SWZ Clausthal – Göttingen
Simulationswissenschaftliches Zentrum

Bericht 2015–2016

Bericht 2015–2016

Simulationswissenschaftliches Zentrum Clausthal-Göttingen

Vorwort

Der vorliegenden Berichtsband 2015-2016 beschreibt die Forschungsaktivitäten des Simulationswissenschaftlichen Zentrums Clausthal-Göttingen (SWZ). Das SWZ wurde am 1.1.2013 von der Technischen Universität Clausthal und Georg-August-Universität Göttingen als Partnern gegründet. In unserem zweiten Bericht stellen wir die aktuellen Forschungsthemen des SWZ vor.

In seinem Positionspapier „**Bedeutung und Weiterentwicklung von Simulation in der Wissenschaft**“[1] vom Juli 2014 hebt der Wissenschaftsrat die Bedeutung der Simulation für den Fortschritt in vielen Wissensgebieten hervor. Um das in dem Positionspapier geforderte Repertoire an Simulationsmethoden an den beiden beteiligten Partneruniversitäten und für den Wissenschaftsstandort Deutschland weiter auszubauen, wird das SWZ in den Jahren 2013 bis 2019 vom Niedersächsischen Ministerium für Wissenschaft und Kultur (MWK) mit insgesamt 4,3 Millionen Euro gefördert. Die Förderung fließt in Forschungsprojekte von Göttinger und Clausthaler Kolleginnen und Kollegen, die sich in einem von externen Gutachtern begleiteten **kompetitiven Verfahren** durchsetzen konnten. Die in diesen Projekten untersuchten Ideen und erzielten Ergebnisse werden in dem vorliegenden Band beschrieben. Sie stammen aus den drei Forschungsfeldern „**Simulation und Optimierung von Netzen**“, „**Simulation von Materialien**“ und „**Verteilte Simulation**“.

Neben den Projekten finanziert das SWZ auch jeweils die Hälfte von **zwei SWZ-Juniorprofessuren**, eine in Göttingen und eine in Clausthal. In Göttingen konnte 2015 Jun.-Prof. Dr. Marcus Baum für diese Stelle gewonnen werden, so dass durch ihn das Gebiet der Simulationswissenschaften in Forschung und Lehre hervorragend vertreten ist. Die Stelle in Clausthal wird voraussichtlich 2017 besetzt.

Um die Forschung am SWZ sichtbar zu machen, wurde neben der Veröffentlichung von Jahresberichten die **Ringvorlesung „Simulationswissenschaften“**[2] eingerichtet, in deren Rahmen im monatlichen Wechsel in Clausthal und in Göttingen Vorträge von externen Gästen gehalten werden. Viele der dort gehaltenen Vorträge wurden aufgezeichnet und können auf der SWZ-Homepage angesehen werden. Des Weiteren ist das SWZ auch durch das in Kooperation mit dem Internationalen Zentrum Clausthal (IZC) organisierte „**International Simulation Science Semester**“[3] in der Lehre an der TU Clausthal vertreten.

Unser Jahrbuch ist in **Deutsch und Englisch** verfasst: Die deutschen Beschreibungen geben einen Überblick über die jeweiligen Projekte und nennen die grundsätzlichen Ideen, Ansätze und Ergebnisse. Die sich anschließenden englischen Beschreibungen sind wissenschaftliche Texte, die detaillierter auf Herleitungen und Methoden eingehen. Wir hoffen, dass die Beschreibungen für Sie interessant sind und wünschen Ihnen: **Viel Spaß beim Blättern und Lesen!**

Prof. Dr. Anita Schöbel
Sprecherin des Vorstands des Simulationswissenschaftlichen Zentrums Clausthal-Göttingen

[1] http://www.wissenschaftsrat.de/download/archiv/4032-14.pdf
[2] http://www.simzentrum.de/lehre/ringvorlesung/
[3] http://www.simzentrum.de/lehre/isss/

Introduction

This report presents the research activities of the Simulation Science Center Clausthal-Göttingen (SWZ) in the years 2015 and 2016. The center was founded in 2013 by the two partners Clausthal University of Technology and Georg-August-University Göttingen. In our second report, we describe the SWZ research projects we are currently working on.

In its position paper "**Bedeutung und Weiterentwicklung von Simulation in der Wissenschaft**"[1] (The importance and development of simulation in science) of July 2014 the German Wissenschaftsrat underlines the importance of the research areas to be examined at the SWZ for the progress in many fields of science. To expand the toolbox of simulation methods at the two partner universities and for research in Germany, the SWZ is supported by the **Niedersächsischen Ministerium für Wissenschaft und Kultur** (MWK) in the years 2013-2019 with a total of 4.3 million euros. The funding is mainly used to support research projects in Göttingen and Clausthal. The applications for the funded projects have been evaluated and accepted by external reviewers in a competitive procedure. The ideas and results of the funded projects are described in this report. They come from the three research areas "**Simulation and optimization of networks**", "**Simulation of Materials**", and "**Distributed Simulation**".

In addition to the projects, SWZ also funds **two SWZ-Juniorprofessorships**, one in Göttingen and one in Clausthal with fifty percent each. In Göttingen, this position was filled in 2015 with Jun.-Prof. Marcus Baum, who excellently represents simulation science in research and teaching. The position in Clausthal is expected to be filled in 2017.

To make the research at SWZ visible, in addition to the publication of the annual reports, the **lecture series "Simulation Sciences"**[2] was set up in which talks are provided by external guests every month, altering between Göttingen and Clausthal. Many of the talks held in this lecture series have been recorded and can be viewed at the homepage of SWZ. Furthermore, SWZ is represented in teaching at TU Clausthal through the "**International Simulation Science Semester**"[3] which is organized in cooperation with the International Center Clausthal (IZC).

This report is written in English and in German. The German texts are short overviews on the projects sketching their basic ideas, approaches, and main results. The English sections contain the more scientific part in which models, results, and methodologies are described in detail. We hope that the descriptions are interesting for you and that you **enjoy reading it.**

Prof. Dr. Anita Schöbel
President of the board of directors of the Simulation Science Center Clausthal-Göttingen

[1] http://www.wissenschaftsrat.de/download/archiv/4032-14.pdf
[2] http://www.simzentrum.de/lehre/ringvorlesung/
[3] http://www.simzentrum.de/lehre/isss/

Inhalt

Table of contents

Simulation und Optimierung von Netzen

7

Die Simulation ist heute eine der wichtigsten und in vielen Fällen einzig praktikablen Techniken zur Analyse und Optimierung von großen Netzen. Telekommunikationsnetze, Verkehrs-/Logistiknetze und Energienetze haben vieles gemeinsam. Die Komplexität der Netze mit ihren vielen parallel existierenden Knoten und den Strömen zwischen den Knoten ist schwer überschaubar und oft mit anderen Techniken als Simulation nicht beherrschbar. Das tatsächliche Verhalten eines solchen Netzes unterscheidet sich oft von dem vermuteten Verhalten. Der Aufbau, Betrieb, Modifikation und Optimierung solcher Netze stellt in der Regel eine Infrastrukturaufgabe dar, die mit erheblichen Kosten verbunden ist. Um hier Fehlentwicklungen zu vermeiden, wird vor einer physischen Installation die Simulation als wichtigstes Hilfsmittel eingesetzt, um die Eigenschaften eines Netzes, das Verhalten, die kritischen Leistungskenngrößen und Parameter in einem frühen Stadium zu ermitteln. Die 2005 in Kraft getretene EU-Verordnung z.B. zu Ausgleichszahlungen an Fluggäste bei Annullierungen oder großen Verspätungen (EG 261/2004) unterstreicht die Notwendigkeit geeigneter Verfahren und Systeme für den praktischen Einsatz.

8

Strukturuntersuchungen zur Entstehung und Fortpflanzung von Verspätungen in Verkehrsnetzen

Michael Kolonko, Anita Schöbel, Jonas Harbering, Fabian Kirchhoff

Verspätungen und verpasste Anschlüsse bei Bahn- und Flugreisen sind ein Ärgernis für die Reisenden und nehmen regelmäßig breiten Raum in der öffentlichen Diskussion ein. Sie verringern die Akzeptanz des jeweiligen Verkehrsmittels und führen bei der Bahn zu einem Ausweichen auf den Individualverkehr. Auf Seiten der Betreiber stören Verspätungen den Betriebsablauf, es entstehen Produktivitätsverluste. Verspätungen bei den zentralen Verkehrsmitteln haben daher gravierende negative Auswirkungen sowohl auf die Mobilität der Gesellschaft wie auch auf die Wirtschaftlichkeit der Verkehrsbetriebe. In dem vorliegenden Projekt wurden daher grundlegende Untersuchungen zur Entstehung und Verbreitung von Verspätungen in Verkehrsnetzen vorgenommen. Gleichzeitig wurden Gegenmaßnahmen zur Stabilisierung des Verkehrs untersucht.

Einzelne Verspätungen entstehen zunächst typischerweise durch externe Störungen wie z.B. besondere Wetterereignisse oder Verzögerungen beim Ein- und Aussteigen. Diese primären Verspätungen (Quellverspätungen) übertragen sich auf weitere Züge, so dass eine einzelne Verspätung zu einer ganzen Kaskade von propagierten Verspätungen (Folgeverspätungen) führen kann. Im Wesentlichen gibt es dabei zwei Übertragungsmechanismen für Verspätungen. Sie können sich fortpflanzen, weil Anschlusszüge auf verspätete Zubringerzüge warten und weil aus Sicherheitsgründen Mindestdistanzen zwischen Zügen eingehalten werden müssen. Zusätzlich kann es noch weitere Übertragungsmechanismen, wie z.B. Fahrzeugumläufe, geben.

Um Verspätungen auf allen Ebenen (vor, während und nach dem Auftreten von Quellverspätungen) zu begegnen, wurden verschiedene Ansätze entwickelt. Zu diesem Zweck sind in Göttingen und Clausthal existierende Softwareprojekte miteinander verglichen und fortgeführt worden (LinTim in Göttingen, HiTT in Clausthal). Sie beschäftigen sich mit der Planung verschiedener Elemente von öffentlichen Verkehrssystemen, und damit unter anderem mit der Analyse der Auswirkungen von Verspätungen.

Um die Unterschiede und Gemeinsamkeiten der existierenden Systeme in Clausthal und in Göttingen vergleichbar zu machen, werden gemeinsame Komponenten der Modellierung in einem Ereignis-Aktivitäts-Netzwerk (EAN) festgehalten. Der wesentliche Unterschied in der weiteren Modellierung besteht in der Darstellung der Verspätungen. Während auf Clausthaler Seite Zufallsvariablen mit dazugehörigen Wahrscheinlichkeitsverteilungen angenommen werden (stochastischer Ansatz), so werden auf Göttinger Seite feste Verspätungswerte entsprechend einer Wahrscheinlichkeitsverteilung festgelegt (Szenarien-basierter Ansatz). Entsprechend unterschiedlicher Art sind die Ergebnisse, die sich für die resultierenden übertragenen Verspätungen ergeben: Verteilungen auf der einen, Szenarien auf der anderen Seite. Ein wesentlicher Teil der gemeinsamen Arbeit besteht darin, diese unterschiedlichen Modellierungsarten gegeneinander zu validieren.

Für diese Validierung werden Verspätungsszenarien für das Szenarien-basierte Modell mit den Verteilungen simuliert, die die Eingangsgrößen für das stochastische Modell bilden. Die resultierenden Verspätungsszenarien werden dann mit den errechneten Verteilungen des stochastischen Modells verglichen, siehe Abbildung 1. Es ergibt sich eine sehr hohe Übereinstimmung. Damit ist im Kern die Gleichwertigkeit beider Herangehensweisen in Bezug auf die Verspätungsmodellierung gezeigt. Je nach konkretem Anwendungsfall kann das Modell, welches die spezifischen Eigenschaften der konkreten Fragestellung besser erfasst,

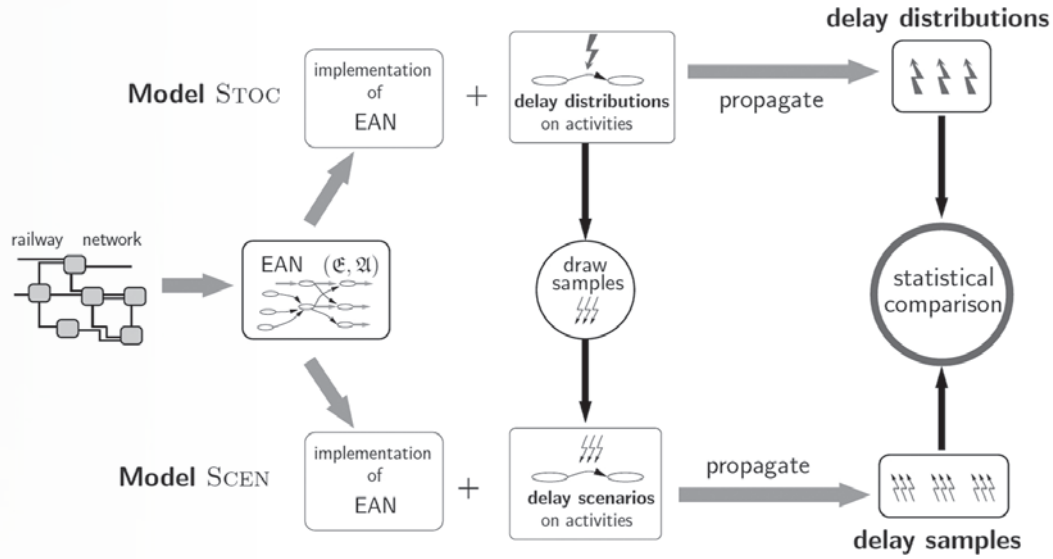


Abbildung 1: Vorgehensweise beim Vergleich des Szenarien-basierten mit dem stochastischen Modell

gewählt werden. Gleichzeitig werden dadurch beide Modelle und Implementierungen validiert, siehe auch [13] für eine detailliertere Beschreibung. Die gemeinsame Modellierung bildet die Grundlage für weitere Untersuchungen zur Stabilisierung des Verkehrssystems unter Verspätungen auf verschiedenen Ebenen.

So wurde in verschiedenen Forschungsarbeiten untersucht, inwieweit sich ein verspätungsresistentes Verkehrssystem erstellen lässt. Ein öffentliches Verkehrssystem basiert dabei wesentlich auf den zugrunde liegenden Linien. Die Linien sind die Wege, die die Fahrzeuge im Netzwerk abfahren. Das Linienplanungsproblem wird üblicherweise in zwei Teilprobleme unterteilt. In dem ersten Teilproblem wird eine Menge von potentiellen Linien vorgeschlagen. Im Rahmen dieses Projektes wurde eine Forschungsarbeit veröffentlicht (siehe [7]), die einen Algorithmus zum Lösen dieses Teilproblems vorschlägt. Der Algorithmus errechnet eine Menge von potentiellen Linien – den Linienpool – dessen Qualitäten anhand verschiedener Parameter kontrolliert werden kann. In dem zweiten Teilproblem werden Frequenzen für alle potentiellen Linien gefunden. In [12] wird ein Modell vorgeschlagen, in dem die Linien so ausgewählt werden, dass möglichst wenige Passagiere umsteigen müssen. Es wird gezeigt, dass dieser Ansatz die Verspätungsübertragung reduziert. Die beiden Arbeiten gehen damit einen ersten Schritt zur integrierten Planung des öffentlichen Verkehrs. Außerdem werden in weiteren Arbeiten bestehende Robustheitskonzepte einander gegenüber gestellt ([8]) und ein neues Robustheitskonzept am Beispiel von Fahrplänen erarbeitet ([9]).

Neben Maßnahmen, die vor dem Eintreten von Verspätungen vorgenommen werden können, gibt es auch Maßnahmen, die ergriffen werden können im Fall von bereits bestehenden Verspätungen. Auch in diesem Bereich sind neue Forschungsarbeiten entstanden. So wird in der Arbeit [1] ein simulationsbasierter Lern-Ansatz vorgeschlagen, der basierend auf der Qualität von vergangenen Warte-Entscheidungen aktuelle Warte-Entscheidungen trifft. Alle bisher erwähnten Arbeiten stellen eine makroskopische Sicht auf das Infrastrukturnetzwerk dar. In weiteren Arbeiten werden die bei Verspätungen an den Bahnhöfen entstehenden Konflikte aus mikroskopischer Perspektive untersucht ([5,6]).

Existieren Verspätungen einmal im System ist es keine leichte Aufgabe, herauszufinden wodurch sie ursprünglich ausgelöst wurden. In [18] wird mittels statistischer Verfahren untersucht, inwieweit sich Verspätungen im Netz der Deutschen Bahn zurückverfolgen lassen. Eines der Verfahren lässt sich auch bei allgemeinen Ursprungsbestimmungen von Ausbreitungsmechanismen auf Netzwerken (insbesondere Krankheiten) anwenden.
Einige der erwähnten Arbeiten sind in den Dissertationen von J. Harbering ([11]) und F. Kirchhoff ([15]) erschienen, die beide im Rahmen dieses Projektes entstanden sind.

Basierend auf initialen Forschungsarbeiten im Rahmen dieses Projektes ist in Göttingen die Forschergruppe FOR 2083 zum Thema „Integrierte Planung im Öffentlichen Verkehr" genehmigt worden.

Referenzen

[1] R. Bauer and A. Schöbel. Rules of thumb -practical online strategies for delay management. Public Transport, 6(1):85-105, 2014.

[5] T. Dollevoet, D. Huisman, L. Kroon, M. Schmidt, and A. Schöbel. Delay management including capacities of stations. Transportation Science, 49(2):185-203, 2015.

[6] T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel. Delay management with rerouting of passengers. Transportation Science, 46(1):74-89, 2012.

[7] P. Gattermann, J. Harbering, A. Schöbel. Line Pool Generation. Public Transport, 1-26, 2016, published online.

[8] M. Goerigk and A. Schöbel. Algorithm engineering in robust optimization. In Algorithm Engineering: Selected Results and Surveys, Editors: L. Kliemann, P. Sanders, LNCS State of the Art, 9220, 2016, in print, http://arxiv.org/abs/1505.04901.

[9] M. Goerigk and A. Schöbel. Recovery-to-optimality: A new two-stage approach to robustness with an application to aperiodic timetabling. Computers and Operations Research, 52: 1-15, 2014.

[11] J. Harbering. Planning a Public Transportation System with a View Towards Passengers' Convenience. PhD thesis, University of Göttingen, Germany. 2016

[12] J. Harbering. Delay Resistant Line Planning with a View Towards Passenger Transfers. Technical report, University of Göttingen, 2014.

[13] J. Harbering, F. Kirchhoff, M. Kolonko, and A. Schöbel. Delay propagation in public transport - stochastic modeling meets scenario approach. Technical report, 2014. zur Veröffentlichung vorgesehen.

[15] F. Kirchhoff. Verspätungsfortpflanzung in Bahnnetzen, Modellierung und Berechnung mit abgeschlossenen Familien von Wahrscheinlichkeitsverteilungen. PhD thesis, University of Technology Claustal, Germany. 2015.

[18] J. Manitz, J. Harbering, M. Schmidt, T. Kneib, A. Schöbel. Source Estimation for Propagation Processes on Complex Networks with an Application to Delays in Public Transportation Systems. Journal of the Royal Statistical Society: Series C, 2016, published online.

Projektdaten

Das Projekt wurde von April 2013 bis März 2016 vom SWZ mit insgesamt 1,75 TV-L E13 Stellen an den Standorten Clausthal und Göttingen gefördert. Beteiligte Wissenschaftler sind:

Prof. Dr. Anita Schöbel,
Arbeitsgruppe Optimierung
Institut für Numerische und Angewandte Mathematik,
Universität Göttingen

Jonas Harbering, M.Sc.
Arbeitsgruppe Optimierung
Institut für Numerische und Angewandte Mathematik,
Universität Göttingen

Prof. Dr. Michael Kolonko,
Arbeitsgruppe Stochastische Optimierung
Institut für Angewandte Stochastik und Operations Research, Technische Universität Clausthal

Dr. Fabian Kirchhoff
Arbeitsgruppe Stochastische Optimierung
Institut für Angewandte Stochastik und Operations Research, Technische Universität Clausthal

Prof. Dr. Thomas Hanschke,
Arbeitsgruppe Stochastische Modelle in den Ingenieur-wissenschaften
Institut für Angewandte Stochastik und Operations Research, Technische Universität Clausthal

12

Sichere Kommunikation in Internet of Things (IoT) Umgebungen

Xiaoming Fu, Dieter Hogrefe, Salke Hartung, Jiachen Chen

Die Gewährleistung eines sicheren Betriebs von vernetzten Geräten stellt insbesondere im Bereich des Internets der Dinge (Internet of Things, IoT) einen schwieriges Problem dar, da die IoT-Geräte meist über eine deutlich geringere Rechenleistung verfügen als normale Netzwerkgeräte. Insbesondere wenn auf Basis von IoT-Geräten in Krisenfällen z.B. nach Naturkatastrophen ein dezentrales Kommunikationsnetz aufgebaut werden soll, ist es wichtig, dass die knappen Ressourcen nicht durch Spam und andere niedrig zu priorisierende Nachrichten blockiert werden sowie dass die Authentizität von Nachrichten mit möglichst wenig Rechenaufwand verifiziert werden kann.

Ein weiterer wichtiger Aspekt in vielen IoT-Netzen stellt die Lokalisierung der einzelnen Geräte, die z.B. Sensoren sein können, dar. Vielfach ist ein Messwert nur in Verbindung mit der Information, an welcher Stelle er erhoben wurde, von Nutzen. Um die einzelnen IoT-Geräte möglichst klein und günstig produzieren zu können sowie eine möglichst lange Betriebsdauer mit einer Batterie zu ermöglichen, kommt vielfach GPS zur Bestimmung der Position nicht in Frage. Daher muss auch hier auf Ansätze, die die Vernetzung der Geräte mit einbeziehen, zurückgegriffen werden: Einige wenige IoT-Knoten kennen per GPS oder durch Installation an einem festen Ort ihre Position. Die anderen Knoten leiten ihre Position im Netz aus diesen Referenzpunkten ab.

Die Schwierigkeit in der Umsetzung solcher Konzepte besteht meist weniger in der Implementierung entsprechender Algorithmen als viel mehr in dem realitätsnahen Testen der Systeme. Eigenschaften wie eine gute **Skalierbarkeit** (von einigen wenigen IoT-Knoten auf viele Hundert Knoten) und eine **Robustheit** des Gesamtsystems gegenüber Störungen (wie z.B. Paketverluste, Reflexionen usw.) können nicht unter Betrachtung nur eines einzelnen IoT-Knotens untersucht werden, sondern nur in vernetzten Verbünden aus vielen IoT-Komponenten. Ein wichtiges Hilfsmittel zur Untersuchung dieser Probleme stellt die Netzwerksimulation dar, die es in einer einfach zu handhabenden und günstigen Weise ermöglicht, virtuell Netzwerke aus tausenden von Knoten aufzubauen und das Verhalten des Systems in der Simulation zu untersuchen. Der Bereich der in diesem Projekt verwendeten Simulationssoftware reicht dabei von klassischen Netzwerksimulatoren bis hin zu komplexen mathematischen Werkzeugen und schließt auch selbst entwickelte Programme mit ein.

Methoden zur Bestimmung von Abständen in drahtlosen Sensornetzwerken

Ein Sensorknoten berechnet in einem drahtlosen Sensornetzwerk seine eigene Position meist in dem er die Abstände zu anderen Knoten, deren Positionen bereits bekannt sind, bestimmt. In einer 2013 durchgeführten Studie wurde untersucht, wie gut sich der Signalstärke-Indikator (RSSI) zur Bestimmung des Abstands zu anderen Netzwerkknoten eignet. Auch wenn der RSSI in den meisten Lokalisierungsalgorithmen eingesetzt, so konnte doch gezeigt werden, dass dieser in vielen Fällen nur eine grobe Abschätzung der Abstände liefert.

Einen verbesserten Ansatz zur Bestimmung des Aufenthaltsortes eines Sensors stellt die **sensorunterstützte Monte Carlo Lokalisierung** (SA-MCL) dar. Die SA-MCL ermöglicht gegenüber klassischen Methoden der Lokalisierung entweder bei gleicher Anzahl an Basisknoten den Lokalisierungsfehler zu reduzieren oder aber bei gleicher Lokalisierungsgüte mit weniger Basisknoten zu arbeiten. Die Idee hinter der SA-MCL besteht darin, zusätzliche Sensorinformationen (wie z.B. die Bewegungsgeschwindigkeit und -richtung) für die Bestimmung des Aufenthaltsortes eins Sensorknotens mit heran zu ziehen. Auf Basis dieser Zusatzinformationen lässt sich der Aufenthaltsort eines Sensorknotens sogar dann noch fortschrei-

ben, wenn kein Kontakt zu einem Basisknoten mehr besetzt.

Ein weiterer untersuchter Ansatz stellt die **pfad-orientierte Monte Carlo Lokalisierung** (PO-MCL) dar. PO-MCL kann immer dann eingesetzt werden, wenn sich die Sensorknoten meist auf statischen Pfaden bewegen. Das Ziel ist auch hier die Reduktion des Lokalisierungsfehlers und die Berücksichtigung von Situationen, in denen keine Basisknoten zum Abgleich der Ortsinformationen zur Verfügung stehen. Bei der Verwendung von PO-MCL werden, so lange eine exakte Lokalisierung mit Hilfe der Basisknoten möglich ist, die Aufenthaltswahrscheinlichkeiten eines Sensorknotens erfasst. Fällt der Kontakt zu den Basisknoten aus, so kann die Bewegung des Sensorknotens über die vorab berechneten Aufenthaltswahrscheinlichkeiten prognostiziert werden.

Methoden zur sicheren Auffindung von Nachbarknoten in einem drahtlosen ad-hoc Netzwerk

Es gibt eine Reihe von Angriffsmöglichkeiten, um die Genauigkeit der Ergebnisse von Lokalisierungsalgorithmen zu verschlechtern. Da für die Bestimmung der eigenen Position der Kontakt zu den Basiskonten von großer Bedeutung ist, zielen viele dieser Angriffsszenarien darauf ab, diesen Kontakt zu verhindern oder falsche Informationen zu den Sensorknoten zu leiten. Eine häufige Angriffsmethode stellen die sogenannten Wurmloch-Angriffe dar, dabei initiiert der Angreifer eine eigene Kommunikationsstrecke; an einem Ende zeichnet er die dort erreichbaren Funksignale auf und spielt sie an der anderen Seite der Kommunikationsstrecke wieder ab. Damit suggeriert er einem Sensorknoten eine falsche Position der Basisknoten.

Um dies zu verhindern, ist es für Sensorknoten wichtig erkennen zu können, ob es sich bei einem vermeintlichen Nachbarknoten um einen tatsächlichen Nachbarn handelt. Hierfür setzen wir das 2013 auf der IEEE International Conference on Communications (ICC) vorgestellte Robust Secure Neighbor Discovery (RSND) ein. RSDN setzt auf einen verteilten Ansatz zur Bestimmung von Inkonsistenzen, die möglicherweise durch einen Angriff hervorgerufen wurden, und um falsche Nachbarknoten zu identifizieren. Da das Verfahren verteilt über alle Sensorknoten arbeitet, ist es nicht für Angriffe, die auf einzelne Punkte abzielen, die das gesamte System kompromittieren können, (Single point of failure, SPF) anfällig. Um dies zu erreichen, bestimmt RSDN in jedem Knoten eine vermutliche lokale Topologie auf Basis der geschätzten Distanzen zu den jeweiligen Nachbarn. Ein einzelner falscher Wert erhält dadurch nur ein geringes Gewicht und hat so nur einen geringen Einfluss auf das Gesamtergebnis. Die mathematische Basis für diese Erkennung von Abweichungen in den Abständen (zur Erkennung von falschen Nachbarn) stellt die multidimensionale Skalierung (MDS) dar.

Projektdaten

Das Projekt wurde von April 2013 bis März 2016 vom SWZ mit insgesamt 0,75 TV-L E13 Stellen am Standort Göttingen gefördert. Beteiligte Wissenschaftler sind:

Prof. Dr. Dieter Hogrefe
Arbeitsgruppe Telematik
Institut für Informatik
Universität Göttingen

Dr. Mayutan Arumaithurai
Arbeitsgruppe Computer Networks Group
Institut für Informatik,
Universität Göttingen

Prof. Dr. Xiaoming Fu
Arbeitsgruppe Computer Networks Group
Institut für Informatik,
Universität Göttingen

Dr. Jiachen Chen
Arbeitsgruppe Computer Networks Group
Institut für Informatik,
Universität Göttingen

Salke Hartung, M. Sc.
Arbeitsgruppe Telematik
Institut für Informatik,
Universität Göttingen

Dr. Somayeh Taheri
Arbeitsgruppe Telematik
Institut für Informatik,
Universität Göttingen

Dekomposition von offenen Warteschlangennetzen mit „Batch-Processing“

Wiebke Klünder, Thomas Hanschke, Horst Zisgen

Die Bedeutung von Produktionsplanungs- und -steuerungswerkzeugen wächst in der industriellen Fertigung seit Jahrzehnten kontinuierlich an. Als Ursachen hierfür sind vor allem die stetig komplexer werdenden Produktionsprozesse sowie gleichzeitig die Forderung nach einer Verkürzung der Fertigungszyklen zu nennen. Daher besteht die Notwendigkeit, Fertigungsprozesse bereits von Anfang an zu optimieren, da nachträgliche Korrekturen an einem Produktionsprozess meist nur schwer umsetzbar sind.

Zur Modellierung von Fertigungsprozessen werden meist **Nichtproduktformnetze** (siehe Abbildung 1) eingesetzt. Diese Netze besitzen jedoch eine Reihe von Eigenschaften, die eine analytische Berechnung der wichtigen Kenngrößen wie der mittleren Warteschlangenlänge an einer Maschine verhindern. Da diese Kenngrößen z.B. für die Planung der Pufferkapazitäten notwendig sind, muss nach alternativen Lösungsmethoden gesucht werden. Eine Möglichkeit zur näherungsweisen Lösung von Nichtproduktformnetzen stellt die **Monte-Carlo Simulation** dar, die jedoch je nach Größe der Produktionsanlage rechenzeitintensiv sein kann und keine funktionalen Zusammenhänge zwischen den Parametern und den Kennzahlen des Modells liefert. Eine wichtige analytische Lösungsstrategie ist die **Dekompositionsmethode**. Diese nutzt approximative Berechnungen, um die gesuchten Kennzahlen eines Modells näherungsweise zu bestimmen und stellt außerdem funktionale Zusammenhänge zwischen Parametern und Kenngrößen her, die für ein generelles Modellverständnis meist sehr hilfreich sind.

Die in [1] vorgestellte Dekompositionsmethode erlaubt es, ein Nichtproduktformnetz in einzelne Bausteine zu zerlegen und diese dann adäquat durch „traffic equations“ zu verbinden. Jeder Baustein repräsentiert ein Warteschlangensystem, das aus Maschinen und einem Wartebereich besteht. Die Maschinen können dabei optional Lose vom Umfang größer als 1 verarbeiten, d.h. eine **Batch-Verarbeitung** kann in der Methode abgebildet werden. Die Bedien- und Ankunftszeiten der Lose werden durch stochastische Prozesse modelliert, die wie die Losgrößen zwischen den Bausteinen variieren können. Die Dekompositionsmethode beinhaltet das Aufstellen und Lösen zweier linearer Gleichungssysteme, deren Lösungen Werte zur näherungsweisen Berechnung der Kenngrößen mittels der Approximationsformel von Allen und Cunneen [2] liefern. Das erste lineare Gleichungssystem behandelt die Ströme innerhalb des Produktformnetzes zwischen den Bausteinen

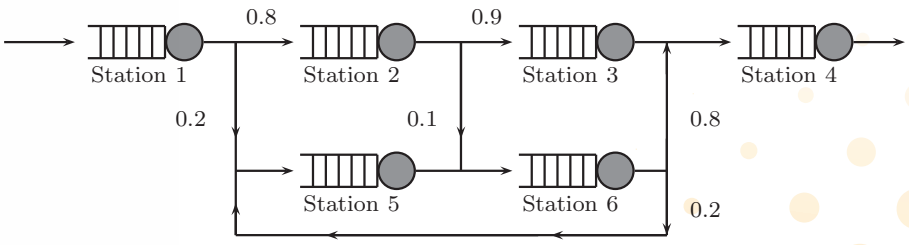


Abbildung 1: Beispiel eines Nichtproduktformnetzes

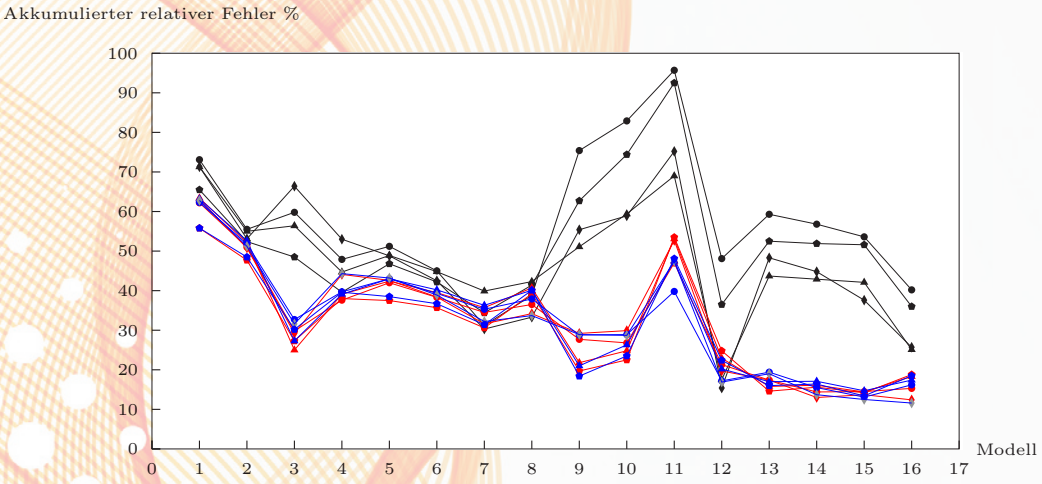


Abbildung 2: Abweichung zwischen Simulations- und Approximationsergebnissen

(„traffic equations"). Die Lösungen fließen in das zweite lineare Gleichungssystem ein, welches die Fertigungsströme zu einem Baustein und innerhalb des Bausteins beschreibt. Zur Beschreibung dieser Ströme wurden in [1] Approximationen von Pujolle und Ai [3] verwendet.

Batch-Verarbeitung

Ziel dieses Projektes ist es, die in [1] beschriebene Dekomposition um weitere Approximationen zur Beschreibung der Ströme zu einem Baustein und innerhalb eines Bausteins zu erweitern. Dazu wurden die Approximationen von Chylla [4], Whitt [5] und Kühn [6], die keine Losgrößen-Verarbeitung zulassen, um diese Eigenschaft ergänzt. Die Dekomposition wurde durch die Verwendung aller zwölf möglichen Kombinationen der Approximationen für die Ströme auf Robustheit untersucht. Dabei wurde die Simulationsstudie aus [1] verwendet, die 16 Modelle umfasst. Es wurden jeweils die Abweichungen der Näherungen der Kennzahlen zu den Ergebnissen der Simulationen bestimmt.

Es konnte beobachtet werden, dass die Approximationen von Pujolle/Ai (blau) und Kühn (rot) für die Ströme in einen Baustein des Nichtproduktformnetzes gute Näherungen liefern, während die Approximation von Chylla (schwarz) teilweise hohe Abweichungen besitzt. Die Ströme innerhalb eines Bausteins werden hingegen von allen vorliegenden Approximationen gut umgesetzt. Problematisch sind signifikant ausgeprägte Losgrößenwechsel zwischen den Bausteinen, die unabhängig von der Wahl der Approximationen zu hohen Abweichungen führen können. Der Grund dafür liegt darin, dass in den Approximationen angenommen wird, dass es sich bei den Strömen um Erneuerungsprozesse handelt. Wird diese Annahme verletzt, wie im Fall von Losgrößenwechseln in bestimmten Situationen, sinkt die Approximationsgüte.

Produktklassen

Da in einer Fertigung häufig mehr als ein Produkt eines Typs hergestellt wird, stellt ein weiteres Ziel die Verallgemeinerung der Dekomposition aus [1] dar. Die Verallgemeinerung besteht in der Betrachtung mehrerer Produktklassen, die unterschiedliche Losgrößen und auch unterschiedliche Ankunfts- und Bedienzeiten an den Maschinen besitzen. Da beobachtet werden konnte, dass die Approximationen von Pujolle und Ai annehmbare Näherungen liefern, wurde dieser Ansatz auf den Mehr-Produktfall der Dekomposition erweitert. Die durchgeführte Simulationsstudie nutzt zwei Referenzmodelle. Das erste Modell bildet eine Fertigungsstraße mit drei Produktklassen und das zweite Modell ein Netz mit zwei Produktklassen. Die aus den Referenzmodellen gewonnenen Simulationsmodelle lassen sich in sechs unterschiedliche Kategorien unterteilen, die eine Untersuchung

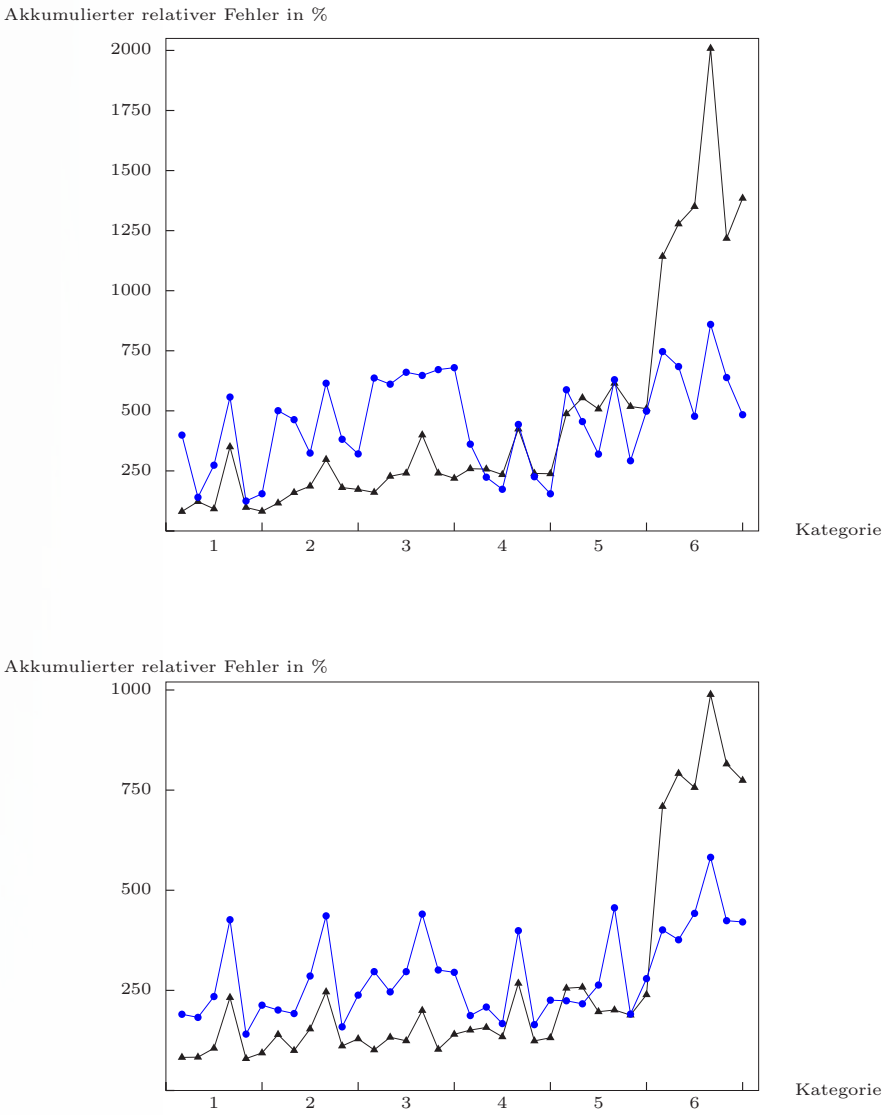


Abbildung 3: Abweichung zwischen Simulations- und Approximationsergebnissen im Mehrproduktfall

der Nützlichkeit der Verallgemeinerung in verschiedenen Situationen ermöglichen. Die ersten drei Kategorien beschreiben Modelle, bei denen die Produktklassen jeweils denselben Wert für die Losgröße an einem Baustein aufweisen, aber sich die Bedienzeiten der Werkstücke der Produktklassen an einem Baustein unterscheiden. Bei den letzten drei Kategorien variieren zusätzlich die Losgrößen der Produktklassen an einem Baustein. Die Kategorien 1 und 4 besitzen die Eigenschaft, dass die Losgrößen im Verlauf der Fertigung von Station zu Station zunehmen und am Austrittspunkt des Nichtproduktformnetzes ihr Maximum erreichen. Die Kategorien 2 und 3 bzw. 5 und 6 beinhalten dagegen Modelle mit schwankenden Losgrößen im Fertigungsverlauf. Als Vergleich, um den Mehrwert der Verallgemeinerung zu bestimmen, wurde die Dekomposition aus [1] genutzt. Die Umsetzung erfolgte durch gemittelte Eingabewerte und das Aufteilen der Approximationsergebnisse auf die Produktklassen mit Hilfe von relativen Besuchshäufigkeiten.

Abbildung 3 zeigt die akkumulierten relativen Abweichungen zwischen den Simulations- und Approximationswerten. Die Verallgemeinerung (schwarz) ist der etablierten Dekomposition (blau) aus [1] in den ersten vier Kategorien durch ihre Produktklassenbetrachtung überlegen und liefert Näherungen, die sich für eine Prä-Evaluierung eines Modells eignen. Zwar besitzt die Dekomposition aus [1] mit der einfacheren Modellierung bessere Approximationsgüten für die Kategorien 5 und 6. Jedoch sind diese generell sehr niedrig und eignen sich nicht für eine Prä-Evaluierung. Die Schwankungen der Größe der Lose innerhalb des Fertigungsverlaufs führen zu Fehlern, die in der Verletzung der Annahme, dass die Ströme Erneuerungsprozessen bilden, begründet ist. Die Dekomposition aus [1] ist der Verallgemeinerung in den Kategorien 5 und 6 überlegen, da eine Produktklassenbetrachtung im Anschluss des Verfahrens vorgenommen und der Fehler nur einmalig innerhalb der Methode verursacht wird. Die Dekomposition mit Produktklassenbetrachtung verursacht den Fehler mehrfach innerhalb des Verfahrens, wodurch sich der Fehler stärker aufbauen kann.

Zusammenfassend sind die genannten Dekompositionsmethode als Prognose- und Evaluierungswerkzeug geeignet, wenn das zu untersuchende Nichtproduktformnetz keine starken Schwankungen der Losgrößen besitzt.

Literatur

[1] Th. Hanschke and H. Zisgen, Queueing networks with batch service, European Journal of Industrial Engineering, 5(3):313-326, 2011

[2] Th. Hanschke, Approximations for the mean queue length of the $GI^X/GI^{(b,b)}/c$ queue, Operations Research Letters, 34(2): 205-213, 2006

[3] G. Pujolle and W. Ai, A solution for multiserver and multicall open queueing networks, INFOR, 24(3):221-230, 1986

[4] Peter Chylla, Zur Modellierung und approximativen Leistungsanalyse von Vielteilnehmer-Rechensystemen, Dissertation, TU München, 1986

[5] Ward Whitt, The queueing network analyzer, Bell system technical Journal, 62(9): 2779-2815, 1983

[6] Paul J. Kühn, Approximate analysis of general queueing networks by decomposition, IEEE Transaction on Communications, 27(1):113-126, 1979

Projektdaten

Das Projekt wird seit April 2014 vom SWZ mit insgesamt 0,5 TV-L E13 Stellen am Standort Clausthal gefördert. Beteiligte Wissenschaftler sind:

Prof. Dr. Thomas Hanschke
Arbeitsgruppe Stochastische Modelle in den Ingenieurwissenschaften
Institut für Angewandte Stochastik und Operations Research, Technische Universität Clausthal

Dipl.-Wirt.-Inf. Wiebke Klünder
Arbeitsgruppe Stochastische Modelle in den Ingenieurwissenschaften
Institut für Angewandte Stochastik und Operations Research, Technische Universität Clausthal

Simulation unsicherer Optimierungsprobleme mit Anwendung in der Fahrplangestaltung und der Maschinenbelegung

Martin Dahmen, Stephan Westphal, Anita Schöbel

In der mathematischen Optimierung beschäftigt man sich mit Lösungsverfahren von Optimierungsproblemen, die hinsichtlich einer gegebenen Fragestellung oder Zielfunktion ein bestmögliches Ergebnis liefern sollen. Dazu gehört beispielsweise die Bestimmung eines Produktionsplans mit minimalen Kosten oder eines Busfahrplanes, der möglichst geringe Verspätungen verspricht. Viele dieser Fragestellungen weisen zudem Unsicherheiten in den Eingabedaten auf, was dazu führt, dass strategische Entscheidungen getroffen werden müssen, bevor alle relevanten Daten bekannt sind. Ein Busfahrplan muss beispielsweise bereits vor Antritt der Fahrt feststehen, und auch in der Fließfertigung von Automobilproduzenten werden Produktionszyklen und Lagerkapazitäten lange im Voraus geplant, bevor die Aufträge der Kunden eingetroffen sind. Das Ziel bei der Optimierung unter Unsicherheit ist es, diese Unsicherheiten bereits während des Prozesses zur Entwicklung einer Lösungsstrategie zu berücksichtigen.

Es gibt verschiedene Konzepte, die sich mit solchen Optimierungsproblemen unter Unsicherheiten beschäftigen. Der wesentliche Unterschied dieser Konzepte liegt dabei in der Modellierung der Unsicherheiten. Ein bewährtes Konzept aus der robusten Optimierung sucht nach einer Lösung, die für jedes mögliche eintretende Szenario zulässig ist und dabei das Szenario mit den höchsten Kosten in der Zielfunktion minimiert [1]. Man bezeichnet diese Absicherung im schlechtesten Fall allgemein auch als konservative „worst-case“ Analyse. Auf diese Art und Weise können Optimierungsprobleme, wie zum Beispiel die Produktionsprogrammplanung, in einer robusten Version formuliert werden, bei der Modellparameter – wie die Nachfrage oder die Produktionszeiten – unsicher sind [6]. Im Falle der stochastischen oder auch robusten stochastischen Optimierung können diese Unsicherheiten, wie z.B. die Nachfragemengen anhand von Vergangenheitsdaten und Prognosen, durch Zufallsvariablen ausgedrückt werden, um mithilfe der Wahrscheinlichkeitsverteilungen die erwarteten einzutretenden Szenarien zu optimieren (z.B. die Minimierung der Kosten im Erwartungswert). In beiden Fällen muss eine Lösung entwickelt werden, die nach der Realisierung eines Szenarios möglichst gute Ergebnisse liefert. In der Online-Optimierung werden die Unsicherheiten durch Anfragesequenzen modelliert und nicht gleichzeitig, sondern nacheinander aufgedeckt. Nach jeder Anfrage muss der Online-Algorithmus geeignet reagieren, was wegen der Unbekanntheit der noch folgenden Anfragen dazu führen kann, dass so Entscheidungen getroffen werden, die sich durch nachfolgende Nachfragen als suboptimal herausstellen. Die Realisierung eines Szenarios entspricht einer vollständig aufgedeckten Eingabesequenz. Die Online-Optimierung zählt zu den konservativen Konzepten, bei denen immer nur der „worst-case“, also die Eingabesequenz mit den für einen bestimmte Algorithmus höchsten Kosten betrachtet wird. Gesucht ist ein Algorithmus, der für gegebene Eingabesequenzen Lösungen berechnet und den Quotienten der Zielfunktionswerte aus der Online- und der Offline-Lösung minimiert. Die Offline-Lösung ist dabei die optimale Lösung für das Problem, bei der alle Parameter und Unsicherheiten bereits von Beginn an bekannt sind.

Die große Herausforderung in der konservativen „worst-case“ Analyse entsteht durch die Vielzahl an Strategien und Szenarien, die für ein endgültiges Resultat untersucht werden müssen. In der Regel ändert sich das „worst-case“ Szenario für ein Problem durch die Wahl einer anderen Strategie bzw. eines anderen Algorithmus. Dadurch werden anspruchsvolle praktische Optimierungs-

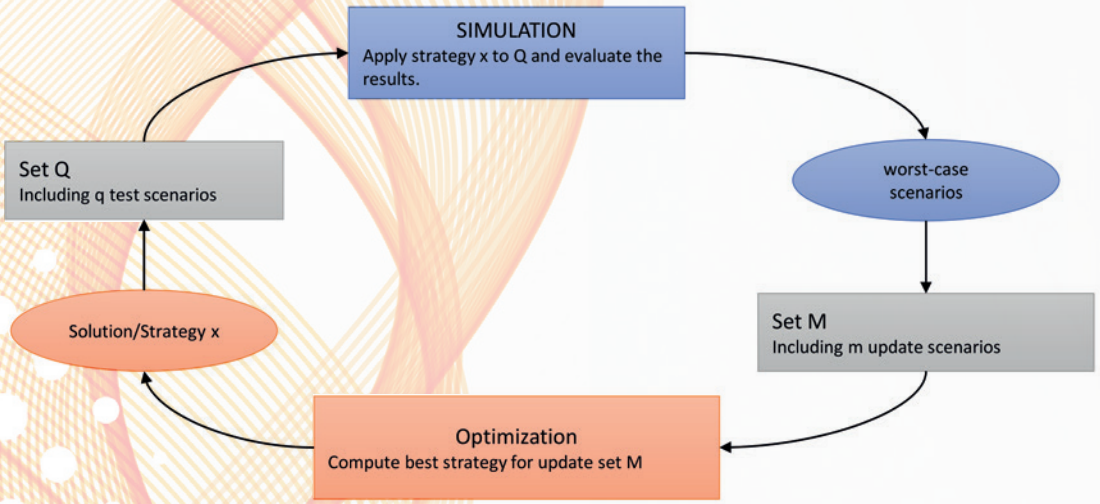


Abbildung 1: Der simulationsbasierte Optimierungsyzklus

probleme, wie Produktionsnetzwerke mit mehreren tausend Varianten und Produktionen am Tag schnell zu komplex, um sie allein mit theoretischen Mitteln zu analysieren. Im Rahmen dieses Projektes wird daher ein integrativer Simulations- und Optimierungsansatz vorgestellt und untersucht, der die fortschrittlichen theoretischen Entwicklungen der mathematischen Optimierung mit den praxisorientierten Analysemethoden und Techniken der Simulation verbindet.

Der simulationsbasierte Optimierungsansatz
Das grundlegende Konzept eines simulationsbasierten Optimierungsansatzes liegt in einem Wechselspiel zwischen einem Optimierungsprozess und einer Simulation. Dabei sollen die Ergebnisse der Optimierung als Input für die Simulation dienen und umgekehrt. Im Fokus steht dabei der iterative simulationsbasierte Optimierungszyklus aus Abbildung 1. Abwechselnd wird eine geeignete Strategie auf Basis von Inputdaten entwickelt (Optimierung) und anschließend im Rahmen einer Simulation getestet, bzw. bewertet. Hierbei wird versucht die Szenarien zu finden, für die die Strategie sehr schlecht geeignet ist (Simulation), um anschließend eine Verbesserung vornehmen zu können. Für ähnliche Fragestellungen robuster Optimierungsprobleme sind bereits Konvergenzresultate bekannt [4]. Dieser iterative Prozess ermöglicht, dass nicht von Beginn an alle denkbaren Szenarien in der Optimierung oder der Simulation berücksichtigt werden müssen, sondern nur ausgewählte Szenarien in den Zyklus aufgenommen werden.

Das beruht auf der Annahme, dass bereits ein Bruchteil aller Daten und Informationen ausreicht, um auch bei komplexen Fragestellungen schon qualitative Aussagen über die Güte ihrer Strategien zu machen. Extremresultate in der Auswertung der Strategien werden häufig nur bei sehr speziellen und konstruierten Szenarien beobachtet, wie z.B. die „worst-case" Sequenz des GRAHAM-Algorithmus aus Abbildung 2, ein Greedy-Verfahren für das Online-Scheduling auf identischen Maschinen [2].

Der Simulationszyklus startet mit einer Menge M, bestehend aus endlich vielen (zufällig) gewählten Szenarien für das gegebene Optimierungsproblem. Im Optimierungsschritt wird auf Grundlage dieser Menge eine optimale Strategie x, also ein Algorithmus oder eine Funktion, die angewendet auf ein Szenario eine Lösung für das Problem berechnet, bestimmt. Im Falle der Online-Optimierung werden hier Probleme betrachtet, bei denen sich die Menge der Algorithmen parametrisieren lässt. Im Optimierungsschritt wird ein

geeigneter Parameter berechnet und mithilfe der gegebenen Update-Menge M optimiert. Anschließend wird die neue Strategie bzw. der neue Parameter durch Simulation auf Testszenarien untersucht. Dazu wird wieder (zufällig) eine Menge Q von Testszenarien erstellt und die Auswirkungen von x auf diese Szenarien werden berechnet. Die Auswertung erfolgt dabei anhand der gewünschten Zielfunktion. So wären beispielsweise die Produktionskosten ein geeignetes Mittel zur Bewertung eines Algorithmus für die Produktionsplanung. Verschiedene Robustheitskonzepte und auch die Online-Optimierung verfolgen dabei den Ansatz, die Kosten der Optimierungslösungen in Relation zu den Kosten einer bestmöglichen Strategie ohne Unsicherheiten zu stellen. Es wird also immer im Nachhinein berechnet, wie der optimale Produktionsplan ausgesehen hätte, um die Lösung unter Unsicherheit damit zu vergleichen. Im Rahmen der Simulation sollen möglichst viele Szenarien mit besonders großem Abstand (bzgl. der Zielfunktion) zwischen der Optimallösung und der Lösung unter Unsicherheiten ermittelt werden. Das sind die sogenannten „bösartigen" Szenarien. Anhand der Update-Menge M und dieser neuen Szenarien beginnt der zyklische Prozess mit der Verbesserung der Strategie x dann von neuem. Im optimalen Fall wird in jedem Optimierungsschritt eine Strategie x bestimmt, sodass das Maximum der Auswirkungen von x auf die Menge aller Szenarien minimiert wird. Umgekehrt wird für eine bestmögliche Analyse der Strategie in der Simulation ein Szenario bestimmt, so dass das Minimum über die Auswirkungen aller Strategien auf dieses Szenario maximiert wird. Letzteres sind exakt die Szenarien, die die Menge der „worst-case" Szenarien für die Strategie x bilden.

Anwendung des simulationsbasierten Optimierungsansatzes auf Online Probleme

Die Online-Optimierung wird häufig als Spiel betrachtet, bei dem der Optimierer, oder auch Online-Spieler, anhand einer unbekannten Eingabesequenz optimale bzw. gute Entscheidungen treffen muss. Ihm gegenüber steht der Offline-Gegner. Er kennt die komplette Eingabesequenz von Anfang an, bzw. kann sie anhand der Entscheidungen des Online-Spielers zu seinen Vorteilen verändern. Den Faktor, der am Ende den Unterschied zwischen den Zielfunktionen der Online- und der Optimallösung ausmacht, nennt man die Kompetitivität der Online-Lösung. Ist dieser Faktor unbeschränkt, so nennt man den zugehörigen Online-Algorithmus nicht kompetitiv. Gesucht sind folglich untere und obere Schranken für die Kompetitivität eines Algorithmus bezogen auf ein Online-Problem. Mithilfe einer unteren Schranke lässt sich beweisen, dass es keinen Online-Algorithmus gibt, der näher an die optimale Offline-Lösung herankommen kann. Umgekehrt definiert jeder zulässige Online-Algorithmus durch seine Kompetitivität eine obere Schranke, und derjenige mit der geringsten Kompetitivität gilt als der beste Online-Algorithmus.

Im Rahmen des Projektes wird das Spiel des Online-Spielers und Offline-Gegners durch den Zyklus aus Abbildung 1 simuliert. Der Optimierungsschritt repräsentiert den Zug des

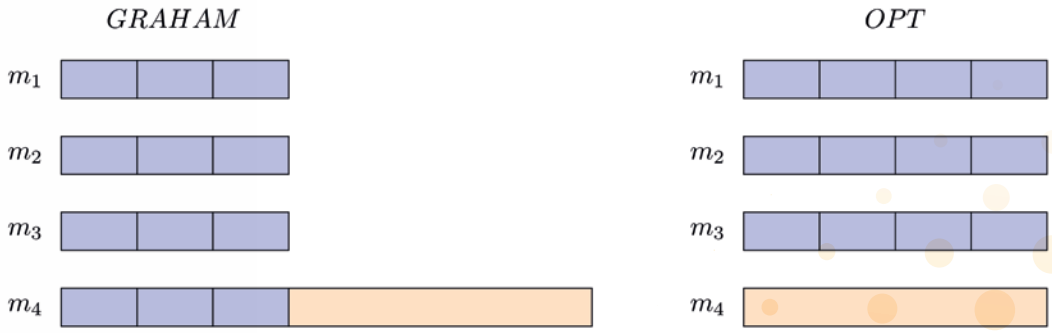


Abbildung 2: Die Graham-Heuristik im Vergleich zur Optimallösung einer „worst-case" Sequenz.

Abbildung 3: Die Zielfunktionen des Online-Spielers und des Offline-Gegners.

Online-Spielers und damit die Wahl einer Online-Strategie. Anschließend sucht der Offline-Gegner im Simulationsschritt die Sequenz, mit der er dem Online-Spieler am meisten schadet. Abbildung 3 zeigt die Zielfunktionen der Spieler. Im Falle des Online-Spielers spricht man auch von einem Minimax-Problem.

Anwendung in der Maschinenbelegung

Eine der meist erforschten Fragestellungen des Operations Research ist das sogenannte Scheduling. Dabei beschäftigt man sich mit der Frage, wie eine Menge von Aufträgen auf eine gegebene Menge von Maschinen verteilt und wann deren Bearbeitung jeweils beginnen soll, so dass eine Reihe von Nebenbedingungen eingehalten werden. Dabei können sowohl die Aufträge als auch die Maschinen unterschiedliche Eigenschaften haben. Die Maschinen haben dabei z.B. unterschiedliche Geschwindigkeiten, es gibt Reihenfolgebeziehungen zwischen den Aufträgen, Zeitfenster in denen die Aufträge bearbeitet werden müssen oder Einschränkungen bzgl. der Zuweisungsmöglichkeiten der Aufträge zu den Maschinen. Im untersuchten Fall kommt hier die Online-Charakteristik zum Tragen, bei der die Aufträge nacheinander eintreffen und bei jedem Eintreffen sofort entschieden werden muss, auf welcher Maschine der jeweilige Auftrag bearbeitet werden soll. Durch die Verteilung der Aufträge auf die Maschinen entsteht ein Belegungsplan, der im Folgenden auch Profil genannt wird.

Das Konzept zur simulationsbasierten Optimierung wurde zunächst auf die Online-Fragestellung zur Minimierung des sogenannten Makespan auf identischen Maschinen angewendet. Dabei weist jeder der nacheinander bekannt werdenden Jobs eine bestimmte Bearbeitungszeit auf. Ein Online-Algorithmus muss nun entscheiden, auf welcher der gegebenen Maschinen dieser Job bearbeitet werden soll, so dass die maximale Auslastung, der sogenannte Makespan, der einzelnen Maschinen minimiert wird.

Eine bewährte Strategie zur Verteilung der Aufträge ist die Greedy-Heuristik von Graham [2], die neue Jobs immer auf die Maschine mit der zu diesem Zeitpunkt geringsten Last verteilt. In einigen Fällen kann es aber sinnvoll sein, dass man die freie Kapazität auf den weniger ausgelasteten Maschinen für besonders große Aufträge freihält, statt eine gleichmäßige Verteilung von Jobs auf allen Maschinen anzustreben. Die dazu gewählte Parametrisierung der Algorithmen basiert auf dieser Idee und entscheidet für jeden ankommenden Job, ob er zu den kurzen oder langen Jobs gehört und damit auf einer Maschine mit großer oder geringer Last bearbeitet werden sollte. Verschiedene Konzepte zur Anpassung des Parameters in der Simulation mit den Testsequenzen wurden hierbei verwendet.

Die naheliegende Methode ist die Berechnung des optimalen Parameters basierend auf der Update-Menge M. Eine der größten Schwierigkeiten ist dabei aber die Skalierbarkeit des Verfahrens. Für das Schedulingproblem gilt, dass sowohl die Bestimmung der optimalen Strategie x für eine gegebene Szenarienmenge M, als auch die exakte Berechnung der optimalen Offline-Lösung bereits auf zwei Maschinen NP-schwer ist. In diesem Fall muss man sich in der Praxis oft mit einer Approximation zufriedengeben, die einen Parameter möglichst nah am Optimum bestimmt, dabei aber sehr schnell und effizient berechnet werden kann. Eine weitere Methode ist die Annäherung an einen „optimalen" Parameter durch Verbesserungsschritte wie z.B. lokale Suche. Der Vorteil ist die geringe Laufzeit der Iterationen im Simulationszyklus. Allerdings werden in der Regel durch die

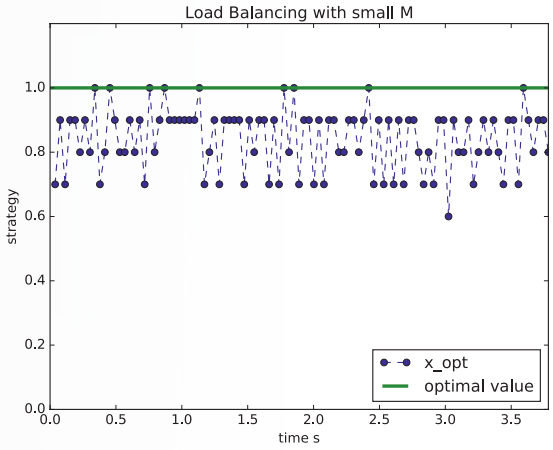


Abbildung 4: Das Konvergenzverhalten des Simulationszyklus. M wird hier in jeder Iteration vollständig überschrieben.

langsame Verbesserung deutlich mehr Iterationen benötigt, bevor ein erkennbares Ergebnis zur Konvergenz des Verfahrens vorliegt. Es werden auch leichter lokale Optima erreicht, was allerdings von der Wahl der Update-Menge M abhängt, die entscheidenden Einfluss auf Konvergenz oder Nicht-Konvergenz des Verfahrens haben kann. So kann eine Update-Menge M, die nur aus den letzten „worst-case" Sequenzen besteht, zu alternierenden Parameterwerten führen, die zwischen zwei oder mehreren lokalen Optimalwerten hin und her springen (Abbildung 4).

Besser ist daher ein stetiges Wachstum der Update-Menge, was sich aber negativ auf die Performance des Zyklus auswirkt, und damit erneut die Wahl von wenigen aber möglichst einflussreichen Update-Sequenzen motiviert. Es bleibt zu zeigen, welche und wie viele dieser Szenarien für eine qualitative Aussage über das Problem ausreichend sind. Die Konzepte wurden implementiert und auf das Schedulingproblem angewendet. Die Jobsequenzen wurden dabei zufällig, aber mit unterschiedlichen Eigenschaften und Restriktionen erstellt.

Verbesserungsverfahren

In dem bisher verwendeten Modell wurde die Menge der Testszenarien Q zufällig und ohne Einsicht in die bisherigen Ergebnisse des Optimierungsprozesses erstellt. Insbesondere in der Online-Optimierung ist das gesuchte „worst-case" Szenario aber häufig eine sehr spezielle und besonders bösartig konstruierte Sequenz, die durch Zufall nur schwer gefunden werden kann (vgl. Graham [2] und [9]). Für eine Verbesserung des Konvergenzprozesses der Verfahren ist daher auch eine Verbesserung zur Bestimmung der Test-Szenarien erforderlich. Dazu wurde ein analytischer und ein genetischer Ansatz entwickelt, die auf spezielle Problemklassen angewendet werden sollen.

Die erste Idee basiert auf Linearer Programmierung zur Maximierung der „worst-case" Sequenzen im Simulationsschritt. Die gewählten Strategien lassen sich in den Nebenbedingungen des linearen Programms modellieren, und anhand einer geeigneten Zielfunktion kann die Sequenz mit möglichst großem Abstand zwischen Online- und Offline-Lösung bestimmt werden. Allerdings handelt es sich hierbei zunächst um ein ganzzahliges lineares Programm, das durch die Vielzahl an Algorithmen exponentiell viele Nebenbedingungen hat. Da zu diesem Zeitpunkt bereits eine Online-Strategie gewählt wurde, ist nur noch eine passende Sequenz und die dazugehörige beste Strategie für das Optimum, also den Offline-Gegner, zu finden. Zur Vereinfachung wurde der Raum der Sequenzen entlang der Profile der Online-Lösung und der bestmöglichen Profile

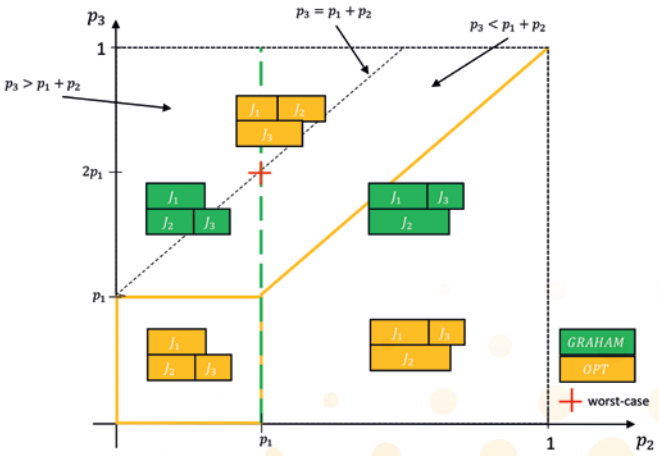


Abbildung 5: Die Scheduling-Profile der Optimallösung und der Graham-Heuristik zur Minimierung des Makespan von 3 Jobs auf 2 Maschinen. Die Achsen geben die Bearbeitungszeiten des zweiten und dritten Jobs in Abhängigkeit des ersten an.

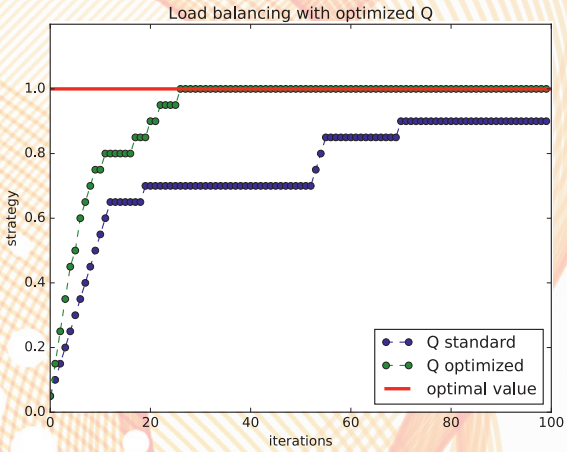


Abbildung 6: Das Verfahren zur Sequenzoptimierung im Vergleich zu vorher.

partitioniert (siehe Abbildung 5). Innerhalb jeder Partition entsteht dadurch ein fraktionales lineares Programm mit einer quasikonvexen Zielfunktion, was dazu führt, dass sich die „worst-case" Sequenzen innerhalb einzelner Partitionen durch Lösungsverfahren von linearen Programmen leicht bestimmen lassen. Dadurch ist zwar noch kein globales Maximum für die Menge aller Sequenzen gefunden, aber untere Schranken für die Kompetitivität einzelner Algorithmen lassen sich schneller berechnen. Das wiederum reduziert die Anzahl der Algorithmen und Szenarien, die im weiteren Verlauf des Simulationszyklus betrachtet werden müssen. Der Konvergenzprozess ist in Abbildung 6 zu sehen. Ebenso der Vergleich zu den bisherigen Ergebnissen. Bisher konnte dieses Verfahren allerdings nur auf spezielle Problemklassen aus dem Bereich des Scheduling angewendet werden und auch das Partitionieren des Szenarienraumes ist sehr aufwändig.

Genetische Verfahren

Die zweite Verbesserung verfolgt einen genetischen Ansatz, der möglichst allgemein formuliert wurde und sich auf fast alle Problemklassen anwenden lassen soll. Die Grundidee, dass Testszenarien durch Randomisierung erzeugt werden, wurde aus dem Optimierungszyklus im weitesten Sinne übernommen. Die Sequenzen sollen dabei aber nicht komplett zufällig und unabhängig von der Update-Menge M gewählt werden, sondern die bereits als bösartig identifizierten Szenarien mithilfe genetischer Methoden wiederverwendet werden. Genetische Algorithmen werden schon seit langer Zeit zur Lösung komplexer Fragestellungen und als heuristisches Verfahren für Optimierungsprobleme eingesetzt [7].

Unter speziellen Konvergenzannahmen lassen sich genetische Verfahren auch auf Minimax-Probleme anwenden [8]. Im Rahmen des Projektes sollen nun speziell die Szenarien für die Menge Q aus dem Zyklus bestimmt werden. Das Verfahren konzentriert sich dabei auf Strukturen von „worst-case" Sequenzen und erzeugt neue, noch „schlimmere" Sequenzen, anhand dieser Merkmale. Dazu zählen unter anderem Extremwerte im Szenarienraum, aber auch kritische „Pattern" innerhalb der Sequenzen. Eine Wiederholung von immer wiederkehrenden Zufallssequenzen soll dadurch größtenteils vermieden werden. Verschiedene Operatoren zur „Selektion",

Genetic Operators	
Random	Create a set of (uniformly distributed) random sequences
Crossover	Create a new sequence as a combination of several previous sequences e.g. σ1=[B, A, D, F, A, B, D, E, D], σ2=[A, F, E, D, F, F, B, A] → σ=[A, A, E, F, A, B, F, B, A]
Mutation	Exchange some genes of a sequence with random elements e.g. σ1=[B, D, F, D, A, A, E, C, C] → σ=[B, D, C, D, A, A, B, C, C]
Double	Create a new sequence by duplication e.g. σ1=[A, F, D, E, C] → σ=[A, F, D, E, C, A, F, D, E, C]
Cut	Reuse certain subsequences e.g. σ1=[D, C, B, A, E, E, A, B] → σ=[D, C, B, A, C, B, F, A]
Invert	Create the inverse of a sequence e.g. σ1=[A, A, B, A, B, B, A, A,] → σ=[B, B, A, B, A, A, B, B, A]
Boundary	Create extreme sequences e.g. σ1=[A], σ2=[C, C, C, C, C, C, C, C]

Abbildung 7: Eine Auswahl der verwendeten genetischen Operatoren

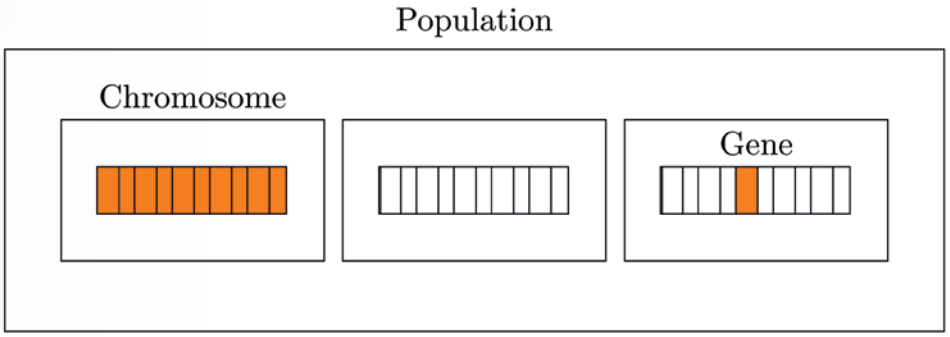


Abbildung 8: Der klassische Aufbau einer Population für die Menge der Testszenarien Q.

„Mutation“ und des „Crossovers“ von Sequenzen kommen dabei zum Einsatz (Abbildung 7) [7]. Die Online-Sequenzen entsprechen dabei den „Chromosomen“, die zusammen die „Population“ des genetischen Algorithmus ergeben (Abbildung 8). In der Simulation werden die Sequenzen mithilfe der Online-Strategie ausgewertet und ihre „Fitness“ bestimmt. Die verwendeten Methoden wurden dabei nur in Abhängigkeit der Sequenzen und nicht für einzelne Probleme implementiert, sodass das Konzept zunächst auf fast alle Problemklassen anwendbar ist. Das entwickelte Framework erlaubt eine Gewichtung einzelner Methoden manuell oder automatisch für spezielle Probleme einzustellen, um dadurch eine gewünschte Priorisierung zwischen den Methoden zu ermöglichen. Auch dieser Ansatz wurde implementiert und mit den bisherigen Ergebnissen verglichen (Abbildung 9 und 10). Darüber hinaus ist auch eine Kombination dieser beiden Verfahren denkbar, die aber bisher noch nicht untersucht wurde.

Literatur

[1] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. Robust Optimization . Princeton University Press, Princeton and Oxford, 2009.

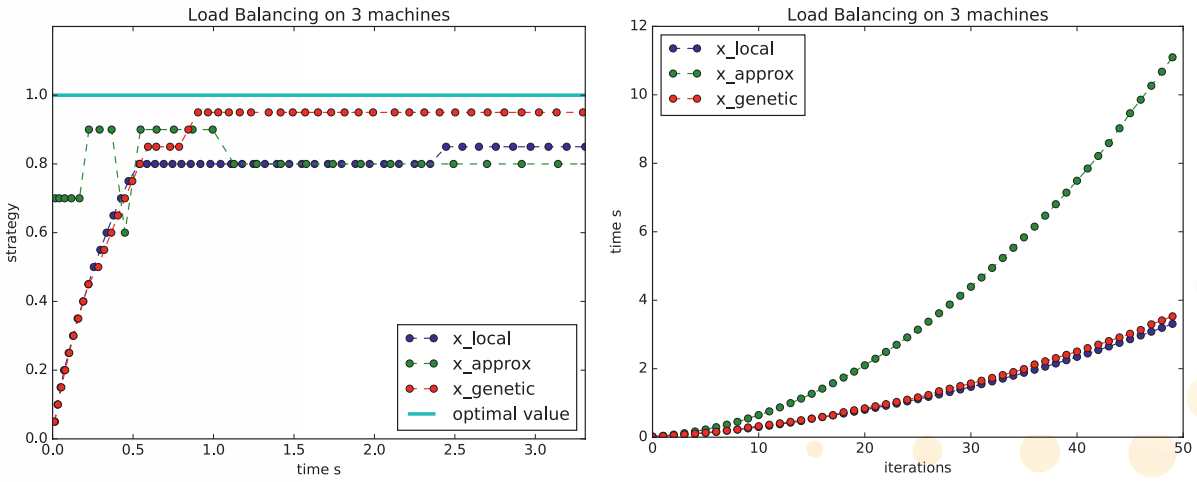


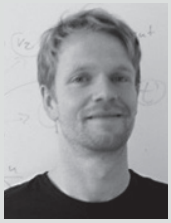
Abbildung 9: Die Optimierungsverfahren im Vergleich. Das Konvergenzverhalten im zeitlichen Verlauf (links) und die benötigte Zeit für die Iterationen (rechts).

[2] Pinedo, Michael. Scheduling: theory, algorithms and systems. Englewood Cliffs, N.J.: Prentice Hall, 1995
[3] A. Borodin and R. El-Yaniv. Online Computation and Competitive Analysis . Cambridge University Press, 1998.
[4] A. Mutapcic and S. Boyd, Cutting-set methods for robust convex optimization with pessimizing oracles. Optimization Methods and Software, 24(3):381–406, 2009.
[5] S. Albers. Online Scheduling: *Introduction to Scheduling*, edited by Yves Robert and Frederic Vivien. Chapman and Hall/CRC Press, 57-84, 2009.
[6] Claus, *Herrmann*, Manitz. *Produktionsplanung und -steuerung*, Forschungsansätze, Methoden und deren Anwendungen. 2015
[7] Goldberg, D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Massachusetts, 1989.
[8] W. Jeffrey, Herrmann. A Genetic Algorithm for Minimax Optimization Problem. Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress onVol.2, 1999.
[9] Irene Fink, Sven O. Krumke, Stephan Westphal. New lower bounds for online k-server routing problems, Information Processing Letters, v.109 n.11, p.563-567. May, 2009

Projektdaten

Das Projekt wird seit August 2015 vom SWZ mit insgesamt einer TV-L E13 Stelle an den Standorten Clausthal und Göttingen gefördert. Beteiligte Wissenschaftler sind:

Prof. Dr. Anita Schöbel
Arbeitsgruppe Optimierung
Institut für Numerische und Angewandte Mathematik
Universität Göttingen



Martin Dahmen, M.Sc.
Arbeitsgruppe Diskrete Optimierung,
Institut für Angewandte Stochastik und Operations Research
Technische Universität Clausthal

Prof. Dr. Stephan Westphal
Arbeitsgruppe Diskrete Optimierung,
Institut für Angewandte Stochastik und Operations Research
Technische Universität Clausthal

Simulation and Optimization of Networks

By now simulation is one of the most important and in many cases the only viable techniques for analysis and optimization of large networks. Telecommunications networks, traffic and logistic networks and energy networks have much in common. The complexity of the networks with its many parallel existing nodes and the flows between nodes is difficult to comprehend and often cannot be controlled with other techniques than simulation. The actual behavior of such a network is often different from the assumed behavior. The construction, operation, modification and optimization of such networks usually represent an infrastructure task that is associated with considerable costs. To avoid undesirable developments, simulation is used as an important tool to determine the properties of a network, the behavior of the critical performance characteristics and parameters at an early stage.

28

Analysis of source delays and their propagation in public transport networks

Michael Kolonko, Anita Schöbel, Jonas Harbering, Fabian Kirchhoff

Delayed or missed rail and flight connections are a nuisance for all travelers and therefore delays in public transport get much attention in the media. They tend to reduce the acceptance of public transport and result in a shift towards private transportation. For the companies operating the network, delays lead to disturbances of their operations and cause productivity losses due to unproductive blocking of rare resources such as platforms and runways. Constant delays in central transportation systems therefore have a serious negative impact on both the mobility of society as well as on the profitability of transport companies. In this project we therefore deal with delays in passenger rail transportation.

Source delays are typically caused by external disturbances such as extreme weather conditions, technical malfunctions or delays in boarding of passengers. These primary delays (source delays) may be carried forward through the network to the next trains, so that a single delay can result in a whole cascade of secondary delays (propagated delays) in the observed network.

A reasonable planning of countermeasures against delays requires modeling and simulation of the transportation network, including the delays encountered in actual operation. Such a modeling of delays has to take into account the interaction of traffic movements on different levels: In the railway traffic, delays can spread through the network by trains waiting for delayed feeder trains or by maintaining minimum distances to other delayed trains on the same track. In addition, the capacity of stations and tracks is limited and the vehicles may have limited availability (due to their circulations). Similar dependencies also exist in air traffic.

Different investigations have been carried out in the present project. The reconstruction of source delays from propagated delays helps to understand the reasons for delays in greater detail. Furthermore, various measures preventing the propagation of delays have been investigated: inserting time buffers in the timetable, decoupling process chains (e.g., no waiting for passengers from delayed feeder trains), improvement of process elements (e.g., adjusting the de facto necessary travel times), consideration of limited capacities during the line planning or increasing the capacities by improving the infrastructure network.

Additional difficulties arise from the fact that the evaluation of schedules is a multi-objective question with partially contrary objectives: Additional delays (due to missed connections, e. g.) can be reduced by increasing the time buffers in the timetable (e.g., by longer stopping times), but this also increases the scheduled traveling time. Conversely, reducing the time buffers in order to have shorter travel times can lead to a system which will react very sensitively even to small delays and delays will spread immediately throughout the network. Possible countermeasures often require substantial investments that need to be weighed against the benefit of delay reductions.

Scenario-based and stochastic approaches

The present project aims at modeling the formation and propagation of delays, particularly in rail networks. To reduce the impact of delays on both long-term infrastructural and short-term operational level, different approaches have to be used and, if necessary, combined suitably. For the long-term optimization during early planning stages, methods of stochastic processes may be applied. A corresponding stochastic model has been developed at Clausthal University. For the operational optimization of an appropriate disposition in the case of delayed trains, a scenario-based approach is more appropriate. This approach has been used by the group at Göttingen University.

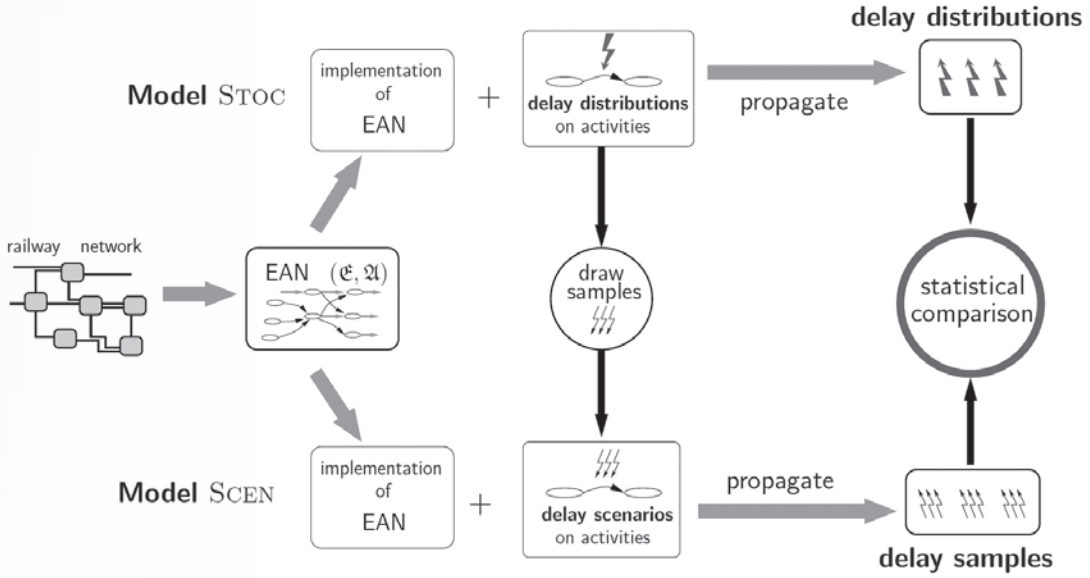


Figure 1: Concept of comparing the scenario-based with the stochastic model

To be able to compare these two approaches with their different target objectives, first the common elements of both models were identified. Both models are based on an event-activity network (EAN). The main differences between the two approaches lie in the way they handle random source delays. In the stochastic model, the source delays are modeled by random variables with probability distributions. In the scenario-based model, this is done by different scenarios with fixed delay values drawn from a probability distribution. Correspondingly, the secondary delays are of different type in the two models: Probability distributions on one hand, scenarios on the other hand.

It could be shown that the two models essentially lead to the same results. For this proof, delay scenarios for the scenario-based model have been simulated with random distributions that also served as input parameters for the stochastic model. The resulting delay scenarios have been compared to the delay distributions calculated in the stochastic model, see Figure 1. There was a very high matching between the two methods, which is shown in Figure 2 for a typical example network. With this, the essential equivalence of both approaches is shown with respect to the delay modeling. Therefore, the model actually used can be selected depending on the specific application. At the same time, the two models and implementations have been validated against each other by this comparison, see also [13].

Estimation and combination of delay distribution functions

If we use a stochastic model, source delays will be modeled by random variables and their distributions. The delays that appear later on as propagated and accumulated delays are also expressed by random variables which are functions of the source delay random variables. To determine their distributions, we need three operations on distribution functions: convolution (for delays that add), multiplication (for the maximum of two delays, e.g., of the departure delay of a train that has a delayed feeder train) and a shift operation to account, e.g., for limited waiting times. If we assume that all delays that may meet in the network are independent, then these three operations are sufficient to determine the delay distributions in tree-like networks. This independence assumption will approximately hold in large networks as, e.g., the nationwide train networks.

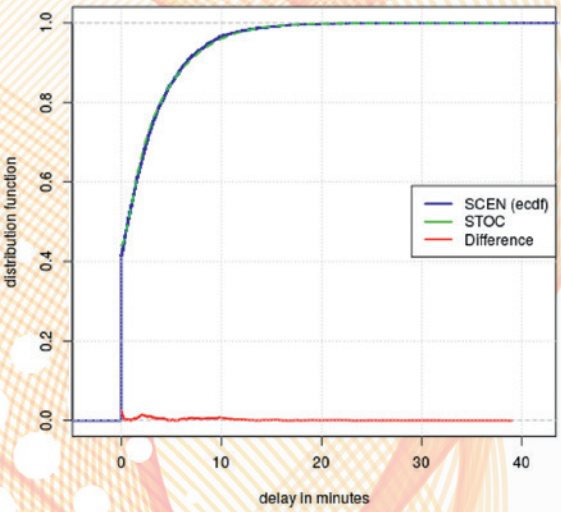


Figure 2: Comparison of the delay distribution function calculated in the stochastic model and the empirical distribution function of delay values from the scenario based model.

To be able to efficiently perform these operations on distributions, one has to restrict the distributions to certain families. In [15] a family Theta of parametrized distributions was used that is closed under these operations, this means that, e.g., the result of the convolution of two distributions of Theta is again an element of Theta. The distribution functions of distributions from Theta can be expressed by so-called theta-exponential polynomials. It is well-known, that - though Theta is closed under the aforementioned operations - the complexity of the parameters of a distribution is growing exponentially with each application of an operation. This is overcome in [15] (see also [14]) by approximating the distributions from Theta by (extended) Hyper-Erlang-distributions after each application of one of the operations. As Hyper-Erlang-distributions form a subset of Theta, subsequent operations can be carried out in Theta, followed again by an approximation by a Hyper-Erlang distribution to simplify the result. Hyper-Erlang-distributions are simpler to parametrize, in fact a fixed small number of parameters is sufficient, and they are easily adaptable to real delay data (see [23]).

This essentially solves the problem to determine delay distribution in cycle-free networks. However, in practice even small traffic networks will lead to an event-activity network with cycles created, e.g., by arcs for passengers that want to change trains. Technically speaking, cycles prevent a topological sorting, i.e., we cannot make sure that all predecessors of a node may have been visited before the node itself is visited. Here, ‚visit' means to determine the delay distribution. E.g. in a cycle it may not be possible to determine the arrival delays of all feeder trains of a departing train before the departure delay itself is determined. In [15], (see also [16]) an iterative approximation of the long-term equilibrium delay distributions in cycles is found. If the sum of all time buffers in a cycle is larger than the sum of all expected source delays and if the delays propagated into the cycle from outside are bounded, then an iteration that simulates the propagation of delays in the cycle over time will converge. This generalizes well-known stability results from queueing theory to cycles of servers.

This approach can be generalized to connected components of EAN consisting of more than one cycle. The overall procedure to determine the distributions of all propagated delays in an arbitrary network is then to first isolate connected components of the network, determine their delay distributions iteratively and then connect these using the tree-like skeleton of the remaining network.

In this way we also obtain an easy to check condition for the stability of a network: in each of its cycles, the convergence condition mentioned above must hold.

Once source delays are propagated to other stations it becomes a difficult task to determine their source. This task is even more complicated if delay information is only accessible indirectly via the actual timetable. Based on this data, in [18] we investigate different methods to reconstruct the source station of propagated delays. Two newly developed methods for the source detection of propagation mechanisms on networks are compared with another existing method (Comin). In the first method, the weighted network median is estimated to be the source (EDM), while the second method tries to trace back the delays along their path of propagation (backtracking). We compared both methods based on different delay scenarios, consisting of a set of source delays in

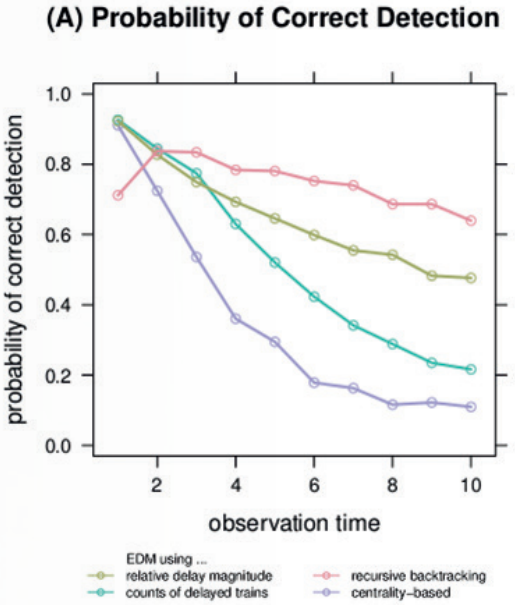


Figure 3: Comparison of methods EDM, backtracking and Comin for the source detection of delay propagation mechanisms

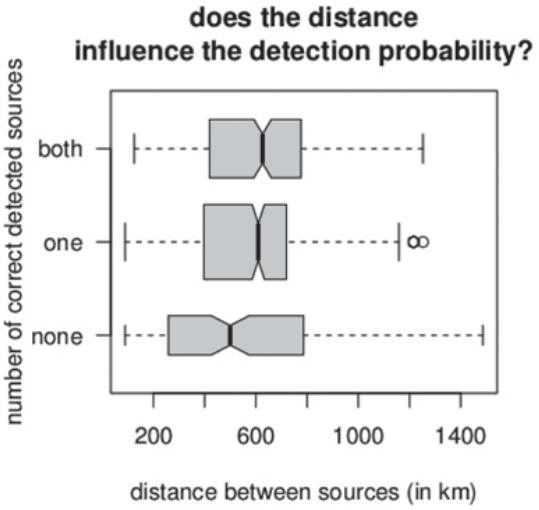


Figure 4: Quality of the method EDM for the source detection in case of two stations with source delays

a fixed station. The source delays are propagated in the network during a period of four hours and the delay situation is recorded at ten different time points within the four hours. For each method at each time point we measure the number of scenarios for which the correct source is detected. This relative incidence is depicted in Figure 3 (probability of correct detection). With respect to the existing method, both newly developed methods are able to improve the performance considerably. This performance improvement is also shown on real data of the largest German railway operator Deutsche Bahn.

We developed this basis methodology to determine single delay sources. The methodology has been extended to allow the detection of multiple sources. Hereby, the network including the counted propagated delays is divided into cluster. In each of the cluster the source detection method EDM is executed with respect to one single source. In Figure 4 the results of the performance of the method based on two sources is depicted. We recognize a tendency of a dependence of the quality of the source detection and the distance between the two sources.

Determination of delay resistant timetables

For the determination of timetables, two already existing systems (HiTT in Clausthal, LinTim in Göttingen) were used and extended, which are described in the following.

In this project HiTT was extended and improved. It now contains an event oriented simulation for the propagated delays as well as a tool to derive the delay distributions as described above. Having two simulations tools based on different approaches allows to verify and calibrate the tools.

In the system HiTT, timetables are obtained from a heuristic optimization combining genetic algorithms and simulated annealing. A multi-criteria approach is used, which for each timetable uses as cost functions the average delays both from simulation and analytical calculation as well as the scheduled total travel time, see Figure 5. It turned out that a reliable determination of average delays by simulation is much slower than to determine mean delays from the delay distributions calculated analytically in the stochastic model.

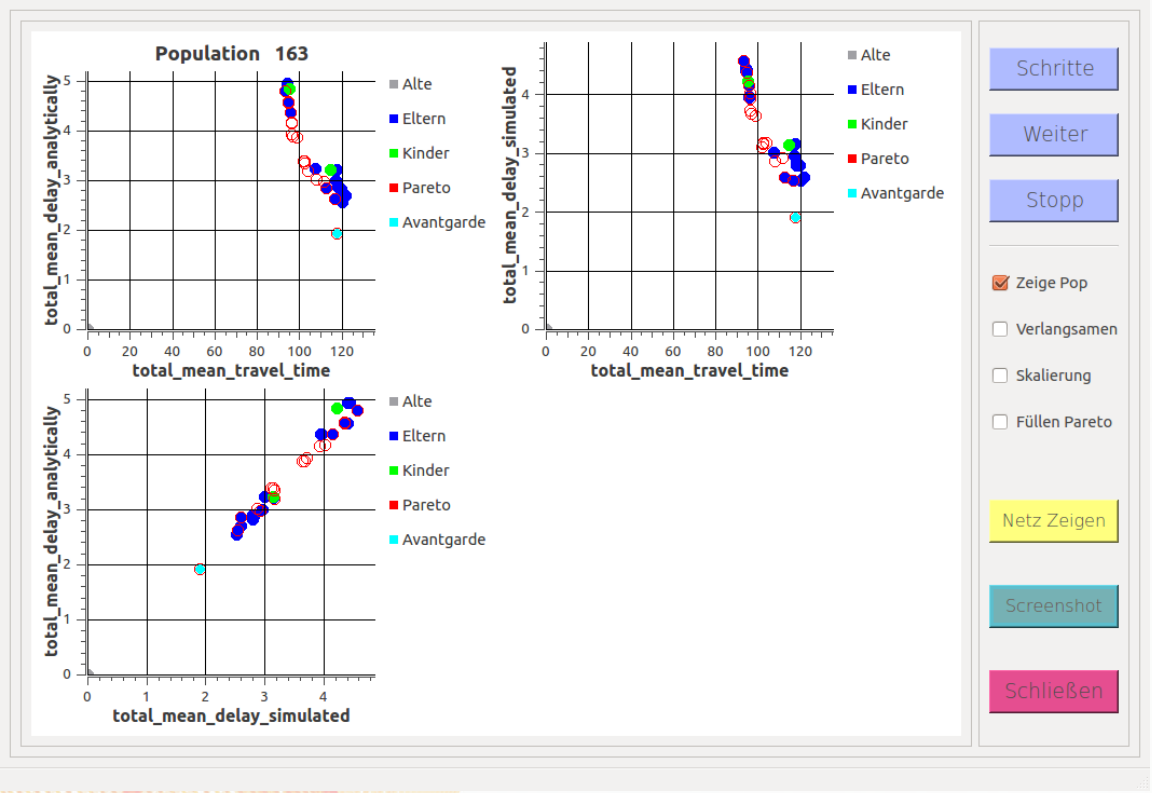


Figure 5: The evolution of the population of timetables in a 3-dimensional costs space, shown are the 2d projections.

