



SWZ  Clausthal - Göttingen
Simulationswissenschaftliches Zentrum

**Bericht
2013–2014**

Bericht 2013–2014

Simulationswissenschaftliches Zentrum
Clausthal-Göttingen



Vorwort

Bei dem vorliegenden Berichtsband 2013–2014 des Simulationswissenschaftlichen Zentrums Clausthal-Göttingen (SWZ) handelt es sich um den ersten Jahresbericht des am 1.1.2013 von den beiden Partneruniversitäten TU Clausthal und Georg-August-Universität Göttingen gegründeten Zentrums. In dem Bericht werden die Forschungstätigkeiten im Bereich Simulation an den beiden Standorten des SWZ vorgestellt.

In seinem Positionspapier „Bedeutung und Weiterentwicklung von Simulation in der Wissenschaft“¹ vom Juli 2014 hebt der Wissenschaftsrat die Bedeutung der Themen, die am SWZ untersucht werden, für den Fortschritt in vielen Wissensgebieten hervor. Um genau dieses geforderte Repertoire an Simulationsmethoden sowohl an den beiden beteiligten Partneruniversitäten als auch ganz allgemein für den Wissenschaftsstandort Deutschland weiter auszubauen, wird das SWZ in den Jahren 2013 bis 2017 vom Niedersächsischen Ministerium für Wissenschaft und Kultur (MWK) mit insgesamt 4,3 Millionen Euro als Anschubfinanzierung für Forschungsprojekte gefördert. Das SWZ Clausthal-Göttingen ist aus einer Clausthaler Vorläufereinrichtung hervorgegangen. Bereits diese Vorläufereinrichtung wurde 2009 vom BMBF in der Foresight Studie² als einer der aktivsten Akteure in Deutschland auf dem Gebiet der Simulation benannt. Durch die jetzt erfolgte Erweiterung zu einem von zwei Universitäten getragenen Zentrum wurde das Spektrum der bereits in Clausthal etablierten Themen durch den nun standortübergreifenden Ansatz erheblich erweitert.



Prof. Dr. Thomas Hanschke
Sprecher des Vorstands des Simulationswissenschaftlichen Zentrums
Clausthal-Göttingen

Die im Folgenden beschriebenen Projekte stammen aus den beiden Forschungsfeldern „Simulation und Optimierung von Netzen“ und „Verteilte Simulation“. Die 2015 neu startenden Projekte bauen diese Themenfelder weiter aus und werden außerdem das dritte für das SWZ vorgesehene Themengebiet „Simulation von Materialien“ initiieren.

Die Projektförderung durch das SWZ wird dabei als Anschubfinanzierung zur Einrichtung von interdisziplinär und standortübergreifend arbeitenden Arbeitsgruppen und zur Einwerbung von Forschungsprojekten, wie etwa DFG Graduierten Kollegs, Forschergruppen, EU Projekten und Industrieprojekten verstanden.

Um die Forschung am SWZ sichtbar zu machen, wurde neben der Veröffentlichung dieses Jahresberichts die Ringvorlesung „Simulationswissenschaften“³ eingerichtet, in deren Rahmen im monatlichen Wechsel in Clausthal und in Göttingen Vorträge zu verschiedenen Themen von externen Gästen gehalten werden. Des Weiteren ist das SWZ auch bereits durch das in Kooperation mit dem Internationalen Zentrum Clausthal (IZC) organisierte „International Simulation Science Semester“⁴ in der Lehre an der TU Clausthal vertreten. In Göttingen wird 2015 eine durch das SWZ finanzierte Junior-Profeur in der Informatik eingerichtet, so dass auch dort eine Verzahnung des SWZ in Forschung und Lehre stattfindet.

¹ <http://www.wissenschaftsrat.de/download/archiv/4032-14.pdf>

² [http://www.bmbf.de/pubRD/Foresight-Prozess_BMBF_Zukunftsfelder_neuen_Zuschnitts\(5\).pdf](http://www.bmbf.de/pubRD/Foresight-Prozess_BMBF_Zukunftsfelder_neuen_Zuschnitts(5).pdf)

³ <http://www.simzentrum.de/lehre/ringvorlesung/>

⁴ <http://www.simzentrum.de/lehre/iss/>

Introduction

The present report band 2013–2014 of the Simulation Science Center Clausthal-Göttingen (SWZ) has been the first annual report since the center was founded beginning of 2013 by the two partner universities TU Clausthal and Georg-August-University Göttingen. In the report the research in simulation at the two sites of the SWZ is presented.

In its position paper *“Bedeutung und Weiterentwicklung von Simulation in der Wissenschaft”*¹ (The importance and development of simulation in science) from July 2014 the Wissenschaftsrat underlines the importance of the research areas to be examined at the SWZ for the progress in many fields of science. To expand this required repertoire of simulation methods, at both the two involved partner universities and for research in Germany, the SWZ is funded by the Niedersächsischen Ministerium für Wissenschaft und Kultur (MWK) in the years 2013–2017 with a total of 4.3 million euros for initiating research projects. The SWZ Clausthal-Göttingen emerged from a precursor center at TU Clausthal. Already this precursor institution was named in 2009 by the BMBF Foresight Study² as one of the most active players in Germany in the field of simulation. By expanding the SWZ to a center carried by two universities the spectrum of topics has been greatly enhanced.

The projects described below are from the two research areas *“Simulation and optimization of networks”* and *“Distributed Simulation”*. The projects starting in 2015 will continue developing these topics and will also initiate the third planned SWZ research area *“Simulation of Materials”*. The project funding by the SWZ is understood as initial funding for establishing interdisciplinary and cross-location operating research groups and to acquire research projects, such as DFG graduate schools, research groups, EU and industrial projects.

To make the research at SWZ visible, in addition to the publication of annual reports, the *lecture series “Simulation Sciences”*³ was set up in which at a monthly change between Clausthal and Göttingen lectures on various topics are held by external guests. Furthermore, the SWZ is also represented in teaching at TU Clausthal by the *“International Simulation Science Semester”*⁴ which is organized in cooperation with the International Center Clausthal (IZC). In computer science at the University of Göttingen a junior professorship funded by the SWZ will be set up in 2015 so that research and teaching gets stronger connected within the SWZ in Göttingen, too.



Prof. Dr. Thomas Hanschke
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.bmbf.de/pubRD/Foresight-Prozess_BMBF_Zukunftsfelder_neuen_Zuschnitts\(5\).pdf](http://www.bmbf.de/pubRD/Foresight-Prozess_BMBF_Zukunftsfelder_neuen_Zuschnitts(5).pdf)

³ <http://www.simzentrum.de/en/teaching/lecture-series/>

⁴ <http://www.simzentrum.de/en/teaching/international-simulation-science-semester/>

Inhalt

Vorwort	2
Simulation und Optimierung von Netzen	6
Strukturuntersuchungen zur Entstehung und Fortpflanzung von Verspätungen in Verkehrsnetzen	8
Verspätungsfortpflanzung in Verkehrsnetzen – Modellierung und Berechnung mit abgeschlossenen Familien von Wahrscheinlichkeitsverteilungen	20
Sichere Kommunikation in Internet der Dinge Umgebungen	24
Diffusionsapproximation und Dekomposition von Warteschlangennetzen mit Batch-Processing	42
Verteilte Simulation	62
Eine cloud-basierte Software für verteilte Simulation <i>Teilprojekt – Leistungsfähigkeit der Cloud bei verteilter Simulation</i>	64
Eine cloud-basierte Software für verteilte Simulation <i>Teilprojekt - Schnittstellenentwicklung und Zuverlässigkeitstechnik</i>	80
Simulationsbasierte Qualitätssicherung für Software Systeme	90
DeSim: Dezentrale Architekturen und Konzepte für die Simulation von Systems of Systems	106
Ringvorlesung „Simulationswissenschaften“	120
Übersicht über die bisherigen Vorträge	122
Mitglieder	129
Vorstand des SWZ	129
Beirat des SWZ	129
Mitglieder und Angehörige des SWZ	130
Veröffentlichungen	132

Table of contents

Introduction	3
Simulation and Optimization of Networks.....	7
Delays in public transport: formation and propagation of delays in networks.....	9
Delay Propagation in Transport Networks – Modeling and Calculation using Closed Families of Probability Distributions.....	21
Securing Communications in Internet of Things (IoT) Environments.....	25
Diffusion approximation and decomposition of queueing networks with batch processing	43
Distributed Simulation	63
A Cloud-Based Software Infrastructure for Distributed-Simulation <i>Sub Project – Cloud Performance for Distributed-Simulation</i>	65
A Cloud-Based Software Infrastructure for Distributed-Simulation <i>Sub Project – Interface Development and Reliability Engineering</i>	81
Simulation-Based Quality Assurance for Software Systems	91
DeSim: Distributed Architectures and Concepts for the Simulation of Systems of Systems.....	107
Lecture series „Simulation Sciences“	121
International Simulation Science Semester	126
Members.....	129
Board of Directors	129
Advisors	129
Members.....	130
Publications.....	132

Simulation und Optimierung von Netzen

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.

Simulation and Optimization of Networks

By now simulation is one of the most important and in many cases the only viable technique for analysis and optimization of large networks. Telecommunication 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.



Strukturuntersuchungen zur Entstehung und Fortpflanzung von Verspätungen in Verkehrsnetzen

Anita Schöbel, Michael Kolonko, 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 – zumindest bei der Bahn – zu einem Ausweichen auf den Individualverkehr. Auf Seiten der Betreiber stören Verspätungen den Betriebsablauf, es entstehen zusätzlich Produktivitätsverluste durch unnötiges Freihalten knapper Ressourcen wie Bahnsteige bzw. Rollbahnen. Verspätungen bei den zentralen Verkehrsmitteln haben daher gravierende negative Auswirkungen sowohl auf die Mobilität der Gesellschaft wie auch auf die Wirtschaftlichkeit der Verkehrsbetriebe.

Einzelne Verspätungen entstehen zunächst typischerweise durch externe Störungen wie z.B. besondere Wetterereignisse, technische Störungen, Verzögerungen beim Be- und Entladen bzw. beim Ein- und Aussteigen. Diese primären Verspätungen übertragen sich auf weitere Züge oder Flugzeuge, so dass eine einzelne Verspätung zu einer ganzen Kaskade von Folgeverspätungen in dem betrachteten Netzwerk führen kann. Die Übertragungen entstehen dabei durch das Warten auf verspätete Zubringer oder die verspätete Freigabe einer blockierten Ressource (z.B. einer Rollbahn oder eines Gleisabschnitts). Verschiedene Maßnahmen können einer Verbreitung der Verspätungen entgegenwirken: das Einplanen von Zeitpuffern in die Fahrpläne, die Entkopplung der Prozessketten (z.B. kein Warten auf die Passagiere verspäteter Zubringer), die Verbesserung bestimmter Prozessglieder etwa durch eine Stabilisierung der tatsächlich notwendigen Zeitdauern, die Berücksichtigung der beschränkten Kapazitäten schon bei der Linienplanung oder die Erhöhung der Kapazitäten durch Verbesserung der Infrastruktur.

Besondere Schwierigkeiten ergeben sich daraus, dass es sich bei der Bewertung von Fahrplänen um eine multikriterielle Fragestellung mit teilweise zuwiderlaufenden Zielen handelt: Zusätzliche Wartezeiten aufgrund von Verspätungen (z.B. bedingt durch verpasste Anschlüsse) lassen sich durch eine Erhöhung der Pufferzeiten (z.B. längere geplante Haltezeiten) verringern, dies erhöht jedoch die fahrplanmäßige Reisezeit. Umgekehrt führt eine Verringerung der Puffer von einem gewissen Punkt an dazu, dass das System sehr empfindlich auch auf kleinere Verspätungen reagiert und sich diese sofort im gesamten Netz ausbreiten. Mögliche Gegenmaßnahmen erfordern oft umfangreiche Investitionen, die gegen die Vorteile der Verspätungsverringerung abgewogen werden müssen.

Eine sinnvolle Planung möglicher Gegenmaßnahmen erfordert daher eine Modellierung und Simulation des Verkehrsnetzes unter Einbeziehung der im realen Betrieb auftretenden Verspätungen. Eine solche Modellierung der Verspätungen muss insbesondere die Verzahnung der Verkehrsbewegungen auf den verschiedenen Ebenen angemessen berücksichtigen: Im Bahnverkehr können sich Verspätungen u.a. durch das Warten auf verspätete Zubringerzüge oder durch das Einhalten von Mindestabständen zu anderen Zügen in dem gesamten Netz fortpflanzen. Hinzu kommen die Kapazitätsengpässe an den Bahnhöfen, in Bezug auf die Gleise sowie die eventuell nur beschränkte Verfügbarkeit von Fahrzeugen (Umlauf). Noch komplexer sind die Abhängigkeiten im Flugverkehr, wo zusätzlich auf individuelle Passagiere und umsteigende Crews gewartet werden muss und Maßnahmen der Flugsicherung sowohl in Bezug auf den Flughafen als auch in Bezug auf die Flugrouten ebenso wie die komplexen Bodenprozesse weitere Quellen für Verspätung darstellen.

Delays in public transport: formation and propagation of delays in networks

Anita Schöbel, Michael Kolonko, 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.

Source delays are typically first caused by external disturbances such as extreme weather conditions, technical malfunctions or delays in boarding of passengers. These primary delays may be carried forward through the network to the next trains or aircrafts, so that a single delay can result in a whole cascade of secondary delays in the observed network. Propagation of delays may be caused by waiting for delayed feeder trains or the delayed freeing of a blocked resource (like a runway or a section of a track). Various measures can be used to counteract a propagation of delays: Additional time buffers can be allocated to the schedule, process chains can be decoupled (e.g., no waiting for passengers of delayed feeder trains) and critical links in the process can be improved by stabilizing their duration. It is also possible to prevent delays by taking into account the limited capacity of transport resources at an early stage of the planning, e. g. during line planning or by additional investments into the infrastructure to increase the capacity.

For the systematic construction of delay resistant timetables, additional difficulties arise from the fact that the evaluation of schedules is a multi-objective question with partially contrary objectives: Addi-

tional 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.

A reasonable planning of countermeasures therefore 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). Even more complex dependencies arise in aviation, where an aircraft may have to wait even for individual delayed transit passengers or delayed crews. Local and global air traffic control and the flight paths represent additional sources of delay.

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 are used, for the operational optimization of an appropri-

Szenarien-basierte und stochastische Ansätze

Um die Auswirkungen von Verspätungen sowohl auf langfristiger, infrastruktureller Ebene als auch kurzfristig, operational reduzieren zu können, müssen verschiedene Ansätze verwendet und gegebenenfalls geeignet kombiniert werden. Für die langfristige Optimierung, die im Wesentlichen durch die Veränderung von Linien- und Fahrplänen realisiert wird, werden Methoden der stochastischen Modellierung eingesetzt. Für die operationale Optimierung wird mit Hilfe einer Szenarien-basierten Herangehensweise eine geeignete Disposition für den Fall verspäteter Züge gesucht.

Um diese beiden, auf verschiedene Zielsetzungen hin ausgerichtete Modelle vergleichen zu können, wurden die gemeinsamen Elemente sowohl der stochastischen als auch der Szenarien-basierten Modelle identifiziert und in einem Ereignis-Aktivitäts-Netzwerk (EAN) festgehalten. Die wesentlichen Unterschiede zwischen den Ansätzen bestehen in der Art, wie zufällige Quell-Verspätungen dargestellt werden. In dem Szenarien-basierten Modell geschieht dies durch unterschiedliche Szenarien mit festen Verspätungswerten, in dem stochastischen Modell werden dafür Zufallsvari-

ablen und Wahrscheinlichkeitsverteilungen eingesetzt. Entsprechend unterschiedlicher Art sind die Ergebnisse, die sich für die resultierenden übertragenen Verspätungen ergeben: Szenarien auf der einen, Verteilungen auf der anderen Seite.

Es konnte nachgewiesen werden, dass die beiden Modelle im Wesentlichen zu denselben Resultaten führen. Für diesen Nachweis wurden Verspätungsszenarien für das Szenarien-basierte Modell mit den Verteilungen simuliert, die die Eingangsgrößen für das stochastische Modell bilden. Die resultierenden Verspätungsszenarien wurden dann mit den errechneten Verteilungen des stochastischen Modells verglichen, siehe Abbildung 1. Es ergab sich eine sehr hohe Übereinstimmung, die anhand eines typischen Beispiels in Abbildung 2 dargestellt ist. 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, gewählt werden. Gleichzeitig sind dadurch beide Modelle und Implementierungen validiert worden und die Angemessenheit der Modellierung ist durch das Ereignis-Aktivitäts-Netzwerk belegt worden, siehe auch [10].

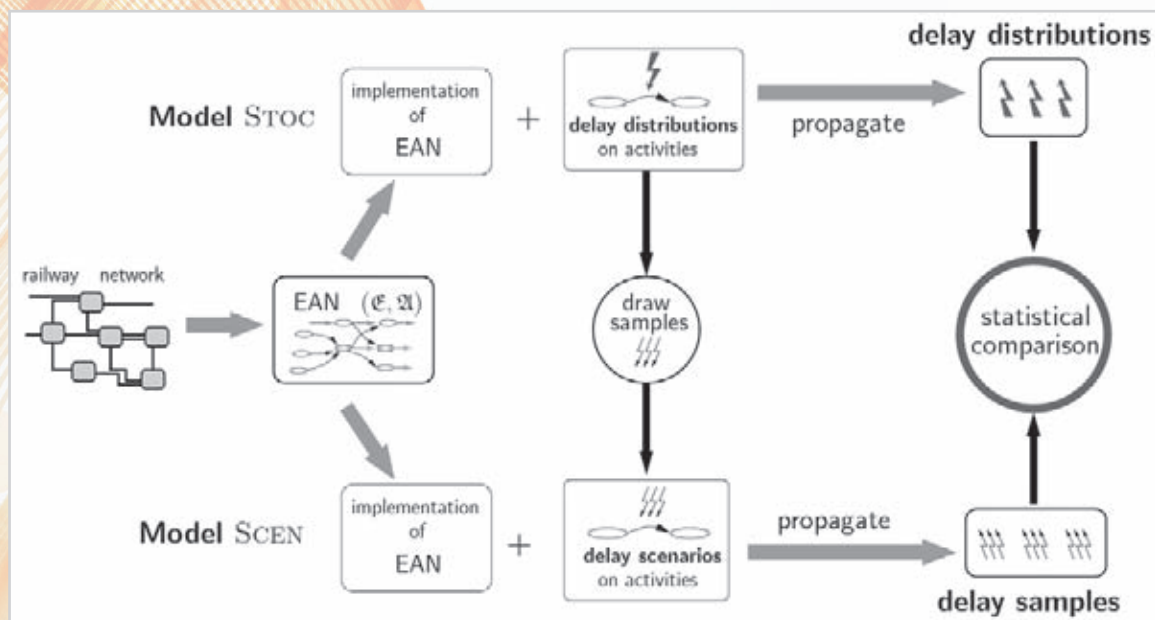


Abbildung 1: Vorgehensweise beim Vergleich des Szenarien-basierten mit dem stochastischen Modell.

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

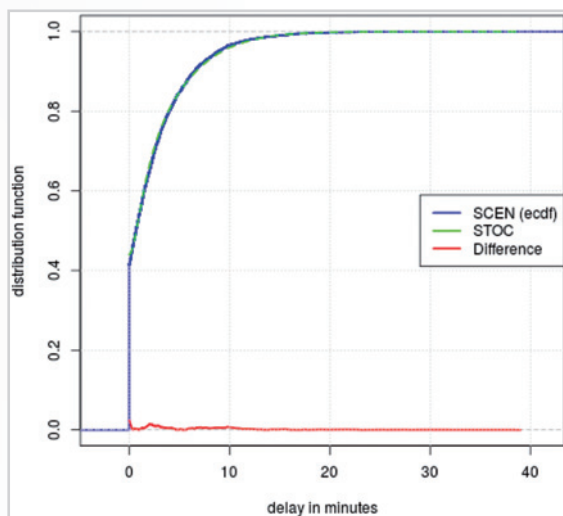


Abbildung 2: Vergleich der errechneten Verspätungsverteilungsfunktion und der empirischen Verteilungsfunktion für übertragene Verspätungen.

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.

ate disposition in the case of delayed trains, a scenario-based approach is appropriate. To be able to compare these two models respecting their different target objectives, first the common elements of both models were identified and recorded in an event-activity network (EAN). The main differences between the two approaches lie in the way they handle random source-delays. In the scenario-based model, this is done by different scenarios with fixed delay values. In the stochastic model, random variables and probability distributions are used for this. Correspondingly, the secondary delays are of different type in the two models: Scenarios on one hand, probability distributions on the other side.

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. So the 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 validity of both models and implementations has been shown by this comparison, see also [10].

Estimation and combination of delay distribution functions

As mentioned above, in the stochastic model source delays are modeled by random variables and distribution functions. For the estimation of the delay carried over from one train to the next, the distribution functions have to be combined in a suitable way. If we assume that all interacting delays are approximately independent, the following operations have to be performed on the delay distribution functions: Folding (for adding delays), multiplication (for maximization if, for example a train is waiting for passengers from two delayed feeder trains) and shift operations. Based on an approach in [4], we use distribution families which are closed under these operations. This means that if two distributions of this family are combined by one of the above operations, we again get a distribution from the same distribution family. At the same time, the selected family must allow a fitting of empirical source delay data. These conditions are fulfilled by the distributions which can be built from theta exponential polynomials and, in a restricted sense, by an extension of the Hyper-Erlang distributions. The fitting and estimation of distributions was examined in [18].

If using only theta exponential polynomials, the terms are getting very complex and will lead to numerical problems after only a few steps. Therefore, an approximation of the resulting distributions by a simpler Hyper-Erlang distribution based on moment matching was proposed. This allows to calculate delay distributions in circle-free nets. In nets with circles there is no topological order and thus no clear order of calculation for determining the resulting delays. In [12] an iterative approximation of long-term delay distributions was used, which reproduces the actual transfer of

Schätzung und Verknüpfung von Verspätungsverteilungen

In dem zur Netzplanung eingesetzten stochastischen Modell werden Quell-Verspätungen mit Zufallsvariablen und Verteilungen modelliert. Zur Berechnung der von einem Zug auf den nächsten übertragenen Verspätungen müssen diese Verteilungen verknüpft werden. Wenn man voraussetzt, dass alle zu verknüpfenden Verspätungen näherungsweise unabhängig sind, werden Faltungen (zur Addition von Verspätungen), Multiplikationen (zur Maximumbildung, wenn z.B. ein Zug auf Passagiere aus zwei Zubringerzügen wartet) und Verschiebeoperationen benötigt. Basierend auf einer Herangehensweise in [4] wurden Verteilungsfamilien verwendet, die unter diesen Verknüpfungen abgeschlossen sind, d.h. dass sich bei der Verknüpfung von zwei Verteilungen aus der jeweiligen Familie wieder eine Verteilung aus der gewählten Verteilungsfamilie ergibt. Gleichzeitig erlauben die gewählten Verteilungsfamilien eine Anpassung an vorgefundene empirische Quellverspätungen. Hierzu sind besonders Erweiterungen der Hyper-Erlangverteilungen und allgemeiner die mit Thetaexponentialpolynomen darstellbaren Verteilungen geeignet. Die Anpassung und Schätzung von Verteilungen wurde in [18] untersucht.

Die Berechnungen allein mit Hilfe der Thetaexponentialpolynomen führen aber nach wenigen Schritten zu sehr komplexen Ausdrücken und numerischen Problemen. Es wurde daher eine auf Momentenanpassung beruhende Approximation der resultierenden Verteilungen mit Hilfen von (erweiterten) Hyper-Erlangverteilungen eingeschaltet. Damit sind Berechnungen von übertragenen Verspätungsverteilungen in kreisfreien Netzen möglich.

In Kreisen gibt es keine topologische Sortierung und damit keine eindeutige Reihenfolge für die Bestimmung der resultierenden Verspätungen. In [12] wurde daher eine iterative Approximation der langfristigen Verspätungsverteilung verwendet, die die tatsächliche Übertragung der Verspätungen nachvollzieht. Mit Hilfe klassischer Resultate der Warteschlangentheorie konnte gezeigt werden, dass die Verspätungen in dieser Approximation nach Verteilung konvergieren,

falls in jedem Kreis die Summe der Zeitpuffer größer als die Summe der mittleren Quellverspätungen ist und die Höhe der übertragenen Verspätungen beschränkt ist. Dies ist gleichzeitig eine praktisch überprüfbare Voraussetzung für den stabilen Betrieb des Netzwerks (vgl. auch [11] und [13]).

Ermittlung verspätungsresistenter Fahrpläne

Zur Bestimmung von Fahrplänen wurden die beiden schon existierenden Systeme (HiTT in Clausthal, LinTim in Göttingen) weiter entwickelt, die im Folgenden beide beschrieben werden.

HiTT wurde um eine neue Komponente zur ereignisorientierten Simulation von Verspätungen erweitert, deren Genauigkeit sich mit Hilfe von Konfidenzintervallen abschätzen lässt. Die ermittelten durchschnittlichen Verspätungen wurden mit den analytisch berechneten Werten verglichen, um so eine Verifizierung der analytischen Berechnung und den dabei vorgenommenen Approximationen zu erhalten. Neben dem Abgleich mit dem Szenarien-basierten Modell konnte so auch ein interner Vergleich mit einer ereignisorientierten Simulation vorgenommen werden.

Im System HiTT werden Fahrpläne mit Hilfe genetischer Algorithmen iterativ verbessert. Es wird dabei ein multikriterieller Ansatz verfolgt, der neben der fahrplanmäßigen Gesamtreisezeit auch die mittlere simulierte Verspätung und die analytisch berechnete mittlere Reiseverspätung miteinbeziehen kann, siehe Abbildung 3. Eine Einbeziehung der mittleren Verspätung in die heuristische Optimierung wird dabei erst durch die schnelle analytische Bestimmung der Verspätungsverteilungen sinnvoll möglich.

Des Weiteren wurden in dem Projekt Ansätze zur Optimierung simulierter Zielfunktionen weiterentwickelt sowie das langfristige Verhalten heuristischer, sogenannter modellbasierter Optimierungsverfahren untersucht (vgl. [16] und [17]). In [8] wird ein Verfahren zur Verringerung der Anzahl an Simulationen, die erforderlich sind, um die besten Lösungen aufgrund einer simulierten Zielfunktion mit einer vorgegebenen Fehlerwahrscheinlichkeit

delays. Using classical results of queueing theory it could be shown that delays converge in distribution for this approximation, if in all circles the sum of the buffer times is greater than the sum of the mean source delays and the amount of the transmitted delays is bounded. This is also a practically testable condition for the stable operation of the network (see also [11] and [13]).

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.

HiTT was expanded to include a new component for event-oriented simulation of delays. The precision of this approach can be estimated by using confidence intervals. The average delays determined by simulation were compared with the

values calculated analytically as described above. This provided a verification of the analytical calculation and the approximations made here. Thus, in addition to the comparison with the scenario-based model from Göttingen, an internal comparison with an event-driven simulation could be achieved.

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 3. 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.

We further developed more general approaches to optimize simulated target functions as well as studied the long-term behavior of heuristics, in

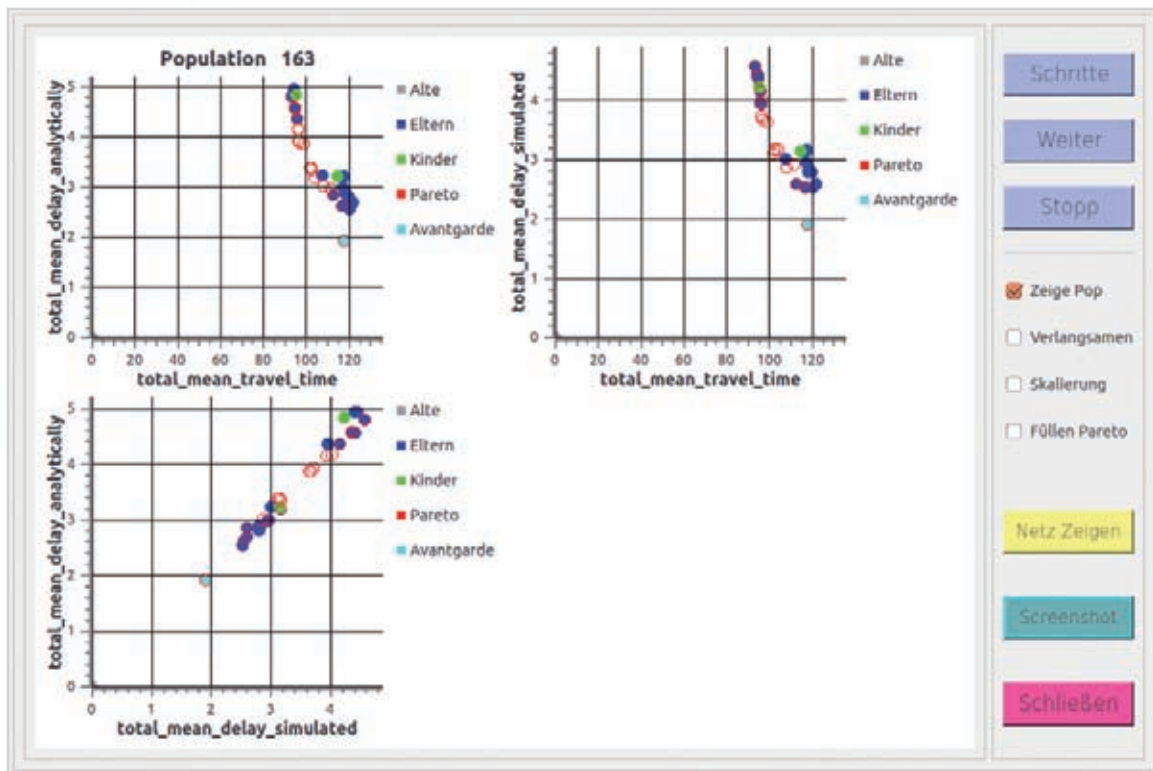


Abbildung 3: Die Entwicklung der Population von Fahrplänen im 3-dimensionalen Kostenraum.

Figure 3: The evolution of the population of timetables in a 3-dimensional costs space.

erkennen zu können, entwickelt. Beide Ansätze haben das Potential, in Zukunft zur Optimierung robuster Fahrpläne eingesetzt werden zu können. LinTim beschäftigt sich nicht nur mit der Konstruktion von Fahrplänen sondern setzt schon bei der Planung des Netzwerkes an und enthält außerdem Routinen zur Umlaufplanung sowie zum Erzeugen von Verspätungen und Optimierungsverfahren für das Verspätungsmanagement. Im Rahmen des Projektes wurde an der konkreten Erzeugung robuster Fahrpläne gearbeitet. Das grundlegende Modell der strengen Robustheit verlangt von einem Fahrplan, dass er beim Auftreten jeder Quell-Verspätung zulässig bleibt. Um dies zu gewährleisten, müssten große Mengen an Pufferzeiten in Form von Fahrzeitzuschlägen eingeplant werden, die den resultierenden Fahrplan wegen der sich daraus ergebenden langen Fahrtzeiten unattraktiv machen würde. Daher wird diese strenge Anforderung in der Praxis meist relaxiert. Zur Verteilung der verfügbaren Pufferzeiten in dem Fahrplan werden dabei zwei Verfahren (siehe auch [2] und [14]) angewandt:

- Um Verspätungen zu begegnen, die früh in der Fahrt einer Linie auftauchen und somit starke Auswirkungen auf den weiteren Fahrtverlauf hätten, ist es sinnvoll, Puffer eher am Anfang einer Fahrt einzuplanen (exponentielle Verteilung).
- Um Verspätungen, die erst im späteren Verlauf der Fahrt einer Linie auftreten, begegnen zu können, ist es sinnvoll, ebenfalls Puffer eher am Ende einer Linienfahrt einzuplanen (reverse-exponential).

Diese beiden Methoden wurden in LinTim implementiert und getestet. Zum Vergleich wurden noch zwei weitere Ansätze implementiert, in denen die Pufferzeiten gleichverteilt auf jeder Kante eingeplant wurden. Dabei wurde zwischen einer absoluten Gleichverteilung (uniform-random) und einer prozentualen Gleichverteilung (proportional) auf jeder Kante unterschieden.

Die Ergebnisse der mit diesen Verteilungen erstellten Fahrpläne wurden bezüglich der planmäßigen (also im störungsfreien Fall realisierten) Fahrtzeit und der Fahrtzeit unter simulierten Störungen verglichen. Als Referenz wurde außerdem ein Fahrplan ohne zusätzlichen Puffer (none) beurteilt.

Abbildung 4 zeigt, dass die absolute Höhe der eingeplanten Pufferzeiten eine viel größere Auswirkung hat als ihre Verteilung.

Generell geht es bei der Recovery-Robustheit um die Entwicklung von Fahrplänen, die zwar nicht jede Verspätung abfangen können, deren Qualität sich aber im Verspätungsfall einfach wiederherstellen lässt (vgl. [7]). Das Problem kann bei Vorliegen einer endlichen Menge an wahrscheinlichen Verspätungsszenarien als Standortproblem im Raum aller Fahrpläne angesehen werden: Für jedes Verspätungsszenario berechnet man einen optimalen Fahrplan und sucht im Raum aller zulässigen Fahrpläne anschließend denjenigen, der möglichst dicht an der Menge der optimalen Fahrpläne für die einzelnen Szenarien liegt. Es lässt sich zeigen, dass ein so gewonnener Fahrplan recovery-robust ist. Derartige Fahrpläne lassen sich mit Hilfe linearer Programmierung im aperiodischen Fall verhältnismäßig schnell auffinden.

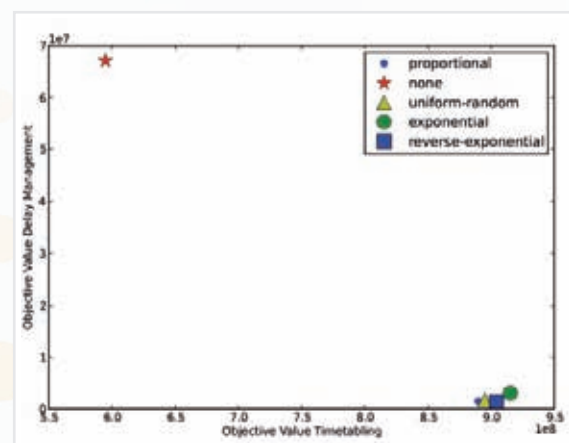


Abbildung 4: Verschiedene Möglichkeiten, Puffer zu verteilen. Aufgetragen ist die simulierte Fahrtzeit im Verspätungsfall gegen die planmäßige Fahrtzeit des jeweiligen Fahrplans.

Figure 4: Different options for buffer distribution. The simulated traveling time is plotted against the scheduled traveling time of each timetable.

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, developed in Göttingen, is not restricted to the construction of timetables, but may also be used in early stages of the planning of networks. It contains routines for vehicle scheduling and optimization methods for the delay management used after a severe disturbance of the operations. Within the project, we worked on creating concrete robust timetables. The basic model of strict robustness requires a timetable that remains valid under each possible source delay. To ensure this, large additional buffer times (extended scheduled driving times) would be needed, which would make the resulting timetable unattractive due to long regular traveling times. Therefore, this strict requirement is usually relaxed in practice. To distribute the available buffer time in the timetable two methods (see [2] and [14]) are applied:

- In order to address delays that occur early in the rotation and thus have a strong impact on the continuing tour, it makes sense to plan buffer times at the beginning of a journey (exponential distribution).
- To address delays that accumulate later in the rotation, it would also make sense to schedule more buffer times at the end of a tour (reverse-exponential).

These two methods were implemented and tested in LinTim. Two other approaches in which the buffer times were distributed equally on each edge have been implemented for comparison. In the implementation there was a distinction between an absolute equal distribution (uniform-random) and a proportional equal distribution (proportional) on each edge.

The results of the timetables created with these distributions were compared to the regular timetables (i.e. to times from a fault-free operation) and the travel times under simulated delays. As a reference, also a road map without additional

buffers (called none) was evaluated. Figure 4 shows that the absolute amount of the scheduled buffer times has a much greater impact than their distribution.

The general idea of recovery robustness is to develop timetables, which may not be able to intercept any delay, but which can easily be restored to normal condition after a breakdown (see [7]).