We further developed more general approaches to optimize simulated target functions as well as studied the long-term behavior of heuristics, in particular so-called model-based optimization methods (see [16] and [17]). In [8], a method for reducing the number of simulations which is needed to identify the best solution based on a simulated target function with a predetermined error probability was developed. Both approaches have the great potential for the optimization of robust timetables.

LinTim is a software project designed for the planning of public transportation networks. As such, it consists of planning stages, like stop location, line planning, timetabling, vehicle scheduling and delay management (see Figure 6). Throughout this project we added some functionalities to LinTim.

In a new functionality buffer times are added to the durations of the activities of the timetable. These buffer times allow to construct a more robust transportation system with respect to delays. We compared different buffer time distributions (see [2,17]) and tested the robustness of the outcoming system. We found that the robustness of a transportation system is mainly determined by the amount of buffer time rather than the distribution.

Furthermore, we carried out investigations in the area of robust timetabling. [8] gives an overview on older and more recent robustness concepts. As a new robustness concept, the recovery-robustness with an application to timetables is studied in [9]. The idea of this concept is to provide timetables which – in contrast to absorbing every single delay – can easily be recovered in case of delays. Given a finite set of reasonable delay scenarios, this problem can be modeled as a location problem in the space of all timetables: we compute an optimal timetable for each delay scenario and then search for a feasible timetable which minimizes the distance to the set of optimal timetables for the single scenarios. We show that

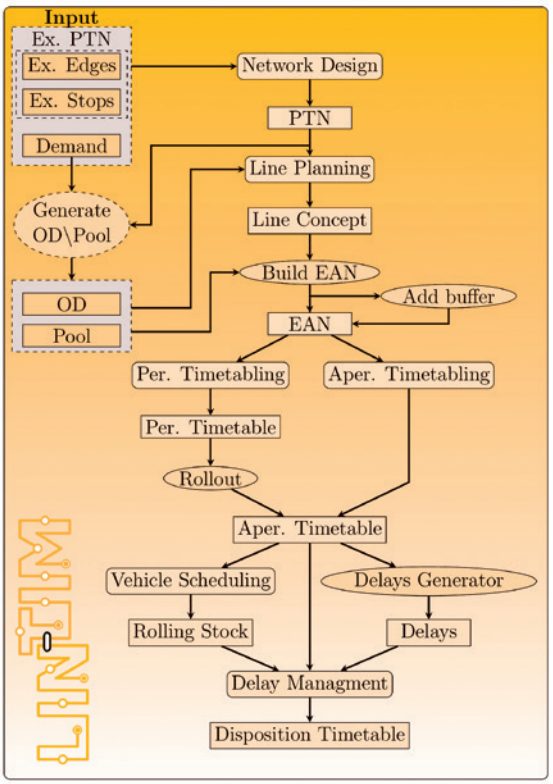


Figure 6: Depiction of the main relations of the data and the functionalities within LinTim

this resulting timetable is recovery-robust. Then, based on linear programming, fast approaches are developed and proposed which compute such a timetable in the aperiodic case.

Investigation on stabilization measures

If a timetable contains buffer times the effect of delays will decrease. If a timetable is recovery-robust it can be recovered easily in most cases. However, both measures cannot guarantee that a timetable can be realized as planned. Hence, more strategies preventing the propagation of delays have to be planned and analyzed. In [1] we propose and analyze a rule-of-thumb. This rule evaluates for a set of given delay scenarios how long a train waits for transferring passengers in an optimal decision. We then applied this rule to new scenarios. In a study based on LinTim we were able to show that this simulation based learning approach leads to very good rules-of-thumb. Furthermore, we extended LinTim by a more realistic model for the optimal planning of wait and sequencing decisions. In the previous activity based model (see [19]) we assume that each passenger has to wait one period for every missed transfer on his path. This is only an approximation because a passenger is counted multiple times even though he only missed one transfer. We improve this activity based model by the following more realistic path based model (see [6]). In the path based model each passenger has a planned path and pursues this path also in case of delays. If any of the transfers in the path can not be maintained a penalty of a single period is accounted for this passenger. It is much more costly to compute this model, since for each passenger a path has to be tracked. However, this model prevents counting the waiting time of passengers multiple times. In Figure 7 the objective function values (y-axis) for different delay scenarios (x-axis) are depicted. A value above the x-axis represents the amount by which the objective value is lower in the new model compared to the old model. The two lines represent different assumptions on the arrival of the passengers at their start station. Firstly, we assumed that all passengers arrive at the start station exactly at the point in time when their train was planned to leave (passenger_delay(realistic)). Secondly, we considered that passengers arrive at the start station according to a uniform distribution (passenger_delay(uniformly)).

The previous stabilization measures represent a macroscopic view on the railway traffic and in particular on the area around stations. In [5] first approaches towards decisions on a microscopic level in station areas have been investigated. We propose two different integer linear programs – a set packing and an assignment formulation. Experimental studies based on the network of the dutch railways show the relevance of the approach.

Integration with other planning steps

The previously described research works only deal with questions related to single planning stages within the planning of public transportation systems. In a novel approach we try to solve different planning stages at the same time, hence integrated. Within the framework of this project we carried out research works which compose a part of the PhD thesis of J. Harbering ([11]). In this PhD

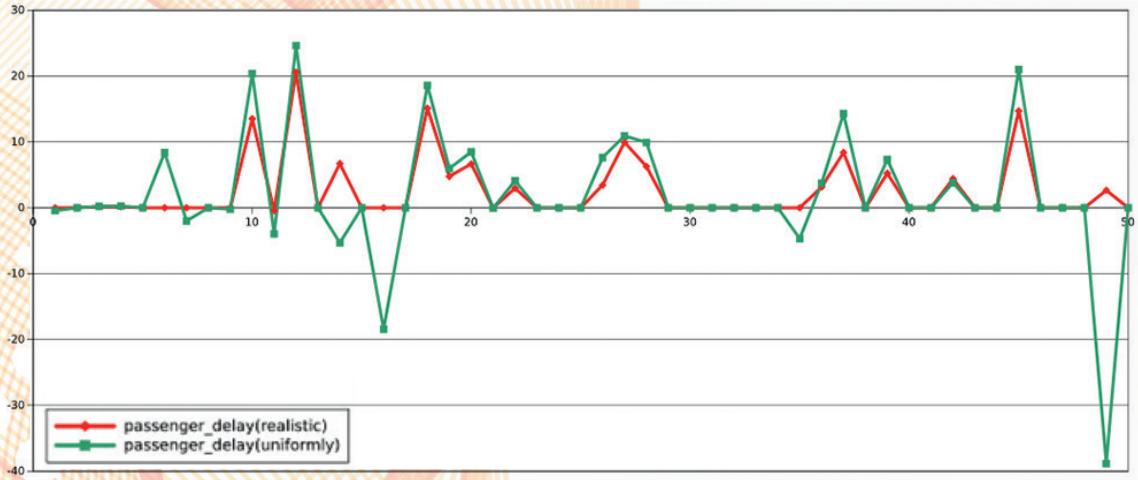


Figure 7: Comparison of activity and path based models for the decision of stabilization measures in the delay management problem

thesis the relation between the following planning stages with a view towards passengers' convenience is elaborated more detailedly.

Usually the design of the infrastructural network and thus the planning of the location of the stations is the first stage (see Figure 6). [3] investigates how stations can be located such that passengers have shortest possible traveling times.

Based on an existing infrastructure network the lines of the public transport are planned in the next stage. This stage consists of two different steps. In the second step the frequencies of potential lines from a predefined pool, the line pool, are computed. The solution, which is a subset of the set of possible lines with associated frequencies, is called a line concept. In [12] we investigated how lines can be found – without an existing timetable – such that the resulting transportation system is not prone to delays. An existing line planning model is adapted to minimize the number of passenger transfers. The number of passenger transfers is minimized because they represent a significant mechanism for delay propagation in the delay management stage. In order to compute this model we have to explicitly compute the paths of the passengers which is solved by a column generation approach. This approach can be applied successfully because the line pool is considered as given. The task to compute such a set of potential lines is the first step of the line planning stage.

This optimization problem is investigated in [7]. This problem poses a crucial component within the planning of public transportation since the problems from the following stages are limited to lines generated from this problem. The line pool generation problem consists in finding a set of potential lines which is neither too small nor too big. On the one hand the set shall not be too big so that the subsequent second step line planning problem can be solved in appropriate time. On the other hand the line pool shall not be too small so that the same line pool contains good solutions for different line planning problems. Furthermore, we wish to be able to control the quality of the line pool with respect to usual objective functions of the line planning (e.g., costs, traveling time of the passengers, delay resistance, …). In order to tackle this novel problem we formulate an integer program and develop a heuristic. Based on an empty line pool the heuristic computes stepwise whether the current line admits a feasible line concept. If this is not the case edges of the network which prevent the feasibility are identified. These edges are then preferably considered in the generation of the lines. The lines have to agree with several conditions which are controlled by parameters. We show that this algorithm frequently gives optimal solutions on small star graphs. Furthermore, this algorithm was tested on instances similar to the one from the Deutsche Bahn. We tested many different parameter settings and thus resulting line pools. Based on this we

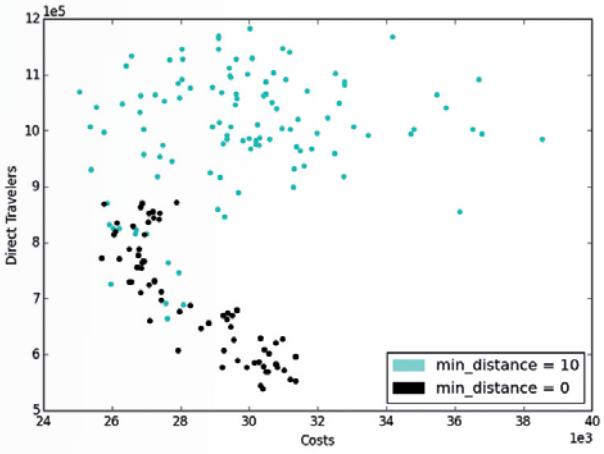


Figure 8: Analysis of different line pools color coded by the minimal distance between start and end station of the lines

could show that with certain parameter settings also certain properties of the line pools can be obtained. For example, we could show that both long lines and a minimal euclidean distance between the start and end station of a line lead to many direct travelers (see Figure 8).

The research work in the area of integrated public transportation planning initiated in this project is followed up by the research project DFG - FOR 2083.

References

[1] R. Bauer and A. Schöbel. Rules of thumb -practical online strategies for delay management. Public Transport, 6(1):85-105, 2014.

[2] V. Cacchiani, A. Caprara, and M. Fischetti. A lagrangian heuristic for robustness, with an application to train timetabling. Transportation Science, 46(1):124-133, 2012.

[3] E. Carrizosa, J. Harbering, and A. Schöbel. The Stop Location Problem with Realistic Traveling Time. Journal of the Operational Research Society, 67(10): 1325-1337, 2016.

[4] B. Fuhr. Robuste Flugplanung mittels stochastischer Modellierung. PhD thesis, TU Clausthal, 2007.

[5] T. Dollevoet, D. Huisman, L. Kroon, M. Schmidt, and A. Schöbel. Delay management including capacities of stations. Transportation Science, 49(2):185-203, 2015.

[6] T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel. Delay management with rerouting of passengers. Transportation Science, 46(1):74-89, 2012.

[7] P. Gattermann, J. Harbering, A. Schöbel. Line Pool Generation. Public Transport, 1-26, 2016, published online.

[8] M. Goerigk and A. Schöbel. Algorithm engineering in robust optimization. In Algorithm Engineering: Selected Results and Surveys, Editors: L. Kliemann, P. Sanders, LNCS State of the Art, 9220, 2016, in print, http://arxiv.org/abs/1505.04901.

[9] M. Goerigk and A. Schöbel. Recovery-to-optimality: A new two-stage approach to robustness with an application to aperiodic timetabling. Computers and Operations Research, 52: 1-15, 2014.

[10] B. Görder and M. Kolonko. Ranking and selection: A new sequential Bayesian procedure for use with common random numbers. 2014. submitted, http://arxiv.org/abs/1410.6782.

[11] J. Harbering. Planning a Public Transportation System with a View Towards Passengers' Convenience. PhD thesis, University of Göttingen, Germany. 2016

[12] J. Harbering. Delay Resistant Line Planning with a View Towards Passenger Transfers. Technical report, University of Göttingen, 2014.

[13] J. Harbering, F. Kirchhoff, M. Kolonko, and A. Schöbel. Delay propagation in public transport - stochastic modeling meets scenario approach. Technical report, 2014. zur Veröffentlichung vorgesehen.

[14] F. Kirchhoff. Modelling delay propagation in railway networks. In Operations Research Proceedings 2013, pages 237 - 242, 2014.

[15] F. Kirchhoff. Verspätungsfortpflanzung in Bahnnetzen, Modellierung und Berechnung mit abgeschlossenen Familien von Wahrscheinlichkeitsverteilungen. PhD thesis, University of Technology Claustal, Germany, April 2015.

[16] F. Kirchhoff and M. Kolonko. Modeling delay propagation in railway networks with closed families of distributions. submitted, 2014.

[17] L. Kroon, G. Maroti, M. Retel Helmrich, M. Vromans, and R. Dekker. Stochastic improvement of cyclic railway timetables. Transportation Research Part B: Methodolog-

ical, 42(6):553-570, 2008.
[18] J. Manitz, J. Harbering, M. Schmidt, T. Kneib, A. Schöbel. Source Estimation for Propagation Processes on Complex Networks with an Application to Delays in Public Transportation Systems. Journal of the Royal Statistical Society: Series C, 2016, published online.
[19] A. Schöbel. Optimization in public transportation. Stop location, delay management and tariff planning from a customer-oriented point of view. Optimization and Its Applications. Springer, New York, 2006.
[20] A. Schöbel and S. Schwarze. Finding delay-resistant line concepts using a game-theoretic approach. Netnomics, 14(3):95-117, 2013.
[21] Z. Wu and M. Kolonko. Asymptotic properties of a generalized cross entropy optimization algorithm. IEEE Transactions on Evolutionary Computation, 18:1-16, 2014.
[22] Z. Wu and M. Kolonko. Absorption in model-based search algorithms for combinatorial optimiztion. In Evolutionary Computation (CEC), 2014 IEEE Congress on, pages 1744-1751. IEEE, 2014.
[23] Z. Yang. Modellierung und Simulation von Verspätungsdaten mit PH-Verteilungen. Masterarbeit, TU Clausthal, 2012.

Project data

The project was funded from SWZ with 1.75 TV-L E13 staff positions from April 2013 to March 2016 at the sites Clausthal and Göttingen. Involved scientists are:

Prof. Dr. Anita Schöbel,
Research Group Optimization
Institute for Numerical and Applied Mathematics
University of Göttingen

Jonas Harbering, M.Sc.
Research Group Optimization
Institute for Numerical and Applied Mathematics
University of Göttingen

Prof. Dr. Michael Kolonko,
Research Group Stochastic Optimization
Institute of Applied Stochastics and Operations Research
Clausthal University of Technology

Dr. Fabian Kirchhoff
Research Group Stochastic Optimization
Institute of Applied Stochastics and Operations Research
Clausthal University of Technology

Prof. Dr. Thomas Hanschke,
Research group Stochastische Modelle in den Ingenieurwissenschaften
Institute of Applied Stochastics and Operations Research
Clausthal University of Technology

Securing Communications in Internet of Things (IoT) Environments

Dieter Hogrefe, Xiaoming Fu, Salke Hartung, Jiachen Chen

1 Introduction and Research Goals

1.1 Project motivation

Infrastructure security is an important ongoing field of research in computer networks. Especially security issues in resource-constrained Internet of Things (IoT) environments is prone to security holes as its architecture design, hardware capabilities and applications are different from ordinary networks.

Emergency and natural disaster are a prominent example of scenarios where a communication infrastructure that could prevent spam from consuming valuable resources is vital. In the aftermath of catastrophic events, including earthquakes, tsunami, terroristic attacks, the standard communication infrastructure (landline, mobile phones, Internet, etc.) could be severely damaged and fragmented. In such a scenario, it is all the more important to ensure that the resources are used to disseminate important information (e.g., emergency warning messages, information regarding food and shelter, and safety confirmation) and spam and other non-priority information should not be forwarded. Moreover, there is a need for a lightweight signature generation and verification algorithms.

As more and more devices get connected to the Internet, it is all the more important to consider decentralized approaches that could either be an alternate or compliment a server based approach. In order to decentralize, we consider the following approaches: i) Information Centric support for providing virtualized third party services to end-users/Internet of things - Here, the motivation is the virtualized services could be provided closer to where the service (firewall, spam filtering, QoS) is needed instead of having to go all the way to the server; ii) Distributed Social Networks based privacy and load sharing - Here the motivation is the nodes that are part of a social network could in fact support one another to provide storage, reliability as well as ensure that the privacy is not violated.

The term localization IoT environments denotes the process of determining the own position in space while being part of a network. In Wireless Sensor Networks which are a possible application of IoT collected information usually only makes sense if it is associated with spatial information. Since equipping all nodes in the network with GPS has several drawbacks, more efficient localization algorithms are required. Furthermore, more efficient algorithms spare computational time which can then be used for running additional security operations.

A common technique to reduce production and deployment costs, to save battery power and to keep the hardware at a reasonable size is to use only a few seed nodes which are aware of their position (either by static positioning or by using GPS) and help others by sending out location announcements including their own position. How exactly the location of the unknown nodes is determined is defined by the localization algorithm.

Security in WSNs often only focused on securing data transmission, data integrity and DDoS attacks. Nowadays the research focus switched to protecting key items in the network. These are nodes with special responsibilities which might cause the network to fail or behave unexpected if they are compromised. In the context of localization these special items are the seed nodes which are indispensable for running the localization process. Compromised seed nodes could send out fake location announcements or refuse any communication at all to prevent other nodes from estimating their position. To address this issue our work also focuses on solutions to prevent attacks against localization services.

1.2 Importance of simulation

Although implementing algorithms designed for IoT applications is usually easy, field tests in real testbeds are hard to conduct [56]. Significant specifications of an algorithm like scalability and robustness can only be tested with hundreds of nodes which is almost impossible to do in a real world scenario, because usually one does not have access to a network of this size or it is impossible to disrupt the ongoing operations.

Fortunately, network simulation provides an easy-to-use and cheap method for constructions of networks built by thousands of nodes and is theoretically only limited by the hardware capabilities of the machine running the simulation software. In addition to that simulation allows to construct special case scenarios and to repeat experiments with different parameter settings in reasonable time. State of the art simulation software is able to model all kinds of network influences like typical network events (e.g. broken links, data congestion, data loss), physical effects (e.g. weather, path-loss, reflection, scattering), user behavior (e.g. mobility of nodes, repeating data patterns) and provides implementations of all common protocols of the ISO/OSI layer model.

Another important aspect of using simulation software is that usually it is easy to extend or even replace the existing models and protocols by custom implementations without reprogramming all underlying layer models. Therefore implementing new algorithms can be done with way less effort compared to a real testbed implementation.
In our research work we use simulation tools fitting best for the main aspect in focus. This ranges from specialized software for network simulation [63] to profound mathematical tools [64] and even custom written tools for very specialized simulation tasks.

2 Completed Work

All of the following described work has been published as full papers or demos on international research conferences. A complete list of the publications associated with this project can be found at the end of the report.

2.1 Evaluating methods for distance estimation in WSNs

A common practice in localization algorithms is to measure distances to other nodes and calculate the own position using these measurement results. An initial study which was done in the context of the university project "Forschungsorientiertes Lehren und Lernen" and evaluated the quality of the received signal strength indicator (RSSI) as a distance estimator for the use in range-based active localization algorithms. This work was presented and published at the IEEE ICWiSe 2013 conference in Kuching, Malaysia [5].

RSSI is used in lots of localization algorithms but mostly only evaluated via simulation. The fineness of the evaluation in these cases is depending on the quality of the system models used to represent the environment. Our study shows that in a real implementation RSSI only can be used as a very rough estimator and is subject to lots of outer influences which can affect the measured results (see figure 2.1). In our paper we list possible reasons for measurement impacts and compare our results to the outcomes in related work [47], [48], [49], [50].

2.2 Efficient localization algorithms for WSNs – SA-MCL

Our second completed work in the field of localization algorithms for WSNs is Sensor-Assisted Monte Carlo Localization (SA-MCL) which is an extension of the well-known MCL algorithm. The

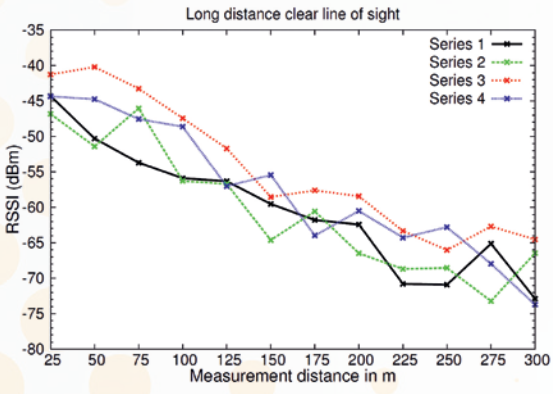


Figure 2.1: RSSI levels in long distance measurement series

aim of improving a localization algorithm is either to decrease the localization error or to reduce the number of seed nodes necessary to achieve stable localization in the network. In SA-MCL we achieve both by using additional sensors to gather information about the velocity and direction of movement of a sensor mote. With the help of these sensor readings a mote is able to update its position even in the absence of seed nodes.

2.2.1 Design

The classic MCL approach represents the localization estimation of a mote by using a set of weighted particles. Each particle represents a possible location of the mote; the weighted average of all particles is assumed to be the estimated position of the mote. The only parameters known by all motes in the network are the deployment area dimensions, the communication range r and the maximum velocity vmax of a mote. Further it is assumed that all motes in the network are mobile and can move freely in every direction. Only a few motes are always aware of their position, either by being equipped with GPS or by having a fixed but therefore static position. These special motes are called seed nodes and are used to help all others determining their position. Seed nodes regularly broadcast location announcements including their position, a unique ID and a timestamp. Motes receiving these announcements record these as new observations and use them in the MCL algorithm.

MCL works as a two step algorithm: In the prediction step new particles are generated from the existing set by drawing a new particle in a circular area with radius vmax around the previous particle location. This step is used to account for the growing uncertainty of the real location due to movement of the sensor mote. In the filtering step the recorded observations are used to filter particles which are not in communication range of a seed node. These two steps are repeated until N valid particles are generated which represent the new localization estimation of the mote. The main problem of MCL are situations where no new observations have been recorded. In these cases only the prediction step is executed which leads to a growing uncertainty of the motes position and consequently to high localization errors.

In SA-MCL we account for these situations by using additional sensor information. The sensors are only activated if the number of new observations received during the last localization interval tends to reach 0. The sensor information is used to record a rough estimation of the path taken by the mote relative to its last estimated location. Once no new observations have been received the recorded path is used to drag the whole particle set along. By taking this action the degeneration of the particle set is prevented and the position estimation cannot get worse. When receiving location announcements again the sensors are turned off and the mote switches back to the more efficient MCL approach.

2.2.2 Simulation Results

Our approach has been implemented as an extension of the original simulation software used in [52]. The simulator is implemented in Java and provides support for the random waypoint model [55], [56] to account for node mobility. Our simulations are done in a 500m x 500m square region with 200 network nodes. The radio range is set to 50m. All experiments are repeated 25 times and the results are averaged. We mainly study 3 simulation parameters:

1. Node Velocity:
   When analyzing the impact of mobility in the network the speed of a node is chosen randomly from the interval $[V_{min}, V_{max}]$
2. Seed degree:
   The aim of our improvement is to account for situations with a low seed degree. Therefore we constantly lower the average number of seeds in the neighborhood (S_d) of a node. Additionally we examine the behavior of both algorithms if we increase the number of seed nodes.
3. Error in sensor readings:
   We account for inaccuracies in sensor readings by adding a sensor error of α. We study the effect of increasing a in the interval of [0 %, 40%], for all other experiments, α is fixed at 15 %.

The key metric for a localization algorithm is the localization error which we give as an average of all nodes:

$$Error = \frac{1}{n}\sum_{i=1}^{n}||e_i - l_i||$$

where n is the number of all nodes, e_i is the location estimation of the i^{th} node and l_i is the real location of node i.

For this report we only show the most important of our results. The key aim of SAMCL is to account for situations where the seed node degree is low. In figure 2.2 we show how using the sensor information helps SA-MCL to perform much better than MCL especially if $S_d < 1$. SA-MCL performs about 40% better in low seed degree situations ($S_d < 1$) compared to MCL. It can be found that for higher seed degrees MCL and SA-MCL tend to converge to a single graph as there are only few situations left where no seed information is available, i.e. SA-MCL cannot benefit from its additional features any longer.

We present the localization error for different V_{max} in figure 2.3. As expected for higher node velocities the localization error for both algorithms is slightly higher, since seed nodes will move out of communication range faster when having a higher velocity. However, we need to point out that there is a threshold situation, since both algorithms require a certain velocity for providing enough

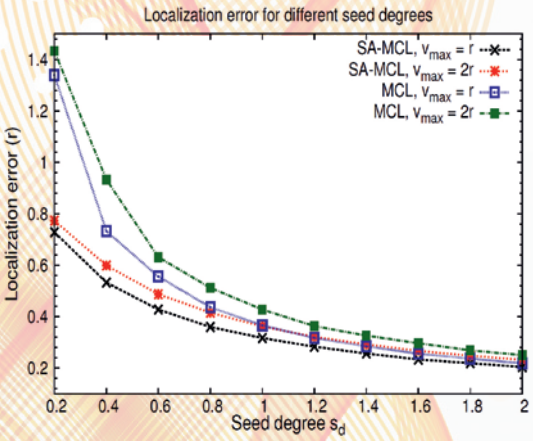


Figure 2.2: Localization error for different seed degrees

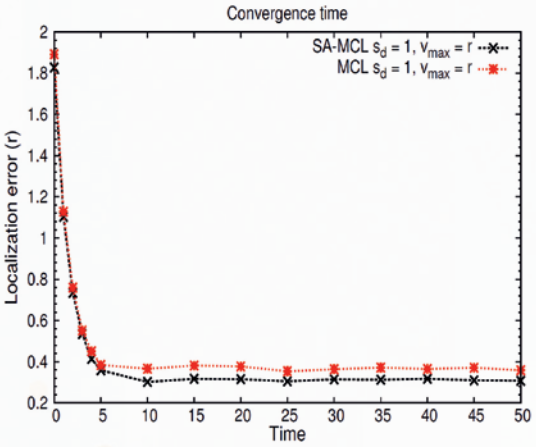


Figure 2.4: Convergence Time

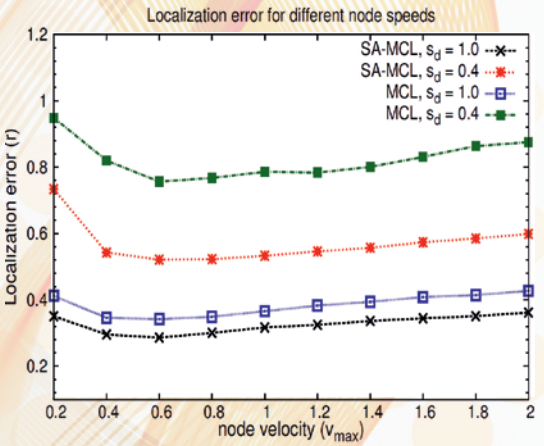


Figure 2.3: Localization error for different node velocities

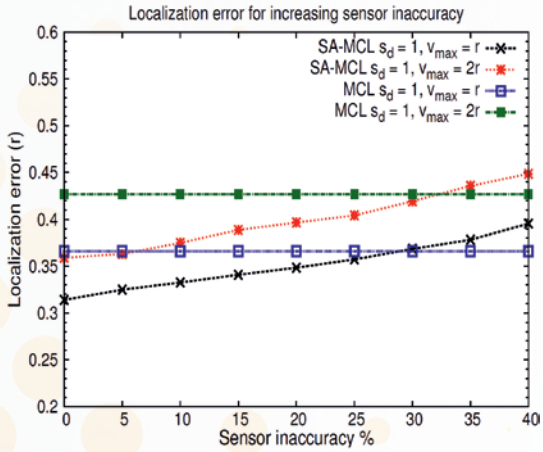


Figure 2.5: Localization error for different sensor precisions

variability in receiving location announcements from different seed nodes. From figure 2.3 we can infer the optimal velocity (i.e. lowest localization error) for all nodes seems to be 60 % of the transmission range.

In figure 2.4 we explore the convergence time of both algorithms. While our outcomes confirm the results in [52] it also can be seen that SA-MCL behaves in a similar way as MCL and reaches a stable state after about 10 time steps. After that, almost no further improvement can be found. The main problem which could arise in SA-MCL is the precision of the sensor readings. To account for imprecise hardware we simulate sensor errors from 0 – 40 %. The results are shown in figure 2.5. MCL does not make use of sensor information, therefore the results do not change and are only given as a reference. It can be found that even when increasing the sensor error up to 30 % SA-MCL performs better than MCL.

2.2.3 Conclusion

SA-MCL has been initially verified by simulation to be a suitable extension for MCL in low seed degree situations. With these initial results we went on to a practical field test implementation which is described in the next section.

2.2.4 Practical implementation of SA-MCL

Going beyond simulation, we built a practical implementation of MCL and SA-MCL and experimentally evaluate its performance in a field test. In this part of the paper, we first describe our mobile wireless sensor network testbed which is based on radio controlled cars (RC cars). Secondly, we describe the implementation of our approach and particularize how the sensor data is gathered. Finally, we describe our approach of parameter optimization and present our final results.

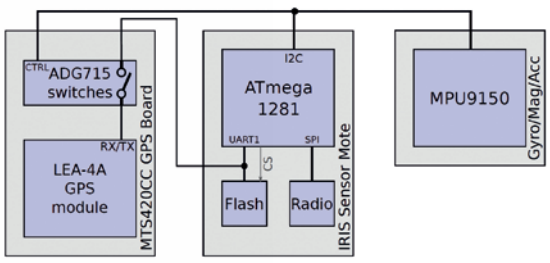


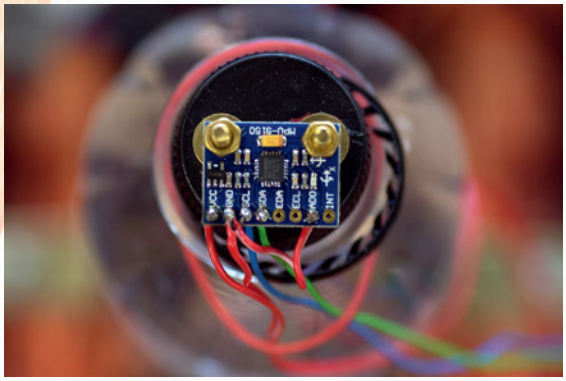
Figure 2.6: Data bus connections on the mobile node. Power supply lines not shown.

The hardware platform used for the field test is comprised of a small radio controlled electric vehicle carrying a sensor mote connected to a GPS receiver, accelerometer, gyroscope (angular rate sensor) and magnetometer. The sensor mote is an IRISWireless Measurement System manufactured by MEMSIC Inc. The mote's main board contains an Atmel ATmega1281 MCU, an 802.15.4/ZigBee-compliant 2.4GHz radio and a 512 kB flash ROM for storing measurement results. It also includes an expansion connector for connecting additional sensor boards. GPS functionality is provided by the Crossbow Technology MTS420CC sensor board (Figure 2.7a) which contains a LEA - 4A GPS receiver module by uBlox AG which is connected to the mote's UART1 serial port through the expansion connector (Figure 2.7c). As the flash chip and the GPS module are connected to the same UART, bus arbitration is necessary. This is done via a separate chip-select line for the flash chip and via I2C-controlled ADG715 switch ICs for the GPS board (see Figure 2.6 for an overview).

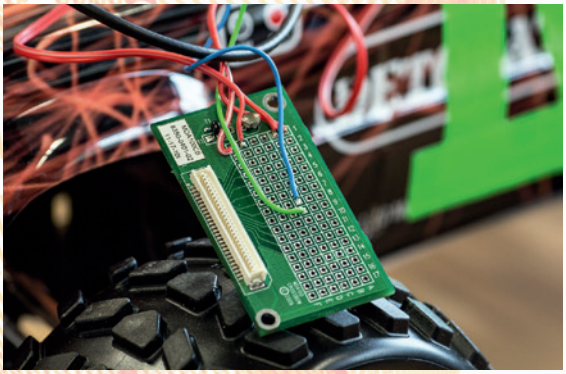
An Invensense MPU9150 9 - axis motion sensor as shown in Figure 2.7a provides accelerometer and gyroscope sensors as well as the AKG8975 magnetometer which is included on-chip. A breakout board containing the MPU9150 is mounted on a vertical plastic spacer giving about 20 cm of clearance from the vehicle's body using non-magnetic (brass) screws to minimize magnetic interference originating from the frame and motor. It is connected to the mote's I2C-bus through wires and an interface board plugged into the expansion connector. The completed sensor mote stack including the mote itself, the GPS board and the expansion board is shown in Figure 2.7d.

Power is supplied to the mote and all sensors from the vehicle's 7.2V NiMH-battery through a LM2596-based DC-DC step-down circuit, whose output has been manually adjusted to be in between 3.290V and 3.300V using a MASTECH M9803R multimeter. The final car assembly is shown in Figure 2.8.

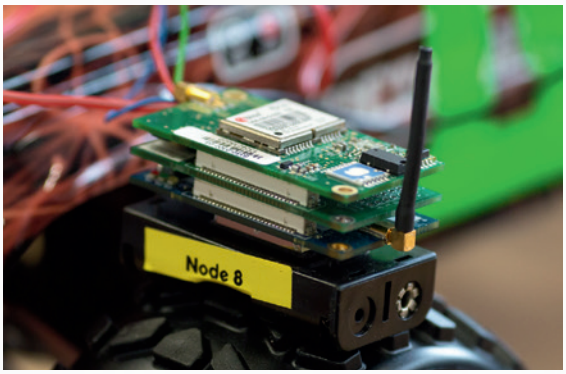
(a) GPS module



(b) MPU-9150



(c) Expansion board



(d) Full mote stack

Figure 2.7: Mobile node hardware components.

While estimating the compass heading of a sensor node by employing a simple approach that uses only a magnetometer, several challenges arise. Sensor noise leads to unsteady, fluctuating sensor data even at rest. Magnetic interference both from the RC car and the environment can lead to flawed magnetometer data. Finally, tilting the magnetometer introduces noticeable errors and can easily occur while driving.

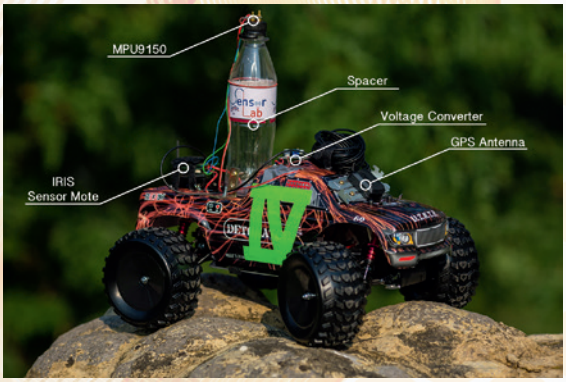


Figure 2.8: Fully assembled RC car mobile node.

Sensor noise can, to some degree, be suppressed by means of low pass filters. Magnetic interference from the RC car is compensated by mounting the magnetometer on a spacer, as seen in Figure 2.8, keeping it away from the magnetic fields originating from the RC car. To mitigate magnetic interference from the environment and situations in which the magnetometer is tilted, accelerometer and gyrometer data is employed. Additionally, the accelerometer is used to determine if the sensor node is moving, which is important for running SA-MCL.

The gyrometer sensor provides angular velocity data, relative to the local orientation of the

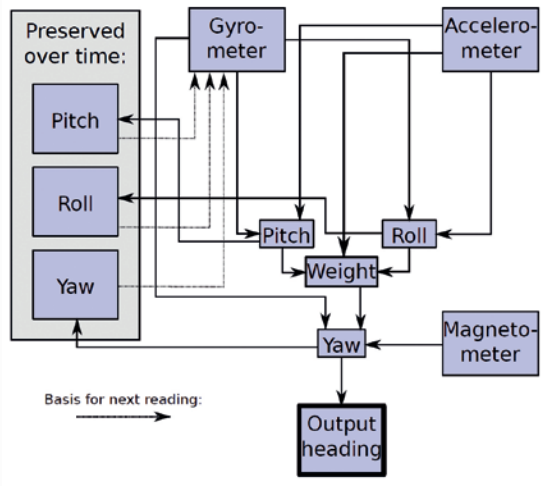


Figure 2.9: Motion sensing overview.

sensor node. This makes it possible to update a known orientation estimate using gyrometer data. To do this, the relative orientation offset has to be rotated to an absolute orientation offset and added to a previous estimate.

By measuring the vector of gravity using the accelerometer, it becomes possible to calculate pitch and roll of the sensor node while at rest. Together with the yaw reading from the magnetometer, these values provide a base orientation of the sensor node that can be updated using the gyrometer. Since the gyrometer's accuracy is not impacted by acceleration, magnetic fields or tilt, a weighted average of the gyrometer's yaw estimate and the magnetometer yaw estimate provides a robust estimate of the sensor node's compass heading.

The weights are adjusted according to which adverse conditions are detected. If the sensor is moving or tilted, accelerometer and magnetometer receive lower weights. When not tilted or in motion, the gyrometer is aggressively weighted less and the magnetometer is used to reset the orientation, eliminating accumulated sensor drift. In addition to the situational weighting approach, low pass filters are employed to reduce the noise of sensor data. An overview of the approach can be seen in Figure 2.9.

Due to the limited processing power and working memory, care has to be taken to avoid the use of expensive floating point calculations as much as possible. While anchor nodes receive their location information from GPS sensors attached to the extension connector, the longitude and latitude information is projected into a cartesian coordinate system by means of the Universal Transverse Mercator (UTM) map projection. The coordinates are centered around those of the site where the experiment takes place and stored in a 16 bit fixed point format with a resolution of approximately 3mm. This allows relatively efficient computation of distances even within the limits of an 8 bit microcontroller without any significant loss in precision.

Our implementation is able to complete one run of MCL with fifty samples within 70 ms to 80 ms on IRIS sensor motes. When running SA-MCL and using the previously described dead reckoning approach, the runtime is reduced even further to 6 ms to 7 ms, as no resampling of samples has to be performed. Both algorithms are executed directly on the hardware platform.

MCL and SA-MCL are both run once per second after which an announcement message is broadcast, containing anchor node information. GPS readings are also taken at a rate of one per second, while motion information, i.e. compass

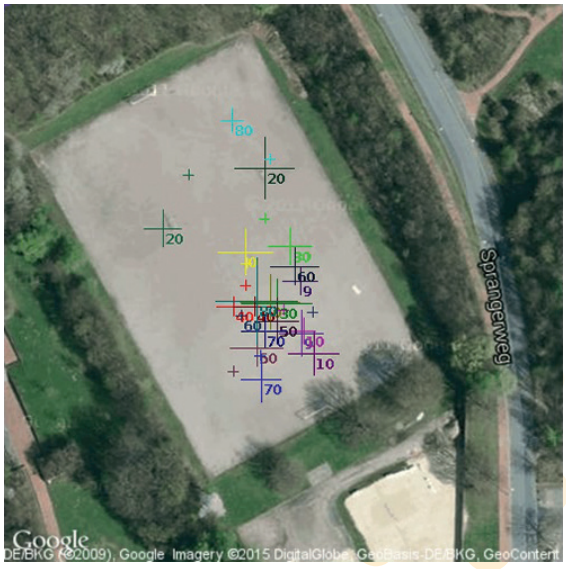


Figure 2.10: Screenshot of live view mode.

heading and a flag that determines if the sensor is moving at all, is measured about five times per second. All events such as measured sensor data and received or sent messages are logged to flash memory to allow for detailed analysis of everything that happened during the experiment. Furthermore, our implementation features a live view mode which displays the positions of all sensors in the testbed as show in Figure 2.10. The live view mode is not required for determining the performance results of our evaluation, but provides a comfortable way to quickly detect problems.

Ten remote controlled cars, which are configured as described above, are driven over a flat sports field, controlled by volunteers. The dimensions of the experimental area are 100 m x 50 m. The drivers of the cars are instructed to imitate the random waypoint mobility model. This means a car is supposed to drive a more or less straight line, pause for a short time, change direction and start over. To maintain control of the car and ensure safe driving the maximum velocity is limited to V_{max} ≈ 6km/h. Data is collected over a period of about 18 min to 20 min.

To enable the live view mode it is necessary to forward the positioning information, which otherwise is stored only in every motes' flash memory, to a central base station. Therefore, a grid of relay nodes is required. These relays do not have any other functionality than forwarding packets to the base station. Figure 2.11 shows the field test setup with the base station located in the center, eight relay nodes arranged in a grid and the mobile nodes are allowed to move arbitrarily on the field.

To allow fine grained control of the nodes' radio ranges, an artificial limit is introduced by means of an RSSI cut-off value. If a node receives a location announcement, it will always record it in its flash memory log. However, if the RSSI is below a certain, configurable threshold, the announcement will not be processed further by MCL. Experimentally determined RSSI values for different ranges are given in Table 2.1. We are aware that this mapping is only a very rough estimation and does not hold true in general. However, packets misjudged based on this method are considered to be a consequence of the varying antenna characteristics on real hardware. To provoke many situations without seed information, the RSSI cutoff

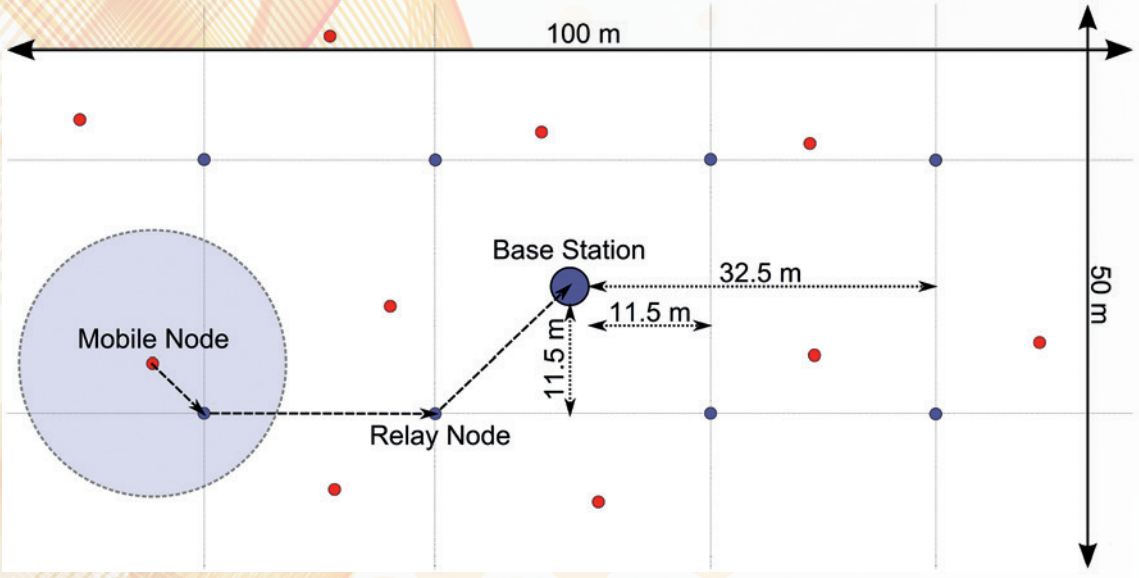


Figure 2.11: General field test setup.

is set to 29, which corresponds to a radio range of ≈ 5 m according to our provided table.

Table 2.1: Mapping of RSSI value to distance.

RSSI	50	33	26	12	9	6
Dist.	1m	4m	8m	16m	22m	26m

To maximize the usable data gained from the experiment, all seed nodes also run both MCL and SA-MCL, i.e. every seed node is a simple node at the same time. To avoid a car localizing itself based on its own location announcements, announcements originating from itself are ignored.

All cars are equipped with GPS sensors to allow determining the localization error with respect to the reference value provided by the GPS readings. We are aware of the fact that GPS introduces a localization error on its own. However, more precise reference systems usually involve multiple cameras and are beyond our budget. Seed nodes will announce their position also based on their current GPS information.

All data collected during the field test, including GPS data, sent and received location announcements and location estimations of MCL and SA-MCL are stored in the flash memory of each node. Using the logs gained from the field experiments, it becomes possible to run further simulations with very high fidelity based on the collected data afterwards.

2.2.5 Results

After performing the field test, we evaluate the performance of SA-MCL and MCL according to various error metrics.

Absolute localization error

Again, the absolute localization error is calculated as the Euclidean distance between the estimated position and the real position given by our GPS ground truth data. Our field test clearly shows that the performance of SA-MCL is superior to that of MCL. A per-car breakdown of the absolute localization error is given in Figure 2.12.

The average absolute localization error of SA-MCL during this experiment is 11.37 m, while the average absolute localization error for MCL is 27.1 m, which corresponds to an average improvement of 58 %. The maximum improvement is as high as 66 %, while even in the worst case there is still an improvement of 46 %.

Grid localization error

Intuiting that SA-MCL and MCL performance is different in different regions of the field test area, besides the absolute localization error another type of metric is developed, called grid error. For this error measure, the test field is subdivided into cells on a grid. The cells have a dimension of 3m 3 m. For each cell the average error at the times

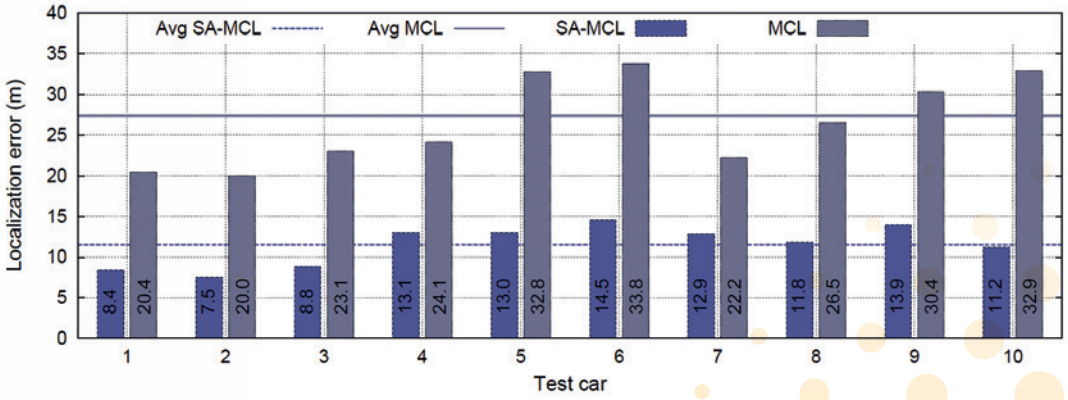


Figure 2.12: Absolute Localization Error.

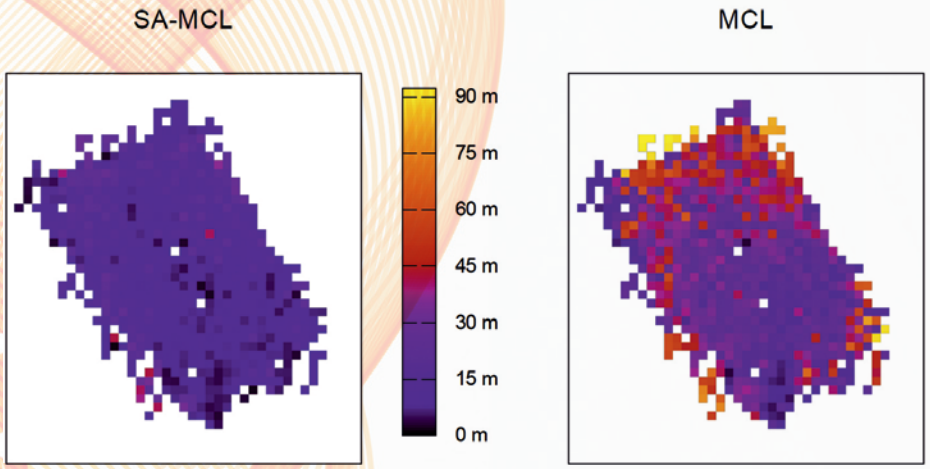


Figure 2.13: Average grid error over all cars.

that the GPS coordinates are within the given cell is determined.

The resulting grid is then plotted with color coded errors in the style of a heat map. The grid error plot showing the averages of all ten cars can be seen in Figure 2.13, with the average error given in meters according to the scale. At a glance, it becomes obvious that especially at the outskirts of the field test area, SA-MCL outperforms MCL. This is likely due to a lower seed node density in those regions, which results in MCL's location estimates rarely entering those areas. The effect would be even more drastic on a larger test field.

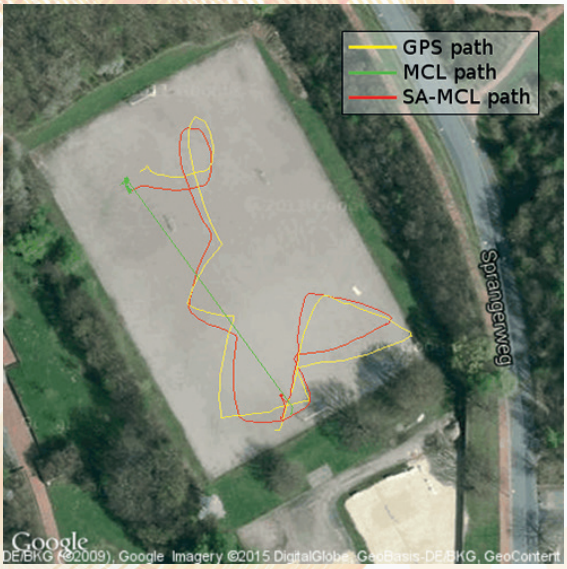


Figure 2.14: Path taken by RC car during experiment.

Optical trace

Figure 2.14 shows a partial plot of the path taken by one of the RC cars overlayed on a satellite picture of the experiment's locale over a period of 3.5 min. The figure illustrates a period in which no seed information is available to the node. The yellow line corresponds to the location provided by the GPS sensor, while the red line gives the location estimate of SA-MCL and the green line shows the estimate calculated by MCL. Only a part of the full path is given, as a plot of the full path would make it hard to visually discern details due to its complexity.

It can be found that SA-MCL is almost perfectly imitating the real path of the car as given by the GPS data. Using the combined data of all sensors to determine the new heading SA-MCL is able to react immediately to changes in orientation. MCL on the other hand is unable to account for the missing seed information and therefore jumps from its last known location at the top left corner directly to the next determined location after receiving seed information again at the bottom of the test field.

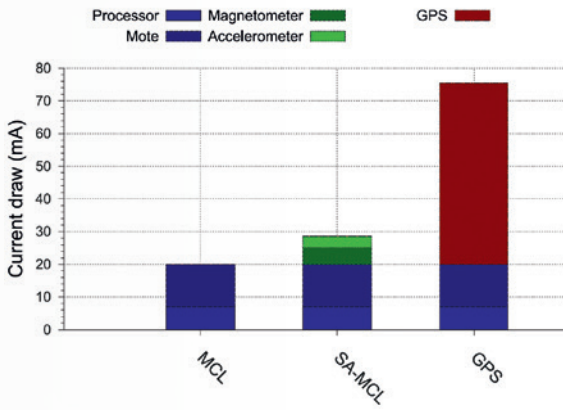


Figure 2.15: Current draw for MCL, SA-MCL and GPS.

Current Draw

One of the most quoted reasons for justifying the usage of localization algorithms instead of GPS is the high current draw of the latter. However, only few researchers provide a study of the energy consumption of their algorithms, often due to a missing real implementation. We measured the current flow of the fully assembled node when executing MCL, SA-MCL and when using GPS with a MASTECH M9803R multimeter. The current draw is measured over a period of 1 min with a frequency of 4 Hz. The gathered values are averaged to produce the final results which are shown in Figure 2.15. It can be found that the GPS device indeed is consuming a lot of power while the current draw introduced by our additional sensors is reasonably low. To be precise, the power consumption overhead of SA-MCL is about 8.6mA compared to 55.38mA for the GPS device. All in all, the total current draw for a node running MCL is about 20mA which includes powering the mote itself. For SA-MCL the current draw is about 28.59mA and 75.38mA for a mote equipped with GPS respectively. This means, the consumed power of a node localizing using GPS is about 3 times higher.

2.2.6 Further evaluation

Based on the data and GPS traces collected in the field test, we are able to run further offline experiments to analyze the performance of SA-MCL and MCL with different parameters. The logs stored in the nodes' flash memory are fed into an especially designed simulator based on the code running on the actual sensor nodes. In this way, it becomes possible to evaluate the performance of the algorithms when configured with different parameter sets.

The main parameters of interest are the radio range, which is represented by the RSSI cut-off value, and the speed value, which describes how fast the samples are moved in SA-MCL when no seed information is available. The quality of this parameter is crucial for the performance of SA-MCL. The value of V_{max} also depends on this parameter and should be set to twice the value of the speed parameter, to ensure that the randomly resampled particles in the particle filter on average spread with a speed corresponding to that of the speed parameter.

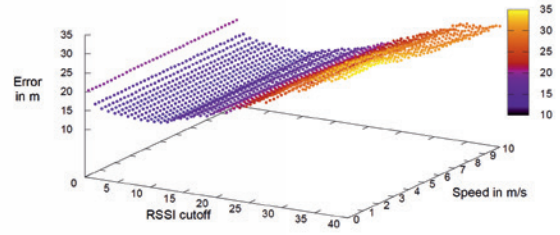


Figure 2.16: Absolute MCL error with varying parameters.

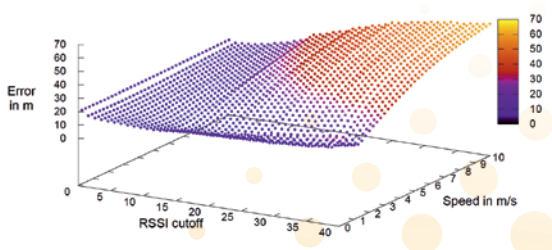


Figure 2.17: Absolute SA-MCL error with varying parameters.

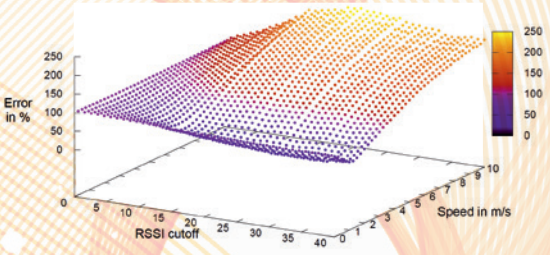


Figure 2.18: SA-MCL error percentage compared to MCL.

Figures 2.16 and 2.17 show the absolute localization error in meters for both algorithms with different parameters. SA-MCL is examined with speed values in the range 0.075m/s to 3m/s with a step size of 0.075m/s and RSSI cut-off values in the range [0, 40]. It can be found that SA-MCL is robust against a decrease in radio range as long as its speed value is configured to match the average speed of the sensor node while in motion. In contrast, the localization error of MCL strongly increases if the radio range and thus the seed density decreases.

Figure 2.18 shows the performance of SA-MCL in comparison to MCL. This comparative error is calculated as the ratio between the average absolute error of SA-MCL and MCL as given in Equation(2.1). Here, it can be seen that speed settings below the optimal value still at worst make SA-MCL's performance approach that of MCL, but do at no point make it perceivably worse, while well calibrated values lead to consistently superior performance.

$$\epsilon_{\text{loc }\%} = \frac{\epsilon_{\text{loc SA-MCL}}}{\epsilon_{\text{loc MCL}}} \times 100\,\% \tag{2.1}$$

2.3 Effcient localization algorithms for WSNs - PO-MCL

Path-Oriented Monte Carlo Localization (PO-MCL) exploits the movement behavior of nodes in certain applications where nodes mostly move on a set of static paths, which are initially unknown to the node. Our aim is to reduce the localization error and to account for situations where no seed information is available.

2.3.1 Key Ideas

In PO-MCL all nodes are maintaining a prediction grid, which divides the whole deployment area into grid cells. A node is always located in exactly one of these cells. A grid cell has exactly eight neighboring cells except for the cells at the borders of the deployment area. Each of the neighboring cells is labeled with its corresponding cardinal direction (N, NE, E, SW, ..., NW). We assign a value to all grid cells, which represents the probability of moving to this cell next. Based on observations from seed nodes the grid is updated such that the probability of the cell the node has moved to is increased and the values of all other neighboring

(a) Grid directions.

(b) Grid update process.

(c) Grid movement prediction.

Figure 2.19: Grid techniques in PO-MCL.

cells are decreased. As long as seed node information is available, the original MCL algorithm is executed except that samples are assigned the weight of their corresponding grid cell. Therefore, samples located in cells where the node currently is or has been before (i.e. cells corresponding to the path the node is moving on) will have a higher weight. In situations without seed information the node relies on the prediction grid information using an initial orientation determined by a magnetometer. The node will try to follow the path on the grid by looking for cells with high probabilities until seed information is available again.

2.3.2 Prediction Grid Construction

The dimensions of the grid cells are an important parameter as they mainly decide over the memory overhead of PO-MCL.We are adapting the size of the grid cells based on V_{max} and t_{check}. Since the maximum distance a node can travel between two localization estimations is $d = V_{max} \times t_{check}$ we also define the grid cell dimension as d as shown in in Figure 2.19a. It is obvious that with smaller values of d the resolution of the grid is growing and the traveled paths can be mapped to the grid with more precision. In the beginning all grid cells are initialized with the value 0.1, since no information about paths has been gathered yet and an arbitrary block of nine cells of the grid sums up to exactly 1. If a node can update its location estimation based on seed information, it checks if it has moved from its previous cell c_{t-1} to a different grid cell c_t. If yes, the probability of c_t is increased while the probability of all other neighboring cells of c_{t-1} is decreased. The amount of probability increase Δ_{inc} and decrease Δ_{dec} is determined based on the cell value of ct using Equations (2.2) and (2.3) where β is a tunable parameter regulating the level of increase.

$$\Delta_{\text{inc}} = \frac{\beta}{cellValue} \tag{2.2}$$

$$\Delta_{\text{dec}} = \frac{\Delta_{\text{inc}}}{8} \tag{2.3}$$

Cells with a low probability therefore will be increased faster than cells with high probabilities. A cell can have a maximum probability of 0.5 and we limit Δ_{inc} to be 0.2 at max. The effect of different values for β is shown in Figure 2.20. Smaller values of β will result in less probability increase and therefore slower grid convergence. An example of updating the grid is given in Figure 2.19b. Here, the node is moving North to c_t,

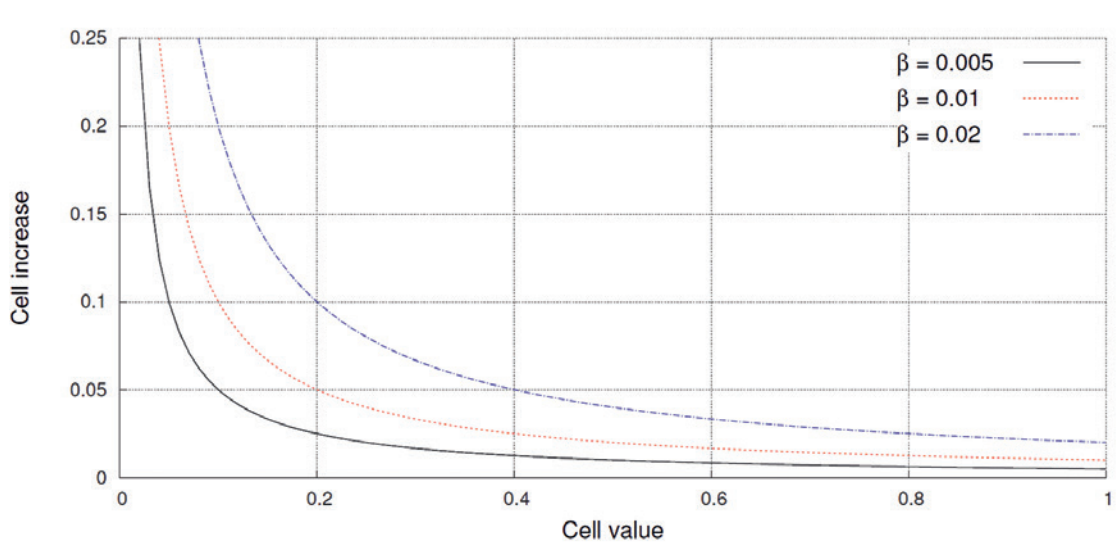
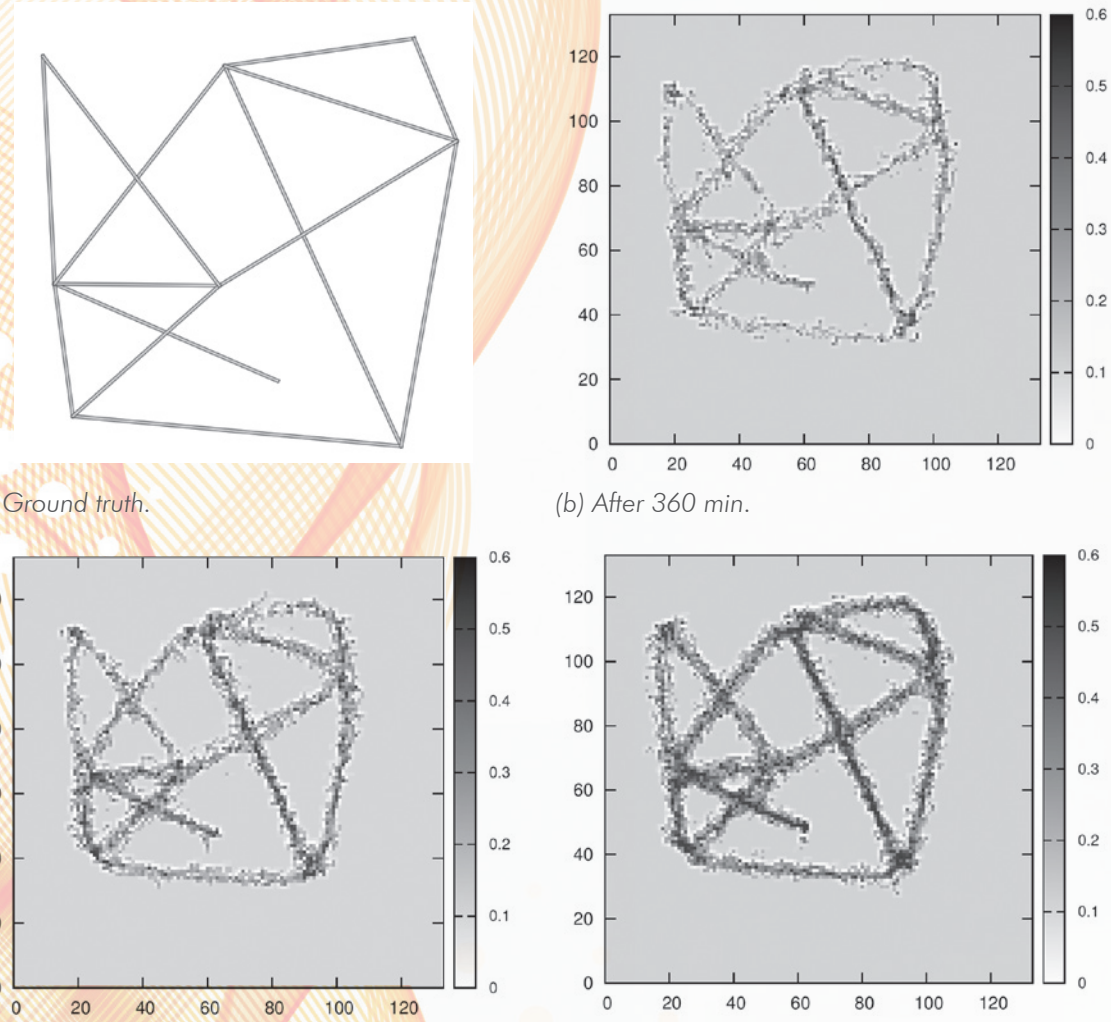


Figure 2.20: Effect of different values for β.



(a) Ground truth. (b) After 360 min.

(c) After 720 min. (d) After 1440 min.

Figure 2.21: Grid convergence of a random path scenario.

therefore we increase the probability of c_t and decrease the probability of all other neighbors of c_{t-1} as described above.

Over time, the prediction grid will converge to a representation of the traveled paths of the nodes. An example of the convergence process is shown in Figure 2.21. Ideally, the values of all eight neighbors of a cell sum up to 1.0. However, since cells are affecting each other this is unlikely to happen. In this respect the term probability used in this paper is not entirely accurate, but used for the ease of presentation.

2.3.3 Grid Movement Prediction

In standard MCL without seed information the sample set L will degenerate more and more over time as uncertainty of the real position of the node is growing. In PO-MCL our constructed prediction grid can account for these situations as we have generated additional information about the node's behavior. Using a magnetometer we can determine the node's current heading. As the sensor readings might not be exactly correct and the prediction grid is only a very rough representation of the ground truth model, we choose not only the

neighboring grid cell representing the direction determined by the magnetometer, but also its left and right neighbors. We then determine the highest probability of these three cells and move all samples by Δd in direction of the determined grid cell. Δd is calculated from the average of V_{min} and V_{max} as shown in Equation (2.4).

$$\Delta d = \frac{v_{min} + v_{max}}{2} \times t_{check} \quad (2.4)$$

Sample weights are reassigned based on the grid cells where the samples might have been moved to and the location estimation of the node is recalculated based on the updated sample set. The process is illustrated in Figure 2.19c: Assume the magnetometer determined a current direction of SW as indicated by the black arrow. The node selects its corresponding three neighboring cells and looks for the highest value. In the left example the result is consistent with the magnetometer direction (0.32). In the right example the cell in direction S has the highest probability (0.35). Therefore the predicted direction of the node indicated by the darker cell is S instead of SW.

2.3.4 Magnetometer Query Interval

Usage of a magnetometer is required for determining the heading of the node as soon as no more seed information is available. Since the magnetometer is consuming additional power it is desirable to use it as little as possible. Hardware tests in our experimental lab have shown that the time from powering on the magnetometer sensor to getting a first reading is negligibly low (10 ms) while GPS took almost 90 s to synchronize. Consequently, the magnetometer can be put into sleep mode and will be activated when required only. As the node might change its direction when reaching an intersection, it is necessary to query the magnetometer from time to time. The magnetometer query interval determines how often this is done. The most precise but also most power consuming solution would be to keep the magnetometer powered on. However, our simulation results show that it is possible to use the magnetometer only every 4th to 5th time while keeping a reasonable amount of precision. Detailed information on the trade-off between precision and magnetometer usage is given in Section 2.3.8.

```
1: procedure PO-MCL
2:     if |o_t| > 0 then
3:         MCL()
4:         magInterval = 0
5:         updatePredictionGrid()
6:     else
7:         if magInterval % magQuery == 0 then
8:             direction = getDirectionFromMagnetometer()
9:         else
10:            direction = getDirectionFromGrid(posOnGrid)
11:        end if
12:        moveSamplesByDirection(direction)
13:        magInterval++
14:    end if
15:    posOnGrid = determineGridCell()
16: end procedure
```

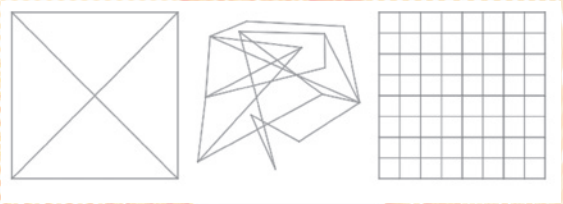
Figure 2.22: PO-MCL algorithm.

2.3.5 Formal Description

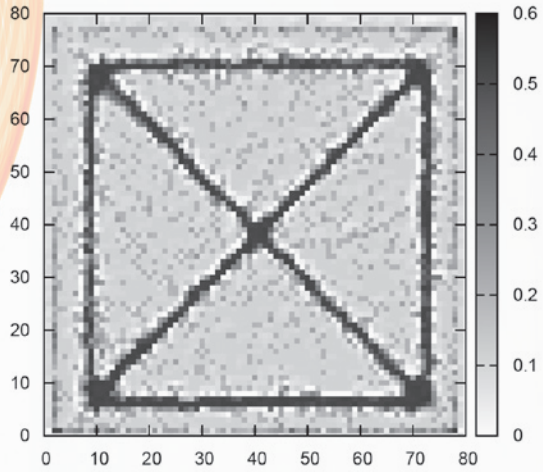
In Figure 2.22 we show a pseudo code listing of PO-MCL. Depending on the circumstance if the node receives one or more location announcements o_t from seed nodes, either the MCL algorithm will be executed or the grid will be used to update the sample set. In the former case the prediction grid is updated if the node has moved to a neighboring cell. As an additional improvement MCL has been slightly modified to weight samples according to the grid cell values. This is done by assigning the value of the grid cell where the sample is located to the sample weight. The parameter magQuery determines how often the magnetometer is checked and magInterval is a simple counter to keep track of the number of executed localization attempts in periods where no location announcements are heard. It is always reset to zero as soon as MCL can be executed again. After the sample set has been updated in either way the new grid cell in which the node is located is determined.

2.3.6 Implementation Details and Evaluation Results

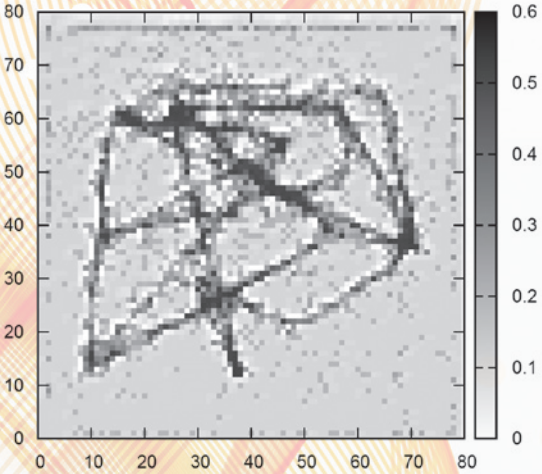
We implement and evaluate MCL and PO-MCL in Qualnet. MCL and existing improvements have only been validated in a custom Java simulation software [52] provided by the authors of MCL. We



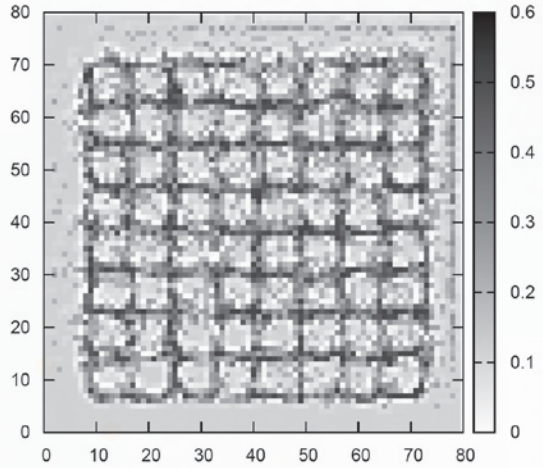
(a) Ground truth.



(b) Square scenario.



(c) Random paths 2 scenario.



(d) Grid scenario.

Figure 2.23: Evaluation scenarios.

strongly encourage researchers to use network simulation software well-established in research or industry to validate new algorithms in more reliable simulation environments. Custom network simulators often lack essential features like a full implementation of the protocol stack or path loss models for wireless transmissions and therefore might show distorted results. Nevertheless, we also modify the Java simulation environment used for evaluating MCL to support PO-MCL and receive similar results. Our code also provides a rudimentary GUI to visualize the ideas of MCL and PO-MCL which is suited for understanding the concepts of MCL and PO-MCL.

Since Qualnet only provides a random waypoint mobility model we had to add our path-based mobility model described previously. ESRI shapefiles are used as input files. A shapefile provides vector information for point, line and polygon data and is a common format used in geoinformatics. Consequently, shape files can be gener-

ated with a variety of tools including popular geographic information system software (GIS) like ESRI ArcGis. The shapefile is parsed and represented by an internal graph data structure. When executing a scenario the mobility model generates a random path walk on the graph based on the configuration parameters, e.g. starting vertex, V_{max}, pause time, etc.

2.3.7 Simulation Parameters

All experiments are conducted in a deployment area of 1000m x 1000 m. The default parameters unless otherwise stated for all experiments are given in Table 2.2.

In our simulations we investigate different path pattern scenarios, node velocities, numbers of seed nodes, magnetometer query intervals, radio ranges and quantities of maintained samples. All experiments use 200 ordinary nodes trying to localize themselves and mostly 25 or 50 seed nodes. We would like to point out that the number of seed nodes is set to a very low number compared to the size of the deployment area since we are interested in generating situations where no seed information has been acquired. Every experiment lasts 1 d (1440 min) to provide sufficient time for building the prediction grid. All experiments are repeated 10 times and averaged over all 200 ordinary nodes to get the final results.

$$\epsilon = \frac{\sum_{i=1}^{N} d(P_i, \hat{P}_i)}{N} \quad (2.5)$$

Table 2.2: Simulation parameters and default values

V_{max}	2m/s	tcheck	2.5s	N	25
magQuery	4	r	50m		

The localization error is given in multiples of r as the radio range is the main parameter for determining the absolute localization error (see Section 2.3.8). The error ε is given by averaging the error of all ordinary nodes as shown in Equation (2.5) where d(..) is the Euclidean distance between the real position P_i and the estimated position $\hat{P}_i$ of a node and N denotes the number of ordinary nodes.

2.3.8 Simulation Results

Different Path Characteristics

We study the effect of four different path characteristics to see how well PO-MCL adapts to these scenarios. A square with diagonals is a simple test scenario and provides only four vertices and therefore only has limited expressive power. More realistic scenarios are the random path scenarios in which we generate a set of vertices and connect them using arbitrary edges. The last scenario is a grid with a cell size of 100m² to test the behavior of PO-MCL in situations where a lot of changes in direction can be expected. The first random

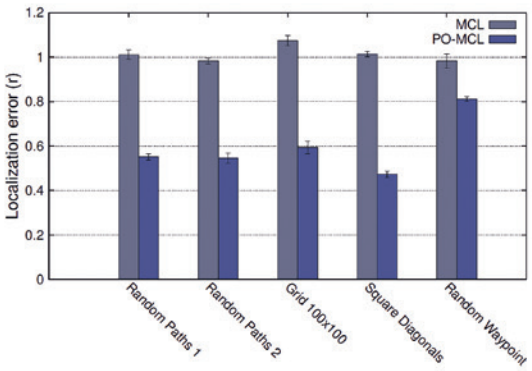


Figure 2.24: LE for different shapefile scenarios.

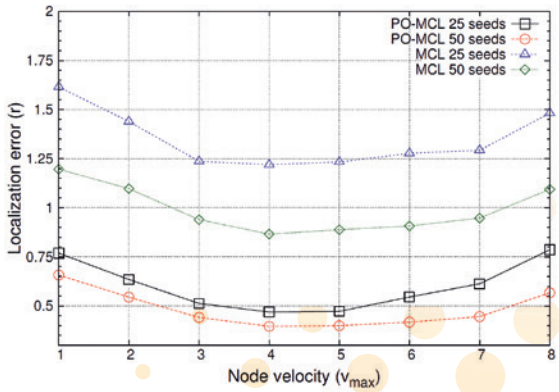


Figure 2.25: LE for different node velocities.

path scenario is already shown in Figure 2.21. The converged grids for the remaining scenarios are illustrated in Figure 2.23. The localization error for each scenario when executing MCL and PO-MCL is shown in Figure 2.24. It can be seen that PO-MCL outperforms MCL in all scenarios. Especially in the grid scenario, PO-MCL benefits from using the magnetometer and the prediction grid. On average, the localization error is reduced by about 40% and even halved in the square scenario. Additionally we studied the behavior of PO-MCL if the random waypoint model is applied. Although PO-MCL cannot efficiently use the prediction grid in this case, it still benefits from its magnetometer.

Different Node Velocities

In Figure 2.25 we investigate different node velocities. Both algorithms benefit from an increasing node velocity in the beginning, since periods without seed information are getting shorter as nodes are moving faster. However, since the radio range is kept the same (50 m) for higher node velocities of >4m/s the localization error is growing as nodes lose contact to seed nodes more often. Depending on the radio range the local minimum of the curve might be found at a different node velocity, but the characteristics of the curve will be the same for other simulation parameters.

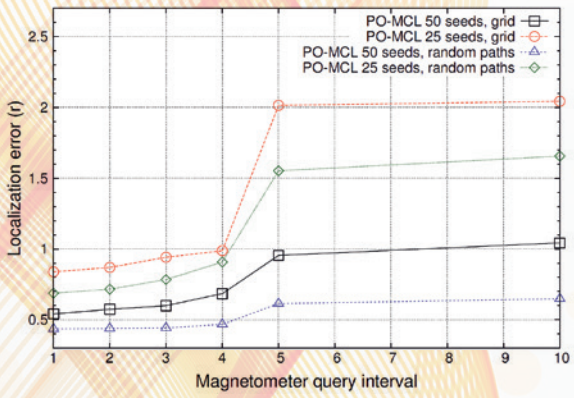


Figure 2.26: LE for different magnetometer intervals.

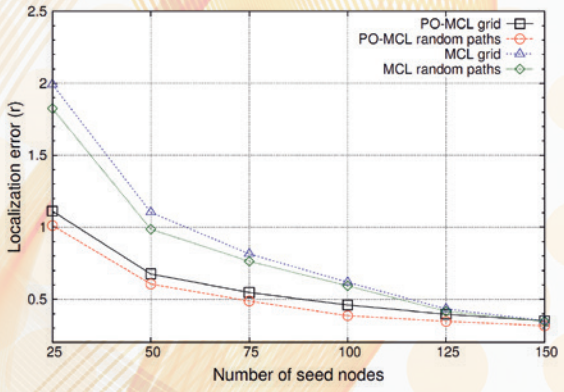


Figure 2.27: LE for different number of seed nodes.

Magnetometer Query Interval

A very important parameter for the power consumption of PO-MCL is the magnetometer query interval, which describes how frequent PO-MCL will query the magnetometer to retrieve the node's orientation. The results are shown in Figure 2.26. The best results will be achieved, if the magnetometer is kept on all the time. In this case the grid is only used for sample weighting and not for predicting the movement of a node. When increasing the magnetometer query interval, only slight increase of localization error can be found up to values of 3 to 4. Mainly for the grid scenario, higher values result in increase of the error, because changes of direction happen more often than detected by the magnetometer. The characteristics of the curve heavily depend on the other parameters. If t_{check} is decreased while keeping the same V_{max} PO-MCL will be executed in shorter intervals and therefore the magnetometer will be queried more often while the same distance is traveled by a node. Ergo, for smaller values of t_{check} bigger magnetometer query intervals are possible.

Number of Seed Nodes

A crucial parameter for the overall precision of a localization algorithm is the number of seed nodes available to an ordinary node on average. In Figure 2.27 we show how both MCL and PO-MCL behave if the number of seed nodes in the scenario is constantly reduced. While the localization error for MCL tremendously increases, PO-MCL can compensate missing location announcements by using the magnetometer and its grid prediction techniques. If seed nodes are constantly available, the localization error of MCL and PO-MCL will almost converge to a single

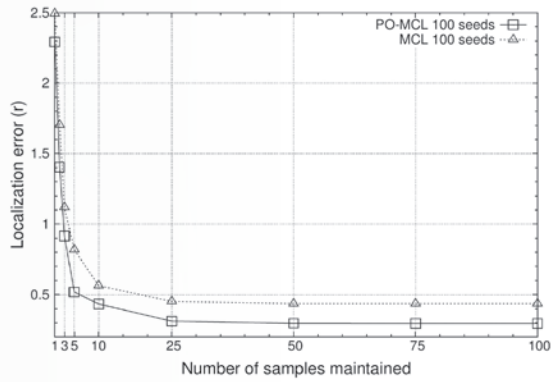


Figure 2.28: LE for different sample set sizes.

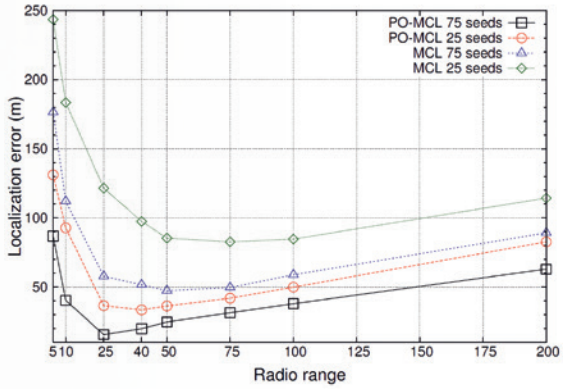


Figure 2.29: LE for different radio ranges.

curve, although PO-MCL still benefits little from its improved particle weighting.

Number of Samples Maintained

Computational time of both MCL and PO-MCL mainly depends on the number of maintained samples. There is a trade-off between the number of samples and localization error and it is desired to keep the localization error and the number of samples both as low as possible. In Figure 2.28 it can be seen that the localization error is rapidly decreasing when increasing the number of samples. This is due to the fact that larger sample sets can account for single imprecise samples. However, after a sample set size of 25 is reached there is no further improvement of the localization error. In contrast to Hu and Evens who recommend an optimal sample set size of 50 [52], we find this to be 25 which halves the computational overhead for the sample set.

Radio Range

Since MCL and PO-MCL both are connectivity-based algorithms the absolute localization error is mainly determined by the radio range r of the nodes. Smaller values of r will result in smaller absolute localization error given that a sufficient number of seed nodes is available. On the other hand with smaller r a bigger number of seed nodes is required to ensure the same level of seed node coverage. Figure 2.29 shows the absolute localization error when increasing the radio range.

2.4 Secure Neighbor Discovery for Wireless Ad Hoc Networks

2.4.1 Motivation

Attacking localization algorithms can be achieved in manifold ways and usually leads to inaccurate location estimations or no localization at all. Seed nodes are essential for almost all localization algorithms and therefore might be a target of great demand. A well-known attack and security issue in all wireless networks is the wormhole attack as illustrated in figure 2.30. Two physically different parts of the network are connected by an attacker using a very fast link. Packets issued in the part to the right are then transmitted via the attackers link and replayed in the network part to the left. As a consequence all nodes to the right are virtually becoming neighbors of node A. Applying this attack to a localization scenario means that seed information can be replayed in a different part of the network which in return will lead to inaccurate location estimations.

2.4.2 Proposed solution: RSND

To prevent this from happening it is desirable for nodes to be able to verify all of its neighbors being real neighbors. For this purpose, we proposed Robust Secure Neighbor Discovery (RSND). The full paper Robust and Scalable Secure Neighbor Discovery for Wireless Ad Hoc Networks has been presented at the IEEE International Conference on Communications (ICC), 2013. Compared to other solutions [57], [58], [59]

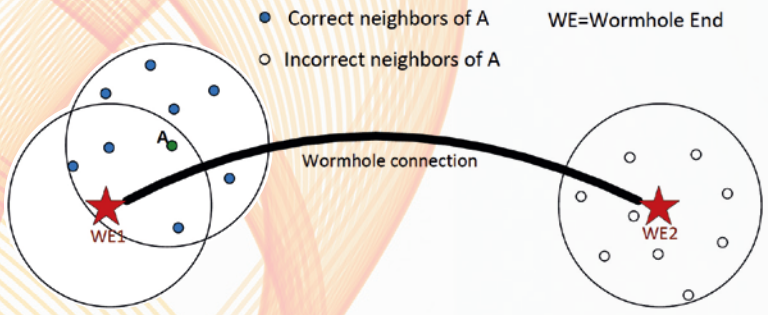


Figure 2.30: Wormhole Attack

RSND features a distributed approach to identify anomalies and detect incorrect neighbors of a node and therefore is not subject to single point of failure (SPF) attacks as in centralized solutions. RSND constructs an estimate of the local topology using the measured distances between neighboring nodes such that the effect of wrong ranging values is down-weighted. This allows a node to distinguish which neighbors of it are introduced by a wormhole. The mathematical base for the detection of anomalies (incorrect neighbors) is multi-dimensional scaling (MDS).

RSND constructs an estimate of the local topology using the measured distances between neighboring nodes such that the effect of wrong ranging values is down-weighted. This allows a node to distinguish which neighbors of it are introduced by a wormhole. In configuration of a set of points by MDS the goal is to find the relative positions of the points while preserving the given distances as far as possible. This goal is achieved by minimizing the sum of the residuals (differences between input and output distance values). We would like to modify this goal to minimizing the sum of weighted residuals in which the residual of incorrect links are down-weighted. The algorithm thus would intend to preserve the ranging values of correct links while letting the wrong ranging values be much more free to change because of having a down-weighted role in the error function. After doing such a visualization, we will be able to detect the wormhole affected links by checking which links have changed more significantly.

2.4.3 Evaluation and results

In our simulations nodes are uniformly distributed in a 2D field having radio (and ultrasound) range of R = 300m. The ranging error has a Gaussian distribution with the variance of e. In each set-up there is a single two-end wormhole attack located in arbitrary places in the network. First, we evaluate the correctness of the algorithm in determining if there is any wormhole attack in a node's neighborhood. The evaluation criteria are false-positive and false-negative metrics. We investigate the effect of ranging error standard deviation from e = 0 to e = 2.5.

According to Figure 2.31, with larger ranging error the false-negative rate shows a very slight change while the false-positive rate grows more significantly. This is because a high ranging error itself can introduce inconsistencies to the ranging information of a correct set. The second part of simulations investigates the performance of RSND in detecting the correct/incorrect neighbors of a wormhole affected node. First with N = 12 we increase the number of nodes tunneled by the wormhole as M = 1 to 11. We run the experiment in the neighborhood of 80 nodes for each M. The result is shown in Figure 2.32 in terms of the accuracy of detections. As can be seen, by increasing M for a fixed N the accuracy of the algorithm degrades. This is because with a fixed N as M grows the wrong part of the ranging information grows while the consistent part is kept fixed.

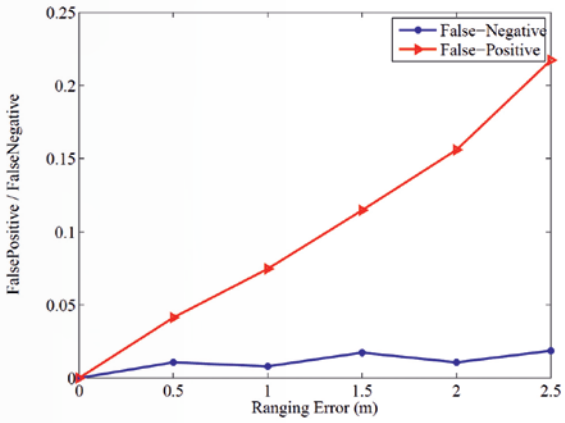


Figure 2.31: Accuracy of RSND in detecting the presence of wormhole

With random M from $1 \leq M \leq N/2$ values and running over 150 neighborhoods, we got the results presented in Figure 2.33. It shows that the performance is not significantly affected by the network degree and it even improves slightly with N. This is because although M is chosen to be proportional to N and thus the ratio of true and wrong ranging information does not change, as R is fixed, for larger N values more nodes are located in one node's radio range (higher node density) and therefore more neighbors of the verifier node would be neighbors themselves and can measure their distances. Thus, the ratio of unknown edges would be less and with such a richer ranging information

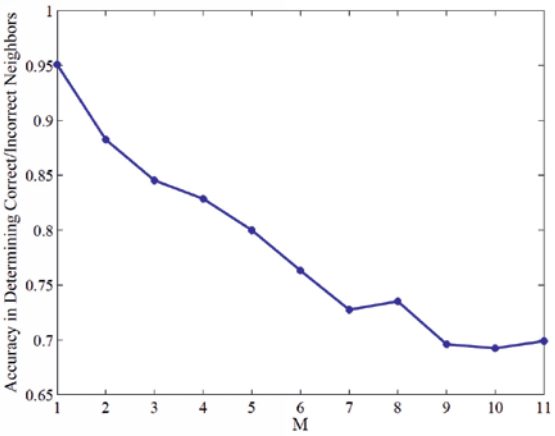


Figure 2.32: Accuracy of RSND in determining correct and incorrect neighbors of a node for N=12

the topology could be configured more correctly by the algorithm. This indicates the very high scalability of RSND protocol.

2.5 Lightweight Integrity Verifcation Architecture for ICN

Named data networking (NDN) is a new paradigm for the future Internet wherein interest and data packets carry content names rather than the current IP paradigm of source and destination addresses. Security is built into NDN by embedding a public key signature in each data packet to enable verification of authenticity and integrity of the content. However, existing heavyweight signature generation and verification algorithms prevent universal integrity verification among NDN nodes, which may result in content pollution and denial of service attacks. Furthermore, caching and location-independent content access disables the capability of a content provider to control content access, e.g., who can cache a content and which end user or device can access it. We propose a lightweight integrity verification architecture (LIVE), an extension to the NDN protocol, to address these two issues seamlessly. LIVE enables universal content signature verification in NDN with lightweight signature generation and verification algorithms. Further, it allows a content provider to control content access in NDN nodes by selectively distributing integrity verification tokens to authorized nodes. We evaluate the effectiveness of LIVE with open source CCNx project. Our study shows that LIVE only incurs average 10% delay in accessing contents. Compared to traditional public key signature schemes, the verification delay is reduced by more than 20 times in LIVE. The results of this work titled "LIVE: Lightweight Integrity Verification and Content Access Control for Named Data Networking" was published in IEEE Transactions on Information Forensics and Security [10].

2.6 Distributed/Decentralized Approaches

2.6.1 Decentralizing the provision of virtualized services

Networks are becoming increasingly complex and service providers incorporate additional functionality (e.g., firewall, spam filtering, QoS) in the network to protect, manage and improve

service performance. Software Defined Networking (SDN) seeks to manage the network with the help of a (logically) centralized control plane. We observe that current SDN solutions pre-translate policy (what) into forwarding rules at specific switches (where). We argue that this choice limits the dynamicity, flexibility and reliability that a software based network could provide. Information Centric Networking (ICN) shifts the focus of networks away from being predominantly location oriented communication environments. We believe ICN can significantly improve the flexibility for network management. In this paper, we focus on one of the problems of network management – service chaining – the steering of flows through the different network functions needed, before it is delivered to the destination. We propose Function-Centric Service Chaining (FCSC), a solution that exploits ICN to provide flexibility in managing networks that utilize virtualization to dynamically place functions in the network as required. We use a simple topology as well as a real-world topology on a simulation environment to compare the performance of FCSC and a more traditional SDN solution. We show that FCSC reacts to failures with fewer packet drops, adapts to new middleboxes more quickly, and maintains less state in the network. The results of this work was published in the 1st ACM Conference on Information-Centric Networking (ICN 2014), Paris, France, September 2014 [9]. The conference is highly competitive with an acceptance ratio of 17%. This work received the "Best Paper Award" at the conference.

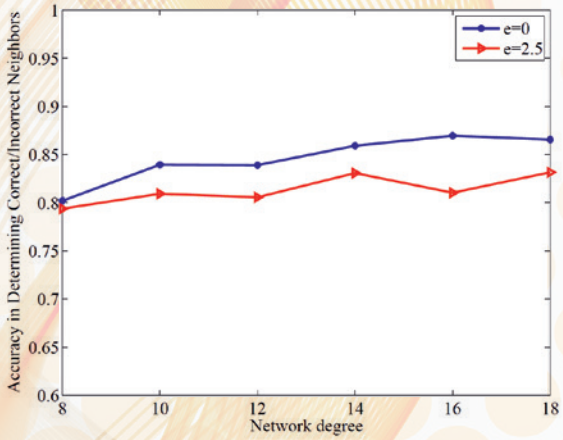


Figure 2.33: Accuracy of RSND in determining correct and incorrect neighbors of a node for N=8 to 18

2.6.2 Decentralized Social Networks Based Privacy

Concomitant with the tremendous growth of online social networking (OSN) platforms are increasing concerns from users about their privacy and the protection of their data. As user data management is usually centralized, OSN providers nowadays have the unprecedented privilege to access every user's private data, which makes large-scale privacy leakage at a single site possible. One way to address this issue is to decentralize user data management and replicate user data at individual end-user machines across the OSN. However, such an approach must address new challenges. In particular, it must achieve high availability of the data of every user with minimal replication overhead and without assuming any permanent online storage. At the same time, it needs to provide mechanisms for encrypting user data, controlling access to the data, and synchronizing the replicas. Moreover, it has to scale with large social networks and be resilient and adaptive in handling both high churn of regular participants and attacks from malicious users. While recent works in this direction only show limited success, we introduce a new, decentralized OSN called the Self-Organized Universe of People (SOUP). SOUP employs a scalable, robust and secure mirror selection design and can effectively distribute and manage encrypted user data replicas throughout the OSN. An extensive evaluation by simulation and a real-world deployment shows that SOUP addresses all aforementioned challenges. The results of this work was published in the 15th Annual ACM/IFIP/USENIX Middleware Conference (Middleware 2014), Bordeaux, France, December 2014 [4]. The conference is highly competitive and has an acceptance rate of 18.75 %.

2.6.3 Geo Distributed Cloud

Geo-distributed clouds provide an intriguing platform to deploy online social network (OSN) services. To leverage the potential of clouds, a major concern of OSN providers is optimizing

the monetary cost spent in using cloud resources while considering other important requirements, including providing satisfactory quality of service (QoS) and data availability to OSN users. In this paper, we study the problem of cost optimization for the dynamic OSN on multiple geo-distributed clouds over consecutive time periods while meeting pre-defined QoS and data availability requirements. We model the cost, the QoS, as well as the data availability of the OSN, formulate the problem, and design an algorithm named "cosplay". We carry out extensive trace driven simulation with a large-scale real-world Twitter trace over 10 geo-distributed clouds all across the US. Our results show that, while always ensuring the QoS and the data availability as required, our approach can reduce much more one-time cost than the state-of-the-art methods, and it can also significantly reduce the accumulative cost when continuously evaluated over 48 months, with OSN dynamics comparable to real-world cases. The results of this work was published in the premier IEEE/ACM Transactions on Networking in September 2014 [8].

2.6.4 Recommendation based differentiation between network paths

This work investigates the rating of network paths, i.e. acquiring quantized measures of path properties such as round-trip time and available bandwidth. Comparing to fine grained measurements, coarse-grained ratings are appealing in that they are not only informative but also cheap to obtain. Motivated by this insight, we firstly address the scalable acquisition of path ratings by statistical inference. By observing similarities to recommender systems, we examine the applicability of solutions to recommender system and show that our inference problem can be solved by a class of matrix factorization techniques. A technical contribution is an active and progressive inference framework that not only improves the accuracy by selectively measuring more informative paths but also speeds up the convergence for available bandwidth by incorporating its measurement methodology. Then, we investigate the usability of rating-based network measurement and inference in applications. A case study is performed on whether locality awareness can be achieved for overlay networks of Pastry and BitTorrent using inferred ratings. With the help of simulation based evaluations and analytical approaches, we show that such coarse-grained knowledge can improve the performance of peer selection and that finer granularities do not always lead to larger improvements. Parts of this work was published in the premier IEEE/ACM Transactions on Networking in July 2014 [12].

2.6.5 Content Retrieval At the Edge: A Social-aware and Named Data Cooperative Framework

Recent years with the popularity of mobile devices have witnessed an explosive growth of mobile multimedia contents which dominate more than 50% of mobile data traffic. This significant growth poses a severe challenge for future cellular networks. As a promising approach to overcome the challenge, we advocate Content Retrieval At the Edge, a content-centric cooperative service paradigm via device-to-device (D2D) communications to reduce cellular traffic volume in mobile networks. By leveraging the Named Data Networking (NDN) principle, we propose sNDN, a social-aware named data framework to achieve efficient cooperative content retrieval. Specifically, sNDN introduces Friendship Circle by grouping a user with her close friends of both high mobility similarity and high content similarity. We construct NDN routing tables conditioned on Friendship Circle encounter frequency to navigate a content request and a content reply packet between Friendship Circles, and leverage social properties in Friendship Circle to search for the final target as inner-Friendship Circle routing. The evaluation results demonstrate that sNDN can save cellular capacity greatly and outperform other content retrieval schemes significantly. This solution could also help process data emerging from IoT devices. Part of this work as published in the IEEE Transactions on Emerging Topics in Computing journal [13].

2.6.6 Comparison of Naming Schema in ICN

Information-Centric Networking (ICN) treats content as a first-class entity - each content has a unique identity and ICN routers forward traffic based on content identity rather than the locations of the content. This provides benefits like

dynamic request routing, caching and mobility support. The choice of naming schema (flat vs. hierarchical) is a fundamental design choice in ICN which determines the functional separation between the network layer and the application layer. With hierarchical names, the network layer is cognizant of the semantics of hierarchical names. Name space management is also part of network layer. ICN architectures using flat names leave these to the application layer. The naming schema affects the performance and scalability of the network in terms of forwarding efficiency, routing table size and name space size. This paper provides both qualitative and quantitative comparison on the two naming schemas using these metrics, noting that they are interdependent. We seek to understand which naming schema would be better for a high-performance, scalable ICN architecture. This will help us develop a naming solution that is more applicable for resource constrained IoT nodes. Part of this work was published in the 22nd IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN 2016) [16].

2.6.7 Triadic Closure Pattern Analysis and Prediction in Social Networks

We study the problem of group formation in online social networks. In particular, we focus on one of the most important human groups - the triad – and try to understand how closed triads are formed in dynamic networks, by employing data from a large microblogging network as the basis of our study. We formally define the problem of triadic closure prediction and conduct a systematic investigation. The study reveals how user demographics, network characteristics, and social properties influence the formation of triadic closure. We also present a probabilistic graphical model to predict whether three persons will form a closed triad in a dynamic network. Different kernel functions are incorporated into the proposed graphical model to quantify the similarity between triads. Our experimental results with the large microblogging dataset demonstrate the effectiveness (+10 % over alternative methods in terms of F1-Score) of the proposed model for the prediction of triadic closure formation. This study will be extended to understand group formation in social networks based IoT environments. Part of this work was published in the IEEE Transactions on Knowledge and Data Engineering [14].

2.6.8 Cloud-Assisted Data Fusion and Sensor Selection for Internet-of-Things

The Internet of Things (IoT) is connecting people and smart devices on a scale that was once unimaginable. One major challenge for the IoT is to handle vast amount of sensing data generated from the smart devices that are resource-limited and subject to missing data due to link or node failures. By exploring cloud computing with the IoT, we present a cloud based solution that takes into account the link quality and spatio-temporal correlation of data to minimise energy consumption by selecting sensors for sampling and relaying data. We propose a multi-phase adaptive sensing algorithm with belief propagation protocol (ASBP), which can provide high data quality and reduce energy consumption by turning on only a small number of nodes in the network. We formulate the sensor selection problem and solve it using both constraint programming (CP) and greedy search. We then use our message passing algorithm (belief propagation) for performing inference to reconstruct the missing sensing data. ASBP is evaluated based on the data collected from real sensors. The results show that while maintaining a satisfactory level of data quality and prediction accuracy, ASBP can provide load balancing among sensors successfully and preserves 80% more energy compared with the case where all sensor nodes are actively involved. Parts of this work was published in the IEEE Internet of Things Journal [15].

3 Final Words

This report gave an overview of our completed projects and presented our results achieved so far. We listed all of the published papers, live demonstrations and talks given at international research conferences. Our ongoing work and future projects are described in our application for extension.

Publication list

[1] S. Hartung, K. Rieck, A. Kellner, and D. Hogrefe, “Monte Carlo Localization for

Path-Based Mobility in Wireless Sensor Networks." in In IEEE Wireless Communications and Networking Conference (WCNC), Dha, Qatar, April 2016.

[2] S. Hartung, A. Bochem, A. Zdziarstek, and D. Hogrefe, "Applied Sensor-Assisted Monte Carlo Localization for Mobile Wireless Sensor Networks," in In International Conference on Embedded Wireless Systems and Networks (EWSN), Graz, Austria, February 2016.

[3] A. Bochem, A. Zdziarstek, S. Hartung, and D. Hogrefe, "Demo: Dead Reckoning for Monte Carlo Localization in Low Seed Density Scenarios," in In International Conference on Embedded Wireless Systems and Networks (EWSN), Graz, Austria, February 2016.

[4] D. Koll, J. Li, and X. Fu, "SOUP: An Online Social Network By The People, For The People," in Proc. of 15th Annual ACM/IFIP/USENIX Middleware Conference (Middleware 2014), Bordeaux, France. (acceptance rate: 18.75 %), December 2014.

[5] S. Hartung, H. Brosenne, and D. Hogrefe, "Practical RSSI Long Distance Measurement Evaluation inWireless Sensor Networks," in The 2013 IEEE Conference on Wireless Sensors (ICWiSe 2013), Kuching, Malaysia, December 2013.

[6] S. Hartung, A. Kellner, A. Bochem, and D. Hogrefe, "Sensor-Assisted Monte Carlo Localization for Wireless Sensor Networks," in 6th IFIP International Conference on New Technologies, Mobility and Security (NTMS) - Poster + Demo Session, Dubai, UAE, April 2014.

[7] S. Hartung, S. Taheri, and D. Hogrefe, "Sensor-Assisted Monte Carlo Localization for Wireless Sensor Networks," in 6th IEEE International Conference on Cyber Technology (CYBER), Hong Kong, HK, June 2014.

[8] L. Jiao, J. Li, T. Xu,W. Du, and X. Fu, "Optimizing Cost for Online Social Networks on Geo-Distributed Clouds," IEEE/ACM Transactions on Networking (accepted), September 2014.

[9] M. Arumaithurai, J. Chen, E. Monticelli, X. Fu, and K. K. Ramakrishnan, "Exploiting ICN for Flexible Management of Software-Defined Networks," in Proc. of 1st ACM Conference on Information-Centric Networking (ICN 2014), Paris, France (Won the Best Paper Award, acceptance rate: 17%), September 2014.

[10] Q. Li, X. Zhang, Q. Zheng, R. Sandhu, and X. Fu, "LIVE: Lightweight Integrity Verification and Content Access Control for Named Data Networking," IEEE Transactions on Information Forensics and Security, IEEE, to appear., October 2014.

[11] S. Taheri and D. Hogrefe, "Robust and Scalable Secure Neighbor Discovery for Wireless Ad Hoc Networks," in The 2013 IEEE International Conference on Communications (ICC 2013), Budapest, Hungary, June 2013.

[12] W. Du, Y. Liao, N. Tao, P. Geurts, X. Fu, and G. Leduc, "Rating Network Paths for Locality-Aware Overlay Construction and Routing," IEEE/ACM Transactions on Networking, July 2014.

[13] L. Pu, X. Chen, J. Xu, and X. Fu, "Content Retrieval At the Edge: A Social-aware and Named Data Cooperative Framework," IEEE Transactions on Emerging Topics in Computing, June 2016.

[14] H. Huang, J. Tang, L. Liu, J. Luo, and X. Fu, "Triadic Closure Pattern Analysis and Prediction in Social Networks," IEEE Transactions on Knowledge and Data Engineering, December 2015.

[15] F. H. Bijarbooneh,W. Du, E. Ngai, X. Fu, and J. Liu, "Cloud-Assisted Data Fusion and Sensor Selection for Internet-of-Things," IEEE Internet of Things Journal, June 2016.

[16] S. S. Adhatarao, J. Chen, M. Arumaithurai, X. Fu, and K. K. Ramakrishnan, "Comparison of Naming Schema in ICN," in IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 2016.

[17] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Tech. Rep., 2000.

[18] K. Fall, "A delay-tolerant network architecture for challenged internets," in SIGCOMM. ACM, 2003.

[19] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in MobiHoc. ACM, 2004.

[20] C. Boldrini, M. Conti, and A. Passerella, "Context and resource awareness in opportunistic network data dissemination," in WoWMoM. IEEE, 2008.

[21] J. Wu, M. Xiao, and L. Huang, "Homing spread: Community home-based multicopy routing in mobile social networks," in INFOCOM. IEEE, 2013.
[22] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," ser. WDTN. ACM, 2005.
[23] I. Psaras, L. Wood, and R. Tafazolli, "Delay-/disruption-tolerant networking: State of the art and future challenges," Tech. Rep., 2012.
[24] Z. Feng, "Data dissemination in delay tolerant networks," Ph.D. dissertation, School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, 2012.
[25] K. Scott, The MITRE Corporation, S. Burleigh, and NASA Jet Propulsion Laboratory, "RFC 5050: Bundle protocol specification," IETF, Tech. Rep., 2007.
[26] F. Li and J. Wu, "Mops: Providing content-based service in disruption-tolerant networks," in ICDCS. IEEE, 2009.
[27] A. Ker¨anen, J. Ott, and T. K¨arkk¨ainen, "The ONE Simulator for DTN Protocol Evaluation," in SIMUTools. ICST, 2009.
[28] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe, "A survey of mobility in information-centric networks: Challenges and research directions," in INFOCOM NOM Workshop. ACM, 2012.
[29] G. Tyson, J. Bigham, and E. Bodanese, "Towards an information-centric delay tolerant network," in NOMEN. IEEE, 2013.
[30] G. Grassi, D. Pesavento, L.Wang, G. Pau, R. Vuyyuru, R.Wakikawaa, and L. Zhang, "Vehicular inter-networking via named data," in HotMobile Poster, 2013.
[31] F. Neves dos Santos, B. Ertl, C. Barakat, T. Spyropoulos, and T. Turletti, "Cedo: Content-centric dissemination algorithm for delay-tolerant networks," in MSWiM. ACM, 2013.
[32] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. K. Ramakrishnan, "COPSS: An efficient content oriented pub/sub system," in ANCS, 2011.
[33] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in SIGCOMM, 2007.
[34] J. Chen, M. Arumaithurai, and Me, "G-COPSS: A content centric communication infrastructure for gaming," in ICDCS, 2012.
[35] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in CoNEXT, 2009.
[36] L. Zhang, D. Estrin, J. Burke, V. Jacobson, and J. Thornton, "Named Data Networking (NDN) Project," PARC, Tech. Report NDN-0001, 2010.
[37] W. Fenner, D. Srivastava, K. K. Ramakrishnan, D. Srivastava, and Y. Zhang, "XTreeNet: Scalable Overlay Networks for XML Content Dissemination and Querying," in WCW, 2005.
[38] I. Psaras, L. Saino, M. Arumaithurai, K. Ramakrishnan, and G. Pavlou, "Name-Based Replication Priorities in Disaster Cases," in INFOCOM NOM Workshop, 2014.
[39] Focus Group-DR&NRR, "Technical report on telecommunications and disaster mitigation," ITU-T, Tech. Rep., 2013.
[40] K. Cho, C. Pelsser, R. Bush, and Y.Won, "The japan earthquake: The impact on traffic and routing observed by a local isp," in SWID. ACM, 2011.
[41] E. Monticelli, B. Schubert, M. Arumaithurai, X. Fu, and K. Ramakrishnan, "An information centric approach for communication in disaster situations," in LANMAN. IEEE, 2014.
[42] J. D. Hunter, "Matplotlib: A 2d graphics environment," Computing In Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.
[43] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of information (netinf) - an information-centric networking architecture," Comput. Commun., vol. 36, no. 7, pp. 721–735, Apr. 2013.
[44] D. Trossen and G. Parisis, "Designing and realizing an information-centric internet," IEEE Communications Magazine, vol. 50, no. 7, pp. 60–67, 2012.
[45] I. Amundson and X. D. Koutsoukos, "A survey on localization for mobile wireless sensor networks," in Proceedings of the 2Nd International Conference on Mobile Entity Localization and Tracking in GPS-less

Environments, ser. MELT'09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 235–254. [Online]. Available: http://dl.acm.org/citation.cfm?id=1813141.1813162

[46] G. Han, H. Xu, T. Duong, J. Jiang, and T. Hara, "Localization algorithms of wireless sensor networks: a survey," Telecommunication Systems, vol. 52, no. 4, pp. 2419–2436, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11235-011-9564-7

[47] O. G. Adewumi, K. Djouani, and A. M. Kurien, "RSSI Based Indoor and Outdoor Distance Estimation for Localization in WSN," in 14th IEEE International Conference on Industrial Technology (ICIT), 2013.

[48] A. T. Parameswaran, M. I. Husain, and S. Upadhyaya, "Is RSSI a Reliable Parameter in Sensor Localization Algorithms - An Experimental Study," in 28th International Symposium on reliable distributed Systems, 2009.

[49] N. A. Dieng, C. Chaudet, M. Charbit, L. Toutain, and T. B. Meriem, "Experiments on the RSSI as a Range Estimator for Indoor Localization," in 5th International Conference on New Technologies, Mobility and Security (NTMS), 2012.

[50] Y. Chen and A. Terzis, "Calibrating RSSI Mesaurements for 802.15.4 Radios," in Proceedings of the Seventh European Conference on Wireless Sensor Networks (EWSN), 2010, pp. 256–271.

[51] G. Teng, K. Zheng, and W. Dong, "MA-MCL: Mobile-Assisted Monte Carlo Localization for Wireless Sensor Networks." IJDSN, vol. 2011, 2011.

[52] L. Hu and D. Evans, "Localization forWireless Sensor Networks," in 10th Annual International Conference on Mobile Computing and Networking (MobiCom 2004), Phildadelphia, USA, 2004, pp. 45–57.

[53] S. Zhang, J. Cao, C. Li-Jun, and D. Chen, "Accurate and energy-efficient range-free localization for mobile sensor networks," Mobile Computing, IEEE Transactions on, vol. 9, no. 6, pp. 897–910, 2010.

[54] M. Rudafshani and S. Datta, "Localization in wireless sensor networks," in Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on, 2007, pp. 51–60.

[55] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," Wireless Communications & Mobile Computing (WCMC): Special Issue On Mobile Ad Hoc Networking: Research, Trends And Applications, vol. 2, pp. 483–502, 2002.

[56] E. Kulla, M. Ikeda, L. Barolli, F. Xhafa, and J. Iwashige, "A survey on manet testbeds and mobility models," Computer Science and Convergence, vol. 114, pp. 651–657, 2012.

[57] R. Shokri, M. Poturalski, G. Ravot, P. Papadimitratos, and J.-P. Hubaux, "A practical secure neighbor verification protocol for wireless sensor networks," in Proceedings of the Second ACM Conference on Wireless Network Security, ser. WiSec '09. New York, NY, USA: ACM, 2009, pp. 193–200.

[58] R. Stoleru, H. Wu, and H. Chenji, "Secure neighbor discovery in mobile ad hoc networks," in Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on, Oct 2011, pp. 35–42.

[59] W. Znaidi, M. Minier, and J.-P. Babau, "Detecting wormhole attacks in wireless networks using local neighborhood information," in Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on, Sept 2008, pp. 1–5.

[60] Advanticsys. (2014, Nov.) Advanticsys website. [Online]. Available: http://www.advanticsys.com/

[61] Memsic. (2014, Nov.) Memsic website. [Online]. Available: http://www.memsic.com/

[62] TinyOS Alliance. (2014, Nov.) TinyOS website. [Online]. Available: http://www.tinyos.net/

[63] QualNet by SCALABLE Network Technologies Inc. (2014, Nov.) SCALABLE Network Technologies website. [Online]. Available: http://www.web.scalablenetworks.com/

[64] Matlab by The MathWorks Inc. (2014, Nov.) The MathWorks website. [Online]. Available: http://www.mathworks.com/

Project data

The project was funded from SWZ with 0.75 TV-L E13 staff positions from April 2013 to March 2016 at the site Göttingen. Involved scientists are:

Prof. Dr. Dieter Hogrefe
Telematics Group
Institute of Computer Science
University of Göttingen

Dr. Mayutan Arumaithurai
Computer Networks Group
Institute of Computer Science
University of Göttingen

Prof. Dr. Xiaoming Fu
Computer Networks Group
Institute of Computer Science
University of Göttingen

Dr. Jiachen Chen
Dr. Jiachen Chen
Computer Networks Group
Institute of Computer Science
University of Göttingen

Salke Hartung, M. Sc.
Telematics Group
Institute of Computer Science
University of Göttingen

Dr. Somayeh Taheri
Telematics Group
Institute of Computer Science
University of Göttingen

Decomposition of open queueing networks with batch service

Wiebke Klünder, Thomas Hanschke, Horst Zisgen

1. Introduction

The importance of analysis of non-product form networks by applying approximations has increased steadily in recent years. The most important strategy approach is given by the decomposition method. The decomposition method enables an isolated treatment of the nodes within the network. The method is particularly applied in planning and optimization of production systems by calculating characteristics of each node. In this article a decomposition method will be presented serving primarily as a pre-evaluation tool. If the calculated characteristics move in acceptable ranges Monte-Carlo simulation can be performed.

Until now the decomposition method for open queueing networks with batch service in the single class case was developed by using the approach of Pujolle/Ai[1]. The first aim is to expand the method to common approximations developing the approach to batch service and to transfer them to the developed method of [1]. This includes the approximately formation of the merging of the input streams by Kühn[2] and Chylla[3] as well as the approximation of the flow by Whitt [4] and Chylla. The second aim is the generalization of the method to the multi class case of networks with batch service.

2. Decomposition method

The decomposition method consists of three steps (figure 1) and can be applied only if the utilization of all queueing systems in a network is high. This assumption is necessary to satisfy the condition that the departure streams building also the arrival streams of subsequent stations in the network form renewal processes and enables an isolated observation of the queueing systems. In the first step of the method overlying arrival streams were merged by forming the superposition. Next the flow of single queueing systems are approximately computed and the last step split the determined flow in respect to the transition probabilities inside the network. If the phases of the decomposition method are inserted into each other a system of linear equations is built. The solutions of the system of linear equations provide quantities for calculations of performance measures of the single queueing systems within the network.

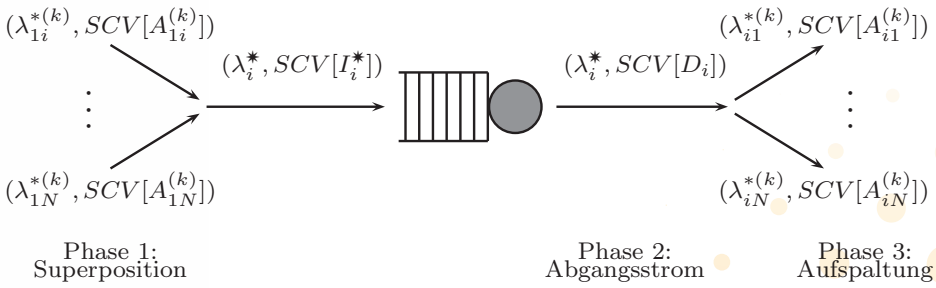


Figure 1: decomposition method

3. Single class open queueing networks

3.1 Description of the model

The open network consists of 1 to N nodes numbering successively and presenting $GI^X/GI^{((b,b))}/c$ queueing systems. Jobs arrive in groups of size b_0 from outside the network according a renewal process with arrival rate $\lambda_0 < \infty$ and the squared coefficient of variation $SCV[I_0] < \infty$. $0 \le p_{ij} \le 1$ describes the transition probability that an arriving batch reaches node *j* from node *i* and

$$\sum_{i=1}^{N} p_{0i} = 1$$

applies meaning jobs enter the network from outside. Each queueing system have c_*i* identical servers, an unlimited waiting room and the FCFS queueing discipline. The service starts if a batch of the required size b_i was generated at node *i*. The service times are presented as random variables S_i with service rate $\mu_i < \infty$ and $SCV[S_i] < \infty$. It is assumed that the interarrival and service times are independent. After a complete service of a batch it will arrive in a form of a batch to the subsequent node according to the transition probabilities. X_i is described by an integer random variable and represents the input size of the groups at node *i*. The first and second moment are calculated by

$$E[X_i] = \frac{\sum_{j=0}^{N} b_j \tau_j p_{ji}}{\sum_{j=0}^{N} \tau_j p_{ji}} \quad \text{and} \quad E[X_i^2] = \frac{\sum_{j=0}^{N} b_j^2 \tau_j p_{ji}}{\sum_{j=0}^{N} \tau_j p_{ji}}.$$

τ_i denotes the relative throughput and can be determined by the modified traffic equation:

$$\tau_i = p_{0i}\frac{b_0}{b_i} + \sum_{j=1}^{N} \tau_j p_{ji} \frac{b_j}{b_i} \quad \text{and} \quad \tau_0 := 1.$$

The modified arrival rate of batches can be calculated by $\lambda_i^* = \lambda_0 \cdot \tau_i$ and the modified utilization by $\rho_i^* = \lambda_i^* / c_i \mu_i < 1$.

3.2 Decomposition

Merging

There exist three different approaches to form the superpositions of the arrival streams. The first approach was developed by Pujolle/Ai[5]. The counting process representing the incoming jobs and incoming groups is respectively described by knowledge of the asymptotic behavior of renewal processes:

$$SCV[I_i] \approx \left(\sum_{j=0}^{N} \tau_j \cdot p_{ji}\right)^{-1} \cdot \sum_{j=0}^{N} \tau_j \cdot p_{ji} \cdot SCV[A_{ji}].$$

Chylla uses the approach in order to approximate the splitting of the departure stream (see splitting):

$$SCV[I_i] \approx 1 + \sum_{j=0}^{N} \frac{\lambda_j^*}{\lambda_i^*} p_{ji} (SCV[A_{ji}] - 1).$$

The approach of Kühn is based on a case-by-case analysis which depends on the values of $SCV[A_{ji}]$ (see splitting):

$$SCV[I_i] \approx 2 \cdot \frac{t_1 + t_2}{(t_1 \cdot t_2)^2} (I^1 + I^2 + I^3 + I^4)$$

$$t_j = \frac{1}{p_{ji}\tau_j}, \quad j = 1, 2.$$

The components $I^1, \ldots, I^4$ are either a composition of hypoexponentially, hyperexponentially distributed sub-processes or a mixture. For details see [2].

After the interarrival times of the single jobs has been determined the interarrival times of batches will be approximated by [6]:

$$SCV[I_i^*] \approx \frac{E[X_i]}{b_i} (SCV[X_i] + SCV[I_i]),$$

where $SCV[X_i]$ is calculated by $(E[X_i^2]/E[X_i]^2)-1$.

Flow

There are fundamentally two approaches to approximate the departure stream in a non-product form network. Pujolle/Ai and Chylla use the approach

$$D_i \approx \begin{cases} \frac{S_i}{c_i} & \text{with probability } \rho_i^* \\ \frac{S_i}{c_i} + I_i^* & \text{with probability } 1 - \rho_i^* \end{cases}$$

and it results for Pujolle/Ai according to the calculation of the first and second moment of the process D_i

$$SCV[D_i] \approx \rho_i^{*2} SCV[S_i] +$$

$$(1 - \rho_i^*) SCV[I_i^*] + (1 - \rho_i^*)\rho_i^*$$

and a slightly modified version of Chylla

$$SCV[D_i] \approx 1 + P_i^{*2}(SCV[S_i] - 1) +$$

$$(1 - P_i^*)(SCV[I_i^*] - 1),$$

where P_i is described by the Erlang-C formula. Whitt and Kühn use the approach of Marshall [7] to approximate the departure stream basing on Lindley's recursion of waiting times. The formula

$$SCV[D_i] \approx 1 + (1 - \rho_i^{*2})(SCV[I_i^*] - 1) +$$

$$\frac{\rho_i^{*2}}{\sqrt{c_i}}(SCV[S_i] - 1).$$

represents the approximation of Whitt and Kühn developed the approximation

$$SCV[D_i] \approx SCV[I_i^*] + 2\rho_i^{*2} SCV[S_i] -$$

$$\rho_i^{*2}(SCV[I_i^*] + SCV[S_i])g_{KLB},$$

where g_{KLB} is the correction factor given by Krämer and Langenbach-Belz [8].

Splitting

The splitting of the departure stream in accordance with the transition probabilities can be considered as a Bernoulli experiment. After completion of service at node *j* jobs are directed to node *i* with probability p_{ji} and with probability $1\text{-}p_{ji}$ they are routed elsewhere. The number of the first batch be directed to node *i* is geometrically distributed. The first moment and the variances of the splitting process are calculated by using Wald's equation respectively the Blackwell-Girshick equation. The squared coefficient of variation results by:

$$SCV[A_{ji}] = 1 + p_{ji}(SCV[D_j] - 1).$$

If the phases are inserted successively into each other a system of linear equations is formed whose solution provide the squared coefficient of variation of the interarrival times of the batches. Characteristics like the average number of individual jobs in the queueing systems can be determined by the modified formula of Allen and Cunneen [6]:

$$E[N_i] \approx E[Z_{\infty,i}] + b_i \cdot E[Q]_{GI/GI/c}(\rho_i^*, SCV[I_i^*],$$

$$SCV[S_i])_{g_{KLB}} + b_i c_i \rho_i^* + E[Y_{\infty,i}]$$

3.3 Numerical results

Due to the independence of the phases the presented approaches can be arbitrarily combined, e. g. merging will be approximated by Kühn and flow by Whitt. The benchmark which was done in regard to [1] was used to evaluate the decom-

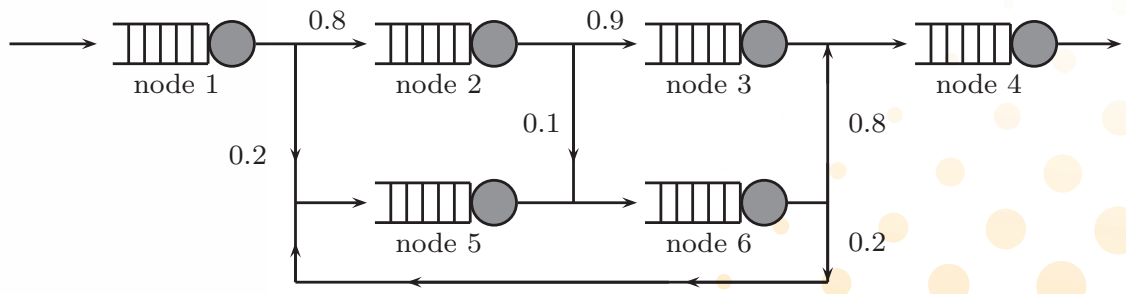


Figure 2: reference model

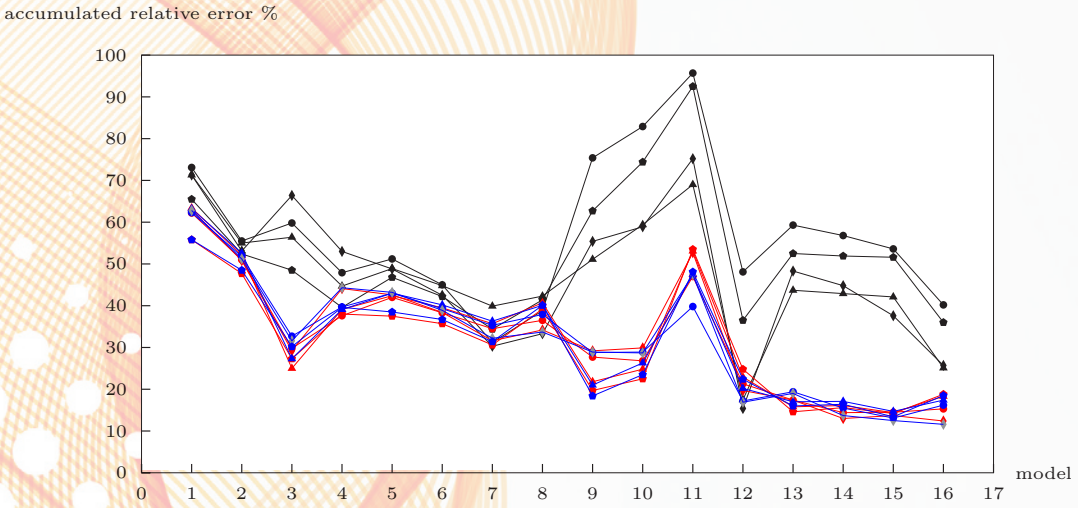


Figure 3: accumulated error, black: merging by Chylla, blue: merging by Pujolle/Ai, red: merging by Kühn

position method using all possible combinations of the presented approximation approaches. The figure 2 shows the reference network. All in all 16 cases were investigated in detail differing in the characteristics of the utilization, batch sizes, number of servers and $SCV[S_i]$.

The relative error is the discrepancy between calculated approximate values by the decomposition method and the mean of the simulation results. Figure 3 shows all possible combinations. The merging by Chylla (black) irrespective of the choice of the approximation of the flow has the inexact results. The formation of the superpositions of the arrival streams by Pujolle/Ai (blue) and Kühn (red) provide similar results.

Exemplarily case 11 ([1] benchmark: table 3, case 3) will be evaluated shortly. Table 1 presents the parameterization of the open queueing network. Figure 4 summarizes the results and shows the relative errors at the nodes. The ratio of mean input batch size ($E[X_4]=6.048$) and batch size $b_4=5$ at node 4 explains the increased discrepancies. These situation affects an overestimation of the $SCV[I_i^*]$ and at last of the approximate characteristics. A similar phenomenon occurs at the node 2 ($E[X_2]=3>b_2=1$). This case shows that the approximations of the merging by Pujolle/Ai and Kühn are robust. In contrast Chylla's approximation caused larger errors at node 2. It is also clearly shows that Chylla's approximation combined with the flow developed by Whitt and Kühn does not work well if

table 1: queueing network with $\lambda_0 = 2$, $SCV[I_0] = 1$, $b_0 = 1$, $E[S_1] = 3.6$, $E[S_2] = 0.45$, $E[S_3] = 4.5$, $E[S_4] = 4$, $E[S_5] = 8.889$ and $E[S_6] = 11.4$

Node	b_i	c_i	p_i	SCV $[S_i]$
1	3	3	0.8	0.25
2	1	1	0.72	0.25
3	10	1	0.648	0.25
4	5	2	0.8	0.25
5	2	4	0.6	0.25
6	3	3	0.887	0.25

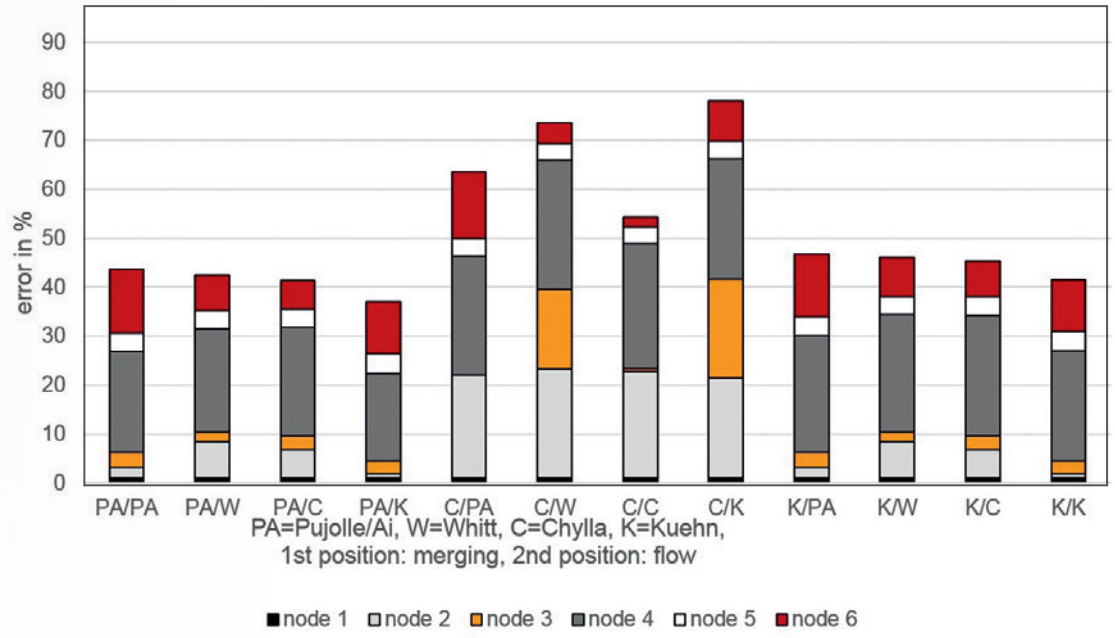


Figure 4: case 11, relative errors

there exist large changes of batch size (node 3). If there are larger batch size changes the merging by Kühn revealed weaknesses (node 4).

3.4 Conclusions

All approaches yield acceptable results being useful as pre-evaluation for optimization. Generally it has been shown that the approximation of the merging by Pujolle/Ai and Kühn are robust. The approximately approach of Chylla on the other hand caused in cases of larger batch size changes high errors ([1] benchmark: cases 9-16, table 3 and 4). The approach of Kühn which has a complex case distinction is more difficult to handle than the approach of Pujolle/Ai.

The two approaches of the approximation of the flow yield similar results which could be expected since the approaches provide similar approximation formulas in case of high utilizations.

4. Multi class open queueing networks

4.1 Description of the model

The open network to be examined consists of 1 to N nodes numbering successively and presenting $GI^{X(k)}/GI^{b(k),b(k)}/c$ queueing systems. Jobs arrive in groups of size $b_o^{(k)}$ from outside the network according a renewal process with arrival rate $\lambda_0^{(k)}<\infty$ and the squared coefficient of variation $SCV[I_0^{(k)}]<\infty$. $0 \le p_{ij}^{(k)} \le 1$ describes the transition probability that an arriving batch belonging to a product class k, $k \in \{1,\ldots,K\}$, reachs node *j* from node *i* and

$$\sum_{i=1}^{N} p_{0i}^{(k)} = 1$$

applies meaning jobs enters the network from outside. Each queueing system have c_i identical servers, an unlimited waiting room and the FCFS queueing discipline. The service starts if a batch of the required size $b_i^{(k)}$ was generated at node *i*. The mixing of jobs having different product classes in one batch is forbidden. Furthermore it is assumed that all servers serve batches of all product classes. The service times of a product class *k* are presented as random variables $S_i^{(k)}$ with service rate $\mu_i^{(k)}<\infty$ and $SCV[S_i^{(k)}]<\infty$. It is assumed that the interarrival and service times of one product class are independent and pairwise independent

between the product classes. After a complete service of a batch it will arrive in a form of a batch to the subsequent node according to the transition probabilities. $X_i^{(k)}$ is described by an integer random variable and represents the input size of the groups of product class *k* at node *i*. The first and second moment are calculated by

$$E[X_i^{(k)}] = \frac{\sum_{j=0}^{N} b_j^{(k)} \tau_j^{(k)} p_{ji}^{(k)}}{\sum_{j=0}^{N} \tau_j^{(k)} p_{ji}^{(k)}} \quad \text{and} \quad E[X_i^{(k)2}] = \frac{\sum_{j=0}^{N} b_j^{(k)2} \tau_j^{(k)} p_{ji}^{(k)}}{\sum_{j=0}^{N} \tau_j^{(k)} p_{ji}^{(k)}}.$$

$\tau_i^{(k)}$ denotes the relative throughput of a product class k and can be determined by the modified traffic equation:

$$\tau_i^{(k)} = p_{0i}^{(k)} \frac{b_0^{(k)}}{b_i^{(k)}} + \sum_{j=1}^{N} \tau_j^{(k)} p_{ji}^{(k)} \frac{b_j^{(k)}}{b_i^{(k)}} \quad \text{and} \quad \tau_0^{(k)} := 1.$$

The modified arrival rate of batches can be calculated by $\lambda_i^{*(k)} = \lambda_0^{(k)} \cdot \tau_i^{(k)}$ and the modified utilization by $\rho_i^{*(k)} = \lambda_i^{*(k)} / c_i \, \mu_i^{(k)}$.

4.2 Decomposition

Merging

In order to form the superposition of the arrival streams [5],[1] the counting process $(N_{(i,t)}^{(k)})_{t \in R+}$ for the associated renewal process $I_i^{(k)}$ is first described:

$$Var[N_{i,t}^{(k)}] = \sum_{j=0}^{N} p_{ji}^{(k)} \tau_j^{(k)} SCV[I_i^{(k)}] t.$$

The counting process can also be represented as the splitting process of the departure streams of the predecessors:

$$Var[N_{i,t}^{(k)}] = \sum_{j=0}^{N} p_{ji}^{(k)} \tau_j^{(k)} SCV[A_{ji}^{(k)}] t.$$

From the equalization of the descriptions the wanted measure $SCV[I_i^{(k)}]$ results

$$SCV[I_i^{(k)}] = \left(\sum_{j=0}^{N} \tau_j^{(k)} p_{ji}^{(k)} \right)^{-1} \sum_{j=0}^{N} \tau_j^{(k)} p_{ji}^{(k)} SCV[A_{ji}^{(k)}].$$

Flow

The departure stream can be approximated by the approach of Pujolle/Ai [5], [1]:

$$D_i \approx \begin{cases} \frac{S_i^*}{c_i} & \text{with probability } \rho_i^* \\ \frac{S_i^*}{c_i} + I_i^* & \text{with probability } 1 - \rho_i^*. \end{cases}$$

S_i^* describes the aggregated average service times, I_i^* the aggregated modified interarrival times and ρ_i^* the aggregated modified utilizations:

$$\rho_i^* = \sum_{k=1}^{K} \rho_i^{*(k)} < 1.$$

The first moment of the process D_i is calculated by

$$E[D_i] = \rho_i^* \frac{E[S_i^*]}{c_i} = E[I_i^*]$$

and the second moment is given by

$$E[D_i^2] = \rho_i^* \frac{E[S_i^{*2}]}{c_i} + (1 - \rho_i^*) E\left[\left(\frac{S_i^*}{c_i} + I_i^* \right)^2 \right].$$

Then the wanted measure can be determined:

$$SCV[D_i] = \frac{E[D_i^2]}{E[D_i]^2} - 1 = \rho_i^{*2} SCV[S_i^*] +$$

$$(1 - \rho_i^*) SCV[I_i^*] + (1 - \rho_i^*) \rho_i^*.$$

$SCV[I_i^*]$ describes the aggregated form of the squared coefficient of variation of the interarrival times of the batches:

$$SCV[I_i^*] \approx \sum_{k=1}^{K} \frac{\lambda_i^{*(k)}}{\lambda_i^*} \frac{E[X_i^{(k)}]}{b_i^{(k)}} (SCV[X_i^{(k)}] + SCV[I_i^{(k)}]).$$

To be able to specify the squared coefficient of variation of the aggregated service times the average service rate must be first calculated:

$$\mu_i^* = \left(\sum_{k=1}^{K} \frac{\lambda_i^{*(k)}}{\lambda_i^*} \frac{1}{c_i \mu_i^{(k)}} \right)^{-1} \quad \text{with}$$

$$\lambda_i^* = \sum_{k=1}^{K} \lambda_i^{*(k)}.$$

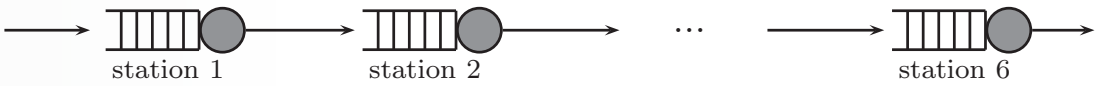


Figure 5: reference model 1

The squared coefficient of variation of the aggregated service times is then computed by

$$SCV[S_i^*] = -1 + \sum_{k=1}^{K} \frac{\lambda_i^{(k)}}{\lambda_i^*} \left(\frac{\mu_i^*}{c_i \mu_i^{(k)}} \right)^2 (SCV[S_i^{(k)}] + 1).$$

Splitting

The splitting of the departure stream D_i can be considered as a Bernoulli experiment. After completion of service at node *j* jobs are directed to node *i* with probability $p_{ji}^{(k)}$ and with probability $1\text{-}p_{ji}^{(k)}$ they are routed elsewhere. The number of the first batch be directed to node *i* is geometrically distributed. Since the process D_i is assumed as a renewal process the expected value and the variance of the splitting process $A_{ji}^{(k)}$ can be determined by Wald's equation respectively Blackwell-Girshick equation:

$$E[A_{ji}^{(k)}] = E\left[\sum_{v=1}^{V} D_j\right] = \frac{1}{p_{ji}^{(k)}} E[D_j]$$

$$Var[A_{ji}^{(k)}] = Var\left[\sum_{v=1}^{V} D_j\right] = \frac{1}{p_{ji}^{(k)}} Var[D_j] + \frac{1 - p_{ji}^{(k)}}{p_{ji}^{(k)2}} E[D_j].$$

The measure $SCV[A_{\_ji}^{(k)}]$ can be calculated by

$$SCV[A_{ji}^{(k)}] = \frac{Var[A_{ji}^{(k)}]}{E[A_{ji}^{(k)}]^2} = 1 + p_{ji}^{(k)} (SCV[D_j] - 1).$$

If the phases are inserted successively into each other a system of linear equations is formed whose solution provide the squared coefficient of variation of the interarrival times:

$$SCV[I_i^{(k)}] = \left(\sum_{j=0}^{N} \tau_j^{(k)} p_{ji}^{(k)} \right)^{-1} \sum_{j=0}^{N} \tau_j^{(k)} p_{ji}^{(k)} \left(1 + p_{ji}^{(k)} \left(\rho_j^{*2} SCV[S_j^*] + \right. \right.$$

$$\left. \left. (1 - \rho_j^*) \left[\sum_{k=1}^{K} \frac{\lambda_j^{*(k)}}{\lambda_j^*} \frac{E[X_j^{(k)}]}{b_j^{(k)}} (SCV[X_j^{(k)}] + SCV[I_j^{(k)}]) \right] + (1 - \rho_j^*) \rho_j^* \right) - 1 \right).$$

Characteristics like the average number of individual jobs in the system of the various queueing systems can be determined by the modified formula of Allen and Cunneen [9] and the correction factor of Krämer and Langenbach-Belz:

$$E[N_i] \approx E[Z_{\infty,i}^*] + b_i^* E[Q]_{GI/GI/c}$$

$$(\rho_i^*, SCV[I_i^*], SCV[S_i^*])_{g_{KLB}} + b_i^* c_i \rho_i^* + E[Y_{\infty,i}^*]$$

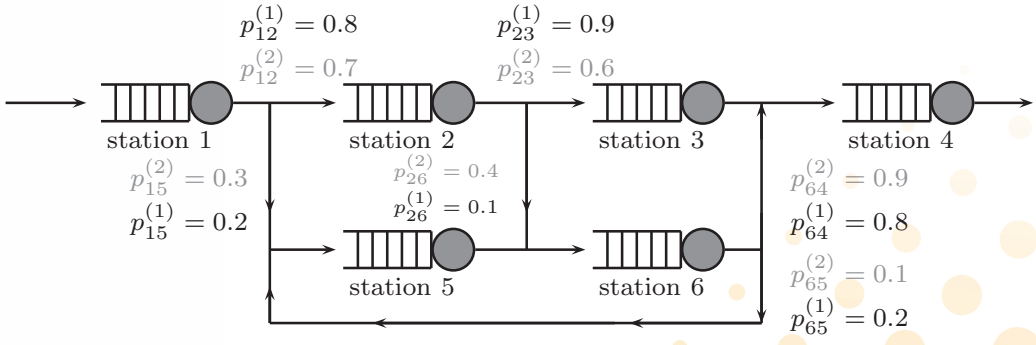


Figure 6: reference model 2

with

$$E[Z^{*}_{\infty,i}] = \sum_{k=1}^{K} E[Z^{(k)}_{\infty,i}],$$

,

$$b^{*}_{i} = \sum_{k=1}^{K} \frac{\lambda^{*(k)}_{i}}{\lambda^{*}_{i}} b^{(k)}_{i}$$

and

$$E[Y^{*}_{\infty,i}] \sum_{k=1}^{K} E[Y^{(k)}_{\infty,i}].$$

To calculated the average number of individual jobs belonging to one product class the formula has the form

$$E[N^{(k)}_{i}] \approx \frac{\lambda^{*(k)}_{i}}{\lambda^{*}_{i}} E[Z^{(k)}_{\infty,i}] + \frac{\lambda^{*(k)}_{i}}{\lambda^{*}_{i}} b^{*}_{i} E[Q]_{GI/GI/c}(\rho^{*}_{i}, SCV[I^{*}_{i}],$$

$$SCV[S^{*}_{i}])_{gKLB} + b^{(k)}_{i} c_{i} \rho^{(k)}_{i} + \frac{\lambda^{*(k)}_{i}}{\lambda^{*}_{i}} E[Y^{(k)}_{\infty,i}]$$

4.3 Numerical results

The realized simulation study includes two reference models. The first represents a six-stage series with three product classes.

The second reference model represents a network with two product classes.

The simulation models constructing of the reference models can be divided into six different model classes:

class 1	$b^{(k)}_{i}$ are equal at a station i, $E[X^{(k)}_{i}] \leq b^{(k)}_{i}$
class 2	$b^{(k)}_{i}$ are equal at a station i, $E[X^{(k)}_{i}] > b^{(k)}_{i}$
class 3	$b^{(k)}_{i}$ are equal at a station i, $E[X^{(k)}_{i}] \stackrel{!}{>} b^{(k)}_{i}$
class 4	$b^{(k)}_{i}$ are different at a station i, $E[X^{(k)}_{i}] \leq b^{(k)}_{i}$
class 5	$b^{(k)}_{i}$ are different at a station i, $E[X^{(k)}_{i}] > b^{(k)}_{i}$
class 6	$b^{(k)}_{i}$ are different at a station i, $E[X^{(k)}_{i}] \stackrel{!}{>} b^{(k)}_{i}$.

$<^{!}$ means that the $E[X^{(k)}_{i}]$ are significantly larger than $b^{(k)}_{i}$. Each model class consists of six different simulation models only differing in the $SCV[S^{(k)}_{i}]$.

A comparison method is used to investigate the additional value of the generalization of the decomposition (multi class case). The comparison method is the decomposition describing in [1] and section 3. The method uses average input values, merging and flow by Pujolle/Ai and the approximation results were afterward converted to the product classes.

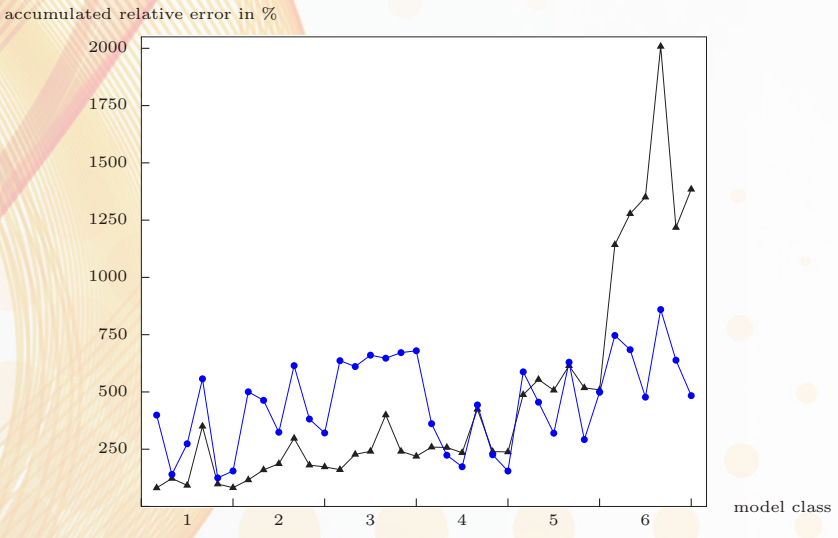


Figure 7: series – accumulated relative error of the generalization (black) and of the comparison method (blue)

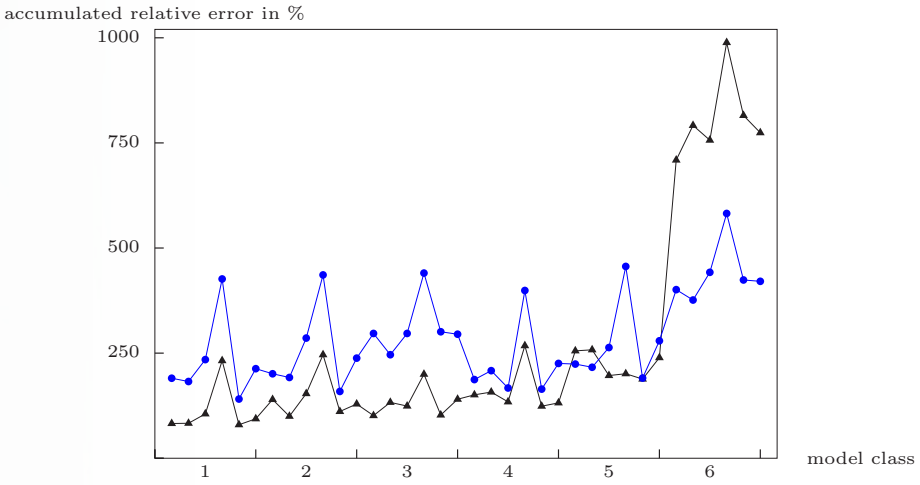


Figure 8: network – accumulated relative error of the generalization (black) and of the comparison method (blue)

The figures 6 and 7 show the accumulated relative error. The errors are calculated from the discrepancies between means of the simulations and the approximations of the $E[N_i^{(k)}]$. The black data points represent the generalization of the decomposition and the blue ones the comparison method. It is shown that the generalization is almost superior to the comparison method for the first four model classes. The model classes 5 and 6 are better approximated by the simpler modeling with the method from [1]. However, none of the decompositions for problems corresponding to the model classes 5 and 6 is useable since the quality of the approximations are too low.

Exemplary a simulation model which is based on the second reference model and belongs to the model class 3 will be analyzed in detail. Table 2 shows the parametrization of the network and figure 9 presents the relative error of the product classes at a node i for the generalization of the decomposition.

High errors for both product classes are observed at nodes 3 and 5. The nodes are common that they have high values of the $SCV[I_i^*]$ at node 3 and 5 ($SCV[I_3^*]=4.39$, $SCV[I_5^*]=2.58$). This is caused by the ratio of the mean input group size and the batch size ($E[X_3\hat{}((k))]=5>b_{\_3}\hat{}((k))=2$,

table 2: queueing network with $\lambda_0^{(1)} = 1$, $\lambda_0^{(2)} = 1$, $b_0^{(1)} = 1$, $b_0^{(2)} = 1$, $SCV[I_0^{(1)}] = 1$, $SCV[I_0^{(2)}] = 1$

Network - 2 product classes										
Knoten i	c_i	ρ_i^*	product classes							
			1				2			
			b_i^1	μ_i^1	ρ_i^1	$SCV[S_i^{(1)}]$	b_i^2	μ_i^2	ρ_i^2	$SCV[S_i^{(2)}]$
1	3	0.9	3	0.19	0.6	1.2	3	0.37	0.3	0.3
2	1	0.9	5	0.8	0.2	0.3	5	0.2	0.7	1.2
3	5	0.9	2	0.14	0.5	1.2	2	0.105	0.4	0.3
4	2	0.9	7	0.09	0.8	0.3	7	0.714	0.1	1.2
5	4	0.9	1	0.23	0.3	1.2	1	0.152	0.6	0.3
6	3	0.9	4	0.04	0.7	0.3	4	0.269	0.2	1.2

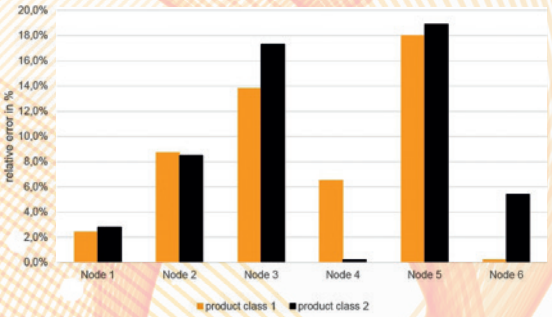


Figure 9: relative error of the generalization of the decomposition

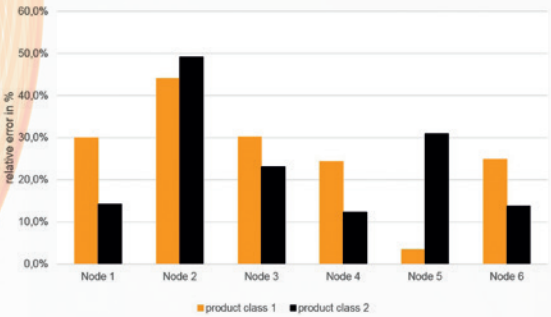


Figure 10: relative error of the established decomposition

$E[X_5^{(k)}]=3>b_5^{(k)}=1$) effecting an overestimation. By creation of a probability mass in the 0 the process is not a renewal process being an assumption for the decomposition method. The unbalanced utilizations on the product classes have no negative effects. The small inaccuracies at the nodes 4 on product class 1 and node 6 on product class 2 can not be clearly explained.

Figure 10 represents the relative errors of the well-established method for the model which is described in table 2. It can be clearly seen that the generalized method provides significantly better results than the comparison method.

4.4 Conclusions

The generalized decomposition by treating the individual product classes leads to good approximations and is almost superior to well-established decomposition methods. This makes the method to an useful pre-evaluation tool for optimization and planning. There is no method of decomposition for problems that deal with queueing networks with different batch sizes of the product classes at a station and the situation that the average input group size of a product class is larger than the respective batch size. The condition of renewal processes is violated and the approximations partly differ strongly from the exact results.

5. References

[1] Th. Hanschke and H. Zisgen, Queueing networks with batch service, European Journal of Industrial Engineering, 5(3):313-326, 2011

[2] Paul J. Kühn, Approximate analysis of general queueing networks by decomposition, IEEE Transaction on Communications, 27(1):113-126, 1979

[3] Peter Chylla, Zur Modellierung und approximativen Leistungsanalyse von Vielteilnehmer-Rechensystemen, Dissertation, TU München, 1986

[4] Ward Whitt, The queueing network analyzer, Bell system technical Journal, 62(9): 2779-2815, 1983

[5] G. Pujolle and W. Ai, A solution for multi-server and multicall open queueing networks, INFOR, 24(3):221-230, 1986

[6] Th. Hanschke, Approximations for the mean queue length of the GI^X/GI^((b,b))/c queue, Operations Research Letters, 34(2): 205-213, 2006

[7] Kneale T. Marshall, Some inequalities in queueing systems, Operation Research, 16(3):651-668, 1968

[8] W. Krämer and M. Langenbach-Belz, Approximate formulae for the delay in the queueing system GI/G/1, 8th International Teletraffic Congress, 1976

[9] T. Gröger, Warteschlangensysteme mit Gruppenankünften, Gruppenbedienung und heterogenen Kunden, Diplomarbeit, TU Clausthal, 2007

Project data

The project is funded from SWZ with a 0.5 TV-L E13 staff positions since April 2014 at the site Clausthal. Involved scientists are:

Prof. Dr. Thomas Hanschke
Research Group Stochastische Modelle in den Ingenieurwissenschaften
Institute of Applied Stochastics and Operations Research
Clausthal University of Technology

Dipl.-Wirt.-Inf. Wiebke Klünder
Research Group Stochastische Modelle in den Ingenieurwissenschaften
Institute of Applied Stochastics and Operations Research
Clausthal University of Technology

A simulation-based approach for optimization problems under uncertainties

Martin Dahmen, Stephan Westphal, Anita Schöbel

Mathematical optimization is about finding a feasible solution to a given optimization problem with the best objective value possible. This, for example, includes minimizing costs for a production plan in automotive industries or scheduling a timetable for buses with as little delay as possible. Many of those optimization problems are subject to uncertainties in the input data. According to this, the optimizer must take decisions without knowledge of all important data. Passengers have to know about the bus timetable before they begin their trip. The same holds for automobile producers, who create their production schedules before customer's orders have been specified. The aim of optimization under uncertainties is to take all uncertainties into account during the optimization process.

Different approaches for optimization problems under uncertainties have been studied in the past. The main difference between those approaches lies in the way of modeling the uncertainties. A well-known concept in robust optimization is to find a feasible solution for all possible scenarios and minimize the objective value of the scenario with the highest costs [1]. This type of approach is called "worst case analysis", as it is about minimizing the costs in the worst possible case. Robust formulations of optimization problems under uncertainties have already been studied in the past, such as production planning in automotive industries with uncertain demand and production times [5]. In the case of stochastic or robust stochastic optimization, those uncertainties are expressed by random variables with the aim to minimize the costs of the expected scenarios. In both cases, the solution has to yield optimal results after the realization of the random variables. Another approach is online optimization.

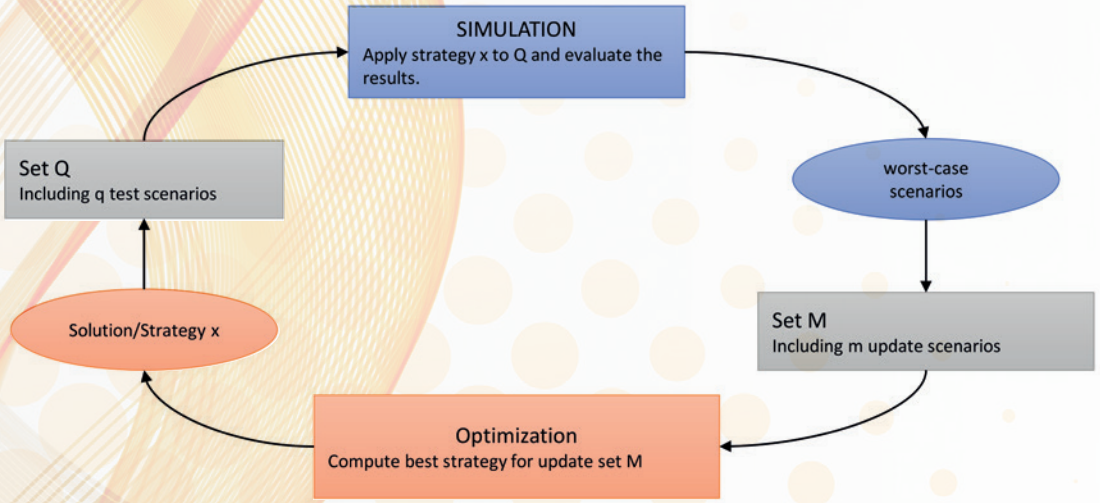


Figure 1: A simulation-based optimization approach

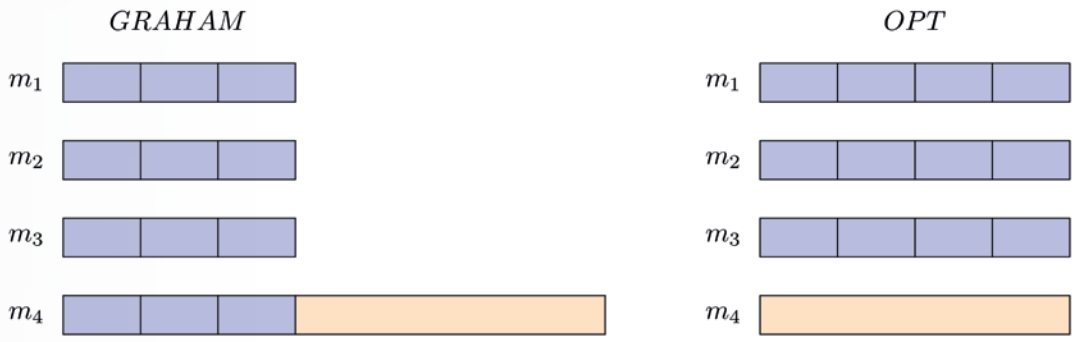


Figure 2: Graham's heuristic (left) compared to the optimal solution (right).

In online optimization, the uncertainties are given as a so-called online sequence. This means that uncertainties are not uncovered all at once but appear as requests one after the other. An online algorithm has to take strategic decisions after each request appears, without knowledge of what the rest of the sequence looks like. This may result in bad decisions during the optimization process with respect to the following requests. A fully uncovered sequence represents the realization of an online scenario. The aim of online optimization is to determine the worst case sequence and to find an online algorithm that minimizes the quotient of the computed online solution and the best possible offline solution for this sequence. The offline solution for a given problem is the optimal solution, with full knowledge of the sequence and its request from the beginning.

The challenge of worst case analysis is to deal with the great number of possible strategies and scenarios that have to be examined. Often the worst-case scenarios depend on the strategies and change with the choice of a different algorithm. Many practical problems, for example, production networks with many different variants and production cycles each day, are often too complex for a theoretical worst-case analysis. As part of this project, we want to present an approach that combines theoretical principles of mathematical optimization with practice-oriented analytic methods of simulation to solve worst-case problems in terms of robust and online optimization.

Simulation-based approach

The alternating interaction of an optimization process and a simulation process is the fundamental concept of a simulation-based optimization approach. Output data of the optimization process forms the input for the simulation and vice versa. The proposed iterative simulation-based optimization procedure is illustrated in Figure 1. First, a strategy is chosen, based on the input data for the optimization process (optimization). Afterwards, the strategy is evaluated in the simulation process, by determining the scenarios with the most impact on the strategy's objective value (simulation). Those scenarios form the input of the next optimization process which is about to find a new and better strategy. Convergence of similar approaches for robust optimization problems has been studied in [4]. As a result of the iterative process, only a selection of scenarios and strategies is taken into account. The goal of the study is to show that even in the case of complex optimization problems a small fraction of the data is sufficient to evaluate algorithms and strategies. In many cases, the scenarios with the greatest impact on the objective value have a very particular structure. For example, the worst-case sequence of the Graham-Algorithm in figure 2, a greedy heuristic for online scheduling on identical machines [2].

The simulation cycle in Figure 1 begins with a finite set M of (randomly) selected update scenarios for our optimization problem. During

Figure 3: The objective functions of an online player and the offline adversary.

optimization, an algorithm is chosen that optimally solves the scenarios from M. In the case of online optimization, we consider problems, which allow algorithms that can be represented by a parameter. During the optimization process, we compute a strategy or parameter based on the scenarios from set M. A new strategy will then be evaluated in the simulation process. For this, a set Q of (randomly chosen) test scenarios is created and the impact of the strategy x on Q is computed by evaluating the objective value of the solutions produced by x and the scenarios from Q. Many approaches from online and robust optimization focus on the relative value of the algorithm x and the optimal solution without uncertainties. In the case of production planning, the schedule with the minimized costs according to our strategy x is compared to the optimal strategy with the knowledge of all relevant data. In the simulation, we want to find scenarios that maximize the distance between the objective value of the strategy x and the optimal strategy. With those scenarios as an input, the next iteration of the simulation-based optimization approach begins with an improvement of our strategy x. In the ideal case, in each iteration of the optimization process, we find a strategy that minimizes the maximum objective value for x applied to the scenarios from M. In the simulation process, we want to find a scenario that maximizes the objective value applied to x. These are exactly the worst-case scenarios for our strategy x.

Application to online problems

Online-Optimization is often considered as a two-player game, consisting of an online player and the offline adversary. The input sequence is unknown to the online player, and he must take decisions for each request without knowledge of the rest of the sequence. The offline adversary plays against him. He knows the whole sequence from the beginning or rather can determine what the sequence looks like. He can solve the sequence to optimality while trying to maximize the gap between his objective value and the objective value of the online player. The largest possible ratio of the objective function values of these solutions is called the competitiveness of the online algorithm. The smaller the competitiveness the better the algorithm (in the case of minimization problems). Our aim is to find lower and upper bounds for the competitiveness of an online algorithm for an optimization problem.

We adapted the idea of a two-player game between the online player and the offline adversary to our simulation-based approach from Figure 1. The optimization process represents the choice of an online strategy by the online player. Afterwards, in the simulation, the offline adversary chooses his sequence that best counters the online strategy. The objective function for both players is illustrated in Figure 3. In the case of the online player, we also call it a minimax problem.

Application to online machine scheduling
Scheduling is one of the most studied problems in operations research. In a scheduling problem, a set of jobs has to be assigned to a set of machines and different timeslots, under the restriction that various constraints have to be fulfilled. The jobs, as well as the machines, may have different characteristics and properties. The machines, for example, operate at different speed levels, or the jobs can only be processed in specific timeslots. In our case, we have an additional online-characteristic. This means that jobs will appear subsequently and as soon as a job appears an online algorithm has to decide which machine it

will we assigned to. A complete schedule, so that every job is assigned to exactly one machine, will be called scheduling profile in the following. We applied our simulation-based optimization approach to the online optimization problem "minimizing the makespan on identical machines". Each job is given a processing time. An online algorithm has to assign each appearing job to one of the machines such that the makespan, the maximum load of all machines will be minimized. A well-known greedy heuristic to solve this problem is the Graham-Algorithm [2]. Each job is assigned to the machine with the smallest load. Graham's algorithm is very useful to solve many practical problems, although, in some cases, it might be better to have remaining capacities on less loaded machines to deal with very long jobs (Figure 2). Based on this idea, we choose a parameter that decides whether a job belongs to the set of short jobs or long jobs. This, in turn, will define whether the job is processed on a machine with a high load or a low load. We implemented and tested different approaches for adjusting the parameter. The obvious idea is to find the optimal parameter for the set M which contains all update sequences. The problem is that the procedure is hardly scalable because of its complexity. Minimizing the makespan on identical machines is an NP-hard problem and is therefore hard to solve. In order to solve this optimization problem in each iteration of the simulation-based optimization approach, we have to use approximation algorithms to find a good solution in acceptable time. Another approach is to improve the parameter in each iteration by a local search heuristics. The advantage here is the short runtime but we may experience only slight improvements. It is also more likely that we end up in a local optimum, which depends on the choice of our update-set M. As we see in Figure 4, a set M that contains only worst-case sequences from the last iterations may lead to alternating strategies. Therefore, the set M should consist of a reasonable selection of sequences from all previous iterations. The question to be solved is, which and how many of those sequences are necessary for convergence and a proof of optimality.

The approaches we have used so far did not use a sophisticated concept to create the set Q of test sequences. Especially in the case of online optimization, most worst-case scenarios have a very specific structure and it is not very likely to find them simply by sampling. Therefore, it is necessary to improve the way of how test scenarios are chosen

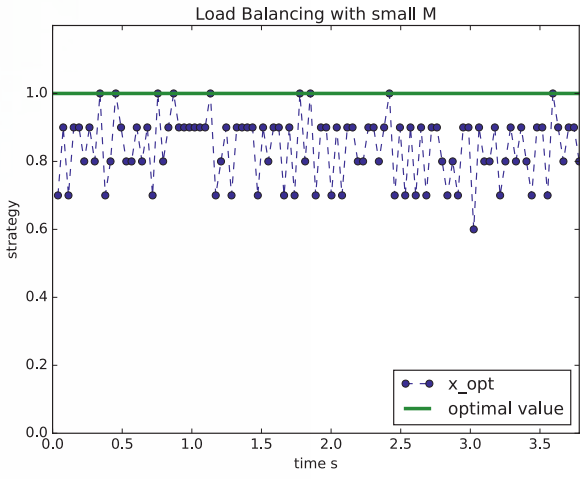


Figure 4: Behavior of the simulation-based optimization approach. M does not contain any sequences from previous iterations. Improvement procedures

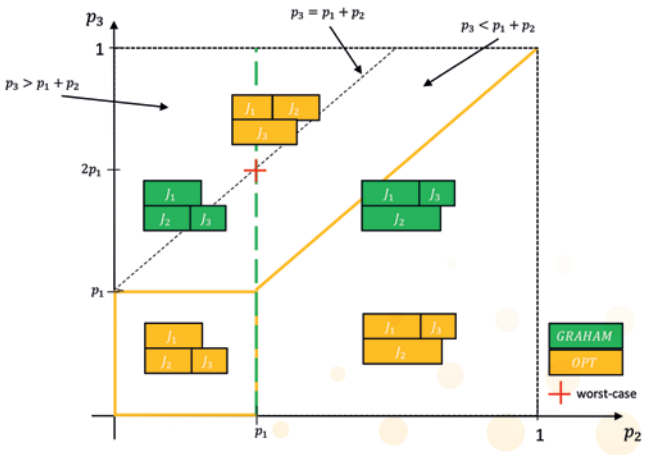


Figure 5: All possible schedules for the graham heuristic and the optimal solution for scheduling 3 jobs on 2 machines.

in order to improve convergence of the simulation based approach. In the following, we present two approaches on how to create the test scenarios Q. The first one is an analytic approach using linear programming, while the second one is based on ideas from genetic algorithms. Both can be applied to certain online optimization problems. The first approach uses a linear programming model to maximize the objective value of worst-case sequences in the simulation process. Strategies are represented by the constraints of the linear program. The objective value is given by the quotient of the online and offline solution so that by maximizing the objective function the program computes the desired worst-case sequence. Unfortunately, this is a mixed-integer linear program with an exponential amount of constraints. Solving this model is very time-consuming. In order to solve this problem, we partitioned the sequences according to their profiles given by the online and optimal offline solutions (Figure 5). For each partition, we obtain a fractional linear program that can be solved by methods from linear programming. Thus, we can compute the worst-case sequences for each partition and obtain lower bounds for the competitiveness of our algorithms. This way, the number of algorithms and scenarios we have to consider in the simulation-based optimization approach can be reduced significantly. The process of convergence is illustrated in Figure 6. Unfortunately, the partitioning of the scheduling profiles is a complex task and this approach could only be applied to a very limited type of scheduling problems.

Genetic Operators	
Random	Create a set of (uniformly distributed) random sequences
Crossover	Create a new sequence as a combination of several previous sequences e.g. σ1=[B, A, D, F, A, B, D, E, D], σ2=[A, F, E, D, F, F, B, A] → σ=[A, A, E, F, A, B, F, B, A]
Mutation	Exchange some genes of a sequence with random elements e.g. σ1=[B, D, F, D, A, A, E, C, C] → σ=[B, D, C, D, A, A, B, C, C]
Double	Create a new sequence by duplication e.g. σ1=[A, F, D, E, C] → σ=[A, F, D, E, C, A, F, D, E, C]
Cut	Reuse certain subsequences e.g. σ1=[D, C, B, A, E, E, A, B] → σ=[D, C, B, A, C, B, F, A]
Invert	Create the inverse of a sequence e.g. σ1=[A, A, B, A, B, B, A, A,] → σ=[B, B, A, B, A, A, B, B, A]
Boundary	Create extreme sequences e.g. σ1=[A], σ2=[C, C, C, C, C, C, C, C]

Figure 7: A selection of implemented genetic operators.

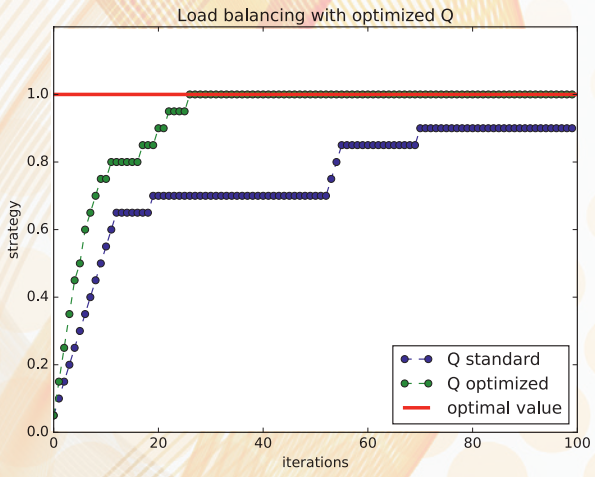


Figure 6: The improvement heuristic in comparison to a randomly chosen Q.

Genetic Approach

The second approach uses genetic methods to manipulate worst-case scenarios from the set M. This way, the information about the effect of online strategies on sequences from the set M are conserved and even revised. This idea is based on the observation mentioned before, i.e. that worst-case sequences have a very specific structure. We try to reproduce this structure. Genetic algorithms have been studied as heuristic approaches to complex optimization problems for a long time [7]. Under certain assumptions, genetic algorithms can be applied to minimax problems [8].

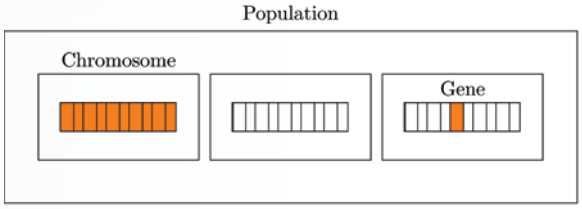


Figure 8: The structure of our population of test scenarios.

Our idea is to use genetic methods to create the set of test scenarios Q in our simulation-based optimization approach. We create new sequences by manipulating and combining the specific structures of the previous found worst-case sequences. A repetition of certain structures, as we experienced by randomization, should be prevented. Our genetic operators are based on the typical methods of genetic algorithms, such as "selection", "mutation" and "crossover" (Figure 6) [7]. Online sequences correspond to the "chromosomes" of an evolutionary "population". The simulation process is used to evaluate the "fitness" of our online algorithm, depending on the population Q. The operators are independent of the optimization problem and can be applied to almost every kind of online optimization problem. Figures 8 and 9 show the improved results we achieved with the genetic approach.

References

[1] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. Robust Optimization . Princeton University Press, Princeton and Oxford, 2009.

[2] Pinedo, Michael. Scheduling: theory, algorithms and systems. Englewood Cliffs, N.J.: Prentice Hall, 1995

[3] A. Borodin and R. El-Yaniv. Online Computation and Competitive Analysis . Cambridge University Press, 1998.

[4] A. Mutapcic and S. Boyd, Cutting-set methods for robust convex optimization with pessimizing oracles. Optimization Methods and Software, 24(3):381–406, 2009.

[5] S. Albers. Online Scheduling: Introduction to Scheduling, edited by Yves Robert and Frederic Vivien. Chapman and Hall/CRC Press, 57-84, 2009.

[6] Claus, Herrmann, Manitz. Produktionsplanung und -steuerung, Forschungsansätze, Methoden und deren Anwendungen. 2015

[7] Goldberg, D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Massachusetts, 1989.

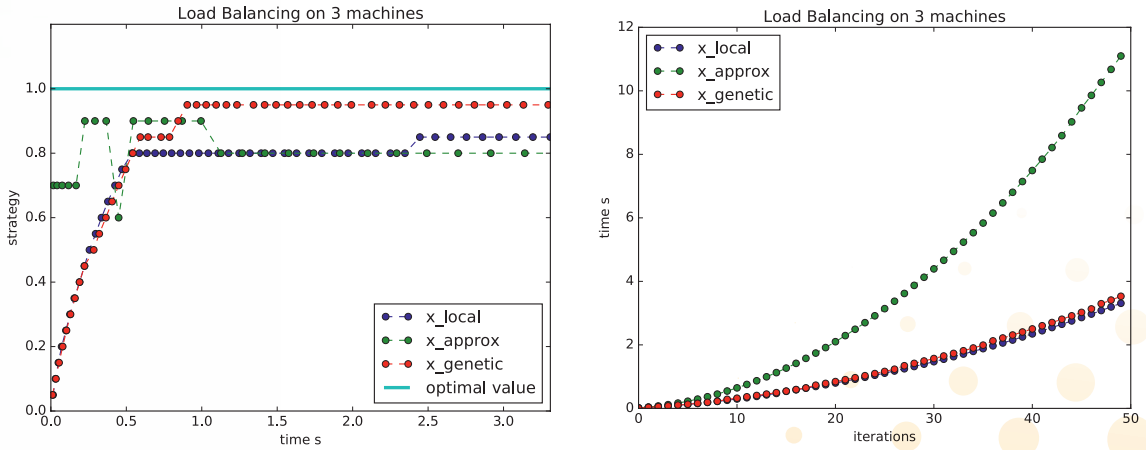


Figure 9: Comparison of the simulation-based optimization approaches. The figure on the left shows the chosen strategies over time. The figure on the right shows the cumulated time needed for all iterations.

[8] W. Jeffrey, Herrmann. A Genetic Algorithm for Minimax Optimization Problem. Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress onVol.2, 1999.

[9] Irene Fink, Sven O. Krumke, Stephan Westphal. New lower bounds for online k-server routing problems, Information Processing Letters, v.109 n.11, p.563-567. May, 2009

Project data:

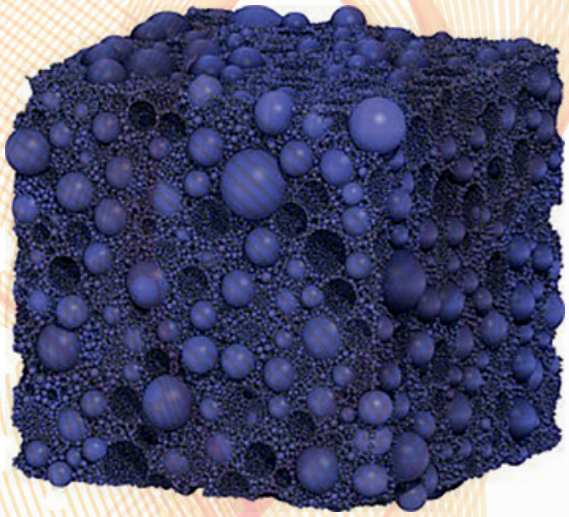
The project is funded from SWZ with one TV-L E13 staff position since August 2015 at the sites Clausthal and Göttingen. Involved scientists are:

Prof. Dr. Anita Schöbel
Research Group
Optimization
Institute for Numerical and
Applied Mathematics
University of Göttingen

Martin Dahmen, M.Sc.
Research group Discrete
Optimization
Institute of Applied Stochastics
and Operations Research
Clausthal University of
Technology

Prof. Dr.
Stephan Westphal
Research group Discrete
Optimization
Institute of Applied Stochastics and Operations Research
Clausthal University of
Technology

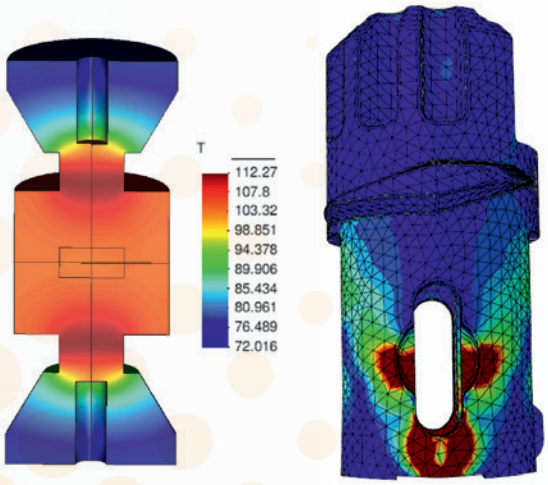
Simulation von Materialien

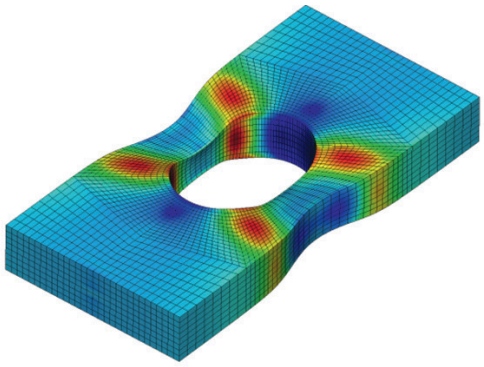


Die Fortschritte der Materialwissenschaften und angrenzender Ingenieurdisziplinen sowie Naturwissenschaften mit dem Fokus der Einbeziehung von Materialeigenschaften haben seit jeher den Entwicklungsstand einer Gesellschaft definiert. Materialwissenschaften sind gerade in der deutschen Industrielandschaft eines der zentralen Themen, welche die Grundlage für viele Innovationen in anderen Industriezweigen liefern.

In den Materialwissenschaften sowie materialorientierten Ingenieurdisziplinen und Naturwissenschaften hat sich bereits eine stark interdisziplinäre Arbeitsweise herausgebildet, welche in den stark überlappenden Bereichen „Computational Materials Science“, „Computational Physics“, Computational Mechanics und „Computational Chemistry“ ausgeprägt ist. Simulationen haben sich in den Materialwissenschaften und den benachbarten naturwissenschaftlichen Disziplinen zu einem lebendigen und forschungsstarken Wissenschaftszweig herausgebildet, der vermehrt auch von der Industrie wahrgenommen und aktiv gefördert wird. Probleme der Materialwissenschaften und materialorientierten Energiewissenschaften sind sehr vielfältig und spielen sich typischerweise auf unterschiedlichsten Längen- und Zeitskalen ab. Zudem liegen die unterschiedlichsten physikalischen Ursachen vor, wie Temperatur, Strahlung, elektromagnetischer und mechanischer Felder. Deshalb zeichnet sich das Feld der Materialsimulationen durch eine Vielzahl verschiedener Simulationsmethoden aus, welche auf die jeweilige Problemklasse zugeschnitten sind:

Auf der kleinsten Längenskala werden in sogenannten ab-initio Simulationen atomare Prozesse parameterfrei auf der Grundlage von Naturgesetzen simuliert. Diese quantenmechanischen Methoden erfordern meistens Hochleistungsrechner, erlauben aber vielfältige und quantitative Aussagen zu treffen. Ein solches Programmpaket wird in Clausthal entwickelt und vertrieben.
Auf der makroskopischen Längenskala hingegen wird das Material als ein Kontinuum betrachtet, dessen Verhalten von Materialmodellen bestimmt wird. Die darin auftretenden Materialparameter müssen anhand ausgewählter Experimente mit Hilfe von Optimierungsverfahren bestimmt werden. Während eine Vielzahl von Simulationstechniken vereinheitlicht ist, kommen hierbei vorwiegend kommerzielle Programmpakete zum Einsatz. Da andererseits solche Programme auch großen Einschränkungen unterworfen sind, ist die Neu- und Weiterentwicklung von numerischen Berechnungsverfahren erforderlich. Das Hauptinteresse bei der Anwendung kommerzieller Programme





richtet sich auf die Modellierung des Problems und die Bestimmung der dabei verwendeten Parameter sowie die Validierung der Resultate. Bei der Neuentwicklung numerischer Berechnungsverfahren hingegen stehen die Analysis, die Effizienz und die Stabilität der Berechnungsmethoden und die Verifikation der Programme im Fokus.

Zwischen diesen Polen befinden sich eine Vielzahl weiterer Methoden die hier nur einige als Stichworte genannt werden sollen: Stochastische Simulationen, Molekulardynamik, Mehrfeldsimulationen, Phasenfeldtheorien, Mikrostruktursimulation von Versetzungen, Simulation chemischer Prozesse, usw.

Eine der großen Herausforderungen im Bereich Computational Materials Science“ ist daher die Multiskalensimulation, die im Idealfall von der ab-initio Simulation bis zur Simulation von Umformprozessen und Herstellungsverfahren reicht. Diese Herausforderung wird einerseits dadurch angegangen, dass Simulationsparameter gezielt von den auf atomarer Ebene bestimmten Größen über mikroskopischen Berechnungen zu den makroskopischen Simulationen durchgereicht werden. Andererseits werden unterschiedliche Simulationsmethoden in eine einheitliche Simulationsumgebung integriert, um Effekte zu beschreiben, bei denen unterschiedliche Längen- und Zeitskalen nicht mehr entkoppelt werden können. Diese Aktivitäten erfordern die Zusammenarbeit unterschiedlicher Wissenschaftsdisziplinen und profitieren von Zusammenschlüssen, wie sie im Simulationswissenschaftlichen Zentrum geplant sind.
Im Gegensatz zu etlichen anderen Anwendungen von Simulationsverfahren sind Simulationen im Bereich der materialorientierten Ingenieurdisziplinen sehr häufig dadurch gekennzeichnet, dass sie alle verfügbaren Rechnerressourcen bis an die Grenzen ausschöpfen müssen, um gewünschte Ergebnisse zu erzielen. Fragen der algorithmischen Komplexität und der Effizienz von Implementierungen sind daher für Fortschritte der Simulationsmethoden in diesem Teilgebiet von entscheidender Bedeutung.

86

Kopplung multi-physikalischer Prozesse zur Simulation von Gasbohrungen

Stefan Hartmann, Leonhard Ganzer, Jithin Mohan, Birger Hagemann

Die Wasserstoffverpressung in ehemalige Gasreservoirs dient zur Zwischenspeicherung von Energie. Dabei wird überschüssige elektrische Energie in chemische Energie umgewandelt und auf vergleichbar geringem Raum in Kavernen gespeichert. Zunächst wird die überschüssige elektrische Energie genutzt, um durch Elektrolyse Wasser in Wasserstoff und Sauerstoff zu zersetzen. Der Wasserstoff wird anschließend komprimiert und in eine geschlossene geologische Formation injiziert. In Zeiten geringer Stromproduktion, aber hoher Nachfrage, wird der Wasserstoff aus der Gesteinsformation wieder entnommen und kann durch Motorgeneratoren oder Brennstoffzellen in elektrische Energie umgewandelt werden. Analoge Vorgehensweisen treten bei POWER-to-GAS oder CO2-Speicherung auf.

In Deutschland gibt es derzeit insgesamt 51 Untergrundgasspeicher, davon 21 in ehemaligen Erdgaslagerstätten oder Aquiferen und die restlichen 30 in Salzkavernen. Mit etwa 24 Mrd. m^3 bei Normalbedingungen verfügt Deutschland über das größte Arbeitsgasvolumen in der EU, daher ist davon auszugehen, dass diese verfügbaren Speichervolumina bei der Gestaltung der Energiewende eine wesentliche Rolle spielen werden. Etwa 2/3 des Arbeitsgasvolumens in Deutschland befinden sich in Porenspeichern, der Rest ist in Kavernenspeichern verfügbar. Kavernenspeicher sind international nur von untergeordneter Bedeutung, jedoch durch die geologische Situation in Norddeutschland hier besonders häufig. Zusätzlich gibt es in Deutschland eine Vielzahl von nahezu ausgeförderten Erdgaslagerstätten, die potentiell zur Errichtung neuer und weiterer Untergrundspeicher geeignet sind.

Für all die genannten Anwendungen in Untergrundspeichern repräsentieren Gasbohrungen, deren dauerhafte technische Integrität sowie die Eigenschaften der geologischen Speicherformationen (geologische Integrität) systemkritische Komponenten, zu deren Beurteilung und Bewertung eine gekoppelte Simulation des Verbundsystems (Bohrungsmaterial, Komplettierung, Zementation, geologische Formation, Formationsfluide) notwendig ist.

Um die kritischen Stellen der Gasbohrungen zu untersuchen, sind numerische Untersuchungen mit Hilfe moderner Simulationswerkzeuge unabdingbar. Es wird daher in diesem Projekt eine moderne und effiziente numerische Vorgehensweise herangezogen, um ein Kopplungstool für die vorliegende multi-physikalische Mehrfeldproblematik zu entwickeln.

Stand der Forschung

Eine wissenschaftliche Fragestellung stellt die Integrität von Bohrungen dar, d.h. welche Spannungszustände liegen aufgrund von äußeren Belastungen sowie Temperaturänderungen vor. Auch führen Korrosions- und Alterungsprozesse zu Undichtigkeiten bzw. Leckagen, die unerwünscht sind und zum Ausfall der Bohrung führen können. Mögliche Leckagewege befinden sich dabei besonders an den Verbindungen zwischen (Deck-) Gestein und Zement sowie zwischen Zement und der Verrohrung (siehe Abb. 1).

Daher sind thermomechanische Berechnungen für den Injektions- und den Produktionsprozess der Bohrungen erforderlich. Drei Gebiete können in Abb. 1 identifiziert werden. Das poröse Medium (umgebendes Gestein), welches mit der Open Source Software DuMuX behandelt werden kann, die Verrohrung selbst bestehend aus Stahl und Zement (sowie einem Elastomer-Dichtungsbauteil, dem sogenannten Packer; Berechnung mit Inhouse-Finite Elemente Programm Tasafem) und dem Gebiet innerhalb des Rohres, in welchem sich das zu verpressende Gas befindet

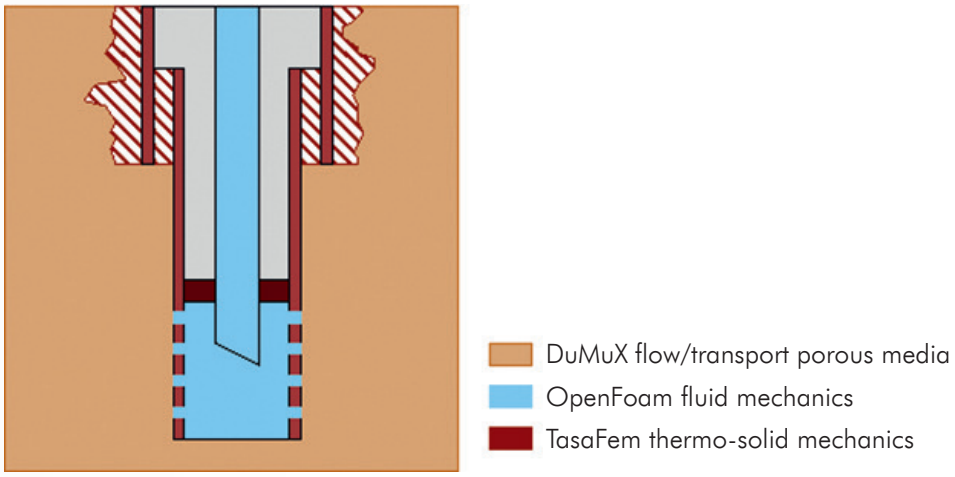


Abbildung 1: Schnitt durch eine Gasspeicherbohrung und Berechnungsgebiete

(Simulation mit OpenFoam). Insgesamt liegen in den unterschiedlichen Gebieten partielle Differentialgleichungen vor, die mit unterschiedlichen Diskretisierungsverfahren gelöst werden müssen. Die Anwendung einer gemeinsamen impliziten Zeitintegration auf das nach der räumlichen Diskretisierung auftretende gekoppelte System aus algebraischen und differentiellen Gleichungen führt auf ein gekoppeltes System aus nichtlinearen Gleichungen. Diese sollen mit beschleunigten Gauss-Seidel Verfahren gelöst werden.

In der ersten Projektphase sind die nichtlinearen Gauss-Seidel Verfahren mit unterschiedlichen Beschleunigungstechniken untersucht worden. Zudem sind die für das Teilproblem der Gasrohrströmung erforderlichen Randbedingungen für ein Demonstrator-Beispiel hergeleitet worden, um die Strömungseigenschaften und die notwendigen Diskretisierungen für den Fluid-Löser herauszufinden. Parallel hierzu haben wir zu Verifikationszwecken analytische Vergleichslösungen für das dickwandige Rohr unter Innen- und Außendruck sowie thermischer Ausdehnung entwickelt. Nebenbei ist auch einer Inkonsistenz bei inkompressiblen Fluiden unter thermischer Ausdehnung gefolgt worden.