If there is a finite set of realistic delay scenarios, the problem can be regarded as a location problem in the space of all schedules: For each delay scenario an optimal timetable is calculated in the space of all valid timetables and then the one which is closest to the set of optimal timetables is selected. It can be shown that a timetable obtained this way is recovery-robust. Such timetables can be found relatively quickly by means of linear programming in the aperiodic case.

Investigation on stabilization measures

In case a train has a severe delay, a dispatcher has to decide online how possible conflicts are solved. E.g., he/she has to decide whether trains wait for a delayed feeder train, or which train may run first through a segment of tracks required by several trains. So far, stabilization measures examined in the literature neglect the online nature of the problem: New delays may occur at any time, without being previously known. In [1], this aspect has been investigated and various rules have been tested by means of simulation in LinTim. Using machine learning has been found to be a very promising approach in this case. Furthermore in 2013 ([5]), for the first time the capacity of the stations was included into these decisions and experimentally tested using data from the Dutch railway. In addition, the processes of evaluating the passenger satisfaction was refined (see [6]) and implemented in LinTim.

Integration with other planning steps

In order to be able to use the knowledge gained through the analysis of delays for strategic planning and reasonable operational dispatching, an integrated network including line planning

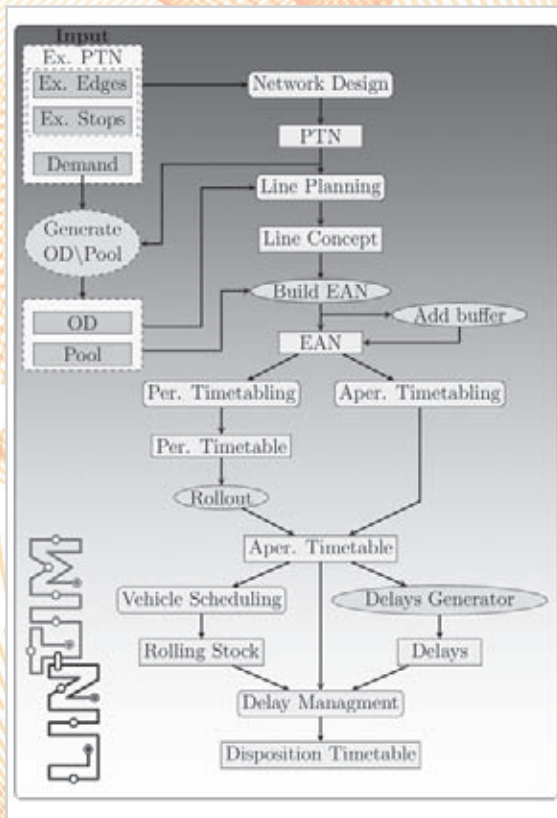


Abbildung 5: Schematischer Zusammenhang der Daten und Planungsschritte im ÖPNV.

Figure 5: Schematic connection between data and planning steps in public transport.

Untersuchung von Stabilisierungsmaßnahmen

Im Fall einer Verspätung im Zugverkehr muss ein Disponent schnell Entscheidungen treffen. Diese beinhalten, ob auf einen verspäteten Zubringer gewartet werden soll, und welcher Zug bei Konflikten zuerst fahren darf. Die in der Literatur bisher untersuchten Stabilisierungsmaßnahmen vernachlässigen dabei den online-Charakter des Problems: Neue Verspätungen können jederzeit auftreten, ohne dass man das vorher weiß. In der Arbeit [1] wurde dieser Aspekt aufgegriffen und verschiedene Regeln mittels einer Simulation in LinTim getestet. Als besonders vielversprechend hat sich ein auf einem maschinellen Lernverfahren beruhender Ansatz gezeigt.

Weiterhin wurden in 2013 ([5]) das erste Mal die Kapazitäten der Bahnhöfe bei diesen Entscheidungen mitberücksichtigt und experimentell anhand von Daten der holländischen Bahn getestet. Außerdem wurden die Verfahren zur Evaluation der Passagierzufriedenheit verfeinert (siehe [6]) und in LinTim implementiert.

Integration mit anderen Planungsschritten

Um die bei der Analyse der Entstehung und Fortpflanzung von Verspätungen in einem Netzwerk gewonnenen Erkenntnisse für die strategische Planung und für operative Reaktionen sinnvoll nutzen zu können, ist eine integrierte Netzwerk-, Linien- und Umlaufplanung notwendig. In Abbildung 5 ist der schematische Einfluss dieser Planungsschritte auf die Fahrplan-Erstellung und die Behandlung von Verspätungen dargestellt, wie er in dem Simulationstool LinTim verwendet wird.

Bei dem Aus- und Umbau von Infrastruktur muss einerseits die Planung von Bahnhöfen unter realistischen Fahrzeitbedingungen (vgl. [3]), aber andererseits auch ein optimierter Streckenausbau berücksichtigt werden, um dadurch mehr Kapazitäten und weniger Verspätungen zu erzielen. Für die Wege der Passagiere und damit auch für das Verspätungsmanagement spielt die Linienplanung eine grundlegende Rolle. So führt eine Linienplanung mit ausgeglichenen Auslastungen der Gleise zu geringeren Verspätungsverbreitungen, da die Zugfolgezeiten zwischen aufeinander folgenden Fahrten mehr Puffer enthalten können. Eine ausgeglichene Auslastung der Gleise lässt sich mit einem spieltheoretischen Ansatz erreichen. Die daraus folgenden positiven Auswirkungen auf die Verspätungen konnten mit Hilfe von Simulationen mit LinTim experimentell bestätigt werden (siehe [15]). Darüber hinaus erscheint es auch sinnvoll, die Linien bereits so zu planen, dass die Passagiere auf den von ihnen gewählten Wegen möglichst wenig umsteigen müssen (vgl. [9]). Es müssen dann auch weniger Umstiege gehalten werden (durch Warten auf die Zubringer), so dass die Ausbreitung von Verspätungen daher unwahrscheinlicher ist. Außerdem kann das sich ergebende Optimierungsproblem zum Auffinden der besten Stabilisierungsmaßnahmen schneller gelöst werden, weil weniger Entscheidungsvariablen (für

and scheduling is required. Figure 5 shows the schematic influence of these planning steps on the scheduling process. Additionally, the treatment of delays used in the simulation tool LinTim is shown.

When expanding and renewing infrastructure, the planning of stations has to take into account realistic driving times (see [3]), but on the other hand also an optimal track planning is required to achieve greater capacity and less delays. For the itineraries of the passengers and for the delay, line planning is fundamental. Thus, a line design with balanced utilization of the tracks leads to less propagation of delays, as the headways between successive trains may contain more buffer times. A balanced utilization of the tracks can be achieved with a game-theoretic approach. The positive impacts on the delays achieved by this method were confirmed experimentally by simulations with LinTim (see [15]). In addition, it appears to be reasonable to plan lines such that passengers face as few train transfers as possible (see [9]). Then less connections have to be kept (by waiting for the feeder), so that there is less delay propagation

in the whole network. In addition, the resulting optimization problem to find the best stabilization measures can be solved faster, because fewer decision variables (one for each transition) have to be considered. Line planning with minimal transfers can efficiently be determined by the method of column generation (see [9]). This approach also allows to solve real-world instances including the actual traveling routes of the passengers. Simulation results from LinTim confirm that the delay robustness of timetables can be increased when respecting the passenger routes during line planning, but sometimes this will result in higher costs.

Figure 6 compares the computation time and the target function values of the IP solutions to the values obtained from the column generation approach of [9]. It should be noted that even for more complex test instances the target function values of the two algorithms show only small differences. The computation time of the IP increases in a way that larger test instances may not be tractable.

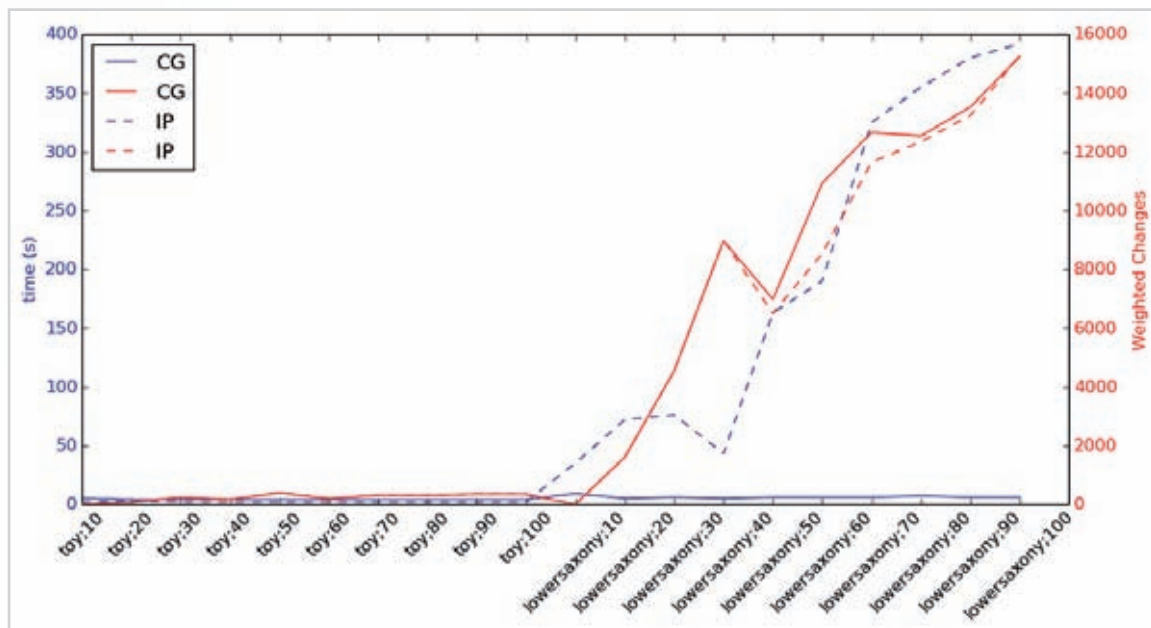


Abbildung 6: Vergleich der Rechenzeit und der Qualität des IPs und des Spaltengenerierungsverfahrens zur Minimierung der Umstiege in der Linienplanung.

Figure 6: Comparison of computation time and quality of the IP and the column generation method for the minimization of the number of transfers in the line planning.

jeden Umstieg eine) zu betrachten sind. Linienpläne mit minimaler Anzahl an Umstiegen lassen sich effizient mit der Methode der Spaltengenerierung (vgl. [9]) bestimmen. Dieser Ansatz erlaubt es auch, die Fahrten der Passagiere inklusive ihrer Umstiege in realitätsnahen Instanzen integriert in der Optimierung mit zu berücksichtigen. Simulative Ergebnisse mit LinTim bestätigen auch hier, dass bei Berücksichtigung der Passagierwege bei der Linienplanung die Verspätungsresistenz von Fahrplänen erhöht werden kann, dies mitunter aber zulasten höherer Kosten.

In Abbildung 6 werden die Rechenzeit und der Zielfunktionswertes des IPs und des Spaltengenerierungsansatzes aus [9] verglichen. Dabei ist festzustellen, dass auch bei zunehmender Komplexität der Testinstanz die Zielfunktionswerte der Algorithmen geringe Unterschiede aufweisen. Die Rechenzeit zur Berechnung des IPs nimmt aber so stark zu, so dass größere Instanzen nicht mehr berechenbar sind.

Projektdate

Das Projekt wird seit April 2013 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

Literatur // 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. 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 für Informatik.
- [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*, 2013. Available online before print, <http://dx.doi.org/10.1287/trsc.2013.0506>.
- [6] T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel. Delay management with rerouting of passengers. *Transportation Science*, 46(1):74-89, 2012.
- [7] 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.
- [8] 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>.
- [9] J. Harbering. A Line Planning Model for Delay Resistance. Technical report, University of Göttingen, 2014.
- [10] 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.
- [11] F. Kirchhoff. Modelling delay propagation in railway networks. In *Operations Research Proceedings 2013*, pages 237 - 242, 2014.
- [12] F. Kirchhoff. Verspätungsfortpflanzung in Bahnnetzen, Modellierung und Berechnung mit abgeschlossenen Familien von Wahrscheinlichkeitsverteilungen. PhD thesis, University of Technology Clausthal, Germany, to be submitted Jan. 2015.
- [13] F. Kirchhoff and M. Kolonko. Modeling delay propagation in railway networks with closed families of distributions. submitted, 2014.
- [14] Leo Kroon, Gabor Maroti, Mathijn Retel Helmrich, Michiel Vromans, and Rommert Dekker. Stochastic improvement of cyclic railway timetables. *Transportation Research Part B: Methodological*, 42(6):553-570, 2008.
- [15] A. Schöbel and S. Schwarze. Finding delay-resistant line concepts using a game-theoretic approach. *Netnomics*, 14(3):95-117, 2013.
- [16] Z. Wu and M. Kolonko. Asymptotic properties of a generalized cross entropy optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 18:1-16, 2014.
- [17] Zijun Wu and Michael Kolonko. Absorption in model-based search algorithms for combinatorial optimization. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 1744-1751. IEEE, 2014.
- [18] Z. Yang. Modellierung und Simulation von Verspätungsdaten mit PH-Verteilungen. Masterarbeit, TU Clausthal, 2012.

Verspätungsfortpflanzung in Verkehrsnetzen – Modellierung und Berechnung mit abgeschlossenen Familien von Wahrscheinlichkeitsverteilungen

Fabian Kirchoff

Die Erstellung von Fahrplänen im Bahnverkehr unterliegt verschiedenen Herausforderungen. Zufällige Einflüsse können Betriebsabläufe stören und letztlich zu Abweichungen vom Fahrplan, d.h. Verspätungen, führen. Dies können beispielsweise nicht schließende bzw. öffnende Türen während des Aufenthalts eines Zuges in einer Station sein. Ein anderes Beispiel sind außergewöhnliche Wetterumstände während der Fahrt eines Zuges. Durch externe Störungen verursachte Verzögerungen werden Quellverspätungen genannt. Sowohl aus Sicht des Dienstleisters als auch aus Sicht der Bahnreisenden ist es wünschenswert, wenn ein Fahrplan das tatsächliche, reale Fahrgeschehen der Züge möglichst genau widerspiegelt. Dies bedeutet, dass er gegen moderate Störungen robust sein muss. In solchen Fahrplänen steht für kritische Betriebsabläufe, zusätzlich zu deren Mindestzeiten, ein Zeitpuffer zur Verfügung.

Die Grundlage der Modellierung in dieser Dissertation stellt ein Ereignis-Aktivitätsnetzwerk dar. Den Aktivitäten werden Quellverspätungen zugeordnet. Darüber hinaus betrachten wir auch fortgepflanzte Verspätungen, die, direkt oder indirekt, durch Quellverspätungen verursacht werden. Diese beziehen sich immer auf ein Ereignis. Verspätungen werden grundsätzlich durch Zufallsvariablen bzw. die zugehörigen Verteilungsfunktionen repräsentiert. Ziel ist es, für alle Ankunfts- und Abfahrtsereignisse, d.h. bezüglich aller Kombinationen von Linien und zugehörigen Stationen, die Verteilungsfunktionen der auftretenden Verspätungen zu bestimmen. Neben den Quellverspätungen fließen hier die im Fahrplan enthaltenen Zeitpuffer sowie Regeln für die Wartezeit von Abbringerzügen auf die jeweiligen Zubringer ein. Mittels der resultierenden Verspätungsverteilungen können Kenngrö-

ßen für die Robustheit eines Fahrplans berechnet werden, welche im Rahmen der Optimierung von Fahrplänen Anwendung finden.

Mit der Hyper-Erlangverteilung wird eine Verteilungsfamilie verwendet, die es ermöglicht, die empirischen Verteilungen der Quellverspätungen hinreichend genau zu approximieren. Dazu wurde ein aus der Literatur bekanntes Verfahren modifiziert, welches für die Anpassung empirischer Verteilungen einen EM-Algorithmus verwendet. Die erhaltenen Verteilungsfunktionen werden als Theta-Exponentialpolynom dargestellt. Diese Polynome sind abgeschlossen gegen die für die Propagation der Verspätungsverteilungsfunktionen benötigten Operationen. Im Zuge der Berechnung der Verteilungsfunktionen werden diese miteinander verknüpft bzw. modifiziert. Die Komplexität der Repräsentation der resultierenden Verteilungsfunktionen steigt dabei im Allgemeinen stark an. Daher ist es nötig, ein Verfahren zur Komplexitätsreduzierung zu nutzen, welches die Verteilungsfunktion hinreichend gut approximiert. Zur Validierung der Ergebnisse wurde im Rahmen dieser Dissertation eine ereignisgesteuerte stochastische Simulation entwickelt. Mittels dieser lassen sich Verspätungen relativ einfach propagieren. Jedoch ist sie im Vergleich zur Verspätungsfortpflanzung auf Ebene der Verteilungsfunktionen sehr zeitaufwendig.

Den Kern dieser Dissertation stellt die Untersuchung der Verspätungsfortpflanzung in starken Zusammenhangskomponenten des Ereignis-Aktivitätsnetzwerkes dar. In diesen Teilen des Netzwerkes ist es nicht möglich, eine topologische Sortierung der Ereignisse herzustellen. Zunächst wurde ein iteratives Verfahren entwickelt, welches es ermög-

Delay Propagation in Transport Networks – Modeling and Calculation using Closed Families of Probability Distributions

Fabian Kirchhoff

The construction of railway timetables is faced with various challenges. Random factors may have an impact on the scheduled procedures and might possibly result in deviations from the timetable, i.e. delays. For instance, this can be a defect door during the halt of a train in some station. Also the train run can be influenced by extraordinary weather conditions. Such external disturbances are called source delays. Both the service provider and the customers require the reliability of a given timetable. That means such a timetable needs to be robust against moderate source delays. Hence, a timetable should contain a certain amount of time buffers for critical procedures.

Modeling basis is an event-activity network. The source delays are assigned to drive and stop activities. Propagated delays are (directly or indirectly) caused by source delays. These delays always refer to an arrival or departure event of the network. Delays are nondeterministic and represented by random variables or the corresponding distribution functions. The aim is to determine the long-run distribution functions of the delays of all events. Besides source delays, this demands knowledge of time buffers and waiting time rules for connecting trains. The resulting distribution functions may be used for computing measurements of the robustness of timetables which are employed in the context of time table optimization.