Als weitere Schritt ist zunächst die Kopplung aus der Rohrströmung und dem porösen Gestein geplant und numerischen Verfahren durch Verifikationsbeispiele zu überprüfen.

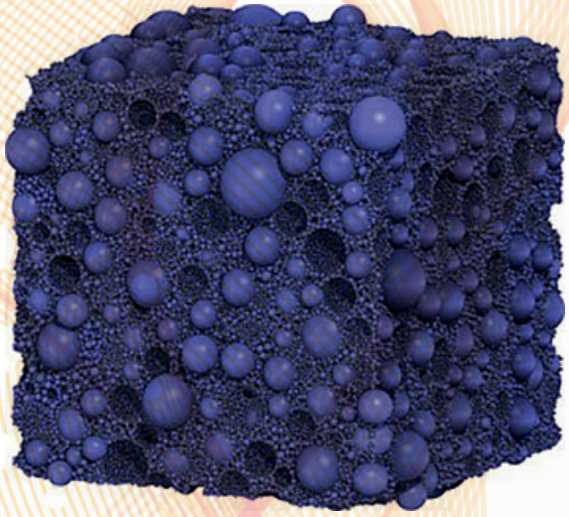
Projektdaten

Das Projekt wird seit August 2015 vom SWZ mit insgesamt einer TV-L E13 Stelle an dem Standort Clausthal. Beteiligte Wissenschaftler sind:

Prof. Dr. Leonhard Ganzer
Arbeitsgruppe
Lagerstättentechnik
Institut für Erdöl- und
Erdgastechnik
Technische Universität
Clausthal

Prof. Dr.-Ing.
Stefan Hartmann
Arbeitsgruppe
Festkörpermechanik
Institut für Technische Mechanik
Technische Universität
Clausthal

Simulation of Materials

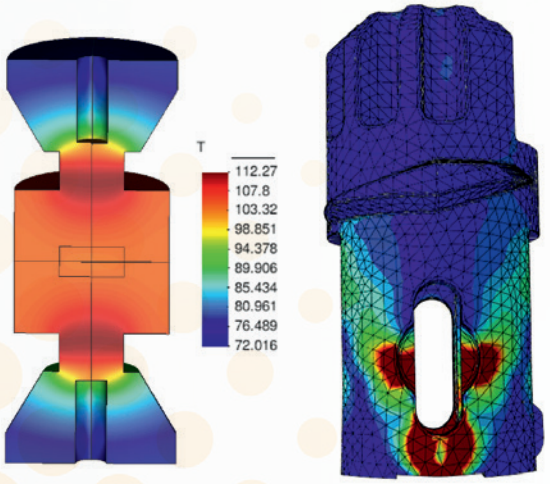


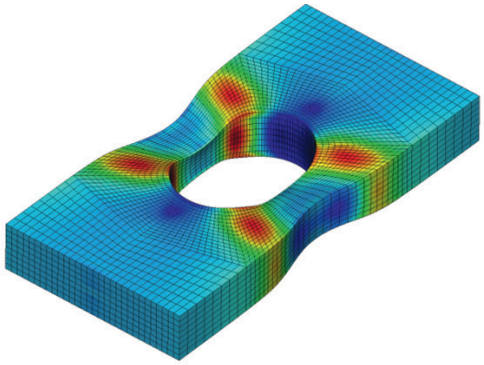
The advances in Materials Science and both related Engineering disciplines as well as Natural Sciences with focus on considering material properties always have defined the state of development of a society. Materials Science is one of the key topics in the German industrial landscape, which provides the basis for many innovations in other areas of industry.

In Materials Science and related disciplines, a highly interdisciplinary approach is already established, which is divided into the strongly overlapping areas of „Computational Materials Science“, „Computational Physics“, “Computational Mechanics” and „Computational Chemistry“. Simulations have been developed into a living and research-oriented scientific branch in material science and their neighboring Natural Sciences, which is also increasingly perceived and actively promoted by the industry.

Problems in Materials Sciences and material-oriented energy sciences are very diverse and are typically based on multiple length and time scales. Furthermore, different physical reasons are apparent, such as temperature, radiation, electromagnetic and mechanical fields. Therefore, the field of material simulations is characterized by a variety of different simulation methods, which are tailored to the respective problem class:
On the smallest scale, atomic processes are simulated parameter-free on the basis of natural laws in so-called ab-initio simulations. These quantum mechanics methods usually require high-performance computers, but allow achieving a variety of quantitative results. Such a program package is developed and distributed in Clausthal.
On the macroscopic length scale, the material is considered to be a continuum, whose behavior is determined by material models. The material parameters, which occur in the models, have to be identified at specific experiments using optimization procedures. Since the simulation techniques are largely uniform, one concept is to draw on commercial program packages. On the other hand, these programs have limitations, so that one major goal lies in the development of new numerical methods. The main interest in applying commercial programs is the modeling of the problem, the determination of the parameters as well as validating results. In the case of code developing, the interest are seen in the analysis, efficiency and stability, and the verification of the codes.

Between these poles, there are a number of other methods. Some will be listed as following: stochastic simulations, molecular dynamics, phase





field methods, microstructural simulation of dislocation, simulation of chemical processes, etc. Thus, one of the „Grand Challenges in Computational Materials Science" is the multiscale simulation, which ideally extends from ab initio simulation to the simulation of forming processes and manufacturing processes. On the one hand, this challenge is addressed by the fact that simulation parameters are deliberately transmitted from the atomic level through microscopic computations to the macroscopic simulation. On the other hand, different simulation methods are integrated into a uniform simulation environment in order to describe effects in which different length and time scales can no longer be decoupled. These activities require the cooperation of different scientific disciplines and benefit from mergers as planned at Simulation Science Center.

In contrast to several other applications of simulation methods, simulations in the field of material science are very often characterized by the fact that they have to exploit all available information processing resources to their limits in order to achieve usable results. Questions of algorithmic complexity and efficiency of implementations are therefore of crucial importance for the progress of the simulation methods in this area.

Coupling of multi-physical processes for the simulation of gas wells

Stefan Hartmann, Leonhard Ganzer, Jithin Mohan, Birger Hagemann

1. Objectives

In September 2010 the German government announced a new energy concept. The goals comprise essential changes of the power supply until 2050, see (Klaus et al., 2010). The existing challenges of this huge project are represented by both a failure-resistant energy-supply encompassed by strongly fluctuating energy production drawing on wind and solar power plants, as well as a sustainable reduction of greenhouse gas emissions. For this goal, underground storage units can be utilized in different ways:

- Hydrogen storage might be used to store huge amounts of electrical energy in geological sub-surfaces. In this technology, electrical energy is transformed into chemical energy and, accordingly, can be stored in comparably small areas, (Kepplinger et al., 2011). The surplus electrical energy is utilized to separate water into hydrogen and oxygen by means of electrolysis, (Sherif et al., 2005). The hydrogen is condensed and pressed into a geological formation. During periods of less power production, but high demand, the hydrogen is extracted and transformed into electrical energy by generators or fuel cells.
- POWER-to-GAS: Hydrogen and bio-gas can be injected into the current natural gas grid. Studies have shown that concentrations in the range of single-digit percentage are conceivable. In this concept, the natural gas grid and existing underground gas-storage are used as storage facilities, (DVGW, 2011).
- CO_2 -storage: The greenhouse gas carbon dioxide, which frequently is a byproduct of energy plants based on fossil fuel, but also in various industrial processes, can be separated and stored in geological formations.

The storage of natural gas is performed for more than 100 years in Germany to balance the daily and seasonable fluctuations of the gas consumption. Thus, it is a well-established technology. The experience using other gases such as hydrogen, carbon dioxide or mixtures of gases is limited. In Great Britain and the US examples of storage in caverns exist, in which comparably small total volumes of pure hydrogen (> 95%vol) are stored. It is reported that this works well. However, using porous formations like former gas storage and aquifers, where the gas in not stored in a large cavity, difficulties might arise for carbon dioxide and hydrogen. There is some experience using town gas with less than 50%vol of hydrogen, (Sedlacek, 2014; Leonhard, 2008).

Storage wells in Germany are mostly older than thirty years and were only established for natural gas. Their usage for other gases might lead to problems for different reasons:

- Molecular hydrogen might penetrate into the metal pipes' lattice leading to hydrogen embrittlement.
- Hydrogen storage might lead to micro-biological activities. In this context produced hydrogen sulfide is dangerous, since it leads to corrosive solvents.
- Moreover, CO_2 and biogas might contain corrosive components.
- Hydrogen can disappear much faster through small cracks due to their high diffusivity and small molecule size.
- The injection of particular gas components (e.g. hydrogen and carbon dioxide), which have not been initially in the reservoir, might lead to a number of bio-and geochemical processes, which change the properties of the rock. If this is coupled with the stress and temperature changes, cracks in the concrete or in the connections can be introduced. This is very dangerous in the capstone region, since a leakage of the gas deposit might emerge.

For all of the applications mentioned above, gas wells and geological storage formations are the

system critical components. In order to assess and predict their technical integrity, coupled simulations of the compound system are required.

2. Methods

The underlying entire project scope can be interpreted as a surface coupling approach of three different regions, where the methods are mainly driven by applications in the field of fluid-structure interaction. Additionally to the surface coupling, a volumetric coupling exists in each domain, which is dominated by thermal and mechanical fields. In the surrounding rock also chemical and hydro-dynamical agencies occur. In the following, we abbreviate these regions by the expressions solid-region, representing the casing itself, the fluid-region, domain inside the conduit, where the gas flow occurs, and the rock-region represented by the surrounding area, see Fig. 1. The casing is composed by steel and concrete layers, which mainly undergoes internal and external pressures under axial stresses, also under thermal cycles and temperature differences from inside to outside. The gas pressure drives the internal pressure in the casing during the injection and production phase. Additionally, a packer has to be considered, isolating the production area and the remaining tube, which can be seen as a seal. This might also cause problems if the separation is not guaranteed.

In order to gain the final possibility of simulating the production and injection phase, several subproblems have to be treated. We apply the method of vertical lines, i.e. the partial differential equations (PDE) in each region are discretized in space. Here, either the finite element (FEM) or the finite volume method (FVM) is applied, which depends on the applied solver. This results in a large system of differential-algebraic equations (DAEs). We have the discretized equilibrium conditions in the solid (discretized principle of virtual displacements) as the algebraic part, and the transient (discretized) heat equation representing the differential part. In dependence of the constitutive model for the steel and the concrete, for example, applying a plasticity model, additional ordinary differential equations (ODEs) might appear. A similar structure occurs in the fluid part, i.e. the gas flow. There, the mass balance, balance of linear momentum and energy lead, in dependence of various assumptions (stationary, transient, compressible, incompressible, ...) to a DAE-system or an ODE-system as well. Analogously, this holds also for the rock region. Further algebraic equations arise from the interface conditions between solid and gas, solid and rock, as well as gas and rock.

Since all applied programs are able to perform a Backward-Euler time integration step, first order accuracy in the time domain can be obtained for the entire DAE-system, which is result of all spatially discretized PDEs. Moreover, according to (Birken et al., 2010), we can apply time-adaptive, diagonally-implicit Runge-Kutta methods as well. This offers two advantages: higher order accuracy, and

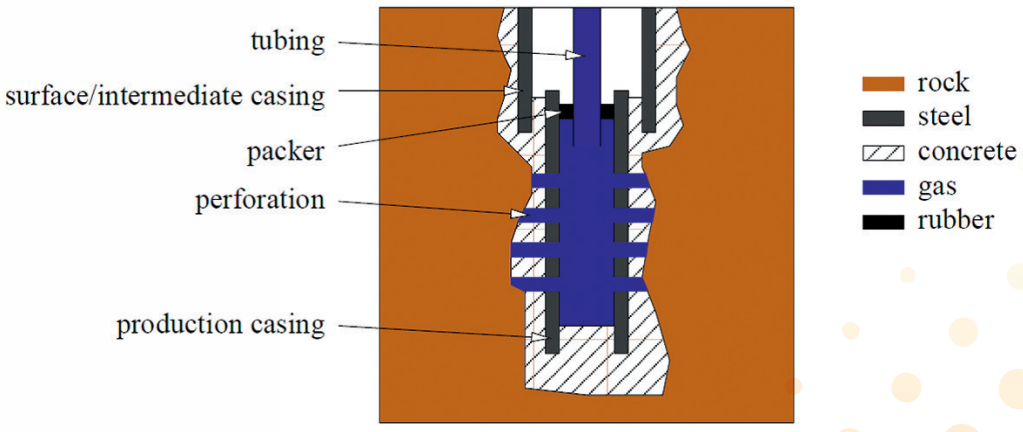


Figure 1: Principal sketch of production casing. Additionally, the regions of the solid, fluid and hydrodynamical solvers are identified

step-size selection according to an error estimator, which does not require additionally computations. In other words, the time integration adapts to the most sensitive part of the solution. Of course, each spatially discretized PDE requires its own step-size and there might be different time scales, for example, if in the perforation region of the gas well, turbulence has to be considered. However, this is out of the scope of the first steps in the project.

The result of the time integration scheme leads to a coupled system of non-linear equations in each (temporal) stage. In this respect, a partitioned approach has to be applied, see (Rothe et al., 2015a), i.e. each solver has to be exploited. Here, we draw on the non-linear Gauss-Seidel approaches, which are studied in detail in (Erbts and Düster, 2012; Erbts et al., 2015; Wendt et al., 2015).

Several sub-problems have to be considered in the region of the casing shoe before the entire coupling scheme is implemented:

- formulation and identification of the thermo-mechanical material properties (steel, concrete, rock, gas, ...)
- formulation of the initial and boundary conditions of a demonstrator well (particularly, the production casing, i.e. the casing shoe)
- pure computation of the stationary gas flux (globally – entire well – and locally – perforations)
- formulation of the (thermal and mechanical) loads acting on the casing
- pure computation of the rock region with the flow source of the gas well

2.1. Time Integration

Singly, diagonally-implicit Runge-Kutta methods, see (Hairer et al., 1993; Hairer and Wanner, 1996; Hairer et al., 1989; Strehmel and Weiner, 1995), have the advantage that

- the size of the system for higher order schemes remains the same as it is for the DAE-system (this is different for fully implicit Runge-Kutta methods),
- Backward-Euler based programs can be modified, only the starting vector and the time-step have to be adapted,
- time-adaptivity is provided for free.

Of course, there are also disadvantages. Within one time-step, there are evaluation points (stages). For order two and three, there are two or three stages required, which seems to be sufficient, see (Alexander, 1977; Cash, 1979). For a fourth-order method, five stages are necessary, so that efficiency becomes questionable. Since the accuracy is connected to the overall solution of the resulting system of non-linear equations, we restrict ourselves to a second-order method of Alexander (1977) with the extension to step-size control by Diebels et al. (1999); Ellsiepen (1999). Alternatively, Rosenbrock-type methods might be interest leading to a scheme, where no iteration is required (only for the linear solvers if an iterative scheme is applied). In this respect and for coupled and uncoupled problems, see (Lang and Verwer, 2001; Hartmann and Hamkar, 2010; Hamkar et al., 2012; Netz et al., 2013; Hamkar, 2013) and the literature cited therein. A comparison of different methods is provided in (Rothe et al., 2012). A further disadvantage might be non-linear Dirichlet boundary conditions since they show an order reduction phenomenon, see (Rothe et al., 2015b) for an approach proposed by (Alonso-Mallo, 2002; Alonso-Mallo and Cano, 2004) and covered by (Quint, 2012). These problems have to be investigated in detail.

A first successful coupling of different codes, the in-house finite element program Tasafem and the DLR τ-code for the fluid part, was studied in (Birken et al., 2009, 2010). For further coupling in the sense of a monolithic approach for different fields, see (Hartmann and Rothe, 2013; Hartmann et al., 2013).

2.2. Accelerated Gauss-Seidel Schemes

The simulation of the production and injection phases of the gas well requires the treatment of several sub-problems as mentioned in Section 2. The governing PDEs of each sub-problem is spatially discretized with the FEM or FVM resulting in large systems of DAEs. The application of time integration schemes on such DAEs leads to a coupled system of non-linear equations, which are solved using a partitioned approach. The partitioned solution strategy requires three loops, the outer time loop, the coupling iteration loop and the inner local iteration loop. The coupling itera-

tion loop can be either an implicit coupling loop for strongly coupled problems or explicit coupling for weakly coupled cases. The general solution procedure for a coupled problem is done with the help of non-linear block-Gauss-Seidel method, which is commonly used in Fluid-Structure-Interaction problems; see (Degroote et al., 2010; Joosten et al., 2009; Wood et al., 2010).

Considering a generic problem with four equations and four unknowns partitioned equally into two sub-problems with two variable vectors of size 2 x 1, the Gauss-Seidel iteration procedure is carried out, in which the first sub-problem is solved for the first variable vector keeping the second variable vector fixed. After the local iteration, the resulting update for the iterated variable vector is given as an input to the second sub-problem. After the local iterations for the second sub-problem, we obtain the updates for the both the variable vectors under consideration. The iteration is stopped if the residual satisfies the prescribed tolerance criterion. Although the partitioned approach offers many advantages, it is only conditionally stable, (Armero and Simo, 1992), and in case of strongly coupled problems, large number of iterations may be needed to attain convergence leading to large computational costs. To circumvent these prob¬lems, acceleration methods can be used to drive the solution faster to a converged solution. The basic idea behind any acceleration method is to improve the intermediate solution with previously computed solution estimates. In the generic problem discussed previously in this section, the solution estimate after the local iterations of the first sub-problem is modified with the solutions of its previous iterations. This modified solution is then passed on to the second sub-problem and the Gauss-Seidel procedure is continued. This process is repeated until a converged solution is attained. The modification of the updated solution is based on the kind of acceleration method used. In literature, there exists a wide variety of acceleration techniques like Aitken-type methods, see (Aitken, 1926; Macleod, 1986; Irons and Tuck, 1969), vector sequence acceleration methods, (Brezinski, 1997; Wynn, 1962), and Quasi-Newton methods, (Broyden, 1965; Degroote et al., 2009).

A simple system with four non-linear equation and four unknowns partitioned into two blocks has been solved with Gauss-Seidel scheme with different acceleration methods in (Erbts, 2016). The convergence performance with the application of acceleration methods improved by a factor of two in comparison with the standalone Block-Gauss-Seidel schemes. With the classical Aitken-type acceleration methods, the equation system was solved with only 24 global iterations in comparison to the 62 iterations taken by block Gauss-Seidel method to reach a converged solution. All other variants of the Aitken-type acceleration methods showed comparable results. The vector extrapolation methods show slightly worse convergence properties in comparison to the Aitken-type methods as they need on an average of five iterations more to reach convergence. Further investigations are required to see whether Quasi-Newton methods show an improved performance.

2.3. Gas Flow

As mentioned above, older gas wells can be used for storing energy in the form of hydrogen (injection phase). This stored gas can be later pumped back to cater the energy demands at peak loads (production phase). A key aspect in both phases is the flow of the gas through the gas well itself. We are interested in a first step to understand the flow behavior through the pipes and through the perforations (outlet). Many factors affect the flow behavior like the type of gas and its fluid and thermal properties, type and number of perforations, chemical properties etc. The understanding of the various complex interactions of flow behavior will help in determining the efficacy of the well and in shedding light on its integrity. Thus, the effect of the gas flux in both the injection and production phases is of prime importance and needs to be numerically simulated. Flow simulations have to be carried out both globally to understand flow behavior in the entire well as well as locally at the perforations. Since the perforations have a significant influence on the efficiency of gas storage, initial simulations are done at a local level close to the perforations.

Governing equations

The fluid flow in the gas well is simulated with the help of open source Computational Fluid Dynamics (CFD) program called Openfoam. Openfoam

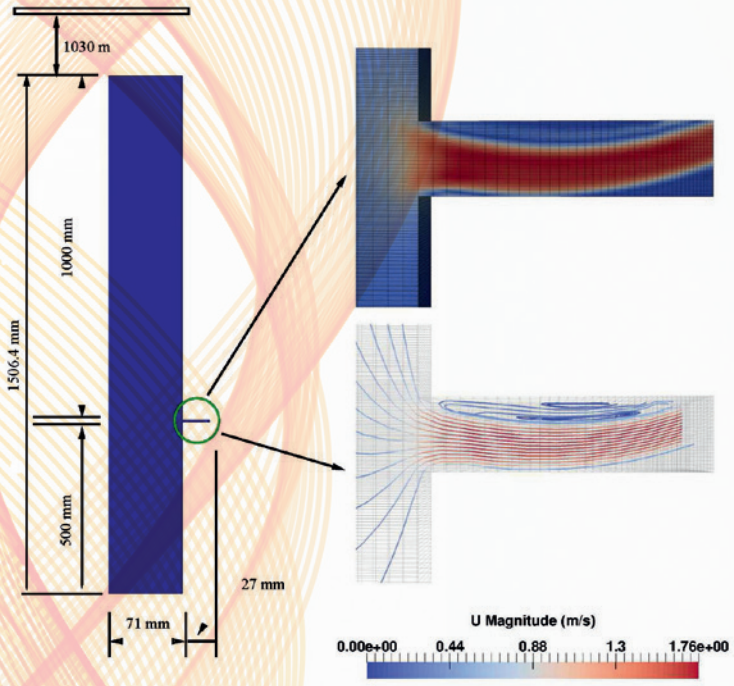


Figure 2: First approach of a gas well and its outlet

offers different solver options to deal with a wide variety of flow problems. Specific solvers are available for flows ranging from incompressible, to compressible, steady, transient, laminar, turbulent, single and multi-phase, combustion, etc., (Greenshields, 2012b). The problem under consideration is solved with the help of simplefoam solver which solves the incompressible Navier-Stokes equation with the Semi-Implicit Method for Pressure-Linked Equation (SIMPLE) algorithm (Ferziger and Peric, 2012). The continuity and momentum

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \vec{v}) = 0$$

$$-\operatorname{grad} p + \eta \operatorname{div}(\operatorname{grad} \vec{v}) + \eta \operatorname{grad}(\operatorname{div} \vec{v}) + \rho \vec{k} = \rho \frac{\partial \vec{v}}{\partial t} + \rho(\operatorname{grad} \vec{v})\vec{v}$$

equation for incompressible flow have to be solved numerically, where v is the velocity, ρ the density, η the viscosity of the fluid, and p the pressure. In order to obtain an explicit equation in pressure, the divergence of the momentum equation is taken and substituted into the continuity equation. The general solution strategy can be summarized by the following steps, see (Greenshields, 2012a):

1. Setting the boundary conditions
2. Solving the discretized momentum equation to compute the intermediate velocity field.
3. Computing mass fluxes at the cell interfaces
4. Solving the pressure equation and applying under relaxation
5. Correcting the mass fluxes at the cell interfaces
6. Correcting the velocities on the basis of the new pressure field
7. Update the boundary conditions
8. Repeat until convergence

Some of these steps are explained in the following.

Geometry and Mesh

The gas well encompasses dimension scales ranging from 1000m to 1.6mm. Thus, a multi-scale problem in space is present. As an initial step, focus is given on the region near the perforations. The geometry used for simulation is axisymmetric and is shown in Fig. 2. In a first step, the simulated geometry is assumed to have only one perforation, whose height is determined so that the entire mass flow through the total number of perforations in a real geometry is emulated. Since we are dealing with fluid effects as the gas leaves the perfora-

tions into the storage rocks, the perforation length is only taken up to the end of the cement casing. The perforation has a height of 6.4mm and length of 27mm. A total pipe length of 1000mm is also included above and 500mm below the perforation so that the effect of flow entrance from the pipe to the perforation can be accurately modeled. The pipe section has a radius 71mm. The geometry is meshed with hexahedral elements with a total mesh density of 26000 cells. The perforation is sufficiently refined to capture the significant flow features with 40cells each in the axial and radial directions. The overall aspect ratio of the cells is 21.

Boundary conditions

The flow properties for the injection phase is known at the surface of the well. The pressure of injection p_0 is 15MPa at a surface temperature $\Theta_0 = 288.16$K. The volume flow rate q_{Std} is given as 1500m^3 h^{-1} at standard conditions of 0.101325MPa and 273.16K. The actual volume flow rate q is recalculated as 2.97×10^{-3} m^3 s^{-1}. The region of the pipe with the perforation simulated is at a depth of 1030m from the surface of the well. The radius at the surface casing changes from 34mm to the production casing of 71mm in a depth of 1027.5m. Thus, the injection pressure at the surface is multiplied by the area ratio to obtain the pressure in the simulated region. The effect of gravity is negligible in comparison to pressure scales at the surface. Accordingly, the pressure in the simulated domain is calculated to be 12MPa.

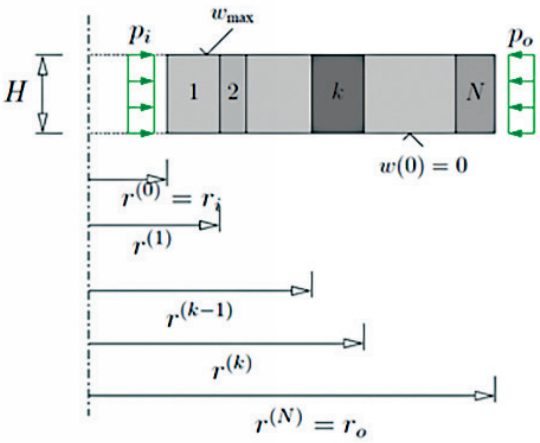


Figure 3: Multi-layered thick-walled tube under internal and external pressure

The simulated domain is assumed axisymmetric, which is prescribed with the help of the wedge boundary condition available in Openfoam and is applied to front and back faces of the geometry. The inlet is prescribed with a volumetric flow rate boundary condition which is calculated from the flow rate at the surface and the area ratio. The flow rate at the inlet is calculated to be 3.6×10^{-5} m^3 s^{-1} which corresponds to 0.19ms–1. The outlet at the perforation is given static pressure of 12MPa. The remaining faces are prescribed with noSlip wall boundary condition.

Material properties and solver settings

The simulated fluid is hydrogen with a kinematic viscosity of 8.8×10^{-6} Pa s (Stewart and Roder, 1964). The pressure equation is solved with the Generalized-Algebraic-Multigrid solver and the velocity field equations are solved with the symmetric Gauss-Seidel solver available in Openfoam. The solver tolerances are defined by 10^{-4}.

Results

The top right sketch in Fig. 2 shows the velocity field near the perforation. The flow is accelerated as it enters the perforation from the production pipe. The maximum velocity at the outlet is 1.7ms^{-1}. The flow appears to be detached from the top and bottom walls of the perforation leading to low velocity regions. The bottom right picture in Fig. 2 shows the velocity streamlines in the perforations. The vortex seen near to the top wall of the perforation clearly suggests the unsteady nature of the flow. Moreover, the effect of turbulence is not taken into consideration. Hence, the next step would be simulating the flow field considering the unsteady nature of the flow. The effect of turbulence and compressibility will also be investigated in the preceding steps.

2.4. Thick-walled Tube with Several Layers

The entire gas well is divided into surface casing, intermediate casing, and production casing (summarized from the top to the bottom), see Fig. 1. Each casing is surrounded by concrete to save the casing (from outside and inside) and to seal the different rock layers. The casing can undergo different mechanical and thermal loads such as bending, torsion, as well as internal and external

pressure. To obtain more insight into the problems, only the casing combined with the concrete encircling the metal casing for the case of internal pressure and external pressure is of interest. This can, of course, be solved with finite elements, however, for both practical issues as well as for code verification purposes, analytical equations are of interest. For one layer and a fixed axial deformation, the solution is provided in (Lehmann, 1984), see Fig. 3. For several layers with and without axial strains a solution is provided in (Clifton et al., 1976; Yeh and Kyriakides, 1986; Bourgoyne, 1986; Bai et al., 1997; Shildip et al., 2015). It is also possible to extend these investigations for multi-layered thick-walled tubes under internal and external pressure with axial strains under stationary heat conduction, where internal and external different temperatures exist.

3. Outlook

After investigating each region for the specified demonstrator well in view of the physical observations and the numerical properties of each of the three programs, the coupling program explained in the previous sections will be developed. The most critical simulation is the fluid part. Here, different approaches are looked at to see their influence on the numerical results. Regarding the coupling tool, a first step is to address only a sequential coupling. Later on, parallel coupling scheme can be studied as well, in order to obtain a much more efficient computational tool. Furthermore, the step-size control technique, which makes use of the same temporal step-sizes for all programs, can be extended to a multiscale approach in time, so that a most efficient computational tool can be provided.

References

Aitken, A. (1926). On Bernoulli's numerical solution of algebraic equations. Proceedings of Royal Society of Edinburgh, 46:289–305.

Alexander, R. (1977). Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s. SIAM Journal on Numerical Analysis, 14:1006–1021.

Alonso-Mallo, I. (2002). Runge-Kutta methods without order reduction for linear initial boundary value problems. Numerische Mathematik, 91:577–603.

Alonso-Mallo, I. and Cano, B. (2004). Avoiding order reduction of Runge-Kutta discretizations for linear time-dependent parabolic problems. BIT Numerical Mathematics, 44:1–20.

Armero, F. and Simo, J. C. (1992). A new unconditionally stable fractional step method for non-linear coupled thermomechanical problems. International Journal for numerical methods in Engineering, 35(4):737–766.

Bai, Y., Igland, T. R., and Moan, T. (1997). Tube collapse under combined external pressure, tension and bending. Marine Structures, 10(5):389–410.

Birken, P., Quint, K. J., Hartmann, S., and Meister, A. (2009). On coupling schemes for heat transfer in FSI applications. In Hartmann, S., Meister, A., Schäfer, M., and Turek, S., editors, Fluid-Structure-Interaction. Theory, Numerics and Applications, pages 21 – 30, Kassel, Germany. Kassel University Press.

Birken, P., Quint, K. J., Hartmann, S., and Meister, A. (2010). A time-adaptive fluid-structure interaction method for thermal coupling. Computing and Visualization in Science, 13:331–340.

Bourgoyne, A. (1986). Applied Drilling Engineering. SPE textbook series. Society of Petroleum Engineers.

Brezinski, C. (1997). Projection methods for systems of equations. Studies in Computational Mathematics. Elsevier, Amsterdam, New York, 1997.

Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations. Mathematics of computation, 19(92):577–593.

Cash, J. R. (1979). Diagonally implicit Runge-Kutta formulae with error estimates. Journal of the Institute of Mathematics and its Applications, 24:293–301.

Clifton, R. J., Simonson, E. R., Jones, A. H., and Green, S. J. (1976). Determination of the critical-stress-intensity factor kic from internally pressurized thick-walled vessels. Experimental Mechanics, 16(6):233–238.

Degroote, J., Bathe, K. J., and Vierendeels, J. (2009). Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction. Computers & Structures, 87(11):793–801.

Degroote, J., Haelterman, R., Annerel, S., Bruggeman, P., and Vierendeels, J. (2010). Performance of partitioned procedures in fluid-

structure interaction. Computers & structures, 88(7-8):446–457.
Diebels, S., Ellsiepen, P., and Ehlers, W. (1999). Error-controlled Runge-Kutta time integration of a viscoplastic hybrid two-phase model. Technische Mechanik, 19:19–27.
DVGW (2011). Nutzung von Gasen aus regenerativen Quellen in der öffentlichen Gasversorgung. DVGW-Arbeitsblatt G 262 (A) (Deutscher Verein des Gas-und Wasserfaches e. V.), Bonn, Germany.
Ellsiepen, P. (1999). Zeit- und ortsadaptive Verfahren angewandt auf Mehrphasenprobleme poröser Medien. Doctoral thesis, Institute of Mechanics II, University of Stuttgart. Report No. II-3.
Erbts, P. (2016). Partitioned solution strategies for electro-thermo-mechanical problems applied to the field-assisted sintering technology, volume 18 of Mechanik/Bruchmechanik. VDI Verlag, Düsseldorf.
Erbts, P. and Düster, A. (2012). Accelerated staggered coupling schemes for problems of thermoelasticity at finite strains. Computers & Mathematics with Applications, 64(8):2408–2430.
Erbts, P., Hartmann, S., and Düster, A. (2015). A partitioned solution approach for electro-thermo-mechanical problems. Archive of Applied Mechanics, 85:1075–1101.
Ferziger, J. H. and Peric, M. (2012). Computational methods for fluid dynamics. Springer Science & Business Media.
Greenshields, C. J. (2012a). OpenFOAM -The Open Source CFD Toolbox-Programmer's Guide.
Greenshields, C. J. (2012b). OpenFOAM -The Open Source CFD Toolbox-User's Guide.
Hairer, E., Lubich, C., and Roche, M. (1989). The numerical solution of differential-algebraic systems by Runge-Kutta methods. Springer-Verlag, Berlin.
Hairer, E., Norsett, S. P., and Wanner, G. (1993). Solving Ordinary Differential Equations I. Springer, Berlin, 2nd revised edition.
Hairer, E. and Wanner, G. (1996). Solving Ordinary Differential Equations II. Springer, Berlin, 2nd revised edition.
Hamkar, A.-W. (2013). Eine iterationsfreie Finite-Elemente Methode im Rahmen der finiten Thermoviskoelastizität. Phd-thesis, report no. 1/2013, Institute of Applied Mechanics, Clausthal University of Technology, Clausthal-Zellerfeld.
Hamkar, A.-W., Hartmann, S., and Rang, J. (2012). A stiffly accurate Rosenbrock-type method of order 2 applied to FE-analyses in finite strain viscoelasticity. Applied Numerical Mathematics, 62(12):1837 – 1848.
Hartmann, S. and Hamkar, A.-W. (2010). Rosenbrock-type methods applied to finite element computations within finite strain viscoelasticity. Computer Methods in Applied Mechanics and Engineering, 199(23-24):1455–1470.
Hartmann, S. and Rothe, S. (2013). A rigorous application of the method of vertical lines to coupled systems in finite element analysis. In Ansorge, R., Bijl, H., Meister, A., and Sonar, T., editors, Recent Developments in the Numerics of Nonlinear Hyperbolic Conservation Laws, volume 120 of Notes on Numerical Fluid Mechanics and Multidisciplinary Design, pages 161–175. Springer Berlin / Heidelberg.
Hartmann, S., Rothe, S., and Frage, N. (2013). Electro-thermo-elastic simulation of graphite tools used in SPS processes. In Altenbach, H., Forest, S., and Krivtsov, A., editors, Generalized Continua as Models of Materials, volume 22 of Advanced Structured Materials, pages 143 – 161. Springer, Berlin.
Irons, B. M. and Tuck, R. C. (1969). A version of the Aitken accelerator for computer iteration. International Journal for Numerical Methods in Engineering, 1(3):275–277.
Joosten, M. M., Dettmer, W. G., and Perić, D. (2009). Analysis of the block Gauss–Seidel solution procedure for a strongly coupled model problem with reference to fluid–structure interaction. International Journal for Numerical Methods in Engineering, 78(7):757–778.
Kepplinger, J., Crotogino, F., Donadei, S., and Wohlers, M. (2011). Present trends in compressed air energy and hydrogen storage in Germany. Clarks Summit, PA, USA. Solution Mining Research Institute, SMRI.
Klaus, T., Vollmer, C., Werner, K., Lehmann, H., and Müschen, K. (2010). Energieziel 2050: 100% Strom aus erneuerbaren Quellen. Umweltbundesamt, Dessau.
Lang, J. and Verwer, J. (2001). ROS3P – An accurate third-order Rosenbrock solver designed for parabolic problems. BIT, 41:731–738.
Lehmann, T. (1984). Elemente der Mechanik II,

Elastostatik. Vieweg, Braunschweig, 2nd edition.
Leonhard, W. (2008). Energiespeicher in Stromversorgungssystemen mit hohem Anteil erneuerbarer Energieträger. VDE Studie, Frankfurt.
Macleod, A. J. (1986). Acceleration of vector sequences by multi-dimensional δ methods. Communications in Applied Numerical Methods, 2(4):385–392.
Netz, T., Hamkar, A.-W., and Hartmann, S. (2013). High-order quasi-static finite element computations in space and time with application to finite strain viscoelasticity. Computers and Mathematics with Applications, 66:441 – 459.
Quint, K. J. (2012). Thermomechanically coupled processes for functionally graded materials: experiments, modelling, and finite element analysis using high-order DIRK-methods. Phd-thesis, report no. 2/2012, Institute of Applied Mechanics, Clausthal University of Technology, Clausthal-Zellerfeld.
Rothe, S., Erbts, P., Düster, A., and Hartmann, S. (2015a). Monolithic and partitioned coupling schemes for thermo-viscoplasticity. Computer Methods in Applied Mechanics and Engineering, 293:375 – 410.
Rothe, S., Hamkar, A.-W., Quint, K. J., and Hartmann, S. (2012). Comparison of diagonal implicit, linear-implicit and half-explicit Runge-Kutta methods in non-linear finite element analyses. Archive of Applied Mechanics, 82(8):1057 – 1074.
Rothe, S., Schmidt, J.-H., and Hartmann, S. (2015b). Analytical and numerical treatment of electro-thermo-mechanical coupling. Archive of Applied Mechanics, 85:1245–1264.
Sedlacek, R. (2014). Untertage-Gasspeicherung in Deutschland. Erdöl Erdgas Kohle, 130(11):402–410.
Sherif, S. A., Barbir, F., and Veziroglu, T. N. (2005). Wind energy and the hydrogen economy – review of the technology. Solar Energy, 78(5):647–660.
Shildip, D. U., Bhope, D. V., and Khamankar, S. D. (2015). Stress analysis of multilayer pressure vessel. Journal of Applied Mechanical Engineering, 4(2):1–6.
Stewart, R. B. and Roder, H. M. (1964). Properties of normal and parahydrogen. Pergamon Press, New York.
Strehmel, K. and Weiner, R. (1995). Numerik gewöhnlicher Differentialgleichungen. Teubner Verlag, Stuttgart.
Wendt, G., Erbts, P., and Düster, A. (2015). Partitioned coupling strategies for multi-physically coupled radiative heat transfer problems. Journal of Computational Physics, 300:327–351.
Wood, C., Gil, A. J., Hassan, O., and Bonet, J. (2010). Partitioned Block-Gauss–Seidel coupling for dynamic fluid–structure interaction. Computers & Structures, 88(23):1367–1382.
Wynn, P. (1962). Acceleration techniques for iterated vector and matrix problems. Mathematics of Computation, 16(79):301–322.
Yeh, M. and Kyriakides, S. (1986). On the collapse of inelastic thick-walled tubes under external pressure. ASME. Journal of Energy Resources and Technology, 108(1):35–47.

Projektdaten

The project is funded from SWZ with one TV-L E13 staff position since August 2015 at the site Clausthal. Involved scientists are:

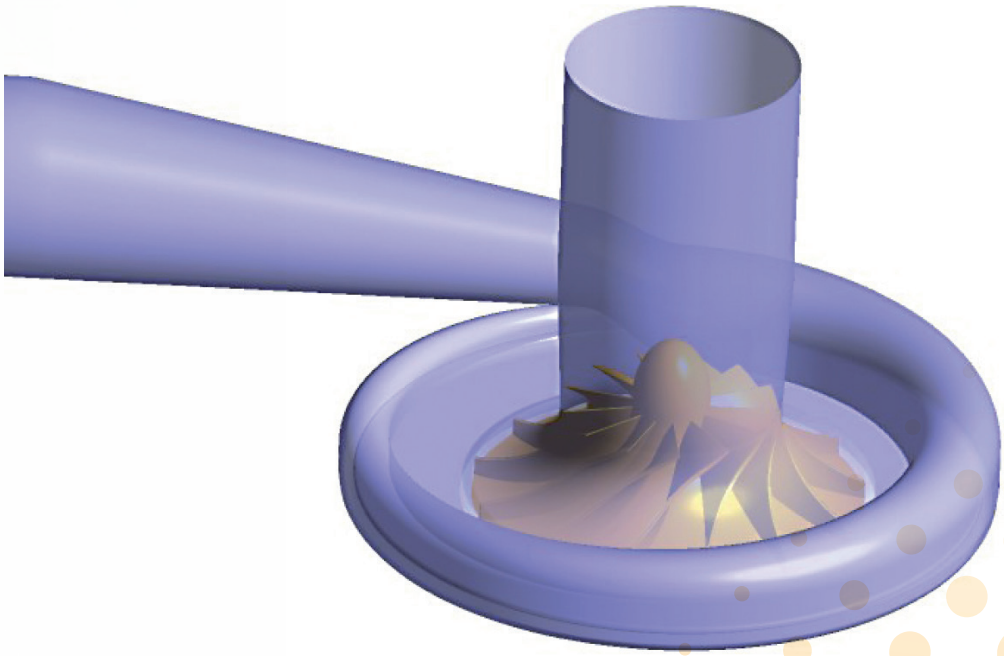
Prof. Dr. Leonhard Ganzer
Research Group
Reservoir Engineering
Institute of Petroleum
Engineering
Clausthal University
of Technology

Prof. Dr.-Ing.
Stefan Hartmann
Division of Solid Mechanics
Institute of Applied Mechanics
Clausthal University
of Technology

Verteilte Simulation

Die numerische Simulation technisch-wissenschaftlicher Probleme gehört traditionell zu den Disziplinen mit dem höchsten Bedarf an Rechenleistung. Dementsprechend werden solche Probleme auf Supercomputern mit Vektor- und Parallelrechnerarchitektur bearbeitet. Die größten Parallelrechner der Welt wurden in den USA und inzwischen in China für Simulationsanwendungen installiert. Die größten Parallelrechner in Deutschland sind deutlich kleiner und sind an den drei Standorten München (LRZ und RZG), Stuttgart (HLRS) und Jülich (FZ) konzentriert. In Niedersachsen stehen für große Simulationsaufgaben der HLRN III (Norddeutscher Verbund für Hoch- und Höchstleistungsrechnen) und zahlreiche kleinere Anlagen an universitären Rechenzentren zur Verfügung. Diese geographisch verteilten und heterogenen Ressourcen können für zahlreiche Fragestellungen besser genutzt werden. Man spricht dabei von verteilter Simulation oder Grid Computing. Der Aufwand für die Erstellung geeigneter Software und Middleware, die zwischen den Anwendungen und den Betriebssystemen sitzt, ist im Allgemeinen sehr hoch und es gibt enorme Einschränkungen für die Anwendungen. Bislang hat es sich deshalb nur in Einzelfällen gelohnt, Simulationen auf mehrere PC Cluster zu verteilen. Mit der Entwicklung leistungsfähigerer Hardware und neuer Berechnungsmethoden zeichnet es sich jedoch ab, dass der Aufwand für die Verteilung von Simulationsanwendungen auf existierende PCs und Workstations in einen Bereich kommt, wo er für viele Anwender interessant wird.

In diesem Projektbereich sollen daher Modelle und Methoden untersucht werden, mit denen Simulationen auf ein Grid verteilt werden können und die dabei entstehenden Probleme des Software-Tests und der Qualitätssicherung gelöst werden können. „Verteilte Simulation" meint aber auch die Simulation hochgradig verteilter realer Systeme, wie z.B. Lieferketten im Supply Chain Management, die nicht durch ein herkömmliches, geschlossenes Modell erfasst werden können.



Eine Cloud-basierte Software Infrastruktur für verteilte Simulationen – Interface Entwicklung

Fabian Glaser, Jens Grabowski

Zusammenfassung

Durch seine Skalierbarkeit und Flexibilität ist Cloud Computing eine interessante Option um den Ressourcenbedarf in den Simulationswissenschaften zu befriedigen. Die effiziente Nutzung von Cloud Ressourcen erfordert allerdings ein hohes Maß an technischer Expertise und stellt somit eine hohen Mehraufwand für Wissenschaftler dar, wenn sie ihre Berechnungen in die Cloud auslagern wollen. Daher werden Technologien benötigt, die das Bereitstellen von Simulationsanwendungen in der Cloud erleichtern. Um überdies eine Unterversorgung mit Cloudressourcen, welche unerwünschte Effekte auf die Performanz der Applikation hat und eine Überversorgung, welche unnötige Kosten verursacht, zu vermeiden, werden außerdem Methoden benötigt, um die bereitgestellten Ressourcen automatisch passend zu skalieren.

Modell-getriebene Entwicklung (englisch: Model-driven development (MDD)) ist eine Methodik welche formale Modelle benutzt um das zu entwickelnde System zu beschreiben. Hierbei kommen domänen-spezifische oder universelle Modellierungssprachen zum Einsatz. MDD hilft den Abstraktionsgrad zu erhöhen und führt dadurch zu einer besseren Verständlichkeit und Handhabbarkeit des modellierten Systems. Durch die Formalität der Modelle ist es außerdem möglich Modelltransformationen zu definieren, welche die automatische Generierung von ausführbarem Code oder Dokumentation ermöglichen. Überdies können Modelltransformationen dazu eingesetzt werden, die Modelle für einen bestimmten Anwendungsfall, wie zum Beispiel eine spezielle Repräsentation für eine bestimmte Nutzergruppe, anzupassen.

MDD ist eine passende Methodik, um die Komplexität der Bereitstellung von Simulationsanwendungen in der Cloud zu reduzieren und den Simulationswissenschaftler von komplizierten cloud-spezifischen Details abzuschirmen. Als Teil des SWZ-Projekt „Eine Cloud-basierte Infrastruktur für verteilte Simulationen“, haben wir deshalb analysiert, wie Simulationsanwendungen modelliert werden können um ihre automatische Bereitstellung in der Cloud zu ermöglichen. Überdies haben wir einen modellgetriebenen Ansatz entwickelt um den Umfang der bereitzustellenden Ressourcen automatisch zu ermitteln.

Publikationen

[a1] F. Glaser, H. Neukirchen, T. Rings, J. Grabowski, Using MapReduce for High Energy Physics Data Analysis , Proceedings of the 2013 International Symposium on MapReduce and Big Data Infrastructure (MR. BDI 2013), 03-05 December 2013, Sydney, Australia 2013, DOI: http://dx.doi.org/10.1109/CSE.2013.189 , 2013.

[a2] F. Glaser, J. N. Serrano, J. Grabowski, A. Quadt, ATLAS user analysis on private cloud resources at GoeGrid , Proceedings of the 21st Conference on Computing in High Energy Physics and Nuclear Physics (CHEP), 13-17 April, Okinawa, Japan, available online: http://iopscience.iop.org/article/10.1088/1742-6596/664/2/022020, 2015.

[a3] F. Glaser, Towards Domain-Model Optimized Deployment and Execution of Scientific Applications in Cloud Environments, Proceedings of the Doctoral Symposium at the 5th Conference on Cloud Computing and Services Sciences (DCCLOSER 2015), Lisbon, Portugal, 2015. (Best PhD project award).

[a4] M. Göttsche, F. Glaser, S. Herbold, J. Grabowski, Automated Deployment and Parallel Execution of Legacy Applications in Cloud Environments, The 8th IEEE International Conference on Service Oriented Computing & Applications (SOCA 2015), 2015.

[a5] F. Glaser, Domain Model Optimized Deployment and Execution of Cloud Applications with TOSCA, Proceedings of the 9th System Analysis and Modelling Conference (SAM 2016), Springer, 2016.

Studentische Arbeiten

[a6] M. Göttsche, Automated Deployment and Distributed Execution of Scientific Software in the Cloud using DevOps and Hadoop MapReduce, Masterarbeit im Studiengang Angewandte Informatik am Institut für Informatik, ZAI-MSC-2015-02, ISSN 1612-6793, Zentrum für Informatik, Georg-August-Universität Göttingen, 2015.

[a7] J. M. Erbel, Transformation of UML Deployment Models into executable Cloud Orchestration Templates, Bachelorarbeit im Studiengang Angewandte Informatik am Institut für Informatik, ZAI-BSC-2015-08, ISSN 1612-6793, Zentrum für Informatik, Georg-August-Universität Göttingen, 2015.

Awards

„Best PhD project award" für die Präsentation und das Paper „Towards Domain-Model Optimized Deployment and Execution of Scientific Applications in Cloud Environments" auf dem Doctoral Symposium der 5th Conference on Cloud Computing and Services Sciences (CLOSER 2015).

Projektdaten

Das Projekt wurde von Juli 2013 bis Juni 2016 vom SWZ mit insgesamt 0,5 TV-L E13 Stellen an dem Standort Göttingen gefördert. Beteiligte Wissenschaftler sind:

Prof. Dr. Jens Grabowski
Arbeitsgruppe Softwaretechnik für Verteilte Systeme
Institut für Informatik
Universität Göttingen

Fabian Glaser, M.Sc.
Arbeitsgruppe Softwaretechnik für Verteilte Systeme
Institut für Informatik
Universität Göttingen

Xiaowei Wang, M.Sc.
Arbeitsgruppe Softwaretechnik für Verteilte Systeme
Institut für Informatik
Universität Göttingen

Simulations-Basierte Qualitätssicherung von Softwaresystemen

Jens Grabowski, Verena Herbold, Daniel Honsel, Stephan Waack, Marlon Welter

Einleitung

Die Qualitätssicherung von Softwaresystemen ist eines der Forschungsgebiete, das in den letzten Jahrzehnten an Popularität dazugewinnen konnte. Mittlerweile ist Software zu einem essentiellen Bestandteil des alltäglichen Lebens geworden. Die Komplexität von Software steigt an, was auch daran liegt, dass bestehende Softwaresysteme kontinuierlich den aktuellen Anforderungen angepasst und erweitert werden müssen. Aus diesem Grund besteht großes Interesse an der Überwachung der Qualität eines Systems während dessen Evolutionsprozesses sowie der Vorhersage der Qualität für einen konfigurierbaren Projektverlauf. Je eher ein Projektmanager auf ein Problem hingewiesen wird, umso eher kann er darauf reagieren und entsprechende Gegenmaßnahmen einleiten. Dies könnten z.B. Veränderungen an der Zusammensetzung des Entwickelteams sein. Durch die Wahl eines simulationsbasierten Ansatzes können unterschiedliche Szenarien durchgespielt und deren Resultate miteinander verglichen werden.

Ansatz

Unser Ansatz besteht aus drei unterschiedlichen Forschungsbereichen: Datengewinnung aus Software Repositorien, Agentenbasierte Modellierung und Simulation, Bewertung mittels bedingter Markovscher Zufallsfelder. Wie diese drei Bereiche zusammenspielen, mit der Absicht die Qualität eines simulierten Softwareprojekts zu bewerten, wird im Folgenden genauer beschrieben.
Zu Beginn haben wir eine konkrete Fragestellung, die es zu beantworten gilt. Eine interessante Frage könnte z.B. sein: Was passiert, wenn ein Hauptentwickler das Projekt nach einer bestimmten Zeit verlässt? Als erstes wird ein agentenbasiertes Modell erstellt, welches die Frage beantworten soll. Um das Modell zu parametrisieren, benötigen wir Daten, die aus ausgewählten Open Source Repositorien gewonnen werden. Mit den daraus geschätzten Parametern kann die Simulation ausgeführt werden. Sollte das Modell noch nicht zur Fragestellung passen, kann es beliebig angepasst werden bis es zu einer Antwort führt. Abschließend werden die simulierten Ergebnisse mit Hilfe von bedingten Markovschen Zufallsfeldern bewertet. Nachfolgend werden die drei Bereich genauer erläutert.

Verantwortlich für den Evolutionsprozess von Software sind Entwickler und ihr Commit-Verhalten. Eine Generalisierung von Entwicklern ist nicht möglich, da die Arbeit von Individuen oft vom Durchschnitt abweicht. Aus diesem Grund werden Entwickler in verschiedene Typen eingeteilt. Die Klassifizierung basiert auf den Daten der Repositorien und jeder Typ weist eine andere Commit-Häufigkeit sowie eine andere Wahrscheinlichkeit einen Fehler in der Software zu beheben auf.
Um eine Qualitätsaussagen über die Software machen zu können, benötigen wir die Anzahl der Fehler, die einer Datei zugeordnet sind. Dazu werden unterschiedliche Fehlertypen sowie deren Entstehungsraten ermittelt.

Um Abhängigkeiten zwischen Dateien zu beschreiben wird ein sogenannter Change-Coupling-Graph benutzt. Jeder Knoten repräsentiert eine Datei und jedes Mal, wenn Dateien zusammen in einem Commit verändert werden, wird eine neue Kante zwischen ihnen gezogen. Existiert bereits eine Kante, so wird ihr Gewicht erhöht. Diese Netzwerke und dessen Veränderungen über den Projektverlauf werden ebenfalls aus den Software Repositorien gewonnen. Enthaltene Cluster stellen logische Unterteilungen der Software da, wie z.B. Module oder Pakete.

Passend zu der oben genannten Fragestellung wird ein agentenbasiertes Simulationsmodell entwickelt und mit den Daten aus den Repositorien parametrisiert. Ein Agent ist ein Individuum, das autonom in seiner Umgebung agiert. Es wird

zwischen aktiven und passiven Agenten unterschieden. Die aktiven können den Zustand der passiven verändern, umgekehrt geht das nicht. Wir haben nur einen aktiven Agententyp, den Entwickler. Passive Agenten sind Dateien und Fehler. Abhängigkeiten zwischen den Agenten werden mit Netzwerken modelliert. Unter anderem wird auch der Change-Coupling-Graph simuliert. Das bietet die Möglichkeit die Struktur der simulierten Software mit der realen Software zu vergleichen. Zu Beginn der Simulation wird eine bestimmte, feste Anzahl von Entwicklern jedes Typs instanziiert. Diese sind dann, beschrieben durch ihr jeweiliges Commit-Verhalten, verantwortlich für die Entstehung der Software. Des Weiteren können mit einem Commit auch Fehler behoben werden. Ob ein Fehler gefixt wird ist abhängig vom Fehlertyp, Entwicklertyp und Vernetzungsgrad der fehlerbehafteten Datei.
Um die simulierte Software bewerten zu können, wird jede Datei vorläufig als akzeptabel oder problematisch gekennzeichnet. Diese Kennzeichnung basiert auf der Anzahl und dem Typ der einer Datei angehefteten Fehler. Der simulierte, gekennzeichnete Change-Coupling-Graph dient als Grundlage für die weitere Auswertung.
Der Zustand der simulierten Software wird mit bedingten Markovschen Zufallsfeldern bewertet. Hierbei fließen auch Abhängigkeiten zwischen den Dateien in die Bewertung ein. Hat z.B. die Datei A eine schlechte Qualität und die Datei B hängt von A ab, so ist die Qualität von B davon beeinträchtigt. Des Weiteren sollen größere, zusammenhängende Teile der Software eine homogene Bewertung erhalten.

Ergebnisse

Basierend auf durchgeführten Fallstudien konnten wir nachweisen, dass Agent-basierte Simulation ein vielversprechender Ansatz zur Softwarequalitätssicherung ist. Mit dem hier beschriebenen Ansatz ist es möglich einen Qualitätstrend eines Softwareprojekts, bezüglich einer konkreten Fragestellung, zu analysieren. Für die Frage nach dem Wegfall eines Hauptentwicklers haben wir unser Simulationsmodell mit Parametern eines Open Source Projekts eingestellt und zwei verschieden Verläufe simuliert. Einen Lauf komplett mit Hauptentwickler und einen Lauf bei dem der Hauptentwickler nach der Hälfte Projektlaufzeit wegfällt. Als Resultat erhielten wir, dass ohne den Hauptentwickler etwa 20% weniger Fehler behoben werden.

Um das Ergebnis zu validieren haben wir nur wenige Parameter des Modells (Aufwand, Projektgröße und Projektdauer) entsprechend eines anderen Open Source Projekts eingestellt und erhielten einen ähnlichen Qualitätstrend.

Insgesamt haben wir unsere Ergebnisse für zwei weitere Projekte bestätigen können. Diese Projekte haben eine ähnliche Laufzeit und eine ähnliche Größe wie das Originalprojekt, unterscheiden sich jedoch deutlich in der Anzahl der Instanzen der unterschiedlichen Entwicklertypen.

Stipendien

ACM SIGSOFT Travel Grant: $1.200 für Verena Honsel zur Teilnahme an der International Conference on Automated Software Engineering (ASE) inklusive des Workshops on Software Mining (SoftMine) 2015 in Lincoln, Nebraska, USA

Preise

ACM SIGSOFT Distinguished Paper Award auf der MSR 2016 für das Paper „Adressing Problems with External Validity of Repository Mining Studies Through a Smart Data Platform“

Projektdaten

Das Projekt wurde von Mai 2013 bis April 2016 vom SWZ mit insgesamt einer TV-L E13 Stelle an dem Standort Göttingen gefördert. Beteiligte Wissenschaftler sind:

Prof. Dr. Jens Grabowski
Arbeitsgruppe Softwaretechnik
für Verteilte Systeme
Institut für Informatik,
Universität Göttingen

Dipl.-Math. Verena Herbold
Arbeitsgruppe Softwaretechnik
für Verteilte Systeme
Institut für Informatik
Universität Göttingen

Prof. Dr. Stephan Waack
Arbeitsgruppe Theoretische
Informatik und Algorithmische
Methoden
Institut für Informatik,
Universität Göttingen

Marlon Welter, M.Sc.
Arbeitsgruppe Theoretische Informatik
und Algorithmische Methoden
Institut für Informatik,
Universität Göttingen

Dipl.-Inf. Daniel Honsel
Arbeitsgruppe Theoretische
Informatik und Algorithmische
Methoden
Institut für Informatik,
Universität Göttingen

DeSim: Dezentrale Architekturen und Konzepte für die Simulation von Systems of Systems

Jürgen Dix, Jörg P. Müller

Zusammenfassung

Das DESIM-Projekt befasst sich mit der Modellierung, Analyse und Simulation komplexer Systems of Systems (SoS), von Systemen also, die wiederum aus komplexen und heterogenen Systemen bestehen, die in loser Kopplung miteinander interagieren und in vielen Fällen emergentes Verhalten aufweisen. Typischerweise ist der Mensch Bestandteil eines SoS, muss also in seinem Verhalten mitsimuliert werden. Ein gutes Beispiel für ein SoS ist ein Verkehrssystem, in dem Menschen, aber auch automatisierte Akteure wie autonome Fahrzeuge Routenwahlentscheidungen treffen, die wiederum einen Einfluss auf Verkehrsfluss und Effizienz des Gesamtsystems haben (McCluskey et al., 2016). Weitere Anwendungen liegen in den Bereichen dezentral organisierter Produktionsplanungs- und Steuerungsmethoden im Bereich Industrie 4.0, in der selbstorganisierenden Logistik sowie in der Medizintechnik (intelligente Implantate).

Traditionelle Methoden und Paradigmen der Simulation wie System Dynamics und Diskrete Modelle stoßen bei den Anforderungen der SoS an ihre Grenzen, da sie Konzepte wie Autonomie und Dezentralität und komplexes, kognitives Verhalten weder in der Ausführung, noch in der

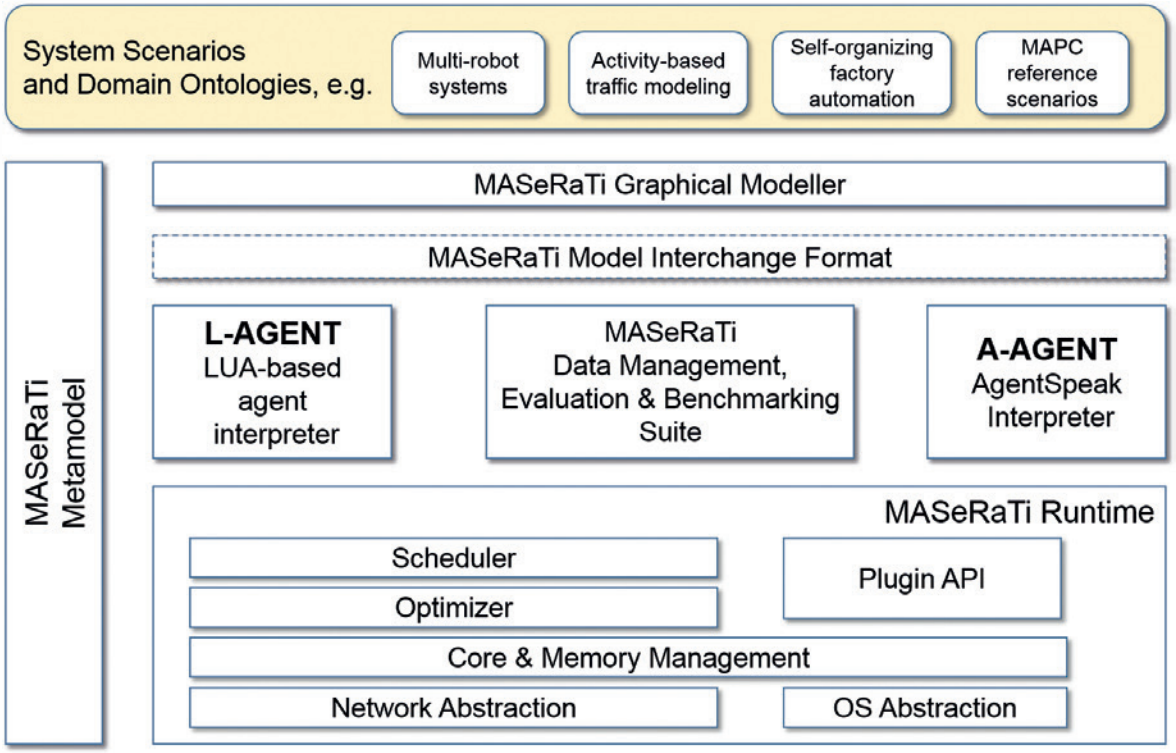


Abb. 1: Architektur des MASeRaTi Modelierungs- und Simulations-Frameworks

Modellerstellung angemessen unterstützen. Einen vielversprechenden Ansatz stellt hier die Multiagenten-basierte Modellierung und Simulation (MABS) dar, einer Form der Mikrosimulation, die Modellierungsabstraktionen für Konzepte wie wissensbasierte Entscheidung, Planung und Planausführung, Kooperation, Koordination, Autonomie und die Darstellung gemischter Szenarien mit menschlichen und maschinellen Akteuren und deren Interaktion bietet (Bazzan and Klügl, 2013). Allerdings fehlt es hier an standardisierten Modellen unterhalb der konzeptuellen Ebene und an performanten, skalierbaren Laufzeitplattformen, die es ermöglichen, die sehr rechenintensiven MABS-Modelle für größere Problemstellungen mit beschränkten Ressourcen in akzeptabler Zeit zu simulieren.

Ziel des DESIM-Projekts war es, einen Beitrag zur Entwicklung nutzbarer und skalierbarer MABS-Methoden zu leisten. Nach einer Analyse der Anforderungen und des Standes der Forschung in diesem Bereich wurde die Notwendigkeit festgestellt, eine einheitliche flexible Gesamtarchitektur für ein agentenbasiertes Modellierungs- und Simulationsframework zu konzipieren. Abb. 1 zeigt das erste wesentliche aus diesen Aktivitäten resultierende Ergebnis, die Simulationsarchitektur MASeRaTi (Ahlbrecht et al., 2014).

Der innovative Kern des Frameworks ist die verteilte Laufzeitumgebung (MaSeRaTi Runtime), die letztlich für das Scheduling der Agenten und die Ausführung ihrer Aktionen verantwortlich ist. Schwerpunkt liegt dabei auf der Nutzbarkeit massiv paralleler Hardware, die durch Nutzung des MPI-Standards verwendet wird.

In der ersten Projektphase erfolgte die Spezifikation des MaSeRaTi-Frameworks und eine prototypische Realisierung (Ahlbrecht et al., 2014). In der zweiten Projektphase wurde zum einen anhand von Beispielanwendungen (dem Cows-Szenario aus dem internationalen Multi-Agent Programming Contest (MAPC) und einem Verkehrssimulationsszenario (s. Abb. 2) die MaSeRaTi Runtime mit der State-of-the-Art Softwarelösung Jason verglichen (Ahlbrecht et al., 2016).

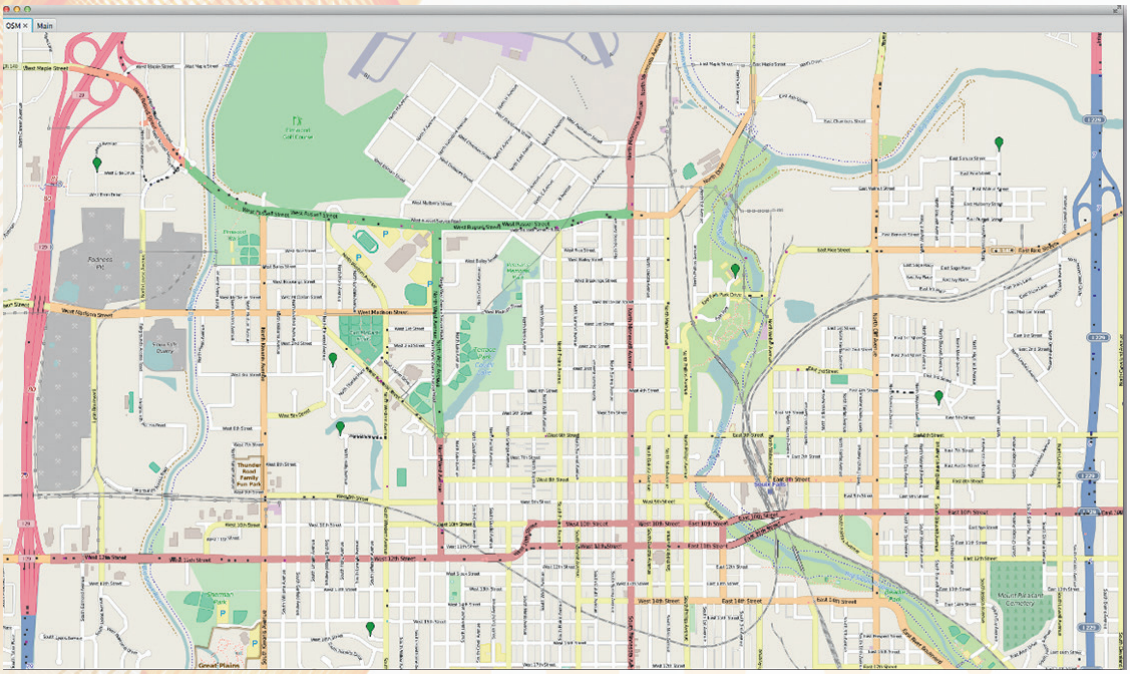


Abb. 2: Das Sioux Falls Benchmark-Szenario für Verkehrssimulation (Fahrzeugagenten sind als kleine schwarze Punkte dargestellt)

Die Tests belegen eine deutlich verbesserte Performanz und Skalierbarkeit von MaSeRaTi. Dies bedeutet, dass mit MaSeRaTi deutlich größere Modelle simuliert werden können. Zum zweiten erfolgte eine Portierung der MaSeRaTi Runtime in die weit verbreitete Sprache Java. Zum dritten wurde der prototypische AgentSpeak(L)-Interpreter in die Agentensprache LightJason weiterentwickelt und realisiert, die die Modellierung / Programmierung kognitiver Agenten mit dem Belief, Desire, Intentions Modell (Rao and Georgeff, 1991) und ihre effiziente Ausführung in der MaSeRaTi Architektur ermöglicht. Das letzte nennenswerte Ergebnis des Projekts ist auf der Modellierungsseite angesiedelt. Basierend auf einem zuvor definierten Metamodell wurde ein grafisches Modellierungswerkzeug für Agentenprogramme definiert (Ahlbrecht 2016).

Zusammenfassend ist im DESIM-Projekt ein wichtiger erster Schritt zur Entwicklung einer innovativen Plattform für die multiagenten-basierte Simulation komplexer Systeme mit kognitiven Agenten gelungen. Die lauffähige Software steht der wissenschaftlichen Community unter http://lightjason.org zur Verfügung.

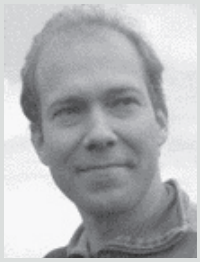
Ein Bedarf für weitere Forschung auf diesem Gebiet besteht in vieler Hinsicht, um die Simulation noch größerer und komplexerer Szenarios mit noch reichhaltigerem und komplexerem kognitiven Verhalten (wie lernende Agenten oder effiziente Kommunikation) zu unterstützen. Neben Weiterentwicklungen der Ausführungsplattform sind auch eine verbesserte Werkzeugunterstützung sowie die leichtere Anbindung domänenspezifischer Funktionen und Modelle von großer Bedeutung. Um letztlich nicht nur Einzelsysteme (wie ein städtisches Straßenverkehrssystem), sondern eine ganze Smart City mit all ihren Teilsystemen und Interaktionen modellieren und simulieren zu können, sind weitere innovative Modelle und Methoden, z.B. in den Bereichen Multilevel-Simulation und Hybride Modelle erforderlich.

Projektdaten

Das Projekt wurde von Juli 2013 bis Juni 2016 vom SWZ mit insgesamt einer TV-L E13 Stelle an dem Standort Clausthal gefördert. Beteiligte Wissenschaftler sind:

Prof. Dr. Jürgen Dix
Arbeitsgruppe Computational Intelligence

Institut für Informatik
Technische Universität Clausthal



Prof. Dr. Jörg P. Müller
Arbeitsgruppe Wirtschaftsinformatik

Institut für Informatik
Technische Universität Clausthal

Auf dem Weg zu einer Multi-Level-Simulation mit Hilfe dynamischer Cloud-Umgebungen

Stefan H. A. Wittek, Michael Göttsche, Andreas Rausch und Jens Grabowski

Die Konstruktion von cyber-physikalischen Systemen erfordert die Anwendung holistischer Simulationen. Um der Komplexität dieser Systeme gerecht zu werden, streben wir eine effiziente Simulationsmethodik an. Die benötigte holistische Perspektive wird auf dem Groblevel erreicht, welches mit mehreren Modelle auf dem Detaillevel co-simuliert wird, um Teile des Systems zu fokussieren, die von besonderem Erkenntnisinteresse sind. Welches diese Teile sind, in die „hineingezoomed" wird, ist dynamisch während eines Simulationslaufes. Dieser Dynamik wird auf Recheninfrastrukturebene Rechnung getragen, indem die resultierende Multi-Level-Simulation in einer Cloud-Umgebung mit dynamisch allokierten Ressourcen ausgerollt und somit die Verschwendung von Ressourcen vermieden wird. Unsere Forschung gliedert sich in die Untersuchung der Methodik der Multi-Level-Simulationen und in die Entwicklung eines Deploymentansatzes zur dynamischen Ausbringung solcher Simulationen in Cloud-Umgebungen.

Für eine erste Evaluation unseres Ansatzes haben wir einen Prototyp entwickelt, der einen Fahrstuhl in Form einer Multi-Level-Simulation nachbildet. Auf dem Groblevel simuliert er den Lift und das Treiberprogramm, auf dem detaillierten Level das Kabel. Zum aktuellen Zeitpunkt sind die Abbildungsfunktionen noch statisch, die ersten Ergebnisse sind aber vielversprechend hinsichtlich der Synchronität der einzelnen Komponenten. Ebenso noch statisch ist derzeit das Deployment des Prototyps. Jede der Komponenten wird auf einer eigenen Maschine, mithin also verteilt, ausgebracht. Die Kommunikation zwischen den einzelnen Teilen wird dabei über den auf RMI basierenden Simulationsbus abgewickelt.

Diese prototypische Implementation ist hilfreich, um erste Ergebnisse zur Methodik zu gewinnen.

In unseren nächsten Schritten werden wir sukzessive auf einen realweltlichen Anwendungsfall umstellen. Hierzu werden die handkodierten Abbildungsfunktionen durch maschinell gelernte ersetzt werden. Eine Herausforderung hierbei wird die Berücksichtigung von nutzerdefinierten Restriktionen sein. Des Weiteren ist die dynamische Aktivierung und Deaktivierung von Komponenten zur Laufzeit vorgesehen.

Der nächste Schritt hinsichtlich des dynamischen Cloud-Deployments wird die Evaluation geeigneter Strategien für die Skalierung der Ressourcen sein. Hierzu werden wir reale Anwendungsfälle untersuchen, um aus diesen die lastdeterminierenden Parameter zu extrahieren. Diese werden im Anschluss auf ihre Prognosekraft untersucht werden und Teil eines integrierten Frameworks für das dynamische Deployment sein.

Publikationen

[a1] S. Wittek, M. Göttsche, A. Rausch, J. Grabowski, Towards Multi-Level-Simulation using Dynamic Cloud Environments, Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2016), Lisbon, Portugal, July 29-31, 2016. SciTePress 2016, ISBN 978-989-758-199-1.

Studentische Arbeiten

[a2] R. Tasik, Agent-based Simulation of Workflow Scheduling and Cloud Deployments, Bachelorarbeit im Studiengang Angewandte Informatik am Institut für Informatik, ZAI-BSC-2016-17, ISSN 1612-6793, Zentrum für Informatik, Georg-August-Universität Göttingen, 2016.

Awards

„Best Power Award“ für die Präsentation des Papers „Towards Multi-Level-Simulation using Dynamic Cloud Environments“ auf der 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2016)

Projektdaten

Das Projekt wird seit August 2015 vom SWZ mit insgesamt einer TV-L E13 Stelle an den Standorten Clausthal und Göttingen gefördert. Beteiligte Wissenschaftler sind:

Prof. Dr. Andreas Rausch
Software Systems Engineering
Institut für Informatik
Technische Universität
Clausthal

Stefan Wittek, M.Sc.
Software Systems Engineering
Institut für Informatik
Technische Universität Clausthal

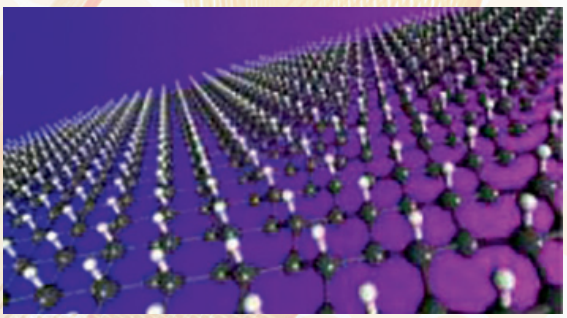
Prof. Dr. Jens Grabowski
Arbeitsgruppe Softwaretechnik
für Verteilte Systeme
Institut für Informatik
Universität Göttingen

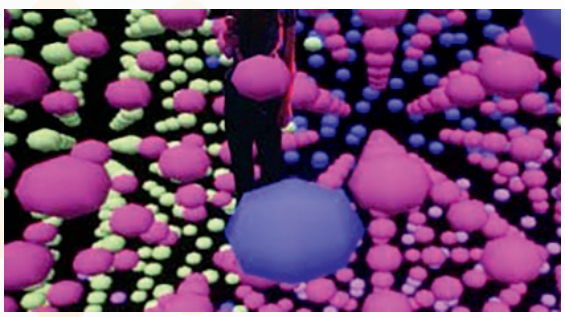
Michael Göttsche, M.Sc.
Arbeitsgruppe Softwaretechnik
für Verteilte Systeme
Institut für Informatik
Universität Göttingen

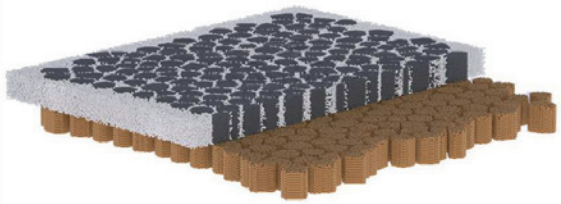
Cloud-Efficient Modelling and Simulation of Magnetic Nano Materials

Pavle Ivanovic, Harald Richter, Arezoo Bozorgmehr

Der vorliegende Bericht folgt einem trans-disziplinären Ansatz, indem er gleichberechtigt zueinander sowohl magnetische Nano-Materialien, deren mathematische Modellierung, die dafür verfügbaren Löser und deren effiziente Ausführung in einer Cloud beschreibt. Der Bericht ist folgendermaßen strukturiert: im ersten Kapitel wird erklärt, warum magnetische Nano-Materialien von hohem Interesse für zukünftige Rechnerarchitekturen sind. Es werden allerdings für Forschungen auf diesem Gebiet ausgeklügelte Ultrahochvakuum-Apparaturen und Rastertunnel-Mikroskope benötigt, sowie ein Zugang zu einem Supercomputer oder Parallelrechner. Deswegen schlagen wir vor, magnetische Nano-Materialien auf einer Cloud, als dem derzeit vorherrschen- den Rechenparadigma, zu simulieren, um so aufwendige Laborarbeiten zu verringern. In Kapitel 2 des Berichts wird der Stand der Technik dargestellt, und zwar sowohl bei der Modellierung magnetischer Nano-Materialien als auch bei den Fähigkeiten von Clouds zu effizientem Hochleistungsrechnen (HPC). Die mathematische Modellierung beruht auf der sog. Landau-Lifschitz-Gilbert-Gleichung (LLGE), die bereits in die 1930er Jahre zurückgeht. Trotzdem gibt es bis heute für die LLGE nur eine kleine Zahl von Lösern, die das Message Passing Interface (MPI) verwenden, was es ermöglicht, einen Solver auf die Server einer Cloud verteilen zu können. Zu dieser kleinen Solver-Zahl gehören die Softwarepakete „Nmag“, „Magpar“ und „Vampire“. Im Kapitel 2 des Berichts wird ebenso ein kurzer Überblick zu den Solver gegeben, sowie eine Zusammenfassung zum Thema „Virtualisierung“ und zu „OpenStack“ erstellt, was das Cloud-Betriebssystem ist, das wir im Projekt benützen. In Kapitel 3 wird die LLGE hinsichtlich ihrer Inter-Prozesskommunikation neu bewertet, um damit die Solver-Algorithmen Cloud-effizienter machen zu können. Darüber hinaus haben wir herausgefunden, dass in den Lösern verschiedene algorithmische Änderungen wünschenswert wären und haben diese im Detail beschrieben. Nach dieser Beschreibung folgt im Report eine kurze Schilderung der Hard- und Software unserer Cloud, die wir benützt haben, um die Rechenzeiten der Solver zu messen. In unserer privaten Cloud werden wir auch in den kommenden zwei Jahren Messungen vornehmen, insbesondere nachdem alle algorithmischen Änderungen vollzogen worden sind. Bisher lag der erste wichtige Schritt im Projekt darin, den Verwaltungszusatzaufwand benennen und im Kapitel 3 genau beschreiben zu können, der für die Inter-VM-Kommunikation notwendig ist. Aufgrund dieses Overheads haben wir festgestellt, dass eine Standard-Cloud einen Supercomputer oder einen Parallelrechner nicht ersetzen können wird. Im Projekt ist es unsere Aufgabe dies zu ändern. In Kapitel 4 des Berichts werden die Messungen beschrieben, die wir gemacht haben,







um die HPC-Effizienz einer Standard-Cloud besser quantifizieren zu können. Dazu haben wir ein rechteckiges Stück ferromagnetischen Materials (Permalloy) der Größe 30x30x100 nm3 im Rechner simuliert. Das Gitter, in das der Permalloy-Quader im Rechner zerlegt wurde, bestand aus 4,100 Punkten, die insgesamt 18,780 kleine Tetraeder bildeten. Die Eingabedateien, die wir für die Solver Nmag, Magpar und Vampire definiert haben, beschreiben dieselbe Geometrie, dieselbe Anzahl von Gitterpunkten, dasselbe ferromagnetische Material, dieselbe Anfangsmagnetisierung, sowie dieselben Simulationsparameter, wie z.B. Anzahl und Dauer der Zeitschritte. Damit haben wir es schließlich erreicht, dass die simulierten Ergebnisse vollständig miteinander übereinstimmten. Mit Hilfe von vier verschiedenen Simulationsszenarien konnten wir außerdem zeigen, dass Speedup und Effizienz in unserer Cloud zu klein für effizientes HPC sind, und dass die Solver auch nicht hin zu einer größeren Zahl von vCPUs in der Cloud skalierbar sind. Im Kapitel 5 schließlich schlagen wir drei Methoden vor, um eine beliebige Standard-Cloud HPC-effizient zu machen. Diese Methoden beruhen darauf, allen Cloud-Overhead bei der Inter-vCPU-, Inter-VM- und Inter-Serverkommunikation zu beseitigen. Letztere konnte dadurch eliminiert werden, dass eine Direktverbindung zwischen MPI und Infiniband in den Host-Betriebssystemen aller Teile der parallelen Solver geschaffen wurde. Für diese Direktverbindung wurde ein virtuelles PCI-Gerät namens „ivshmem“ benützt, das einen gemeinsamen Speicher zwischen VMs etabliert, und das von Qemu emuliert wird. Außerdem wurden alle MPI-Bibliotheksaufrufe in den Gastbetriebssystemen von der libibverbs-Bibliothek, die für Infiniband notwendig ist, abgekoppelt, indem ein tiefer Schnitt in die Solver-Software gemacht wurde. Schließlich schlagen wir vor, ivshmem auch zwischen den VMs auf demselben Server einzuführen, wofür Infiniband nicht notwendig ist, und wozu der Software-Schnitt sowie unsere Adapter dazwischen entfallen können. Schließlich waren wir in der Lage, eine ähnliche Methode auch für die Inter-vCPU-Kommunikation auf demselben Server einzuführen. Es wird erwartet, dass durch diese Methoden wesentliche Verbesserungen in Bezug auf die HPC-Effizienz von Standard-Clouds erzielt werden. Der Projektbericht endet mit einer Schlussfolgerung und einer Literaturliste.

Projektdaten

Das Projekt wird seit August 2015 vom SWZ mit insgesamt 0,5 TV-L E13 Stellen an dem Standort Clausthal gefördert. Beteiligter Wissenschaftler ist:

Prof. Dr. Harald Richter
Arbeitsgruppe Technische Informatik und Rechnersysteme
Institut für Informatik
Technische Universität Clausthal

Numerisch intensive Simulationen auf einer integrierten Recheninfrastruktur

Fabian Glaser, Jens Grabowski, Alexander Bufe, Gunther Brenner, Christian Köhler, Philipp Wieder

Zusammenfassung
Die Anwender von Simulationssoftware sind angesichts der heterogenen Infrastruktur, wie sie oft in heutigen Rechenzentren vorzufinden ist, mit der Aufgabe konfrontiert die am besten geeigneten Rechenressourcen für ihren Anwendungsfall sowie die Skala, auf welcher diese bereitzustellen sind, zu bestimmen, letztendlich aber auch die technischen Details für ihre Verwendung zu erlernen, was in jedem Fall eine Investition von Zeit voraussetzt.

Die von uns gewählte Kombination aus Anwendungsfall, zunächst betrachteter Infrastruktur und technischem Ansatz um diese auf einfache Art und Weise verfügbar zu machen ist durch mehrere Forschungsdisziplinen motiviert:
Aus ingenieurwissenschaftlicher Sicht ist es Ziel des Projektes, das Verständnis von Stofftransport- und Umwandlung in porösen Medien zu verbessern, begründet durch den Bedarf aus verschiedenen Bereichen der Ingenieurwissenschaften wie der chemischen Verfahrenstechnik poröse Materialien gezielt zu designen und beispielsweise als Träger für Katalysatoren mit optimierten Umsatzraten und Selektivität einzusetzen. Hierfür ist es notwendig, die Morphologie-Transport-Beziehungen unter Einbeziehung von chemischen Reaktionen genauer als bisher möglich vorhersagen zu können.

Die relevanten mathematischen Methoden aus der numerischen Strömungsmechanik schließen insbesondere die Lattice-Boltzmann Methode (LBM) unter Anwendung adaptiver Gitterverfeinerung ein. Diese müssen jedoch noch weiterentwickelt werden um auch für die technisch relevanten porösen Strukturen die auf sehr unterschiedlichen Längenskalen ablaufenden Prozesse auflösen zu können.

Die durchgeführten Simulationen dienen hierbei nicht nur zur Verbesserung des ingenieurwissenschaftlichen Verständnisses, sondern auch zur Evaluation der weiteren in diesem Projekt durchgeführten Entwicklungen. Sie sind hierfür hervorragend geeignet, da sie eine Vielzahl von Anwendungsfällen mit unterschiedlichen Anforderungen an die Infrastruktur darstellen. Parameterstudien, bei denen das gleiche Programm mehrmals mit einem anderen Parameter gestartet wird, stellen beispielsweise wesentlich andere Anforderungen an die Netzwerkverbindung als eine einzelne Simulation mit einem einzigen großem Rechengebiet und der Übergang vom Entwicklungs- in den Produktivbetrieb kann eine Verwendung anderer Ressourcen sinnvoll sein.

Aus Sicht der der Informatik ist ein Aspekt die Parallelisierung des Simulationscodes um die Methode auf aktuellen HPC Clustern mit GPU-Beschleunigerkarten anwendbar zu machen. Außerdem sollte die Möglichkeit die Fähigkeiten moderner heterogener Infrastrukturen, insbesondere IaaS Clouds und HPC Cluster, zu nutzen dem Simulationsanwender auf einfach nutzbare Weise zugänglich sein einschließlich einer automatischen Auswahl der am besten geeigneten Ressource.

Eine zentrale Komponente der im Rahmen dieses Projektes durchgeführten Entwicklungen ist die Verteilungsschicht („Distribution Layer"), welche die Aufgabe übernehmen soll zwischen den Applikationsmodellen für Simulationscode auf der einen Seite und den verschiedenen Ressourcentypen auf der anderen Seite zu vermitteln, ob diese nun auf einer Cloud basieren oder HPC Cluster mit evtl. Beschleunigerkarten sind. Dazu sind Anforderungen an die Ressourcen, die entweder vom Simulationsanwender direkt festgelegt oder aus den Simulationsparametern und Eingabedaten extrahiert werden, auszuwerten und eine sinnvolle Wahl unter den verfügbaren Ressourcen zu treffen.

Die Kombination aus einem Applikationsmodell mit gewählten Parametern und dem Infrastruktur-

modell liefert anschließend u.a. die instanzierten Modelle, die für Verteilungs-, Ausführungs- und Abwicklungsschritte relevant sind, welche durch den Simulationsanwender per Kommandozeilenschnittstelle ausgelöst werden können. Insbesondere sind mehrere Ausführungsschritte mit variierenden Parametern möglich, um bspw. Parameterstudien zu realisieren, für welche keine pauschal beste Wahl der Infrastruktur existiert, oder um mit erhöhter Menge des zu reservierenden Speichers der Gitterverfeinerung Rechnung zu tragen, die sich in einem vorhergehenden Simulationsschritt ergeben hat.

In Anbetracht des vorhergehenden SWZ Projekts 11.4.1 soll die darin entwickelte Lösung zur automatischen Bereitstellung von Cloudressourcen zu Simulationszwecken auf die Interaktion mit den im HPC-Bereich üblichen Batchsystemen erweitert werden.

Projektdaten

Das Projekt wird von März 2016 bis Februar 2019 vom SWZ mit insgesamt 2 TV-L E13 Stellen an dem Standort Clausthal gefördert. Beteiligte Wissenschaftler sind:

Prof. Dr.-Ing. Gunther Brenner
Arbeitsgruppe Strömungsmechanik
Institut für Technische Mechanik
Technische Universität Clausthal

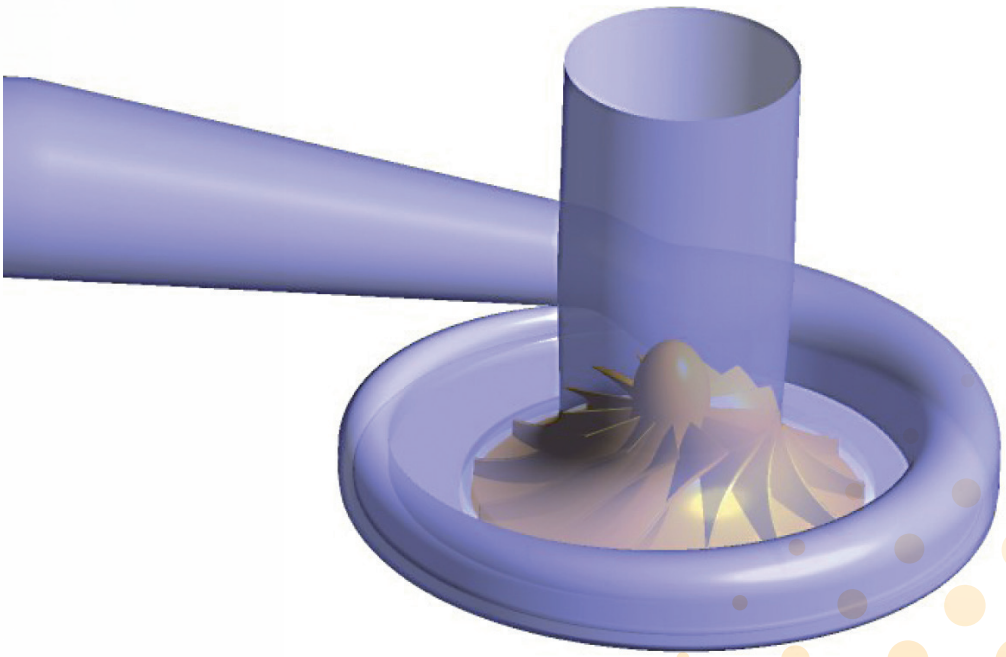
Prof. Dr. Ramin Yahyapour
Arbeitsgruppe Praktische Informatik
Institut für Informatik
Georg-August-Universität Göttingen

Prof. Dr. Jens Grabowski
Arbeitsgruppe Softwaretechnik für Verteilte Systeme
Institut für Informatik
Georg-August-Universität Göttingen

Distributed Simulation

Numerical simulation of technical and scientific problems traditionally is one of the disciplines with the highest demand of computing power. Accordingly, such problems are processed on supercomputers with vector and parallel computing architecture. The most powerful parallel computers in the world have been installed in the US and in China for simulation applications. The most powerful parallel computers in Germany are much smaller than these machines and are concentrated at three sites in Munich (LRZ and RZG), Stuttgart (HLRS) and Jülich (FZ). In Niedersachsen there are HLRN II (North German Network for High-Performance Computing) and many smaller systems in university computer centers available for large simulation tasks. These geographically dispersed and heterogeneous resources can be used more efficiently for numerous questions. This is referred to as distributed simulation or grid computing. Unfortunately generally the effort needed to create the software and middleware that sits between applications and operating systems is very high and there are enormous constraints on the applications. Therefore it has paid only in individual cases, to distribute simulations on multiple PC clusters so far. However, with the development of more powerful hardware and new calculation methods, it is apparent that the effort to distribute simulation applications to existing PCs and workstations is an area where it is of interest to many users.

In this project area we will examine models and methods with which simulations can be distributed on a grid and the resulting problems of software testing and quality assurance can be solved. "Distributed Simulation" also means the simulation of highly distributed real systems, such as Supply chains, supply chain management, which cannot be detected by a conventional, closed model.



A Cloud-Based Software Infrastructure for Distributed Simulation" – Interface Development

Fabian Glaser, Jens Grabowski

1 Introduction

Due to its scalability and flexibility, Cloud Computing is an interesting option to satisfy the computational demand in simulation sciences. However, using Cloud Computing requires a lot of technical expertise and hence puts a huge overhead on simulators willing to offload their computations to the Cloud. Therefore, technologies are needed that simplify the deployment of simulations applications in cloud environments. Furthermore, to avoid *underprovisioning* of cloud resources which would lead to undesired performance of the executed simulation and *overprovisioning* which would result in unused resources and unnecessary infrastructure costs, we argue that also methods are needed to determine the appropriate scale of the provisioned cloud resources automatically.

Model-driven development (MDD) is a methodology that utilizes formal models of the system to be developed using domain-specific or general-purpose modeling languages. This helps to increase the level of abstraction resulting in better comprehensibility and manageability of the modeled system. Given the formality of the model, it becomes possible to develop *model-transformations* that enable the automated generation of executable code or documentation from the models. Furthermore, model-transformations can be used to refine or tailor a model for a specific use-case, such as a specific representation suitable for a specific group of users.

We argue that MDD is also a suitable methodology to tame the complexity of the deployment of simulation applications for the Cloud and to shield the simulator from complicated cloud internals. As part of the SWZ project "A Cloud-Based Software Infrastructure for Distributed Simulation", we therefore analyzed how simulation applications can be modeled to be automatically deployed in a Cloud. Furthermore, we developed a model-driven approach to automatically determine an appropriate scale of the provisioned cloud resources.

In the following, we will give an overview of our activities in this direction. Parts of this text have been adapted from the corresponding publications made in scope of the project. For more details, we refer the interested reader to the publications as they are listed in Section 9. Before, we present the outcome of our work, we provide a general short background on cloud computing and how our approach fits into the cloud stack in the next section.

2 General terms

Cloud computing grew from a hot research topic and a disruptive resource provisioning paradigm in the industry to a mature technology over the last decade. While several authors defined cloud computing, the definition most cited was published by the U.S. National Institute of Standards and Technology (NIST) [1] and can be seen as the de-facto standard:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Thereby, the NIST defines three service models for cloud computing, which operate on different levels of abstraction. On the highest level of abstraction is Software-as-a-Service (SaaS), where fully fledged applications are delivered to the user e.g., via a web-browser. Below that, Platform-as-a-Service (PaaS) offers programming environments or platforms, such as pre-configured databases, as services. On the lowest level of abstraction

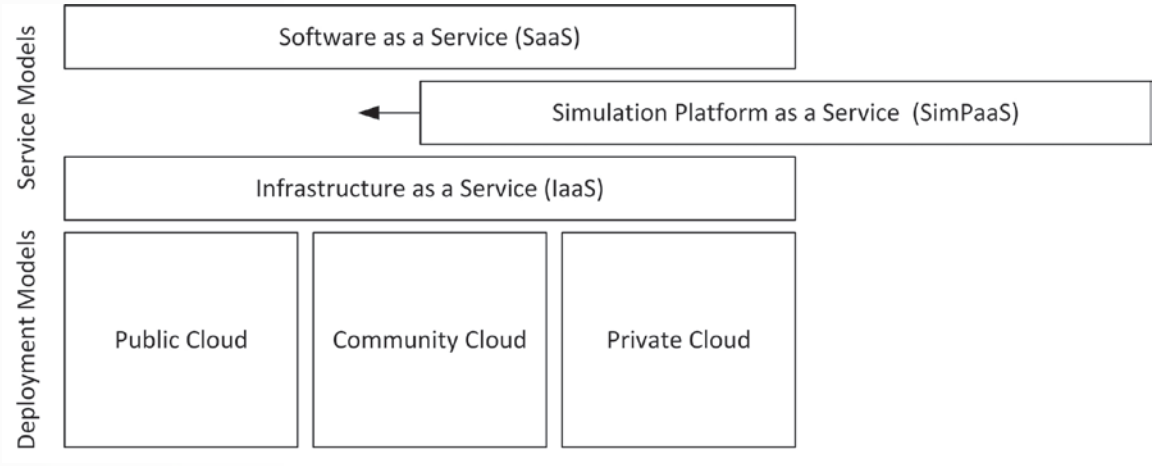


Figure 1: Positioning SimPaaS inside the cloud stack.

the user is able to directly acquire computing resources (e.g., virtual machines, virtual network, and virtual storage) on demand via Infrastructure-as-a-Service (IaaS). All these different service models can be arbitrarily deployed on top of different cloud deployment models. The cloud deployment models are differentiated depending on who owns and uses them. Private clouds are owned and used exclusively by a single company or organization in contrast to Public clouds, which can be utilized by the public (usually for a fee). Community clouds are clouds that are shared among several organizations often built around special requirements. The term Hybrid clouds is often used to denote a mixture of the introduced cloud deployment models.

When considering moving simulation applications to the cloud, several hardware and software dependencies need to be taken into account which vary heavily from domain to domain. Therefore, we decided to build our framework on top of the least restricted service model IaaS with the goal to provide a "Simulation Platform as a Service" (SimPaaS), which enables the simulator to easily move his/her computations to a cloud environment. Figure 1 depicts how our framework fits into the cloud stack.

To determine an appropriate scale for the resources to be deployed, one commonly distinguishes between two types of scaling: *horizontal* and *vertical* scaling. Vertical scaling thereby refers to scaling the size of a single resource, e.g. the amount of RAM in a single virtual machine, whereby horizontal scaling refers to scaling the number of instances of a single resource type, e.g. the number of virtual machines used for computation. Which type of scaling is applicable depends on the software and hardware a simulation depends on. We call this the *computation framework*, which is used to execute a simulation.

When characterizing the computation frameworks for the cloud, one needs to distinguish between two characteristics: *scalability,* which can be defined as "the ability of the system to sustain increasing workloads by making use of additional resources" [2] in contrast to *elasticity,* which can be defined as "the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible." [2] Hence, scalability is a prerequisite for elasticity, but does not take certain temporal aspects into account and also is no measure on how well the provisioned resources actually match the demand.

Having introduced these foundational terms, we describe our notion of the computation framework and the envisioned workflow for deploying simulations in the cloud in more detail in the next Section.

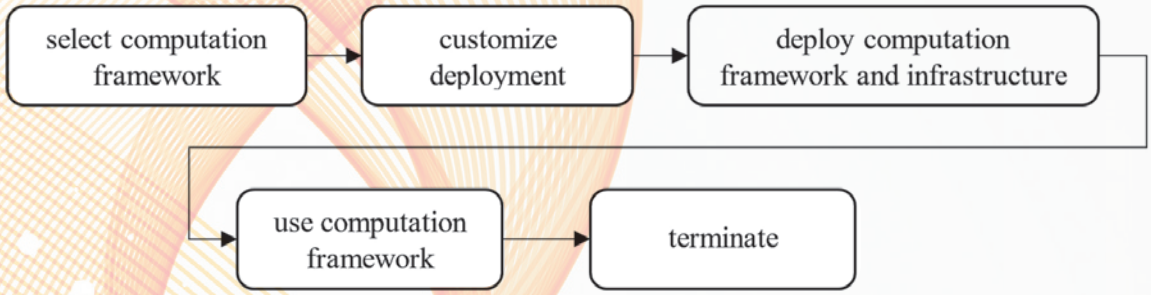


Figure 2. Workflow for using cloud infrastructure.

3 Envisioned Workflow for Deploying Simulation Computation Frameworks

Highly specialized *computation frameworks* for developing and running simulations are common in many scientific domains. These frameworks encode years of experience and scientific knowledge. Simulators working in these specific domains are often very familiar with these frameworks and have the knowledge to modify or extend their code to solve a specific research question. However, they are less familiar with setting up the required cloud infrastructure to execute these computation frameworks efficiently. Obtaining the necessary administrative knowledge to do so places an unnecessary burden on the individual scientist. To add complexity, the computational demand might vary heavily and additional computing resources might be necessary, depending on the research question that is going to be solved.

We experienced this in the early phases of this project, where we were concerned with the evaluation of the suitability of cloud resources and the widely used computational framework Hadoop for Monte-Carlo Simulation and Data Analysis in High Energy Physics. Our case study showed that cloud computing is a viable option and we were able to efficiently utilize a few hundred virtual machines in the Amazon Cloud for computation [a1]. However, we experienced that setting up the infrastructure and scaling it according to the changing computational requirements is a complex problem. As a follow-up of this work, we hence defined a framework that allows to automatically deploy predefined compute clusters for high energy physics in a cloud environment [a2].

This work adapted the workflow depicted in Figure 2, which can be generalized to the usage of different simulation computation frameworks.

In the first step, the simulator selects the desired computation framework from a repository, which stores models of different computation frameworks. In a second step, the deployment of the computation framework can be customized to suit the needs of the simulator. Here the needed resources can be configured, such as the number of cores in the virtual machines or the desired size of the file system. Afterwards, the selected computation framework and the corresponding infrastructure are automatically deployed according to the model in a cloud environment. In the following, the simulator can run his/her simulations on the infrastructure and collect the necessary data and finally terminate the whole infrastructure.

Excursus: Automated Deployment and Distributed Execution of Legacy Applications

Applications that have not been built to be executed in the Cloud and hence neither have built in mechanism for scaling or elasticity can still benefit from using cloud resources. This can be e.g., for doing parameter studies or the processing of multiple input data in parallel. To distribute the load to several machines, existing frameworks for distributed computations can be used. In a master thesis [a6] and a follow-up publication [a4], we evaluated how Cloud resources and Hadoop can be utilized to parallelize these kind of use-cases. The evaluation showed that we can achieve significant speedup, while the overhead for setting up such a system can be reduced to a minimum using automation.

Table 1: Use cases from the project.

Scientific Domain	Collaborators	Computation framework	Scaling Options
Mathematics	Prof. Dr. Schoebel (Uni Göttingen)	LinTim [4], XPress [5]	vertical
Material Science	Prof. Dr. Brenner (TU Clausthal)	OpenFOAM [6], MPI	vertical, horizontal
High Energy Physics	Prof. Dr. Quadt (Uni Göttingen)	ROOT/PROOF [8]	vertical, horizontal

By providing a suitable model for the deployment of such computation framework in a repository, all information on the required infrastructure can be encoded persistently and reused in case the redeployment of the software on another infrastructure is required. Such a model needs to get developed once for each computation framework, which could be done as joint effort between the simulator who is an expert in his domain and an expert who has the administrative knowledge to deploy the computation framework efficiently. The resulting model can be stored in a model repository and used for automated redeployment of the software framework in the future.

In scope of the project, we were working with computation frameworks from three different scientific domains that serve as case studies for the approach. Their characteristics are summarized in Table 1.

We will use the material science use-case as an example in scope of this article.

Modelling Simulation Computation Frameworks

A deployment model of the simulation computation framework needs to include the hardware configuration, the software configuration and a mapping of the application configuration to the hardware components. To be able to fully benefit from the scalability and elasticity offered by cloud systems, it also needs to incorporate information on how to scale the hardware and software components, such that the simulation deployment can be easily adapted to changing performance requirements without redeploying the computation framework from scratch.

After evaluating the use of the general-purpose modeling language UML [9] for modeling simulation computation frameworks for the cloud, we realized that a more specialized modeling language is needed to capture all the details for an automatic deployment. Therefore, we adopted a model based on the Topology and Orchestration Specification for Cloud Applications (TOSCA) [10], which is an approach of the standardization organization OASIS to develop a standardized template language to describe application topologies. A simplified meta-model for TOSCA is shown in Figure 3.

A *ServiceTemplate* captures the structure and the life-cycle operations of the application. It consists of a *TopologyTemplate* and a *Plan*. Plans define how the cloud application is managed and deployed. TopologyTemplates contain *EntityTemplates*, which are either *NodeTemplates* that define e.g., the virtual machines or application components, *RelationshipTemplates* that encode the relationships between the NodeTemplates, e.g., that a certain application component is deployed on a certain virtual machine, or *GroupTemplates* that define groups of NodeTemplates, which e.g. should be scaled together. EntityTemplates have *Properties*, e.g., the IP address of a virtual machine, and a certain type that references an *EntityType*. The *EntityType* defines the allowed Properties through *PropertyDefinitions*, and has *Interfaces*, which define the *Operations* that can be executed on the type, e.g., the termination of a certain application component, or the restart of

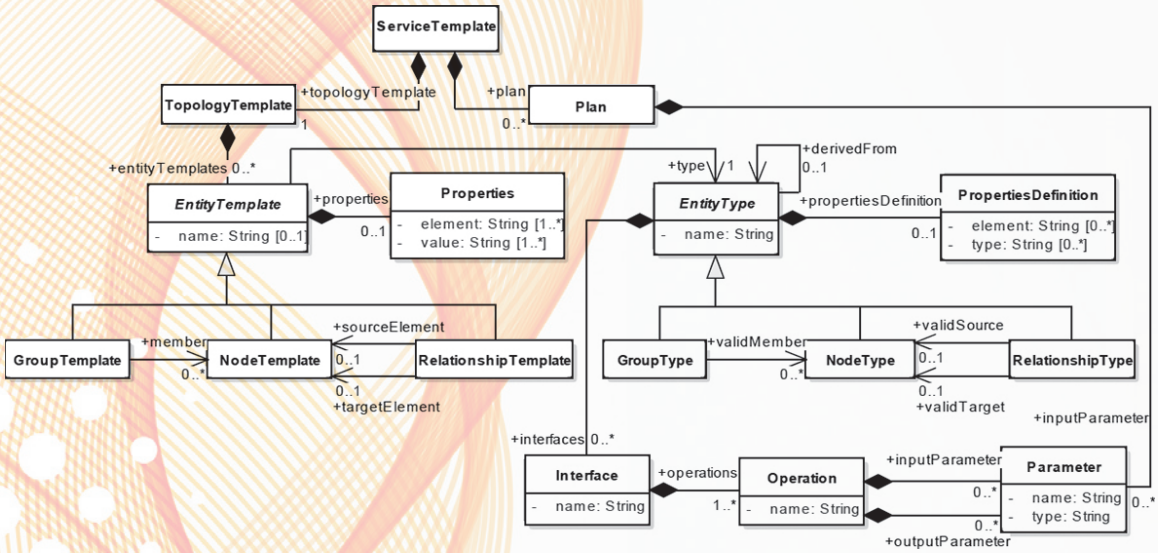


Figure 3: Meta-model of TOSCA (adapted from Bergmayr et al. [13]).

a virtual machine. Operations have *Parameters* that define their input and output. In addition to parameters for operations, TOSCA also allows to define input parameters for Plans. These parameters can be used to parameterize the deployment workflow of the model and can e.g., include the virtual machine type to use or the number of instances of a certain type to launch.

Figure 4 shows a simplified deployment model of the OpenFOAM framework from the material science use-case using TOSCA. The deployment model uses specialized NodeTypes that can be interpreted by the cloud orchestrator Cloudify [11] which is able to start the corresponding instances, e.g., the virtual machines in an orchestrated manner, and the configuration management tool Ansible [12] that is responsible for enforcing a certain software configuration on the virtual machines.

Since there is no standardized graphical syntax for TOSCA available, we use the following notation: NodeTemplates are depicted by boxes with solid lines, RelationshipTemplates are visualized by connections between the boxes, and GroupTemplates are depicted with boxes with dashed lines. For the NodeTemplates and the Groups we additionally list the type and a subset of the Properties. One virtual machine serves as a gateway node. This node gets a public IP address (floating IP) assigned and is reachable from the outside of the cloud. The gateway node is connected to an extra volume which provides the storage for the simulation data. An arbitrary number of virtual machines is deployed to serve as worker nodes in the cluster to do the calculations. The gateway node exports its volume via a Network File System (NFS), which is then mounted and shared by the worker nodes. The software configuration for the gateway and mpiworker nodes is modeled with help of a NodeTemplate of type ansible.nodes.Application. With help of these NodeTemplates the corresponding Ansible roles for the software configuration are associated to the host in which the NodeTemplate is contained. Since the software configuration for the worker nodes is dependent on the software configuration of the gateway, we use an additional depends on relationship between the Ansible nodes. In the abstract deployment model, several parameters can be adjusted to provide an appropriate scale for the required computational power. Hence, the following parameters are kept as input parameters of the model:

P1: The size S of the NFS.

P2: The virtual machine type T of the gateway and worker nodes. The virtual machine type or virtual machine flavor is the common way of IaaS providers to encode the hardware configuration

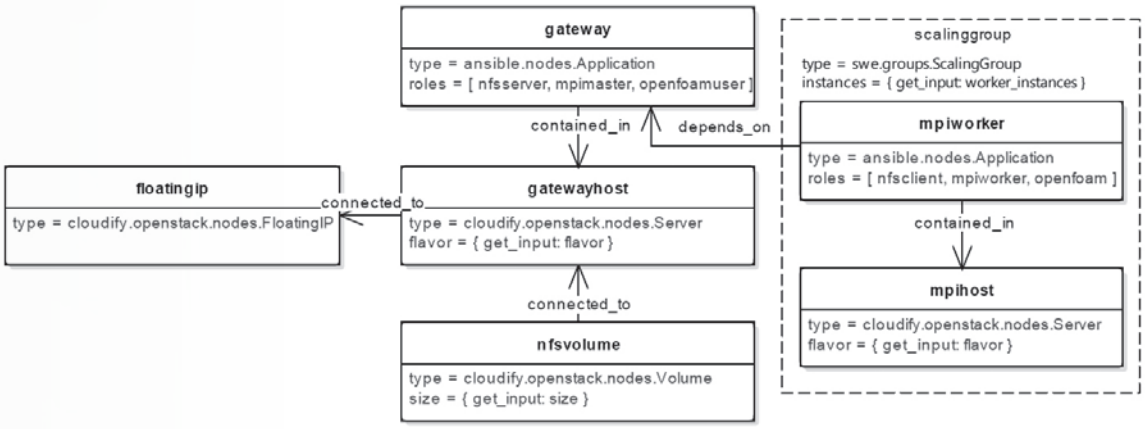


Figure 4: Deployment model of OpenFOAM.

of a virtual machine. This includes RAM size, number of compute cores, and disk space.

P3: The number of worker nodes N. MPI can be used to distribute computation within a single machine across multiple cores, or across multiple machines.

Excursus: Model Transformation from UML to TOSCA

The problem with newly developed modeling languages is that no mature modelling tools are available to assist the user to create model artefacts. This can be solved by defining a model transformation from an existing broadly accepted modelling language that allows to translate models defined in the first language to a representation in the second language. We evaluated this solution as part of a Bachelor thesis [a7] with was supervised in scope of this project. The work showed that deployment models defined in UML can be transformed into TOSCA templates that can be used to automatically deploy applications in cloud environments. A similar approach has been evaluated and implemented by Bergmayr et al. [13] with similar results.

5 Model-Driven Scaling of the Computation Frameworks

Given a model with which the computation framework can be automatically deployed in the cloud, automatic adaption of the deployment to the resource demand becomes possible. The resource demand is implicitly defined by what the simulator wants to simulate with help of the computation framework. We call this the *domain model* of the simulator. For our use case from material science using OpenFOAM, the domain model consists of the following parts:

1: The geometry of the domain on which the mathematical model should be solved and how the mesh on this domain is created.
2: The initial and boundary conditions for the problem for each parameter.
3: The physical properties for the system of partial differential equations to be solved.
4: The control of the simulation, such as the simulation time and the reading and writing of the solution.
5: A domain decomposition that describes how the domain should be decomposed for parallel computation.

If we are able to extract the parameters from the domain model that have an influence on its required scale, we can adapt the scale of its deployment accordingly. The framework for determining an appropriate scale for the cloud resources is depicted in Figure 5, with a possible (technical) instantiation for the material science use-case on the left-hand side.

At its core is the modelling of the deployment of the computation framework. This can be done

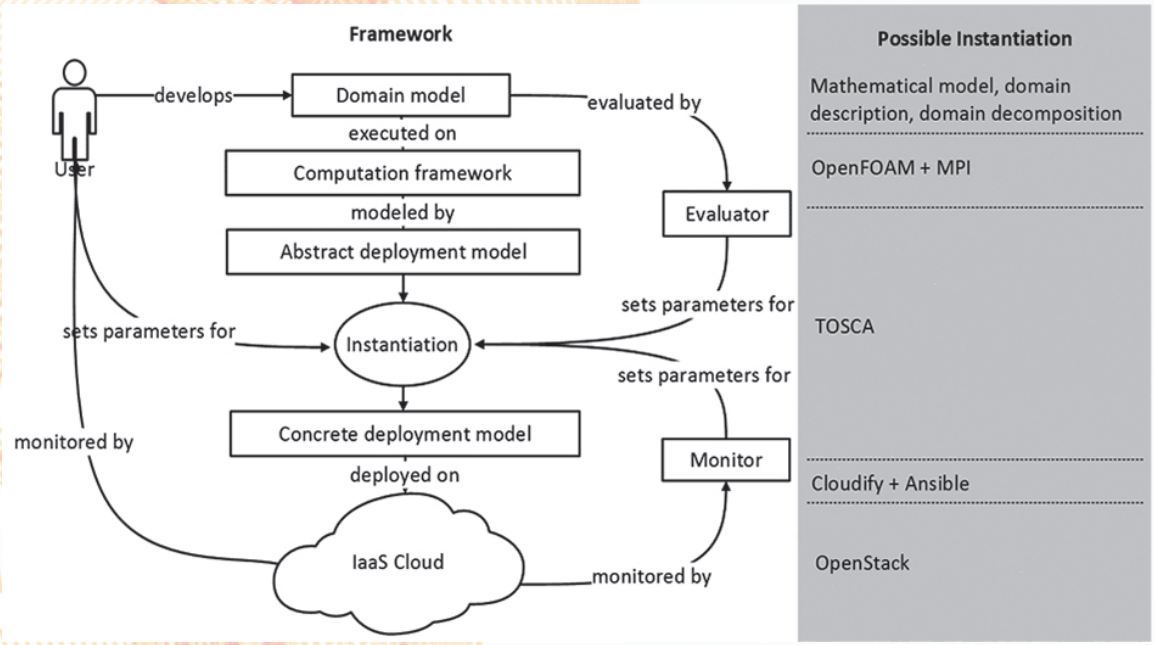


Figure 5: Approach for model-driven scaling of simulation computation frameworks in the cloud.

with help of TOSCA as described in the last section. We hereby distinguish between an *abstract deployment model*, which keeps certain parameters of the deployment that define its scale, unset, and an *concrete deployment model* which has concrete values for these parameters set. To set these parameters, we define three sources: the simulator (user) can set these parameters manually, an *evaluator* can be used that extracts suitable values from the domain model before the computation framework is deployed, and a *monitor* that monitors the performance of the deployed computation framework at runtime and adjusts the parameters of the deployment model, in case the performance goals are not met.

As part of the project, we focused on the evaluator. The evaluator needs to parse the domain model, extract the values for the parameters that have an influence on the resource demand, and map the values to the corresponding parameters of the deployment model. With the goal to define a domain independent evaluator, we defined a domain-specific language that allows us to specify mappings between parameters of the domain model and deployment model parameters. The language enables us to define arithmetic expressions to be able to calculate a value for a deployment model parameter from several domain model parameters. For the material science use-case, the parameters P1-P3 can be set automatically from domain model parameters. A appropriate size for the NFS (P1), for example, can be calculated as

$$size = \frac{total\ simulation\ time}{write\ frequency\ of\ partial\ results} \times size\ of\ partial\ results.$$

The values used on the left-hand side can be automatically determined from the domain model. A detailed description of the language and a small evaluation for the material science use-case have been published in a conference paper [a5]. Our evaluation showed that we are able to determine an appropriate scale of the automatically deployed infrastructure with help of the evaluator and the mapping language. As future work, we are going to extend the framework to incorporate the monitor and a dynamic adaption of the deployment model at runtime.

6 Related Work

Besides TOSCA, other cloud-related standardization attempts exist. In the MDD community, the

Open Cloud Computing Interface (OCCI) [14] receives the most attention. Merle et al. [15] define a metamodel for OCCI to provide a common basis for the generation and conformance testing of OCCI tools. This metamodel is used by Paraiso et al. [16] to model the deployment of applications with help of containers. Several works extend the Unified Modeling Language (UML) to capture cloud-specifics [17][18][19]. Bergmayr et al. [13] show how to convert refined UML models to TOSCA templates. With the Cloud Application Management Framework (CAMF) [20], Loulloudes et al. attempt to build a whole IDE to manage cloud applications with the help of TOSCA. Other approaches developed completely new cloud-specific modelling languages. Brandtzaeg et al. introduce CloudML [21], Silva et al. define the CloudDSL [22], and Hamdaqa et al. present the StratusML [23]. All of these languages are specifically tailored for the modeling of cloud applications. Bunch et al. define Neptune [24], a domain specific language especially to deploy scientific applications in the cloud. While our approach in modeling the computation framework is similar to the works introduced above, the definition of the mapping between the domain of the user and the deployment model is new. Similar to the concept we defined for the dynamic update of the deployment during runtime, Ferry et al. [25] define a Models@Runtime approach for the deployment of cloud applications. We will evaluate the work of Ferry et al. when we extend our implementation to be able to utilize runtime information of the computation framework.

7 Conclusion and Outlook

We argue that the efficient deployment of simulation computation frameworks in a cloud environment should not be burdened on the simulator. Methods need to be provided that simplify this process without loosing the possibility to adapt the provisioned resources according to the computational demand. Our solution tries to automate and scale the deployment of the computation framework by utilizing a model-driven approach as much as possible. We envision a model repository from which the simulator can select the desired computation framework, trigger the automated deployment, work with the provided infrastructure, and finally simply shut down the provisioned resources with a push of a button. We combine MDD and a mapping mechanism to bridge the gap between the domain model to be computed and the required cloud infrastructure to enable the appropriate scaling according to the domain model demands. The introduced mechanism determines an appropriate scale of the infrastructure before it is deployed in the cloud. Our initial experiences show that it is possible to scale the infrastructure appropriately with information extracted from the domain model. However, some information on the runtime behavior of the domain model cannot be predicted by static evaluations. As future work, we will move towards the automated modification and adaptation of our deployment models during runtime.

To further foster an easy-to-use infrastructure for simulations conducted in scope of the SWZ, we are collaborating with the Gesellschaft für wisschenschaftliche Datenverarbeitung Göttingen (GWDG), which serves as the IT-infrastructure provider for the University and the Max-Planck Society in Göttingen. In the follow-up project "Numerically Intensive Simulations on an Integrated Compute Infrastructure" which is a joint work of our group, the GWDG, and the TU Clausthal, we are extending the methodology and technology developed in this project to enable the transparent use of the heterogeneous resource types provided by the GWDG, including the cloud system, the HPC cluster and specialized CPU-enabled resources.

8 References

[1] P. Mell, T. Grance: The NIST definition of cloud computing. Technical report 800-145, National Institute of Standards and Technology (NIST), Gaithersburg, MD, September 2011, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[2] N.R. Herbst.,S. Kounev, R. Reussner. Elasticity in cloud computing: What it is, and what it is not. In Proceedings of the 10th International Conference on Autonomic Computing (ICAC 13) (pp. 23-27), 2013.

[3] Hadoop: http://hadoop.apache.org/, last accessed: 30.11.2016.

[4] LinTim: http://lintim.math.uni-goettingen.de, last accessed: 30.11.2016.

[5] FICO Express Optimization: http://www.fico.

com/en/products/fico-xpress-optimization-suite
[6] OpenCFD: OpenFOAM - The Open Source Computational Fluid Dynamics (CFD) Toolbox. http://www.openfoam.com/, last accessed: 30.11.2016.
[7] Message Passing Interface: http://www.mcs.anl.gov/research/projects/mpi/
[8] Ganis, G., Iwaszkiewicz, J., and Rademakers, F. (2008). Data Analysis with PROOF. In Proceedings of XII International Workshop on Advanced Computing and Analysis Techniques in Physics Research, number PoS(ACAT08)007 in Proceedings of Science (PoS).
[9] UML: http://www.uml.org/, last accessed: 30.11.2016.
[10] OASIS: Topology and Orchestration Specification for Cloud Applications (TOSCA) 1.0, November 2013. http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html.
[11] Cloudify: http://getcloudify.org/, last accessed: 30.11.2016.
[12] Red Hat, Ansible is Simple IT Automation. https://www.ansible.com/, last accessed 30.11.2016.
[13] Bergmayr, A., Breitenbücher, U., Kopp, O., Wimmer, M., Kappel, G., Leymann,F.: From architecture modeling to application provisioning for the cloud by combining UML and TOSCA. In: 6th International Conference on Cloud Computing and Services Science (CLOSER) (2016).
[14] Nyren, R., Edmonds,A., Papaspyrou, A., Metsch, T.: Open Cloud Computing Interface - Core, April 2011. http://ogf.org/documents/GDF.183.pdf.
[15] Merle, P., Barais, O., Parpaillon, J., Plouzeau, N., Tata, S.: A precise metamodel for open cloud computing interface. In: 8th IEEE International Conference on Cloud Computing (CLOUD), pp. 852–859. IEEE (2015).
[16] Paraiso, F., Challita, S., Al-Dhuraibi, Y., Merle, P.: Model-driven management of docker containers. In: 9th IEEE International Conference on Cloud Computing (CLOUD), San Francisco, United States, June 2016. https://hal.inria.fr/hal-01314827.
[17] Bergmayr, A., Troya, J., Neubauer, P.,Wimmer, M., Kappel, G.: UML-based cloud application modeling with libraries, profiles, and templates. In: 3rd International Workshop on Model-Driven Engineering on and for the Cloud (CloudMDE), pp. 56–65 (2014).
[18] Kamali, A., Mohammadi, S., Barforoush, A.A.: UCC: UML profile to cloud. computing modeling: using stereotypes and tag values. In: 7th International Symposium on Telecommunications (IST), pp. 689–694. IEEE (2014).
[19] Guillén, J., Miranda, J., Murillo, J.M., Canal, C.: A UML profile for modeling multicloud applications. In: Lau, K.-K., Lamersdorf, W., Pimentel, E. (eds.) ESOCC 2013. LNCS, vol. 8135, pp. 180–187. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40651-5_15.
[20] Loulloudes, N., Sofokleous, C., Trihinas, D., Dikaiakos, M.D., Pallis, G.: Enabling interoperable cloud application management through an open source ecosystem. IEEE Internet Comput. 19(3), 54–59 (2015).
[21] Brandtzæg, E., Mosser, S., Mohagheghi, P.: Towards CloudML, a model-based approach to provision resources in the clouds. In: 8th European Conference on Modelling Foundations and Applications (ECMFA), pp. 18–27 (2012).
[22] Silva, G.C., Rose, L.M., Calinescu, R.: Cloud DSL: a language for supporting cloud portability by describing cloud entities. In: CloudMDE 2014, p. 36 (2014).
[23] Hamdaqa, M., Tahvildari, L.: Stratus ML: a layered cloud modeling framework. In: 2015 IEEE International Conference on Cloud Engineering (IC2E), pp. 96–105, March 2015.
[24] Bunch, C., Drawert, B., Chohan, N., Krintz, C., Petzold, L., Shams, K.: Language and runtime support for automatic configuration and deployment of scientific computing software over cloud fabrics. J. Grid Comput. 10(1), 23–46 (2012).
[25] Ferry, N., Brataas, G., Rossini, A., Chauvel, F., Solberg, A.: Towards bridging the gap between scalability and elasticity. In: 4th International Conference on Cloud Computing and Services Science (CLOSER), pp. 746–751 (2014).

9 Publications

Parts of the work described above, have been peer-reviewed and published in the proceedings of

international conferences:

[a1] F. Glaser, H. Neukirchen, T. Rings, J. Grabowski, Using MapReduce for High Energy Physics Data Analysis , Proceedings of the 2013 International Symposium on MapReduce and Big Data Infrastructure (MR.BDI 2013), 03-05 December 2013, Sydney, Australia 2013, DOI: http://dx.doi.org/10.1109/CSE.2013.189 , 2013.

[a2] F. Glaser, J. N. Serrano, J. Grabowski, A. Quadt, ATLAS user analysis on private cloud resources at GoeGrid , Proceedings of the 21st Conference on Computing in High Energy Physics and Nuclear Physics (CHEP), 13-17 April, Okinawa, Japan, available online: http://iopscience.iop.org/article/10.1088/1742-6596/664/2/022020 , 2015.

[a3] F. Glaser, Towards Domain-Model Optimized Deployment and Execution of Scientific Applications in Cloud Environments, Proceedings of the Doctoral Symposium at the 5th Conference on Cloud Computing and Services Sciences (DCCLOSER 2015), Lisbon, Portugal, 2015. (Best PhD project award).

[a4] M. Göttsche, F. Glaser, S. Herbold, J. Grabowski, Automated Deployment and Parallel Execution of Legacy Applications in Cloud Environments, The 8th IEEE International Conference on Service Oriented Computing & Applications (SOCA 2015), 2015.

[a5] F. Glaser, Domain Model Optimized Deployment and Execution of Cloud Applications with TOSCA, Proceedings of the 9th System Analysis and Modelling Conference (SAM 2016), Springer, 2016.

Additionally, the following student theses were supervised in scope of the work:

[a6] M. Göttsche, Automated Deployment and Distributed Execution of Scientific Software in the Cloud using DevOps and Hadoop MapReduce, Masterarbeit im Studiengang Angewandte Informatik am Institut für Informatik, ZAI-MSC-2015-02, ISSN 1612-6793, Zentrum für Informatik, Georg-August-Universität Göttingen, 2015.

[a7] J. M. Erbel, Transformation of UML Deployment Models into executable Cloud Orchestration Templates, Bachelorarbeit im Studiengang Angewandte Informatik am Institut für Informatik, ZAI-BSC-2015-08, ISSN 1612-6793, Zentrum für Informatik, Georg-August-Universität Göttingen, 2015.

Project data

The project was funded from SWZ with 0.5 TV-L E13 staff positions from July 2013 to June 2016 at the site Göttingen. Involved scientists are:

Prof. Dr. Jens Grabowski
Research Group for Software Engineering for Distributed Systems
Institute of Computer Science
University of Göttingen

Xiaowei Wang, M.Sc.
Research Group for Software Engineering for Distributed Systems
Institute of Computer Science
University of Göttingen

Fabian Glaser, M.Sc.
Research Group for Software Engineering for Distributed Systems
Institute of Computer Science
University of Göttingen

Simulation-Based Quality Assurance for Software Systems

Jens Grabowski, Verena Herbold, Daniel Honsel, Stephan Waack, Marlon Welter

1 Introduction

Software Quality Assurance has risen in popularity over the last decades. Software has become an integral part of almost everybody's life, in economics, in science and in everyday life. The complexity of software has grown and also the certainty that programs have been extended over the years. It is therefore only natural to explore ways to better monitor software quality and predict possible problems that might arise in the future. The sooner a bug or unsafe part of the software code can be detected, the sooner a project manager is able to react to it and implement countermeasures. Even if there is no immediate problem with the software, the detection of unclean code can prevent problems in the future.

Once a problem has been detected the next step is to find effective countermeasures (e.g. hire more developers). Due to the nature of our simulation-based approach, we can evaluate different scenarios that a project manager can execute and predict their likely result in terms of software quality. Although there are many unknown variables that cannot be covered by a simulation tool (e.g. new external requirements), the results still have a value as a general guideline towards the question which measures could be the most effective ones.

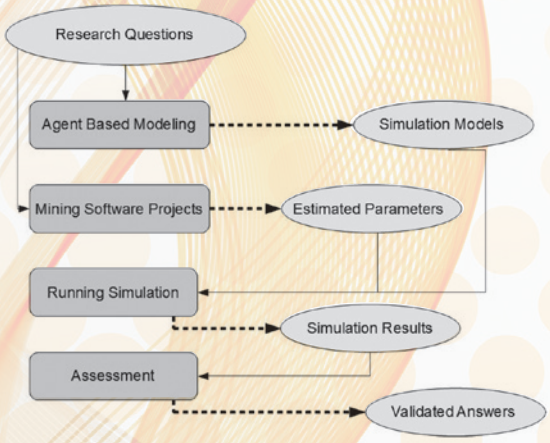


Figure 1: Process of building assessed simulation models.

2 Approach

In this section we describe our approach for evaluating the quality of simulated software projects. Therefore, we propose an interaction of three different research areas. The overall process is depicted in Figure 1.

Starting with a concrete research question about software evolution in mind, e.g., how are different kinds of changes distributed over time, a tailored model is built (Agent-Based Modeling). With the model in mind, we select and mine open source software projects which helps us estimate appropriate model parameters describing the software evolution process under investigation (Mining Software Repositories). Once the model is adjusted with the estimated parameters, the simulation can be run (Running Simulation). This step also belongs to the agent-based modeling, since it gives feedback about the model itself, e.g., if it needs to be adapted, if the so gained model does not fit the real world. Finally, the simulated results get assessed in terms of their quality (Assessment). Concretely, we use mining of software repositories to parameterize the simulation model according to one particular open source project. By this, we observe the size and effort, i.e., number of different developer types, of other open source projects with the aim of simulating them under similar settings. Then, an agent-based simulation model is used to describe the behavior of the different developer types as well as the dependencies between software entities and developers. Finally, to evaluate the simulation, Conditional Random Fields (CRFs) are applied on the simulated software dependency graphs.

In the following, we describe the steps Mining Software Repositories, Agent-Based Modeling, and Assessment in more detail.

3 Mining Software Repositories

The main driver of the simulation are the developers, which trigger the simulation by their changes on the software entities. Thus, the description of the developers as well as their contribution behavior is of great importance. In our preliminary study [1], we focused on the growth of software systems based on the commit distribution. Since the focus of this report lies in the methodology of the approach and the interplay of the different disciplines, we concentrate on mining aspects which work closely together with the simulation and assessment.

In a software project, the work of an individual often differs significantly from the average behavior. It is influenced by their experience, position, involvement in the project, and personal work style. To take this into account, we derive developer types by mining. Understanding developer roles and related contribution is important for the software development process, e.g., making trend predictions [2]. In this work, the roles are classified based on heuristics depending on the commit frequency and bug fixing activity of individuals.

Since the simulation makes statements about the overall quality of a software project, the introduction of bugs is an important factor. From software repositories, one can retrieve the information, at which point in time a bug occurs or gets fixed, but often the information about the change introducing the bug is missing [3]. The current model reflects bug report and closing frequencies based on mining issue tracking systems (ITS).

To model dependencies between software entities, e.g., classes, files, and modules, we make use of change coupling networks, i.e., dependencies based on common changes. For this, we create, for example at the file level, a node for every entity, i.e., file. If two files are changed together, an edge is created and its weight is increased with every further change. To reduce noise, we omit large commits. To trace the evolution of these networks, we measure the average degree, modularity, diameter, and the number of clusters over time. The degree is the number of dependencies of a node, the modularity represents an indicator of how good the graph can be divided into clusters, and the diameter describes the distance, i.e., the maximum shortest path in the whole network. The clusters are assigned by the node's modularity class [4]. This way, the clusters represent logical dependencies of software entities, e.g., user interface elements or tests.

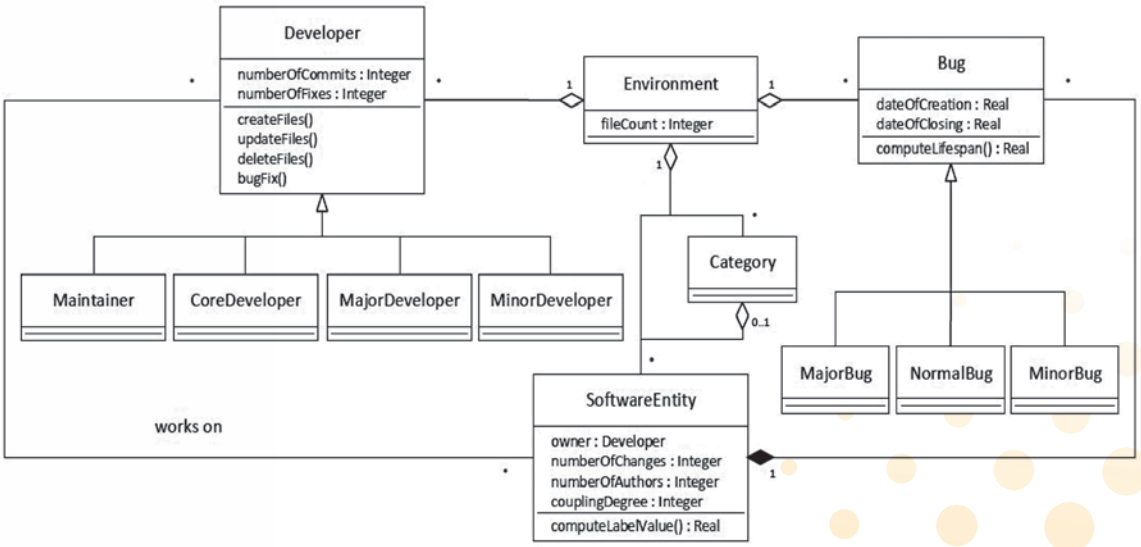


Figure 2: Agent-Based Simulation Model for Software Evolution.

4 Agent-Based Modeling

Since we are on a more abstract level in the context of simulation, we speak in general of software entities instead of files. For developing our model, we use Repast Simphony [5], an open source agent-based modeling and simulation platform. We develop agent-based simulation models corresponding to the specific problem formulated in research questions. An agent is an individual that acts autonomously in an environment. Thereby, it has dynamic interactions with other agents and perceives the environment. In the domain of software evolution the environment is a software project and individual agents are developers, entities, and bugs. Agents can be considered as active or passive. Active agents can change the state of passive agents.

Our simulation model depicted in Figure 2 contains all agents which are divided into active and passive ones. The only active agents are the different developers who work on the software entities, i.e., software entities and bugs are passive agents. The environment knows all instances and is responsible for the creation of a configured number of developers at start-up and instantiating bugs at scheduled points in time. This is based on the previously mined bug occurrence rates from the ITS.

To represent dependencies between the entities we use networks. The most important networks are the *DeveloperEntityNetwork*, *BugEntityNetwork*, and the *ChangeCouplingNetwork*.

- *DeveloperEntityNetwork*. This network represents the dependencies between entities and developers. If a developer creates an entity, an edge between the developer and the new entity will be created. If a developer changes an existing entity again, the weight of the edge is increased, otherwise a new edge will be created. Hence, this network provides the owner, the number of authors, and the number of changes of one entity. The owner is the developer who changed an entity most.
- *BugEntityNetwork*. After the environment has created a bug, an edge between the new bug and the almost randomly selected entity is added to this network. For this selection, the degree, the number of changes, and the cluster of the software entity are also be taken into consideration. It is more probable that a frequently changed entity with many dependencies gets a bug than an isolated and rare changed one. The edge contains information whether a bug is fixed or not.
- *ChangeCouplingNetwork*. This network represents dependencies between software entities that are changed together several times. Since one can consider it as the semantic description of the simulated software, this network serves as input for the evaluation.

The core of the model is the commit behavior of the developers. Commits are responsible for entity changes through the following actions: create, delete, and update. These actions make the software evolve over time. Furthermore, commits are responsible for fixing bugs. To model the commit behavior of developers, we assume that in each simulation round a developer can perform a commit according to the type and the corresponding commit probability. If the developer commits, then the number of entities to be changed has to be determined. For that, we assume that the number of entity changes follows a geometric distribution [1]. The probability of creating and deleting entities decreases with the increasing system growth and the probabilities are restricted through the project size and adjusted for each type of developer such that the average update behavior is restrained like the mined data. Through the mining, we identified four different types of developers:

- *Core Developer*. The founder and the project leader. Most entities are known by him and he performs most commits.
- *Maintainer*. A person who does primarily maintenance work. He fixes a large number of bugs in the project.
- *Major Developer*. A developer who knows only parts of the project and works mostly on entities known by him.
- *Minor Developer*. A developer with less than 100 commits and only few bugfixes, e.g., a specialist that only implements one specific task or feature.

Modeling the entity selection is a difficult task, because we have no information about the

intention of the developer. For the entities to be updated, we select the first entity randomly, with a slightly higher probability for entities that the developer owns or has worked on before. The selection of further entities is additionally based on information about the first one, e.g., the owner of the entity, the category that entity is assigned to, and other former touched files in the category. According to Hattori and Lanza [6], corrective activities, such as bug fixes, generate many tiny commits including one to five entities and about 80 % of all commits are tiny ones. Thus, instead of performing a commit as described above, a developer can have the intention to fix a bug. For this, one entity with at least one bug and additionally up to five connected entities is selected randomly.

If a developer has the intention to fix a bug or another commit contains a bug, then the bug can get fixed with a certain probability. This probability depends on the ownership, the coupling degree of the entity, and the developer type. The highest probability to fix the bug has a core developer that is in addition the owner of the entity.

To assess the quality of the simulated software, a label representing the quality (acceptable/problematic) is assigned to each entity. The label is computed as follows: For each bug type (major, normal, and minor) values are set indicating how much a certain bug type should decrease the label. The different bug factors are set as follows: 0.825 for major bugs, 0.9 for normal bugs, and 0.98 for minor bugs. For each bug of a type assigned to an entity the according factor is multiplied with the current value. This means that an entity without a bug has the label value of 1 and every bug decreases the value. Afterwards each entity is preliminarily labeled as acceptable (label higher than 0.8) or problematic (else). The simulated change coupling graph with labeled entities serves as input for further assessment.

5 Assessment

The assessment of the current state of a project is done via CRFs. The main idea here is that we want to account for dependencies between different entities of the software (e.g., if entity *A* has a bad quality and *B* depends on *A*, then entity *B*'s quality is also affected). The dependencies among entities are modeled in the change coupling graph and the local quality of entities is provided as a preliminary label by the simulation.

Given the graph and preliminary labels for each entity, we use a method inspired by the Ising model of statistical mechanics to create a new final labeling of the graph, that includes the neighborhood information. Analogous to the Ising model, we have conformity weights, that reward if entities are labeled the same as their preliminary labeling, and a coupling parameter, that rewards if neighboring entities are labeled in the same way.

In the reference project we adjusted these weights to achieve the following goals.

- Dependencies between software entities influence the quality of each other, particularly in a way that a problematic software entity negatively influences those entities that depend on it.
- Providing a broadly homogeneous labeling of big parts of the software, such that an observer has a clear overview of the quality state of the software as a whole.

The first point was achieved by first choosing the coupling parameter in a way that labeling neighboring sites in the same way provides a higher score than labeling them differently and, secondly, by choosing a higher weight for the conformity weight of the label problematic than for the label acceptable since this results in a higher score for negative influences, i.e., entity *a* negatively influences the quality of entity *b*.

The second point simply is a result of the coupling weight alone, since this leads to a homogenization of highly connected parts of the graph. To provide a clearer overview of the quality state of the software, we further introduce another level of abstraction of the software graph. Derived from the software graph, we create the categories graph where each node represents one category of entities as provided by the simulation. This category graph is now visualized in our tool and each category is colored in a tone between green and red according to the ratio of acceptable to problematic entity labellings resulting from the CRF evaluation.

6 Case Studies and Results

Study 1: Evolution of Change Coupling Networks

In one study [7], we focused on software dependency analysis through agent-based simulation of software development. We extended our initial framework described in [1], where we already simulated the growth and bugs of software systems. For getting parameters that encode the reality, we mine the evolution of change coupling of two open source software projects: the cd and dvd burning tool K3b for the estimation of parameters and the java logging framework Log4j for validation. Hence, the simulation model reflects the behavioral rules of software dependencies detected by the mining process. Since the overall aim is to describe the evolution of software dependencies, we need to perform analysis on different levels. The work examines two research questions:

- RQ1: How do software dependency networks based on change coupling evolve?, and,
- RQ2: Can change coupling be simulated by an agent based simulation?

For answering these questions, we investigated the project in terms of file dependency networks. For the calculation of the dependencies, we consider change coupling, i.e., two files (nodes) are dependent, if they have been changed together frequently. In this case, the two nodes are linked with an edge.

Results

For modeling the dependencies between files we examined the evolution of the yearly degree for each file. To put this into a manageable format we calculated the average for all files.

The results are visualized in Figure 3. The colors are given according to the community structure derived by the modularity algorithm. These communities reveal logically dependent units, e.g., tests. The average degree d, i.e., the average number of dependencies, rises fast and stagnates after some years. In Log4j the values are more unsteady in the beginning. Although K3b is comprised of less files, the average degree is higher than for Log4j. The cause for this is that the corresponding networks contain more weakly connected files, which lower the average. The nodes represent the files existing in each year and the edges represent the dependencies. The figure shows that the graph evolves in a dense way and develops strong file dependencies. In contrast the graph of Log4j, although developing a strong core, evolves in a more fragmented way. There are also some small individual clus-

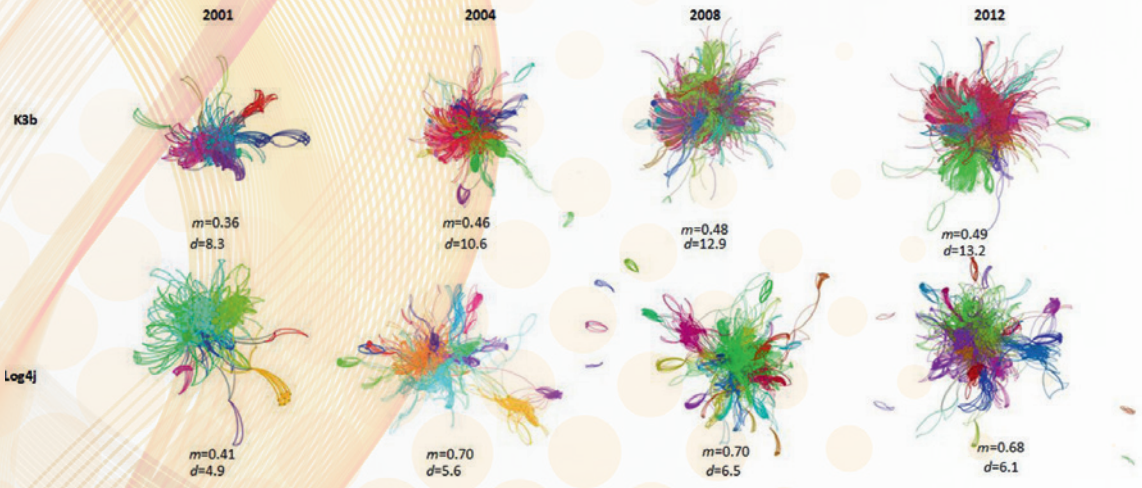


Figure 3: File Dependency Networks of K3b and Log4j.

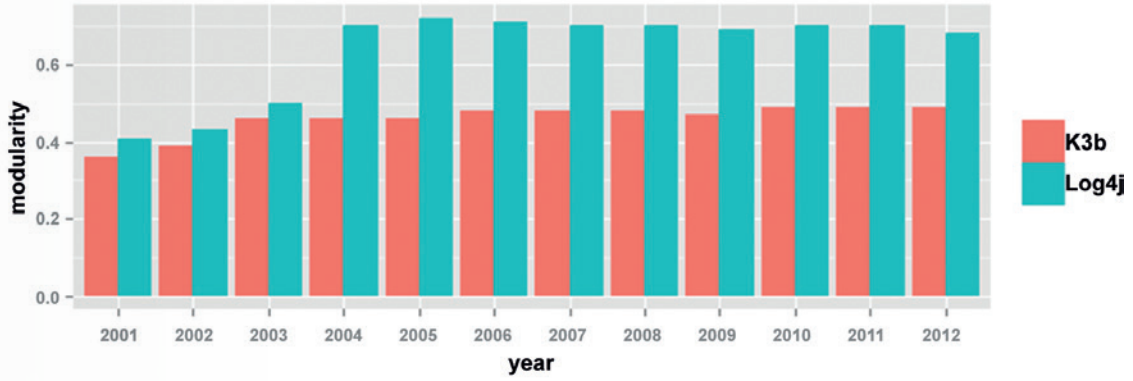


Figure 4: Network modularity for K3b and Log4j.

ters around. This observation is supported by the modularity measure *m*, which is higher for Log4j and also determines the clusters. The network modularity is an indicator of how good the network can be divided into clusters. The modularity values compared for each year are depicted in Figure 4.

We observed that the average coupling degree over the time resembles a square function. We verified this by fitting a regression model to the empirical function and retrieved the best fit for a second order model with an adjusted R-squared value of 0.97. These observations state the answer to RQ1. For the comparison of empirical and simulated behavior, we consider the results of the simulation. In Figure 5 the average coupling degree of the project under simulation is shown, which relies on our *ChangeCouplingNetwork*. The average coupling degree of the simulation follows the same trend as the observed empirically from K3b. Thus, the general evolution of file dependencies is the same (R Q2).

Furthermore, we investigated, whether we can reproduce the system growth and average coupling degree in the evolution of Log4j. For this purpose, we tried out our simulation with an experimental parameter setting which suits Log4j. We increased the maximum number of files, adapted the number of simulation rounds, and changed the number of the distinct developer types according to the mined values. For the latter, we have one core developer, five major developers, and fourteen minor developers in Log4j. K3b has one core developer, one maintainer, 17 major developers and 106 minor developers.

With this changed simulation parameters we get simulation results close to the empirical behavior. We reach values similar to that in Figure 3 with the simulation. The result is shown in Figure 5 with the simulated curve in blue. The unsteadiness in the beginning observable in the real project data, is also visible in the simulation.

As a conclusion, we can say that a project similar in size and duration can be reproduced with simulation by changing few parameters.

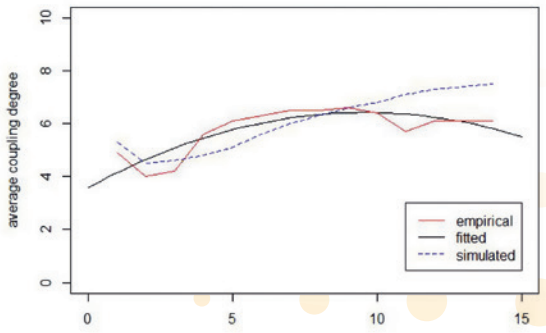


Figure 5: Empirical, Fitted, and Simulated Average Coupling Degree of Log4j.

Table 1: Overview of projects.For developers: (core|maintainer|major|minor). Duration: in years.

Project	#Commits	#Files	#Developers	Duration
K3b	6142	1046	(1|1|6|116)	12
Log4j	8082	620	(1|1|6|13)	13
Kate	14282	681	(1|1|10|328)	11

Study 2: Monitoring Software Quality

The second study presented in [8] is designed according to two main objectives. First, we are interested in a realistic performance of the current simulation model. Therefore, we conduct experiments which are able to show if the simulation behaves in the expected way, e.g., to show the impact of the loss of a core developer. Second, based on the experiments and our experience on the simulation of software evolution we declare requirements for future simulation models.

For this, we analyzed three open source projects: K3b [14], Log4j [18], and Kate [19] which are similar in the duration and size, but differ in the effort spent by the developers. Table 1 gives an overview of the project properties.

For each, we calculated the metrics we need to estimate parameters to employ the current simulation model. For this experimental setup, we set some parameters to be fixed and some to be changeable according to the project properties. Fixed parameters originate from our preliminary studies retrieved from K3b. Changeable parameters are, e.g., the number of developers of a certain type, the project duration, the size of the system, and the initial number of clusters, i.e., interrelated parts of the software. Overall, the study aims to answer the following research question:

RQ3: Can we simulate effects like the loss of a core developer realistically?

To answer this question, we employ the simulation model (Figure 2) with parameters derived from the three open source projects, performed the described experiments on each model and automatically assess the results with our assessment tool. In the following we describe results and requirements to the simulation as implication of the experiments.

Results

From the mining perspective important parameters for simulating the loss of a core developer are first of all the team constellation and, for monitoring the impact on quality, the bug introducing rates and the dependency graphs including the bug information.

Changes in the team constellation itself, especially in the leadership, also have an impact on the overall software quality. In Figure 6, the commit frequencies of developer types in K3b is depicted. There we observed that a very active core developer became inactive and the maintainer took over the project lead. A different development pattern is observable in Figure 7 where the commits of developer types in Kate are visible. There the core developer stays active over the whole time with peaks of more work done and the same holds true for the other types. For modeling these effects, we build heuristics from the mined data to manually categorize the types and build averages over their commit and bug fix behavior to describe the general behavior of an agent of the distinct types. The heuristic analysis yields the following categorization: core developers perform more than 20 % of all commits; maintainers perform more than 10 % of all commits and more than 25 % of their commits are bug fixes; major developers own more than 2 % of all commits; minor developers perform less commits. Using these heuristics we derived the number of developers of each type listed in Table 1.

Bugs are also modeled using the average over all reported and closed bugs originating from the ITS.

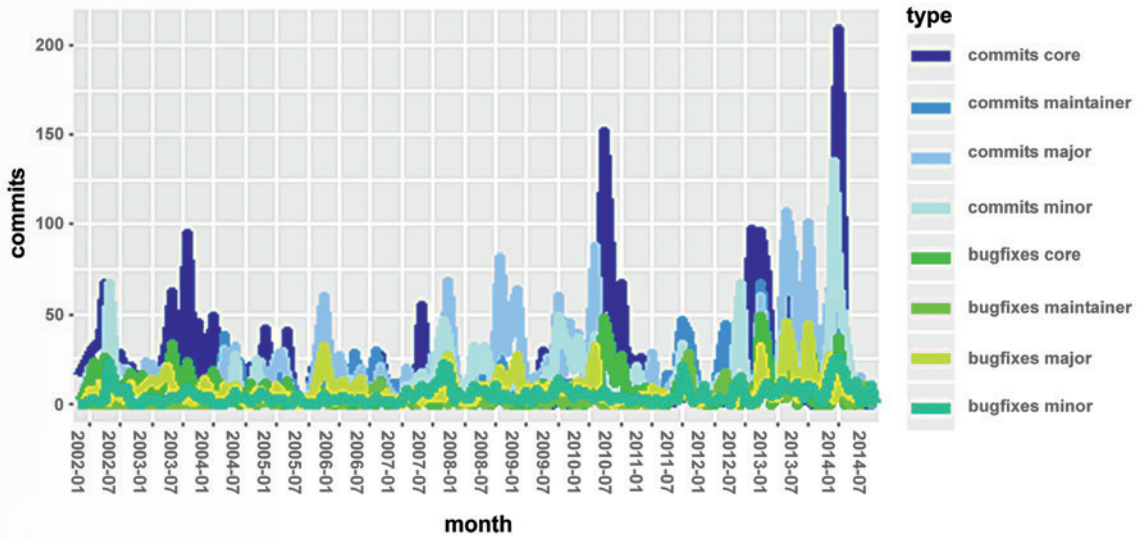


Figure 6: Monthly commits by developer type of K3b.

For this, we consider minor, normal, and major bugs. Critical bugs are also considered as major. Overall, we derived a number of 0.87 reported bugs and 0.81 closed bugs in average per day which presents one round in the simulation.
We investigated the evolution of software dependency networks in our preliminary study. Although we were able to simulate the average node degree, i.e., the average number of co-changes between files, different structural properties only matched in terms of the general trend. Therefore, we added the number of clusters derived by the modularity class to the parameter set. Since we are most interested in areas of heavier dependencies, we omit very small clusters covering less than 5 % of the graph. This way, we observed the evolution of clusters shown in Figure 8.

There we observed an unsteady rise in the beginning for all projects that stabilizes in the case of K3b. For Kate, the growth is similar, but at the end, some clusters merge, so that there are a few

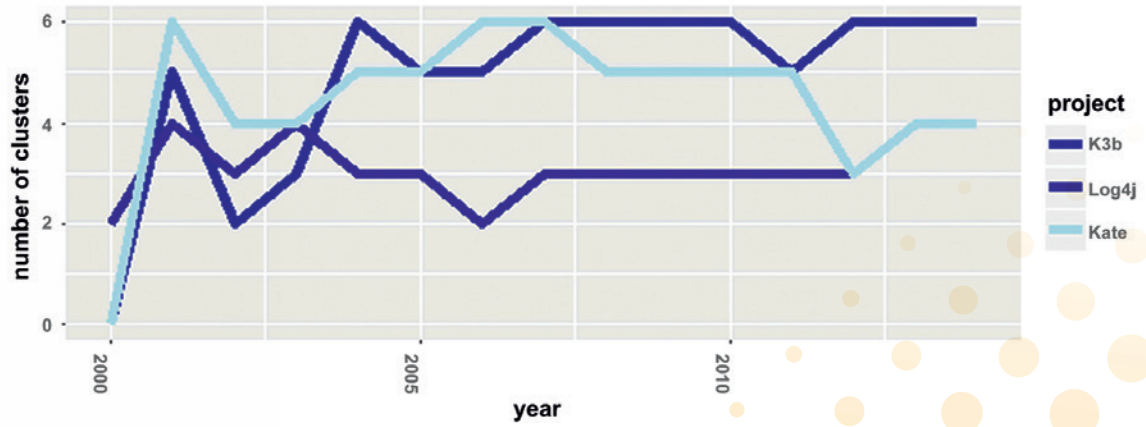


Figure 7: Monthly commits by developer type of Kate.

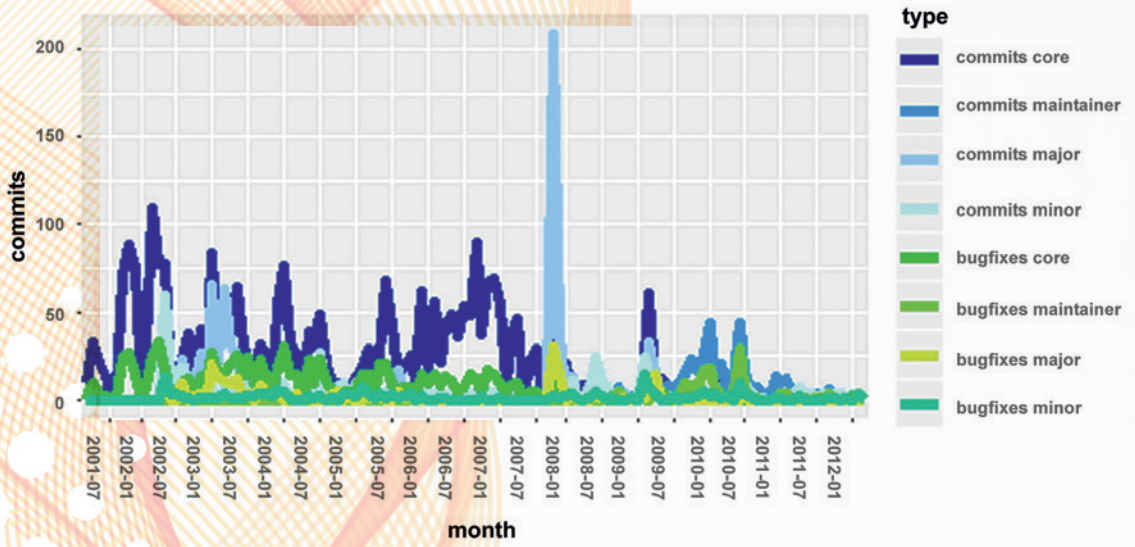


Figure 8: Number of clusters over the time.

larger clusters. Log4j has a very high modularity and its dependency graph branches into many tiny clusters with the effect that only a few bigger clusters exist over the whole time.

With the gained insights, the work can be pursued on the simulation model. For this, we first simulated the reference project K3b and afterwards, we changed particular parameters according to the mined data to simulate the other projects. The parameters we changed are the project size and duration as well as number of developers per type. To answer RQ3, we simulated two runs. The first one without changes to the effort and the second one with reduced effort after one half of the project's duration. To achieve this, we assume that the core developer leaves the project. With reduced effort 12 % to 20 % fewer bugs are fixed which matches our intuition.

For the assessment of the different simulation runs we had to determine the configuration parameters for the CRF (coupling and conformity weights). We set the parameters according to some prior work done on K3b where we found a set of parameters that result in an overall acceptable state when simulating K3b with the parameters mined from the real data, but would result in several clusters being evaluated as critical when omitting developers or increasing bug introduction rates. So we used this set of parameters for the assessment tool to see if it could be used for other similar projects as well.

For all three simulated projects the evaluation of the quality of the software was in an overall acceptable state when running with the mined parameters, but the removal of a core developer resulted in significantly worse evaluations of the quality state of the software as can exemplary be seen in Figure 9. Thus, we can answer RQ3 as follows: Effects, like the loss of a core developer can be simulated using our simulation framework. The impact matches our intuition.

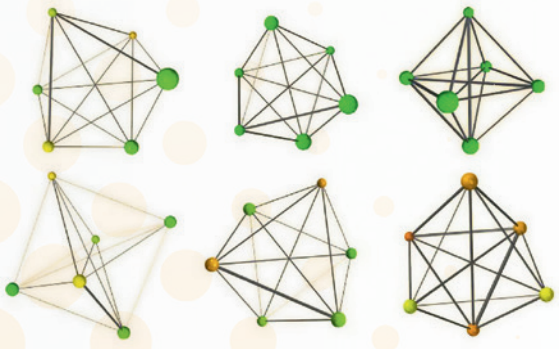


Figure 9: Screenshots from the assessment tool of 3 points in time of two simulation runs of K3b.

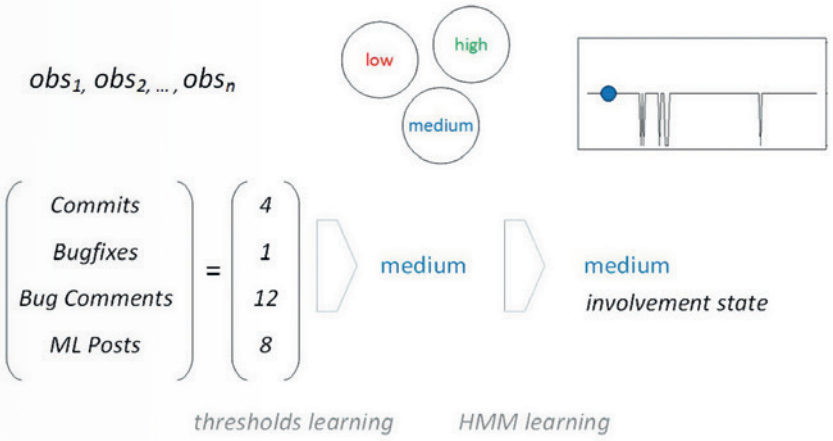


Figure 10: Learning of Developer's Involvement State Sequence.

The 3 images in the top row result from a run with mined parameters, whereas the bottom row results from a run where a core developer was removed. Nodes represent clusters, colors represent relative ratio of acceptable labels to critical labels (green/red), the size of the edges represents the number of dependencies between files of clusters.

Study 3: Dynamic Model of Developer Behavior

Our third study [9] deals with a dynamic description of developers involvement and workload, since our preliminary studies posed a problem with the linear effort spent by developers in the simulation. Developers' actions are not solely visible in their commit behavior. Besides the Version Control System (VCS), a huge amount of data is available for analyses: ITSs, Mailing Lists (ML), forums, and also the activity of developers on social networks like Twitter can be traced. However, given the whole history of a project, it is hard to derive a complete picture of the behavior of developers. Often, the different data sources have no links among each other, e.g., names and email addresses for the same developer can differ across platforms. This is often solved by tools developed for that purpose or manual mapping. To describe the evolution of development activity and involvement of developers in a dynamic way, i.e, the extent of involvement in a project varies over time, we make use of the data describing the observable behavior of developers. We propose to employ Hidden Markov Models (HMMs) to gain insights about the underlying behavior to answer the following question:

RQ4: Can we model developers' contribution behavior accurately using HMMs?

We focus on analyzing the following important involvement factors for each developer over the time: the monthly number of commits and bug fixes, the amount of mailing list posts, and the number of comments in the ITS as observation sequence. This way, we get a broad picture of the evolution of open source software (OSS) developers.

In Figure 10, the broad learning process is illustrated. We start with a sequence of monthly activity vectors as observations. We use a threshold learner and KNN (K-Nearest Neighbor) to classify the observations into low, medium, and high for each metric and with a majority vote for each observation. With the Baum-Welch and the Viterbi algorithm we calculate the transitions between the involvement states (e.g., low involvement to medium involvement) and the emissions for all states (i.e., the workloads).

Figure 11a) mr for Individual Models.

We build a HMM for each developer of a project, as well as one general model for all developers. This way, we can describe the activity and workload of developers dynamically, which we will use to extend our simulation model to allow for changes in the project team during the simulation.

Results

We evaluated our approach in a case study including six OSS projects. The goal of our case study is to examine our main hypothesis, i.e., that we can describe developer contribution dynamics with HMMs and its usefulness for the prediction of developer contribution.

We trained a HMM for each developer, who performs a minimum number of actions in the project. Since we assume to derive similar models for same developer roles, we compared the retrieved models role-wise and also project-wise. Finally, we are interested to know to which extent the HMMs derived by a project are applicable for other software projects.

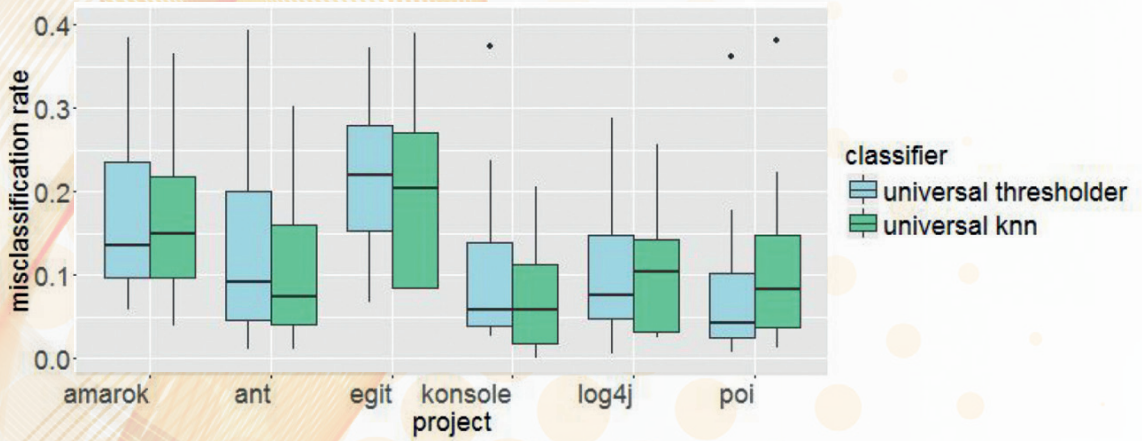


Figure 11b) mr for General Models.

To assess the goodness of fit of the derived models, we measure the misclassification rate *mr* which counts the number of observations where the predicted state differs from the classification done by the threshold learner and K-Nearest Neighbor (KNN). The mean misclassification rate of all fitted HMMs for developers' contribution behavior is $mr = 0.098$ with threshold classification and $mr = 0.109$ with KNN, i.e., the prediction performs well in general, which presents the answer to RQ4. Project-wise results are depicted in Figure 11a. The average misclassification rate using general models comes up to 14.5 % for the threshold classification and 13.5 % for the KNN classification model. Project-wise results are depicted in Figure 11b.

In both cases all 106 developer models could be created. This emphasizes the major advantage of an universal model. It is appropriate for all developers, e.g., with the universal models we were successful in predicting the sequence of involvement states also for the developers, where no individual model could be calculated. In contrast to the individual models the general model performs about 5 % worse. Therefore, the general model performs well, but for an explicit description of the workload, also other factors should be taken into account.

Application

To demonstrate the potential impact of such a HMM and its usage for software engineering, we now demonstrate a sample application for estimating the output of a project. Concretely, we consider a software project from the starting point of a project manager who aims to estimate the workload and involvement of the team members for the next 24 months. The manager plans with one core developer, two major developers, and five minor developers. Using the general models for the distinct types, the manager can predict the activity of the development team. This can facilitate decisions about the team resource planning. In Figure 12a the produced outcome of commits for one simulation run is represented, where we consider the average contribution per developer of the types major and minor. The resulting involvement states of the run are shown in Figure 12b also as average for major and minor developers. Overall, we estimated 358 commits of the core developer, 187 commits of the two major developers, and 71 distributed among the five minor developers as an average over ten simulation runs.

We now introduce two tools which we developed to support and facilitate our research.

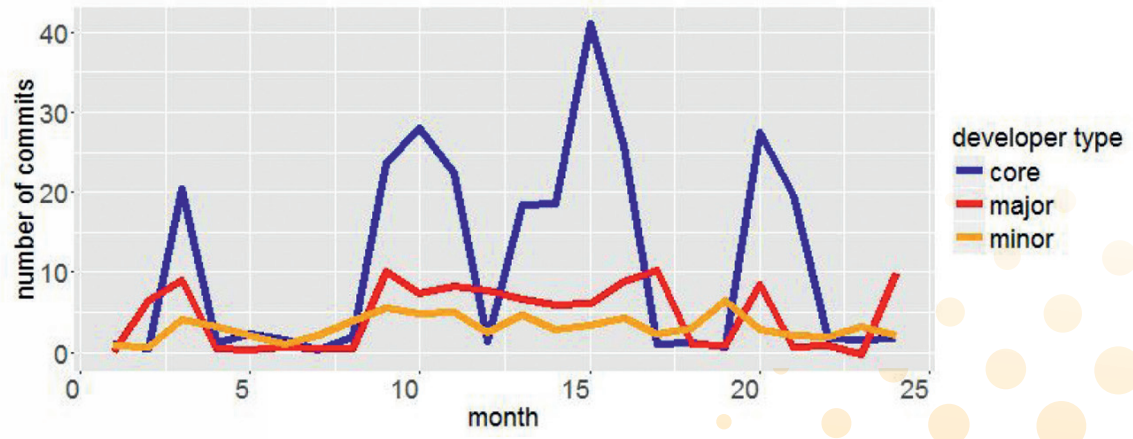


Figure 12a) Prediction of Commits.

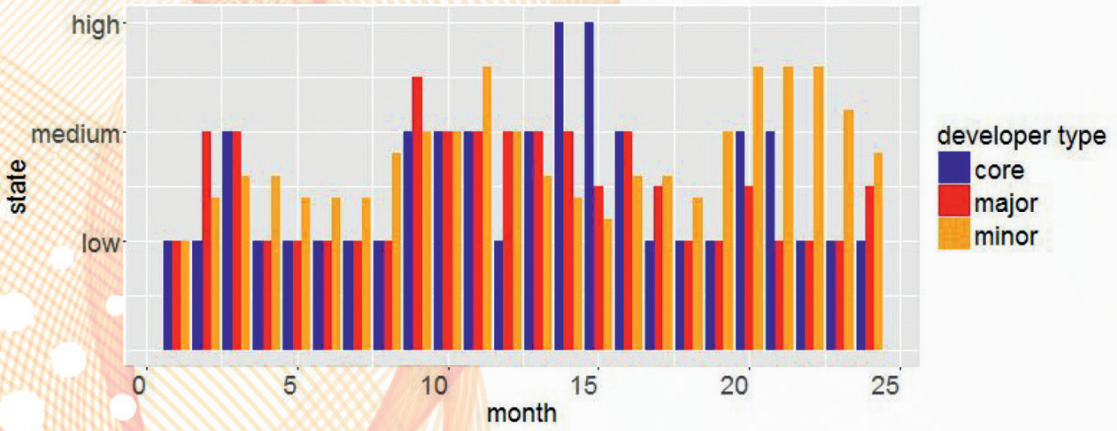


Figure 12b) Prediction of Involvement States.

Tools: Automated Assessment Tool

The assessment of the overall quality of the software using CRF's as described above, is a step that we think should be executed automatically once a user has chosen the parameters for the simulation process (e.g. number of developers, experience level of developers, ...). To achieve this goal we have to define default parameters for all open parameters of the CRF assessment (e.g. coupling- and conformity-parameter in the Ising model). As described in [8], we so far used a reference project with known development to adjust these parameters and use them as default values. However, one project is not enough to satisfy all cases. So the next steps to the automated assessment tool is the determination of more robust default values for the CRF. Once these have been found, the tool, which so far is a stand-alone tool with its own result visualisation, will be integrated into the simulation tool.

Tools: Mining Framework

An essential constituent for building simulation models based on experiences extracted from current and past software projects is the extraction and evaluation of data related to these experiences. We outlined an approach towards a model-based software mining infrastructure [10]. The approach was extended to incorporate further data and refine existing data descriptions.

Besides the use for mining of simulation parameters, the DECENT [10] software mining architecture was integrated in the SmartSHARK platform and used for addressing problems with external validity of repository mining studies [11] (awarded as ACM SIGSOFT Distinguished Paper at the 13th International Conference on Mining Software Repositories (MSR)). Five major problems in the state of practice of software repository mining were discussed and the SmartSHARK smart data platform was proposed and evaluated as a potential solution. The problems related to the external validity include:

- *Heavy re-use of data sets*. Re-use of the same data is important and necessary for the comparability of results. However, relying too much on data re-use poses a threat to the external validity of the results. For example the NASA defect data for software defect prediction is used in at least 58 different studies on software defect prediction [12]. Problems with the quality of this data, as indicated by e.g., Shepperd et al. [13], could threaten the validity of these 58 studies at once.
- *Non-availability of data sets*. In some studies, the opposite is true - the data that is used for the study is not available for validation and comparison to other researchers, making replication impossible.
- *Non-availability of implementations*. The implementations of approaches are not always

publicly available as open source. Re-implementation of complex approaches can require a high amount of resources, limiting the ability of others to replicate a study. In addition, different interpretations of the paper contents due to insufficient level of detail may lead to differences in the re-implementation threatening any replication.

– *Small data sets*. The lack of readily available preprocessed public data limits the size of many studies, often relying on less than ten software projects, threatening to the generalization of results. Only very few studies use larger data sets with more than 100 software projects.
– *Diverse tooling*. In most studies, researchers create their own tool environment based on their preferences, which are often based on existing solutions for data analytics, e.g., R for data mining or WEKA for machine learning. The tool environments range from prototypes that can only be applied to the data used in the case study but nothing else, to close-to-industrial level solutions that can be applied in a broad range of settings. However, these tool environments are usually largely incompatible to each other. Hence, even if all data, implementations, etc. required for a replication are available, the diversity of the required tooling puts a heavy burden on the replication process, especially if multiple results need to be replicated, while still restricting their direct comparability the to incompatibility between the tool environments.

To address these problems, we proposed a smart data platform for conducting experiments, including the estimation of simulation parameters. It provides a solution for both data collection and storage, as well as flexible functionalities for data analytics. The smart data platform can serve as a common data store and thereby, directly enable researchers to share data sets, without additional effort, simply by using the platform. The integration of software analytics directly into the platform gives researchers a common ground for their implementations and thereby facilitates the sharing and replication of approaches, without the problem of different environments for the analytics. Through the automation of the data collection, constant maintenance of the underlying

Table 2: Overview of projects for SmartSHARK evaluation.

Project	Lang.	#Commits	Size	#Files	#ML Messages	Type
guice	Java	1442	89 MB	713	n/a	Programming
openage	C++	1761	8 MB	560	n/a	Game
Minesweeper	Java	65	7 MB	207	n/a	Game
HackerNews	Java	12	1 MB	78	n/a	News
SMSSync	Java	1395	40 MB	557	n/a	Messaging
cursynth	C++	219	3 MB	185	n/a	Audio
passivedns	C	220	1 MB	50	n/a	Network
oryx	Java	372	48 MB	537	n/a	Machine Learning
ohmu	C++	226	3 MB	185	n/a	Programming
libxcam	C++	250	3 MB	242	n/a	Camera
libyami	C	487	4 MB	248	108	Media
wds	C++	238	3 MB	193	27	Media
oclint	C++	733	3 MB	349	n/a	Programming
xgboost	C++	1847	8 MB	373	n/a	Mathematics
elasticsearch-hadoop	Java	1243	9 MB	576	n/a	Search Engine
cxxnet	C++	852	4 MB	173	n/a	Machine Learning
mxnet	C++	236	1 MB	124	n/a	Machine Learning
osquery	C++	2208	9 MB	555	n/a	Instrumentation
swift	Java	496	8 MB	427	n/a	Programming
fatal	C++	401	3 MB	141	n/a	Programming
k3b	C++	5508	47 MB	1069	587	Media
ksudoku	C++	668	7 MB	233	12544	Game
log4j	Java	3266	18 MB	643	54546	Programming

data, as well as the addition of new projects to the underlying database, problems due to heavy data re-use can be mitigated.

To evaluate how well SmartSHARK can address the problems noted above, we conducted several experiments with the platform. We collected data from multiple projects to see if such a high degree of automation for such complex data is feasible. We defined three kinds of software analytics selected in a way to cover a broad area of research topics, including 1) visual analytics of evolutionary trends during the development and within the mailing list usage; 2) a complex machine learning based defect prediction approach; and 3) a statistical evaluation of data in order to support effort estimation. Finally, we evaluated the usability of the platform.
For the data collection, we randomly selected 23 projects from GitHub that were programmed in Java, C, or C++. Table 2 lists the chosen projects, including number of commits, number of files in the repository, size of the repository, programming language, number of stored mails, and a very brief project description. The selected projects can be used for subsequent simulation studies as well.

For the analytics, we created five different visualizations for the added projects:

- *Change history* showing how many software artifacts (i.e., files, classes, functions) were changed at which point in time (i.e., commit).
- *File-level bug overview* depicting the number of changed files over the project lifespan together with the number of defects.
- *Number of sent messages* in the mailing list indicating how many messages were sent at which point in time. The mailing list activity is shown together with the number of changed files for the project in order to allow researchers to look for correlations.
- *Activity of contributing people* in the mailing list showing how many people were active on the mailing list on a monthly basis. The visualization allows to differentiate between new users on the mailing list and those also active in the previous month.
- *Defect prediction results* showing which changes were predicted as defective by the created model compared to which changes were marked as defective by the data collection process. We relied on a change-based defect prediction model based on a where we use the first part of a project as training data, then leave a gap and predict the remainder of the project using a prediction model trained on the first part of the data. This visualization is intended to showcase that results computed by complex analytics can also be visualised with the platform.

The visualizations give a first impression about the project and provide researchers insights regarding several evolutionary questions, e.g., patterns on the mailing list activity, the evolution of the size of a project or when bugs were created. They can also serve as a foundation for simulation parameter identification. In future development of the platform visualisation of simulation outcomes can be integrated in a similar fashion as the visualisation of the defect prediction results, where the performance of different simulation models can be visualised and compared against the observed data for a given project. Having the simulation integrated in the platform can enable quick turnaround on future simulation studies with the data for the parameter estimation and for results comparison readily available within the same platform.

Future development of the SmartSHARK platform may include migration of SmartSHARK to a larger cloud environment to scale even further. A plug-in system can streamline the definition and usage of new data sources, processing steps, analytics, and visualizations. While the DECENT infrastructure provides the overall architecture for the mining process, its deployment and extension across different platforms and cloud environments may require additional overhead, it's embedding into SmartSHARK aims to streamline the process. A plug-in system taking into account the deployment in a cloud environment can streamline its extension. While we performed initial studies to evaluate the usability of the platform by researchers, usage of the platform by additional user groups, e.g., students as part of lectures or other researchers can provide further feedback about usability shortcomings and requirements for future extensions the overall functionality of the platform.

7 Conclusion

We described our simulation-based approach in software quality assurance and prediction as a decision help for project managers in charge of a software project. For this, we explained the overall process, which consist of parameter mining, agent-based modeling and simulation, and automatic assessment. In summary, we can conclude the following points:

- We have shown that it is possible to make a quality-trend-analysis of simulated software according to adjusted parameters like the effort. To make more detailed analyses, we have to extend the simulation model.
- In terms of the assessment part of the project we conclude that we are able to visualize the quality state of the project in a clear way for an observer including dependencies of single entities. So far, however, our reference data is too little to be certain of the parametrization that we have to use for the CRF to achieve robust, comparable statements about the quality state of the software in focus.
- The applicability of HMMs for developer contribution behavior was demonstrated. Moreover, we were able to calculate a general model describing the dynamics developers perform in OSS projects. Interestingly, the emission distribution describing the workload of developers in the different states is more difficult to model generally, but it can be reflected using a general model. A major benefit of the general model is that it is able to calculate the states for every developer.
- We illustrated the use of the general model as a predictor for project managers to estimate the effort of developers in the project.
- Furthermore, we introduced and showed the applicability of our self-developed assessment tool and mining framework.
- Finally, we are able to simulate file dependencies in terms of change coupling with the inferred parameters. The empirical change coupling degree behaves in most instances like the simulated degree.

References

[1] Verena Honsel, Daniel Honsel, and Jens Grabowski. Software Process Simulation Based on Mining Software Repositories. Proceedings of the Third International Workshop on Software Mining (SoftMine), 2014.

[2] G. Gousios, E. Kalliamvakou, and D. Spinellis. Measuring developer contribution from software repository data, in Proceedings of the International Working Conference on Mining Software Repositories (MSR), 2008.

[3] S. Kim, E. J. Whitehead, and Y. Zhang. Classifying Software Changes: Clean or Buggy?, IEEE Transactions on Software Engineering, 2008.

[4] S. Fortunato, Community detection in graphs, Physics Reports, vol. 486, no. 3-5, pp. 75-174, 2010.

[5] M. J. North, N. T. Collier, J. Ozik, E. R. Tatara, C. M. Macal, M. Bragen, and P. Sydelko, Complex adaptive systems modeling with repast simphony, Complex Adaptive Systems Modeling, vol. 1, no. 1, pp. 1–26, 2013.

[6] L. Hattori and M. Lanza. On the nature of commits. in ASE Workshops. IEEE, 2008, pp. 63-71.

[7] Verena Honsel, Daniel Honsel, Steffen Herbold, Jens Grabowski, Stephan Waack. Mining Software Dependency Networks for Agent-Based Simulation of Software Evolution, The 4th International Workshop on Software Mining (SoftMine), 2015

[8] Daniel Honsel, Verena Honsel, Marlon Welter, Stephan Waack, Jens Grabowski, Monitoring Software Quality by Means of Simulation Methods, ESEM '16 Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2016

[9] Verena Honsel, Steffen Herbold, Jens Grabowski, Hidden Markov Models for the Prediction of Developer Involvement Dynamics and Workload , 12th International Conference on Predictive Models and Data Analytics in Software Engineering (PROMISE), 2016

[10] Philip Makedonski, Fabian Sudau, Jens Grabowski, Towards a Model-based Software Mining Infrastructure, ACM SIGSOFT Software Engineering Notes 40(1), ACM, 2015

[11] Fabian Trautsch, Steffen Herbold, Philip Makedonski, Jens Grabowski, Addressing

Problems with External Validity of Repository Mining Studies Through a Smart Data Platform, 13th International Conference on Mining Software Repositories (MSR), 2016

[12] T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell. A systematic literature review on fault prediction performance in software engineering. IEEE Transactions on Software Engineering, 38(6):1276–1304, Nov 2012.

[13] M. Shepperd, Q. Song, Z. Sun, and C. Mair. Data Quality: Some Comments on the NASA Software Defect Datasets. IEEE Transactions on Software Engineering, 39(9):1208–1215, 2013.

Project data

The project was funded from SWZ with one TV-L E13 staff position from May 2013 to April 2016 at the site Göttingen. Involved scientists are:

Prof. Dr. Jens Grabowski
Research group for
Software Engineering for
Distributed Systems
Institute of Computer Science
University of Göttingen

Dipl.-Math. Verena Herbold
Research group for
Software Engineering for
Distributed Systems
Institute of Computer Science
University of Göttingen

Prof. Dr. Stephan Waack
Theoretical Computer Science
and Algorithmic Methods
Institute of Computer Science
University of Göttingen

Marlon Welter, M.Sc.
Theoretical Computer Science
and Algorithmic Methods
Institute of Computer Science
University of Göttingen

Dipl.-Inf. Daniel Honsel
Theoretical Computer Science
and Algorithmic Methods
Institute of Computer Science
University of Göttingen

DeSim: Scalable simulation of systems of cognitive agents

Tobias Ahlbrecht, Jürgen Dix, Niklas Fiekas, Philipp Kraus, Jörg P. Müller

1 Abstract

Using purely agent-based platforms for any kind of simulation requires to address the following challenges: (1) scalability (efficient scheduling of agent cycles is difficult), (2) efficient memory management (when and which data should be fetched, cached, or written to/from disk), and (3) modelling (no generally accepted meta-models exist: what are essential concepts, what just implementation details?). While dedicated professional simulation tools usually provide rich domain libraries and advanced visualisation techniques, and support the simulation of large scenarios, they do not allow for "agentization" of single components. The objective and contribution of the DESIM project was to make a contribution towards bridging this gap by developing a distributed, scalable runtime platform for multiagent simulation, MASeRaTi, addressing the three problems mentioned above. It allows to plug-in both dedicated simulation tools (for the macro view) as well as the agentization of certain components of the system (to allow a micro view). We describe the system architecture developed in the project, which consists of a lightweight kernel and an agent-modeling layer, on top of which applications reside. An evaluation of the platform has been provided by (i) a proof-of-concept implementation of the well-known cow scenario used in the multi-agent programming contest, and (ii) experimentally investigating scalability in comparison to the Jason platform, which represents the state-of-the-art benchmark.

2 Introduction

The SWZ project Decentralized Architectures and Concepts for the Simulation of Systems of Systems (DeSim) started on the 1st of June, 2013 as a cooperation between the research groups of Prof. Dr. Jürgen Dix and Prof. Dr. Jörg P. Müller at the Department of Informatics at TU Clausthal. In addition, Tobias Ahlbrecht M.Sc., Niklas Fiekas B.Sc., Dipl.-Inf. Philipp Kraus and Dipl.-Inf. Michael Köster were involved as scientific staff. The DeSim project, belonging to the Distributed Simulation branch of the SWZ, centers on modelling, analyzing and simulating complex Systems of Systems (SoS), i.e. systems which in turn consist of complex and heterogenous systems interacting loosely coupled and often exhibiting emergent behavior. Conventional simulation methods and methodologies cannot cope with the requirements of SoS, since they do not sufficiently support concepts like autonomy and decentralization concerning the execution and moreover the modelling of simulations.

The overall project goals can be summarised as eliciting requirements for the simulation of SoS, analyzing those requirements regarding essential modelling concepts and decentralized runtime architectures, the analysis, assessment and extension of methods and tools from the Multiagent Systems and Simulation paradigm, and the evaluation of those with the help of selected application scenarios.

Multiagent-based Modelling and Simulation (MABS) is a kind of micro simulation paradigm providing modelling abstractions for concepts like knowledge-based decision making, planning and plan execution, cooperation, coordination autonomy and the description of mixed scenarios with human and machine actors and their interaction.

It uses the concept of a multiagent system as the basic metaphor of the simulation model and has been applied to various domains, such as sociology, economics, spread of diseases, crowd simulation and emergency rescue, and traffic management (see [6] for an overview).

Due to its ability to support fine-grained modeling of a wide range of actors in a simulation, MABS has become particularly popular for modeling and simulating socio-technical systems [28], which consist of human and automated actors which interact in complex ways. Traffic engineering and management [4, 18] is a good example of such a system. Also, the capability of MABS to support the modeling of organization models [9] and normative models (electronic institutions, [28]) makes them well-suited for this type of application. The complexity of agents in such systems can be widely diverse, ranging from cells over ants, simple reactive or procedural agents to cognitive agents that possess an information state, motivations and preferences, and planning capabilities, and that reason over these in order to select actions [25]. Belief, Desire, Intention (BDI) Agents are a typical representative of cognitive agents, which aim to strike a balance between expressiveness and tractability by forsaking planning from first principles and rather resorting to the principle of practical reasoning [19].

There is an inherent trade-off in MABS between the granularity / complexity of agents on the one hand and the expected overall performance of the system. While cellular or ant-based approaches easily scale up to 100.000s or even millions of agents, today's cognitive agent platforms such as Jason will only scale up to limited problem sizes (mostly not more than a few hundred agents). However, standardized models are lacking beyond the conceptual level, where e.g. the Belief, Desire, Intentions (BDI) model [31] sees widespread acceptance. Therefore, also scalable runtime platforms which allow for the efficient simulation of computationally expensive MABS models are not yet off-the-shelf available. We found that most platforms are either written in Java or are not scalable for other reasons. Many are only used in academia and simply not designed to run on a high performance computing cluster. Common challenges relate to agent runtime representation and communication performance.

Three major technical challenges to be addressed are (1) the efficient scheduling of agent cycles by a simulation engine, (2) efficient memory management (when and which data should be fetched, cached, or written to/from disk), and (3) modelling (no generally accepted meta-models exist: what are essential concepts, what just implementation details?).

Thus, the main contributions of the project are: (1) an analysis of the state of the art in agent-based simulation platforms, leading to a set of requirements to be imposed on a simulation platform, focusing on runtime scalability and efficient memory management; (2) the proposal of a novel architecture and design of the MASeRaTi simulation platform, bringing together a robust and highly efficient agent kernel (written in Lua) with a BDI agent interpreter including multiagent concepts such as communication and computational norms; and (3) a proof-of-concept evaluation of our approach, describing two application scenarios (the Cows scenario from the multiagent programming contest (MAPC), and an urban traffic simulation scenario) and providing initial experimental results comparing the performance of MASeRaTi against Jason.

3 Use cases for MABS: ATSim and MASSim

We first present two simulation system case studies from our previous research: One to simulate urban traffic management, the other to simulate arbitrary agent systems in one single platform:

3.1 Traffic simulation (ATSim)

In [13], we presented the ATSim simulation architecture that integrates the commercial traffic simulation framework AIMSuN with the multiagent programming system JADE. AIMSuN is used to model and simulate traffic scenarios, whereas JADE is used to implement the informational and motivational states and the decisions of traffic participants (modelled as agents). Thus, all features of AIMSuN (e.g. rich GUI, tools for data collection and data analysis) are available in ATSim, while ATSim allows to simulate the overall behaviour of traffic, and traffic objects can be modelled as agents with goals, plans, and communication with others for local coordination and cooperation.

AIMSuN (Fig. 1(a), left side) provides an API for external applications to access its traffic objects via Python or C/C++ programming languages. But the JADE-based MAS (right side of Fig. 1(a))

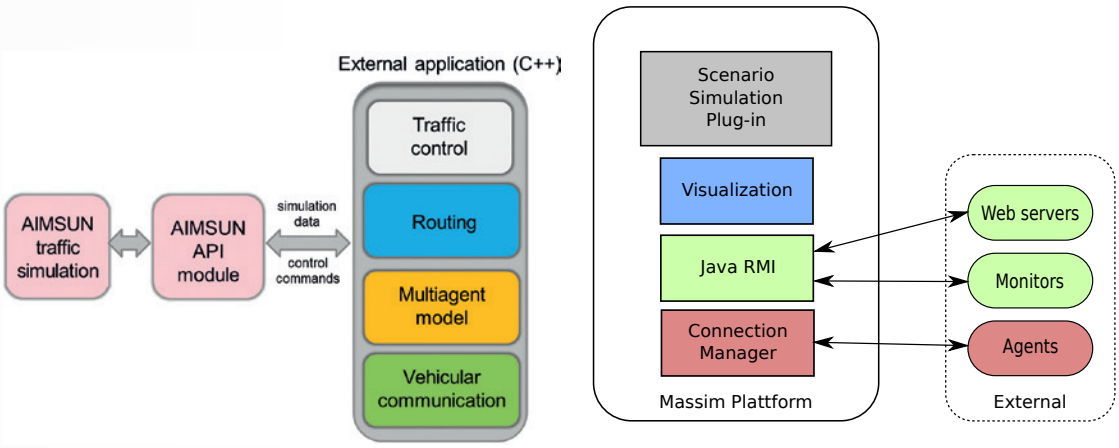


Figure 1: Overview of the platforms

is implemented in Java, which leads to problems with scalability. To enable AIMSuN and the MAS to work together in ATSim, we used CORBA as a middleware. For details on the integration architecture, we refer to [13]. Two application scenarios were modelled and evaluated on top of ATSim: The simulation of decentralized adaptive routing strategies, where vehicle agents learn local routing models based on traffic information [18], and cooperative routing based on vehicle group formation and platooning [20]. The overall system shown in Fig. 1(a) was developed in a larger research project and contained additional components for realistic simulation of V2X communication (extending the OMNET++ simulator), and for formulating and deploying traffic control policies; see [17].