The family of Hyper-Erlang distributions is used because of its capability of approximating the empirical distribution of observed source delays with sufficient accuracy. For the approximation we slightly modified an already existing approach that makes use of an EM algorithm. The resulting

distribution functions can be easily represented as so called Theta-Exponential polynomials. Those polynomials are closed under the needed operations for propagating delays on the level of distribution functions. In the course of applying these operations, the needed number of parameters to represent the results increases very fast. This causes numerical and performance problems. To reduce the complexity of the representation we make use of a moment matching approach. This implies another approximation of distributions. To validate our approach we designed a discrete-event simulation. In this method, propagated delays can be easily calculated. But, compared with delay propagation on the level of distribution functions, this approach is very time-consuming. The crucial part of this work deals with the delay propagation in strongly connected components of the network. If we want to determine the distribution function of the delay of a specific event, we need to determine the distribution functions of the delays of all of its predecessors before. If the network is acyclic, this is possible. As in this case we are able to find a topological sorting of all events. But in strongly connected components, there does not exist such a topological sorting. Our approach is to iteratively approximate the distribution functions in such components. In our experiments, we used different sortings for the events of the strongly connected components. In each iteration, the distribution functions were determined following the specific sorting. Empirically, the sortings did not have an impact on the convergence or the specific asymptotic distributions of the delays. But we observed an impact on the speed of convergence. Therefore, we propose a pseudo-topological sorting that produces good results concerning this matter.

licht, die Verteilungsfunktionen der Verspätungen aller Ereignisse approximativ zu bestimmen. Es wurden verschiedene Sortierungen für die Ereignisse verwendet. Die Verteilungsfunktionen werden in jedem Iterationsschritt in der Reihenfolge der jeweiligen Sortierung bestimmt. Dabei zeigte sich, dass die spezielle Wahl einer Sortierung empirisch keinen Einfluss auf die Konvergenz der Verteilungen sowie insbesondere auf die Grenzverteilungen hat. Lediglich die Konvergenzgeschwindigkeit variierte in Abhängigkeit der gewählten Sortierung. In dieser Arbeit wird eine pseudo-topologische Sortierung vorgeschlagen, die diesbezüglich im Allgemeinen zu guten Ergebnissen führt. Darauf aufbauend wurden die empirisch erhaltenen Ergebnisse mittels eines theoretischen Modellierungsansatzes untersucht. Es gelang, die empirischen Ergebnisse zu erklären bzw. zu

konkretisieren. Für die Stabilität der Verspätungen in einer starken Zusammenhangskomponente ist das Verhältnis zwischen erwarteten kumulierten Quellverspätungen und kumulierten Zeitpuffern wichtig. Ist der Erwartungswert der aufsummierten Quellverspätungen in jedem Kreis kleiner als die Summe der Zeitpuffer, und ist zusätzlich die Verspätungsübertragung zwischen den Kreisen der starken Zusammenhangskomponente beschränkt, so entwickeln sich die Verspätungen stabil. Ihre Grenzverteilungsfunktion ist unabhängig von der jeweiligen Startverteilung zu Beginn der Iteration, falls die miteinander zu verrechnenden Zufallsgrößen stochastisch unabhängig sind. Diese Ergebnisse sind ein guter Ausgangspunkt für weitere Forschung mit dem Ziel, eventuell weniger restriktive Bedingungen für die Stabilität der Verspätungen zu finden.

Based on this, we investigated the empirical results using a theoretical approach. We succeeded in explaining and specifying the results. Regarding the stability of delays in strongly connected components, the ratio between expected cumulated source delays and cumulated time buffers of the contained cycles is important. If the time buffers are bigger than the source delays and if the propagation of delays among the contained cycles is restricted, the long-run delays will be stable. Their asymptotic distribution is independent of the starting distribution in the first iteration, if the delays to be combined with each other are always stochastically independent. These results are a good starting point for further research, aiming to find more conditions that guarantee the stability of delays.



Sichere Kommunikation in Internet der Dinge Umgebungen

Dieter Hogrefe, Xiaoming Fu, Salke Hartung, Mayutan Arumathurai, Jiachen Chen, Somayeh Taheri

Die Sicherheit der eingesetzten Infrastruktur ist ein wichtiges aktuelles Forschungsgebiet im Bereich der Computernetzwerke. Insbesondere ressourcenbeschränkte Internet der Dinge Umgebungen (Internet of Things, IoT) sind anfällig für Sicherheitslücken, da sich deren Architektur, Hardwarefunktionalitäten und Anwendungen stark von denen normaler Netzwerke unterscheiden.

Notfallszenarien und Naturkatastrophen sind prominente Beispiele für Anwendungsfälle, in denen eine Kommunikationsinfrastruktur, die unnötige Nachrichten (wie z.B. Spam) daran hindert, Ressourcen zu verbrauchen, von entscheidender Bedeutung ist. In der Folge von Katastrophen, wie Erdbeben, Tsunamis oder terroristischen Angriffen, kann die Standard-Kommunikationsinfrastruktur (Festnetz, Mobiltelefone, Internet, usw.) schwer beschädigt und zersplittert sein. In solch einem Szenario ist es umso wichtiger sicherzustellen, dass die verbleibenden Ressourcen genutzt werden können, um wichtige Informationen zu verbreiten (wie z.B. Notfall-Warmmeldungen, Informationen zur Nahrungsmittelversorgung, zu Notunterkünften sowie Sicherheitsbestätigungen), und nicht durch Spam und andere Informationen mit niedriger Priorität belegt werden. Darüber hinaus existiert ein Bedarf an einfach einzusetzenden Signaturerzeugungs- und Verifizierungsalgorithmen.

Da die Anzahl an Geräten, die über eine Internetverbindung verfügen, ständig wächst, müssen hierbei vor allem dezentrale Ansätze als Alternative oder als Ergänzung zu einem Server-basierenden Ansatz in Betracht gezogen werden. Um diese Dezentralisierung zu ermöglichen, werden die folgenden Ansätze betrachtet:

- i) Informationszentrierter Ansatz um virtualisierte Drittanbieter-Dienste für den Endanwender bzw. das Internet der Dinge System zur Ver-

fügung zu stellen. In diesem Fall besteht der Vorteil darin, dass die virtualisierten Systeme näher an dem Punkt betrieben werden können, an dem die jeweiligen Dienste (Firewall, Spam-Filter, Quality-of-Service-Steuerung) benötigt werden, anstatt jeweils die vollständige Server-Infrastruktur durchlaufen zu müssen.

- ii) Sicherstellung des Datenschutzes und der Lastverteilung auf Basis eines verteilten sozialen Netzwerkes. In diesem Fall unterstützen sich die Knoten des Netzwerkes gegenseitig, um eine verlässliche Datenspeicherung zu ermöglichen und gleichzeitig sicherzustellen, dass die Datenschutzvorgaben nicht verletzt werden.

In einer Lokalisierungs-IoT-Umgebung versuchen die einzelnen Komponenten des Netzwerkes ihre eigene Position im Raum zu ermitteln. Ein Einsatzgebiet des Internets der Dinge stellen drahtlose Sensornetzwerke (Wireless Sensor Networks, WSN) dar; hier lassen sich die gesammelten Informationen meist nur dann sinnvoll interpretieren, wenn sie einer räumlichen Position zugeordnet werden können. Da die Ausstattung jedes einzelnen Sensorknotens mit einem GPS-Empfänger eine Reihe von Nachteilen mit sich bringen würde, werden hier effizientere Ortsbestimmungsmethoden benötigt. Darüber hinaus lässt sich durch den Einsatz rechenzeitsparsamerer Ortsbestimmungsmethoden die so gewonnene Zeit für die Ausführung zusätzlicher Sicherheitsmaßnahmen verwenden.

Eine übliche Vorgehensweise, um die Produktions- und Bereitstellungskosten zu reduzieren, um Akkuenergie zu sparen und um die Größe des Systems gering zu halten, besteht darin, dass nur einige wenige Basisknoten ihre Position direkt ermitteln (entweder durch eine statische Positionierung oder mit Hilfe eines GPS-Empfängers) und diese dann die anderen Knoten durch das Versenden von Ortsmitteilungen bei der Ermittlung ihrer jeweiligen

Securing Communications in Internet of Things (IoT) Environments

Dieter Hogrefe, Xiaoming Fu, Salke Hartung, Mayutan Arumathurai, Jiachen Chen, Somayeh Taheri

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

Emergency and natural disasters are prominent examples 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, terrorist 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 complement 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. In this case 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. In this case 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 (WSN) 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 DoS 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.

Position unterstützen. Die genaue Vorgehensweise zur Bestimmung der jeweiligen Position der einzelnen Knoten wird über einen Ortsbestimmungsalgorithmus festgelegt.

Überlegungen zur Sicherheit in drahtlosen Sensornetzwerken waren bislang oft nur auf die Absicherung der Datenübertragung, die Sicherstellung der Datenintegrität und der Abwehr von DoS-Angriffen ausgerichtet. Allerdings bewegt sich der Fokus der aktuellen Forschung dahin, wie die wesentlichen Kernknoten eines Netzwerkes geschützt werden können. Als Kernknoten eines Netzwerkes werden die Knoten bezeichnet, die, wenn sie kompromittiert werden, den Ausfall des gesamten Netzes oder zumindest ein unvorhersehbares Verhalten des Gesamtsystems nach sich ziehen und die daher eine besondere Bedeutung für das gesamte Netzwerk besitzen. Im Kontext eines Lokalisierungs-Netzwerkes sind die Kernknoten genau die Basisknoten, die für die Durchführung des Ortsbestimmungsprozesses unerlässlich sind. Kompromittierte Basisknoten können beispielsweise bewusst falsche Ortsmitteilungen versenden oder überhaupt keine Ortsmitteilungen versenden und so die anderen Knoten daran hindern, ihre eigene Position zu schätzen.

Netzwerksimulation

Während die Implementierung von Algorithmen für IoT-Anwendungen in der Regel einfach ist, sind Feldtests in realen Umgebungen häufig nur schwer durchführbar [18]. Relevante Eigenschaften der Algorithmen wie Skalierbarkeit und Robustheit lassen sich häufig nur mit Hilfe hunderter Knoten testen, was jedoch in realen Szenarien meist unmöglich ist, da kein Zugriff auf ein Netzwerk dieser Größe besteht oder es zumindest nicht möglich ist, den normalen Betrieb für die Tests zu unterbrechen.

Glücklicherweise bietet die Netzwerksimulation eine einfach zu nutzende und kostengünstige Methode, um Netzwerke aus mehreren Tausend Knoten zu modellieren, wobei die Anzahl an abbildbaren Knoten theoretisch nur durch die Hardwareausstattung des Simulationssystems und der darauf laufenden Software begrenzt wird. Darüber hinaus ermöglicht die Simulation,

Spezialfälle zu konstruieren und Experimente mit verschiedenen Parametereinstellungen ohne allzu viel Zeitbedarf zu wiederholen. Mit Hilfe aktueller Simulationssoftware lassen sich alle relevanten Arten von Einflüssen auf ein Netzwerk modellieren: Netzwerk interne Ereignisse (z.B. Verbindungsabbrüche, Paketstaus, Datenverluste), physikalische Effekte (z.B. Witterung, Dämpfungen, Reflexionen, Signalstreuungen) und das Verhalten des Nutzers (z.B. Ortsveränderungen eines Knotens und die Abbildung bestimmter Nutzungsmuster). Darüber hinaus beinhaltet aktuelle Simulationssoftware Implementierungen aller gängigen Protokolle des ISO/OSI-Schichtenmodells.

Ein weiterer wichtiger Vorteil bei der Verwendung von Simulationssoftware besteht darin, dass es damit auf einfache Weise möglich ist, Modelle und Protokolle zu erweitern oder durch benutzerspezifische Implementierungen zu ersetzen, ohne alle darunter liegenden Schichten neu programmieren zu müssen. Damit lässt sich die Implementierung neuer Algorithmen mit wesentlich weniger Aufwand realisieren als in einer realen Testumgebung.

Die für die Netzwerksimulation eingesetzte Software reicht dabei von Spezialsoftware [22] über mathematische Standardwerkzeuge bis hin zu selbstentwickelten Programmen für spezifische Fragestellungen.

Ortsbestimmungsdienste

Eine übliche Vorgehensweise bei der Ortsbestimmung eines Knoten besteht darin, die Abstände zu anderen Knoten zu messen und basierend auf den Messergebnissen die eigene Position zu berechnen. In einer anfänglichen Studie dazu [2] wurde die Nutzbarkeit des Signalstärkeindikators (received signal strength indicator, RSSI) als Schätzer für den Abstand im Kontext entfernungsbasierter aktiver Ortsbestimmungsalgorithmen untersucht. RSSI wird zwar in vielen Ortsbestimmungsalgorithmen verwendet, jedoch meist nur in simulierten Umgebungen getestet. Die Genauigkeit dieser Tests hängt dabei von der Qualität des Systemmodells ab, welches die reale Umgebung repräsentieren soll. In der Studie wurde dabei gezeigt, dass in der Realität RSSI nur als grober Schätzer für Abstände genutzt werden kann und einer Vielzahl

Network simulation

Although implementing algorithms designed for IoT applications is usually easy, field tests in real testbeds are hard to conduct [18]. 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.

For this simulation a wide range of tools beginning from specialized software for network simulation [22] to profound mathematical tools and even custom written tools for very specialized simulation tasks is used.

Localization area

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 [2] evaluated the quality

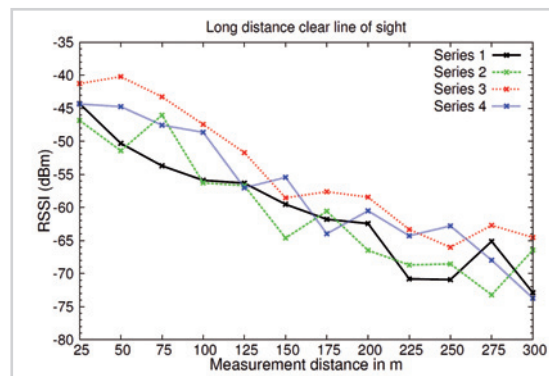


Abbildung 1: RSSI-Level in Abhängigkeit von der Distanz.

Figure 1: RSSI levels in long distance measurement series.

of the received signal strength indicator (RSSI) as a distance estimator for the use in range-based active localization algorithms. 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. The 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 1). Also see [9], [10], [11] and [12].

Efficient localization algorithms for WSNs

The 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.

When using Sensor-Assisted Monte Carlo Localization (SA-MCL) we achieve both by using additional sensors to gather information about the velocity and direction of movement of a sensor node. With the help of these sensor readings a node is able to update its position even in the absence of seed nodes (see [3] and [4] and the related works [13], [15] und [16]).

von äußeren Einflüssen, die die Messgenauigkeit verringern (siehe Abbildung 1), unterliegt (siehe dazu auch [9], [10], [11] und [12]).

Effiziente Ortsbestimmungsalgorithmen für Sensornetzwerke

Das Ziel bei der Verbesserung eines Ortsbestimmungsalgorithmus besteht entweder darin, die Abweichungen zwischen geschätzter und realer Position zu reduzieren oder aber die Anzahl an notwendigen Basisknoten, die in einem Netzwerk für eine stabile Ortsbestimmung notwendig sind, zu reduzieren.

Die sensorunterstützte Monte-Carlo-Ortsbestimmung (Sensor-Assisted Monte Carlo Localization, SA-MCL) ermöglicht die Optimierung auf beide Ziele hin, indem zusätzliche Sensoren an den Knoten eingesetzt werden, um Informationen über die Geschwindigkeit und die Bewegungsrichtung eines Knoten zu erfassen. Mit Hilfe dieser zusätzlichen Sensordaten ist ein Knoten in der Lage, seine Position weiter zu verfolgen, auch wenn kein Kontakt zu den Basisknoten besteht (siehe [3] und [4] sowie damit verbunden [13], [15] und [16]).

Entwurf von SA-MCL

Im klassischen MCL-Ansatz wird die Position eines Knotens durch eine Menge von gewichteten Partikeln repräsentiert. Jeder Partikel stellt eine mögliche Position des Knotens dar; der gewichtete Durchschnitt aller Partikel wird als Schätzwert für die reale Position des Knotens angesehen. Die einzigen Parameter, die allen Knoten in dem Netzwerk bekannt sind, sind die Abmessungen des Gebiets, in dem sie sich befinden, die maximale Kommunikationsdistanz r und die maximale Geschwindigkeit v_{max} eines Knotens. Weiterhin wird angenommen, dass alle Knoten in dem Netzwerk beweglich sind und sich in jede Richtung frei bewegen können. Nur wenigen Knoten in dem Netzwerk ist durchgängig ihre exakte Position bekannt (z.B. weil sie mit einem GPS-Empfänger ausgerüstet sind oder weil sie sich an einer unbeweglichen, festen Position befinden). Diese speziellen Knoten werden Basisknoten genannt und unterstützen alle anderen bei der Bestimmung ihrer jeweiligen Position. Diese

Basisknoten strahlen in regelmäßigen Abständen eine Positionsbenachrichtigung aus, die ihre eigene Position, eine eindeutige ID sowie einen Zeitstempel enthalten. Knoten, die diese Positionsbenachrichtigungen empfangen, zeichnen diese als neue Beobachtungen auf und verwenden diese in dem MCL-Algorithmus.

Der MCL-Algorithmus arbeitet in zwei Stufen:

- Im Vorhersageschritt werden neue Partikel aus einer Menge gezogen, die sich in einem Kreis mit Radius v_{max} um die vorherige Partikelposition befinden. Dieser Schritt wird verwendet, um die zunehmende Unsicherheit über den realen Ort, die auf der Bewegung des Knotens beruht, zu berücksichtigen.
- In dem folgenden Filterschritt werden die aufgezeichneten Beobachtungen verwendet, um die Partikel auszufiltern, die sich nicht in Kommunikationsreichweite eines Basisknotens befinden.

Diese beiden Schritte werden wiederholt, bis N gültige Partikel erzeugt wurden, die die neue Positionsschätzung des Knotens darstellen. Das Hauptproblem des MCL-Algorithmus stellen Situationen dar, in denen keine neuen Positionsbenachrichtigungen aufgezeichnet wurden. In diesen Fällen kann nur der Vorhersageschritt ausgeführt werden, was zu einer wachsenden Unsicherheit in Bezug auf die Position des Knotens führt und damit zu höheren Abweichungen zwischen geschätzter und realer Position.

Bei der SA-MCL-Methode wird diesen Situationen Rechnung getragen, indem zusätzliche Sensorinformationen für die Positionsschätzung herangezogen werden. Diese zusätzlichen Sensoren werden nur aktiviert, wenn die Anzahl der beobachteten Positionsbenachrichtigungen während des letzten Ortsbestimmungsintervalls nahe 0 liegt. Die Sensorinformationen werden verwendet, um eine grobe Schätzung des Weges, den der Knoten relativ zu der letzten geschätzten Position zurückgelegt hat, aufzuzeichnen. Wenn keine neuen Positionsbenachrichtigungen aufgezeichnet wurden, wird der aufgezeichnete Pfad dafür verwendet, um die gesamte Partikelmenge entlang dieses Pfades zu verschieben. Durch dieses Vorgehen kann eine

Design of SA-MCL

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 v_{max} 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 v_{max} 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 the MCL algorithm 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. If 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.

Simulation results

The approach has been implemented as an extension of the original simulation software used in [14]. The simulator is implemented in Java and provides support for the random waypoint models [17] and [18] to account for node mobility. The simulations are done in a 500 m x 500 m square region with 200 network nodes. The radio range is set to 50 m. All experiments are repeated 25 times and the results are averaged. We mainly study three 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 α 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:

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

Verschlechterung der durch die Partikelmenge erzeugten Positionsschätzung verhindert werden. Sobald wieder Positionsbenachrichtigungen empfangen werden, werden die zusätzlichen Sensoren wieder deaktiviert und der Knoten wechselt wieder zu dem effizienteren MCL-Ansatz.

Simulationsergebnisse

Der Ansatz wurde als Erweiterung der ursprünglichen in [14] vorgestellten Simulationssoftware implementiert. Der Simulator wurde in Java implementiert und unterstützt die Zufallswegpunkt-Modelle [17] und [18] (Random waypoint model), um die Knotenbewegungen abbilden zu können. Die Simulationen wurden in einem 500 m x 500 m großen quadratischen Bereich mit 200 Netzwerkknoten durchgeführt. Die Funkreichweite wurde auf 50 m eingestellt. Alle Simulationsexperimente wurden 25 mal wiederholt und als Ergebnisse wurden die Mittelwerte herangezogen. Untersucht wurde der Einfluss der folgenden drei Parameter:

1. Knotengeschwindigkeit:
Für die Untersuchung des Einflusses der Beweglichkeit der Knoten in dem Netzwerk wurde die Geschwindigkeit der einzelnen Knoten zufällig aus dem Intervall $[v_{min}, v_{max}]$ gewählt.
2. Anteil an Basisknoten:
Ziel der Verbesserungen ist es, Situationen mit einem niedrigen Anteil an Basisknoten zu berücksichtigen. Dafür wird der Anteil an Basisknoten in der Nachbarschaft (s_j) eines Knotens kontinuierlich reduziert. Zusätzlich wurde untersucht, wie sich die beiden Algorithmen verhalten, wenn die Anzahl an Basisknoten erhöht wird.
3. Ungenauigkeiten in den Sensordaten:
Zur Berücksichtigung der Ungenauigkeit in den Sensordaten, wird ein Sensorfehler α zu den Daten addiert. Dabei wird untersucht, welchen Effekt die Wahl von α in dem Intervall [0%, 40%] hat. Für alle anderen Versuche wurde α konstant auf 15% gesetzt.

Die zentrale Rolle für einen Ortsbestimmungsalgorithmus spielt der Lokalisierungsfehler, der als

Durchschnitt über alle Knoten angegeben wird:

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

wobei n die Anzahl an Knoten, e_i die Positionsschätzung des i -ten Knoten und l_i die tatsächliche Position des Knoten i sei.

Im Folgenden sollen die wichtigsten der erzielten Ergebnisse vorgestellt werden. Das Hauptziel des SA-MCL-Algorithmus ist es, Situationen mit einem niedrigen Anteil an Basisknoten zu berücksichtigen. Abbildung 2 zeigt, in wie fern die zusätzlichen Sensorinformationen dem SA-MCL-Algorithmus helfen, bessere Ergebnisse als der MCL-Algorithmus zu liefern, insbesondere in Fällen, in denen $s_j < 1$ ist. SA-MCL liefert in diesen Situationen etwa 40% bessere Ergebnisse als MCL. Weiter kann festgehalten werden, dass sich die Ergebnisse von MCL und SA-MCL für einen höheren Anteil an Basisknoten angleichen, da es in diesem Fall nur noch wenige Situationen gibt, in denen keine Positionsmeldungen von Basisknoten vorliegen und so SA-MCL nicht mehr nennenswert von den zusätzlichen Funktionen profitiert.

In Abbildung 3 ist der Positionsbestimmungsfehler für verschiedene Knotengeschwindigkeiten v_{max} dargestellt. Erwartungsgemäß ist der Positionsbestimmungsfehler bei einer höheren Knotengeschwindigkeit bei beiden Algorithmen etwas höher, da sich die Basisknoten bei höheren Geschwindigkeiten schneller aus der Kommunikationsreichweite bewegen. Allerdings muss darauf hingewiesen werden, dass es einen unteren Schwellenwert gibt, bei dessen Unterschreitung die Ergebnisse beider Algorithmen zu einem gemeinsamen Graphen konvergieren, da in diesem Fall nicht mehr genug Variabilität in Bezug auf den Empfang von Positionsmeldungen von verschiedenen Basisknoten besteht. Aus Abbildung 3 kann abgelesen werden, dass die optimale Geschwindigkeit der Knoten (d.h. die Geschwindigkeit, bei der sich die geringsten Positionsbestimmungsfehler ergeben) bei etwa 60% der Kommunikationsreichweite zu liegen scheint.

In Abbildung 4 ist die Konvergenzgeschwindigkeit der beiden Algorithmen dargestellt. Obwohl die Ergebnisse die Aussagen in [14] bestätigen, ist es

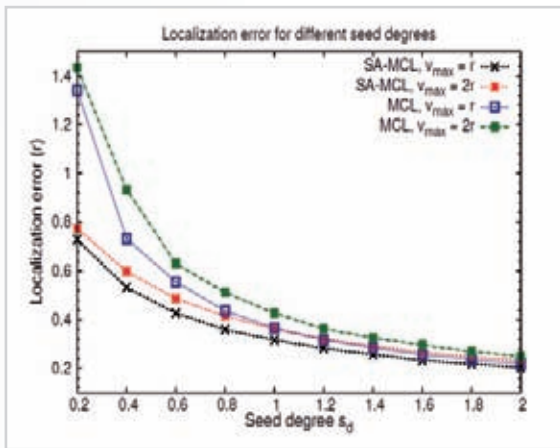


Abbildung 2: Positionsbestimmungsfehler in Abhängigkeit vom Basisknoten-Anteil.

Figure 2: Localization error for different seed degrees.

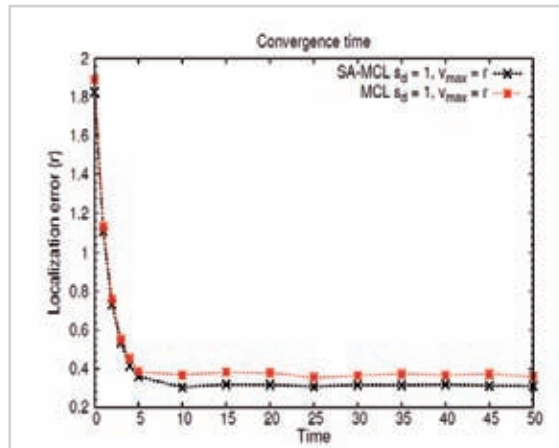


Abbildung 4: Konvergenzgeschwindigkeit

Figure 4: Convergence speed

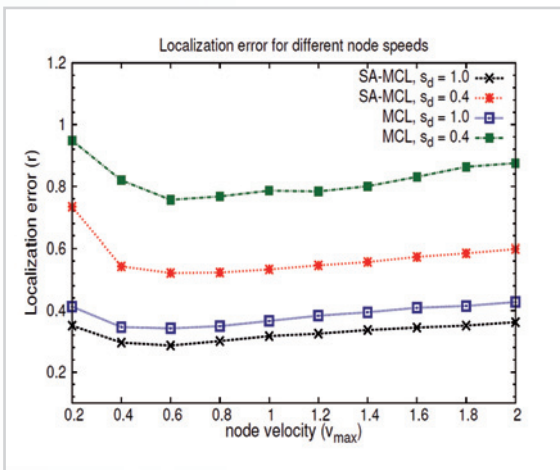


Abbildung 3: Positionsbestimmungsfehler in Abhängigkeit von der Knotengeschwindigkeit.

Figure 3: Localization error for different node velocities.

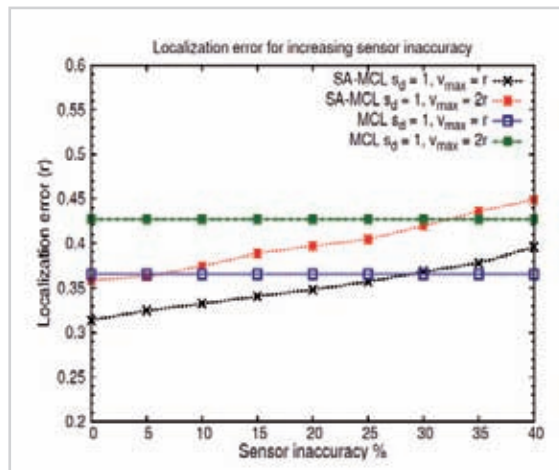


Abbildung 5: Positionsbestimmungsfehler in Abhängigkeit von der Sensorpräzision.

Figure 5: Localization error for different sensor precisions.

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

The most important of the results obtained will be presented next. The key aim of SA-MCL is to account for situations where the seed node degree is low. In figure 2 we show how using the sensor information helps SA-MCL to per-

form 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 noted that for higher seed degrees the position errors of MCL and SA-MCL tend to converge 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.

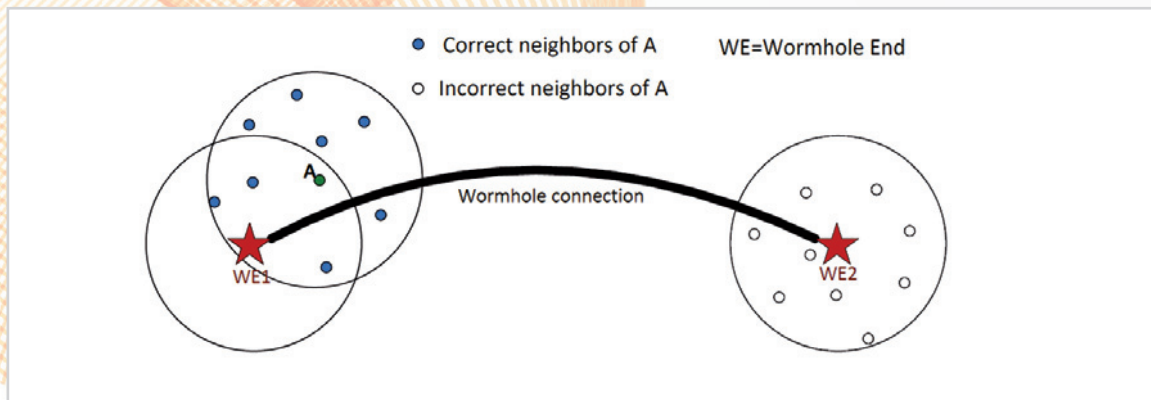


Abbildung 6: Wurmloch-Angriff

Figure 6: Wormhole Attack

ebenfalls ersichtlich, dass sich SA-MCL ähnlich wie MCL verhält und nach etwa 10 Durchläufen einen stabilen Zustand erreicht. Ab diesem Zeitpunkt können fast keine weiteren Verbesserungen der Positionsermittlungsgenauigkeit mehr festgestellt werden.

Lediglich bei der Verwendung von Sensoren mit einer sehr geringen Genauigkeit könnten Probleme bei der Verwendung des SA-MCL-Algorithmus auftreten, so dass in diesen Fällen keine Verbesserung zum MCL-Algorithmus erzielt werden könnte. Um die Auswirkungen von ungenauen Sensoren zu untersuchen, wurden Sensorenfehler im Bereich von 0-40% simuliert. Die Ergebnisse sind in Abbildung 5 dargestellt. Da der MCL-Algorithmus die zusätzlichen Sensorinformationen überhaupt nicht verwendet, hat die Sensorgenauigkeit auch keine Auswirkungen auf die Ergebnisse dieses Algorithmus. Daher sind die MCL-Werte nur als Referenz angegeben. Es kann festgehalten werden, dass SA-MCL selbst bei einer Erhöhung des Sensorfehlers auf bis zu 30% bessere Ergebnisse liefert als MCL.

Sichere Nachbarsuche für drahtlose Ad-Hoc-Netzwerke

Ortsbestimmungsalgorithmen können auf vielfältige Weise angegriffen werden, die in der Regel zu ungenauen Positionsschätzungen führen oder zur Folge haben, dass überhaupt keine Lokalisierung

mehr möglich ist. Da die Basisknoten für fast alle Ortsbestimmungsalgorithmen von essentieller Bedeutung sind, stellen sie ein bedeutsames Ziel für mögliche Angriffe dar. Ein bekanntes Angriffsszenario und Sicherheitsproblem in allen drahtlosen Netzwerken stellen Wurmloch-Angriffe, wie in Abbildung 6 dargestellt, dar. Bei einem Wurmloch-Angriff verknüpft ein Angreifer zwei physisch getrennte Teile des Netzes mit Hilfe einer sehr schnellen Verbindung. Pakete, die in dem rechten Netzwerk ausgestrahlt werden, werden von dem Angreifer über die Verbindung in das linke Netzwerk übertragen und dort erneut ausgestrahlt. In der Folge werden alle Knoten auf der rechten Seite praktisch zu Nachbarn des Knoten A. Angewandt auf ein Ortsbestimmungsnetzwerk bedeutet dieses Angriffsszenario, dass Positionsmeldungen von Basisknoten in einem anderen Teil des Netzwerks erneut ausgestrahlt werden und zu einer fehlerhaften Positionsschätzung seitens der einzelnen Knoten führen können.

Um Wurmloch-Angriffe zu verhindern, ist es von Vorteil, wenn ein Knoten in der Lage ist, zu verifizieren, dass es sich bei den Nachbarn, von denen er Meldungen empfängt, um reale Nachbarn handelt. Dies ermöglicht die robuste sichere Nachbarsuche (Robust Secure Neighbor Discovery, RSND). Im Vergleich zu anderen Lösungen (siehe [19], [20] und [21]) verfügt die RSND über einen verteilten Ansatz zur Identifizierung von Anomalien und zur Erkennung von falschen Nachbarn und ist daher im Gegensatz zu zentralistischen

In figure 3 the localization error is presented for different v_{max} . 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 variability in receiving location announcements from different seed nodes. From figure 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 4 we explore the convergence speed of both algorithms. While our outcomes confirm the results in [14] 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. Only when using sensors with a very low accuracy may problems using the SA-MCL algorithm could occur, so that the MCL algorithm could be better in this case. To account for imprecise hardware we simulate sensor errors from 0-40%. The results are shown in figure 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.

Secure Neighbor Discovery for Wireless Ad Hoc Networks

Localization algorithms can be attacked 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 6. 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 attacker's 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.

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). Compared to other solutions (see [19], [20] and [21]) 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). 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 freer to change because of having a down-weighted role in the error function. After doing such visualization, we will be able to detect the wormhole affected links by checking which links have changed more significantly.

In the simulations nodes are uniformly distributed in a 2D field having radio (and ultrasound) range of $R = 300$ m. The ranging error has a Gaussian distribution with the standard deviation 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 7, with larger ranging error the false-negative rate shows a very slight change

Lösungen nicht anfällig für Single Point of Failure (SPF) Angriffe. RSND verwendet eine auf Basis der gemessenen Abstände erstellte Schätzung der lokalen Topologie, so dass der Effekt einzelner falscher Bereichswerte nur eine geringe Bedeutung hat. Dies ermöglicht es Knoten zu erkennen, von welchen vermeintlichen Nachbarn sie nur durch ein Wurmloch Meldungen empfangen. Die mathematische Grundlage für die Erkennung dieser Anomalien bzw. dieser falschen Nachbarn stellt das Konzept der mehrdimensionalen Skalierung (multi-dimensional scaling, MDS) dar. Bei einer durch MDS vorgegebenen Menge von Punkten besteht die Aufgabe darin, die relativen Positionen der Punkte untereinander zu bestimmen und gleichzeitig die vorgegebenen Distanzen so weit wie möglich einzuhalten. Dieses Ziel wird durch die Minimierung der Summe der Residuen (die Differenzen zwischen den Eingangs- und Ausgangsabstandswerten) erreicht. Dieses Ziel wird im Folgenden dahingehend abgewandelt, dass eine gewichtete Summe der Residuen minimiert werden soll, wobei die Residuen der falschen Verbindungen ein niedrigeres Gewicht erhalten sollen. In Folge dessen würde der Algorithmus die Abstandswerte der korrekten Verbindungen kaum verändern, während die Werte der falschen Verbindungen eher verändert werden, da diese in der Fehlerfunktion nur ein geringes Gewicht besitzen. Auf dieser Basis lässt sich im Nachhinein auch ermitteln, welche Verbindungen durch ein Wurmloch entstanden sind, indem geprüft wird, welche Verbindungsentfernungen wie stark verändert wurden.

In der Simulation wurden die Knoten gleichverteilt auf einem 2-dimensionalen Feld verteilt; die Funk- (und auch Ultraschall-) Reichweite betrug jeweils $R = 300$ m. Der Messfehler der Entfernungen wurde als normalverteilt mit Standardabweichung e angenommen. In jedem Simulationslauf wurde ein Wurmloch mit zwei Enden an einer zufälligen Stelle in das Netzwerk eingefügt. Als erstes wurde untersucht, ob der Algorithmus die Existenz des Wurmloch-Angriffs korrekt erkennt. Als Bewertungsmaßstab wurden die Fehler erster und zweiter Art gezählt. Während der Simulation wurde die Standardabweichung der Messfehler der Entfernungen in einem Bereich von $e = 0$ bis $e = 2,5$ variiert.