Our evaluation of the ATSim platform using a mid-sized scenario (rush hour traffic in Southern Hanover, one hour, approx. 30 000 routes, see [17]) showed that the agent-based modelling approach is intuitive and suitable; it allows for maximal freedom in the implementation of arbitrary properties, preferences, and capabilities of the entities at the micro-level. However, we also found that our integration approach runs into severe scalability issues. Immediate causes identified for this were the computationally expensive representation of agents as Java threads in Jade and the XML-based inter-process communication between Jade and the AIMSuN simulator. In addition, system development and debugging proved difficult because two sets of models and runtime platforms needed to be maintained and synchronised.

3.2 Multi-Agent Programming Contest

The MASSim platform [8, 7] is used as a simulation framework for the Multi-Agent Programming Contest [5](http://multiagentcontest.org). Agents are running remotely on different machines and are communicating in XML with the server over TCP/IP. The server computes the statistics, generates visual output and provides interfaces for the simulation data while the simulation is running. A drawback of dividing the simulation in such a way is the latency of the network that can cause serious delays. Network communication becomes a bottleneck when scaling up; the slowest computer in the network is determining the overall speed of the simulation. Running the simulation in one Java virtual machine leads to a centralised approach that might impede an optimal run (in terms of execution time) of a simulation.

Fig. 1(b) depicts the basic components of the MASSim platform. MASSim will mainly serve us as a reference to compare scalability with MASeRaTi right from the beginning (using the available scenarios). We want to ensure that MASeRaTi outperforms MASSim in both computation time and number of agents.

4 The MASeRaTi Simulation Architecture

We will now elaborate on lessons learned and show how this resulted in the new idea of the scalable runtime platform MASeRaTi [3]. After identifying the key challenges, requirements, and research questions, we present the overview of the MASeRaTi architecture.

4.1 Challenges and requirements

The MASeRaTi simulation platform aims at combining the versatility of an agent-based approach (the micro-view) with the efficiency and scalability of dedicated simulation platforms (the macro-view). We reconsider the three challenges mentioned before for using a purely agent-based approach:

Scalability: Efficient scheduling of agent cycles is a difficult problem. In agent platforms, usually each agent has its own thread. Using e.g. Java, these threads are realised in the underlying operating system which puts an upper limit of approximately 5000 agents on the system. These threads are handled by the internal scheduler and are therefore not real parallel processes. In the MASeRaTi architecture we develop a micro-kernel where agents truly run in parallel. In this way, we reduce the overhead that comes with each thread significantly. We believe that this allows for a much better scalability than agent systems based on (any) programming language, where all processes are handled by the (black-box) operating system. Additionally, many simulation platforms use a verbose communication language (e.g., XML or FIPA-ACL) for the inter-agent communication; this becomes a bottleneck when scaling up. We exploit the efficient synchronisation features of MPI instead.

Efficient memory management: Which data should when be fetched from disk (cached, written)? Most agent platforms are based on Java or similar interpreter languages. When using them we have no control over the prefetching or caching of data (agents need to access and reason about their belief state): this is done by the runtime mechanism of the language. We do not know in advance which available agent is active (random access), but we might be able to learn so during the simulation and thereby optimise the caching mechanism [2]. This is the reason why we are using the scripting language Lua in the way explained below.

Modelling support: As of now, no generally accepted meta-model for multiagent-based simulations exists. We would like to distinguish between essential concepts and implementation details. What are the agents in the simulation? Which agent features are important? We also want the modelling framework to assist a simulation developer in creating a scenario as well as hide the complexity of a parallelised simulation, while not being restrictive in terms of modelling capability.

We formulate three basic requirements for MASeRaTi that directly follow from the identified challenges: (1) the support of a macro and micro view of a simulation, (2) a scalable and efficient infrastructure, and (3) a multiagent-based simulation modelling framework that also supports non-agent components.

So the main problem we are tackling in the MASeRaTi platform is the following: How can we develop a scalable simulation environment, where the individual agents can be modelled / programmed using suitable abstractions and where one can abstract away from specific features? We would like to reason about the macro view (usually supported by dedicated simulation tools) as well as zooming into the micro view when needed. In particular, the overhead for supporting the micro view should not challenge overall system scalability:

1. If no agents are needed (no micro-view required), there should be only a minimal performance penalty for using MASeRaTi on top of the underlying simulation platform ("legacy system").
2. If no legacy code at all is used, MASeRaTi should still perform better or at least comparable to most of the existing agent platforms (and it should have similar functionality).

Note that due to general considerations (Amdahl's law [21]) and the fact that not all processes are completely parallelizable, it is not

possible to achieve requirement (1) perfectly (no agents: performance of MASeRaTi equals performance of legacy code).

In addition to a scalable platform we also provide a meta-model for multiagent-based simulation in order to address the third challenge. We will now focus on the first two challenges. The meta-model serves as a general starting point for the development of a MABS and ensures a certain structure of a simulation that is needed by the underlying platform in order to facilitate scalability.

4.2 Architecture overview

The overall architecture of our framework, as presented in [4], is inspired by concepts from game development. Today's massively multiplayer online role-playing games (MMORPGs) are mostly based on a client-server architecture where the clients are synchronised during game play [14] via a messaging system. Examples include Blizzard's World of Warcraft or EA's SimCity, which supports multiplayer gaming with an "agent-based definition" in its own Glassbox engine (http://andrew-willmott.com/talks/inside-glassbox).

While a game architecture is a good starting point for our purposes, of course we cannot assume a server system with hundreds of hardware nodes, which is powerful enough to handle a MMORPG system. Also, for developing purposes, we need a system running on a single node (think of a common desktop PC). The code for the simulation model developed there should then be transferable to a HPC system, where the real simulation is executed.

Our underlying agent-oriented meta-model uses the well-established concept of a BDI agent [25, 43] in a variant inspired by the agent programming language Jason [11] combined with the concept of artifacts [10] and the idea of an entity [7] that evolved out of experiences gathered in the MAPC. However, these concepts are exchangeable, due to the flexibility of Lua, and developers are not forced to use them. In our agent model, agents connect to these entities in the simulation world. Agents consist of a body and a mind: While the mind (being responsible for the deliberation cycle, the mental state etc.) does not have to be physically grounded, the entity has to be located in an area of the simulation. Thus, an entity is an object with attributes that an agent can control and that might be influenced by the actions of other agents or the overall simulation. Intuitively, an agent can be viewed as a puppet master that directs one (or more) entities.

The architecture of MASeRaTi is composed of three layers as illustrated in Fig. 2:

Scenario layer (SL): The topmost layer represents a specific agent-based simulation model, e.g., a

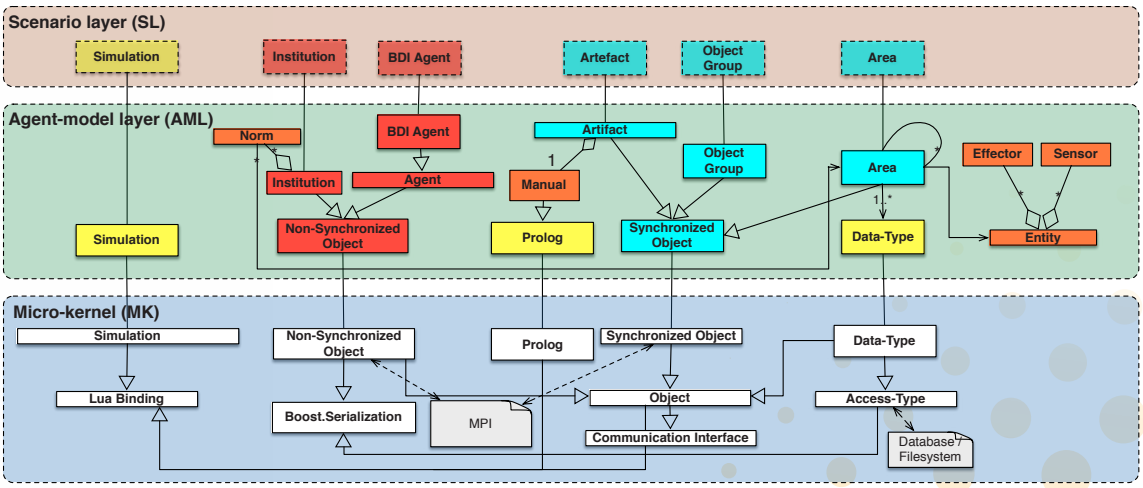


Figure 2: MASeRaTi system architecture: UML class diagram

traffic flow scenario with a number of vehicles and a demand model given by an origin-destination matrix, or the MAPC cow scenario. A scenario layer model refers to model elements provided by the next lower layer, the agent model layer (see below). The SL is illustrated by dotted boxes in Fig. 2.

Agent-model layer (AML): The agent-model layer introduces agent-oriented concepts (e.g. agents, environments, artifacts etc.) to the system and thus describes a model of an agent-based simulation. We provide two implementations of the AML: A version supporting the coding of agents in AgentSpeak, and one implemented in the scripting language Lua(http://www.lua.org/) [23] to ensure maximum flexibility. Due to the multi-paradigm nature of Lua, pure object-oriented concepts are not supported by default. That is, Lua uses only simple data types and (meta-) tables. The agent-oriented abstractions are then introduced through an object-oriented structure in Lua itself. This allows us to work in a uniform fashion with UML models regarding the AML and the scenario layer.

Micro-kernel (MK): The micro-kernel represents the technical backbone of the system. It is written in the C++ programming language to get the necessary performance and provides basic network parallelisation and scheduling algorithms. The layer defines the system's underlying structure con-taining interfaces e.g. for plug-ins, serialisation, Prolog for knowledge representation and reasoning, or statistical accumulation. In short, this bottom layer describes a meta-model for a generic parallel simulation.

An important aspect is the linkage between the three layers, and in particular the connections between the micro-kernel and the AML (illustrated in Fig. 2) which is discussed further in the following.

5 The MASeRaTi micro-kernel layer

The microkernel, the lowest layer, represents the scalable heart of the runtime system of MASeRaTi. It is split up into two main structures (Fig. 3(b)). The core part (below) contains the scheduler algorithms, the core and memory management, the network and operating system layers and the plug-in API together with a Prolog interpreter. Above these core utilities, a scripting language (Lua, AgentSpeak) interpreter (top) is defined and each class structure on the core can be bound to its objects.

While the agent-oriented AgentSpeak maps naturally to agent concepts, the Lua bindings require some more discussion: The Lua runtime is instantiated for each process once, so there is no elaborated bootstrapping. The choice of Lua was mainly motivated by the scaling structure and the game developing viewpoint. Lua, a multi paradigm language, has been used for game development for many years [24]. An advantage of Lua is the small size of its interpreter (around 100 kBytes) and the implementation in native C with the enhancement to append its own data structures into the runtime interpreter with the binding frameworks. The multiparadigm definition of Lua, especially object-oriented and functional [23], can help us to create a flexible metamodel for our simulation model. Lua can also be used with a just-in-time compiler.

5.1 Kernel data structures and components

The kernel defines basic data structures and architectural components as illustrated in Fig. 3(a):

Simulation: A global singleton simulation object, which stores all global operations in the simulation e.g. creating agents or artifacts. It defines the initialization of each simulation; the constructor of the Simulation object must create the world object, agent objects, etc.

Object: Defines the basic structure of each object within the simulation. All objects contain a UUID (Universally Unique Identifier), a statistical map for evaluating statistical object data, the (pre-/post-) tick methods to run the object and the running time data, which counts the CPU cycles during computation (for optimisation).

Prolog: An interface for using Prolog calls within the simulation.

Each class is derived from the Lua Binding class, so the objects are mapped into the AML. The mapping between the micro-kernel and the AML is defined using a language binding concept. The Lua interpreter is written in native C. Based on this structure, a C function can be "pushed" into the

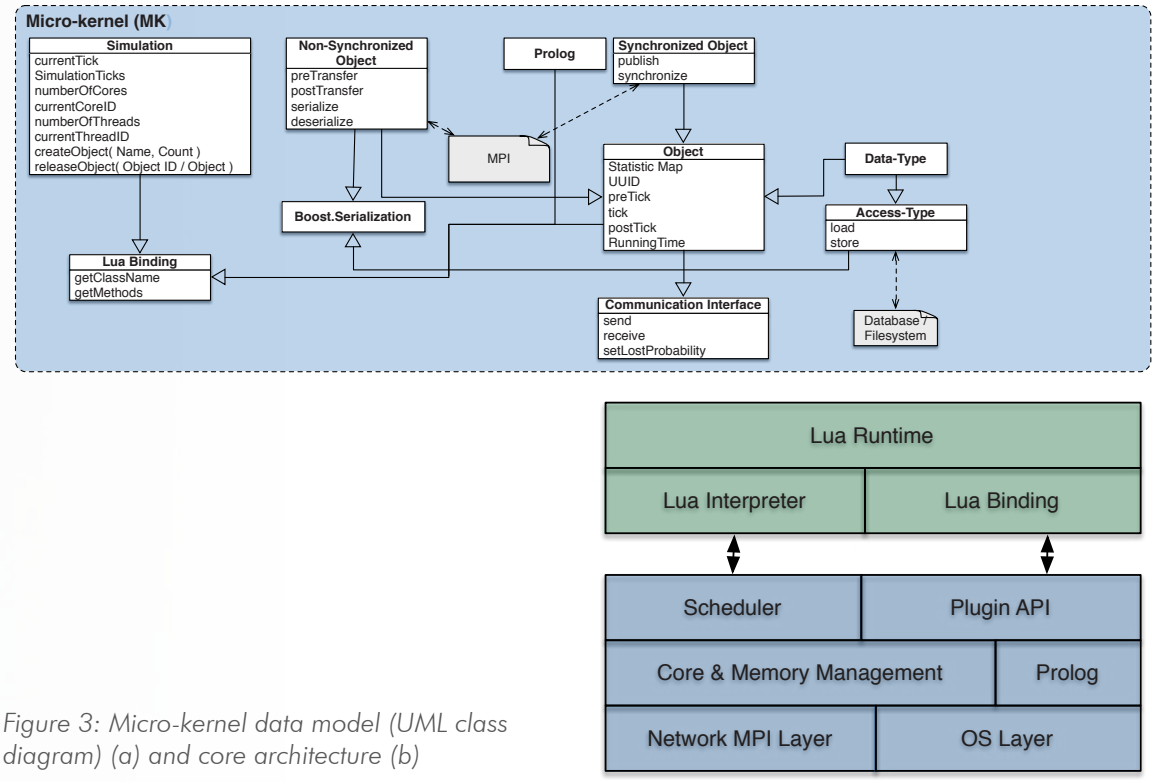


Figure 3: Micro-kernel data model (UML class diagram) (a) and core architecture (b)

Lua runtime. The function is stored into a global Lua table; the underlying C function is used with a script function call.

Our concept defines the micro-kernel in UML; instantiated C++ objects are mapped into the runtime environment by a Lua binding framework (e.g. Lua Bridge or Luabind). Classes and objects in Lua are not completely separateentities, as a class is a table with anonymous functions and properties. If a Lua script creates an object, it calls the constructor, which is defined by a metatable function, the underlying C++ object is also created and allocated to the heap. The destructor call to an object deterministically removes the Lua object and its corresponding C++ object. All C++ objects are heap-allocated and encapsulated by a "smart pointer", as this helps avoiding memory leaks. This concept allows consistent bindings between the different programming languages and the layer architecture.

Each Object comes from the Communication interface, which allows an object to send any structured data to another object. Three subclasses inherit from the central Object. This structure is necessary for creating a distributed and scalable platform with optimisation possibility:

Synchronised Object: An object of this type is synchronised over all instances of the micro-kernel (thread and core synchronised). It exists also over all instances and needs a blocking communication. In the agent programming paradigm the world must be synchronised.

Non-Synchronised Object: This object exists only at one instance of the micro-kernel and can be transferred between different instances of the micro kernel. It should be used e.g. for agents and norms, because the evaluation is independent from other objects. Using the "execution time" of the tick (time complexity), we can group such objects together.

Data-Type: This object represents a data structure, e.g. a multigraph for the traffic scenario with routing algorithms (Dijkstra, A* and D*). The data

types will be pushed into the micro-kernel with the plug-in API. The Access-Type creates the connection to the storing devices.

Synchronised and non-synchronized objects are implemented via Boost.MPI1 structure, and the Access-Type defines the interface to a database or the filesystem for storing / loading object data. The access via the data interface will be defined by the Boost.Serialization library (http://boost.org/doc/libs/release/libs/), so we can use a generic interface.

5.2 Scheduling and optimization algorithms

Next, we discuss alternatives and trade-offs when designing a runtime system to take a deeper look into that of MASeRaTi. During runtime we propose to ask which objects need to be defined as synchronised or non-synchronised datasets. The implementation of the FIPA-ACL definition, e.g., is a blocking operation, because we can update the object only after we have processed the input data, so each external data input creates a slower performance. With the implementation we create additional workload, because parser, lexer and interpreter must also process the data.

One MASeRaTi runtime instance implements a thread-pool, which processes all objects. Scalability is obtained by looking at local instances and taking the global view over all instances into account.

[30] identify a number of generic pitfalls in developing agent-based systems, including:

1. "You forget that you are developing multi-threaded software".
2. "Your design does not exploit concurrency".
3. "You have too few agents".

These three statements underline the importance of scalability, concurrency, and exploiting parallel hardware in the context of developing agent-based systems. We aim at addressing these pitfalls by designing a scalable, multi-threaded and multi-core system which can handle a large number of agents that act concurrently. In order to achieve this under the given technical restrictions (memory and number of threads), we take the following approach:

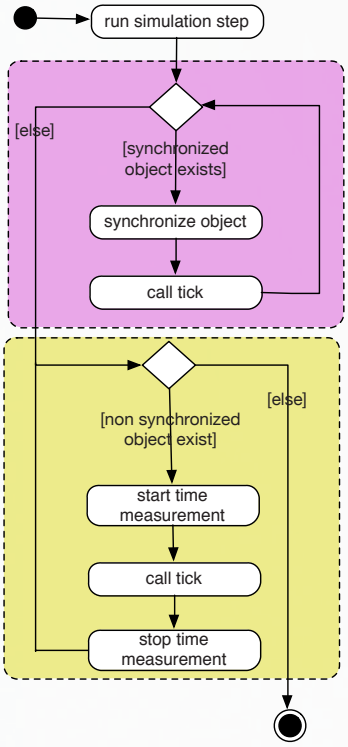


Figure 4: Definition of a simulation step ("tick")

- We create a dedicated scheduler to handle the agents. It is based on a thread pool. In detail we use an asynchronous thread execution structure.
- We measure the average calculation time for each agent when it is active (counting the CPU cycles).
- Based on this, we optimise the number of agents between the micro-kernels with a thread-/core-stealing algorithm (in future work we aim to describe this with a stochastic process).
- As discussed earlier, there are two disjoint sets of objects in our simulation: non-synchronised and synchronised objects. The following pseudo code (Fig. 4) shows the definition of one discrete simulation step, which we denote by the term "tick":

Each simulation object exposes a tick method, which is called by the scheduler (pre/post tick calls are not shown here). There exist only two global blocking operations for synchronisation over all

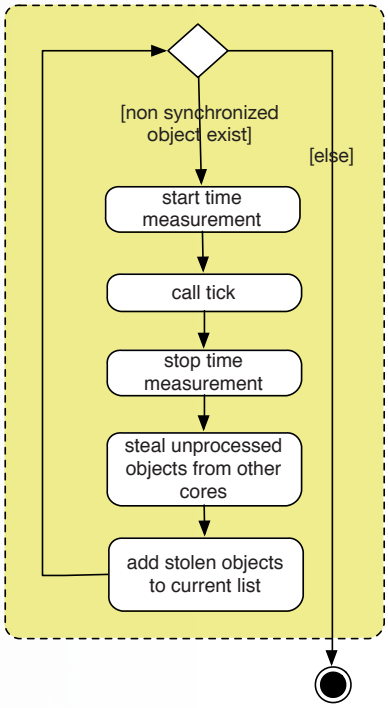


Figure 5: Stealing process on an instance

kernel instances. Each micro-kernel process runs the (global) synchronised objects first. After finishing, the simulation environment is synchronised on each kernel. In the next step, the kernel runs the non-synchronised objects. This second loop can be run in a fast way, e.g. the agents do nothing, so the micro-kernel idles, then the while-loop sends "steal requests" and gets objects from the other instances (Fig. 5).

Fig. 6 shows the stealing process (bullets are the agents, with different calculation times) over all running MASeRaTi instances

This idea allows the processing of a large number of agents with different (complex) plans and belief bases, because we can define the optimisation process with different targets and methods. Plans are sequences of actions and expressions, so action can modify the environment. Because of the asynchronous structure of the thread-pool, there is no deterministic order of action execution. Based on the scenario specification, the ordering of environment modification operations must be set up a-priori by the environment developer. Strategies for this include Map & Reduce, immediate change (first-come-first-serve), or priority based. In detail the agent does not define the timestep when the physical result is finished within the environment; rather, the agent sends a request to the environment for modification. Based on the environment definition, the action result is physically written after all agents have finished their ticks. The runtime model can be represented as a discrete stochastic approach. It is the designer's responsibility to ensure that the environment reaches a stable state (process equilibrium) after all agents have been executed.

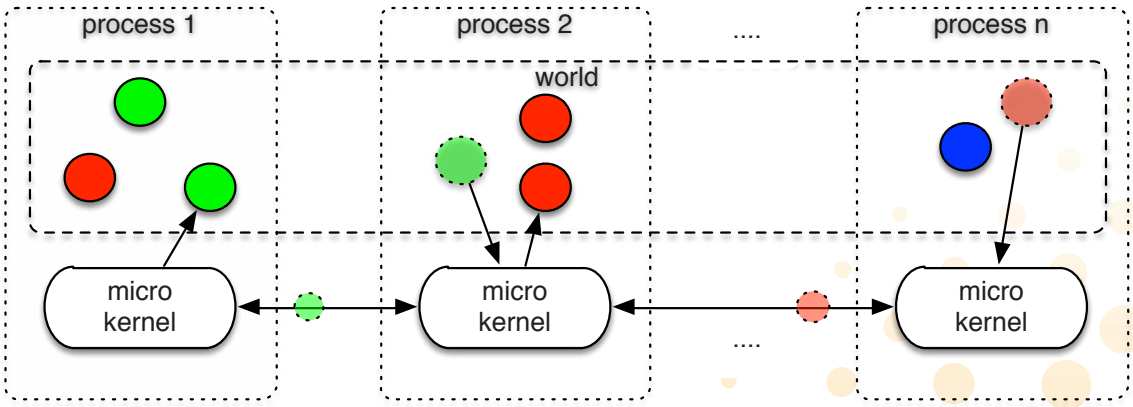


Figure 6: Stealing process over all platform instances

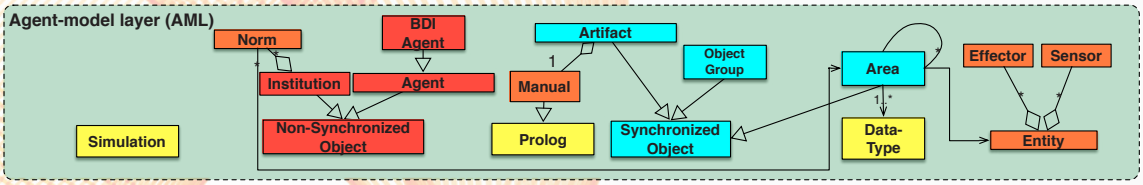


Figure 7: Agent-model layer: UML class diagram

6 Agent-model Layer

The agent-model layer (AML) (depicted in Fig. 7) defines a meta-model of an agent-based simulation. It provides the basic structure and serves as a starting point for an implementation. We start by explaining the structure, followed by the overall system behaviour; we then briefly describe the AML development process and end with a discussion of design tradeoffs. Realization details (pseudo code) can be found in the appendix of [2].

6.1 AML Structure

The structure of the meta-model is heavily influenced by the goal of creating a step(or cycle-) based simulation which can be distributed over several nodes or cores. In such a parallelised multiagent simulation, the developer has to determine for each object whether it has to be present on each and every core or if it is sufficient to have the object running independently on a single core only. We prefer the latter, since that implies less execution time. In contrast, an object like a global artifact has to be accessible by virtually any other object. Thus, it must be made available and therefore executed on each core.

The goal of the AML is to simplify the development of parallel multiagent simulations by defining a number of abstract objects or object categories that normally have to be synchronised and those that can usually run independently. Nevertheless, a developer can easily modify the AML to her needs, in particular redefining the synchronicity of objects.

Fig. 7 illustrates the structure of the AML. Mainly, a simulation consists of a singleton Simulation, the non-synchronised object types Agent, Norm, and the synchronised classes Area, Artifact, ObjectGroup. While for the Simulation only one instance is allowed, the other objects can be instantiated several times. All instantiated objects are being executed in a step-based fashion and therefore implement a tick method that is called exactly once per simulation cycle.

Simulation: The Simulation class in the AML is the Lua-based counterpart to the Simulation class in the MK. It is responsible for the creation, initialisation and deletion of objects. Thus, it is in full control over the simulation.

Agent: As we aim to simulate as many agents as possible we have to ensure that this part of the model can run independently from the rest. Therefore we define two kinds of agents as non-synchronised objects: a generic Agent based on [29] and a more sophisticated BDI Agent [27] inspired by Jason [11]. Both agent concepts are, in spite of their origins, still realised in Lua. The agent interacts with the environment through Entities [7]. In general, an agent can have random access to the simulation world. Therefore, we can only encapsulate some parts of the Agent, namely the internal actions and functions like reasoning. But the effects on the environment have to be synchronised to make them known to all other objects. This is the reason for splitting the agent into two parts: the mind (the Agent) and the body (the Entity). Originally, an Agent is associated with no Entity, but it can be bound to (or bind itself to) one or more Entities to perceive and act in the environment. The generic Agent has three methods that are invoked in the following order: (1) perceive, (2) think, and (3) act. Inside these methods, those of the respective Entity can be called directly while communication between objects has to be realised by a synchronised object (for instance by means of an Artifact). Thus, no specific (other) communication mechanisms

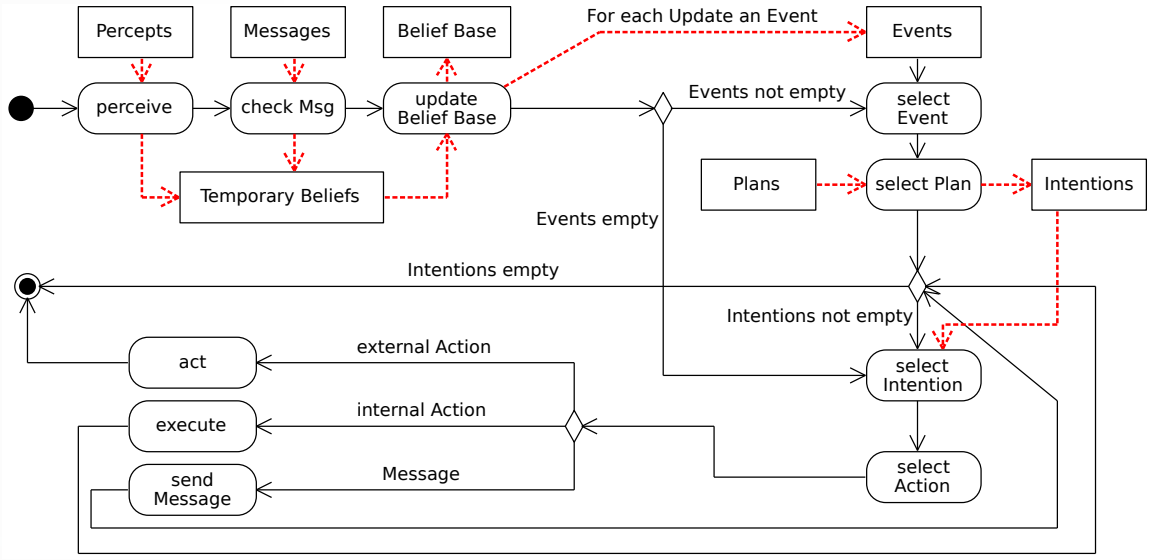


Figure 8: BDI agent cycle: Activity diagram and data flow. Activity is red/dashed and data flow is black/solid

are supported as of now. Agents can be created and destroyed at any point during the simulation. While the Agent exists, it is triggered once per simulation cycle.

BDI Agent: The BDI agent is a little more sophisticated and consists of a Belief Base representing the current world view, a set of events describing changes in the mental state, a set of Plans, and a set of Intentions describing the currently executed plans. Note that events only exist within an agent. They are generated in an agent cycle and processed in the same or one of the following cycles. Fig. 8 shows an overview of the agent cycle, which is a derivation of the Jason agent cycle [11]. One agent cycle is completely processed in one simulation tick once per Agent. Black (continuous) lines represent the activity flow while red (dashed) lines show the data flow. Naturally, the agent cycle is executed from within the agent's tick method. In each tick, the agent first perceives the environment and checks for new messages. Based on this information, the belief base is updated and an event for each update is generated. From the set of events one particular event is selected and a plan that matches this event will be chosen and instantiated. During a simulation run this might result in multiple instantiated Plans at the same time and allows the Agent to pursue more than one goal in parallel. We decided to limit the Agent to the execution of one external action (that affects the environment) but allow several internal actions per simulation tick. The next method selects the next action of an instantiated Plan (i.e. the next action of an intention). As mentioned, in contrast to Jason, the agent cycle does not stop here if it was an internal action or a message, i.e., an action that does not affect the environment. Thus, the agent selects the next event (if possible) or next Intention (if possible) until (1) it reaches a global timeout (set by the simulation) or (2) an external action is executed that forces a synchronisation, or (3) if the set of events and intentions are both empty. This ensures synchronisation of an agent cycle to the tick-based nature of the simulation, while retaining the aspect of performance through the timeout.

As of now, Beliefs, goals and Plans are encoded in simple Lua tables, which can prove a little cumbersome during development. However, as the agent cycle is derived from Jason, using Jason-like syntax in the future seems to be a viable option.

Artifact: For all passive objects of a simulation we use the Artifact methodology defined in [10]. Basi-

cally, each Artifact has a type, a Manual in Prolog (a description of the possible actions associated with it) and a use method that allows an Agent to execute a particular action, i.e. make use of the Artifact. Due to the generality of this approach the developer decides whether the actions are known by the Agents beforehand or not. Additionally, since the Artifact is defined as a synchronous object, one can consider a variation of this object that implements a method for each of its offered capabilities and allows for direct method invocation.

Area: So far, we defined the main actors of a simulation but how are they connected among each other? An Artifact does not have to be located inside a real simulation, i.e., it does not need a physical position (in contrast, most objects do need one). Therefore, we define an Area as a logical or physical space (similar to the term locality introduced by [22]). There can be several Areas, subareas, and overlapping Areas. In the general case, Agents can perform a random access on the environment, so the Areas have to be synchronised and thus be available on all cores of the simulation platform. Within an Area, we define some basic data structures and algorithms for path finding, etc. The most important issue, the connection of the non-synchronised agents with the synchronised Areas is realised by the use of Entities. Agents perceive the environment and execute actions by using the Entities' sensors and effectors.

Entity: An Entity can be seen as the physical body of an Agent located inside an Area. An Agent can register to it, get the sensor data, and execute actions that possibly change the environment. The Entity has some effectors and sensors that are easily replaceable by the simulation developer. Since such an Entity represents the physical body of an Agent and is meant to connect an Agent with the environment, it has to be synchronised over all cores. Entities can be added to and removed from the simulation at any point. Agents who are still registered with the Entity are being notified if the Entity is destroyed. The Agent will continue to exist and can bind to other Entities.

Institutions & Norms: For now, we provided a rudimentary model as a placeholder for future extension: An institution is an object that checks for norm violations and compliance. It operates as a monitor and is also responsible for sanctioning. But a developer can also decide to separate these two tasks. For the future, we are planning to focus only on three kinds of norms: obligations, permissions, and prohibitions. Additionally, we will focus on exogenous norms (events that occur in at least one area) and not on rules that affect the Agent's mind, plans etc.

ObjectGroup: Finally, an ObjectGroup – as the name implies – defines a group of objects. It can be used to group Agents, Artifacts or other objects. Method calls on an ObjectGroup are forwarded to all group members, i.e., with a single method call, all corresponding methods (with the same type signature) of the group members are invoked. In order to reduce overhead and to avoid circular dependencies we only allow a flat list of members at the moment. However, if a hierarchy is needed, it can be easily implemented.

6.2 AML behaviour

So what does the overall behaviour look like? Initially the simulation object creates a number of agents, areas, object groups, norms, etc., and changes the global properties in the three phases: preTick, tick, and postTick. It can delete and create new agents during runtime. But if the simulation developer decides to allow an agent to create another agent, this is consistent with the metamodel. The agent cycles are executed in each tick method. Also, the main procedures of artifacts, norms and areas are executed in this phase. The preTick is intended to be used as a preparation phase and the postTick phase for cleaning up.

6.3 AML Modelling process

The AML is currently targeted by a platform-independent modelling process that aims to differentiate between simulation developer and designer. The core part consists of a yet-to-be-finished metamodel that resembles the AML partially but also includes a number of concepts for simulation scheduling, control, visualisation and statistical analysis.

The simulation developer's task is then to model the domain-specific part of the simulation by

instantiating the metamodel. The result is a model containing types of objects which are to be used in the simulation, i.e. cars or traffic lights. At the same time, the simulation developer can model platform-specific features, e.g. how a certain object type should be displayed or observed. Afterwards, the simulation designer can instantiate another part of the metamodel to specify a concrete simulation instance referencing the objects from the domain model.

To get the model into a form that is usable by the simulation platform, a code generation mechanism is employed. That is, for each class of objects a so called template must be present telling the generation mechanism what files to create and how to fill them. Depending on the target platform, creating templates probably entails some general programming.

To summarise, the process looks as follows (for an illustration, see Fig. 9):

1. The domain-specific and platform-specific model parts are created (simulation developer),
2. the necessary templates are acquired or created (simulation developer),
3. the simulation model is created (simulation designer), and
4. the platform-specific (i.e. AML-compliant) files are generated.

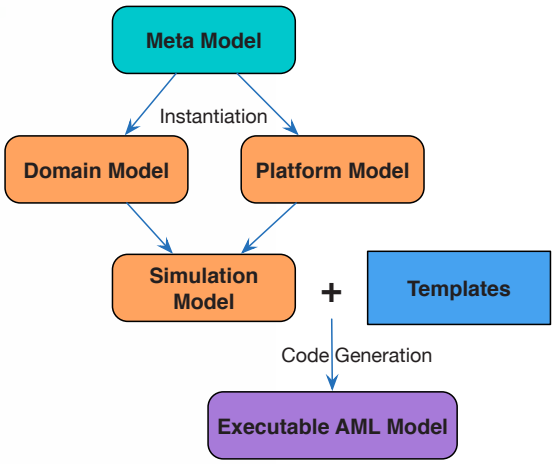


Figure 9: Development process

Surely, this process can also be integrated into most agent-oriented modelling methodologies by either using it directly in the methodology's design phase (if the methodology allows for any metamodel) or by extending the metamodel with the methodology's concepts, so that the modelled results are in turn compatible with the methodology.

One of the advantages of this method is the clear separation of roles. This enables the simulation designer to create simulations possibly without the need for programming. If the designer gets stuck, the developer can create a new object type for the designer to use. Also, reusability is allowed for by the template mechanism when the same platform is targeted and reuse of the domain part is enabled by swapping the platform-specific model part and the templates.

A graphical modelling tool supporting this process is currently being developed [1].

6.4 Design tradeoffs

As we have seen, the AML tries to facilitate the modelling of a parallel multiagent simulation by helping the developer deciding whether objects have to be synchronised or not. Of course, our classification might not fit each and every possible use case. But because of the flexibility of this layer, it is possible to easily adapt the AML to the specific domain.

Also, the layer cannot provide all of the concepts related to the agentoriented paradigm. We tried to identify those which are of utmost importance and thus form something like the least common denominator of all agent-based simulations. If further concepts are needed, they can be easily added on demand or might be readily available if already implemented for another use case.

We mentioned that our BDI agent is restricted to perform at most one external action per simulation cycle, while it is allowed to perform internal actions until it runs out of time. It will be easy to change this behaviour if it proves to be disadvantageous both in terms of agent or platform performance.

We provided a BDI agent in order to (1) show how to transfer an agent concept to the platform

at this level of implementation and (2) ensure that the platform is easily usable if no specific kind of agent is needed. While our platform is open to use any agent concepts, it does not have to.

7 Evaluation

We evaluated the MASeRaTi framework with respect to the following five research questions [RQ]:

- **RQ1**: Is the expressiveness of the meta-model sufficient for modeling typical multi-agent scenarios, which reveal distributed computation, local decision-making, and interaction?
- **RQ2**: Is the overall architectural approach of MASeRaTi feasible? I.e., can it run models designed to answer RQ1?
- **RQ3**: Is the MASeRaTi architecture sufficiently flexible? Can it support simulations with more complex domain-specific models?
- **RQ4**: Is the runtime system sufficiently scalable with respect to system sizes (measured by number of agents) when applied to a generic baseline scenario involving both local computation and communication?
- **RQ5**: Is the runtime system sufficiently scalable with respect to system sizes (measured by number of agents) when applied to a richer, cognitive agent scenario?

To address research question RQ1, we provide a proof-of-concept instantiation of the MASeRaTi architecture layers for the Cows scenario as used in the MAPC. We also address question RQ2, show that the cow scenario model (RQ1) can be run on MASeRaTi.

Then, to show flexibility and the capability to support simulations with more complex models (RQ3), we have implemented a traffic flow simulation based on a Java implementation of the MASeRaTi runtime.

Question RQ4 is addressed afterwards. We experimentally benchmark the AgentSpeak interpreter version of MASeRaTi through the counting scenario used by eJason [15] to establish a simple performance baseline, and comparing it with the performance obtained by using the Jason platform.

Finally, research question RQ5 is investigated by using a slightly simplified version of the Cows scenario as a benchmark problem to assess the performance in a more complex environment. Also here, we compare the results with that obtained by using the Jason platform.

7.1 Proof of concept: The Cows scenario

We test the expressiveness of the proposed meta-model with a first implementation. The cow scenario, as pictured in Fig. 10, is the game that was used in the MAPC from 2008 to 2010. It already contains some elements of more complex scenarios (as in traffic simulation). The task for the agents (red circles (at top) and blue circles (at the bottom)) is to herd cows to a corral in a step-based simulation. The simulated environment is a grid-like world in which the agents can move between adjacent cells, which are not blocked by obstacles (green circles). It contains two corrals – one for each team – which serve as locations where cows should be directed to (red and blue rectangles). The cows (brown circles) are controlled by a flocking algorithm. They form herds on free areas, keeping distance to obstacles and if an agent approaches, they get frightened and flee. The environment also contains fences (x-shapes), which can be opened by letting an

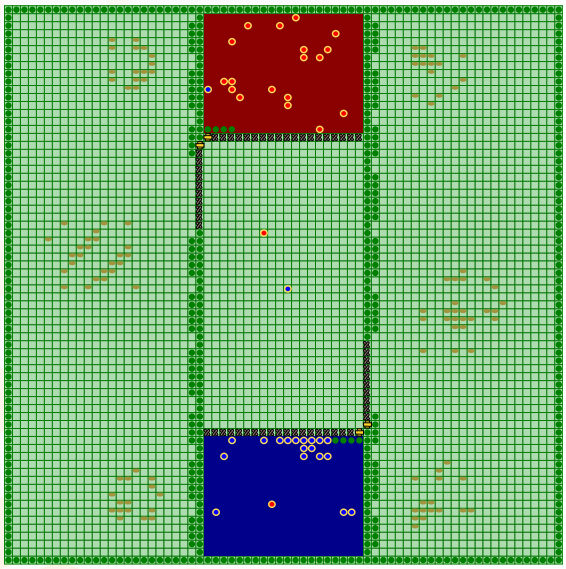


Figure 10: Visualization of the cow scenario

agent stand on a reachable cell adjacent to the button (yellow rectangles). An agent cannot open a fence and then definitely go through it. Instead it needs help from an ally. Lastly, agents only have a local view of their environment and can therefore only perceive the contents of the cells in a fixed vicinity around them. For a detailed description we refer to [8]. By using the proposed meta-model AML we can implement the cow scenario, as outlined in the following. (The corresponding Lua code can be found in the appendix of [2].)

Fig. 11 shows an overview of the mapping from necessary cow scenario elements to appropriate superclasses of the agent-model layer. The grid of the environment is implemented as an Area. Obstacles are defined by a matrix that blocks certain cells. The two corrals are subareas located inside the main area. Fences will become Artifacts. Similarly, we define a switch as an artifact that controls and changes the state (opened or closed) of a fence when getting activated. Each cow is realised by a combination of a cow entity (i.e. the "physical cow") and a respective cow agent controlling this Entity.

The cow agent part is a simple reactive agent that perceives the local environment and reacts upon it and thus, the basic Agent definition together with an Entity representing the cow within the environment are sufficient. In contrast to that, for the cowboy agents we need a more complex behaviour that facilitates coordination and cooperation. For this reason we use the BDIAgent (recall Fig. 8) class and create a cowboy Entity for each of these new (BDI) cowboy agents. Furthermore, for each Entity we create a simple MoveEffector that can be used by the entities to alter their position and a ProximitySensor providing the entities with their percepts. Additionally, we have to define the two teams by using the notion of an ObjectGroup. Finally, the Simulation creates all agents and entities, assigns them to the two teams and creates the simulation world.

To conclude, this preliminary evaluation of the structure of the AML shows that it is possible to express each aspect of the scenario using the predefined classes without the need to derive further ones from the synchronised or nonsynchronised objects. (Nonetheless, doing so still remains a possibility). Regarding the suitability of Lua, it is an extremely flexible language that comes at the cost of a certain degree of usability: any newcomer needs some time to master it. But even then, having appropriate tools and methodologies that support the modelling process is a necessity to ensure an improved workflow and reduced error-proneness.

7.2 Proof of concept (2): A traffic flow simulation

To further proceed towards a more scalable architecture, we need to tackle bigger systems involving larger numbers of more complex agents, with

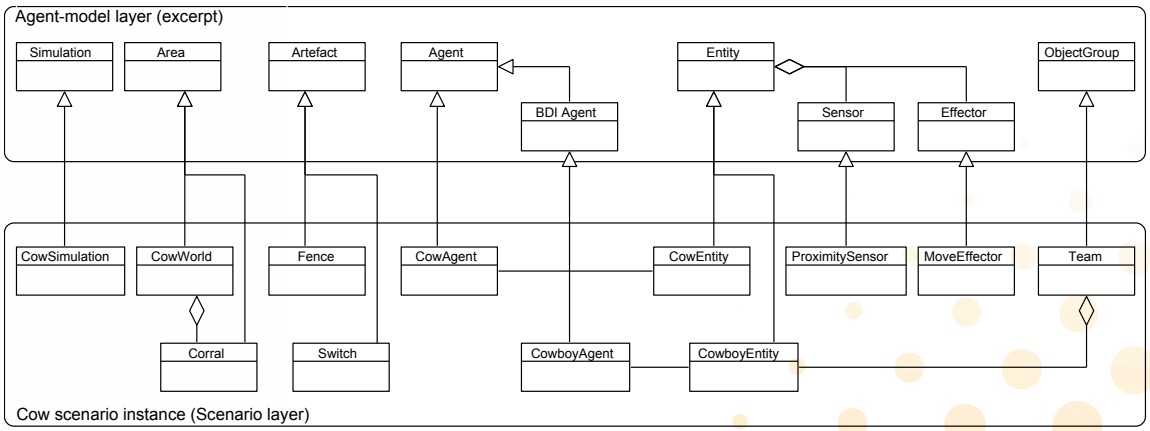


Figure 11: Cow scenario to AML mapping (UML class diagram)

richer decision-making logics (e.g. routing), acting in richer environments. To identify initial requirements on such systems, we developed MecSim, a Java-based implementation of the MASeRaTi architecture that was described previously (http://tu-c.de/mecsim). The implementation uses Jason to model agents in AgentSpeak, and combines this part of Jason with a separate, more efficient implementation of environments. Individual vehicles are represented as agents that use a Nagel-Schreckenberg driver model [26] with various extensions, such as tailgating behaviour. The driving context is very dynamic as it considers traffic jams, different speed ranges, etc. In the following, we address requirements on modeling agent behaviour with MASeRaTi; we then provide a runtime analysis of the system kernel, identifying the distribution of runtime duration among the components (methods) called in the MASeRaTi agent cycle.

7.2.1 Modeling more complex agent behaviour

Cognitive agent-based models are needed to describe the behaviour of a driver. For example, models used by traffic psychologists [12] describe requirements for modeling driver behaviours that is very similar to models based on fuzzy logic known from Artificial Intelligence. Human behaviour is composed of different layers e.g. information filtering, information processing and internal behaviour. In the current version of the MASeRaTi architecture, the following solution is used:

- The information filtering describes data processing (generate the data for each agent on the current context) in the environment.
- The information processing converts simulation data into symbolic representations and generates corresponding percepts for the agent.
- Realistic (human-like) behaviour is scripted in a rather hard-coded way in AgentSpeak. The complete source code of the MecSim implementation can be found at https://mecdev.rz-housing. tu-clausthal.de/gitlab/philipp.kraus/mecsim/blob/master/src/

Fig. 12 illustrates a screenshot of an agent-based traffic flow scenario based on the well-known

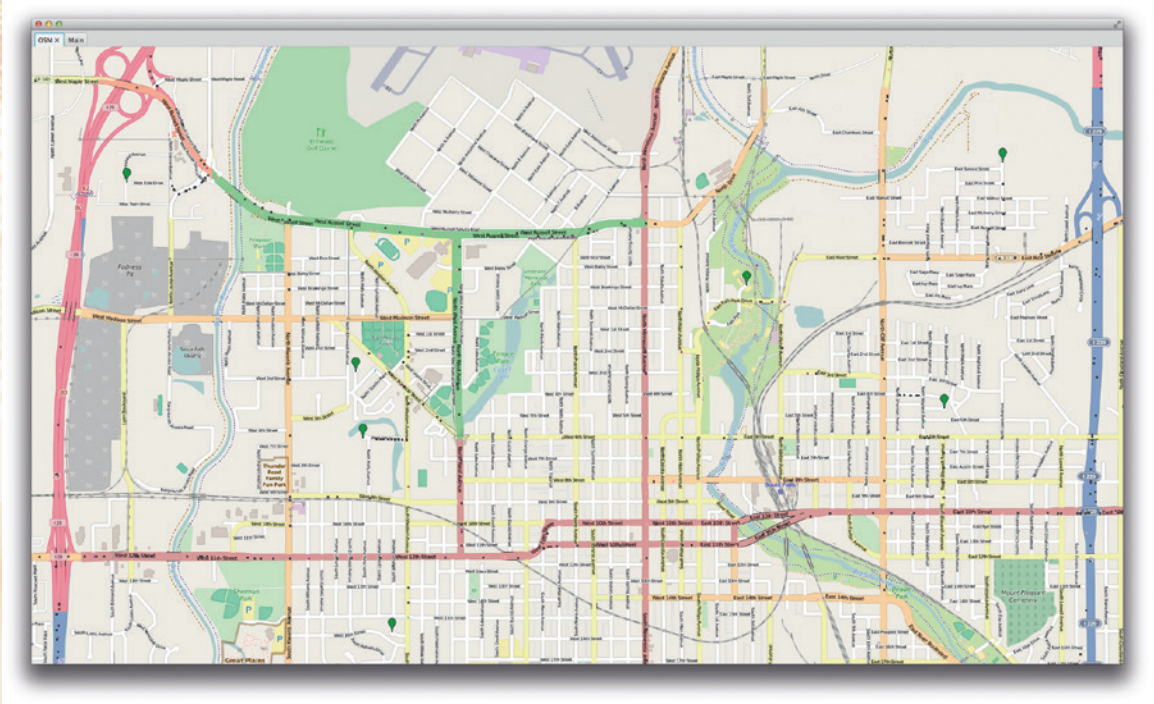


Figure 12: Screenshot of MecSim traffic simulation (Sioux Falls data set)

Sioux Falls benchmark data set. We compared two variants:

In the first variant, the vehicle logic (the "mind") was implemented using native Jason as mentioned above, i.e. where each vehicle is represented by a Jason agent. In the second, a similar logic is encoded in the MASeRaTi environment. When using native Jason code, simulation performance starts seriously degrading once the number of vehicles starts exceeding 20. When using the "non-Jason " implementation of vehicle logic in the MASeRaTi, we can manage about 100 000 vehicles. The main reason for this is that the Jason agent cycle is very resource-consuming, resulting in a slow and non-scalable runtime performance: the simulation slows down drastically as agents are generated.

7.2.2 Runtime analysis of MASeRaTi

The algorithms for perception generation must operate under the worst-case assumption that agents have global perception (or at least: large perception ranges) and will need to access updates of large numbers of percepts. This may create immense data volumes. Fig. 13 shows the results of a runtime analysis of the system. In particular, we record the time spent on different methods called in the simulation (from left to right): (1) execution of driving behaviour (Nagel-Schreckenberg); (2) calculating MAS consistency value; (3) calculating a similarity / metric value of agent belief bases; (4) executing the agent-cycle; (5) generating simulation objects; (6) looping over different simulation architecture layers; and (7) loop over simulation objects

The time consumed by each call is visualized with a box-plot (histogram) in milliseconds in a log-scale. We can conclude that the environment ("Generator" in the blue box above), performs very well. Also the agent-cycle itself is run very efficiently (uses even less time than the environment) and indeed scales up if the agent has to calculate more plans and beliefs. The dashed line on each box shows the method runtime to get information about skewness (median average ratio). However, in each step of an agent cycle, only one action is called, so the reaction time of the agents is too slow for simulating correct traffic behaviour, e.g. the agent perceives the predecessor only without adjusting the speed. The adjusting runs some cycle steps later, so we get an undefined / wrong behaviour of our simulation.

7.3 Experimental analysis of scalability

7.3.1 Counting Scenario

To address research question RQ4, we first benchmark a simple counting scenario to establish a

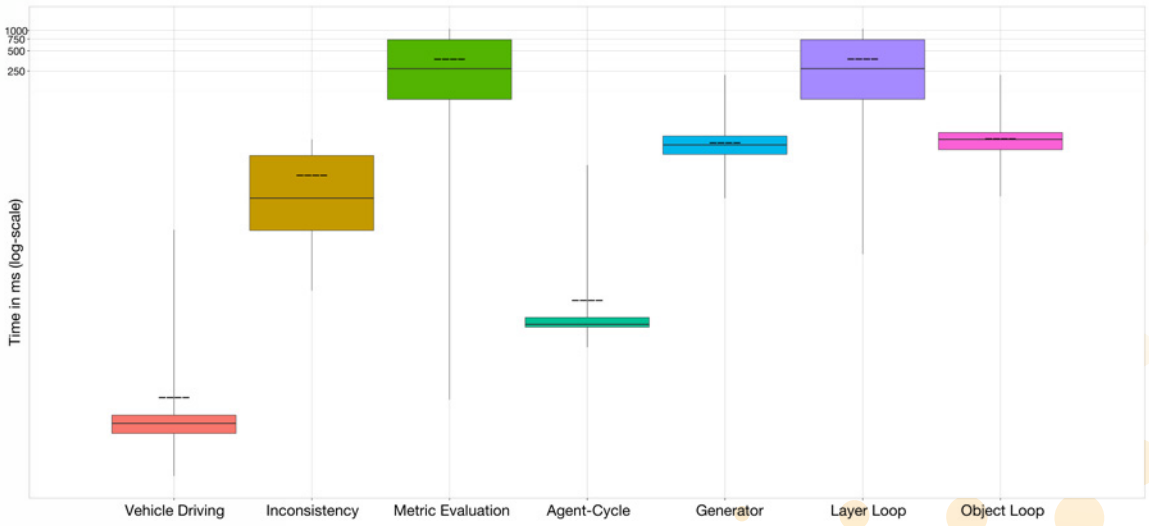


Figure 13: Runtime analysis of the traffic scenario

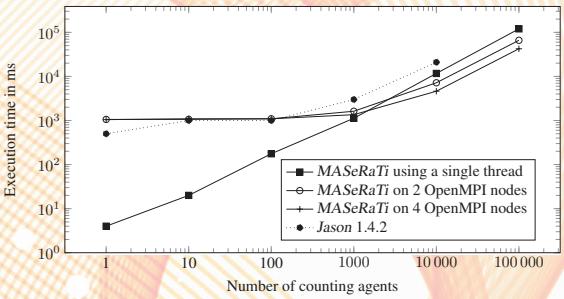


Figure 14: Execution time of the counting scenario for increasing numbers of agents. Jason 1.4.2 does not allow executing the scenario for 100 000 agents.

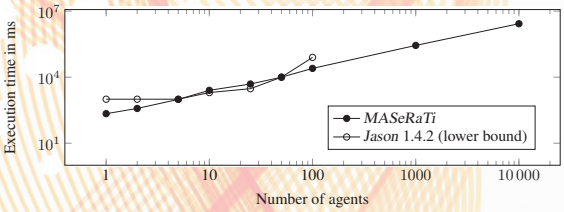


Figure 15: Execution times for 200 steps of the simplified cow scenario with 100 cows and an increasing number of agents using AgentSpeak.

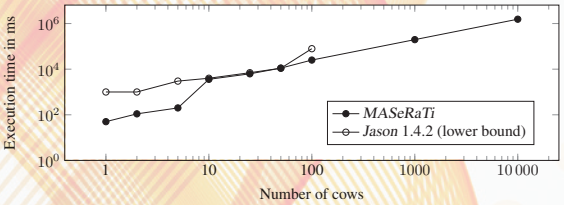


Figure 16: Execution times for 200 steps of the simplified cow scenario with an increasing number of cows and 100 AgentSpeak agents.

performance baseline for systems that involve both computation and communication. (All experiments conducted here were run on Debian Stretch using an Intel Core i7-4800MQ CPU @ 4 x 2.70 GHz and 8 GB RAM.) In the first phase agents count until they reach an upper bound. In order to determine when the simulation can be shut down, there is a second phase: Agents are assembled in a ring. When they have finished counting, they send a token to their successors. Tokens are only passed on by agents that have also finished. Thus, as soon as a token reaches its original sender, all agents must have finished counting and the simulation is complete.

Fig. 14 shows the almost linear scalability of our platform in this scenario. Furthermore OpenMPI can be initialized in constant time to parallelize the computation. This pays off at around 1000 agents. Adding more OpenMPI nodes first brings down the runtime almost linearly, but this effect is limited by the sequential structure of the ring.

7.3.2 Cows scenario

In the counting scenario agents are very simple rather than cognitive; in addition, they do not have to interact with an external environment. To address RQ5, we therefore show a second benchmark where agents are somewhat more complex and communication with the environment is essential. We use a simplified version of the cow scenario to evaluate our AgentSpeak interpreter, because the prototype does not yet come with extensive standard libraries for the more complex planning and communication problems involved in the full cow scenario. In the simplified cow scenario there are no fences or obstacles. At each simulation step, cows move randomly to an adjacent unoccupied field. Whenever a cow moves, the runtime environment updates the agent belief bases and triggers change events.

The experiments in Fig. 15 and Fig. 16 confirm that the execution time of a fixed number of steps scales almost linearly in the number of agents. To put the concrete execution times into perspective we run the same AgentSpeak program on the Jason platform. In order to guarantee a fair comparison, we underestimate the number of calls to the C++ environment. Therefore the given execution times for Jason constitute a lower bound. The result shows that Jason 1.4.2 cannot successfully complete the simulation for scenarios involving 100 or more cows and 1000 or more agents. The same is true for a simulation with 100 agents and 1000 cows (or more).

8 Conclusion and Outlook

In the DESIM project, we carried out important work towards a distributed runtime platform for

multiagent simulation. The main contributions were: (1) an analysis of the state of the art in agent-based simulation platforms, leading to a set of requirements to be imposed on a simulation platform, focusing on runtime scalability and efficient memory management; (2) the proposal of a novel architecture and design of the MASeRaTi simulation platform, bringing together a robust and highly efficient agent kernel (written in Lua) with a BDI agent interpreter including multiagent concepts such as communication and computational norms; and (3) an initial proof of concept realization featuring a simple application scenario.

In terms of the research questions, our insights can be summarized as follows. Concerning RQ1 and RQ2, we believe the meta-model is versatile enough to model typical scenarios and to support typical multiagent features. A MSc thesis describing the meta model in great detail has been completed [1]. The model allows easily for future extensions and is flexible in this respect. The MASeRaTi architecture showed to be flexible enough to do a quite complex traffic flow simulation. Finally, as the results illustrate (we also refer to [16]), scalability in the number of agents is established, both for a generic (counting) scenario and for cognitive agent scenario (a simplified cow and cowboys scenario from MAPC).

Thus, the project has created a solid baseline for agent-based simulation, which can be used, but which also needs to be extended and improved. Issues such as optimisation of the scheduler and the caching mechanisms sketched in the appendix of [2] will be subject for future research. As the ATSim platform can deal with a few thousand (vehicle) agents, we aim MASeRaTi to ultimately scale up to one million agents of comparable complexity (corresponding to the micro-simulation of multimodal traffic in a large city, including public transport, cars, pedestrians, city logistics, and infrastructure). We expect the platform to be used in further projects, such as the SWZ project ASimOV[1], which started in summer 2016, and upcoming versions of the multiagent programming contest.

Also, more systematic experimental evaluation will be carried out using more sophisticated and much larger traffic simulation scenarios. Simulation results obtained this way can be compared to the performance of other simulation frameworks using benchmark data; scalability can also be described by varying certain parameters (e.g. number / complexity of agents) and investigating the resulting degradation of performance. An idea for evaluating our optimisation approach (and in particular the adaptive mechanism for allocating agents to nodes of the runtime system) is the following: By modelling agents' preferences, capabilities, and interactions, a certain structure is imposed on the resulting MAS. We intend to evaluate the degree to which this structure can be mapped to the allocation of agents to the nodes of the distributed runtime system, by a (domain-independent!) entropy measure.

As regards the transition from state-of-the-art approaches such as Jason (and also from the current status of MASeRaTi) to scalable platforms that are able to support cognitive agents models at large scale, there is still a long way to go. Jason defines a logical programming language with no support of fuzzy or stochastic models. This is an important area of future research. Data processing and converting also create a performance problems, because each agent needs to be updated about the current status of the environment in its range of perception. To achieve an efficient access of a large number of agents to the state of the environment, an efficient perception structure is needed. The state-of-the-art event-based perception structures are not useful in such complex simulations (too many datasets are generated). A key improvement we will work on is a refactored agent cycle which can run faster and in parallel, to allow agents to react in a faster way. Given the three initial objectives, our focus has been on the first two: scalability and efficient memory management, whereas we only touched the third, modelling. Here, one avenue of research is to develop appropriate modelling tools to support the MASeRaTi architecture.

References

[1] T. Ahlbrecht. Modular modelling support for multiagent simulation. Department of Informatics, Clausthal University of Technology, July 2016. Master thesis.

1 ASimOV stands for Agentenbasierte Simulation des Passagierverhaltens zur Optimierung des Verspätungsmanagements im Bahnverkehr. It is a joint project with the group of Prof. Anita Schöbel from Göttingen University.

[2] T. Ahlbrecht, J. Dix, M. Köster, P. Kraus, and J. P. Müller. A scalable runtime platform for multiagent-based simulation. Technical Report Ifl-14-02, TU Clausthal, February 2014.

[3] T. Ahlbrecht, J. Dix, M. Köster, P. Kraus, and J. P. Müller. A scalable runtime platform for multiagent-based simulation. In *International Workshop on Engineering Multi-Agent Systems*, pp. 81-102. Springer International Publishing, 2014.

[4] T. Ahlbrecht, J. Dix, N. Fiekas, M. Köster, P. Kraus, and J. P. Müller. An architecture for scalable simulation of systems of cognitive agents. In *International Journal of Agent-Oriented Software Engineering*. Inderscience Publishers, accepted for publication 2016.

[5] T. Ahlbrecht, J. Dix, M. Köster, and F. Schlesinger. Multi-Agent Programming Contest 2013. In M. Cossentino, A. El Fallah Seghrouchni, and M. Winikoff, editors, Engineering MultiAgent Systems, volume 8245 of Lecture Notes in Computer Science, pages 292–318. Springer Berlin Heidelberg, 2013.

[6] ALC Bazzan and F. Klügl. Agent-based Modeling and Simulation. AI Magazine, 33(3):29–40, 2013.

[7] T. Behrens. Towards Building Blocks for Agent-Oriented Programming. PhD thesis, Clausthal University of Technology, 2012.

[8] T. Behrens, M. Dastani, J. Dix, M. Köster, and P. Novák. The Multi-Agent Programming Contest from 2005-2010. Annals of Mathematics and Artificial Intelligence, 59:277–311, 2010.

[9] O. Boissier, M. Colombetti, M. Luck, JJ. Meyer, and A. Polleres. Norms, organizations, and semantics. The Knowledge Engineering Review, 28:107– 116, 2013.

[10] RH Bordini, M. Dastani, J. Dix, and A. El Fallah Seghrouchni, editors. Multi-Agent Programming: Languages, Tools and Applications. Springer, Berlin, 2009.

[11] RH Bordini, JF Hübner, and MJ Wooldridge. Programming multi-agent systems in AgentSpeak using Jason. Wiley & Sons, 2007.

[12] H. Bubb, K. Bengler, RE Grünen, and M. Vollrath. Automobil-Ergonomie. Springer-Verlag, 2015.

[13] V. Chu, J. Görmer, and JP Müller. ATSim: Combining AIMSUN and Jade for agent-based traffic simulation. In Proceedings of the 14th Conference of the Spanish Association for Artificial Intelligence (CAEPIA), volume 1. AEPIA, 2011. Electronic Proceedings.

[14] G. Armitage, M. Claypool, and P. Branch. Networking and online games: understanding and engineering multiplayer Internet games. John Wiley & Sons, 2006.

[15] Á. Fernández Díaz, C. Benac Earle, and LA Fredlund. eJason: An implementation of Jason in Erlang. In M. Dastani, JF. Hübner, and B. Logan, editors, Programming Multi-Agent Systems, volume 7837 of Lecture Notes in Computer Science, pages 1–16. Springer Berlin Heidelberg, 2013.

[16] N. Fiekas. Ein leichtgewichtiger AgentSpeak Interpreter für C++, October 2015. Department of Informatics, Clausthal University of Technology. Bachelor thesis.

[17] M. Fiosins, J. Fiosina, JP Müller, and J. Görmer. Reconciling Strategic and Tactical Decision Making in Agent-oriented Simulation of Vehicles in Urban Traffic. In Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques, SIMUTools '11, pages 144–151, ICST, Brussels, Belgium, 2011. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[18] M. Fiosins, J. Fiosina, JP Müller, and J. Görmer. AgentBased Integrated Decision Making for Autonomous Vehicles in Urban Traffic. In Y. Demazeau, M. Pechoucek, J. Corchado, and J. Perez, editors, Advances on Practical Applications of Agents and Multiagent Systems, volume 88 of Advances in Intelligent and Soft Computing, pages 173–178. Springer Berlin / Heidelberg, 2011.

[19] MP Georgeff and AL Lansky. Reactive reasoning and planning. In Volume 2 of Proc. of the 6th National Conference on Artificial Intelligence, pp. 677—682, AAAI Press, 1987.

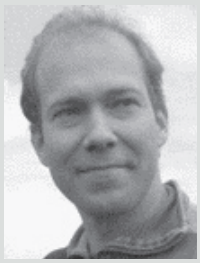
[20] J. Görmer and JP Müller. Group Coordination for Agent-Oriented Urban Traffic MA. gement. In Yves Demazeau et al., editor, Advances on Practical Applications of Agents and Multi-Agent Systems (Proc. of PAAMS 2012), volume 155 of Advances in Soft

Computing, pages 245–248. Springer-Verlag, 2012.
[21] MD Hill and MR Marty. Amdahl's Law in the Multicore Era. IEEE Computer, 41(7):33–38, July 2008.
[22] M. Huhn, JP Müller, J. Görmer, G. Homoceanu, N.-T. Le, L. Märtin, C. Mumme, C. Schulz, N. Pinkwart, and C. Müller-Schloer. Autonomous agents in organized localities regulated by institutions. In Digital Ecosystems and Technologies Conference (DEST), 2011 Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies, pages 54–61, May 2011.
[23] R. Ierusalimschy. Programming with multiple paradigms in Lua. Functional and Constraint Logic Programming, pages 1–12, Springer Berlin Heidelberg, 2009.
[24] R. Ierusalimschy Luiz Henrique de Figueiredo, Waldemar Celes. Lua Programming Gems. Lua. org, ISBN 978-85-903798-4-3. Roberto Ierusalimschy, 2008.
[25] JP Müller. The Design of Intelligent Agents, volume 1177 of Lecture Notes in Artificial Intelligence. Springer-Verlag, 1996.
[26] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. Journal de Physique I, 2(12):2221–2229, December 1992.
[27] Yoav Shoham. Agent-Oriented Programming. Artificial Intelligence, 60(1):51–92, 1993.
[28] MP Singh. Norms as a basis for governing sociotechnical systems. ACM Trans. Intell. Syst. Technol., 5(1):21:1–21:23, January 2014.
[29] G. Weiss, editor. Multiagent systems. MIT-Press, 2013.
[30] MJ Wooldridge and NR Jennings. Pitfalls of agent-oriented development. In Proceedings of the Second International Conference on Autonomous Agents, AGENTS '98, pages 385–391, New York, NY, USA, 1998. ACM.
[31] AS Rao and MP Georgeff. „BDI agents: From theory to practice" In V. Lesser, ed., Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS-95), pp. 312—319, AAAI Press, 1995.

Project data

The project was funded from SWZ with one TV-L E13 staff position from July 2013 to June 2016 at the site Clausthal. Involved scientists are:

Prof. Dr. Jürgen Dix
Research Group Computational Intelligence
Department of Informatics
Clausthal University of Technology



Prof. Dr. Jörg P. Müller
Business Information Technology Unit
Department of Informatics
Clausthal University of Technology

Towards Multi-Level-Simulation using Dynamic Cloud Environments

Stefan H. A. Wittek, Michael Göttsche, Andreas Rausch and Jens Grabowski

1 Introduction

Cyber physical systems (CPS) consist of numerous physical and software components. Autonomous cars and automated production facilities are examples of such systems. The engineering of CPS is a difficult task due to the complexity of these systems.

In engineering, simulation has become a core method. The complexity of a system is abstracted into an executable model that allows us to evaluate designs without the need of building physical prototypes. This reduces the costs and effort involved in this task.

Applying simulation to CPS provides numerous chances. Aside from the possible reduction of prototyping effort, the product can be improved and its costs can be lowered. Real-time simulations can be employed at runtime to infer from a few measurement points to numerous virtual sensors located in between these physical sensors. This allows to reduce the amount and quality of sensory used, which in turn leads to efficient designs. The cost of the system can also be lowered by allowing deviation in the physical part of the CPS. If these deviations (i.e. the bending of a robot arm due to the mass it is lifting) is well known through simulation, it can be compensated using the software part of the system. This compensation in turn can be evaluated in simulation.

Nowadays the simulation of complex systems is done according to specific simulation questions. Scenes are modelled in a particular domain containing only one area of the system using a specific modelling technique. An example of this is the thermal behaviour of the cable in the shaft of a lift using the finite element method, to answer the question if the lengthening of the cable is beyond a certain threshold. By doing this, the interdependency between these scenes is lost, which is problematic in the case of complex CPS.

Simulation for CPS must be both: Holistic enough to capture the dependencies between its components, but only as complex as feasible, regarding modelling effort and computation times. Figure 1 describes this Tradeoff in more detail. In the figure, simulation Scenes are depicted as dots and arranged according to there complexity and holism (x- and y-axis). Without considering resources available, a simulation of the hole system that is modeled, using the most detailed technic available, provides the potentially most accurate results. Under restricted resources and for larger systems, such a simulation is. Instead, the hole system can only be simulated using a coarse simulation (A). If more detail is needed to answer specific questions, only a smaller portion of the system can be considered, ultimately leading to a very small part of the system using the most detailed technic (B). Such a zoom-in could be done with any part of the system, i.e. (C). But, since this scenes are simulated independently, interdependencies between them is lost. To acquire a more holistic view, these scene can be connected directly using a methodology or by building interfaces between these scenes and

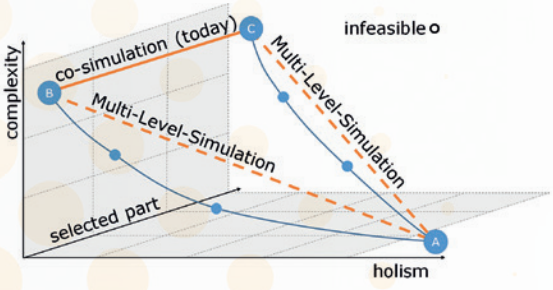


Figure 1: Tradeoff between complexity and holism in simulation.

employing co-simulation. Both approaches are difficult, expensive and often only valid for particular instances of these scenes.

Within the research project Multi-Level-Simulation we aim for a simulation methodology that is efficient regarding complexity. We simulate the CPS on multiple levels of abstraction. On a coarse level, the whole CPS is modelled using a simple semantic (A). To answer questions that require more complex simulations, only relevant areas of the system are chosen to be co-simulated on more detailed levels (B,C). Which area is relevant may vary during the course of the simulation. To reflect this, the connected detail simulations may change dynamically.

Complex simulations are resource intensive and need proper computation infrastructures. If the simulation is dynamic as proposed, this infrastructure needs to be dynamic as well.

In a traditional computing infrastructure setting, the resources have to be designed for the worst case, i.e. to satisfy the requirements of the most resource-intensive possible simulation run in order to generate its results in an acceptable time frame. This poses no problem for simulations with homogeneous requirements. However, for cases where the resource utilization is highly heterogeneous, as in the case of our simulation methodology, the computing infrastructure that accommodates the worst case is vastly oversized for the average simulation, resulting in a low overall utilization and thus cost inefficiency.

A better choice for the computing infrastructure of this use case is one that allows to reserve and release resources on-demand so as to dynamically match the requirements of the simulation. The Cloud Computing paradigm that has emerged and matured in the last few years matches this need. Thus, we will propose a framework for deploying simulations on a Cloud platform in order to achieve a timely as well as cost-efficient solution.

2 Related Research

In this section we will describe research related to our project. The co-simulation of heterogeneous systems is the aim of a variety of tools and frameworks. A selection of these works is presented. The idea to simulate systems on different levels of abstraction can be found in several approaches. Some focus on certain application domains while others aim to provide a general framework. We will discuss both directions. Cloud infrastructures in general and the deployment of simulation into this infrastructure are an active research field. We will provide a brief overview and discuss known approaches in this field.

A variety of works focus on the co-simulation of different simulations tools. Examples of this are the High Level Architecture specification for simulation interoperability [4], the Functional Mockup Interface standard for model exchange and co-simulation [2] and the Mosaik Simulation API [15]. Another approach is to integrate different simulation semantics into a single tool. The Ptolemy project is an example for this approach [6]. All these works aim towards a holistic simulation of the system under investigation. The simulation of different abstraction levels is only addressed in terms of tool integration. The task to provide proper interfaces to connect simulation on different levels has to be done by the modeller.

Much effort is put into approaches that provide such concepts for specific domains such as material flows [5][10], traffic [3] or agent based behavior simulation. They center on the dynamic switching of abstraction levels of model parts at runtime. To do so, explicit mappings between the states of different levels are provided. These mappings are tightly bound to the domain and the simulations they connect and are not designed to be generalizable.

Some research is conducted investigating more general concepts for the problem. The approach of Dynamic Component Substitution describes a co-simulation as a set of connected software components communication through given interfaces [13]. Switching a part of the simulation to a more detailed version corresponds to substituting one such component with another. Both components are required to have the exactly same interfaces. This is a critical limitation. If the components are situated on different levels of abstraction, it is plausible to expect the same for their interfaces. Multi Resolution Entities [14] define a mapping that is used to synchronize the simulation state

on different levels. These mappings are defined as invertible to use them in both directions. This requirement is only meet, if no information is lost mapping a detailed state to a more coarse state, which does not apply in general, as we will describe in Section 3. The concept of Multi Resolution Modelling Space introduces adapters between the interfaces and several mappings between the states of simulations on different levels [9]. However the problem of information loss is not addressed in this approach.

Our approach of Multi-Level-Simulation is different from these approaches, because it does not force the engineer to tailor the coarse level simulations into components connected by interfaces. We consider this approach as too inflexible. The coarse level can be modeled with no dependency on the detailed level. In fact, even cutting arbitrary parts out of existing coarse level simulations to be linked to a detailed level is possible. The detailed simulations are linked into a single simulation on the coarse level using only a state synchronization mechanism. This mechanism also addresses the problem of information loss.
The dynamic deployment of the simulation infrastructure addresses a novel problem with regards to the scaling question. While previous work addresses the scaling of long-running processes, we are more concerned with starting and stopping adequately sized compute resources on-demand depending on the launched and cancelled simulation.

The topic of scaling computing infrastructures in Cloud environments for elastic applications has received a lot of attention. A predestined use case is that of scalable web applications, but more recently the research has shifted to scientific applications. In a thorough review [11] the authors give an overview of the various auto-scaling techniques that have been addressed so far. Our own previous work has dealt with the question of acquiring compute resources and automating simulation deployment and execution for a statically sized infrastructure [8].

Research has also been done in the field of provisioning infrastructure for traditional simulation workflows. One proposal describes a service-oriented binding strategy including a middleware architecture for deploying simulation components [16]. Recently, the TOSCA modelling standard has received more attention as a possibility for automating simulation workflows in a Cloud environment in a model-based way [12]. This has also been employed in research that proposes a domain-model-based deployment and execution framework for scientific applications [7].

3 Multi-Level-Simulation

To describe our approach of Multi-Level-Simulation in more detail, we consider the example of a lift. Figure 2 shows the structure of this example. (A) On the coarse level it consists of a simulation modelling the structure of the lift and a lift program. The structure consists of a shaft in which a cabin can move. The cabin is rigged to a cable. The weight of the cabin (w) is altered when it stops at one of the exits. A motor manipulates the length of the cable (l). The program simulation is connected to the structure and handles the speed and direction of the motor. In this setup all parts of the structure are modelled as rigid bodies. The program has no sensor for l and positions the cabin only indirectly using the last position of the cabin and a timer. On this level, realistic scenarios

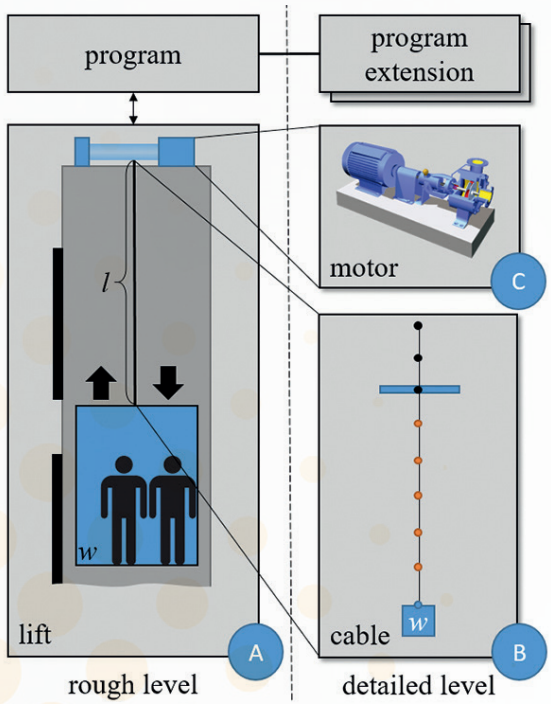


Figure 2 Structure of the elevator example.

of use are modelled. An example of this would be a whole day cycle of an office building. Most persons want to go up in the morning and down in the evening. The simulation on this level is fast.

(B) During the development of the lift and its program the engineers want to investigate, if the stretching of the cable caused by the weight of the cabin and the aging of the cable will lead to a wrong positioning of the cabin. To do this, a detailed but computationally intensive simulation of the cable is activated. This simulation is stateful to reflect the aging of the cable. Only parts of the cable that are stretched in a particular time step age. If the misplacement is a problem, the engineer has to implement an extension to deal with the phenomena in the program.

(C) After this, the dynamics of the cabin are investigated closely. A computationally intense simulation of the motor is activated. This simulation models the acceleration of the motor and allows to precisely determine the travel times of the lift. The simulation is stateful to model the heating of the motor which influences acceleration. Because the stretching of the cable is considered irrelevant for this question, the cable simulation and the corresponding program extension are deactivated. Because the program on the coarse level does not account for the acceleration when calculating the timers, a corresponding extension must be implemented and linked to the program.

Note that the program finally deployed needs to include both extensions.

In both cases, parts of the lift are simulated on two levels at the same time. This leads to the challenge of maintaining the consistence between the states of both levels. If for example in (A) *l* is increased by 0.1m, all elements of the cable in (B) must be placed 0.1m lower. If in (B) the cable is stretched by 10%, displacing the lowest point from -3m to -3.3m, *l* must be set from 3m to 3.3m in (A).

Figure 3 shows a schematic overview of the example. Each simulation consists of two parts. The state of a simulation is defined as a valuation of a fixed set of attributes. The behaviour of a simulation is defined as a mapping which has this state as input and produces a new state as output. Applying the

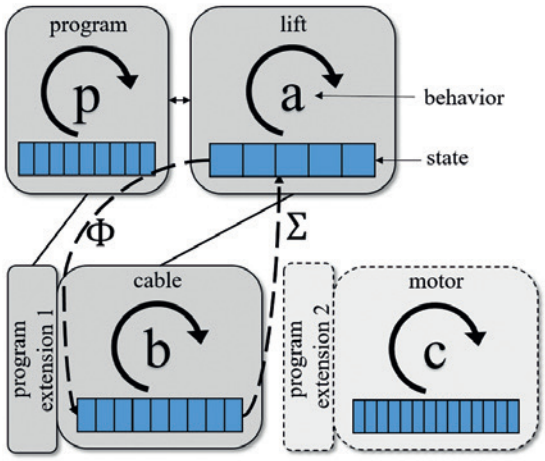


Figure 3: The problem of state synchronisation.

behaviour to a state s_0 of a simulation leads to the state s_1. This corresponds to a step in the simulation. For all simulations the time Δt elapsing in one step is the same. The coarse simulation of the lift is linked to a number of detailed simulations. Note that in the lift example only one of these simulations is connected in a particular simulation run.

Because the simulation models (i.e. the cable) are different, the attributes valuated in a state are different. The states need to be converted between the simulations. This is done using the state mappings Φ and Σ. At the current state of our work, these mappings are given by the modeller. Σ maps the detailed level state to the coarse level state. It is typically not reversible, because information is lost. Referring to the lift example, there are a number of different positions and age levels of the cable elements that map to the same *l*. Φ maps the coarse level state to the detailed level. In the example, Φ restores the position of the cable elements using only *l*. To do so, Φ has to choose among a possible infinite set of states that are mapped to *l* by Σ. To account for this problem, we propose Φ as a mapping of the coarse state and the last state of the detailed state.

Figure 4 gives an overview of the execution of the example. Note that in general changes on different levels accrue concurrently, regarding to simulation time.

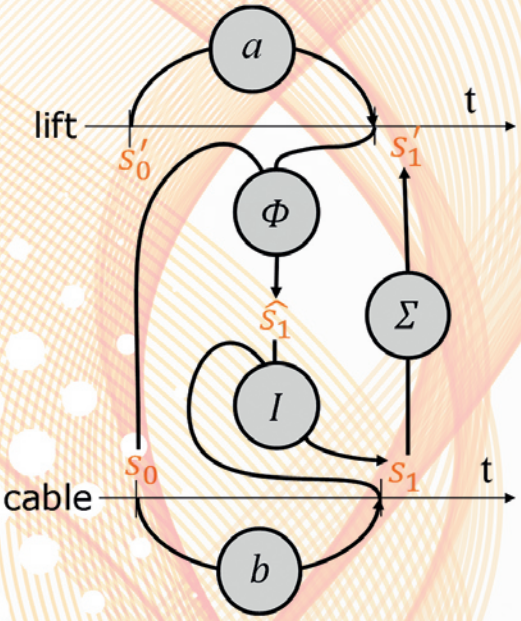


Figure 4: Execution of the lift example.

Let us consider the lift simulation starts with the initial state s_0' and the cable simulation with the initial state s_0. The states are chosen so that $\Sigma(s_0)=s_0'$. This can be seen as that s_0 and s_0' represent 'the same' on both levels. Now both simulations step using the behaviour functions *a* and *b*. The cable simulation ages a number of cable elements, stretching the cable by 0.1m leading to the state s_1. In the same time step, the lift simulation unwinds the cable by 0.2m according to the initial speed of the motor, leading to the state s_1'. Converting s_1' to a state of the cable simulation using Φ results in an intermediate state $\hat{s_1}$. This state is in conflict to s_1 which was calculated using the behaviour *b* of the cable simulation. Simply overwriting s_1 using $\hat{s_1}$ would annihilate the unwinding of the cable. To avoid this, an integrator function *I* must be employed to merge the two states. The resulting state contains both changes. Using Σ on this state leads to an integrated state of the lift simulation that contains again both changes. This state finally overwrites the state of the lift.

4 Cloud-Deployment

In this section, we describe aspects of deploying Multi-Level-Simulations in Cloud environments. Section 4.1 introduces elasticity aspects of Multi-Level-Simulations. In Section 4.2 we outline our plan for a dynamic deployment strategy. Finally, in Section 4.3, we present an initial deployment of our prototype application.

4.1 Elasticity in Multi-Level-Simulations

Multi-Level-Simulations are characterized by their variable resource requirements depending on the simulation question. The fluctuations result from the dynamic nature of Multi-Level-Simulations on two different layers:

System level. A Multi-Level-Simulation consists of multiple components of which not all are operating at the same time. The entirety of these components forms the system level. Specifically, in our example the system consists of program, lift, cable, motor and the communication component. The required components vary (a) between different simulation runs depending on the simulation objective and (b) within the same simulation run when components have finished their simulation.

Component level. The lower level's complexity of a simulation is dependent on the simulation parameters. For example, the cable is one component of our prototype. While a particular component's computational requirements may be low for one run, it can be higher for another.

In such cases, Cloud Computing can help in establishing a dynamic infrastructure to scale the resources in accordance with the simulations' demand. Ideally, at any point only the required computing resources will be allocated. Too few allocated resources ("underprovisioning") will lead either to longer runtimes or even abortion of the simulation. Too many resources ("overprovisioning"), on the other hand, allow for a timely execution of the simulation, but at the cost of dissipation.

4.2 Dynamic Deployment

Contrarily to a static deployment where a fixed set of resources is allocated at the start of the simulation and remains allocated throughout its lifespan, a dynamic deployment is not finished once the required resources have been allocated and the components have been installed and launched on

it. Instead, a framework that fulfils the three following tasks needs to be put in place:

Monitoring. In order to perform runtime adaptations, the framework needs to collect information about the simulation resources and components. Specifically, the utilization of the resources is important to support a judgement.

Reasoning. Using rules and the data collected by the monitor, the framework has to perform reasoning about infrastructure adaptation.

Infrastructure Adaptation. The framework needs to adapt the infrastructure to the simulation requirements in both directions, i.e. by reducing or increasing its size. Also, it needs to adapt the deployment accordingly.

The reasoning pipeline is depicted in Figure 5. The computational complexity of a simulation depends on its model as well as its execution parameters. By combining this with information about the resource usage it is possible to build an execution history that serves as input for the reasoning engine for predicting a suitable deployment for future simulation runs. For this, we employ statistical methods that create rules which are iteratively refined by evaluating their accuracy. For the monitoring and infrastructure adaptation tasks we intend to implement an integrated solution following a models@runtime (Aßmann et al., 2015) approach and the tools we employed for

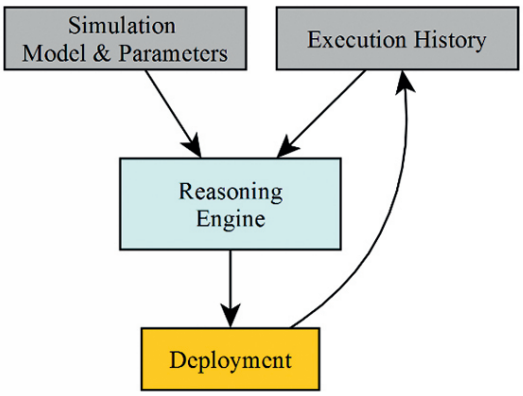


Figure 5:Reasoning Pipeline.

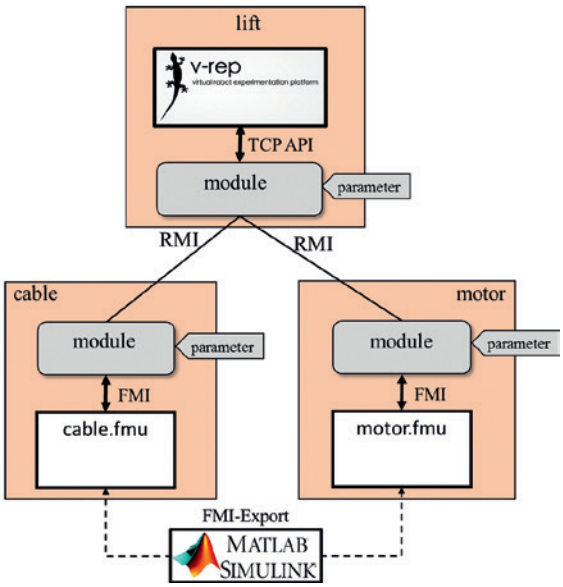


Figure 6: Schematic View of the Prototype

the initial deployment. This approach will allow a strong decoupling of the adaptation logic from the technical steps necessary for enacting deployment changes.

5 Multi-Level-Simulation platform

This section describes the architecture of the Multi-Level-Simulation platform prototype which will implemented within the project. The purpose of the platform is to provide a prove of concept for the Multi-Level-Simulation and the Dynamic Cloud-Deployment. Figure 6 shows a schematic view of the Prototype. For the sack of convenience the depicted simulation setup matches the elevator example.

For each simulation in the example, exists a module instance. The modules are written in Java programming language. They provides an interface to the state and behavior of the actual simulation. Each module is initialized using a set of parameters. This parameter are chosen by the user and may have significant impact of the resources used by the simulation. The modules form a hierarchy according to the simulation layer. Note that this architecture could easily extended

to more than two layers. In the example, the lift module is connected to the cable and the motor module. This connection is realized using the Java Remote Method Invocation Protocol. The lift module orchestrates the execution of the Multi-Level-Simulation and performance the state conversions and integration steps described in Figure 4.

Each module capsulate a simulation model. This model is executed within its original simulation environment. Or the model is executed directly using a compiled binary.

In the example, the lift is modeled using the v-rep simulator[1]. It provides a fast and easy to use environment for the simulation of factory automation and robots. It not only simulates the physical bodys involved in this systems but also the software controlling the behavior of that system. V-rep is connected to the module using its TCP API. The API allows to read and manipulate the state of every element of the simulation scene and to control the execution of the simulation stepwise. The cable and motor are modeled using Matlab Simulink. They are compiled according to the Functional Mockup Interface[2] and wrapped into a fmu file. The module uses the javaFMI[2] library to read the fmu and to execute it.

The describe example may appear simplistic, but it was chosen to contain relevant technical challenges involved in the development of a co-simulation platform. The prototype is platform independent. Java as a programming language, the simulation tools and the fmu technology are at the minimum executable on Windows and Unix systems. V-Rep was chosen because it suits the simulation of a range of cyber physical systems on a coarse level. Matlab Simulink is a central tool used in all kinds of domains for simulations based on differential algebraic equation. It is although used extensively in model based software development in embedded systems. The functional Mockup interface is not limited to Simulink but supported by a variety of simulation tools.

5.1 Initial Deployment

As an initial approach for a Cloud deployment of our prototype application we chose a static setup as depicted in Figure 7.

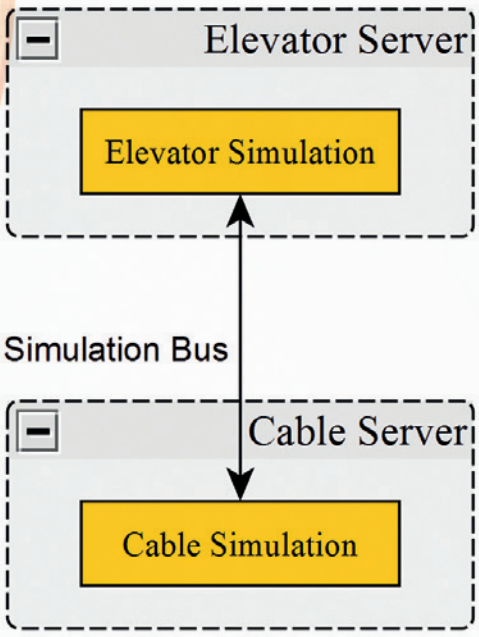


Figure 7: Initial Deployment.

In this setup, each of the two simulation components is served by its own virtual machine in the Cloud. The components exchange status via RMI and therefore only require a shared network for communication.

On a technical level, our static deployment is model-based and agnostic to a particular Cloud platform. Concretely, we use the TOSCA-based Cloud orchestration platform Cloudify[3] as well as the Software Configuration Management tool Ansible[4]. This allows for a deployment of the simulation on the variety of Cloud platforms supported by Cloudify as well as on different operating systems as supported by Ansible.

We intend to employ the technologies used in the initial deployment for the dynamic deployment.

6 Status and Future Work

To get first insights on our concept of Multi-Level-Simulation and the corresponding Cloud deployment mechanism, we build a prototype of the described lift example. The prototype consists of

1 http://www.coppeliarobotics.com
2 Online: https://bitbucket.org/siani/javafmi/wiki/Home
3 Online: http://getcloudify.org/
4 Online: https://www.ansible.com/

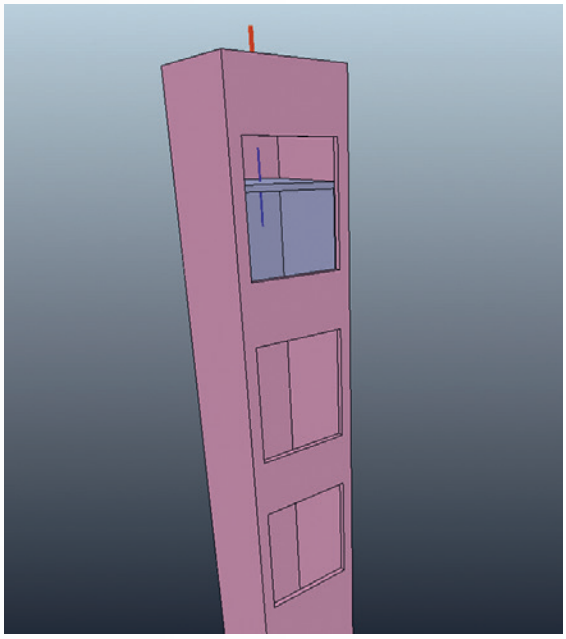


Figure 8: Screenshot of the V-Rep simulation of the lift used in the current prototype.

the lift and the program on the coarse level and the cable on the detailed level. Figure 8 shows a screenshot of the coarse simulation. In the current state of the prototype, the mappings Φ, Σ and I are hand coded for the example. First results of this Multi-Level-Simulation are promising. The simulations stay in synchronization and the results of the simulation meet our expectations.

The deployment of our prototype is distributed, but currently static. Concretely, the lift and the cable component are each deployed on their own virtual machine and the communication between the components is handled by our prototypical simulation bus which is based on Java's RMI. While still in an early stage, this bus will be capable of handling a dynamically deployed simulation in the future.

The provided lift example is useful to get first insights on the correctness of our method, but will be replaced by a real world example in order to provide validated results. Hand coded mapping functions (Φ, Σ) are used as placeholder and will be replaced by a machine learning based approach. We will employ labeled pairs of states of both levels provided by the user as training sets. First ad hoc experiments using this setup and well known machine learning regression techniques such as support vector machines and neuronal nets are very promising. This experiments although stress that a main challenge in doing this will be to use this techniques in such a fashion, that user provided requirements on the functions (such as providing stats within defined boundaries) are met.

Another further direction will be the dynamic activation of detailed simulations at runtime. The coarse level could be executed on its own, until an interesting state is reached. The detailed simulation is connected and is active only as long as needed.

Our next steps with regard to the dynamic Cloud deployment will include the evaluation of suitable strategies for the reasoning pipeline. Concretely, we will evaluate realistic applications built using simulation tools from the machine tool domain to extract parameters that are informative for determining a simulations' resource requirements. Then we will assess their accuracy and build an integrated framework for dynamic deployment.

7 References

[1] Aßmann, U., Bencomo, N., Cheng, B. H., France, R. B., 2015. models@runtime (dagstuhl seminar 11481). Dagstuhl Reports, 1(11).

[2] Blochwitz, T., Otter, M., Akesson, J., Arnold, M., Clauss, C., Elmqvist, H., Friedrich, M., Junghanns, A., Mauss, J., Neumerkel, D., 2012. Functional mockup interface 2.0: The standard for tool independent exchange of simulation models.

[3] Claes, R., Holvoet, T., 2009. Multi-model Traffic Microsimulations, in: Winter Simulation Conference, WSC '09. Winter Simulation Conference, Austin, Texas, pp. 1113–1123.

[4] Dahmann, J.S., Fujimoto, R.M., Weatherly, R.M., 1997. The Department Of Defense High Level Architecture.