In Abbildung 7 ist zu sehen, dass sich die Anzahl an Fehlern zweiter Art (Falsch-negativ-Entschei-

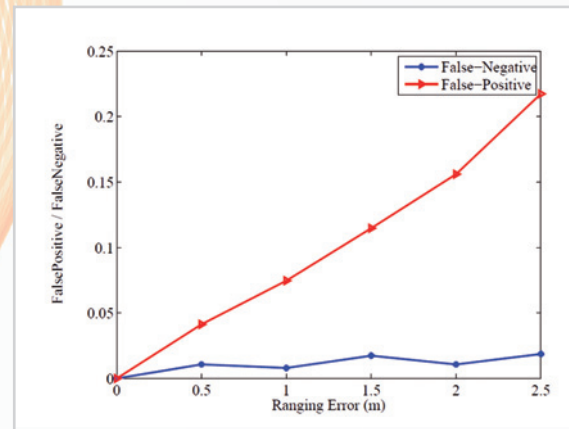


Abbildung 7: Genauigkeit des RSND in Bezug auf die Erkennung der Existenz eines Wurmlochs.

Figure 7: Accuracy of RSND in detecting the presence of wormhole.

dungen) bei einer Erhöhung der Messungenauigkeit kaum verändert, während die Anzahl der Fehler erster Art (Falsch-positiv-Entscheidungen) in einem signifikanten Maß ansteigt. Der Grund hierfür ist die Tatsache, dass große Messfehler bereits Inkonsistenzen in Bezug auf die Entfernungen der Knoten untereinander in einer ansonsten korrekten Menge von Knoten erzeugen können. Im zweiten Teil der Simulation wurde die Leistung von RSND in Bezug auf die Erkennung, ob es sich bei einem vermeintlichen Nachbarn um einen richtigen oder einen falschen Nachbarn handelt, untersucht. Während $N = 12$ fest gewählt wurde, wurde die Anzahl an Knoten, die durch das Wurmloch getunnelt wurden, von $M = 1 \dots 11$ erhöht. Die Simulationen wurden in einer Nachbarschaft aus jeweils 80 Knoten durchgeführt. In Abbildung 8 sind die Ergebnisse der Simulationen in Bezug auf die Genauigkeit der Erkennung von korrekten und falschen Nachbarn dargestellt. Es ist zu sehen, dass sich bei Erhöhung von M für ein fest gewähltes N die Genauigkeit des Algorithmus verschlechtert. Dies liegt daran, da bei festem N und wachsendem M die Anzahl an falschen Abstandsinformationen größer wird, während die Anzahl an korrekten Informationen konstant bleibt.

Mit einem zufällig gewählten M im Bereich $1 \leq M \leq N/2$ und über 150 Nachbarschaften ergeben

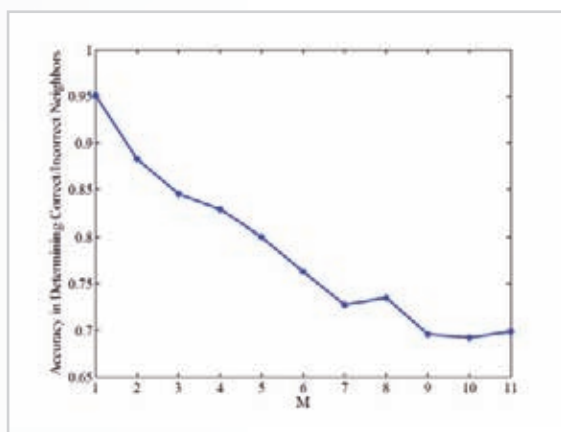


Abbildung 8: Genauigkeit des RSND in Bezug auf die Erkennung von korrekten und falschen Nachbarn eines Knoten für $N = 12$.

Figure 8: Accuracy of RSND in determining correct and incorrect neighbors of a node for $N = 12$.

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 $M = 1 \dots 11$. We run the experiment in the neighborhood of 80 nodes for each M . The result is shown in Figure 8 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 number of the ranging information grows while the consistent number is kept fixed.

With random M from $1 \leq M \leq N/2$ values and running over 150 neighborhoods, we got the results presented in Figure 9. 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

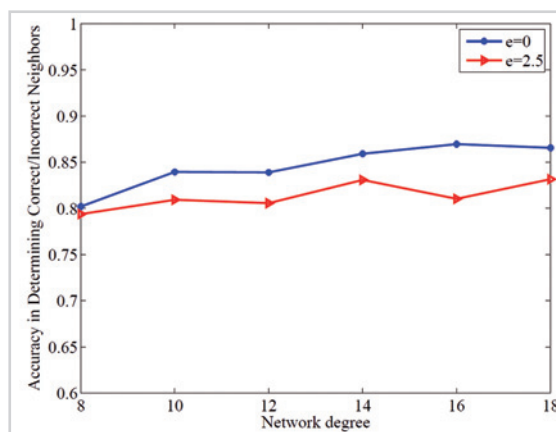


Abbildung 9: Genauigkeit des RSND in Bezug auf die Erkennung von korrekten und falschen Nachbarn eines Knoten für $N = 8$ bis 18.

Figure 9: Accuracy of RSND in determining correct and incorrect neighbors of a node for $N = 8$ to 18.

the verifier node would be neighbors themselves and can measure their distances. Thus, the ratio of unknown edges would be less and with such richer ranging information the topology could be configured more correctly by the algorithm. This indicates the very high scalability of RSND protocol.

Lightweight Integrity Verification 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 an asymmetric 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

sich die in Abbildung 9 dargestellten Werte. Es ist zu sehen, dass die Leistung nicht wesentlich durch den Netzwerk-Grad beeinflusst wird und dass sie sich mit wachsendem N sogar leicht verbessert. Dies liegt daran, dass, obwohl M proportional zu N gewählt ist, sich das Verhältnis von korrekten zu falschen Entfernungsinformationen nicht ändert, da R fest gewählt ist. Für größere Werte von N befinden sich mehr Knoten in der Kommunikationsreichweite eines Knoten (höhere Knotendichte) und damit stehen mehr Knoten zur Verifikation der Position zur Verfügung. Damit sinkt der Anteil an unbekanntem Verbindungskanten und mit diesen verbesserten Entfernungsinformationen kann die Topologie des Netzes besser durch den Algorithmus bestimmt werden. Dies zeigt auch die gute Skalierbarkeit des RSND Protokolls.

Einfache Integritätsprüfungsarchitektur für ICN Benannte Datennetze (Named data networks, NDN) stellen ein neues Modell für die Zukunft des Internets dar, bei dem Datenpakete inhaltsbezogene Namen tragen statt wie bisher Absender- und Zieladresse. Die Sicherheit wird bei NDNs durch das Einbetten von mit asymmetrischer Verschlüsselung erzeugten Signaturen in jedes Datenpaket gewährleistet; auf diese Weise können Authentizität und Integrität der Inhalte sichergestellt werden. Allerdings verhindert der hohe Ressourcenbedarf der Algorithmen zur Generierung und Prüfung der Signaturen einen universellen Einsatz in allen Knoten eines NDN, so dass Paketüberflutungen und Denial-of-Service-Angriffe nicht ausgeschlossen werden können. Darüber hinaus verhindern Systeme zur Zwischenspeicherung und ortsunabhängige Zugriffe, dass der Inhaltsanbieter den Zugriff auf die Daten kontrollieren kann, zum Beispiel um festzulegen, wer die Inhalte zwischenspeichern darf oder welcher Nutzer oder welches Gerät auf die Inhalte zugreifen darf. Um diese beiden Probleme zu behandeln, wird eine lichtgewichtige Integritätsüberprüfungsarchitektur (Lightweight integrity verification architecture, LIVE) als Erweiterung des NDN-Protokolls vorgeschlagen. LIVE ermöglicht eine universelle Signierung und Überprüfung der Inhalte in einem NDN mit ressourcenschonenden Signaturgenerierungs- und Verifikationsalgorithmen. Ferner ermöglicht es Inhaltsanbietern, den Zugriff auf Inhalte in dem NDN durch die selektive Verteilung von Integritätsprüf-Tokens an berechnete Knoten

zu kontrollieren. Die Effektivität von LIVE wurde mit dem Open-Source-System CCNx bewertet. Die Auswertung (siehe [7]) ergab, dass LIVE nur zu einer Verzögerung von durchschnittlich 10% beim Zugriff auf die Inhalte führte. Verglichen mit herkömmlichen Public-Key-Signaturverfahren verringert sich die Verzögerung durch die Prüfung der Inhalte um mehr als das 20-fache.

Verteilte und dezentrale Ansätze

Dezentralisierte Bereitstellung von virtualisierten Diensten

Netzwerke werden immer komplexer und Serviceanbieter bieten immer mehr Zusatzfunktionen (wie z.B. Firewalls, Spamfilter, QoS-Management) um die Dienste zu schützen und zu verwalten sowie um die gebotene Leistung zu verbessern. Mit Hilfe von Software-Defined-Networking (SDN) wird versucht, das Netzwerk auf Basis einer (logischen) zentralen Steuerungsebene zu verwalten. Derzeitige SDN-Lösungen übersetzen Vorgaben (was) in Weiterleitungsregeln zu bestimmten Switches (wo). Diese Wahl schränkt die Dynamik, Flexibilität und Zuverlässigkeit, die ein Software-Defined-Network bieten könnte, ein. Informationszentrierte Netzwerke (Information centric networking, ICN) verschieben hier den Fokus, in dem Netzwerke nicht mehr nur als standortbezogene Kommunikationsumgebungen betrachtet werden. Damit verbessert sich die Flexibilität im Netzwerkmanagement deutlich. Im Folgenden sollen insbesondere die Probleme, die im Netzwerkmanagement bei der Steuerung der Datenflüsse durch die verschiedenen Netzwerkfunktionen auftreten, bevor die Pakete die jeweiligen Ziele erreichen, betrachtet werden. Hierfür wird die funktionszentrierte Dienstverkettung (Function-Centric Service Chaining, FCSC) vorgeschlagen. Dabei handelt es sich um eine Lösung, die das ICN nutzt, um die Flexibilität bei der Verwaltung von Netzwerken zu steigern, indem mit Hilfe von Virtualisierung dynamisch Funktionen nach Bedarf in dem Netzwerk zur Verfügung gestellt werden. Um die Leistungen des FCSC mit dem einer traditionellen SDN-Lösung zu vergleichen, wurden in der Simulationsumgebung sowohl eine einfach als auch eine die Realität abbildende Topologie verwendet. Es konnte nachgewiesen werden, dass FCSC auf Ausfälle mit

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 system CCNx. Our study (see [7]) 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.

Distributed/Decentralized Approaches

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. Current SDN solutions pre-translate policies (what) into forwarding rules at specific switches (where). This choice is limiting the dynamic, 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. We will 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 uses 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 and adapts to new middleboxes more quickly (see [6]).

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 (see [1]).

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

weniger Paketverlusten reagiert und sich gleichzeitig schneller an neue Netzwerkkomponenten anpasst (siehe [6]).

Datenschutz auf Basis dezentralisierter sozialer Netzwerke

Gleichzeitig mit dem enormen Wachstum der sozialen Netzwerke (Online-Social-Networking, OSN) nahem die Sorgen der Nutzer in Bezug auf den Schutz ihrer persönlichen Daten und damit verbunden ihrer Privatsphäre zu. Da die Verwaltung der Nutzerdaten üblicherweise zentralisiert abläuft, ist es den OSN-Anbietern heute uneingeschränkt möglich, auf diese Daten zuzugreifen, was ebenfalls großflächige Datenabgriffe durch Sicherheitslücken an einer einzigen Stelle ermöglicht. Eine Möglichkeit zur Lösung dieses Problems besteht darin, die Nutzerdaten dezentral zu speichern und die Daten auf die jeweiligen Endnutzer-Geräte über das OSN zu verteilen. Allerdings bringt ein derartiger Ansatz neue Herausforderungen mit sich. Insbesondere muss die Verfügbarkeit der Daten bei allen Nutzern mit minimalem Duplizierungsaufwand, ohne die Daten dabei dadurch dauerhaft online verfügbar zu machen, sichergestellt werden. Gleichzeitig müssen Mechanismen für die Verschlüsselung von Benutzerdaten, die Steuerung des Zugriffs auf die Daten und die Synchronisation der Kopien bereitgestellt werden. Darüber hinaus muss das System mit großen sozialen Netzwerken skalieren und robust und anpassungsfähig sowohl in Bezug auf die Nutzung durch reguläre Anwender als auch in Bezug auf Angriffe durch böswillige Benutzer sein. Während die Arbeiten der letzten Zeit auf diesem Gebiet nur begrenzte Fortschritte brachten, wird hier ein dezentrales OSN, welches selbstorganisiertes Universum der Menschen (Self-Organized Universe of People, SOUP) genannt wird, vorgestellt. SOUP verwendet ein skalierbares, robustes und sicheres System zur Spiegelung der Daten und kann die Nutzerdaten effektiv verteilen und deren Verschlüsselung über das gesamte OSN organisieren. Eine extensive simulationsbasierte Auswertung und der Einsatz in der Realität zeigen, dass SOUP all diese Vorgaben erfüllt (siehe [1]).

Geo-verteilte Cloud-Systeme

Geografisch verteilte Cloud-Systeme bieten eine interessante Plattform für die Bereitstellung von OSN-Dienstleistungen. Um das Potenzial der Cloud-Systeme zu nutzen, ist es für OSN-Anbieter wichtig, die Kosten für die Nutzung der Cloud-Ressourcen unter Berücksichtigung wichtiger Nebenbedingungen, insbesondere der Service-Qualität (Quality of service, QoS) und der Verfügbarkeit für die OSN-Nutzer, zu minimieren. In diesem Beitrag werden Fragen der Kostenoptimierung für dynamische OSN auf mehreren geo-verteilten Cloud-Systemen über aufeinander folgende Zeiträume unter Einhaltung vordefinierter QoS-Standards und Datenverfügbarkeitsanforderungen untersucht. Dabei werden die Kosten, der QoS sowie die Datenverfügbarkeit in dem OSM modelliert und es wird ein für diese Problemstellung entworfener Algorithmus „Cosplay“ vorgestellt. Zur simulationsbasierten Untersuchung des Algorithmus wurden Twitter-Daten über 10 geo-verteilte Cloud-Systeme in den USA verwendet. Die Ergebnisse zeigen, dass der neue Ansatz in einem OSN mit einer Dynamik, vergleichbar zu realen Szenarien, bei Einhaltung des QoS und der Datenverfügbarkeit die einmaligen Kosten gegenüber den üblichen Methoden reduzieren kann und außerdem die kumulativen Kosten (bei Betrachtung eines Zeitraums von 48 Monaten) signifikant reduzieren kann (siehe [5]).

Empfehlungsbasierte Abwägung zwischen Netzwerkpfeifen

In dieser Arbeit werden Methoden zur Bewertung von Netzwerkpfeifen, d.h. die Gewinnung von quantitativen Werten bzgl. der Eigenschaften von einzelnen Pfeifen wie Laufzeiten und die verfügbare Bandbreite, untersucht. Im Gegensatz zu feingranularen Messungen besitzen die gröberen Bewertungsmethoden den Vorteil, dass sie einfach anzuwenden sind. Daher soll im Folgenden eine skalierbare Möglichkeit zur Gewinnung von Pfeifen-Bewertungen auf Basis von statistischer Inferenz beschrieben werden. Durch die Analyse von Gemeinsamkeiten mit anderen Empfehlungssystemen wird die Anwendbarkeit der Lösung für

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 (see [5]).

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 (see [8]).

Literatur // References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.

Empfehlungssysteme untersucht und gezeigt, dass das Inferenz-Problem durch eine Klasse von Matrix-Faktorisierungsmethoden gelöst werden kann. Ein technischer Beitrag dabei ist das aktive und progressive Inferenz-Framework, das nicht nur die Genauigkeit durch die selektive Messung von mehr Informationen steigert, sondern außerdem die Konvergenzgeschwindigkeit in Bezug auf die Bandbreite durch die Einbeziehung der Messmethodik. Des Weiteren wurde die Nutzbarkeit der bewertungs-basierten Netzwerk-Messung und -Inferenz in Anwendungen untersucht. In einer

Fallstudie wurde untersucht, ob Kenntnisse über die lokalen Umgebung bei Overlay-Netzwerken von der Art Pastry und BitTorrent durch die Nutzung von Inferenz-Bewertungen gewonnen werden können. Mit Hilfe von simulationsbasierten Bewertungs- und Analyseansätzen lässt sich zeigen, dass bereits mit Hilfe von grobgranularem Wissen die Qualität bei der Auswahl der Kontaktpunkte verbessert werden kann, während feinere Methoden nicht notwendig zu weiteren Verbesserungen führen (siehe [8]).

Projektdaten

Das Projekt wird seit April 2013 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 Arumathurai
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

- [9] 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.
- [10] 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.
- [11] 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.
- [12] Y. Chen and A. Terzis, "Calibrating RSSI Measurements for 802.15.4 Radios," in Proceedings of the Seventh European Conference on Wireless Sensor Networks (EWSN), 2010, pp. 256–271.
- [13] G. Teng, K. Zheng, and W. Dong, "MA-MCL: Mobile-Assisted Monte Carlo Localization for Wireless Sensor Networks." *IJDSN*, vol. 2011, 2011.
- [14] L. Hu and D. Evans, "Localization for Wireless Sensor Networks," in 10th Annual International Conference on Mobile Computing and Networking (MobiCom 2004), Philadelphia, USA, 2004, pp. 45–57.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] QualNet by SCALABLE Network Technologies Inc. (2014, Nov.) SCALABLE Network Technologies website. [Online]. Available: <http://www.web.scalablenetworks.com/>

Diffusionsapproximation und Dekomposition von Warteschlangennetzen mit Batch-Processing

Wiebke Klünder, Thomas Hanschke, Horst Zisgen

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 umzusetzen sind. Um das Eingreifen in die laufende Produktion zu vermeiden, ist die Existenz und der Einsatz von zuverlässigen Prognose- und Modellrechnungen essentiell. Werden die erstellten Modelle in die Geschäftsprozesse integriert, so ist es möglich, Fragen zur taktischen und operativen Produktionsplanung zeitnah zu beantworten und entsprechend auf Veränderungen zu reagieren.

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

ten bzw. gebundenem Material führt. Im Endeffekt verteuern sich die Endprodukte und das Unternehmen büßt Wettbewerbsvorteile ein.

Abbildung 1 zeigt ein typisches Durchlaufzeit-Profil eines Fertigungsprozesses in einer Technologie-Linie, beispielsweise in einer Halbleiterfabrik. Es ist klar ersichtlich, dass die Wartezeit eines Bauteils den größten Anteil an der Durchlaufzeit aufweist und der Bedarf einer Optimierung im Sinne einer Minimierung der Wartezeit den größten Effekt auf die gesamte Durchlaufzeit und der zeitlichen Verkürzung des Produktionsprozesses haben wird. Auch in Herstellungslinien anderer Industriezweige tritt das Problem des hohen Anteils an Wartezeit auf, so dass auch dort Optimierungsbedarf besteht.

Allgemein gibt es zwei Ansätze Produktionsprozesse zu optimieren und Warteschlangen entlang der Wertschöpfungskette zu minimieren. Beiden Ansätzen ist gemein, dass sogenannte Leistungsgrößen, wie z. B. die langfristige mittlere Anzahl an Aufträgen in einem System, ermittelt werden, die Auskunft über Schwachstellen und Engpässe in der Produktion geben, was für Unternehmen von großer Bedeutung ist.

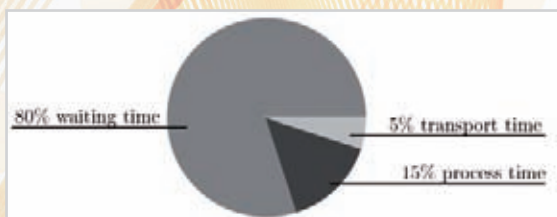


Abbildung 1: Typisches Durchlaufzeit-Profil in Technologie-Linien.

Figure 1: Typical profile of throughput time in a technology line.

Die Simulation wird als eine Methode zur quantitativen Bestimmung der Leistungsgrößen von Produktionssystemen verwendet. Der zu untersuchende Prozess wird in Form eines Computerprogramms mehrfach durchlaufen und der Prozessverlauf protokolliert. Der resultierende Datensatz wird anschließend mit geeigneten Werkzeugen der Statistik auf die gesuchten Kenngrößen hin untersucht. D. h., dass die Leistungsgrößen a posteriori bestimmt werden. Die Ermittlung ist also erst aufgrund gewonnener Erfahrung möglich. Diese Methode, die allgemein unter dem Namen

Diffusion approximation and decomposition of queueing networks with batch processing

Wiebke Klünder, Thomas Hanschke, Horst Zisgen

The importance of simulation methods for production planning and control in industrial manufacturing has steadily increased in recent years. The increase is caused by the complexity of production systems and reduction of production time. This puts the manufacturing companies in need to optimise production processes from the beginning, since subsequent corrections in the manufacturing processes are technically and logistically difficult to manage. To avoid manual corrective actions in the production process, the existence and the use of reliable models for calculation and prediction are essential. Integrating the created models into business processes allows to find answers to questions of tactical and operational production planning to quickly react to changes.

The forming of queues is an effect that should be avoided or minimised from an economic point of view. A queue in an assembly line means a prolongation of total production time by waiting times, which raises storage costs and bound material. In the end the costs of the final product increase and the company loses competitive advantages.

Figure 1 shows a typical profile of processing time in a technology line, e.g. in semiconductor industries. It is clear that the waiting times of modules own the largest part of the total processing time. Minimising the waiting time will have the greatest effect on the throughput time. The problem of long waiting times also occurs at assembly lines, so in other industries there is the need for minimisation too.

Generally there are two approaches to optimise manufacturing processes and minimise queues along supply chains. Both approaches share calculations of parameters of queueing systems

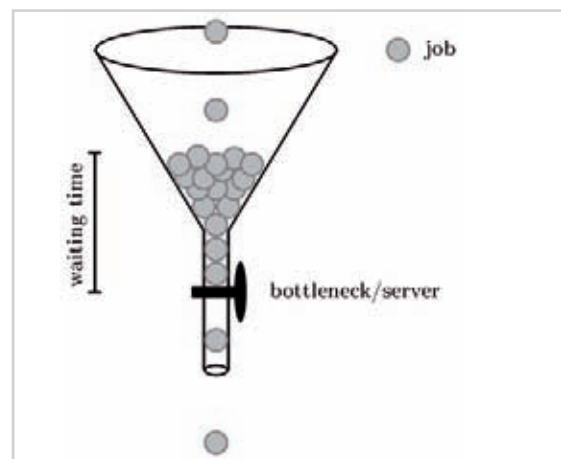


Abbildung 2: Engpass in Produktionsschritt.

Figure 2: Bottleneck in production step.

performance, e.g. the average number of customers in the queueing system. These parameters provide information about bottlenecks and other weaknesses in production systems. This kind of information is of great importance to companies.

The simulation is used as one method to calculate parameters for production systems. The process to be examined will be repeated several times in form of a computer program and the development of the process will be recorded. The resulting data will be analysed using appropriate statistic tools to receive the parameters. I. e. the parameters are determined a posteriori. The gained experience is the basis for the determination of the parameters. This method, commonly known as the Monte Carlo method, has the disadvantage that numerous experiments have to be done to receive parameters by approximate computations. Func-

Monte-Carlo-Simulation bekannt ist, hat den Nachteil, dass zahlreiche Experimente durchgeführt werden müssen, um eine Leistungsgröße durch Iteration näherungsweise zu bestimmen. Funktionale Zusammenhänge zwischen verschiedenen Parametern und den Leistungsgrößen können durch die Monte-Carlo-Simulation nicht ermittelt werden. Ein weiteres Problem stellt der hohe Bedarf an Rechenleistung dar. Dies widerspricht dem Ziel, ein Werkzeug für die taktische und operative Produktionsplanung bereit zu stellen, um möglichst in kurzer Zeit eine Grundlage für insbesondere operative Entscheidungen zu erhalten.

Neben dem Ansatz der Simulation hat sich die analytisch orientierte Methode der Warteschlangentheorie als Alternative zur Monte-Carlo-Simulation etabliert. Der Vorteil der Warteschlangentheorie als Produktionsplanungswerkzeug liegt darin, dass die jeweilige Fragestellung mathematisch modelliert wird, z. B. wieviel Warteraumkapazität bzw. Puffer veranschlagt werden sollte und wie sich eine Erhöhung der Bedienrate durch eine Maschine auf die maximale Warteschlangenlänge auswirkt oder ob eine Erhöhung der Produktion durch zwei Maschinen positivere Effekte auf die Minimierung der Warteschlangenlänge hätte. Durch eine Modellierung mit Hilfe von Warteschlangenmodellen werden die Leistungsgrößen mit einer entsprechenden Genauigkeit a priori bestimmt, ohne dass ein hoher Rechenaufwand entsteht. Neben diesem zeitlichen Vorteil liefern Warteschlangenmodelle einen funktionalen Zusammenhang zwischen Systemparametern und Leistungsgrößen. Das bedeutet, dass sich weitere Leistungsgrößen, welche für weitere Fragestellungen relevant sein können, für das zugrundeliegende Modell ohne zusätzlichen Aufwand bestimmen lassen.

Der Optimierungsansatz mit Hilfe von Warteschlangenmodellen hat jedoch auch Nachteile. Die wesentliche Einschränkung der Verwendungsmöglichkeiten von analytischen Warteschlangenmodellen zur Optimierung von Produktionsprozessen besteht darin, dass sich nur einfache Modelle analytisch exakt lösen lassen, während für allgemeine und komplexere Modelle keine geschlossenen Lösungen bekannt sind. In dem vorliegenden Projekt werden daher bestehende Näherungsverfahren, die auf stochastischen Grenzwertsätzen, Diffusionsapproximation und Dekomposition beruhen, zur Bestimmung der Leistungsgrößen von allgemeinen Warteschlangenmodellen weiterentwickelt.

Die Problemstellungen werden also durch approximative Verfahren behandelt. Es folgt dann eine Simulation, welche die approximativen Ergebnisse validiert (Abbildung 3).

Allgemeine Warteschlangensysteme

In der Warteschlangentheorie wird von einem Grundmodell ausgegangen (Abbildung 4), welches die relevanten Komponenten einer Bedienstation bzw. eines Wartesystems darstellt.

Zusätzlich kann das Modell erweitert werden, um weitere identisch und parallel arbeitende Bediener. Es wird dann von einem Mehrbedienersystem gesprochen. Eine andere Erweiterung sind die Gruppenankünfte und -bedienung, d. h. Kunden treten nicht einzeln in das System ein, sondern in Gruppen unterschiedlicher Größe. Die Gruppenbedienung erfolgt in sogenannten Stapeln (Batches). Ein Stapel muss für den oder die Bediener eine minimale und maximale Größe aufweisen. Die Erzeugung eines Stapels wird Batch-Processing genannt.



Abbildung 3: Sequentielle Verknüpfung von Optimierung und Simulation

Figure 3: Sequential combination of optimising und simulation

tional relationships between various parameters of a queueing system and its performance parameters cannot be determined by the Monte Carlo method. Another problem is the high demand of computing power to receive the performance parameters. This contradicts the goal of providing a tool for tactical and operational planning to create a basis of decision-making, particularly for short-term operational decisions.

In addition to the approach of the simulation the analytically oriented method of queueing theory has been established as an alternative to the Monte Carlo method. The advantage of queueing theory as a production planning tool is that the actual problem is mathematically modeled, e. g. how much waiting room capacity respectively buffer should be provided and how the increase of the service speed by a machine affects the maximum queue length or whether an increase of the production by adding a second machine would have positive effects on minimising the queue length. Using queueing models the performance parameters of production systems will be determined a priori with corresponding accuracy and without requiring high demand of computing power. Furthermore to this speed improvement queueing models provide a functional relationship between system parameters and performance parameters. This means other performance parameters, which may be relevant for production planning and control, are determined without additional effort for the underlying model.

The optimisation approach using queueing models also has disadvantages. The major limitation of the use of analytical queueing models for the optimisation of manufacturing processes is, that only simple models have an analytical and exact solution, while complex queueing models cannot be treated analytically. In this project the existing approximation methods, based on the central limit theorem, will be extended to determine performance parameters of general queueing systems.

Thus the performance parameters will be treated by approximate methods. Then a simulation follows that validates the approximation results (figure 3).

General waiting systems

In queueing theory it is assumed, that a basic model (figure 4) represents all relevant components of an operating station or a queueing system.

In addition, the model can be extended to other identical or parallel servers. This is known as a multi-user system. Other extensions are batch arrivals and batch service, i. e. jobs or customers do not occur individually in the system, but in groups of varying sizes. Groups of varying sizes arrive at servers in so-called batches. A batch must have a minimum and maximum size for the servers. The creation of a batch is called batch processing.

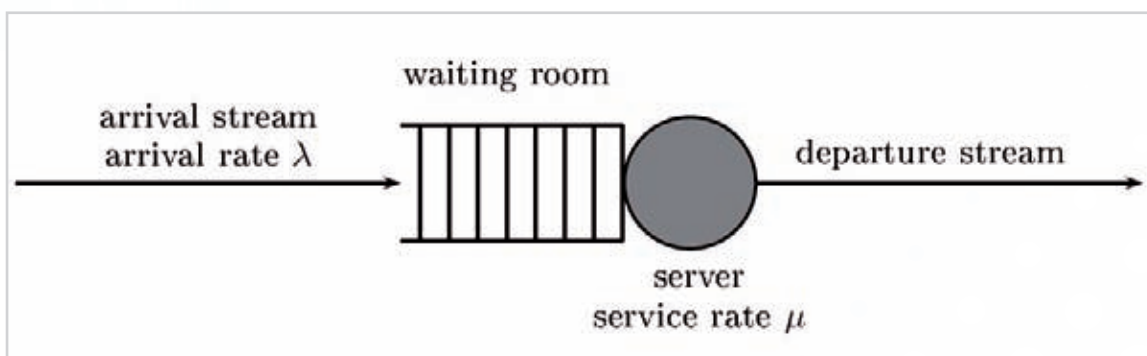


Abbildung 4: Grundmodell einer Bedienstation

Figure 4: Basic model of queueing systems

Des Weiteren können sogenannte heterogene Kunden modelliert werden, welche sich in einem Modell durch relevante Merkmale unterscheiden. Im Zusammenhang mit Gruppenankünften und -bedienung werden während des Batch-Processing Kundentyp-spezifische Stapel in der Batching-Area erzeugt.

Die Ankunfts- und Bedienrate, welche sich aus den Verteilungen der Ankunftszeitpunkte bzw. Zwischenankunftszeiten und Bedienzeiten errechnen, sind weitere bestimmende Faktoren eines Warteschlangenmodells. Sind die Verteilungen des Ankunftsstroms und der Bedienzeiten vom allgemeinem Typ (*general*), so wird das Wartesystem als ein allgemeines Wartesystem klassifiziert. Eine allgemeine Verteilung bedeutet, dass von ihr nur die Maßzahlen wie Erwartungswert und Varianz bekannt sind, jedoch nicht der genaue Typ der Verteilung.

Das Projekt befasst sich mit solchen allgemeinen Mehrbedienerwartesystemen (Abbildung 5), die Gruppenankünfte und -bedienung mit heterogenen Kunden und Batch-Processing beinhalten.

Die Allgemeingültigkeit bzw. der hohe Abstraktionsgrad bezüglich der allgemeinen Verteilungen des Ankunftsstroms und der Bedienzeiten hat den Nachteil, dass es schwierig ist sogenannte stationäre Zustandswahrscheinlichkeiten bzw. Verteilungen in einem System zu bestimmen. Stationäre Zustandswahrscheinlichkeiten sind in der Warteschlangentheorie bzw. auf dem Gebiet der sto-

chastischen Prozesse von besonderem Interesse, da über diese das zeitunabhängige Verhalten eines durch Zufall gesteuerten Systems beschrieben wird. Für allgemeine Wartesysteme ist keine geschlossene analytische Formel für stationäre Verteilungen bekannt. Daher muss auf Approximationsverfahren zurückgegriffen werden, welche seit den 50er Jahren erforscht werden. Um die Güte einer ermittelten Approximation zu beurteilen, werden Simulationsverfahren verwendet. Die Beurteilung der Approximationsgüte mittels Simulation wird durch einen Vergleich der Referenzdaten der Simulation, welche mit statistischen Methoden ausgewertet werden, und den Approximationsergebnissen umgesetzt.

Approximationsverfahren

Im Hinblick auf Warteschlangennetze, die durch den Zusammenschluss mehrerer Wartesysteme entstehen, werden Approximationsverfahren verwendet, denen gemein ist, dass sie von einer hohen Auslastung ausgehen, d. h., dass die Bediener nahezu durchgängig beschäftigt sind. Diese Grundannahme ist in Bezug auf Produktionssysteme realistisch, da ausgelastete Systeme in den meisten Fällen angestrebt werden.

Im Zuge des Projekts werden Approximationen untersucht, die auf der Heavy-Traffic-Approximation beruhen. Allen und Cunneen [1] haben die Erkenntnisse der Heavy-Traffic-Approximation

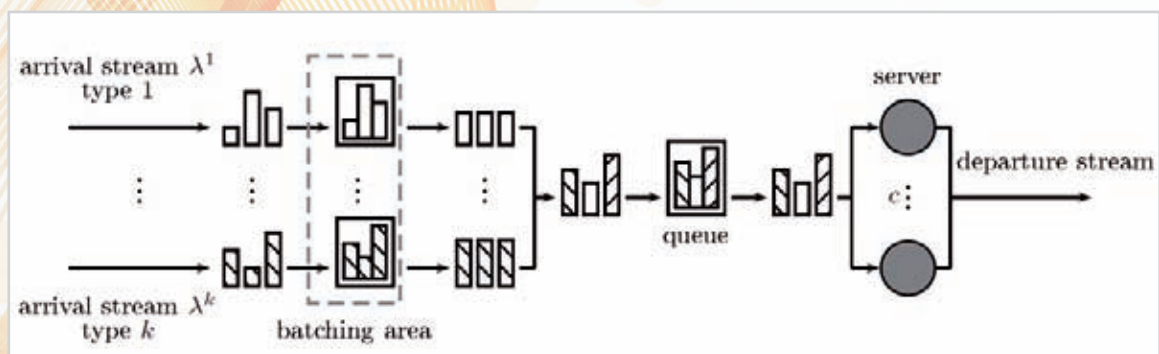


Abbildung 5: Mehrbedienerwartesystem mit Gruppenankünften, -bedienung, Batch-Processing und heterogenen Kunden

Figure 5: Multi-server queueing system with batch arrival and service, batch processing and heterogeneous jobs

In addition heterogeneous clients can be modeled, which differs in relevant characteristics of the queueing model. In connection to batch arrivals and service different type of customer-specific batches are generated during batch processing in the batching area.

The arrival and service rate, which are calculated from the distribution of arrival or interarrival and service times, are additional parameters of a queueing system. If the distributions of the arrival stream and the service times are general, then the queueing system is classified as a general queueing system. A general distribution means that only the parameters such as mean and variance are known, but not the exact type of the distribution. The project deals with such general multi-server queueing systems (figure 5) with batch arrival and service with heterogeneous jobs and batch processing.

The generality and the high level of abstraction with respect to the general distribution of the arrival stream and the service times have the disadvantage that it is difficult to determine so-called steady state probabilities and distributions in a system. Steady state probabilities and stationary distributions in the queueing theory and in the field of stochastic processes are of particular interest, since they describe the time-independent behavior of randomly controlled systems. For general queueing systems no closed analytical formula is known for steady state probabilities and stationary distributions. Therefore there is a need of approximation methods, which are developed since the 50s. Simulation methods may be used in order to assess the quality of calculated approximations. The assessment of the approximations by using methods of simulation is implemented by comparing the reference data of the simulation, which is evaluated by statistical methods, and the results of approximations.

Methods of approximation

In view of queueing networks, which consist of several connected queueing systems, approximation methods are used assuming a high utilization of queueing systems to be analysed. I. e. the servers of a queueing system are almost continu-

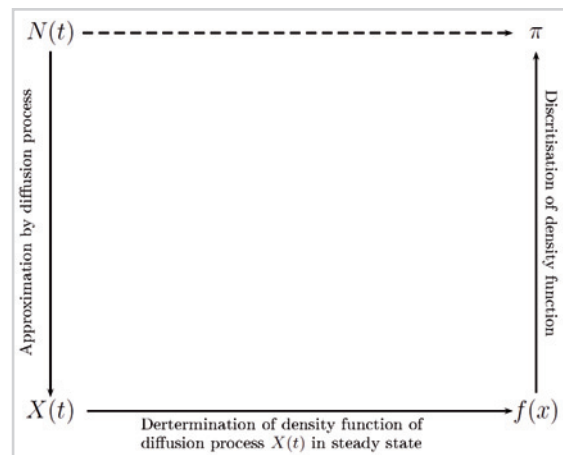


Abbildung 6: Grundidee der Diffusionsapproximation.