[5] Dangelmaier, W., Mueck, B., 2004. Using Dynamic Multiresolution Modelling to Analyze Large Material Flow Systems, in: Proceedings of the 36th Conference on Winter Simulation, WSC '04. Winter Simulation Conference, Washington, D.C., pp. 1720–1727.

[6] Eker, J., Janneck, J.W., Lee, E., Liu, J., Liu, X.,

Ludvig, J., Neuendorffer, S., Sachs, S., Xiong, Y., others, 2003. Taming heterogeneity-the Ptolemy approach. Proc. IEEE 91, 127–144.

[7] Glaser, F., 2015. Towards Domain-Model Optimized Deployment and Execution of Scientific Applications in Cloud Environments. Proceedings of the Doctoral Symposium at the 5th Conference on Cloud Computing and Services Sciences (DCCLOSER 2015), Lisbon, Portugal.

[8] Göttsche, M., 2015. The 8th IEEE International Conference on Service Oriented Computing & Applications, Rome, Italy.

[9] Hong, S.-Y., Kim, T.G., 2013. Specification of multi-resolution modeling space for multi-resolution system. SIMULATION 89, 28–40. doi:10.1177/0037549712450361.

[10] Huber, D., Dangelmaier, W., 2011. A Method for Simulation State Mapping Between Discrete Event Material Flow Models of Different Level of Detail, in: Proceedings of the Winter Simulation Conference, WSC '11. Winter Simulation Conference, Phoenix, Arizona, pp. 2877–2886.

[11] Lorido-Botrán, T., Miguel-Alonso, J., Lozano, J. A., 2012. Auto-scaling Techniques for Elastic Applications in Cloud Environments. Technical Report: University of the Basque Country, 11 – 14. doi:10.1145/2611286.2611314.

[12] Qasha, R., Cała, J., Watson, P., 2015. Towards Automated Workflow Deployment in the Cloud using TOSCA. Towards Automated Workflow Deployment in the Cloud using TOSCA.

[13] Rao, D.M., 2003. Study of Dynamic Component Substitutions (Dissertation). University of Cincinnati.

[14] Reynolds,Jr., P.F., Natrajan, A., Srinivasan, S., 1997. Consistency Maintenance in Multiresolution Simulation. ACM Trans Model Comput Simul 7, 368–392. doi:10.1145/259207.259235.

[15] Schütte, S., Scherfke, S., Tröschel, M., 2011. Mosaik: A framework for modular simulation of active components in Smart Grids, in: Smart Grid Modeling and Simulation (SGMS), 2011 IEEE First International Workshop on. IEEE, pp. 55-60.

[16] Vukojevic-Haupt, K., Karastoyanova, D. Leymann, F.: On-demand Provisioning of Infrastructure, Middleware and Services for Simulation Workflows. In: Proceedings of SOCA 2013.

Project data

The project is funded from SWZ with one TV-L E13 staff position since August 2015 at the sites Clausthal and Göttingen. Involved scientists are:

Prof. Dr. Andreas Rausch
Software Systems Engineering
Department of Informatics
Clausthal University of Technology

Stefan Wittek, M.Sc.
Software Systems Engineering
Department of Informatics
Clausthal University of Technology

Prof. Dr. Jens Grabowski
Research group for Software Engineering for Distributed Systems
Institute of Computer Science
University of Göttingen

Michael Göttsche, M.Sc.
Research group for Software Engineering for Distributed Systems
Institute of Computer Science
University of Göttingen

Cloud-Efficient Modelling and Simulation of Magnetic Nano Materials

Pavle Ivanovic, Harald Richter, Arezoo Bozorgmehr

1 Introduction

In the following, a trans-disciplinary approach will be described that considers in an equal manner the problem of magnetic nano materials, their mathematical modeling, solvers available for it and their efficient execution in a cloud. The project report at hand is organized as follows: In the first chapter, it is explained why magnetic nano materials are of high interest for future computer architectures. However, sophisticated ultra-high vacuum-apparatuses and raster scan microscopes are needed to do research in that field, together with access to a super computer or parallel computer. It is therefore suggested by us in the first chapter to simulate magnetic nano materials in a cloud, which is nowadays the predominant computing paradigm, and to reduce thereby difficult laboratory work. In chapter 2, the state of the art for both, the mathematical modelling of magnetic materials, as well as the cloud's capability for HPC efficiency is outlined. The mathematical model is based on the so-called Landau-Lifschitz-Gilbert equation (LLGE) from the 1930s. However, only a small set of solvers exists for the LLGE utilizing the Message Passing Interface (MPI), which can be distributed onto multiple servers of a cloud. This set comprises the Nmag, Magpar and Vampire software packages. Furthermore, a short overview on these solvers is given in chapter 2, together with a quick summary on virtualization and OpenStack as cloud operating system which we used. In chapter 3, the LLGE is evaluated with respect to its inter-process communication in order to make the solver algorithms more cloud-efficient. We found out that several changes in the existing codes are desirable and described them in detail. Subsequently, a short description of the hardware and software of our cloud follows that was used by us to measure the execution times of the solvers before and after the changes we are going to make in the remaining two years of the project at hand. Finally, the first big step in the project was in chapter 3 to identify the cloud's overhead for inter-VM communication. As a consequence, we concluded that a standard cloud can never be used as a supercomputer and not even as a parallel computer. It is our task in the project to change this. In chapter 4, performance tests are described made by us in order to quantify the HPC effectiveness we can expect from a standard cloud. For that purpose, we simulated a cube of Permalloy with x/y/z-dimensions of 30x30x100 nm3, respectively, as ferromagnetic material. The mesh of the cube consisted of 4,100 grid points and 18,780 finite tetrahedra. Input files were defined by us with the same input geometry, the same number of mesh points, as well as the same ferromagnetic material, initial magnetization and simulation parameters, such as the number and duration of time steps in order to make the output results identical, which we achieved finally. By means of four simulation scenarios, we could demonstrate that the achieved speedups and efficiencies in our cloud are too small for high-performance computing (HPC), and that the solvers do not scale to many virtual CPUs. In the fifth chapter, we are proposing three methods to make a standard cloud HPC-efficient. The methods are based on removing all inter-vCPU, inter-VM and inter-server communication-over-head. The latter is achieved by a direct connection of MPI to Infiniband in the host OSes of all parts of the parallel user code. For that purpose, the inter-VM shared-memory QEMU PCI device (ivshmem) is engaged, and the MPI libraries in each guest OS are separated from the libibverbs library of Infiniband by a deep cut into the solver software. Furthermore, we are suggesting to create ivshmem between two VMs on the same server, without using Infiniband, which does not need a software cut and thus also not our two adapters between. Finally, we were able to propose a similar method for an efficient inter-VM communication on the same server. It is expected that by these methods, significant improvements

can be achieved with respect to HPC-effectiveness of standard clouds. The project report ends with a conclusion and a comprehensive reference list.

1.1 Motivation

According to [1] and [2], magnetic nano materials are thin layers (1-100 nm) or small areas (1-100 nm2) of specific materials such as: iron, cobalt, nickel, gadolinium or dysprosium, which are elements. Alternatively, the materials can also be samarium-cobalt or neodymium-iron-boron, which are alloys, or iron oxide or barium carbonate, which are ceramics. The commonality of these materials is that elementary magnets exist in them, which are smaller than 1 nm in size. Usually these tiny magnets are randomly oriented in the material, and no global magnetic field results. However, each elementary magnet can be oriented in a desired direction by means of an external magnetic write-field. Because of that property, magnetic nano materials are important in technology, since they can be employed as high-capacity computer-storage and as tiny switching elements as well. Physicists consider such materials as the computer's components of the future, though in hard drives and so-called magnetic RAM chips (MRAMs) they are already now in commercial usage (fig. 1). Especially interesting and challenging appears to be the research direction of "Spintronic" in the field of magnetic nano materials. Spintronic may deliver an ultimate bit density for a storage, because it uses the orientation of a single electron's spin to store a bit as "up" or "down". Additionally, Spintronic has the potential to provide for the smallest switching elements at all with trillions of transistors on one chip that are

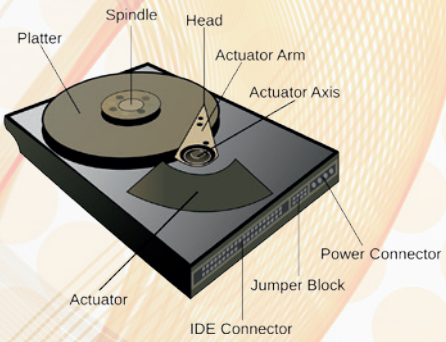


Fig. 1: Example for a commercial hard drive and Magnetic RAM chip.

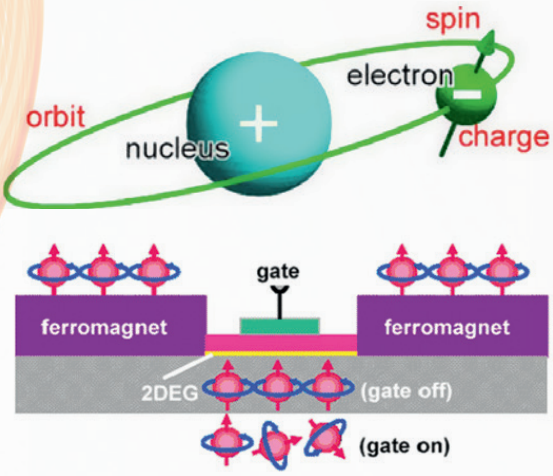


Fig. 2: Spintronic as the ultimate bit storage and as small low-power switching-elements.

operating with very low power (fig. 2). Because of these perspectives, research in magnetic nano materials is important and should be fostered. Unfortunately, research in that field requires special equipment, such as scanning tunnel microscopes and ultra-high vacuum laboratory-setups, together with access to a supercomputer or parallel computer. The latter are needed to simulate in space and time the material on a nano or pico scale. Because of these financial and organizational hurdles, research is limited to a smaller circle of scientists and not open to a world-wide community, which delays progress.

1.2 Simulative Approach

In this project, it is suggested to simulate magnetic nano materials in a cloud instead of using a supercomputer or a parallel computer, and to replace as much as possible experimental work in the lab by computer simulations, because the latter is faster, cheaper and easier. So far, only few software simulators for magnetic nano materials exist, and none of them was written for a cloud. As a consequence, no code runs efficiently in a cloud, which means that it does not scale up to the large number of processing elements that are available in a cloud. On the other hand, clouds are a widespread technology, and using a public cloud is very cost-efficient as well. For instance,

the Cloud provided by the Amazon Web Service (AWS) allows its clients to have Teraflops and Terabytes together with short response times for a few Euros per month as rental fees. With that cost and performance structure, scientists can use their simulation tools more frequently and visit their laboratories fewer than today. Also private or community clouds have in principle the same capabilities.

2 State of the Art

The state of the art for both, the mathematical modeling of magnetic materials, as well as for a cloud's capability for HPC efficiency is outlined.

2.1 Modelling of Magnetic Materials

During a period of 20 years, a mathematical model for magnetic materials was developed by three people. The result was the so-called Landau-Lifschitz-Gilbert equation (LLGE), which is a partial differential equation in $\mathbf{R}^3$. In eq. 1, this LLGE is given, as it was formulated in its first form by [2], [3].

Eq. (1):

$$\frac{\partial}{\partial t}\vec{m} = -|\gamma| \cdot \vec{m} \times \vec{H}_{eff} + \alpha \vec{m} \times \frac{\partial}{\partial t}\vec{m}$$

In eq. 1, $\vec{m}$ is the normalized magnetic dipole moment of the vector field $\vec{M}$, which is created by the elementary magnets in the material. This dipole moment is given by eq. 2. M_s is the saturation magnetization of the material at the temperature, at which the

Eq. (1): $\vec{m} = \frac{\vec{M}}{M_s}$

material is investigated, and can be considered a constant. γ is the gyromagnetic ratio of the material, which is also a constant depending on the material, $\vec{H}_{eff}$ is the effective magnetic field inside of the material, that is caused by the external write-field and some other effects, which will be explained later. α is the so-called Gilbert damping constant of the material.

The LLGE describes the time behavior of the magnetic dipole moment of each elementary magnet under the influence of an external magnetic write field. This behavior can be illustrated by means of a mechanical spinner on which an external force or impulse is exerted. Such a manipulated spinner starts to "gyrate", which means that it rotates around two axes at the same time. This double rotation is called "precession". The difference between the mechanical analogy and the magnetic material is that the external force which is exerted onto the spinner must be replaced by $\vec{H}_{eff}$, and furthermore, that the spinner must be replaced by its torque, which means by the dipole moment $\vec{M}$ of the materials elementary magnets. This means that $\vec{M}$ gyrates in every elementary magnet. As in the spinner analogy, the gyration of $\vec{M}$ is also damped. However, only in the magnetic material, the dipole moment $\vec{M}$ of an elementary magnet starts to orient itself either parallel or anti parallel to $\vec{H}_{eff}$, which is called "relaxation" of $\vec{M}$. The time development of $\vec{M}$ and thus also of $\vec{m}$ is shown in fig. 3. In fig. 4, the same behavior is

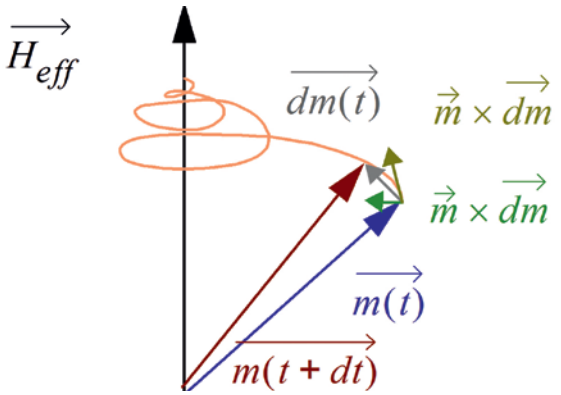


Fig. 3: Time behavior of the normalized magnetic dipole moment $\vec{m}$ of an elementary magnet in a ferromagnetic material in an external magnetic field, which results in $\vec{H}_{eff}$ inside of the material.

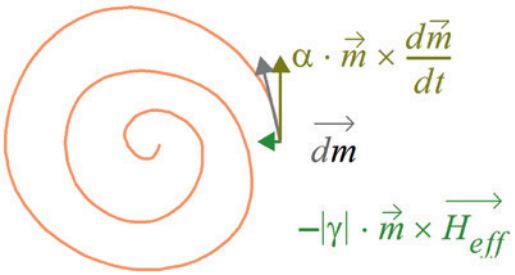


Fig. 4: Movement of the normalized magnetic dipole moment $\vec{m}$ seen from the top.

depicted from the top view. Both figures explain that $\vec{m}$ is moving on a spiral curve into the direction of $\overrightarrow{H_{eff}}$. This can be understood by eq. 1 in the following way: the first term in eq. 1 describes the precession of $\vec{m}$, while the second term denotes how much the amplitude of this precession is damped. To summarize: from eq. 1 and fig. 3 and fig. 4, it becomes clear that any magnetic dipole in a ferro-magnetic material orients itself sooner or later in the direction of an external field, thus writing a non-volatile "0" or "1" bit into a Weiss domain. This is called magnetization. Because of the permanent magnetization, the written bit can be read-out later by a nano-scale read-head. Furthermore, by the given explanations, the mathematical model we are using was outlined, and we will continue with the simulation of this model.

2.2 Simulation of Magnetic Nano Materials

Nowadays several Open-Source software-packages for solving the LLGE exist. However, only a few of them are using parallelized code that can be distributed onto the multiple servers of a cloud. The other packages contain either sequential code, or are using the OpenMP [4] computing library, which is based on loop parallelization in a shared-memory environment. Since shared-memory does not exist between servers, this is not applicable for a cloud. However, a small remainder of solvers is using the Message Passing Interface (MPI), which is the standard mean for communication inside of supercomputers and parallel computers. MPI exists in different implementations, such as mpich [5], [6] or open MPI [7], for example. Besides MPI, a helper tool called Multi-Purpose Daemon (MPD) [8] can additionally be used to launch and manage the parallel user code in an easier way. After a longer literature search, we found out that only three open-source MPI-based solvers exist which are Nmag [9], Magpar [10] and Vampire [11]. Each of them is a complex simulation-package. In the following, a short overview on Nmag, Magpar and Vampire is given, complemented by an evaluation of a cloud's principle capability for HPC.

2.2.1 NMAG

Nmag is using finite elements, is implemented in C and Python and engages mpich2 for parallel execution. It requires an input-geometry and mesh that is created by external tools such as Netgen [12] or Gmsh [13], for example, which discretise the magnetic material into small tetrahedra. Furthermore, Nmag provides for post-processing features after simulation, and stores its output data efficiently in gzip files.

2.2.2 MAGPAR

Magpar is written in C++, uses also mpich2 and can process the same input mesh and geometry files as Nmag do, after having applied some small adaptations via pre-manu-factured scripts. As in Nmag, the Netgen and Gmsh tools can be used. This allows for a simple comparison with Nmag. Magpar uses numerous libraries for static energy minimization, dynamic time integration and post processing. In general, Magpar is comparable to Nmag with respect to functionality.

2.2.3 VAMPIRE

In contrast to the previous two solvers, Vampire was developed to solve the LLGE with atomic resolution, which means it computes the LLGE not on the nano but the pico level. It works with a preciseness of Angström in space and picoseconds in time. This allows to simulate various physical effects in an unprecedented way, but brings additional complexity and computational overhead. Furthermore, it uses OpenMP and is the latest development in LLGE solver codes. Finally, Vampire does not need external tools for creating the input-geometry and mesh.

2.3 HPC in a Cloud

It is a well-known fact from supercomputing and parallel computing that HPC-efficiency is determined by the bandwidth and latency that can be achieved by the communicating parts of a parallel code. In clouds, both bandwidth and latency are influenced by the overhead that is needed for the data exchange between logical processing elements called vCPUs (= virtual CPUs), as well as between virtual machines (VMs) that contain these vCPUs. VMs and vCPUs exist in a so-called guest OS, which is an operating system that is executed as a user process by the real operating system a computer in a cloud has. The real operating sys-

tem is called host OS. The bandwidth and latency achievable in a cloud depend on the physical communication speed and the software overhead that is needed to emulate vCPUs inside of VMs in guest OSes.

2.4 OpenStack

OpenStack [14] is an open-source cloud operating system written in Python that has a comprehensive set of features and tools, which are continuously further-developed every 6 months via new releases. Its constituting parts are first the software on the so-called controller node, and second the tools for its "compute nodes". On the controller node, the main software component is a cloud-wide process-scheduler called "Nova". The compute nodes in turn have "Neutron" [15] as their main part, which provides for a virtual network between vCPUs, VMs, real servers and the Internet. Additionally, there exist several virtual storage capabilities on each compute node.

Finally, the controller node has an administrator GUI called "Horizon dashboard", together with diverse management tools, such as the „Ceilometer" for performance measuring, for example.

3 Project description

We started the project by evaluating in the first step the LLGE equation of eq. 1 with respect to its parallel process communication in order to make the algorithms more cloud-efficient. We found out that some changes in the existing solvers are desirable, which were not performed by us yet but planned. These changes are described in chapter 3.1. In the second step of the project, we used our private cloud and installed Magpar, Nmag and Vampire on it. The solvers were installed as they are, i.e. without any changes in order to measure their execution times before any algorithmic improvements have been made. Their performance figures can then be used subsequently to compare with changed algorithms. Furthermore, the difficult part in the second step was to create proper input files for Magpar/Nmag on the one hand and for Vampire on the other that define the identical physical problem with the same size and the same physical input parameters, and that results in the same output. The reason for this difficulty was that Vampire is different to Magpar/Nmag in all respects, and that all three solvers have completely disjunct default-settings, which must all be changed to make input and output comparable. The found input files are described in chapter 4.1. Before that, the desirable algorithmic changes are explained.

3.1 A Cloud-Efficient LLGE Solver

The LLGE has some structural aspects that can be utilized for better cloud-efficiency, because the LLGE has a second, mathematically equivalent form, which is given by eq. 3:
Eq. (3):

$$\frac{\partial \vec{m}}{\partial t} = -\frac{|\gamma|}{1+\alpha^2} \cdot \vec{m} \times \vec{H}_{eff} - \frac{\alpha|\gamma|}{1+\alpha^2} \vec{m} \times (\vec{m} \times \vec{H}_{eff})$$

3.2 Proper LLGE Selection

Because of eq. 3, Nmag, Magpar and Vampire must be examined which one is using which form of the LLGE. Then that form that is best suited for a cloud must be identified. Afterwards, the solver with that form must be selected. The chosen solver must additionally allow to compute the effective magnetic field $\overrightarrow{H_{eff}}$ inside of the nano material most efficient. This field constitutes of five factors, which are the exchange energy E_{exch}, the anisotropy energy E_{anis}, the Zeeman energy E_{zee}, the demagnetization energy E_{demag} and the magnetoelastic energy E_{me}. In the following, it is explained in detail how to compute these five constituents more efficiently.

3.2.1 Cloud-Efficient Computation of $\overrightarrow{H_{eff}}$ in the Material

The external write-field of the bit memory results in an effective magnetic field $\overrightarrow{H_{eff}}$ inside of the material. Therefore, $\overrightarrow{H_{eff}}$ has to be known for eq. 1 and eq. 3, which means it must be computed. Unfortunately, $\overrightarrow{H_{eff}}$ is determined not only by the external write field, but by several material-internal physical-effects, whose computation is generally difficult. In [3] it is even reported that the ordinary partial differential equation of the LLGE mutates into a highly complex integral/differential equation if $\overrightarrow{H_{eff}}$ is going to be correctly

calculated without any assumptions or restrictions. Fortunately, for ferro-magnetic materials newer research results have revealed that $\overrightarrow{H_{eff}}$ can be computed according to a formula given in [2]. For that case, $\overrightarrow{H_{eff}}$ is "simply":
Eq. (4):

$$\overrightarrow{H_{eff}} = \frac{2A}{\mu_0 \cdot M_s}\nabla^2\vec{m} - \frac{1}{\mu_0 \cdot M_s} \cdot \frac{\partial}{\partial \vec{m}}\overrightarrow{F_{anis}} + \vec{H}_a + \vec{H}_d$$

In eq. 4, A is the so-called "exchange constant", that defines the exchange energy E_{exch}, which will be explained later. μ_0 is the constant vacuum permeability, $\overrightarrow{F_{anis}}$ is the anisotropy energy density, which is a function of the orientation of the magnetization. Its integral is the anisotropy energy E_{anis}. $\vec{H}_a$ is the applied external write field, and $\vec{H}_d$ is the demagnetizing field, that is created by the magnetic material itself.

The expression given in eq. 4 is very compact compared to the formulas used in many older LLGE solvers, such as MICROMAGNUM [16] for example. Because of that, eq. 4 is a good basis for a cloud-efficient computation of the LLGE. Thus, the codes of the solvers under test must be scrutinized to find out, which one is using eq. 4. If the result reveals that no solver uses eq. 4, because it is just too new, then the effort to modify the best candidate according to that method must be estimated. For that purpose, it must be explored how eq. 4 can be split into sub tasks, which can be computed by already existing solver libraries that have the described communication behavior a cloud has. The result of these investigations will lead to the decision, either to follow eq. 4 or to optimise existing solvers, that compute $\overrightarrow{H_{eff}}$ in the "old" way. If eq. 4 is the best path to follow, then we must implement it in our cloud as part of the project. Otherwise, if a decision is made for the old way then the next chapter becomes relevant.

3.2.2 Efficient Computation of $\overrightarrow{H_{eff}}$ in the Old Way

The classical way for computing $\overrightarrow{H_{eff}}$ is explained by means of MICROMAGNUM. This solver does not compute $\overrightarrow{H_{eff}}$ in the compact form of eq. 4, but in a simplified version of the old way, which is nevertheless still highly complex. In the full classical way, $\overrightarrow{H_{eff}}$ is computed by the so-called "effective energy" that is deposed in the magnetic material. This effective energy is according to [2] as follows:
Eq. (5): $E_{eff} = E_{exch} + E_{anis} + E_{zee} + E_{demag} + E_{me}$
eq. 5 means that E_{eff} consists of five components, which are given as follows: 1.) the exchange energy E_{exch}, which is caused by the magnetic dipoles that are interacting with each other. E_{exch} is lowest when all dipoles are oriented in the same direction. 2.) The anisotropy energy E_{anis}, which occurs due to the magnetic anisotropy the material has: each ferro-magnetic material can be easily magnetized in one specific direction, but it can only be magnetized with difficulties in any other direction. E_{anis} is minimised when the material is magnetized in its easy direction. 3.) The Zeeman energy E_{zee}, which is caused by the external write field. It is proportional to the cosine of the angle between the write- field and the material's internal magnetization vector $\vec{M}$. 4.) The demagnetization energy E_{demag}, which is created by the magnetic field the material exerts upon itself because of its dipoles. 5.) The magneto-elastic energy E_{me}, which comes into existence, because the material behaves in some way as a mechanical feather that is elongated or compressed by $\overrightarrow{H_{eff}}$. This effect is called "Magnetostriction". In MICROMAGNUM, however, the Magnetostriction is neglected. Additionally, this simulator directly identifies the mentioned five sub energies of the full classical way with corresponding four sub fields as follows, by neglecting the field that results from material elongation or compression.

Eq. (6):

$$\vec{H}_{eff} = \vec{H}_{exch} + \vec{H}_{anis} + \vec{H}_{zee} + \vec{H}_{demag}$$

The exchange field in MICROMAGNUM is given by eq. 7 with $\vec{M} = M_s \cdot m$ as:

Eq. (7): $$\vec{H}_{exch} = \frac{2A}{\mu_0 \cdot M^2_s}\nabla\vec{M}$$

The demagnetization field is taken from Maxwell's equations according to eq. 8:

Eq. (8): $$\nabla\vec{H}_{demag} = -\nabla\vec{M}$$

In tensor notation, the demagnetization field reads according to [17] as:

Eq. (9): $\vec{H}_{demag} = -\int \underline{N}(\vec{r} - \vec{r}^{\,i})\vec{M}(\vec{r}^{\,i})dV^i$

In a descriptive view, the demagnetization tensor of the material N twists and scales the magnetic moment $\vec{M}$ such that the demagnetization field is obtained. Finally, the uniaxial anisotropy field is computed as given in eq. 10, with K as the anisotropy constant of the material, and with Θ as the angle between the magnetic moment $\vec{M}$ and the easy direction of the material.

Eq. (10): $\vec{H}_{aniso} = K \cdot \sin^2\Theta$

From these equations, it can be seen that eq. 4 is a big progress from the view point of complexity. However, this is not the only method that should be followed: In MICROMAGNUM and other simulators, eq. 9 is discretized by eq. 11. In eq. 11, $\vec{H}_{demag}(\vec{r}_k)$ is the demagnetization field of the volume element k. The field in that *k*-th element is the result of the fields of all volume elements *i* with *i=1,2,...,n* the complete material

Eq. (11):

$$\vec{H}_{demag}(\vec{r}_k) = -\sum_i^{1,n} \underline{N}(\vec{r}_k - \vec{r}_i)\vec{M}(\vec{r}_i)$$

sample has. Furthermore, $\vec{r}_k$ points to that *k*-th element for all *k*, with *k=1,2,...,n*. Because of that, the time complexity of eq. 11 is O(n2) which is too much for being efficient. In addition to that, each product between the tensor N and $\vec{M}$ requires 9 multiplications and 6 additions as indicated in eq. 12, which makes it computational intensive.

Eq. (12):

$$\begin{bmatrix} H_{demag,x}(x) \\ H_{demag,y}(y) \\ H_{demag,z}(z) \end{bmatrix} = \begin{bmatrix} N_{x,x} & N_{x,y} & N_{x,z} \\ N_{y,x} & N_{y,y} & N_{y,z} \\ N_{z,x} & N_{z,y} & N_{z,z} \end{bmatrix} \cdot \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}$$

Fortunately, eq. 11 defines also a discrete convolution between N and $\vec{M}$. A convolution, however, can be substituted by a multiplication in Fourier space. Thus, eq. 11 can be replaced by eq. 13,

which requires only three O(*nlogn*) time complexities instead of

Eq. (13): $F(\vec{H}_{demag}) = F(\underline{N}) \cdot F(\vec{M})$

one $O(n^2)$ complexity, which is much better for larger n. For explanation: two time complexities O(*nlogn*) must be spend to transform N and $\vec{M}$ into the Fourier space by a Fast Fourier transform (FFT), and one time complexity O(*nlogn*) is needed for transforming $F(\vec{H}_{demag})$ back into $\mathbf{R}^3$ by an inverse FFT.

In a cloud, the challenge is to use that forms of FFT and inverse FFT that exhibit a minimum inter-server and a maximum intra-server communication. In [18], it was shown by one of the authors (Richter) that from all canonical forms of the FFT, the Cooley-Tukey and the Gentleman-Sande FFT have the highest locality and thus the lowest inter-server communication. It was also shown by one of us that the performance difference between these two best forms and the worst form is a factor of 4, for the case of 16384 data points that are computed on 16 servers. However, hundreds of thousands of data points are needed for a correct simulation of the magnetic behavior of nano materials, and simulations cannot not be made in 1D but must be done in R3. Both facts boost the performance difference in the two best forms of the FFT to much higher improvement factors. Thus, a proper library function for the parallel computation of the FFT must be employed whenever a convolution has to be calculated. This FFT must be based on MPI and on the Cooley-Tukey or the Gentleman-Sande variant to make a convolution cloud-efficient. If this does not exist, it must be written and tested by ourselves because it pays out a lot.

After substantial algorithmic changes, which were performed either according to eq. 4 or due to FFT instead of convolution, the performance tests must be repeated and their effectiveness must be evaluated again.

3.3 Ultimate Algorithmic Improvements

If all described improvements will prove to be still not enough for a high cloud-efficiency and a good scalability, then the most promising existing LLGE solver candidate must be inspected and changed more deeply. All solvers are based

either on finite elements or on finite differences methods, because these are the standard ways for solving partial differential equations. In [19], it is furthermore pointed out that the usual time-step discretization finite elements or on finite differences methods are using is not applicable in LLGE, because it violates the law of conservation of the magnetization $\vec{M}$. Additionally, the author in [19] summarizes various complex methods that conserve the magnetization and gives a survey on algorithms for solving the LLGE numerically under this boundary condition. Finally, in [20], a simple method for solving the LLGE is shown. Thus, a work package of the project would be to review all existing LLGE solvers, including those which are not MPI-based, under the aspects listed in [19] and [20], and to findout which one could have the best potential for cloud-efficiency. Furthermore, an estimation should be given for the effort needed to convert that candidate into an MPI-based solution, provided that it is not. If this effort is manageable, we must convert this best candidate into an MPI code. After all improvements, performance tests must be redone and the achieved cloud effectiveness must be discussed in detail. After the discussion of the improvements on the mathematical side of the problem, we continue with an explanation of its hardware/software side.

3.4 Hardware of our Private Cloud

Our cloud included 18 servers from Dell and Sun, comprising a total of 84 cores, 324 GB RAM and 19 TB of Disk Storage. The most powerful machine we used was a server from Dell with 2 sockets for Intel Xeon CPUs of type E5620@2.40GHz and four cores each. The cores are with Hyperthreading allowing for 2 logical processing elements per core. They included VT-x Virtualization, L1 data and instruction caches of 32 KB each, L2 caches of 256 KB each, and a shared L3 cache of 12.288 K. The size of the main memory was 32GB. The second strongest machine we used was similar. The only differences were: there were Xeon CPUs of type X5355@2.66GHz, and the cores were without Hyperthreading. Additionally, there was a L2 cache of 4 MB but no 3rd level cache. From these hardware specifications, it can be seen that we have used no new or high-performance machines for our tests. Nevertheless, we believe that our cloud is comparable to many other private clouds, and therefore useful for measurements and drawing conclusions. For this speaks also the fact that the servers in our cloud were originally coupled by Ethernet cards and a switch of 1 Gbits/s transmission speed, respectively. Such simple communication networks are widespread in private clouds.

3.5 Software of our Private Cloud

The host OS for the cloud was CentOS 7.1, and Ubuntu 14.04 was used as Guest OS. As cloud operating system, we have chosen OpenStack with the Juno release installed.

3.6 Inter-VM Communication Overhead

The first big step in the project was to identify the cloud's overhead for inter-VM communication. By studying [21] and [22], we were finally able to draw a block diagram which is shown in fig. 5, and that explains the inter-VM communication in OpenStack. From that diagram, it becomes clear that OpenStack in special, but also standard clouds in general are highly inefficient with respect to their inter-VM communication because of the huge software overhead needed to implement a virtual network based on virtual devices and to isolate VMs from each other for security reasons. For any inter-server data-exchange, three different machines are involved, and according to fig. 5, the following sequence of steps holds: if part 1 of a user application wants to send an intermediate result, which it has calculated for part 2, then this result R has to travel first through the following set of software interfaces: 1.) the MPI process in VM 1, its Berkeley-Sockets, the TCP/IP protocol stack of Guest OS 1, a device driver and a virtual Ethernet-network interface-card (vNIC) [23]. In the second step of the journey, R has to go through multiple Neutron components on the compute node, which are 2.) a tap [24] interface that operates on ISO layer 2 and accepts Ethernet frames from the vNIC, a firewall called "Security Group", a so-called qbr interface, a further "qvo" interface which adds the address tag that is needed for the vNIC's Ethernet Frame in a virtual LAN environment (VLAN), an integration bridge br-int that combines multiple VLANs into one frame stream, and a tunnel bridge br-tun that packs and unpacks VLAN-tagged vNIC-Ethernet Frames into real Ethernet Frames. In

the third step, R has to go through a device driver in the Host OS, a real Ethernet network interface card, and the Ethernet switches in the cloud. In the fourth step, R has to penetrate the controller node with its Host OS and Neutron via a real Ethernet card, a device driver and a second tunnel bridge. To this end, R's journey doubles in order to make the way to the second compute node, where the second part of the parallel user code is in VM 2. This adds another four steps of overhead. As a consequence, we concluded that a standard cloud can never be used as a supercomputer and not even as a parallel computer. It is our task in the project to change this.

4 Performance Tests

In order to quantify the performance we can expect from a standard cloud, we did performance tests by executing Magpar, Nmag and Vampire on our cloud. To make the tests comparable with each other, we had to define for all three solvers the same input geometry, the same number of mesh points, as well as the same ferromagnetic material, initial magnetization and simulation parameters, such as the number and duration of time steps. However, the solvers were developed by different coding teams, in different decades and with different interfaces for user input. Additionally, the solver's default settings are completely different, which is why an intensive study of all solver documents was needed in order to avoid inaccurate results and thus wrong conclusions. In the beginning, we added also a forth solver whose output served for us as a reference for the first three. The forth solver was OOMMF [25], which is based on shared-memory communication in a single server, thus providing a simpler use-case than the others, which have the complication of being distributed across machines. After we found proper input configurations, OOMMF was not needed any more. Finally, we avoided any over-committing of real resources, by launching at most that many vCPUs as real cores were available.

4.1 Input Configuration

We selected a cube of Permalloy with x/y/z-dimensions of 30x30x100 nm3, respectively, as ferromagnetic material. Permalloy has a magnetic saturation of 0.86e6 A/m. Saturation is defined as the material's state, beyond which a further increase of the effective magnetic field $\overrightarrow{H_{eff}}$ cannot increase the magnetization. Furthermore, the value of the exchange-coupling constant A for Permalloy is 13.0e-12 J/m, the dimensionless damping constant of the LLGE was set to 0.5, and the direction of the initial magnetization vector was chosen to be x=1, y=0, z=1. This vector is normalized by the solvers to unity length and results therefore in [0.707, 0, 0.707]. The mesh of the cube is depicted in fig. 6 and consists of 4,100 grid points, 18,780 tetrahedra inside of the bar and 3,460 surface elements outside. The cube was decomposed into pieces of equal size, and each piece was allocated to a logical processing unit which was a vCPU in our cloud. As output, the time course of the magnetisation vector was calculated in the cube for the

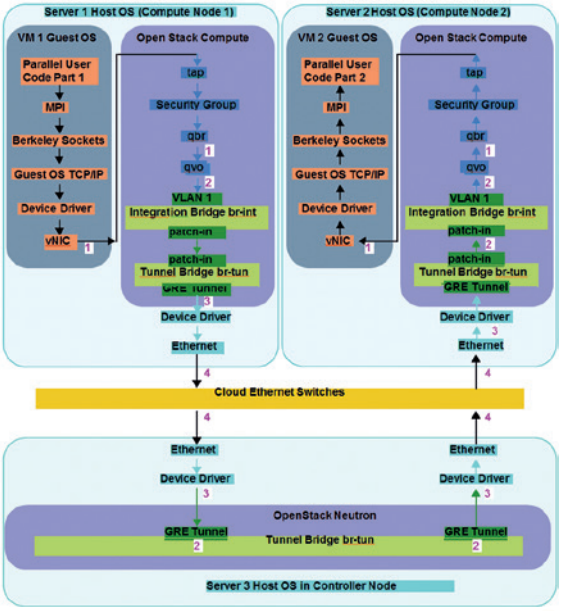


Fig. 5: Software overhead in OpenStack for inter-VM communication. vNIC=virtual Ethernet Card; 1=vNIC Ethernet Data Frames, 2=VLAN-tagged vNIC-Ethernet Frames, 3=Generic Routing Encapsulation Protocol packets (GRE) to transport VLAN-tagged vNIC-Ethernet Frames over real Ethernet Frames, 4=real Ethernet Frames. Furthermore, the integration bridge br-int combines multiple VLANs into one frame stream. Finally, the tunnel bridge br-tun packs and unpacks VLAN-tagged vNIC-Ethernet Frames to and from real Ethernet Frames.

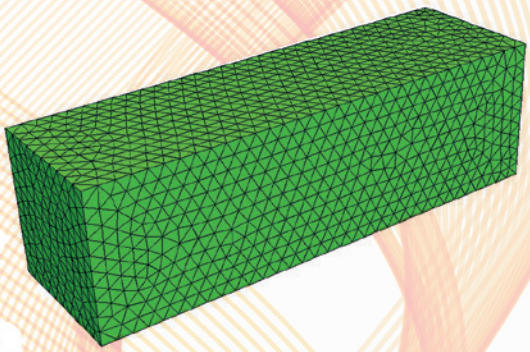


Fig. 6: Input Mesh of the Permalloy bar.

interval of 0-300 ps, which is the point in time when the Permalloy reached saturation. This time course is a function of the write-field $\overrightarrow{H_{eff}}$ that demagnetizes the initial magnetization over time. The magnetisation vector was calculated with 6,000 time steps in Magpar and Nmag and with 120,000 time steps in Vampire, i.e. with a factor of 20 more because of the atomistic approach of the latter.

4.2 Simulation Results

In fig.7, the simulated time function of the three spatial components of the magnetisation vector is depicted. We see that under the influence of $\overrightarrow{H_{eff}}$, Magnetisation aligns parallel to the z-axis until it reaches saturation (= green curve) , while the x and y component approaches to zero. All four solvers delivered the same result. However, Vampire needed days of computing time, while Magpar required only seconds for the same result. The reason is that Vampire simulated atom by atom and picosecond by picosecond. Because of that behavior, we excluded this solver for further investigations and compared in the following only Magpar and NMag with each other. The results of this comparison are presented in the next chapter.

4.3 Performance Results

We measured the elapsed times T of Magpar and NMag for various scenarios on two different machines, which were our most powerful server and the second most powerful one. This allowed us to compare the influence of the computing power and to quantify the impact of communication between vCPUs, VMs and servers in the virtualized environment of OpenStack. Furthermore, on each vCPU, the same code was executed in parallel, but with different parts of the cube of fig. 6 as input. Additionally, MPI was linked as an external library for communication to the solver code in every vCPU. This means that the MPI functionality was not an own process but part of the solver. Finally, we used the so-called Completely Fair Scheduler (CFS) of Linux for our tests [25], which is one of 6 process schedulers available in the kernel.

4.3.1 Scenario 1

The purpose of scenario 1 was to test the inter-vCPU communication inside of the same VM.

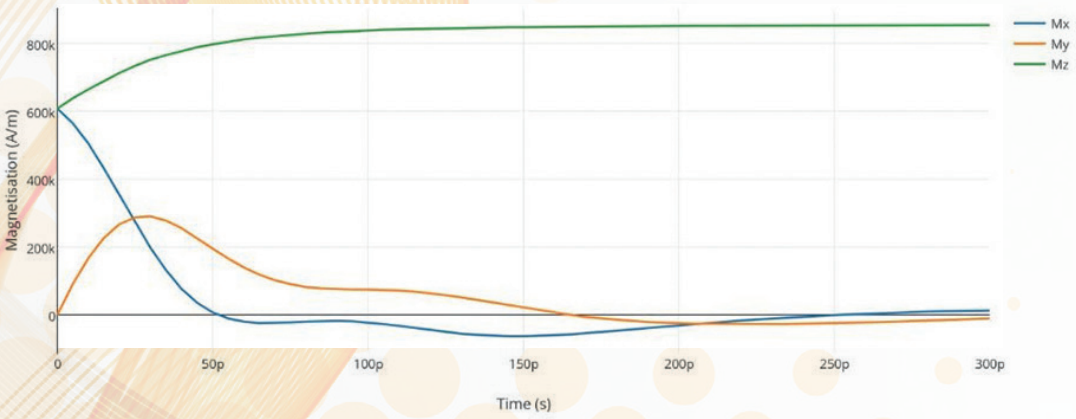


Fig. 7: Time course of the three spatial components of the magnetisation vector.

Table 1: Elapsed time T, speedup S and efficiency E for Magpar and NMag, which were executed on 1 VM with n vCPUs on the same physical CPU on the strongest machine. n is varied from 1 to 4.

	Magpar			Nmag		
Setup	T [s]	S	E [%]	T [s]	S	E [%]
1 vCPU	114	1	100	303	1	100
2 vCPUs	79	1.44	72	249	1.22	61
3 vCPUs	66	1.72	57.3	248	1.22	40.6
4 vCPUs	65	1.75	43.8	239	1.27	31.7

Table 2: The influence of inter-VM communication on T, S and E. The two VMs are executed by two physical CPUs on the strongest server. The number n per VM is varied from 1 to 4, resulting in a total of 2-8 vCPUs.

	Magpar			Nmag		
Setup	T [s]	S	E [%]	T [s]	S	E [%]
2 vCPUs	115	0.99	49.6	309	0.98	49
4 vCPUs	91	1.25	31.3	284	1.07	26.7
6 vCPUs	91	1.25	20.8	280	1.08	18
8 vCPUs	102	1.12	13.9	311	0.97	12.2

Therefore, a VM was statically allocated by us to one physical CPU. Furthermore, the number of vCPUs inside of that VM was varied from 1-4. This was done in order to avoid overcommitting of resources, since not more than 4 cores were available per physical CPU. Furthermore, the used vCPU variation allowed also a 1-to-1 allocation of cores to vCPUs. The measured results are shown in table 1. In this table, the elapsed times T for both, Magpar and Nmag, are given, together with the speedup S and the efficiency E, which were added by us in a post-processing phase. S is defined by us as the execution time on n vCPUs compared to that on 1 vCPU. E is defined as $E = S/n$ and denotes the vCPUs' utilization. Additionally, the first line of table 1 with 1 vCPU shows T, S and E for the sequential case without parallelization, but with virtualization. In this case, the single VM contained only one 1 vCPU. Furthermore, the physical CPU behind the vCPU executed in fact two processes: the first process that represented the VM with the guest OS, and the second process that represented its vCPU. The vCPU was forked-off as a child of the VM and was treated in host OS as a thread. This thread had an own host-OS process-ID (PID), together with a parent process ID (PPID) that was identical to the PID of the VM process. Please note, it can be seen from first line of table 1 that Magpar outperformed Nmag in case of 1 vCPU by a factor of 2,7 with respect to T. The first line serves also as a reference for the subsequent lines with 2-4 vCPUs. From these lines, it can be seen that Magpar scaled always better than Nmag with respect to S and E. However, the speedups in the solvers never reached a factor of 4, although up to 4 vCPUs and thus also 4 physical cores were engaged. This is a remarkable result, because the inter-vCPU communication can theoretically be accomplished via the shared 3rd-level cache of the physical CPU that contains these cores, which would be very fast. Because of that, S and E should be much better than they are. At first sight, it was unclear to us what the reason for this behavior was. We identified the following reason: the huge overhead for inter-VM communication shown in fig. 5 comes also into existence for inter-vCPU communication in the same VM, because OpenStack's standard configuration for VMs does not support shared-memory coupling inside of the VM. This was proven by inspecting the libvirt [26] xml output that was created by the OpenStack dashboard to configure the VM. That inspection revealed that the VM's configuration did not support virtual shared memory between its vCPUs. Because of that xml configuration file, OpenStack did not know about the existing 3rd- level cache-coupling, which is shared CPU memory, and thus could only use the same inefficient mechanism for inter-vCPU coupling as for inter-VM coupling. This issue can easily be solved by us by post-

processing the VMs xml description after creation, because it is very inefficient.

4.3.2 Scenario 2

The purpose of scenario 2 was to test the inter-VM communication between two VMs on the same server via shared main memory. The VMs were located on different physical CPUs in the same server, which was again the most powerful machine. Both VMs were connected with each other similar to fig. 5, but resided the same compute node.

Finally, the VMs were identical in their xml setup file. In table 2, results of scenario 2 are presented. S and E were also calculated with the first line of table 1 as basis. As one can see, Magpar outperformed again Nmag in all respects, but T never reached the minimum value of table 1 which was 65 seconds. The shortest elapsed times in table 1 and table 2 differ by a factor of 1.4. This is a surprising result, because there is a rule of thumb that the fastest communication between processing elements exists between the level 2 or level 3 caches inside of a single physical CPU with multiple cores. This allows 16 bytes to be transferred in about 1 ns. The rule of thumb says also that the second fastest communication is between the CPUs inside of the same server, because it can be performed via shared main memory. It allows 1 byte to get transferred in about 1 ns, which is 1/16 of the previous speed. Although this big speed difference exists, we measured only a factor of 1.4 in practice. The reason is the inefficient mechanism for inter-vCPU coupling that obviously slows down significantly the shared cache coupling. Furthermore, T is shortest in table 2 if the problem is executed by 4 or 6 vCPUs but not by 8 vCPUs. Because of that behaviour, the chosen problem is not scalable, not even to the small number of 8 vCPUs. The reason for that is that in the solvers the proportion between computation time and communication becomes worse, if the 4100 grid points of fig. 6 are distributed onto 8 vCPUs. This problem size is too small for 8 vCPUs. According to that, S and E behave best for 4 or 6 vCPUs but not for 8. Finally, S never reached its maximum value of 1.75 of table 1. This means that the best speedup is obtained if execution happens on the cores of the same physical CPU, although OpenStack does not know about shared cache coupling. The reason for that result is that the 3rd level cache does its job independent of OpenStack. It quickly delivers in many (but not all) cases data to a vCPU, regardless whether these data was obtained via virtual shared memory, which was not the case, or via vNIC which was true. However, it can do its job only in the same physical CPU and not between two CPUs, and it suffers from the huge overhead of fig. 5, which is why we had not an S closer to 8 for 8 vCPUs.

Table 3: The impact of the inter-VM communication between two VMs, which are located on two different servers and coupled via 1 Gbit/s Ethernet cards. The number of vCPU per VM is varied from 1 to 2. The servers were our strongest and second strongest machines.

	Magpar			Nmag		
Setup	T [s]	S	E [%]	T [s]	S	E [%]
2 vCPUs in 2 VMs on 2 servers	176	0.65	32.4	389	0.78	38.9
4 vCPUs in 4 VMs on 2 servers	152	0.75	18.7	362	0.84	20.9

4.3.3 Scenario 3

The target of scenario 3 was to test the influence of inter-server communication for two VMs, which were located on two different servers and that were coupled via 1 Gbit/s Ethernet. In that scenario, the number n of vCPUs per VM was varied from 1 to 2, resulting in a total of 2 and 4 vCPUs that were engaged. Each VM was allocated to one physical CPU, and each vCPU was assigned to one core. Please note, we had to use the command-line interface of Nova instead of the dashboard to manually allocate VMs to servers, because the dashboard does not support this. This way, the cloud scheduler could be bypassed. It was needed to do so, because Nova chooses hosts for VMs due to other criteria. As reference for T and E, the single vCPU execution time from table 1 was again taken. The results are presented in table 3.

One can see, that in table 1 the elapsed time T for two vCPUs is better by a factor of 1.5 than the corresponding T for two vCPUs in table 3. This has to be reconciled with another rule of thumb that says that the slowest communication is between servers, because it requires about 1 ns for 1 bit only, resulting in 1 Gbit/s transmission speed. Thus, a performance difference of approximately 128 should be present between inter-core (=intra CPU) and inter-server communication. However, we measured only a factor of 1.5. The reason is again the inter-server communication-overhead according to fig. 5 that fully dominates all types of communication, from slowest to fastest. Because of that domination, it slows down the fastest type by the highest reduction factor, but the slowest only by the smallest factor, thereby distorting the proportion between slow and fast. Because of that, it is clear that both, the solver's inter-server data-exchange and the software-overhead required for it has to be minimized. We are following both paths in order to be cloud-efficient. Finally, Nmag is again worse than Magpar, which is why we focused in the following only on Magpar as the best solver.

4.3.4 Scenario 4

The goal of scenario 4 was to identify the problem size that is needed in scenario 3 to achieve a speedup of better than 1, because it was really poor. For that purpose, the problem size was varied between 4,100; 40,000; 50,000 and 100,000 grid points. As reference, we used 1 vCPU in 1 VM on our strongest server for the respective problem size. Furthermore, the first line in table 4 was copied from table 1 and table 3 for better comparison. We can see from this line that a problem with size of 4,100 grid points is too small for being distributed on two servers. This changes for problem sizes that are bigger than that by a factor of about 10 or more, but only in case of 2 vCPUs. However, S and E are steadily decreasing with increasing problem size in the constellation of 4 vCPUs. The best S and E we can get for 4 vCPUs is for problems that are bigger than 4,100 but smaller than 40,000 grid points. Additionally, in table 4 the 2 vCPU constellation steadily increases its S and E. However, this increase is small, and the level from where it starts is also small. Nevertheless, it means that it pays out to distribute big problems on multiple servers, but it does not pay out to compute them at the same time on multiple vCPUs of the same VM, at least from a certain problem size on. Finally, we can observe from table 4 that the elapsed time for the nonlinear partial differential-equation eq. 1 does not scale linearly with the problem size but over-proportional, which means that big nanomagnetic problems can be solved only by parallel calculation, because they have a non-linear computing complexity.

Table 4: Influence of the problem size on T, S and E. Scenario 4 is identical to scenario 3, but only Magpar was measured with different number of grid points.

	1 vCPU in 1 VM on 1 server	2 vCPUs in 2 VMs on 2 servers			4 vCPUs in 2 VMs on 2 servers		
# grid points	T [s]	T [s]	S	E [%]	T [s]	S	E [%]
4,100	114	176	0.65	32.4	152	0.75	18.7
40,000	2616	2309	1.13	56.5	1477	1.77	44.3
50,000	3900	3372	1.16	58	2215	1.76	44
100,000	9428	8035	1.17	58.5	6004	1.57	39.3

To summarize, in our standard cloud the achieved speedups and efficiencies proved to be too small for HPC, and the investigated solvers revealed that they do not scale to many vCPUs as well. Our cloud is as it is not suited for HPC. This was already clear from fig. 5 by quality but not by quantity. In the next chapter, it is described how to make our cloud HPC-efficient.

5 Proposed Methods for Cloud-efficient HPC

In the following, three new methods are proposed to make a standard cloud HPC-efficient. However, in the beginning we want to report first which methods were not successful.

5.1 Tuning of Linux Scheduling

Inside of the same server, the Completely Fair Scheduler (CFS) [27] assigns vCPUs equally to available cores in one iteration cycle, and re-assigns them a few seconds later to potentially other cores in the next cycle in order to balance load. During this procedure, CFS does not take into account if two vCPUs of the same parallel process are located on the same physical CPU and could thus communicate quickly with each other via shared 3rd level Cache. As a consequence, re-assigning vCPUs to physical cores will lead to an inefficiency, if the ongoing communication of two vCPUs is interrupted by CFS by de-scheduling, or by separating communicating vCPUs to different VMs. Additionally, every re-assignment to another core requires a cache flush in the source core and the building-up of a set of frequently used cache lines in the target core which takes time. We managed to solve this issue by pinning each vCPU permanently to a predetermined core by using the Linux "taskset" command, together with the help of the virsh virtual machine manager [28] from libvirt. The result was a ten percent perfo mance improvement, which is not significant. Therefore, we searched for other means for more efficiency.

5.2 Replacement of Ethernet

We replaced the 1 Gbit/s Ethernet cards in our cloud, including the Ethernet switch by Infiniband [29] cards, with a speed of 40 Gbit/s each and a 40 Gbit/s Infiniband switch. The cards are of type Mellanox MHQH19B-XTR with QSFP copper cables of 1 m length, and the switch is of type Mellanox Infiniscale IS5023. It has 18 ports and a port- to-port latency of 100 ns. The cards are not capable of performing vt-d [30], [31] or SR- IOV [30], which are hardware-based accelerators for better inter-server communication. Therefore, we integrated in the first attempt the Infiniband cards in the same way as the Ethernet NICs, i.e. according to fig. 5. Unfortunately, a further software overhead was needed to accomplish this: after the tunnel bridge br-tun of each compute node, we had to add a modified Linux TUN interface that encapsulated the GRE [32] protocol as payload into an IP packet and that additionally conveyed the IP packet via a protocol called IP-over-Infiniband (IPoIB) [32]. This was needed because Infiniband can transport nor Ethernet frames neither IP packets. Furthermore, the Ethernet device driver was replaced by us by the OFED verbs driver [34]. At the side of the receiving compute node, the double-packing and transporting over IPoIB had to be undone. Unfortunately, the described attempt to integrate Infiniband into our cloud failed. Measurements revealed that Infiniband's latency was even worse than in the 1 Gbit/s Ethernet case, at least for small block lengths, while the effective data rate did not exceed 3 Gbit/s. The reason was the huge overhead of fig. 5 plus IPoIB, which not even a 40 Gbit/s Infiniband could compensate. Additionally, the key feature of Infiniband, the remote direct memory access (RDMA) could not be used, because it is not supported by OpenStack.

5.3 Efficient Infiniband Integration into a Cloud

With our Infiniband cards, there is probably the only simple option for avoiding the overhead of fig. 5, as well as for bypassing OpenStack, because of its missing support for RDMA. This option is a new software approach we developed, which combines two technologies: the inter-VM shared-memory QEMU [35] PCI device (ivshmem) [36], [37], [38], [39], and 2.) the separation of the MPI libraries in each guest OS from the libibverbs [40] library of Infiniband. This approach improves significantly the inter-server communication-efficiency by removing all overhead of

fig. 5. by means of a direct connection of MPI to Infiniband in host OS in all distributed parts of the parallel user code.

5.3.1 ivshmem

By using libvirt, it is possible to create a so-called ivshmem shared-memory between guest OS and host OS. This memory comes into existence by means of a virtual PCI-device that is emulated by Qemu. This device allows for a zero-copy VM-to-Host communication, which is very efficient with respect to bandwidth and latency, because no internal data buffer exists. Furthermore, ivshmem is compatible with RDMA, and it is even compatible with OpenStack and its way to create VMs. If a user defines a VM by means of the Horizon dashboard, a libvirt configuration file is output by the dashboard in xml format. This file has to be post-processed before it is input to KVM [41] and Qemu, and thus before the VM is launched. The post-processing adds ivshmem shared memory between guest OS and host OS by a few xml tags only. For example, the following code piece creates "ivshmem-plain" (which does not support interrupts):

```
<shmem name='my_shmem0'>
 <model type='ivshmem-plain'/>
 <size unit='M'>4</size>
</shmem>
```

A device with interrupts is created by:

```
<shmem name='my_shmem_server'>
 <model type='ivshmem-doorbell'/>
 <size unit='M'>2</size>
 <server path='/tmp/socket-shmem'/>
 <msi vectors='32' ioeventfd='on'/>
</shmem>
```

In order to create two ivshmem devices for VM 1 and 2 of fig. 8, their configuration files must be edited and enriched as indicated above and due to [36], [37], [38] and [39].

5.3.2 Separation Of MPI From Libibverbs

We are proposing secondly to cut the API between the MPI library and the libibverbs library into two pieces and replace both ends of the cut by our own software. At the side of the MPI library, we replace libibverbs by an own adapter 1 that contains all calls of libibverbs that are made by MPI as its API. Adapter 1 is located in the guest OS. The only purpose of adapter 1 is to forward all MPI calls to libibverbs via shared memory to our second adapter. Adapter 2 is located in host OS and calls libibverbs by means of the data that is passed through it via the shared memory. Adapter 2 has as a API to libibverbs the calls which MPI is performing. In essence, both adapters are doing nothing except forwarding data back and forth from the respective sides of the cut. Adapter 1 thus emulates libibverbs, and adapter 2 emulates MPI by mimicking their APIs. They are empty with exception of data forwarding between guest OS and host OS and vice versa via ivshmem, which is very efficient.

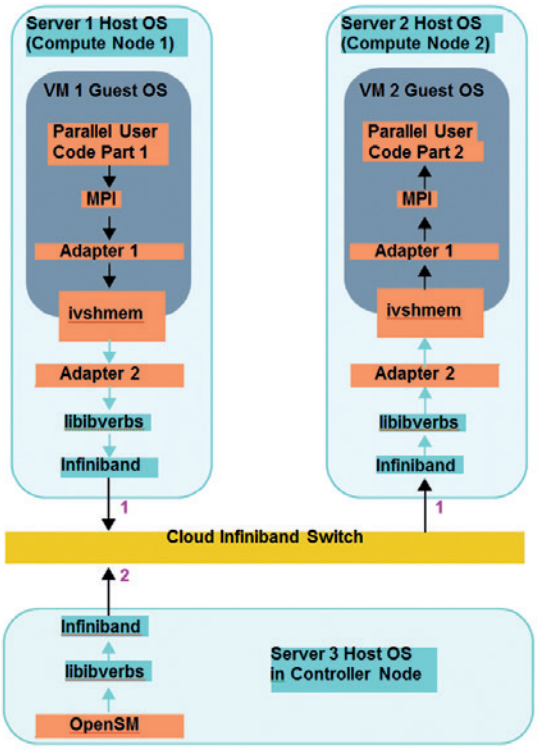


Fig. 8: Proposed solution for inter-VM communication between server 1 and 2 via MPI, ivshmem and Infiniband. 1=RDMA, 2=Switch Setup, OpenSM = InfiniBand Subnet Manager from OFED. ivshmem = Inter-VM Shared Memory QEMU PCI device. Adapter 1 = own code with libibverbs as API procedure heads. Adapter 2 = own code with MPI calls as API procedure heads.

5.3.3 Resulting Approach

The resulting approach is depicted in fig. 8. The Infiniband switch enables two or more VMs to communicate directly with each other via the respective Host OSes and ivshmem as intermediate stages. Furthermore, the ivshmem device enables the VM to access a region of POSIX shared memory (SHM) [42], [43] on the host and performs thus a guest/host bridge. POSIX SHM is better and simpler than the old SystemV shared memory. For SHM, a POSIX semaphore is used for the mutual exclusion of accessing processes in order to synchronize them. The guest/host bridging is accomplished via SHM by memory-mapped I/O from the Guest OS to the host OS. Because of the ivshmem shared memory, RDMA is possible, which is the fastest way to communicate via Infiniband. Finally, the switch in fig. 8 that is needed for a layer 2 switching of Infiniband's data frames have to be configured, for example, by the InfiniBand Subnet Manager [44] from OFED before the switch can be used. OpenStack's Neutron cannot do that, because it does not support Infiniband. The witch setting is made statically and depends on the number of servers that are engaged.

5.3.4 Efficient Inter-VM Communication on the same server

It is easily possible to create ivshmem between two VMs on the same server, without using Infiniband.

We propose the following efficient inter-VM communication on the same server (fig. 9) that does not need a software cut and thus also not our two adapters between. This is a simple solution that allows for multiple MPIs to directly communicate

5.3.5 Efficient Inter-vCPU Communication on the same Chip

Finally, we propose a similar method for efficient inter-vCPU communication on the same server that is shown in fig. 9. It is expected that by these three methods, significant improvements can be achieved with respect to HPC-effectiveness of standard clouds.

6 Conclusions

This project is considering magnetic nano materials, its mathematical modelling, the solvers available for computing it and their efficient execution a cloud. By this manifold approach, substantial insight was achieved in how to improve existing solvers on the algorithmic side, what hampers a standard cloud to be HPC-efficient, and which countermeasures are beneficial to turn it into an "HPC cloud". The latter mainly depends on the cloud's inter-server, inter-VM and inter-vCPU bandwidth and latency. If insufficient, a standard cloud will not speed-up but slow-down many parallel codes with respect to a super computer or parallel computer, because the data exchange

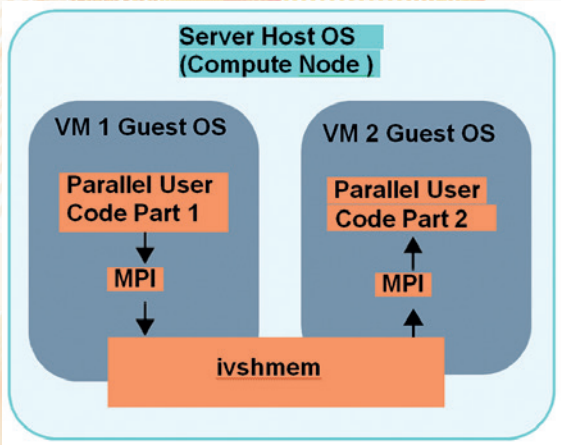


Fig. 9: Proposed solution for inter-VM communication on the same server via MPI and ivshmem. via shared memory, which is its fastest way.

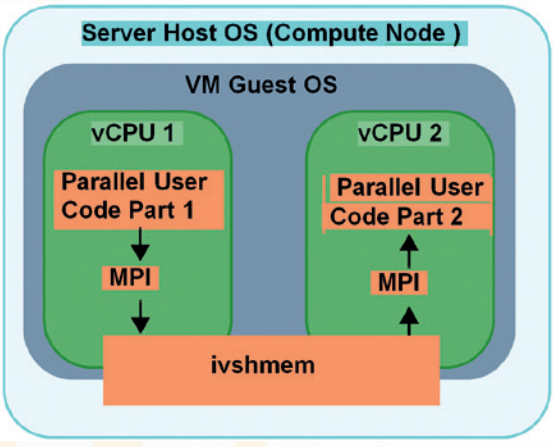


Fig. 10: Proposed solution for inter-vCPU communication in the same socket via MPI and ivshmem.

between the parts of the user code that are located on different servers cannot always be minimized. The reason for that behavior lies in the plethora of software interfaces that are existing in a standard cloud between any two communicating parts of a parallel code. However, we can completely get rid of OpenStacks overhead for inter-server, inter-VM and inter-vCPU communication. We achieve this without modifying the Linux kernel, KVM, Qemu or OpenStack. Furthermore, OpenStack is not involved in HPC communication, which is a big advantage when it comes to efficiency. For that purpose, the inter-VM shared-memory Qemu-PCI device (ivshmem) is engaged, and the MPI libraries in each guest OS are separated from the libibverbs library of Infiniband by a deep cut into the solver software. Furthermore, we are suggesting to create ivshmem between two VMs on the same server, without using Infiniband, which does not need a software cut and thus also not our two adapters between. Finally, we were able to propose a similar method for an efficient inter-vCPU communication on the same server. It is expected that by these methods, significant improvements can be achieved with respect to HPC-effectiveness in standard clouds.

The disadvantage is that OpenStack's benefits it provides by virtualizing the communication are lost. Because of that, we are suggesting to use for an HPC-efficient cloud two parallel networks: 1.) A switched Infiniband network that directly connects the parts of the parallel code and bypasses the cloud OS. This network is highly efficient with respect to bandwidth and latency and is used exclusively for HPC. 2.) A switched Ethernet that is fully integrated into the cloud OS and its virtualization concepts and which is exclusively used for Internet access and VM isolation.

7 References

[1] Vijay K. Varadan, LinFeng Chen, Jining Xie. Nanomedicine: Design and Applications of Magnetic Nanomaterials, Nanosensors and Nanosystems. s.l. : Wiley, November 2008.

[2] Micromagnetics in Wikipedia,http://en.wikipedia.org/wiki/Micromagnetics

[3] http://de.wikipedia.org/wiki/Landau-Lifschitz-Gilbert-Gleichung

[4] OpenMP, http://www.openmp.org/

[5] Mpich, http://www.mpich.org/

[6] MPICH2, https://www.mpich.org/about/overview/

[7] Open MPI, http://www.open-mpi.org/

[8] Multi-Purpose Daemon, https://cs.mtsu.edu/~rbutler/mpd/

[9] Nmag, University of Southampton, http://nmag.soton.ac.uk/nmag/

[10] Magpar, Werner Scholz, http://www.magpar.net/

[11] Vampire,. http://vampire.york.ac.uk/features/

[12] NETGEN - automatic mesh generator, Schöberl Joachim, https://sourceforge.net/proj- ects/netgen-mesher/

[13] Gmsh, Christophe Geuzaine and Jean-François Remacle, http://gmsh.info/

[14] OpenStack, https://www.openstack.org

[15] Neutron, https://wiki.openstack.org/wiki/Neutron

[16] http://micromagnum.informatik.uni-hamburg.de/

[17] A. J. Newell et al., A generalization of the demagnetizing tensor for nonuniform magne- tization, J. Geophys. Res. 98, 9551-9555, 1993

[18] H. Richter, MULTITOP - Ein Multiprozessor mit dynamisch variabler Topologie, Disser- tationsschrift für die Fakultät für Elektrotechnik und Informationstechnik der TU München, 1988

[19] Lubomír Bañas, Numerical Methods for the Landau-Lifshitz-Gilbert Equation, in Zhilin Li, Lubin G. Vulkov, Jerzy Wasniewski (Eds.): Numerical Analysis and Its Applications, Third International Conference, NAA 2004, Rousse, Bulgaria, June 29 - July 3, 2004, Re- vised Selected Papers. Springer 2005 Lecture Notes in Computer Science ISBN 3-540- 24937-0, pp. 158-165.

[20] Won Chang Nam, Min Hyung Cho and YoungPak Lee, Micromagnetic Simulations With Landau-Lifshitz-Gilbert Equation, Quantum Photonic Science Research Center, Hanyang University, Seoul 133-791, Korea,http://www.mathnet.or.kr/real/2007/5/NamWon-Chang.pdf and http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.101.1518

[21] H. Richter and A. Keidel and R. Ledyayev, Über die Eignung von Clouds für das

Hochleistungsrechnen (HPC), in Ifl Technical Report Series ISSN 1860-8477, Ifl-15-03, editor: Department of Computer Science, Clausthal University of Technology, Germany, 2015.
[22] Neutron, https://pinrojas.com/2014/07/29/theres-real-magic-behind-openstack-neutron/
[23] NIC, https://en.wikipedia.org/wiki/Network_interface_controller
[24] TAP, Wikipedia, https://en.wikipedia.org/wiki/TUN/TAP
[25] NIST OOMMF, http://math.nist.gov/oommf
[26] libvirt - virtualization API, https://libvirt.org/
[27] CFS by Molnár Ingo, Wikipedia, https://en.wikipedia.org/wiki/ Completely_Fair_Scheduler
[28] virsh - management user interface, https://linux.die.net/man/1/virsh
[29] Infiniband, http://www.mellanox.com/pdf/whitepapers/IB_Intro_WP_190.pdf
[30] H. Richter, A. Keidel, Hochleistungsrechnen und Echtzeit in virtualisierten Maschinen und Clouds - Die Intel Virtualisierungshilfen, in Ifl Technical Report Series ISSN 1860- 8477, Ifl-14-03, http://www.in.tu-clausthal.de/forschung/technical-reports/ ifi1403richter.pdf, editor: Department of Computer Science, Clausthal University of Technology, Germany, 2014.
[31] Intel VT-D, https://software.intel.com/en-us/articles/intel-virtualization-technology-for-directed-io-vt-d-enhancing-intel-platforms-for-efficient-virtualization-of-io-devices
[32] GRE. https://tools.ietf.org/html/rfc2784
[33] IPoIB, https://www.kernel.org/doc/Documentation/infiniband/ipoib.txt
[34] OFED verbs driver, [https://www.openfabrics.org/index.php/openfabrics-soft- ware.html
[35] QEMU, https://wiki.archlinux.org/index.php/QEMU
[36] ivshmem docu in libvirt, http://libvirt.org/formatdomain.html#elementsShmem
[37] ivshmem, https://qemu.weilnetz.de/qemu-doc.html#pcsys_005fother_005fdevs
[38] ivshmem, http://nairobi-embedded.org/linux_pci_device_driver.html
[39] ivshmem, https://github.com/qemu/qemu/blob/master/docs/specs/ivshmem-spec.txt
[40] Libibverbs, https://www.openfabrics.org/downloads/libibverbs/
[41] KVM, http://www.linux-kvm.org/page/Main_Page
[42] POSIX SHM http://man7.org/linux/man-pages/man7/shm_overview.7.html
[43] POSIX shared memory tutorial, http://man7.org/training/download/posix_shm_slides.pdf
[44] P. Hartman (Intel), Understanding the InfiniBand Subnet Manager, https://software.in- tel.com/en-us/articles/understanding-the-infiniband-subnet-manager

Project data

The project is funded from SWZ with 0.5 TV-L E13 staff position since August 2015 at the site Göttingen. Applying scientist is:

Prof. Dr. Harald Richter
Research Group
Technische Informatik und
Rechnersysteme
Department of Informatics
Clausthal University of
Technology

Numerically Intensive Simulations on an Integrated Compute Infrastructure

Fabian Glaser, Jens Grabowski, Alexander Bufe, Gunther Brenner, Christian Köhler, Philipp Wieder

Introduction

In the heterogeneous infrastructures that are commonly found in today's computing centers, the users of simulation software are confronted with the task of determining which type of computing resource is the most suitable one for their application. They also need to pick the scale at which these resources should be provisioned and ultimately get acquainted with the technical aspects of using the chosen infrastructure, which always requires significant learning effort. We aim to develop an architecture that simplifies this process and eases both infrastructure choices and resource usage for the simulation scientist.

Motivation

Our combination of the simulation application, the infrastructure we are initially focused on and the technical efforts to make them available with ease of use are motivated from various research disciplines:
From the point of view of **engineering**, the goal is to further the understanding of mass transport and conversion processes in porous media which helps to satisfy the demands from e.g. chemical process engineering. Here, the idea is to design porous materials specifically to be applied e.g. as supports for catalysts with optimized conversion rates and selectivity. This necessitates more accurate predictions of morphology-transport relations that also consider chemical reactions.

The relevant **mathematical methods** from computational fluid dynamics include in particular the Lattice Boltzmann Method (LBM) and its application using adaptive mesh refinement. However, these need to be developed further in order to account for the varying length scales on which processes take place in technically relevant porous media.

The simulations of flow in porous media not only have the purpose of increasing understanding of mass transport in porous media, but are also used to evaluate the developed architecture. These simulations are perfectly suitable as they cover a broad range of application cases with different requirements regarding the used infrastructure. Parameter studies for example have other requirements regarding the network connection as a single simulation of a large area or the transition from code development to actual simulations might have different resources requirements. Parallelization of the simulation code to make the method accessible to current High Performance Computing (HPC) setups featuring GPU accelerator cards is one aspect of concern from the view of **Computer Science**. Moreover, the ability to exploit the capabilities of modern heterogeneous infrastructures, including IaaS Clouds and HPC nodes, should be easily accessible available for the simulation scientist, including the automated choice of the most suitable resource type.

Model Driven Architecture

One approach to tackle the heterogeneity of I.T. systems is to use formal models as proposed by the Model Driven Architecture (MDA) [2] developed by the Object Management Group (OMG). Models hereby help to ease the communication between different involved parties and can be used to (semi-) automatically generate code. They can also provide the basis for analyzing or simulating the behavior of a system. Furthermore, they can help to structure information to make it more accessible. For this purpose, the OMG defines a set of standardized modelling languages, including the well-known Unified Modeling Language (UML).

One basic concept in MDA is the assumption of a *platform* on which the I.T. systems are about to be realized. Such a platform can comprise a specialized set of hardware or existing software setups on which the I.T. system depends. Each platform

poses some technical constraints, which need to be fulfilled by the I.T. system to be compatible. To enable decoupling to the systems logic from the technical peculiarities of a platform, the MDA envisions a clear separation between *platform-independent models* (PIMs) *and platform-specific models* (PSMs), and propose to use well-defined *model transformations* that define how to create a PSM from a PIM.

Such an approach is also applicable for handling the complexity of adapting a simulation application to a suitable resource type in a data center. To free the scientist from learning resource type specific technical details, this knowledge can be encapsulated in a model transformation. This way, the scientist can stay focused on the development of the logic of his simulation. To enable such a model transformation, a formal model of the simulation application is needed.

When automatically deploying applications in cloud environments, the required resources (VMs, storage, network, etc.) need to be launched in an orchestrated manner and the desired software configuration needs to be enforced on the launched resources. The Topology and Orchestration Specification for Cloud Applications (TOSCA) [1] aims to provide a standardized template format for orchestration based on XML. In cluster computing, resource management systems such as the Load Sharing Facility (LSF) use different job description formats to define the job to be executed. In contrast to the application models used in cloud computing, they do not define the infrastructure and configuration of the applications itself, but allow to set the path to the application to be executed and the command line parameters that are passed to the application at launch-time. Additional parameters can be set that define the resource requirements, such as the number of worker nodes and the number of cores per worker node that are needed for execution. Since cluster systems are integral parts of grid systems, grid resource management systems like the Globus Resource Allocation Manager (GRAM) or the gLite Workload Management System (WMS) use similar job description formats. These job descriptions are transformed to job descriptions suited for the resource management system that is managing the clusters associated to the grid sites. One main goal of this project, is to abstract from

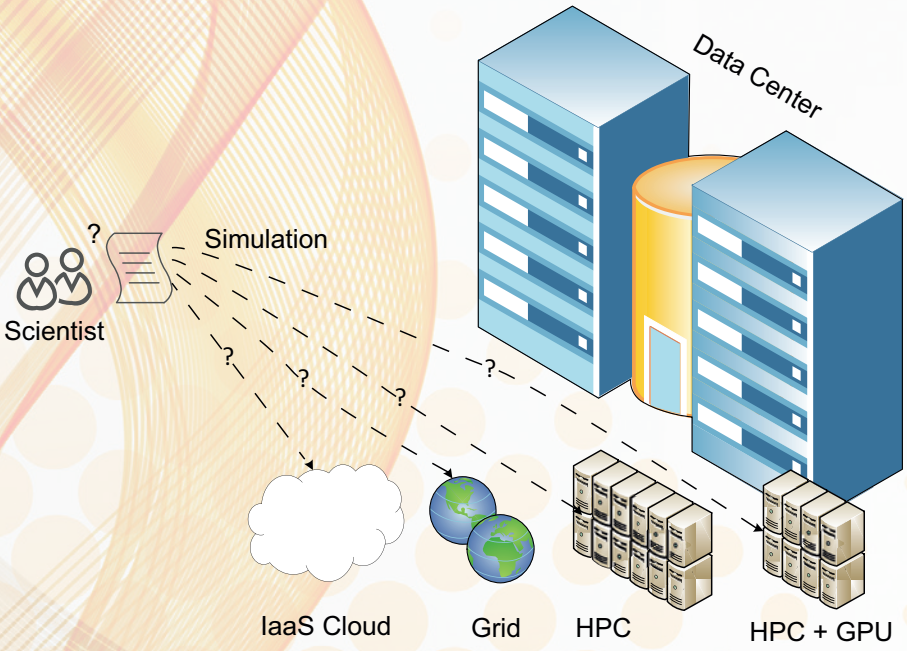


Figure 1: Infrastructure options available to the simulation scientist.

the resource specific formats described above. The goal is to provide a single entry point for the heterogeneous infrastructure and shield the simulation scientist from its inherent complexity.

Project description

When deploying a simulation in a modern data center, the simulation scientist is confronted with several options as depicted in Figure 1. The difficulty in using heterogeneous infrastructures from the simulation scientist's point of view hereby doesn't start with obtaining the know how necessary to utilize the various resource types. Beforehand, an appropriate decision of which resource type is most suitable for the simulation code needs to be made. This choice is influenced by resource availability, application compatibility (for example depending on support for GPU accelerators), user preferences and possibly the current or projected resource workload.

Architecture overview

Addressing this problem by using a model for the simulation application, which can in particular provide estimates of the expected resource demand, as well as models for each resource type, is an approach that can guide or potentially even automate this decision. Our envisioned architecture is therefore organized in separate layers, as depicted in Figure 2: The simulation application described by the Application Layer, in which models of the application topology and behavior as well as facilities to extract its resource requirements from the parameters and input data are contained, ultimately needs to utilize the IaaS Cloud, HPC and other possible resources provided by the Resource Layer and is doing so by using a separate Resource Management Layer, which abstracts the resources' individual ways of application deployment, data transfer, execution control and ultimately resource termination. These layers are connected by the Distribution Layer, in which the Distribution controller as a central point of interaction with the simulation scientist. It brings the simulation application's requirements together with the capabilities of the available resources and, once a resource has been chosen, is able to interact with it on the user's behalf.

Modeling of Cloud Infrastructure and Applications

In the case of IaaS Cloud resources, we make use of the Topology and Orchestration Specification for Cloud Application (TOSCA), which is an already established standard that allows defining the application topology and behavior in a provider-agnostic way. TOSCA provides abstract concepts for modeling cloud resources and services. *NodeTypes* are used to define abstract base classes for resources and services, e.g., a virtual machine or a virtual network and the *Properties* that are allowed for them. *NodeTemplates* are instantiated NodeTypes with fixed values for their Properties. *RelationshipTypes* and the corresponding *RelationshipTemplates* allow to define relationships and dependencies between the NodeTemplates. In addition, TOSCA defines the concept of *Requirements* and *Capabilities*. NodeTemplates can define Requirements, e.g., a software might require a certain amount of RAM in a virtual machine, and Capabilities, e.g. a virtual machine might offer a certain amount of RAM for executing software. If NodeTemplates have matching Requirements and Capabilities, a relationship can be instantiated between them.

Extension to HPC cluster setups

The concept of matching Requirements and Capabilities is not cloud specific and fits the functionality of the Distribution controller very well. We adopt this concept it in the following when defining the functionality of the Distribution Controller. Since the configuration of HPC systems in terms of e.g. job submission queues, the hardware specifications of the compute nodes reached by these as well as the available software modules varies between setups, it is not feasible to construct any kind of universal model for compute clusters. However, the concept of having a platform-independent application model for the simulation, which is supplied with user-specified parameters (either manually set or extracted from the input data) and instantiated to yield a platform-specific model, is analogously applicable to the infrastructure model as well, in particular when compute cluster setups are concerned: The abstract model of the relevant Load Scheduler software should be free of any setup-specific data. Only the admin-

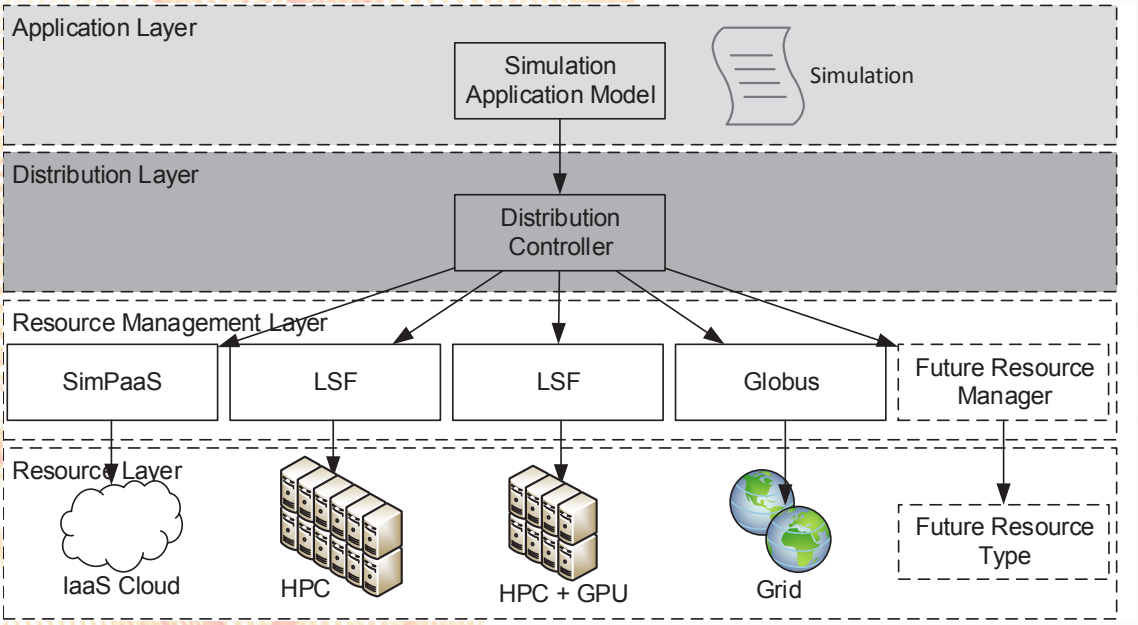


Figure 2: Role of the Distribution Controller.

istrators of a certain setup should specify their setup's design in the form of a Load Scheduler configuration and thereby instantiate the infrastructure model. In addition, the availability of preinstalled applications can, unlike in the case of cloud infrastructures, not be enforced by the user, but becomes a part of the setup-specific infrastructure model.

We orientate on TOSCA in our model development for HPC systems as well, since its metamodel is designed in such a way that it allows constraints on the possible group memberships and relations between the entities constituting the application topology, resembling the way a concrete HPC setup constrains the compute jobs' resource demands that can be met. This information therefore has to become a dynamic part of the deployment model for a batch system, i.e. it is not solely determined by the infrastructure metamodel, but by the configuration specific to a concrete cluster setup.

Figure 3 depicts the envisioned workflow from the viewpoint of the simulation scientist. A PIM encodes the structure and behavior of the simulation application in a target resource independent way and is stored in a model repository. The simulation scientist is then able to select (Step 1) and adapt (Step 2) the existing models from the repository. The selected and instantiated model is then passed to a Distribution Controller (Step 3) that evaluates the parameters of the model (Step 4), selects the suitable target infrastructure accordingly and transforms the selected model into a PSM that matches the requirements of the targeted infrastructure (Step 5). In the next step, the resource provisioning and the automated deployment of the simulation application on the targeted resource is triggered (Step 6). After that, the simulation scientist is given access to the provided resource via a Command Line Interface (CLI) and can execute his simulations accordingly (Step 7). When all simulation runs are done, the simulation scientist can collect the output data and triggers the cleanup and termination of the provided infrastructure (Step 8).

Our envisioned repository architecture is depicted in Figure 4 and it reflects the analogy between the instantiation processes for application and infrastructure models when a compute cluster instead of a cloud-type infrastructure is considered. The configuration-independent model for

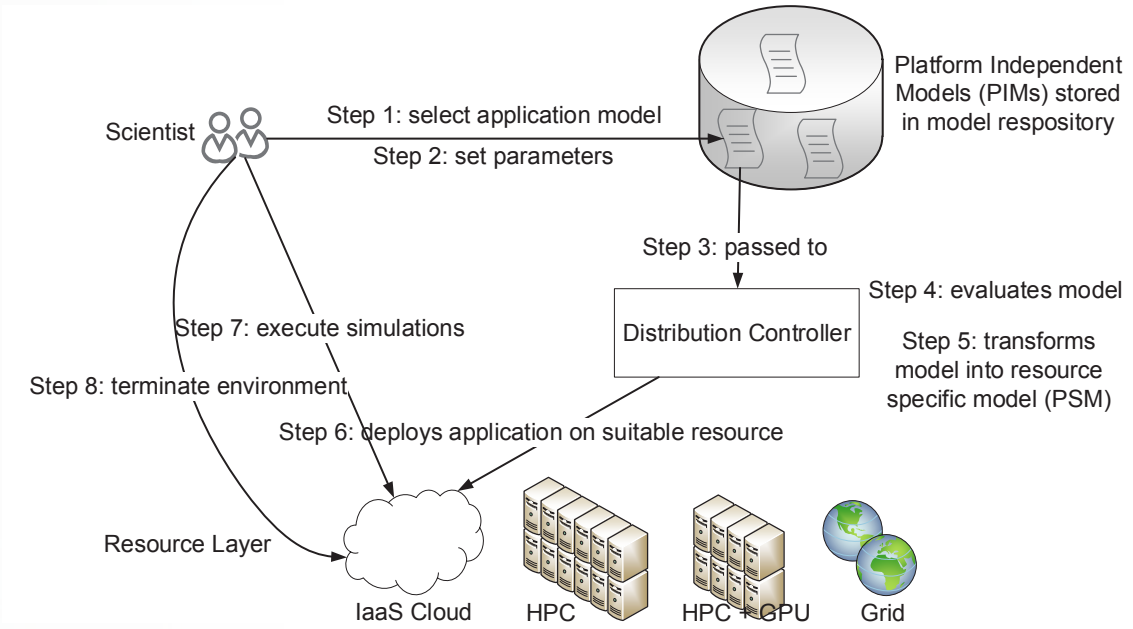


Figure 3: Envisioned Workflow from the viewpoint of the simulation scientist.

a Load Scheduler is instantiated by the cluster administrators and the resulting infrastructure model is uploaded to the responsible institution's repository. From there, it can be accessed by the simulation scientist , who uses it in the Distribution Layer to make a choice of the most suitable infrastructure and eventually instantiate the deployment model which is used for job submission to the cluster.

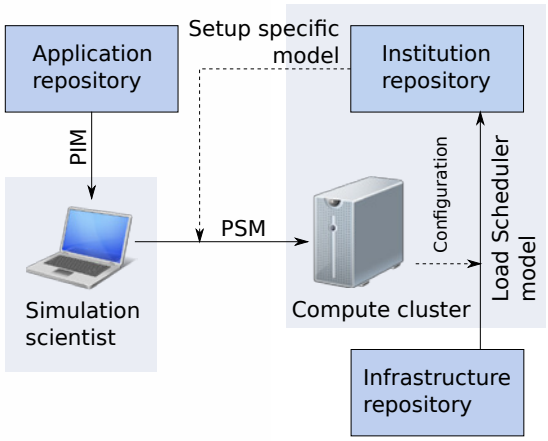


Figure 4: Repository architecture.

By placing the instantiation of the application and infrastructure models under the supervision of the group of people most knowledgeable about the respective systems, we ensure separation of concerns, which is a core principle of MDA.

Application Case: Lattice-Boltzmann method with adaptive mesh refinement

In classical CFD methods, macroscopic flow variables like velocity and pressure are directly obtained from the continuity and the Navier-Stokes equation, which in turn are derived from conservation principles of mass and momentum. Typically, finite volume or finite element methods are used to discretize and solve the resulting non-linear system of second order partial differential equations numerically. In particular, for incompressible flows special attention has to be paid for the coupling of the continuity and Navier-Stokes equation using iterative methods. As an alternative approach, the lattice-Boltzmann (LB) method is employed in our simulation use case. It originates from Boltzmann's kinetic molecular dynamics and may be understood as a discretization in space and time of the velocity-discrete Boltzmann equation. Thus, the LB method determines the

probability distribution function for molecular particles with a certain velocity at a certain position in space. The macroscopic variables such as density and velocity are obtained from the moments of this distribution function. Although the common usage of uniform cartesian grid for LBM might be suitable for some kind of geometries, it generally leads to a unnecessary high number of cells as even uninteresting parts of the flow are resolved with high resolution. In order to overcome this issue, a coarse base grid with an adaptive mesh refinement (AMR) technique can be used. Whereas most implementation use uniform cuboids as refinement blocks, we use the simulation framework AMROC to obtain a general Berger-Collela type AMR system with refinement blocks of arbitrary size.

There are many approaches to extend LBM for multiphase flow. We focus on the widely used Shan-Chen model which is based on a pseudopotential. In our previous work, the model has been extended to chemical reactions in porous media [3]

Current state

As the first scientific issue we investigated the flow in porous media. Since packings of particles are of great importance for chemical engineering e.g. in catalyst beds, knowledge of flow characteristics such as pressure drop is crucial. Despite that importance, a universal correlation of the pressure drop in packed beds is unknown. Even for the special case of spheres in an infinite extended

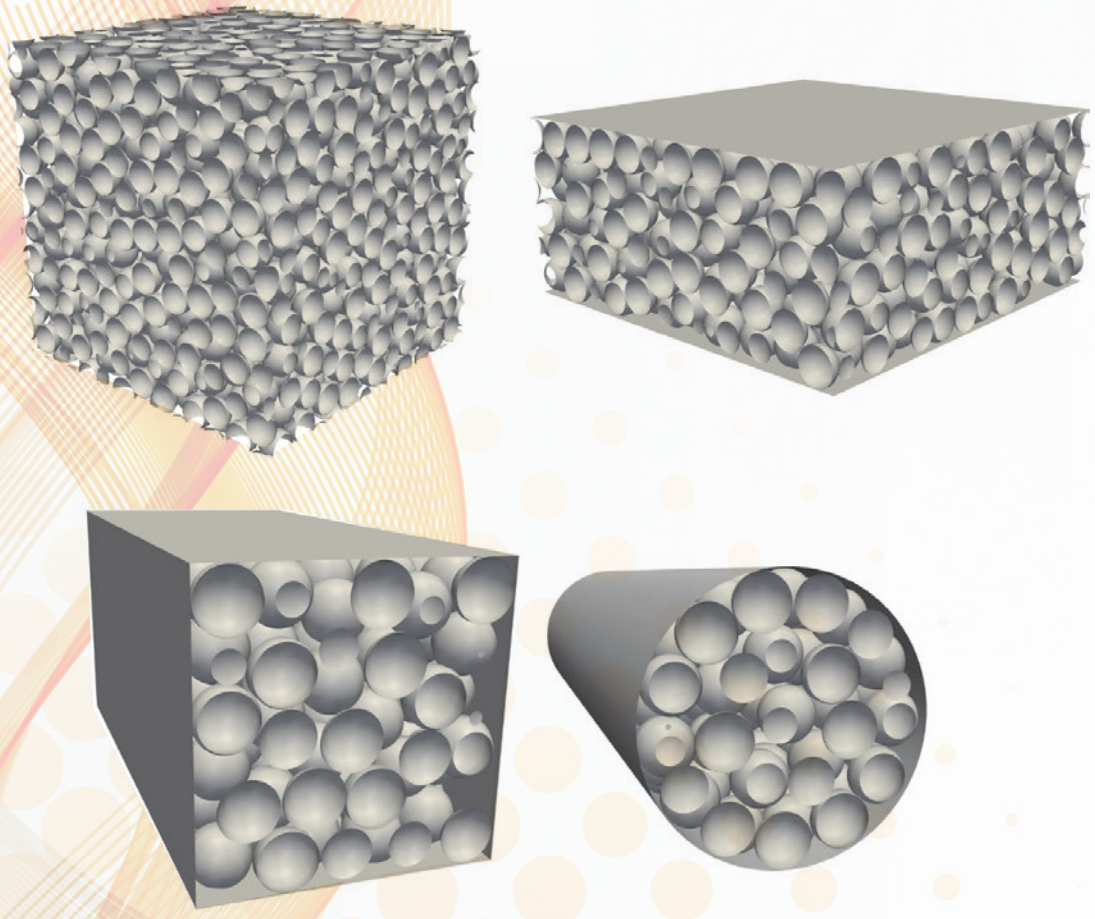


Figure 5: Sample packings of the different configurations

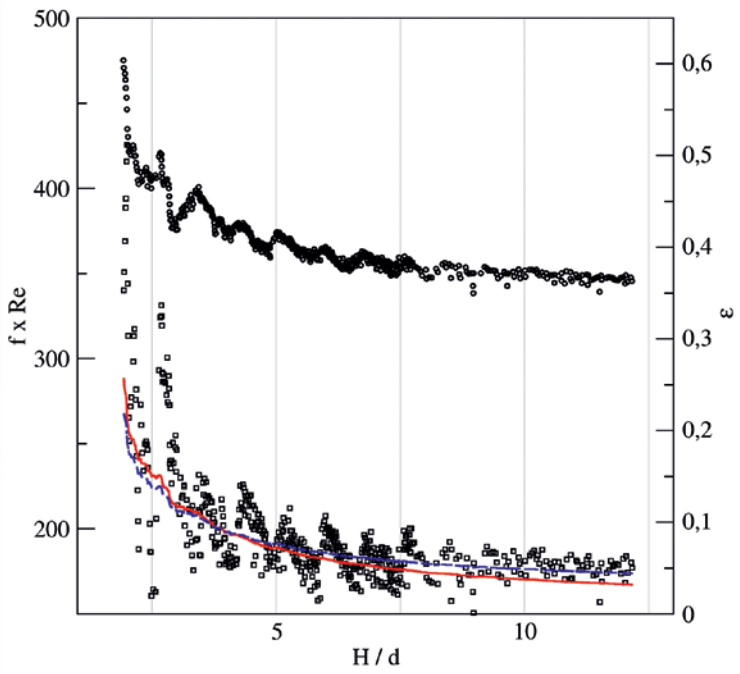


Figure 6: Top: porosity (circles), bottom: dimensionless pressure drop (squares) as function of pipe diameter D to sphere diameter ratio, solid red line: correlation by Eisfeld and Schnitzlein [4], dashed blue line: correlation by Zhavoronkov et al. [5]

bed, the experimental findings and derived correlations differ widely. For example, Ergun proposes a value of 150 for the dimensionless pressure drop, whereas the Carman-Kozeny equation states a value of 180. Most of the research was done experimentally. In the present work, we combine two computer based techniques, the numerical generation of packings and the lattice Boltzmann method to simulate the flow in the artificially created packing. A solely computer based approach allows for simple variation of parameters. In previous work, we compared measured pressure drop of a packing to that calculated with the lattice Boltzmann method and achieved good agreement [3]. As different geometries of the reactor might have an impact on the wall effect, we study not only cylindrical tubes filled with spheres but also rectangular confinings. We used simulated packings of spheres as this approach allow for easy generation of numerous packings, thus allowing to vary parameters easily. To rule out artifacts created by the packing algorithm, two different algorithms are used in this work, namely a modified Lubachevsky-Stillinger algorithm and the algorithm by Desmond and Weeks.

We studied the pressure drop in the flow through four different configurations: An infinite extended packing of spheres, which is modeled by periodic boundaries in all directions, spheres between two infinite extended parallel plates, spheres in a channel with square cross section and a tube filled with spheres. Sample packings of these configurations can be found in Figure 5.

Our simulated dimensionless pressure drop for the infinite extended bed was in good agreement with that predicted by the Carman-Kozeny equation. The pressure drop of all confined geometries tends to oscillate when the spheres become large compared to the diameter or side length of the reactor. These oscillations are unreported in literature but might explain the different results of experimental works as a small change in sphere

diameter to wall distance ratio can lead to a great change in pressure drop. A comparison with correlations for the well-studied case of sphere in a tube show that common correlations average the effect of the oscillations (Figure 6). A paper with detailed results is in preparation.

The cloud-specific part of this project have been developed in scope of predecessor SWZ-Project. A summary of the relevant parts of the predecessor SWZ Projekt 11.4.1 is provided in a separate chapter of this yearbook.

A model of our first new target resource type, the IBM Load Sharing Facility (LSF) has been created using the Eclipse Modeling Framework (EMF). It contains abstractions of the way compute nodes form the cluster and the jobs that can be submitted to it. For example, unlike in a cloud environment, the selection of compute nodes, in particular their CPU and memory resources, is not arbitrary, but subject to the Load Scheduler configuration. Therefore, unlike in an IaaS cloud environment, a job is not in general executed on a set of nodes meeting the exact resource requirements specified, but on the closest match that meets these criteria. In order to make an a priori decision of whether to run the simulation on the LSF cluster at all, or whether for instance a cloud-based infrastructure would be more appropriate, the distribution controller needs this information about the cluster configuration. Should the LSF-based infrastructure be chosen for running the simulation code, the metamodel of a batch job will be instantiated and translated into a jobscript by an M2T transformation.

Outlook

Regarding the lattice Boltzmann method, the aim of the project is twofold: One goal is to extend GPGPU-based parallelization to the AMR method and to implement multiphase models efficiently in the chosen AMR framework. The other aim is to push understanding of flow in and on the boundary layer of porous media especially combined with a chemical reaction in the porous media by application of the developed software. These simulations will also be used as exemplary use cases for evaluation of the developed framework. We are currently developing a resource-agnostic application metamodel based on TOSCA. This metamodel will then be used as an input format for the Distribution Controller.

The distribution layer will fulfill the central task of brokering between the application models of the simulation code on one side and the resource controllers implementing the connection to the various resource types on the other side, be they cloud-based or HPC setups with or without GPU accelerators. This includes evaluating the resource requirements, as they are directly specified by the simulation scientist or can be extracted indirectly from the simulation parameters and input data, and making a suitable choice among the available resource types. Combining the application model with fixed parameters and the infrastructure model yields i.a. the instantiated models relevant for performing the deployment, runtime and termination steps which can be triggered by the simulation scientist using a CLI. In particular, multiple runtime steps with varying parameters are possible, e.g. to realize a parameter study for which no globally best infrastructure choice exists or with an increased memory budget to account for the adaptive mesh refinement that occurred in a previous run.

Finally, we are going to make our framework available to the users and provide documentation and tutorials to facilitate its usage. Another aspect to be ensured in the future is the ongoing user support and maintenance of the software.

References

[1] OASIS (2013). Topology and Orchestration Specification for Cloud Applications (TOSCA) 1.0. [Available online: http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html]

[2] OMG (2014). MDA Guide rev. 2.0. OMG Document ormsc/2014-06-01 [Available Online: http://www.omg.org/cgi-bin/doc?ormsc/14-06-01.pdf].

[3] S. Hofmann, A. Bufe, G. Brenner, T. Turek, Pressure drop study on packings of differently shaped particles in milli-structured channels, Chemical Engineering Science 155 (2016) 376 – 385.

[4] B. Eisfeld, K. Schnitzlein, The influence of confining walls on the pressure drop in

packed beds, Chemical Engineering Science 56 (14) (2001) 4321 – 4329.
[5] N. Zhavoronkov, M. Aerov, N. Umnik, Hydraulic resistance and density of packing of a granular bed, J. Phys. Chem. 23 (1949) 342 – 261
[6] E. Monaco, G. Brenner, Numerische Berechnung von Transport und Reaktionen in einem Fischer-Tropsch-Mikroreaktor mittels der Lattice-Boltzmann-Methode, Chemie Ingenieur Technik, 85(9), 2013, doi: 10.1002/cite.201250675.

Project data

The project will be funded from SWZ with 2 TV-L E13 staff positions from March 2016 to February 2019 at the site Clausthal. Involved scientists are:

Prof. Dr.-Ing. Gunther Brenner
Division of Fluid Dynamics
Institute of Applied Mechanics
Clausthal University of Technology

Prof. Dr. Ramin Yahyapour
Research group Praktische Informatik
Institute of Computer Science
University of Göttingen

Prof. Dr. Jens Grabowski
Research Group for Software Engineering for Distributed Systems
Institute of Computer Science
University of Göttingen

200

Simulationsbasierte Datenfusion & Objektverfolgung

Marcus Baum, Fabian Sigges, Kolja Thormann

Aufgrund steigender Verbreitung und Leistungsfähigkeit moderner Informations- und Kommunikationstechnologien stehen uns immer größere Datenmengen zur Verfügung. Das systematische Extrahieren von Informationen aus diesen Daten ist hierbei eine grundlegende Herausforderung in vielen modernen technischen Gebieten, wie z.B. der Robotik, der Industrie 4.0, dem Autonomen Fahren und der Medizintechnik. In diesem Zusammenhang betrachtet die Datenfusion die systematische Kombination von (verrauschten oder unvollständigen) Daten aus verschiedenen Quellen, mit dem Ziel mehr Information über ein bestimmtes Phänomen zu gewinnen. Datenfusion ist ein interdisziplinäres Forschungsgebiet, welches mit der Signal- und Bildverarbeitung, dem Maschinellem Lernen, der statistischen Datenanalyse und der Schätztheorie verwandt ist. Traditionelle Datenfusionsanwendungen sind in den Bereichen Lokalisierung, Tracking und Navigation mithilfe von Multisensorsystemen zu finden. Aufgrund der steigenden Anzahl an von Computern oder Menschen generierten Daten, werden jedoch neue Anwendungen im Kontext von Big Data und Data Science immer wichtiger.

Die Grundlage zur systematischen Datenfusion bildet ein probabilistisches Modell des betrachteten Problems. Für moderne Anwendungen sind diese Modelle oftmals so komplex (d.h. nichtlinear und hochdimensional), dass keine analytischen Fusionsalgorithmen hergeleitet werden können. Die Grundidee dieses Projekts ist es daher mithilfe von Simulationstechniken, approximative Fusionsalgorithmen für komplexe Problemstellungen zu entwickeln .

In einem ersten Schrittt wird im Rahmen dieses Projekts ein rein datengetriebener Ansatz untersucht: Das zugrundeliegende probabilistische Modell wird verwendet, um eine große Menge an Daten zu erzeugen. Basierend auf diesen Daten wird eine Fusionsregel mithilfe eines Regressions-Algorithmus („Random Forest Regression“) gelernt. Diese Vorgehensweise wurde anhand der Objektverfolgung durch Abstands- und Winkelmessungen illustriert und evaluiert. Es stellt sich heraus, dass der datengetriebene Ansatz insbesondere bei großem Sensorrauschen Standardansätzen überlegen ist.

Simulation-Based Data Fusion & Tracking

Marcus Baum, Fabian Sigges, Kolja Thormann

Introduction

With ongoing advances in modern information- and communication technology, the amount of generated data rises steeply. For this reason, the systematic extraction of usable knowledge from the available data sets is a key challenge in many novel high-tech applications. In this context, the research area of data fusion considers the problem of combining potentially noisy and incomplete data sets from different sources in order to gain knowledge. Data fusion is an interdisciplinary area that overlaps with signal & image processing, machine learning, automated data analysis and state estimation. Traditional applications of data fusion techniques can be found in the areas of localization, tracking, and navigation based on multi-sensor systems. However, with an increasing number of human- and computer-generated data, novel non-traditional applications such as social network and big data analysis are becoming increasingly important.

Statistical data fusion is based on a probabilistic model of the involved quantities. In modern applications, this model is usually extremely complex, e.g., nonlinear and/or high-dimensional, so that it is not possible to derive analytic and efficient fusion algorithms. The basic idea of this project is to bypass this problem with the help of simulations.

In a first step, we investigate a plain data-driven approach [1]: We employ the underlying probabilistic model to simulate a large amount of ground-truth data and observations. Subsequently, a regression algorithm learns a mapping from the observations to the ground-truth. In general, data-driven approaches are becoming increasingly popular in the area of computer vision and pattern recognition. For example, in [2] random forest regression is used to estimate the viewing direction of persons based on generated images. In [3], a deep learning approach is followed to train a tracking algorithm with images. However, for typical data fusion problems, data-driven approaches are still unexplored to a large extent. In this article, we illustrate the above explained simulation-based data-driven approach by considering a traditional data fusion problem, i.e., the tracking of an object based on bearing and distance measurements, as described in the following section.

Example: Tracking Using Bearing-Range Measurements

Due to the vastly increasing interest in autonomous driving, traffic surveillance and mobile robots, a reliable perception of the environment by using sensors is of major importance. Figure 1 illustrates such a scenario, where the objective is to track the two-dimensional Cartesian location of a moving target object over time, e.g., a vehicle. For this purpose, a sensor such as a radar device, collects measurements that consist of (1) the bearing, i.e., angle between the object and a reference direction, and (2) the distance of the object to the sensor. As both measured quantities are usually corrupted with stochastic (Gaussian) noise, a single measurement only specifies the location of the object with uncertainty. Figure 1 illustrates this uncertainty for the Cartesian location obtained from a single bearing-distance measurement using random samples of the noise. The banana-shaped form of the noise in Cartesian space is a result of the Gaussian noise in polar (bearing-angle) space. Hence, it is clear that the problem is highly nonlinear. Intuitively, the task is to fuse several banana-shaped uncertainty regions from different time instants in order estimate the object trajectory. The object is assumed to move approximately linearly, i.e., we assume a so-called (nearly) constant velocity model.

Random Forest Regression for Tracking

Random forest regression [4] is based on decision trees. Each node in a decision tree represents a test that checks if a specific feature of the input

exceeds a threshold. Depending on the answer, the left or right child of the node is followed. This process is performed until the leaf node is reached, which contains a prediction model for the output. A random forest consists of multiple decision trees, where each individual tree is trained with a bootstrap sample from the training data. In this work, we used the Matlab implementation of random forests called TreeBagger.

The basic idea of our approach is to learn a tracker as follows:

1. Simulate a large number of (ground-truth) trajectories and corresponding sensor measurements according to the probabilistic models described above.
2. Train a random forest with the generated data, where the observations are the input and the ground-truth trajectories the output.

Once the training phase is completed, we can use the learned random forest for on-line tracking. By this means, we obtain a batch tracking algorithm that takes all available sensor measurements from all time steps and outputs the complete object trajectory.

Results & Discussion

We compare the results of the random forest tracker with two state-of-the-art tracking methods for bearing-range measurements described in [5]. The first method is a specific version of the Sequential Importance Resampling (SIR) filter tailored to low process noise. The second method transforms the bearing-range measurements to Cartesian measurements in order to render the problem linear so that a Kalman filter (smoother) can be applied.

In the considered evaluation scenario depicted in Figure 1, the target object starts in the lower left corner and its initial position is known to the trackers. The random forest was trained with 40.000 simulations, where 150 trees per forest were chosen. Figure 2 depicts the Root Mean Squared Error (RMSE) for each time step. The RMSE grows over time as the noise becomes larger the farther the object is from the sensor. It can be seen that the random forest tracker shows promising results as it outperforms the other two methods.
Our (preliminary) experiments indicate that a plain simulation-based data-driven approach to data fusion is promising and can beat established problem-specific methods. Of course, our method comes with the disadvantage that it must be trained again if the problem parameters are modified. Additionally, it is not a Bayesian method in the sense that it outputs a posterior density. These disadvantages can probably be tackled by a generalization of the regression problem. Also, we believe that in general a significant larger set of training samples is necessary for which high-performance and big data infrastructures are required.

Outlook

In the future, we will continue to investigate simulation-based approaches for data fusion. In addition to the rather simple illustrative example in this article, we will also consider highly complex settings for which no analytic solutions exist at all. In this work, we took an extreme data-driven approach, i.e., we simulated data to learn the fusion algorithm. However, simulation-based techniques can also be used for performing inference within the Bayesian framework. A problem where this is expected to be especially helpful is the tracking of multiple moving objects such as cars, bicyclists or pedestrians. The main challenge in the so-called multiple object tracking (MOT) problem is that the association of measurements to objects is unknown. By formulating the MOT problem in the Bayesian framework, we aim at calculating the posterior distribution of targets given some sensor measurements. Following Bayes' rule, the posterior is proportional to the product of the prior and the likelihood function. For the MOT problem, the computation of the likelihood function causes already significant problems. Due to the unknown measurement-to-object association, the likelihood function involves a sum over all possible associations, which is intractable for all but small problems. To circumvent the issue of intractable likelihoods, a class of methods called Approximate Bayesian Computation (ABC) or likelihood-free methods [6] was invented in the field of mathematical biology and psychology. ABC algorithms eliminate the need to explicitly calculate the likelihood by means of simulation. A simple ABC algorithm works as follows: A random sample is drawn from the prior distribution. By using the measurement model, a data set fitting to the drawn sample is

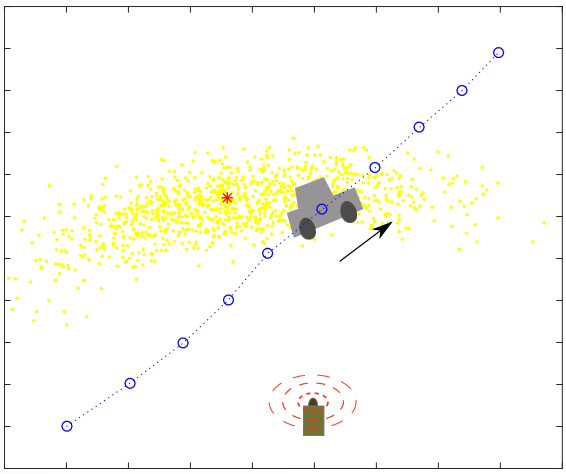


Figure 1. Illustration of the considered problem: The objective is to track an object, e.g., a vehicle, in two-dimensional space. The (discrete) trajectory of the vehicle is indicated by blue circular markers. The sensor collects bearing and distance measurements corrupted with additive Gaussian noise. The banana-shaped uncertainty region for the object location obtained from a single measurement (red asterisk) is illustrated by samples (yellow dots). The size of the tracking area is 110 meters x 110 meters.

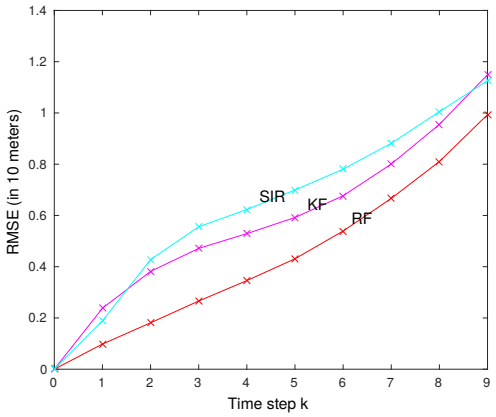


Figure 2. Root Mean Squared Error (RMSE) of the object location for each discrete time instant estimated by an SIR particle filter (blue), modified Kalman filter (magenta), and the random forest estimator (red). The track starts at a known location.

simulated. The simulated data sample set is now compared to the real data sample with some kind of distance function. If the distance between a real data sample and simulated data sample is smaller than a given threshold, we keep the drawn sample, otherwise we discard it. This process is repeated until a suitable number of samples is obtained. The generated samples now represent the posterior distribution. Multiple ABC algorithms using Markov-Chain Monte Carlo (MCMC) methods and Sequential Importance Sampling (SIS) have been invented and applied with great success. So far literature focuses on parameter estimation in the field of biology and psychology. We plan to adopt the idea of simulating the likelihood function and use it in the context of MOT. Therefore, we developed a particle filter including the ideas of ABC. First experiments with simulated data show promising results.

In summary, we believe that simulation-based methods are the key to efficiently solving complex data fusion problems and many interesting research questions are still open and unsolved.

References

[1] Thormann, K., 2016. Random Forest Tracker, Forschungsbezogene Projektarbeit, Institute of Computer Science, University of Göttingen

[2] Fanelli, G., Dantone, M., Gall, J., Fossati, A. and Van Gool, L., 2013. Random forests for real time 3D face analysis. *International Journal of Computer Vision*, *101*(3), pp.437-458.

[3] Ondruska, P. and Posner, I., 2016. Deep tracking: Seeing beyond seeing using recurrent neural networks. *arXiv preprint arXiv:1602.00991*.

[4] Breiman, L., 2001. Random forests. *Machine learning*, *45*(1), pp.5-32.

[5] Romeo, K., Willett, P. and Bar-Shalom, Y., 2015. Particle filter tracking for banana and contact lens problems. *IEEE Transactions on Aerospace and Electronic Systems*, *51*(2), pp.1098-1110.

[6] Csilléry, K., Blum, M.G., Gaggiotti, O.E. and François, O., 2010. Approximate Bayesian computation (ABC) in practice. *Trends in ecology & evolution*, *25*(7), pp.410-418.

Ringvorlesung „Simulationswissenschaften“

Im Rahmen der SWZ Ringvorlesung „Simulationswissenschaften“ berichten externe Gäste über aktuelle Ergebnisse aus ihrem jeweiligen Gebiet. Die Themen erstrecken sich dabei von dem Einsatz von Simulationsmethoden im Flugzeugbau über High-Performance-Computing bis hin zu Verkehrsplanung in Straßen- und Bahnnetzen sowie der Fabrikplanung.

Die Ringvorlesung findet im monatlichen Wechsel in Clausthal und in Göttingen statt. Wann immer möglich werden die Vorträge aufgezeichnet und unter folgende Adresse zum späteren Abruf angeboten:

www.simzentrum.de/lehre/ringvorlesung

Lecture series „Simulation Sciences“

In the SWZ lecture series “Simulation Sciences” external guests are reporting about their latest results in their field of research. The topics extend from the use of simulation methods in the design of aircrafts over high-performance computing to transport planning in road and rail networks and factory planning.

The lectures are hold in Clausthal and in Göttingen. Whenever possible, the lectures are recorded as videos and available for streaming at the following address:

www.simzentrum.de/en/teaching/lecture-series

Übersicht über die bisherigen Vorträge

Übersicht über die bisherigen Vorträge Simulationsbasierte Produktionsplanung in der Halbleiterfertigung: Lösungsansätze und Herausforderungen

Prof. Dr. Lars Mönch
Fakultät für Mathematik und Informatik
FernUniversität in Hagen

Die Produktion von integrierten Schaltkreisen ist einer der komplexesten Produktionsprozesse. Eine einzelne Waferfab umfasst mehrere hundert, zum Teil sehr teure, Maschinen. Bis zu 1000 Jobs werden auf diesen Maschinen bearbeitet. Lange Durchlaufzeiten von bis zu zehn Wochen sind charakteristisch. Reihenfolgeabhängige Umrüstzeiten, schleifenförmige Durchläufe der Jobs, Batchmaschinen, automatisierte Transportsysteme sowie häufige Maschinenausfälle und Nachfrageschwankungen sind typisch. Die diskrete Simulation stellt eine etablierte Methode zur Unterstützung der Produktionssteuerung in Halbleiterfabriken dar. Durch die Bedürfnisse der komplexen Lieferketten in der Halbleiterindustrie motiviert, sind in den letzten Jahren aber Produktionsplanungsansätze stärker in den Mittelpunkt des wissenschaftlichen und praktischen Interesses gerückt. Von besonderem Interesse ist dabei die geeignete Modellierung der lastabhängigen Durchlaufzeiten in linearen Optimierungsformulierungen durch Clearingfunktionen, während konventionelle Formulierungen lediglich feste Durchlaufzeiten als exogene Parameter berücksichtigen. Im Vortrag wird die Leistungsfähigkeit moderner Produktionsplanungsansätze, die Clearingfunktionen verwenden, mit der Leistungsfähigkeit konventioneller Produktionsplanungsansätze verglichen. Der Einsatz von diskreter Simulation zur Bestimmung von Clearingfunktionen sowie zur Bewertung von Produktionsplänen wird diskutiert. Außerdem werden Fragen der simulationsbasierten Leistungsbewertung von Produktionsplanungsansätzen im Rahmen eines rollierenden Ansatzes besprochen. Herausforderungen für den Einsatz von Simulation, die sich insbesondere aufgrund der Größe der Lieferketten in der Halbleiterindustrie für die simulationsbasierten Produktionsplanungsansätze ergeben, werden am Ende des Vortrags dargestellt.

(Der Vortrag fand am 5. März 2014 an der Universität Göttingen statt.)

Von der Mikrostruktur zur Nanostruktur – Werkstoff- und Prozesstechnik für neue Stähle

Prof. Dr.-Ing. Wolfgang Bleck
Institut für Eisenhüttenkunde
RWTH Aachen

Angesichts von mehr als 2200 Stählen in der Werkstoffliste des Stahlinstituts VDEh lässt sich die Frage stellen, welche zukünftigen Anforderungen aus neuen Anwendungsgebieten hervorgehen und welche Optionen für neue Werkstoffentwicklungen bestehen.

Am Beispiel von Anwendungen im Automobilbau und im Bauwesen wird gezeigt, dass vor allem das Engineering von Gefügen auf einer Größenskala unterhalb der Mikroskala attraktive Eigenschaftskombinationen ermöglicht. Neue Untersuchungsmethoden wie die Atom Probe Tomography, High Resolution Transmission Electron Microscopy und die Electron Backscatter Diffraction Methoden resultieren in quantitativen Gefügeinformationen auf der nm-Skala. Die Modellierung, beispielsweise mit ab initio Methoden oder Repräsentativen Volumenelementen, führt zu einem erweiterten Werkstoff- und Prozessverständnis, und daraus abgeleitet zu neuen Produkten. Als Beispiele für neue Werkstoff-Konzepte werden hochfeste kalt umformbare Stähle mit außergewöhnlichem Verfestigungsverhalten, Stähle mit intrinsischer Fehlertoleranz und Stähle mit verbesserter Festigkeits-Zähigkeits-Bilanz besprochen. Die Beispiele für Prozesse betreffen das Warmwalzen mit Auste-

nitkonditionierung und robuster Gefügeeinstellung sowie die Justierung von Phasenumwandlungen bei der Wärmebehandlung.

(Der Vortrag fand am 7. Mai 2014 an der Technischen Universität Clausthal statt.)

Wie kommt Verkehr in den Computer? – Modellierung und Simulation von Verkehrssystemen

Prof. Dr.-Ing. Peter Vortisch
Institut für Verkehrswesen
Karlsruher Institut für Technologie

Bevor eine neue Straße gebaut oder eine neue Ampel in Betrieb geht, wird heutzutage der Verkehr vorher und nachher im Computer simuliert. Diese Simulationsmodelle können im Kleinen abbilden, wie Fahrzeuge über die Kreuzung fahren und aufeinander reagieren, aber auch im Großen, wie viele Pendler sich morgens in eine Stadt bewegen. Wie baut man solche Modelle? Welche Eingangsdaten braucht man dafür und wie bekommt man sie?

Im Vortrag wird zuerst erklärt, wie Verkehr überhaupt entsteht, und wie man erfassen kann, wer sich wann, wo und warum in den Verkehrssystemen bewegt. Anschließend wird die eher mikroskopische Sicht eingenommen: Wie bewegt sich ein Fußgänger? Wie fährt ein Auto dem anderen hinterher? Welche Wege wählen Autofahrer im Straßennetz? Für alle diese Aspekte des Verkehrs werden die gängigen Modellierungsansätze vorgestellt und dargestellt, wo sich die Forschung im Moment bewegt. Begleitend werden einige typische Anwendungen der Simulationsmodelle gezeigt.

(Der Vortrag fand am 4. Juni 2014 an der Universität Göttingen statt.)

Numerische Simulation in der Flugzeugentwicklung – Aktuelle Entwicklungen im DLR

Prof. Dr. Norbert Kroll
Institut für Aerodynamik und Strömungstechnik
DLR Braunschweig

In der Luftfahrtindustrie hat sich die numerische Strömungssimulation (CFD) in den letzten Jahren neben Windkanal- und Flugversuch als unverzichtbarer Bestandteil des aerodynamischen Entwurfsprozesses etabliert. Die kontinuierliche Weiterentwicklung der physikalischen Modelle und numerischen Verfahren sowie die Verfügbarkeit immer leistungsstärkerer Rechner legen nahe, die numerische Simulation wesentlich weitgehender als bisher einzusetzen und den Flugzeugentwicklungsprozess völlig umzugestalten. Daher orientieren sich die aktuellen und zukünftigen Aktivitäten des DLR im Bereich der Verfahrensentwicklung an der Vision eines „Erstflugs im Rechner“ (Digitales Flugzeug).

Primäres Ziel ist die Entwicklung und Bereitstellung einer flexiblen, parallelen Softwareplattform zur multidisziplinären Analyse und Optimierung von Flugzeugen und Hubschraubern unter enger Einbindung von hochgenauen Verfahren aller relevanten Fachdisziplinen. Zum einen soll diese Plattform einen robusten, integrierten Entwurfsprozess von Aerodynamik und Struktur ermöglichen. Das derzeitige noch weitgehend sequentielle Vorgehen im Detailentwurf soll aufgebrochen und das volle Potenzial des multidisziplinären Entwurfs verfügbar gemacht werden. Zum anderen soll mit der Softwareplattform die Möglichkeit geschaffen werden, effiziente und verlässliche Simulationen von Flugmanövern im gesamten Flugbereich einschließlich der Flugbereichsgrenzen durchzuführen und somit die Ermittlung von aerodynamischen und aeroelastischen Datensätzen auf Basis höherwertiger Verfahren zur Bewertung der Flugeigenschaften zu erlauben.

Die sehr aufwendigen, disziplinübergreifenden Simulationen und die Herausforderungen hinsichtlich der physikalischen Modellierung im Bereich der Flugenveloppe erfordern Weiterentwicklungen und Verbesserungen des hybriden DLR-Strömungslösers TAU. Aktuelle Arbeiten zielen darauf ab, die Effizienz, Robustheit und Verlässlichkeit sowie den Automatisierungsgrad des TAU-Codes signifikant zu verbessern und dessen Einsatzspektrum zu erweitern. Vor dem Hintergrund der technologischen Entwicklung der Hochleistungsrechner ist die heute in den meisten Strömungslösern realisierte Parallelisierungsstrategie an die Grenzen der Skalierbarkeit gekommen. Daher sind der

Entwurf und die Implementierung eines Strömungslösers der nächsten Generation ein weiteres wesentliches Ziel zukünftiger Aktivitäten.

Im Vortrag werden einige der zukünftigen Zielanwendungen im Bereich der virtuellen Analyse und Erprobung im Flugzeugentwurf vorgestellt. Basierend auf dem aktuellen Status der CFD-Verfahren werden die wesentlichen Herausforderungen und Anforderungen in der numerischen Simulation abgeleitet und Lösungsansätze für fortschrittliche Simulationsstrategien vorgestellt.

(Der Vortrag fand am 2. Juli 2014 an der Technischen Universität Clausthal statt.)

HPC im Umbruch – Trends und Entwicklungen

Prof. Dr.-Ing. Dr. h.c. Dr. h.c. Michael M. Resch
High Performance Computing Center Stuttgart (HLRS)
Universität Stuttgart

Die theoretische Rechenleistung steigt weiter. Was lange Zeit als Moore's law bezeichnet wurde gilt zwar seit 2004 nicht mehr für einzelne Prozessoren. Trotzdem steigt die Geschwindigkeit von Systemen weiterhin etwa um einen Faktor 2 alle 18 Monate. Daraus ergeben sich neue Möglichkeiten und neue Probleme. In diesem Vortrag sollen diese aufgezeigt und diskutiert werden. Möglichkeiten ergeben sich vor allem durch eine erhebliche höhere Rechenleistung auch für kleinere Systeme. Verbunden damit sind Risiken die uns nicht bewusst sind. Gleichzeitig stellt sich die Frage, ob die Leistungsentwicklung fortgesetzt werden kann. Prognosen, die für das Jahr 2020 ein Exaflop vorhersagen, aber gleichzeitig einen Strombedarf von 50 MW erwarten sind wenig ermutigend. Spezifische Konzepte wie Beschleunigerkarten wirken zunächst wie ein Ausweg, stoßen aber ebenfalls an ihre Grenzen. Alle diese Themen werden angesprochen und in einen Kontext gesetzt. Das HLRS dient dabei als Beispiel für ein Zentrum und die Auswirkungen all dieser Trends auf nationaler Ebene.

(Der Vortrag fand am 2. Juli 2014 an der Technischen Universität Clausthal statt.)

HLRN-III: Hochleistungsrechner für Norddeutschland

PD Dr. Steffen Schulze-Kremer
HLRN und wissenschaftliche Anwendungsbetreuung
Leibniz Universität Hannover

Ende dieses Jahres wird der HLRN-III in der zweiten Ausbaustufe mit 2,5 PetaFlop/s, 85.000 Rechenkernen, 220 TeraByte Hauptspeicher und 8 PetaByte Festplattenspeicher die größte Ressource für wissenschaftliches Rechnen in Norddeutschland sein. Die Länder Berlin, Brandenburg, Bremen, Hamburg, Mecklenburg-Vorpommern, Niedersachsen und Schleswig-Holstein betreiben gemeinsam nun mittlerweile die dritte Generation ihres Hochleistungsrechners, die erfolgreich begutachteten Spitzenforschungsprojekten aus öffentlich-rechtlichen Institutionen ihrer Mitgliedsländer kostenfrei Rechenzeit in großem Umfang zur Verfügung stellt. Der Vortrag stellt die technischen und organisatorischen Besonderheiten des HLRN-III im Kontext des internationalen Supercomputing vor und gibt einen Einblick in seine Nutzung.

(Der Vortrag fand am 16. Juli 2014 an der Technischen Universität Clausthal statt.)

Towards Exascale Simulation Technology

Prof. Dr. Ulrich Rüde
Lehrstuhl für Informatik 10 (Systemsimulation)
Friedrich-Alexander-Universität Erlangen-Nürnberg

Exploiting heterogeneous and hierachically structured extreme scale computer systems to their full capability requires innovation on many levels: New algorithmic paradigms must address unprecedented levels of concurrency and must support asynchronous execution. A new performance-oriented software design technology must be developed to support efficiency, scalability, portability, and flexibility. I will report on our recent work in the waLBerla and HHG frameworks for simulating complex particulate flows based on the lattice Boltzmann method (LBM) and for solving Finite Element Systems using Multigrid Methods. Scalability and performance results for up to a tril-

lion degrees of freedom as well as experiments on accelerator based systems will be presented.

(Der Vortrag fand am 1. Oktober 2014 an der Universität Göttingen statt.)

Modelle und Methoden zur Lösung strategischer Fahrplanprobleme bei DB Netze

Prof. Dr. Karl Nachtigall
Professur für Verkehrsströmungslehre
Technische Universität Dresden

Seit etwa 10 Jahren entwickelt die Professur für Verkehrsströmungslehre im Auftrag von DB Netze prototypische Software zur automatischen Erzeugung und Optimierung von Fahrplänen. Die Programme werden in der Langfristplanung zur Bewertung und Analyse von Infrastrukturmaßnahmen eingesetzt. Als Basis der Modellierung wird ein streng getakteter 2h Fahrplan genutzt. Der Vortrag gibt einen Überblick über die Planungsstufen der strategischen Fahrplanung:

a) streng getaktete 2h-Taktfahrlagen
b) die Integration von Gütersystemtrassen in den Personenverkehr als 24 h Fahrplan
c) Belegungsverfahren, bei denen die als Platzhalter konstruierten Gütersystemtrassen zu konkreten Zugfahrten für Güterverkehrsnachfragen verknüpft werden

Wir stellen die grundlegenden mathematischen Modelle und Lösungsverfahren für diese Anwendungen vor. Streng getaktete Fahrplankonstruktion wird als periodische Ereignisplanung modelliert und dann sehr effizient als Erfüllbarkeitsproblem der Aussagenlogik gelöst. Gütersystemtrassen werden mit linearen Programmen erzeugt. Belegungsverfahren werden durch eine Kombination von linearer Programmierung (Spaltengenerierung) und Heuristiken gelöst.

(Der Vortrag fand am 3. Dezember 2014 an der Universität Göttingen statt.)

Quantitative Morphology-Transport Relationships for Disordered Porous Media by Morphological Reconstruction and High-Performance Computing of Flow and Transport

Prof. Dr. Ulrich Tallarek
Department of Chemistry
Philipps-Universität Marburg

The discovery of the morphology-transport relationships for disordered porous media used in chemical engineering and separation science (packings, monoliths) is a major challenge, because it requires the 3D physical reconstruction and/or computer-generation of the materials followed by 3D mass transport simulations to collect meaningful data for a detailed analysis of morphological and transport properties. This approach is the only direct as well as the most realistic way to understand and optimize materials with applications in chromatography or catalysis. Our latest progress regarding the following issues will be reported: (1) Systematic study of how individual parameters, such as the packing density and packing protocol, affect the morphology of computer-generated packings. (2) Physical reconstruction of packed and monolithic beds to collect information on how experimental parameters of the packing and preparation process influence morphology. (3) 3D mass transport simulations performed on a high-performance computing platform to analyze in detail the hydrodynamics and resulting dispersion. (4) Analysis of computer-generated and physically reconstructed packed and monolithic beds with statistical methods to derive structural descriptors for mass transport (diffusion, dispersion), which have potential for refining the existing theoretical framework.

(Der Vortrag fand am 4. März 2015 an der Technischen Universität Clausthal statt.)

How to order a waiting list?

Dr. Sebastian Stiller
Institut für Mathematik
Technische Universität Berlin

How to order the waiting list for an overbooked flight? This amounts to the problem of packing a knapsack without knowing its capacity. Whenever we attempt to pack an item that does not fit, the item is discarded; if the item fits, we have

to include it in the packing. We show that there is always a policy that packs a value within factor 2 of the optimum packing, irrespective of the actual capacity. If all items have unit density, we achieve a factor equal to the golden ratio R = 1.618. Both factors are shown to be best possible. In fact, we obtain the above factors using packing policies that are universal in the sense that they fix a particular order of the items and try to pack the items in this order, independent of the observations made while packing. We give efficient algorithms computing these policies. On the other hand, we show that, for any R > 1, the problem of deciding whether a given universal policy achieves a factor of R is coNP-complete. If R is part of the input, the same problem is shown to be coNP-complete for items with unit densities. Finally, we show that it is coNP-hard to decide, for given R, whether a set of items admits a universal policy with factor R, even if all items have unit densities. This is joint work with Yann Disser, Max Klimm, and Nicole Megow.

(Der Vortrag fand am 6. Mai 2015 an der Universität Göttingen statt.)

Flüssiger Stahl in silico – numerische Simulation von Prozessen der Stahlherstellung

Prof. Dr.-Ing. Rüdiger Schwarze
Institut für Mechanik und Fluiddynamik
Technische Universität Bergakademie Freiberg

Die Prozesse der Eisen- und Stahlherstellung unterliegen einem permanenten Optimierungsdruck, der aktuell durch ökonomische (möglichst geringe Kosten) und ökologische (möglichst geringer CO2-Ausstoß) Vorgaben definiert wird. Entsprechende Innovationen in der Stahltechnologie erfordern ein vertieftes Verständnis aller beteiligten Prozessstufen, um beispielsweise die Ursachen für eine Qualitätsminderung gezielt bekämpfen zu können. Die Erforschung der Vorgänge in den Prozessstufen der flüssigen Phase ist allerdings aufgrund der physikalischen und chemischen Eigenschaften der Eisen- bzw. Stahlschmelzen sehr schwierig. In der Vergangenheit wurden deswegen vor allem Modellexperimente mit Wasser als Modellschmelze genutzt, um die Strömungen in diesen Prozessstufen zu erforschen. Da die Modellähnlichkeit mit diesem Ansatz nur sehr eingeschränkt erreicht wird, nutzt die moderne Forschung zunehmend die numerische Strömungssimulation (CFD). Mit CFD-Modellen lassen sich auch komplexere physikalische Vorgänge, etwa die elektromagnetische Beeinflussung einer strömenden Stahlschmelze, beschreiben. Im Vortrag wird das anhand verschiedener Beispiele aus den Prozessstufen der Eisen- und Stahlherstellung erläutert. Der aktuelle Forschungsstand und zukünftige Möglichkeiten werden diskutiert.

(Der Vortrag fand am 3. Juni 2015 an der Technischen Universität Clausthal statt.)

Effiziente Lokalisierung für drahtlose Sensornetzwerke

Salke Hartung, M. Sc.
Institut für Informatik
Georg-August-Universität Göttingen

Lokalisierung in Sensornetzwerken bezeichnet die Bestimmung der Position jedes Netzwerkknotens und ist eine grundlegende Notwendigkeit, da gesammelte Daten nur in Verbindung mit ihrer räumlichen Position sinnvoll genutzt werden können. Positionierungssysteme, wie GPS, stellen in der Regel keine Alternative in Sensornetzwerken dar, da die Kapazitäten der einzelnen Knoten nicht ausreichen, um den Betrieb des Netzes langfristig aufrecht zu erhalten. Als Alternative obliegt es den Netzwerkknoten mit Hilfe von Referenzknoten und eines Lokalisierungsalgorithmus ihre Position selbst zu bestimmen. Im Rahmen dieses Forschungsprojektes wurden unter anderem 2 verschiedene Möglichkeiten entwickelt, einen vorhandenen Algorithmus, „Monte Carlo Localization (MCL)“, zu verbessern. In diesem Vortrag werden beide Arbeiten vorgestellt, sowie die laufenden Projekte erläutert.

(Der Vortrag fand am 1. Juli 2015 an der Universität Göttingen statt.)

Simulation der Schallausbreitung in unbegrenzten Räumen

Prof. Dr.-Ing. Stefanie Retka
Juniorprofessur für Computational Dynamics

Institut für Technische Mechanik
Technische Universität Clausthal

Der Vortrag führt in die Anwendung der numerischen Akustik auf komplexe Außenraumprobleme ein. Nach einem kurzen Überblick über die numerischen Grundlagen werden verschiedene Beispiele genutzt, um den Einfluss der Geometrie und deren Modifikation auf die Schallausbreitung und Schallabstrahlung zu verdeutlichen.

Eine Besonderheit unbegrenzter Strukturen liegt in der ersten Eigenfrequenz. Der zugehörige Eigenvektor bildet lediglich ein Viertel einer Wellenlänge ab. Dieser Effekt ist auch als lambda/4 -Effekt bekannt und tritt bei geschlossenen Räumen nicht auf. Weiterhin wird der Einfluss von Änderungen der Raumgeometrie auf die Eigenfrequenzen des Mediums im und um diesen Raum untersucht. Für die oben genannten Effekte werden eine Flasche mit veränderlicher Flaschenhalslänge und -öffnung sowie ein Zimmer mit geöffnetem Fenster, dessen Größe und Position variabel ist, betrachtet.

Abschließend wird der Einfluss der Strömung auf die akustischen Eigenschaften am Beispiel des Klanges einer Blockflöte betrachtet. Die Berücksichtigung eines charakteristischen Strömungsprofils in den numerischen Berechnungen im Frequenzbereich ist eine wesentliche Neuerung und für alle Berechnungen von Bedeutung, bei denen Mechanismen durchströmt bzw. umströmt werden. Exemplarische seien hier Auspuffanlagen oder auch Flugzeugflügel genannt.

(Der Vortrag fand am 7. Oktober 2015 an der Technischen Universität Clausthal statt.)

Große mikroskopische verhaltensorientierte Verkehrssimulationen

Prof. Dr. Kai Nagel
Institut für Land- und Seeverkehr
Technische Universität Berlin

Durch die Fortschritte in der Computertechnik ist es inzwischen recht problemlos möglich, Systeme mit 10^8 Teilchen mikroskopisch zu simulieren. Dies ist aber bereits mehr, als man für die Simulation einer jeden Person und eines jeden Fahrzeugs in einem urbanen oder regionalen Verkehrssystem braucht. Es bleibt also die Aufgabe, die derart definierten synthetischen Personen mit entsprechender Verhaltenslogik auszustatten. Softwaretechnisch ist das mit modernen objekt-orientierten Sprachen recht gut machbar; schwieriger ist es, die Modelle menschlichen Verhaltens zu formulieren. Ein typischer Kunstgriff ist die Annahme eines Nash-Gleichgewichtes, bei dem sich keine reisende Person durch einen unilateralen Wechsel - z.B. auf eine andere Route oder ein anderes Verkehrsmittel - verbessern kann. Dies reduziert die Anforderungen an das Verhaltensmodell, aber um den Preis einer rechentechnisch aufwändigeren Lösungsmethode mit vielen Iterationen, die man als simuliertes menschliches Lernen interpretieren kann, aber nicht muss. Bei anderen Fragestellungen, z.B. bzgl. Betrieb von Taxen oder autonomen Fahrzeugen, ist dieser Kunstgriff ohnehin nicht möglich, und man muss die Reaktionen bestimmter Teilnehmer in simulierter Echtzeit berechnen. Der Vortrag wird beleuchten, was in diesem Rahmen derzeit machbar ist, einschließlich Fragestellungen wie: Was bewirkt eine Autobahnverlängerung in Berlin? Was bewirkt eine Autobahnschließung in Seattle? Welche Emissionen erzeugt das Verkehrssystem in München, und was kann man dagegen tun? Was bewirkt eine Maut im Großraum Johannesburg in Südafrika? Wie simuliert man südafrikanische Minibus-Taxis, wenn man nicht sehr viel über sie weiß? Wie würde man Systeme mit autonomen Fahrzeugen betreiben?

(Der Vortrag fand am 25. November 2015 an der Universität Göttingen statt.)

Dekomposition von Warteschlangennetzen mit Batch-Processing

Dipl.-Wirt.-Inf. Wiebke Klünder
Institut für Angewandte Stochastik und Operation Research
Technische Universität Clausthal

Die Bedeutung von Simulationsmethoden zur Produktionsplanung und -steuerung hat in der industriellen Fertigung in den letzten Jahren ständig zugenommen. Zurückzuführen ist die Zunahme zum einen auf komplexer werdende Fertigungen

und zum anderen auf immer kürzer werdende Fertigungszyklen. Es besteht die Notwendigkeit, dass Unternehmen Fertigungsprozesse von Anfang an optimieren, da nachträgliche Korrekturen am Produktionsprozess in technischer und logistischer Hinsicht schwer umsetzbar sind. Das Entstehen von Warteschlangen ist aus betriebswirtschaftlicher Sicht ein Effekt, der möglichst vermieden bzw. minimiert werden sollte. Eine Warteschlange in einer Produktion bedeutet, dass sich der Produktionsvorgang durch Wartezeiten verlängert, was zu steigenden Lagerhaltungskosten bzw. gebundenem Material führt.

Die analytisch orientierte Methode der Warteschlangentheorie als Produktionsplanungswerkzeug bietet einen Ansatz der Optimierung und besitzt Vorteile gegenüber den etablierten Simulationsmethoden. Die jeweilige Fragestellung, z. B. wieviel Puffer an einer Bedienstation veranschlagt werden sollte, wird mathematisch modelliert und mit Hilfe von analytischen Formeln, die zuvor durch Simulation evaluiert wurden, approximativ gelöst. In diesem Vortrag wird ein Netz aus Produktions- bzw. Bedienstationen betrachtet, in dem Produkttyp-spezifische Aufträge in Stapeln von Bedienern verarbeitet werden. Der Einsatz der Dekompositionsmethode ermöglicht eine isolierte Betrachtung der einzelnen Bedienstationen im Netz, deren Leistungsgrößen anschließend mit Hilfe analytischer Formeln approximativ bestimmt werden können.

(Der Vortrag fand am 2. Dezember 2015 an der Technischen Universität Clausthal statt.)

Agent-Based Modeling and Simulation of Software Processes for Quality Assurance

Dipl.-Inf. Daniel Honsel, Dipl.-Math. Verena Honsel, Marlon Welter, M.Sc.
Institut für Informatik
Georg-August-Universität Göttingen

Project managers have to make decisions concerning the architecture, development strategy or team constellation and they have to estimate the consequences of the decisions. To have tool support for testing the interplay of different parameter constellations would be of great help. We created a simulation tool that can predict possible future scenarios and that makes the project manager aware of risky development trends. For the creation of this tool, understanding software evolution and its drivers is indispensable. To get a model with parameters close to the reality we examined Open Source Software (OSS) projects by repository mining and estimated the simulation's parameters from the gained insights. With the parameters we instantiate an Agent-based simulation of the software project which can be evaluated by project managers using Conditional Random Fields (CRFs) to gain the desired information about development trends.

(Der Vortrag fand am 13. Januar 2016 an der Universität Göttingen statt.)

Optimization problems in DSB long-term and strategic planning

Dr. Natalia Rezanova
Dänische Staatsbahnen (DSB)

Danish State Railways (DSB) is the largest passenger operator in Denmark. DSB Longterm Planning Department is responsible for the long-term traffic planning, including line planning, timetabling and rolling stock (2 years before the day of operations), as well as the strategic traffic planning (10+ years before the day of operations). Many optimization problems arise within the different strategic planning areas. Some of them are solved using commercial optimization software uniquely developed for DSB. Other optimization problems are addressed in-house. The talk will focus on two different optimization projects, where Mixed Integer Programming models formulated, implemented and solved. The line planning optimization model determines an optimal set of train lines, patterns and frequencies, and is used in the strategic planning on DSB S-train network. The facility location optimization model is developed to determine the size and location of maintenance, preparation, cleaning and stabling facilities required to service the fleet of electrical trains, which DSB is planning to acquire in the near future. Both projects, even though quite different in purpose and impact, exploit the advantages of Opera-

tions Research and help to address the real-life challenges of a railway operator.

(Der Vortrag fand am 13. Januar 2016 an der Universität Göttingen statt.)

Verspätungen in Verkehrsnetzen

Prof. Dr. Anita Schöbel
Institut für Numerische und Angewandte Mathematik
Universität Göttingen

Prof. Dr. Michael Kolonko
Institut für Angewandte Stochastik und Operation Research
Technische Universität Clausthal

Jonas Harbering, M.Sc.
Institut für Numerische und Angewandte Mathematik
Universität Göttingen

Die alltäglichen Verspätungen in öffentlichen Verkehrsnetzen können vielfältige Ursachen haben: von klemmenden Türen über unerwarteten Passagierandrang bis zu Baumaßnahmen auf den Gleisen.

Diese kleinen ‚Quellverspätungen' lassen sich im Alltag nicht ganz vermeiden. Ziel einer robusten Fahrplanung ist es daher, durch geschickte Platzierung von Zeitpuffern zu verhindern, dass sich diese Verspätung aufsummieren und über das Netz verbreiten und so zu ernsthaften Verspätungen führen können. Die dazu erforderliche genaue Untersuchung und Simulation des Verkehrsgeschehens unter zufälligen Verspätungen ist aber eine relativ komplexe Aufgabe.

In dem SWZ Projekt „Strukturuntersuchungen zur Entstehung und Fortpflanzung von Verspätungen in Verkehrsnetzen - Modellierung, Simulation und Optimierung eines stochastischen Netzwerkes" wurden von Arbeitsgruppen an der Universität Göttingen und der TU Clausthal verschiedene Ansätze untersucht, mit denen die Auswirkung von Fahrplanpuffern auf die resultierenden Verspätungen der Fahrgäste dargestellt und optimale Pufferallokationen gesucht werden können. Insbesondere konnte in diesem Projekt gezeigt werden, dass die für unterschiedliche Zwecke entwickelten szenario-basierten und stochastischen Modelle zu denselben Ergebnissen führen.

(Der Vortrag fand am 3. Februar 2016 an der Technischen Universität Clausthal statt.)

Modeling, deployment and scaling of simulation applications for and in the cloud

Fabian Glaser, M.Sc.
Institut für Informatik
Universität Göttingen

Often scientific simulations require more computational resources than locally available. While grid computing already offered on-demand access to large-scale distributed computing resources in the past, cloud computing is much more flexible since it offers the possibility to deploy the full hard- and software stack as desired. However, this increased flexibility also places an additional burden on scientists, who are willing to migrate their simulation applications to the cloud, since it requires a deep understanding of the cloud infrastructure and the related technologies. Therefore, our focus in scope of the project "A cloud-based software infrastructure for distributed simulation applications" was to identify the obstacles scientists face when moving their simulation applications to the cloud and develop a framework to simplify this process. The developed solution is based on the Topology and Orchestration Specification for Cloud Applications (TOSCA) and leverages domain-model knowledge of the scientist to scale the deployed simulation infrastructure.

(Der Vortrag fand am 9. März 2016 an der Universität Göttingen statt.)

Simulation-based impact assessment and strategy development in the automotive sector

Prof. Dr. Thomas S. Spengler
Institut für Automobilwirtschaft und Industrielle Produktion
Technische Universität Braunschweig

Electric vehicles are often seen as a promising way for rationalizing the use of fossil energy and cutting down greenhouse gas emissions in the automotive sector. So far, however, the market success of electric vehicles and thus their impact on reducing fossil fuel consumption and greenhouse gas emissions are rather limited. In this talk, the automotive market simulator (AMaSi) is introduced, which combines system dynamics and agent-based simulation to analyze the leverage of manufacturers and policy to support the market diffusion of electric vehicles. Based on real-world data the model is applied to the German car market and different manufacturer and policy measures are simulated, including the currently discussed purchase premium for battery and plug-in hybrid electric vehicles. Additionally, validation issues are discussed when modeling the structure of complex socioeconomic/-technic systems and simulating their behavior.

(Der Vortrag fand am 22. Juni 2016 an der Universität Göttingen statt.)

HPC und Data Science in Göttingen

Prof. Dr. Ramin Yahyapour
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

Forschung gestaltet sich zunehmend rechen- und daten-intensiv, wofür eine geeignete Infrastruktur relevante Voraussetzung ist. Neben der Verfügbarkeit von solchen Infrastrukturen stellt auch das Wissen um den effektiven und effizienten Umgang eine Herausforderung dar. Am Göttingen Campus wird unter dem Begriff eResearch an einer institutionellen Strategie für Computational and Data Science gearbeitet. In dem Vortrag wird Einblick zu den aktuellen Entwicklungen gegeben und Beispiele zu Forschungsfragen geliefert.

(Der Vortrag fand am 26. Oktober 2016 an der Universität Göttingen statt.)

Kostenminimale Flüsse mit beschränktem Budget

Jun.-Prof. Dr. Clemens Thielen
Fachbereich Mathematik
Technische Universität Kaiserslautern

Das Minimalkostenflussproblem ist eines der bekanntesten Probleme der Graphentheorie und besitzt zahlreiche Anwendungen. Für gegebenen Kapazitäten und Kosten pro Flusseinheit auf den Pfeilen eines Netzwerks besteht die Aufgabe in der Bestimmung der kostengünstigsten Möglichkeit, eine vorgegebene Menge eines Gutes von einer gegebenen Quelle zu einer gegebenen Senke durch das Netzwerk zu transportieren. Das Minimalkostenflussproblem stellt somit einen allgemeinen mathematischen Rahmen für viele Distributions- und Transportprobleme dar. Zusätzlich lässt sich auch das Problem der Bestimmung eines maximalen dynamischen Flusses durch ein Netzwerk mit Reisezeiten auf den Pfeilen als Minimalkostenflussproblem formulieren. Daher lassen sich beispielsweise auch Probleme aus dem Bereich der Kapazitätsbestimmung von Abwassernetzen als Minimalkostenflussprobleme lösen.
In vielen Anwendungen des Minimalkostenflussproblems ist jedoch das zu benutzende Netzwerk noch nicht vollständig vorhanden, sondern muss erst (aus-) gebaut werden, bevor es zum Gütertransport verwendet werden kann. Dies motiviert die Erweiterung des Minimalkostenflussproblems um einen zweiten Kostenwert für jeden Pfeil, der die (Aus-) Baukosten des Pfeils pro Kapazitätseinheit beschreibt. Steht nur ein vorgegebenes Budget zum (Aus-) Bau des Netzwerks zur Verfügung, so ergibt sich das Problem, einen Fluss mit minimalen Kosten zu berechnen, der sich in einem Netzwerk mit durch das Budget beschränkten (Aus-) Baukosten realisieren lässt. Das resultierende Netzwerkflussproblem wird als Minimalkostenflussproblem mit Budgetbeschränkung bezeichnet.

Dieser Vortrag beschäftigt sich mit effizienten Algorithmen zur Lösung des Minimalkostenflussproblems mit Budgetbeschränkung. Wir werden zeigen, wie sich kombinatorische Algorithmen für das klassische Minimalkostenflussproblem in Kombination mit binärer oder parametrischer Suche benutzen lassen, um auch das Problem mit Budgetbeschränkung selbst auf großen Netzwerken effizient lösen zu können.

(Der Vortrag fand am 2. November 2016 an der Technischen Universität Clausthal statt.)

Integration based profile likelihood calculation for PDE constrained parameter estimation problems

Prof. Dr. Barbara Kaltenbacher
Institut für Mathematik
Alpen-Adria-Universtät Klagenfurt

Partial diferential equation (PDE) models are widely used in engineering and natural sciences to describe spatio-temporal processes. The parameters of the considered processes are often unknown and have to be estimated from experimental data. Due to partial observations and measurement noise, these parameter estimates are subject to uncertainty. This uncertainty can be assessed using profile likelihoods, a reliable but computationally intensive approach. In this talk, we present the integration based approach for the profile likelihood calculation developed by Chen and Jennrich, 2002, and adapt it to inverse problems with PDE constraints. While exiting methods for profile likelihood calculation in parameter estimation problems with PDE constraints rely on repeated optimization, the proposed approach exploits a dynamical system evolving along the likelihood profile. We derive the dynamical system for the unreduced estimation problem, prove convergence and study the properties of the integration based approach for the PDE case. To evaluate the proposed method, we compare it with state-of-the-art algorithms for a simple reaction-diffusion model for a cellular patterning process. We observe a good accuracy of the method as well as a significant speed up as compared to established methods. While our computational experiments have been done for an application example in systems biology, we emphasize that due to generality of this methodology, integration based profile calculation appears to facilitate rigorous uncertainty analysis for parameter estimation problems with PDE constraints also in many other fields.

(Der Vortrag fand am 7. Dezember 2016 an der Technischen Universität Clausthal statt.)

Lehrangebote an den beiden Partneruniversitäten zum Thema Simulation

Jun.-Prof. Dr.-Ing. Marcus Baum
- Sensor Data Fusion (Universität Göttingen, 501933, Wintersemester 2016/2017)
- Hot Topics in Data Fusion and Analytics (Universität Göttingen, 501934, Wintersemester 2016/2017)
- Simulation-based Data Fusion and Analysis (Universität Göttingen, 502033, Sommersemester 2016)
- Practical Course: Data Fusion (Universität Göttingen, 502082, Sommersemester 2016)

Prof. Dr.-Ing. Gunther Brenner
- Computational Simulation (TU Clausthal, W 8036, Wintersemester 2015/2016 und Wintersemester 2016/2017)
- Simulationsmethoden in den Ingenieurwissenschaften (TU Clausthal, W 8037, Wintersemester 2015/2016 und Wintersemester 2016/2017)

Jun.-Prof. Dr. Anja Fischer
- Kolloquium über Angewandte Mathematik (Universität Göttingen, 500166, Wintersemester 2016/2017)
- Theory of integer programming (Universität Göttingen, 502123, Wintersemester 2016/2017)
- Combinatorial optimization (Universität Göttingen, 501958, Sommersemester 2016)

Prof. Dr. Xiaoming Fu
- Computer Networks (previously Telematik) (Universität Göttingen, 990066, Wintersemester 2016/2017)
- Advanced Computer Networks (Universität Göttingen, 501062, Sommersemester 2016)
- Advanced Topics in Mobile Communications (Universität Göttingen, 501928, Sommersemester 2016)
- Seminar on Internet Technologies (Universität Göttingen, 990039, Sommersemester 2016 und Wintersemester 2016/2017)
- Practical Course Networking Lab (Universität Göttingen, 990086 Sommersemester 2016)
- Practical Course Advanced Networking (Universität Göttingen, 990144, Sommersemester 2016)

Prof. Dr. Leonhard Ganzer
- Numerical Reservoir Simulation (TU Clausthal, S 6102, Sommersemester 2015 und Sommersemester 2016)

Prof. Dr. Jens Grabowski
- Practical Course on Software Testing (Universität Göttingen, 501882, Wintersemester 2016/2017)
- Softwaretechnik (Universität Göttingen, 990045, Wintersemester 2016/2017)
- Software Testing (Universität Göttingen, 990052, Wintersemester 2016/2017)
- Advanced Topics in Software-Engineering (Universität Göttingen, 501216, Sommersemester 2016)
- Requirements Engineering (Universität Göttingen, 501217, Sommersemester 2016)

- Software Evolution (Universität Göttingen, 501562, Sommersemester 2016)
- Practical Course on Parallel Computing (Universität Göttingen, 990179, Sommersemester 2016)
- Data Science and Big Data Analytics (Universität Göttingen, 501425, Wintersemester 2016/2017)

Prof. Dr. Dieter Hogrefe
- Mobile Communications (Universität Göttingen, 990092, Sommersemester 2016)
- Security of Self-organizing Networks (Universität Göttingen, 990125, Sommersemester 2016)

Prof. Dr. Michael Kolonko
- Stochastische Modellbildung und Simulation (TU Clausthal, W 0140, Wintersemester 2015/2016 und Wintersemester 2016/2017)
- Mathematische Methoden des OR: Optimierung und Simulation (TU Clausthal, S 0515, Sommersemester 2016)
- Stochastische Simulation und Statistik (TU Clausthal, S 0260, Sommersemester 2016)

Prof. Dr.-Ing. Dieter Meiners
- Simulation und Modellierung in der Kunststofftechnik (TU Clausthal, S 7920, Sommersemester 2015 und Sommersemester 2015)

Prof. Dr.-Ing. Dietmar P.F. Möller
- Computational Modeling and Simulation (TU Clausthal, W 0506, Wintersemester 2015/2016 und Wintersemester 2016/2017)
- Computational Modeling and Simulation I: Discrete Systems, Transatlantic Course (TU Clausthal, S 0503, Sommersemester 2015 und Sommersemester 2016)
- Grundlagen der Simulationstechnik 1 (TU Clausthal, S 0514, Sommersemester 2016)
- Computational Modeling and Simulation II: Continuous Systems, Transatlantic Course (TU Clausthal, W 0507, Wintersemester 2015/2016)
- Introduction into Stochastic Systems Modeling and Simulation (TU Clausthal, W 0509, Wintersemester 2015/2016)

Prof. Dr. Anita Schöbel
- Seminar on algorithms for (robust) integer programming (Universität Göttingen, 501926, Wintersemester 2016/2017)

Prof. Dr. Stephan Waack
- Algorithmisches Lernen und Probabilistische Datenmodelle (Universität Göttingen, 990002, Wintersemester 2016/2017)
- Conditional Random Fields als Datenstruktur in Bioinformatik und Softwaretechnik (Universität Göttingen, 502106, Sommersemester 2016)
- Grundlagen der Informationstheorie für Informatiker (Universität Göttingen, 990182, Sommersemester 2016)

Prof. Dr. Ramin Yahyapour
- Parallel Computing (Universität Göttingen, 50096, Wintersemester 2016/2017)
- Distributed Storage and Information Management (Universität Göttingen, 50049, Sommersemester 2016)
- Practical Course on Parallel Computing (Universität Göttingen, 990179, Sommersemester 2016)

International Simulation Science Semester

Jeweils im Wintersemester bietet die TU Clausthal das International Simulation Science Semester (IS3) an. Die Veranstaltung wird vom Simulationswissenschaftlichen Zentrum Clausthal-Göttingen (SWZ) und dem Internationalen Zentrum Clausthal (IZC, www.izc.tu-clausthal.de) organisiert. Das englischsprachige Vorlesungsangebot bietet einen sehr guten Einblick in verschiedene Aspekte der Simulationswissenschaften.

Every winter term TU Clausthal (TUC) offers an International Simulation Science Semester (IS3). It is organized jointly by the Simulation Science Center Clausthal / Göttingen (SWZ) and the International Center Clausthal (IZC, www.izc.tu-clausthal.de). This course offer is a great opportunity for students to gain experience and orientation in international study programs required in a globalized world.

Content

Module 1: Introduction into Computational Modeling and Simulation

3 ECTS Credits
Lecturer: Prof. Dr. Dietmar P. F. Moeller (TUC/UHH)

The power of simulation lies in the three R´s of science and engineering, namely: reductionism, repeatability, and refutation. That is why this course is organized as follows: The introductory part focuses on modeling and the essential mathematical methods executed in continuous-time, discrete-time and distributed systems following the fundamental laws in science and engineering. The subsequent training part focuses on the utilization of an industry standard simulation framework and the validation and verification of results of computer-based simulation. To this end, students develop, evaluate and present the simulation results in a plenary workshop at the end of the course.

Module 2: Introduction into Stochastic Systems Modeling and Simulation

3 ECTS Credits (Block Course)
Lecturer: Prof. Dr. Thomas Hanschke (TUC), Dr. Horst Zisgen (IBM)

Stochastic systems modeling and simulation explore stochastic systems which could be defi ned as anything random that changes in time. Stochastic systems are at the core of a number of disciplines in science and engineering, for example communication systems, machine learning, and more. The course will introduce students into the basics of the probability theory: probability spaces, conditional probability time and limits in probability, common probability distributions (binominal, exponential, Poisson, Gaussian), queuing systems models, Markov chains, random processes.

Module 3: Introduction into Computational Modeling and Simulation in Mechanical Engineering

3 ECTS Credits
Lecturer: Prof. Dr. Gunther Brenner (TUC)

In the 21st century decision making increasingly relies on computer simulations. Prominent examples are weather forecasts or predictions of fi nancial or environmental scenarios. Computational methods have become indispensable tools in the context of designing and optimization of processes and products. Therefore, students of engineering sciences have to be made familiar with the ideas of simulation and the use of modern software tools. The goal of the present course is to familiarize students with the basic concepts of computational methods and to provide com-

petences that allow them to utilize these tools in a targeted manner and to assess results critically. Further in-depth knowledge of physical and mathematical details may be imparted subsequently in further lectures.

Module 4: Agent-based Modeling and Simulation
1 ECTS Credit (Block course 2 days)
Lecturer: Prof. Dr. Jörg Müller (TUC)

We are witnesses to growing complexity of today's systems for managing/controlling critical networked infrastructure systems for traffic, logistics, energy, or industry automation. These systems are systems of systems (SoS), i.e., large-scale concurrent and distributed systems that are themselves comprised of complex autonomous systems and that are characterized by operational and managerial independence, geographic distribution, emergent behavior, and evolutionary development. Decentralization and the often stochastic nature of the environment are further properties of such systems. Modeling and simulation of systems of systems require suitable abstractions to express autonomy of systems and often loosely-coupled interaction between these systems. In this lecture, we shall introduce concepts of the multiagent-based modeling and simulation paradigm, which provides these types of abstractions. Starting from the notion of autonomous intelligent agents and multiagent systems, we shall review models for interaction, coordination, and cooperation. Benefi ts but also challenges of agentbased modeling and simulation will be discussed by means of selected application scenarios. In a practical part, attendants of the lecture will have the opportunity to model small examples of multiagent systems using the AgentSpeak modeling language.

Module 5: Transportation Analysis, Modeling and Simulation
3 ECTS Credits
Lecturer: Prof. Dr. Dietmar P. F. Moeller (TUC/UHH)

The transportation systems sector – comprising modes of transportation, each with different operational structures and approaches to security – is a vast, open, interdependent network, moving millions of tons of freight and millions of passengers. Every day, the transportation systems network connects cities, manufacturers, and retailers by moving large volumes of freight and passengers through a complex network of roads and highways, railways and train stations, sea ports and dry ports, and airports and hubs. Thus, the transportation systems sector is the most important component of any modern economy's infrastructure in the globalized world. It is also a core component of daily human life with all of its essential interdependencies, such as demands for travel within a given area and freight transportation in metropolitan areas, which require a comprehensive framework in which to integrate all aspects of the target system. Therefore, transportation systems models enable transportation managers to run their daily businesses safely and more effectively through a smarter use of transportation networks. But the transportation systems sector in today's open, interdependent network encompassing urban and metropolitan areas requires optimization of all operating conditions. This can be successfully achieved if the interactions between transportation modes, the economy, land use, and the impact on natural resources are included in transportation systems planning strategies. But the proposed future of multimodal transportation systems cannot be measured through planning alone. Mathematical models of transportation systems and mobility management, incorporating both realworld and hypothetical scenarios, should be embedded in transportation systems analysis, including the evaluation and/or design the of traffic flows, determining the most reliable mode of operation of physical (e.g., a new road) and organizational (e.g., a new destination) objects, and the interaction between the objects and their impact on the environment. These mathematical models are fundamental to the analysis, planning, and evaluation of small-, medium-, and large-scale multimodal transportation systems. The success of model-based scenario analysis can be evaluated by the resulting forecast or prediction of the transportation system response. An ideal design or operational methodology for a transportation system can be achieved using model-based analysis in conjunction with backcasting or backtracking. Thus, modeling and simulation play a central role in planning, developing, and evaluating multimodal transportation systems, improving transportation efficiency and

keeping pace with the rising demands for optimizing multimodal transportation systems.

Module 6: Student Team Project in Computational Modeling and Simulation
6 ECTS Credits

Student team project, supervisor depends on chosen topic Within the student team project, groups of students which work on a specifi c topic offered by the lecturers of the International Simulation Science Semester are formed. The student team project groups will analyze, present, discuss, and publish (conference or journal) one specifi c topic, such as:

- Dry port development in the maritime domain
- Turnaround optimization at airports
- Mobile autonomous robots in unstructured environments
- Urban mobility concepts for metropolitan areas

Module 7: Intercultural competence seminar
3 ECTS Credits
Various Lecturers

Interacting with people from different cultural backgrounds has become an important part of our daily lives. To benefit from cultural diversity, this course is designed to develop your intercultural competence in two areas: understanding culture and its impact on behavior in an international working environment, and developing communication strategies and skills to work successfully in international teams.

Module 8: Language Training – German A 1.1 Beginners
6 ECTS Credits
Various Lecturers

German language course for beginners or learners with little knowledge of German. This course focuses on developing listening and reading comprehension, active use of German, as well as on acquiring learning techniques and communicative competence needed to study successfully at a German university. Please note that German courses on all levels are available and can be exchanged for this course.

Module 9: Language Training – European and Non-European languages
2–6 ECTS Credits

Apart from German language courses, the course range comprises: Arabic, Brazilian Portuguese, Chinese, English, French, Greek, Italian, Norwegian, Polish, Russian and Spanish. This includes courses for beginners as well as for advanced learners.

Mitglieder // Members

Vorstand des SWZ // Board of Directors

Prof. Dr.-Ing. Gunther Brenner
(Stellvertretender Vorstandsvorsitzender)
Arbeitsgruppe Strömungsmechanik
Institut für Technische Mechanik
Technische Universität Clausthal

Prof. Dr. Thomas Hanschke
Arbeitsgruppe Stochastische Modelle
in den Ingenieurwissenschaften
Institut für Angewandte Stochastik
und Operations Research
Technische Universität Clausthal

Jonas Harbering, M.Sc.
Arbeitsgruppe Optimierung
Institut für Numerische und
Angewandte Mathematik
Georg-August-Universität Göttingen

Prof. Dr. Dieter Hogrefe
Arbeitsgruppe Telematik
Institut für Informatik
Georg-August-Universität Göttingen

Dipl.-Wirt.-Inf. Wiebke Klünder
Arbeitsgruppe Stochastische Modelle
in den Ingenieurwissenschaften
Institut für Angewandte Stochastik
und Operations Research
Technische Universität Clausthal

Prof. Dr. Anita Schöbel
(Vorstandsvorsitzende)
Arbeitsgruppe Optimierung
Institut für Numerische und
Angewandte Mathematik
Georg-August-Universität Göttingen

Beirat des SWZ // Advisors

Herr Nikolaus Bettels
Niedersächsisches Ministerium für Wissenschaft
und Kultur

Prof. Dr. Bernhard Neumair
Steinbuch Centre for Computing
Karlsruher Institut für Technologie

Prof. Dr. Ulrich Rieder
Institut für Optimierung und Operations Research
Universität Ulm

Prof. Dr. rer. nat. Dr. h. c. Kurt Rothermel
Institut für Parallele und Verteilte Systeme
Universität Stuttgart

Univ.-Prof. Dr.-Ing. Sigrid Wenzel
Arbeitsgruppe Produktionsorganisation
und Fabrikplanung
Institut für Produktionstechnik und Logistik
Universität Kassel

**Prof. Dr.-Ing. habil. Dr. h.c. mult.
Peter Wriggers**
Institut für Kontinuumsmechanik
Leibniz Universität Hannover

Mitglieder und Angehörige des SWZ // Members

Professoren

Jun.-Prof. Dr.-Ing. Marcus Baum
Arbeitsgruppe Datenfusion
Institut für Informatik
Georg-August-Universität Göttingen

Prof. Dr.-Ing. Uwe Bracht
Arbeitsgruppe Anlagenprojektierung
und Materialflusslogistik
Institut für Maschinelle Anlagentechnik
und Betriebsfestigkeit
Technische Universität Clausthal

Prof. Dr.-Ing. Gunther Brenner
Arbeitsgruppe Strömungsmechanik
Institut für Technische Mechanik
Technische Universität Clausthal

Prof. Dr. Jürgen Dix
Arbeitsgruppe Computational Intelligence
Institut für Informatik
Technische Universität Clausthal

Jun.-Prof. Dr. Anja Fischer
Arbeitsgruppe Optimierung
Institut für Numerische und
Angewandte Mathematik
Georg-August-Universität Göttingen

Prof. Dr. Xiaoming Fu
Arbeitsgruppe Computer Networks Group
Institut für Informatik
Georg-August-Universität Göttingen

Prof. Dr. Leonhard Ganzer
Arbeitsgruppe Lagerstättentechnik
Institut für Erdöl- und Erdgastechnik
Technische Universität Clausthal

Prof. Dr. Jens Grabowski
Arbeitsgruppe Softwaretechnik
für Verteilte Systeme
Institut für Informatik
Georg-August-Universität Göttingen

Prof. Dr. Thorsten Grosch
Arbeitsgruppe Graphische
Datenverarbeitung und Multimedia
Institut für Informatik
Technische Universität Clausthal

Prof. Dr. Thomas Hanschke
Arbeitsgruppe Stochastische Modelle
in den Ingenieurwissenschaften
Institut für Angewandte Stochastik
und Operations Research
Technische Universität Clausthal

Prof. Dr.-Ing. Stefan Hartmann
Arbeitsgruppe Festkörpermechanik
Institut für Technische Mechanik
Technische Universität Clausthal

Prof. Dr. Dieter Hogrefe
Arbeitsgruppe Telematik
Institut für Informatik
Georg-August-Universität Göttingen

Prof. Dr. Michael Kolonko
Arbeitsgruppe Stochastische Optimierung
Institut für Angewandte Stochastik
und Operations Research
Technische Universität Clausthal

Prof. Dr.-Ing. Dieter Meiners
Institut für Polymerwerkstoffe und Kunststofftechnik
Technische Universität Clausthal

Prof. Dr. Dietmar P.F. Möller
Arbeitsgruppe Stochastische Modelle
in den Ingenieurwissenschaften
Institut für Angewandte Stochastik
und Operations Research
Technische Universität Clausthal

Prof. Dr. Jörg Müller
Arbeitsgruppe Wirtschaftsinformatik
Institut für Informatik
Technische Universität Clausthal

Prof. Dr. Andreas Rausch
Software Systems Engineering
Institut für Informatik
Technische Universität Clausthal

Prof. Dr.-Ing. Dr. rer. nat. habil. Harald Richter
Arbeitsgruppe Technische Informatik
und Rechnersysteme
Institut für Informatik
Technische Universität Clausthal

Prof. Dr. Anita Schöbel
Arbeitsgruppe Optimierung
Institut für Numerische und
Angewandte Mathematik
Georg-August-Universität Göttingen

Prof. Dr. Stephan Waack
Arbeitsgruppe Theoretische Informatik
und Algorithmische Methoden
Institut für Informatik
Georg-August-Universität Göttingen

Prof. Dr. Stephan Westphal
Arbeitsgruppe Diskrete Optimierung
Institut für Angewandte Stochastik
und Operations Research
Technische Universität Clausthal

Prof. Dr. Ramin Yahyapour
Arbeitsgruppe Praktische Informatik
Institut für Informatik
Georg-August-Universität Göttingen

Wissenschaftliche Mitarbeiter

Tobias Ahlbrecht, B.Sc.
Arbeitsgruppe Computational Intelligence
Institut für Informatik
Technische Universität Clausthal

Sebastian Albert
Arbeitsgruppe Optimierung
Institut für Numerische und
Angewandte Mathematik
Georg-August-Universität Göttingen

Alexander Bufe, M.Sc.
Arbeitsgruppe Strömungsmechanik
Institut für Technische Mechanik
Technische Universität Clausthal

Martin Dahmen, M.Sc.
Arbeitsgruppe Diskrete Optimierung
Institut für Angewandte Stochastik
und Operations Research
Technische Universität Clausthal

Niklas Fiekas, B.Sc.
Arbeitsgruppe Computational Intelligence
Institut für Informatik
Technische Universität Clausthal

Fabian Glaser, M.Sc.
Arbeitsgruppe Softwaretechnik
für Verteilte Systeme
Institut für Informatik
Georg-August-Universität Göttingen

Michael Göttsche, M.Sc.
Arbeitsgruppe Software Engineering
for Distributed Systems
Institut für Informatik
Georg-August-Universität Göttingen

Dr. Jonas Harbering
Arbeitsgruppe Optimierung
Institut für Numerische und
Angewandte Mathematik
Georg-August-Universität Göttingen

Dr. Salke Hartung
Arbeitsgruppe Telematik
Institut für Informatik
Georg-August-Universität Göttingen

Dr. Alexander Herzog
Geschäftsführer des
Simulationswissenschaftlichen Zentrums
Arbeitsgruppe Stochastische Modelle
in den Ingenieurwissenschaften
Institut für Angewandte Stochastik
und Operations Research
Technische Universität Clausthal

Dipl.-Math. Verena Herbold
Arbeitsgruppe Softwaretechnik
für Verteilte Systeme
Institut für Informatik
Georg-August-Universität Göttingen

Dipl.-Inf. Daniel Honsel
Arbeitsgruppe Theoretische Informatik
und Algorithmische Methoden
Institut für Informatik
Georg-August-Universität Göttingen

Dipl.-Wirt.-Inf. Wiebke Klünder
Arbeitsgruppe Stochastische Modelle
in den Ingenieurwissenschaften
Institut für Angewandte Stochastik
und Operations Research
Technische Universität Clausthal

Dipl.-Inf. Philipp Kraus
Arbeitsgruppe Wirtschaftsinformatik
Institut für Informatik
Technische Universität Clausthal

Julius Pätzold, M.Sc.
Arbeitsgruppe Optimierung
Institut für Numerische und
Angewandte Mathematik
Georg-August-Universität Göttingen

Dr. Jochen Schulz
Arbeitsgruppe Optimierung
Institut für Numerische und
Angewandte Mathematik
Georg-August-Universität Göttingen

Fabian Sigges, M.Sc.
Arbeitsgruppe Datenfusion
Institut für Informatik
Georg-August-Universität Göttingen

Marlon Welter, M.Sc.
Arbeitsgruppe Theoretische Informatik
und Algorithmische Methoden
Institut für Informatik
Georg-August-Universität Göttingen

Dr. Horst Zisgen
Arbeitsgruppe Stochastische Modelle
in den Ingenieurwissenschaften
Institut für Angewandte Stochastik
und Operations Research
Technische Universität Clausthal

Veröffentlichungen //

Strukturuntersuchungen zur Entstehung und Fortpflanzung von Verspätungen in Verkehrsnetzen – Modellierung, Simulation und Optimierung eines stochastischen Netzwerks

2014

R. Bauer and A. Schöbel. Rules of thumb - practical online strategies for delay management. Public Transport, 6(1):85-105, 2014.

M. Goerigk, M. Knoth, M. Müller-Hannemann, M. Schmidt, and A. Schöbel. The Price of Strict and Light Robustness in Timetable Information. Transportation Science, 48:225-242, 2014.

M. Goerigk and A. Schöbel. Recovery-to-optimality: A new two-stage approach to robustness with an application to aperiodic timetabling. Computers and Operations Research, 2014. to appear.

B. Görder and M. Kolonko. Ranking and selection: A new sequential Bayesian procedure for use with common random numbers. 2014. submitted, http://arxiv.org/abs/1410.6782.

J. Harbering. A Line Planning Model for Delay Resistance. Technical report, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Georg-August Universität Göttingen, 2014.

J. Harbering, F. Kirchhoff, M. Kolonko, and A. Schöbel. Delay propagation in public transport - stochastic modeling meets scenario approach. Technical report, 2014. zur Veröffentlichung vorgesehen.

F. Kirchhoff and M. Kolonko. Modeling delay propagation in railway networks with closed families of distributions. submitted, 2014.

J. Manitz, J. Harbering, M. Schmidt, T. Kneib, and A. Schöbel. Network-based source detection for train delays on railway systems. Technical Report 2014-01, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Georg-August Universität Göttingen, 2014.

J. Manitz, J. Harbering, M. Schmidt, T. Kneib, and A. Schöbel. Network-based source detection: From infectious disease spreading to train delay propagation. In Proceedings of the 29th International Workshop on Statistical Modelling, 2014.

F. Kirchhoff. Modelling delay propagation in railway networks. In Operations Research Proceedings 2013, pages 237 - 242, 2014.

M. Schmidt and A. Schöbel. Timetabling with passenger routing. OR Spectrum, pages 1-23, 2014.

Z. Wu and M. Kolonko. Asymptotic properties of a generalized cross entropy optimization algorithm. IEEE Transactions on Evolutionary Computation, 18:1-16, 2014.

Z. Wu and M. Kolonko. Absorption in model-based search algorithms for combinatorial optimization. In Evolutionary Computation (CEC), 2014 IEEE Congress on, pages 1744-1751. IEEE, 2014.

2013

P. Bouman, M. Schmidt, L. Kroon, and A. Schöbel. Passenger route choice in case of disruptions. In Proceedings of the 16th International IEEE Conference on Intelligent Transport Systems (IEEE-ITSC), 2013. http://www.computr.eu/wp-content/uploads/2013/10/IEEE-ITS2013-PassengerChoice.pdf.

E. Carrizosa, J. Harbering, and A. Schöbel. The Stop Location Problem with Realistic Traveling Time. In Daniele Frigioni and Sebastian Stiller, editors, 13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, volume 33 of OpenAccess Series in Informatics (OASIcs), pages 80-93, Dagstuhl, Germany, 2013. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

T. Dollevoet, D. Huisman, L. Kroon, M. Schmidt, and A. Schöbel. Delay management including capacities of stations. Transportation Science, 2013. Available online before print, http://

dx.doi.org/10.1287/trsc.2013.0506.
M. Goerigk, S. Heße, M. Müller-Hannemann, M. Schmidt, and A. Schöbel. Recoverable Robust Timetable Information. In Daniele Frigioni and Sebastian Stiller, editors, 13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, volume 33 of OpenAccess Series in Informatics (OASIcs), pages 1-14, Dagstuhl, Germany, 2013. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
M. Goerigk and A. Schöbel. Algorithm engineering in robust optimization. Technical report, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Universität Göttingen, 2013.
M. Hintermayer. Column Generation in der Linienplanung. Bachelor Thesis, 2013.
T. Jung. Algorithmischer Ausbau von Schienennetzen - Optimierung im Delay Management. Bachelor Thesis, 2013.
S. Schäfer. A new model for line planning including improvement of infrastructure. Master's thesis, Fakultät für Mathematik und Informatik, Georg August Universität Göttingen, 2013.
A. Schöbel and S. Schwarze. Finding delay-resistant line concepts using a game-theoretic approach. Netnomics, 14(3):95-117, 2013.
R. Wichmann. Online delay management - Heuristiken und Experimente. Bachelor Thesis, 2013.

2012

Z. Yang. Modellierung und Simulation von Verspätungsdaten mit PH-Verteilungen. Masterarbeit, TU Clausthal, 2012.

Sichere Kommunikation in Internet of Things (IoT) Umgebungen

2014

D. Koll, J. Li, and X. Fu, "SOUP: An Online Social Network By The People, For The People," in Proc. of 15th Annual ACM/IFIP/USENIX Middleware Conference (Middleware 2014), Bordeaux, France. (acceptance rate: 18.75%), December 2014.
Q. Li, X. Zhang, Q. Zheng, R. Sandhu, and X. Fu, "LIVE: Lightweight Integrity Verification and Content Access Control for Named Data Networking," IEEE Transactions on Information Forensics and Security, IEEE, to appear., October 2014.
L. Jiao, J. Li, T. Xu,W. Du, and X. Fu, "Optimizing Cost for Online Social Networks on Geo-Distributed Clouds," IEEE/ACM Transactions on Networking (accepted), September 2014.
M. Arumaithurai, J. Chen, E. Monticelli, X. Fu, and K. K. Ramakrishnan, "Exploiting ICN for Flexible Management of Software-Defined Networks," in Proc. of 1st ACM Conference on Information-Centric Networking (ICN 2014), Paris, France (Won the Best Paper Award, acceptance rate: 17%), September 2014.
W. Du, Y. Liao, N. Tao, P. Geurts, X. Fu, and G. Leduc, "Rating Network Paths for Locality-Aware Overlay Construction and Routing," IEEE/ACM Transactions on Networking, July 2014.
S. Hartung, S. Taheri, and D. Hogrefe, "Sensor-Assisted Monte Carlo Localization for Wireless Sensor Networks," in 6th IEEE International Conference on Cyber Technology (CYBER), Hong Kong, HK, June 2014.
S. Hartung, A. Kellner, A. Bochem, and D. Hogrefe, "Sensor-Assisted Monte Carlo Localization for Wireless Sensor Networks," in 6th IFIP International Conference on New Technologies, Mobility and Security (NTMS) - Poster + Demo Session, Dubai, UAE, April 2014.

2013

S. Hartung, H. Brosenne, and D. Hogrefe, "Practical RSSI Long Distance Measurement Evaluation in Wireless Sensor Networks," in The 2013 IEEE Conference on Wireless Sensors (ICWiSe 2013), Kuching, Malaysia, December 2013.
S. Taheri and D. Hogrefe, "Robust and Scalable Secure Neighbor Discovery for Wireless Ad Hoc Networks," in The 2013 IEEE International Conference on Communications (ICC 2013), Budapest, Hungary, June 2013.

Eine Cloud-basierte Software Infrastruktur für verteilte Simulation

2016

F. Glaser, Domain Model Optimized Deployment and Execution of Cloud Applications with

TOSCA , Proceedings of the 9th System Analysis and Modelling Conference (SAM 2016), Saint-Malo, France, 2016.

2015

F. Glaser, J. N. Serrano, J. Grabowski, A. Quadt, ATLAS user analysis on private cloud resources at GoeGrid , Proceedings of the 21st Conference on Computing in High Energy Physics and Nuclear Physics (CHEP 2015), 13-17 April, Okinawa, Japan, available online: http://iopscience.iop.org/article/10.1088/1742-6596/664/2/022020 , 2015

F. Glaser, Towards Domain-Model Optimized Deployment and Execution of Scientific Applications in Cloud Environments , Proceedings of the Doctoral Symposium at the 5th Conference on Cloud Computing and Services Sciences (DCCLOSER 2015), Lisbon, Portugal, 2015.

M. Göttsche, F. Glaser, S. Herbold, J. Grabowski, Automated Deployment and Parallel Execution of Legacy Applications in Cloud Environments, Proceedings of the 8th IEEE International Conference on Service Oriented Computing & Applications (SOCA 2015), Rom, Italy, 2015.

H. Richter, About the Suitability of Clouds in High-Performance Computing, to be published in Proc. ISC Cloud&Big Data, Sept. 28–30, Frankfurt, Germany, 2015.

H. Richter and A. Keidel and R. Ledyayev, Über die Eignung von Clouds für das Hochleistungsrechnen (HPC), in Ifl Technical Report Series ISSN 1860-8477, Ifl-15-03, editor: Department of Computer Science, Clausthal University of Technology, Germany, 2015.

2014

R. Ledyayev, H. Richter, High Performance Computing in a Cloud Using OpenStack, The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization, CLOUD COMPUTING 2014, http://www.iaria.org/conferences2014/CLOUDCOMPUTING14.html, Venice, Italy, 6 pages, May 25 - 29, 2014.

H. Richter, A. Keidel, Hochleistungsrechnen und Echtzeit in virtualisierten Maschinen und Clouds - Die Intel Virtualisierungshilfen, in Ifl Technical Report Series ISSN 1860-8477, Ifl-14-03, http://www.in.tu-clausthal.de/forschung/technical-reports/, editor: Department of Computer Science, Clausthal University of Technology, Germany, 44 pages, 2014.

H. Richter, A. Keidel, Analyses and Methods of High-Performance Computing with Cloud Technology, in Ifl Technical Report Series ISSN 1860-8477, http://www.in.tu-clausthal.de/forschung/technical-reports/, editor: Department of Computer Science, Clausthal University of Technology, Germany, 2014.

2013

F. Glaser, H. Neukirchen, T. Rings, J. Grabowski, Using MapReduce for High Energy Physics Data Analysis , Proceedings of the 2013 International Symposium on MapReduce and Big Data Infrastructure (MR.BDI 2013), 03-05 December 2013, Sydney, Australia 2013, DOI: http://dx.doi.org/10.1109/CSE.2013.189 , 201

Simulationsbasierte Qualitätssicherung für Software Systeme

2016

V. Honsel, S. Herbold and J. Grabowski. Learning from Software Project Histories: Predictive Studies Based on Mining Software Repositories, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML-PKDD) 2016 - NEKTAR Track.

D. Honsel, V. Honsel, M. Welter, S. Waack and J. Grabowski. Monitoring Software Quality by Means of Simulation Methods, 10th International Symposium on Empirical Software Engineering and Measurement (ESEM), 2016.

V. Honsel, S. Herbold and J. Grabowski. Hidden Markov Models for the Prediction of Developer Involvement Dynamics and Workload, 12th International Conference on Predictive Models and Data Analytics in Software Engineering (PROMISE 2016).

2015

V. Honsel, D. Honsel, J. Grabowski and S. Waack. Developer Oriented and Quality Assurance Based Simulation of Software Processes,

Proceedings of the Seminar Series on Advanced Techniques & Tools for Software Evolution (SATToSE) 2015.
V. Honsel, Statistical Learning and Software Mining for Agent Based Simulation of Software Evolution, Doctoral Symposium at the 37th International Conference on Software Engineering (ICSE 2015), Florence, Italy.
V. Honsel, D. Honsel, S. Herbold, J. Grabowski and S. Waack; Mining Software Dependency Networks for Agent-Based Simulation of Software Evolution, The 4th International Workshop on Software Mining, 2015.

2014

V. Honsel, D. Honsel, J. Grabowski. Software Process Simulation Based on Mining Software Repositories; Proceedings of the Third International Workshop on Software Mining (accepted), 2014
Z. Dong, K. Wang L. Dang, M. Gültas, M. Welter, T. Wierschin, M. Stanke, S. Waack. CRF-based models of protein surfaces improve protein-protein interaction site predictions; BMC Bioinformatics 15, 277, 2014.

2013

P. Makedonski, F. Sudau, and J. Grabowski. Towards a model-based software mining infrastructure, Second International Workshop on Software Mining (SoftMine'13), 2013

DeSim: Dezentrale Architekturen und Konzepte für die Simulation von Systems of Systems

2016

Tobias Ahlbrecht, Jürgen Dix, Niklas Fiekas, Michael Köster, Philipp Kraus, Jörg P. Müller: An architecture for scalable simulation of systems of cognitive agents. IJAOSE 5(2/3): 232-265 (2016)
- Malte Aschermann, Philipp Kraus, Jörg P. Müller LightJason: A BDI Framework Inspired by Jason Technical Report Ifl-16-04, Clausthal University of Technology, November 2016.

2015

Ahlbrecht T; Dix J; Schlesinger F. From Testing Agent Systems to a Scalable Simulation Platform. In T. Eiter et al. (Eds.): Brewka Festschrift, LNAI 9060, pp. 47-62. Springer International Publishing Switzerland, 2015.

2014

Ahlbrecht T; Dix J; Köster M; Kraus P; Müller JP. A scalable runtime platform for multiagent-based simulation. In F. Dalpiaz et al., editors, Engineering Multiagent Systems II, volume 8758 of Lecture Notes in Artificial Intelligence (LNAI), pages 81-102, Switzerland, 2014. Springer International Publishing.
Bulling N; Popovici M. A game-theoretic approach to compute stable topologies in mobile ad hoc networks. Journal of Logic and Computation (2014).
Bulling N; A survey of multi-agent decision-making. KI 28, 3 (2014), 147–158.
Dalpiaz F; Dix J; van Riemsdijk B, Eds. Engineering Multi-Agent Systems - Second International Workshop, EMAS 2014, Paris, France, May 5–7, 2014, Revised Selected Papers (2014), vol. 8758 of Lecture Notes in Computer Science, Springer.
Gernert B; Schildt S; Wolf L; Zeise B; Fritsche P; Wagner B; Fiosins M; Manesh R; Müller JP (2014). An interdisciplinary approach to autonomous team-based exploration in disaster scenarios. In Proceedings of 12th IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR 2014). IEEE Press.

2013

Bulling N; Dastani M; Knobbout M. Monitoring norm violations in multi-agent systems. In International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Saint Paul, MN, USA, May 6-10, 2013 (2013), M. L. Gini, O. Shehory, T. Ito, and C. M. Jonker, Eds., IFAAMAS, pp. 491–498.
Fiosina J; Fiosins M; Müller JP (2013). Decentralised cooperative agent-based clustering in intelligent traffic clouds. In M. Klusch, M. Paprzycki, and M. Thimm, editors, Multiagent System Technologies: Proceedings of the 11th

German Conference on Multiagent System Technologies, volume 8076 of Lecture Notes in Artificial Intelligence (LNAI), pages 59-72. Springer Berlin Heidelberg, 2013.

Fiosina J; Fiosins M; Müller JP (2013). Mining the traffic cloud: Data analysis and optimization strategies for cloud-based cooperative mobility management. In J. Casillas et al., editors, Management Intelligent Systems, volume 220 of Advances in Intelligent Systems and Computing, pages 25-32. Springer Berlin Heidelberg.

Fiosins M; Müller JP; Huhn M (2013). A norm-based probabilistic decision-making model for autonomic traffic networks. In J. M. Corchado et al., editors, Highlights on Practical Applications of Agents and Multi-Agent Systems, volume 365 of Communications in Computer and Information Science, pages 49-60. Springer Berlin Heidelberg.

2012

Mechs S; Müller JP; Lamparter S; Peschke J (2012). Networked priced timed automata for energy-efficient factory automation. In Proceedings of the 2012 American Control Conference (ACC 2012), pages 5310-5317. IEEE Press.

Multi-Level Simulation

2016

S. Wittek, M. Göttsche, A. Rausch, J. Grabowski. Towards Multi-Level-Simulation using Dynamic Cloud Environments, SIMULTECH 2016 - 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, 2016 (S. 297-303, DOI: 10.5220/0005997502970303).

231

Postanschrift
Simulationswissenschaftliches Zentrum Clausthal-Göttingen
Geschäftsstelle
Dr. Alexander Herzog
Arnold-Sommerfeld-Straße 6
38678 Clausthal-Zellerfeld
E-Mail: alexander.herzog@tu-clausthal.de
Telefon: +49 5323 72-2966
Fax: +49 5323 72-5299

Impressum
Herausgeber: Simulationswissenschaftliches Zentrum Clausthal-Göttingen
Redaktion: Dr. Alexander Herzog
Layout und Satz: Melanie Bruchmann, TU Clausthal
Bildnachweis: Titelbild – Fotolia.com (© marigold_88)

Hier nicht erwähnte Bilder stammen von den Autoren.

ISBN 978-3-946340-79-9

www.simzentrum.de