Figure 6: Basic idea of diffusion approximation.

ously busy. This assumption is realistic in terms of production systems, as an utilized system will be gained in most cases.

In the course of the project approximations are investigated, which are based on the heavy traffic approximation. Allen and Cunneen [1] have used the previous results on the heavy-traffic approximation to develop an analytical formula for determining the average number of customers in the system, $E[N]$, which was later extended by Krämer/Langenbach-Belz [2] with a correction term. The heavy-traffic approximation is also used for the approximation by the diffusion approximation using the Fokker-Planck equation to determine the average number of customers in a queueing system [3].

It suffices to determine $E[N]$ a priori since all other relevant parameters can be determined by the theorem of Little[4], which describes the general law between system parameters and performance parameters.

Other approximation methods, such as the fluid approximation are not appropriate for general queueing systems because this method uses recursive calculations, which does not consider effects appearing in relation with batch arrival and service.

genutzt und eine analytische Formel zur Bestimmung der mittleren Anzahl von Kunden im System, $E[N]$, entwickelt, welche später von Krämer/Langenbach-Belz [2] um einen Korrekturterm erweitert wurde. Die Heavy-Traffic-Approximation wurde ebenfalls zur approximativen Bestimmung der Leistungsgröße $E[N]$ durch die Diffusionsapproximation mittels der Fokker-Planck-Gleichung [3] verwendet.

Die Bestimmung der Leistungsgröße $E[N]$ a priori genügt, da alle weiteren relevanten Kenngrößen über den Satz von Little [4], welcher die allgemeine Gesetzmäßigkeit zwischen den Systemmerkmalen und den Leistungsgrößen beschreibt, bestimmt werden können.

Andere Approximationsverfahren, wie z. B. die Fluidapproximation, sind für die hiesige Betrachtung von allgemeinen Warteschlangensystemen nicht geeignet, da sie rekursive Berechnungen zum Ansatz haben, welche im Zusammenhang mit Gruppenankünften und -bedienung bestimmte Effekte nicht berücksichtigen.

Die Grundidee der Diffusionsapproximation ist die Approximation des mathematisch schwer zu handhabenden stochastischen Warteschlangenprozesses N zur Bestimmung der mittleren Anzahl von Kunden im System $E[N]$ durch einen Wiener-Prozess, der auch Brownsche Bewegung genannt wird. Der Wiener-Prozess gehört zur Klasse der Diffusionsprozesse. Die stationären Zustandswahrscheinlichkeiten des Wiener-Prozesses mit Drift 0 lassen sich mittels der Fokker-Planck-Gleichung bestimmen. Da der Wiener-Prozess jedoch einen kontinuierlichen Zustandsraum besitzt, der mathematische einfacher zu beherrschen ist, der Warteschlangenprozess N aber einen diskreten Zustandsraum hat, muss anschließend diskretisiert werden, um den Prozess N zu approximieren.

Die Anwendung der Approximation des Warteschlangenprozesses N durch die Brownsche Bewegung führt zu dem Problem, dass die Brownsche Bewegung auch negative Werte annehmen kann, was für den Warteschlangenprozess N jedoch nicht zugelassen ist. Um das Problem zu beheben, wird die Fokker-Planck-Gleichung durch das Hinzufügen einer absor-

bierenden Schranke und sofortiger Rückkehr modifiziert, so dass der Prozess in die positive Halbebene projiziert wird um sicher zu stellen, dass fast sicher nur nicht-negative Werte angenommen werden (Abbildung 7).

Aus der Fokker-Planck-Gleichung mit einer absorbierenden Schranke und sofortiger Rückkehr ergeben sich Differentialgleichungen, deren Lösung zu einer stationären Verteilung der modifizierten Brownschen Bewegung bzw. dem Wiener-Prozess führen, welche anschließend diskretisiert wird. Danach ist es möglich, die Leistungsgröße $E[N]$ des Warteschlangenprozesses N zu bestimmen. Die Methode kann jedoch nur angewandt werden, wenn das betrachtete Wartesystem eine hohe Auslastung besitzt.

Ein weiteres nützliches Approximationsverfahren von Allen/Cunneen beruht ebenfalls auf der Heavy-Traffic-Approximation. Auch hier wird angenommen, dass das Warteschlangensystem eine hohe Auslastung aufweist. Die Heavy-Traffic-Approximation von Kingman [S. 217, 1] besagt, dass die Wartezeit eines Auftrags in einer Warteschlange eines allgemeinen Wartesystems unter der Bedingung einer hohen Auslastung näherungsweise exponentialverteilt ist.

Da sich die gesamte Verweilzeit eines Auftrags in einem Wartesystem aus der Summe der Wartezeit in der Warteschlange und der Bedienzeit zusammensetzt (Abbildung 8), haben Allen und Cunneen die Erkenntnis über die Wartezeit in der Warteschlange genutzt und über empirische Analysen eine Approximationsformel für die mittlere Verweilzeit gefunden.

Die Approximationsformel liefert für Warteschlangensysteme, die eine exakte analytische Lösung besitzen, das exakte Resultat. Krämer/Langenbach-Belz haben die von Allen und Cunneen gefundene Approximationsformel um einen Korrekturterm erweitert, der eine Fallunterscheidung bezüglich der Ausprägung des quadrierten Variationskoeffizienten der Zwischenankunftszeiten, der die normierte Abweichung vom Erwartungswert misst, berücksichtigt.

Über den Satz von Little können anschließend die weiteren Leistungsgrößen bestimmt werden.

The basic idea of diffusion approximation is the approximation of the stochastic queueing process N for determining the average number of customers in $E[N]$ by a Wiener process also called Brownian motion, which is mathematically difficult to handle. The Wiener process belongs to the class of diffusion processes. The steady state probabilities of a Wiener process with drift 0 can be determined by means of the Fokker-Planck equation. Since the Wiener process has a continuous state space, which is mathematically easier to control, but the queueing process N has a discrete state space, there is the need to discretise in order to approximate the process N .

The utilisation of the approximation of the queueing process N by the Brownian motion leads to the problem that the Brownian motion can obtain negative values, which are not permitted in the queueing process N . To solve the problem, the Fokker-Planck equation is modified by adding an absorbing boundary and immediate return, thus the process is projected on to positive half-plane to ensure that almost certainly only non-negative values are assumed (figure 7).

From the Fokker-Planck equation with an absorbing boundary and immediate return occur differential equations, whose solutions lead to a stationary distribution of the modified Brownian

motion or the Wiener process that is then discretised. With the discretised result it is possible to determine the average number of jobs in the queueing system, $E[N]$ of the queueing process N .

The method can be applied only if the queueing system considers a high utilization.

Another useful approximation developed by Allen/Cunneen is also based on the heavy-traffic approximation. It is also assumed that a queueing system has a high utilization. The heavy-traffic approximation was found by Kingman [p. 217, 1] and provides the information, that the waiting time of a job in a queue of a general queueing system is exponentially distributed under the condition of high utilization approximation. Since the total time of a job spent in a queueing system is determined by the sum of the waiting time in the queue and the service time (figure 8), Allen and Cunneen have used the knowledge about the distribution of the waiting time in the queue and found by empirical analyzes an approximation formula for the average total time.

The approximation formula provides the exact result for queueing systems having an exact analytical solution. Krämer/Langenbach-Belz extended the approximation formula by Allen/Cunneen by a correction term, which considers a

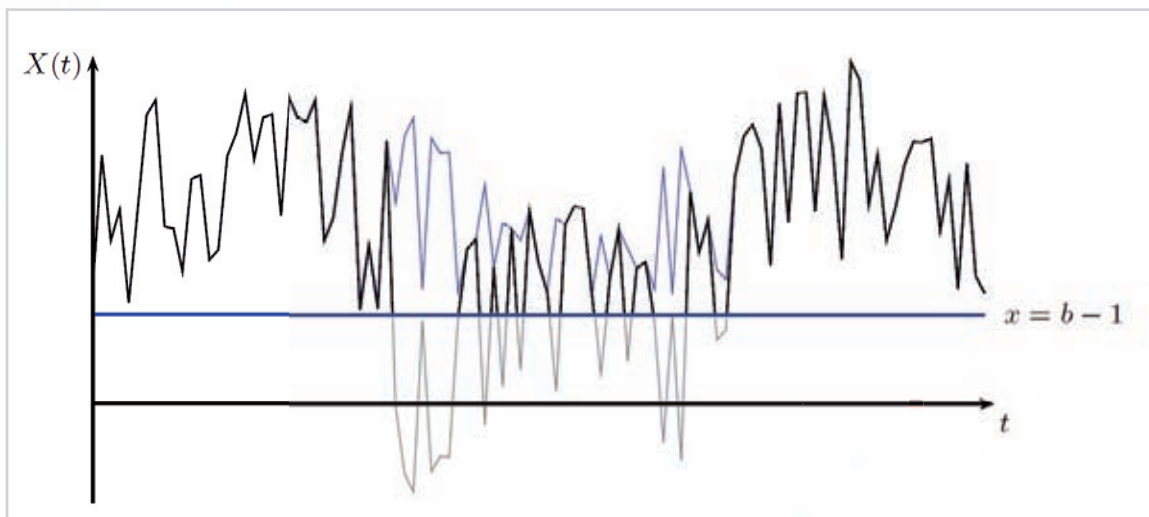


Abbildung 7: Brownsche Bewegung mit absorbierender Schranke und sofortiger Rückkehr.

Figure 7: Brownian motion with an absorbing boundary and immediate return.

Warteschlangennetze

Die beschriebenen Approximationsverfahren zur Leistungsbewertung wurden für einzelne Warteschlangensysteme entwickelt. In sehr vielen Fällen lässt sich ein Produktionsprozess nur über mehrere miteinander verbundene Warteschlangensysteme modellieren (Abbildung 9). Derartige Konstruktionen werden Warteschlangennetz genannt. Die Besonderheit besteht darin, dass der Abgangsstrom eines Systems, das keine Senke darstellt, den Ankunftsstrom der direkt nachfolgenden Systeme bildet. Darüber hinaus müssen bei Verzweigungen im Netz auch Übergangswahrscheinlichkeiten berücksichtigt werden.

Um dennoch die oben beschriebenen Approximationsverfahren verwenden zu können, wird eine Methode genutzt, die eine separierte Leistungsgrößenbewertung einzelner Warteschlangensysteme in einem Netz ermöglicht. Die sogenannte Dekompositionsmethode, welche nur angewandt werden kann, wenn die Auslastung der einzelnen Warteschlangensysteme hoch ist, ermöglicht eine isolierte Betrachtung der Systeme.

Die Dekompositionsmethode lässt sich in drei Schritte unterteilen. Im ersten Schritt werden die Abgangsströme der einzelnen allgemeinen Warteschlangensysteme approximativ gebildet. Um die Übergangswahrscheinlichkeiten innerhalb des Netzes zu berücksichtigen, wird der approximative Abgangsstrom anschließend im zweiten Schritt gemäß den Verzweigungen aufgesplittet. Im letzten Schritt werden, wenn es notwendig ist, d. h. ein Warte-

system mehr als ein Vorgänger-Warteschlangensystem hat, sich überlagernde Ströme zu einem Eingangsstrom zusammengefasst.

Stand der Forschung

Diffusionsapproximation

Die Diffusionsapproximation für allgemeine Einbedienersysteme (GI/GI/1) mit einer absorbierenden Schranke und sofortiger Rückkehr wurde erstmals von Duda [5] gelöst und von Gelenbe [6] zuvor als Idee besprochen. Ausgehend von der Fokker-Planck-Gleichung [S. 347 ff., 3]

$$\frac{d}{dt}f(x, t) = \frac{\alpha}{2} \cdot \frac{d^2}{dx^2}f(x, t) - \beta \cdot \frac{d}{dx}f(x, t),$$

womit sich die Übergangswahrscheinlichkeiten einer Brownschen Bewegung charakterisieren lassen, wird diese durch eine absorbierende Schranke mit sofortiger Rückkehr modifiziert, welche sicher stellt, dass fast sicher keine negativen Werte angenommen werden:

$$\frac{d}{dt}f(x, t) = \frac{\alpha}{2} \cdot \frac{d^2}{dx^2}f(x, t) - \beta \cdot \frac{d}{dx}f(x, t) + \Lambda R\delta(x - 1). \quad (1)$$

Λ beschreibt die durchschnittliche Verweildauer an der Schranke und $R(t)$ die Wahrscheinlichkeit, dass die Schranke zum Zeitpunkt t erreicht wird. Über δ ist die Diracsche δ -Verteilung angegeben. Die Diffusionskoeffizienten α und β werden über die Differenz bzw. Summe der Erwartungswerte und der Varianzen der mit dem Prozess N asso-

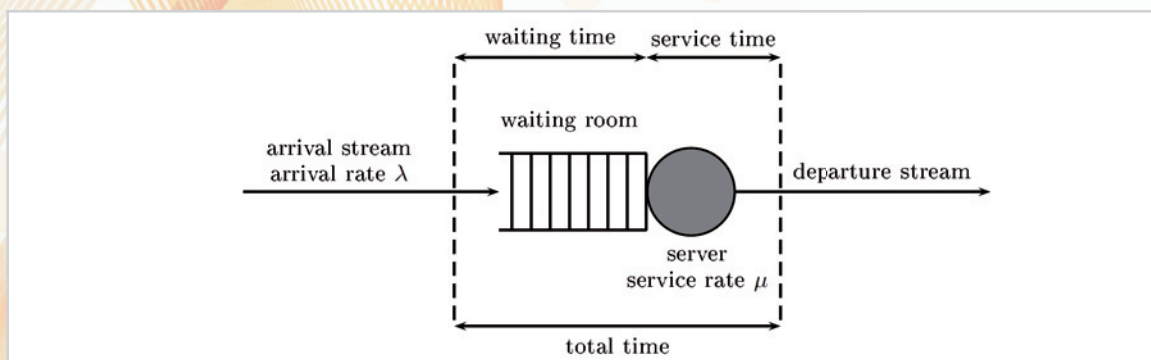


Abbildung 8: Anschauliche Darstellung von Verweil-, Warte- und Bedienzeit.

Figure 8: Graphic representation of waiting, service and total time.

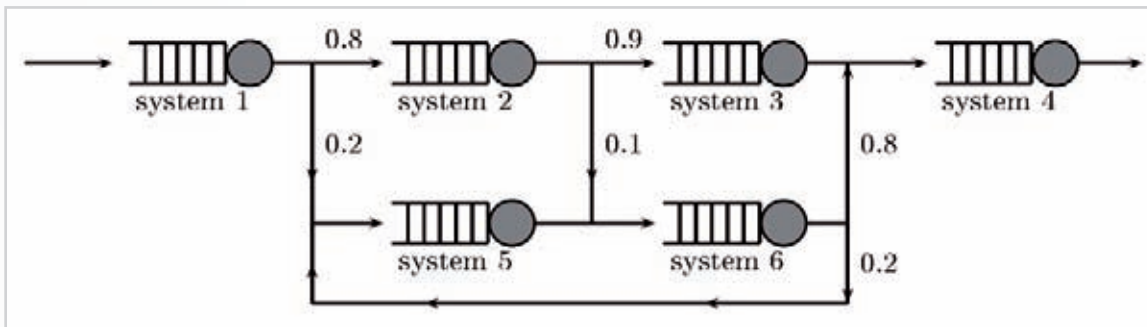


Abbildung 9: Beispiel für ein Warteschlangennetz.

Figure 9: Example for a queueing network.

case-by-case analysis of the squared coefficient of variation of the interarrival times measuring the normalized deviation from the mean.

Then using the theorem of Little further performance parameters can be determined subsequently.

Queueing networks

The approximation methods for performance parameters described above have been developed for single queueing systems. In many cases a production process can only be modeled by using several interconnected queueing systems (figure 9). This structure is called queueing network. The characteristic of queueing networks is that the departure stream of a system is the arrival of the next systems. In addition, transition probabilities between single systems in the network have to be taken into account.

To be able to use the approximation methods described above, a method is used that allows a separated performance parameters determination of single queueing systems in a network. The so-called decomposition method, which can be applied only if the utilisation of all queueing systems in a network is high, allows an isolated consideration of the systems.

The decomposition method consists of three steps. In the first step, the departure streams of single queueing systems are approximately composed.

To consider the transition probabilities inside the network, the departure stream is then splitted into the second step in accordance to the branching of the network. In the last step, if necessary, i. e. if a queueing system has more than one predecessor, the overlying arrival streams are summarised to one arrival stream.

State of the art

Diffusion approximation

The diffusion approximation with an absorbing boundary and immediate return for general single server queueing systems ($GI/GI/1$) was first solved by Duda [5] and pre-discussed as an idea by Gelenbe [6]. Starting from the Fokker-Planck equation [p. 347 et seq., 3]

$$\frac{d}{dt}f(x, t) = \frac{\alpha}{2} \cdot \frac{d^2}{dx^2}f(x, t) - \beta \cdot \frac{d}{dx}f(x, t),$$

allowing the characterisation of transition probabilities of a Brownian motion, which has been modified by an absorbing boundary and immediate return to ensure that almost certainly only non-negative values are assumed:

$$\frac{d}{dt}f(x, t) = \frac{\alpha}{2} \cdot \frac{d^2}{dx^2}f(x, t) - \beta \cdot \frac{d}{dx}f(x, t) + \Lambda R\delta(x-1). \quad (1)$$

Λ describes the average time a job spends at the boundary and $R(t)$ represents the probability that the boundary is reached at time t . δ specifies the Dirac δ distribution. The diffusion coefficients α

zierten Zählprozesse der Erneuerungsprozesse für den Ankunfts- und Abgangsstrom bzw. Bedienzeiten über den zentralen Grenzwertsatz für Erneuerungsprozesse bestimmt.

Die Rand- und Nebenbedingungen sind

$$\begin{aligned} \frac{dR(t)}{dt} &= \left(\beta \frac{d}{dx} f(x, t) - \frac{\alpha}{2} f(x, t) \right) \Big|_{x=0} - \Lambda R(t), \\ f(0, t) &= 0 \quad \text{für alle } t > 0, \\ 1 &= \int_0^\infty f(x, t) dx + R(t). \end{aligned}$$

Durch Lösung der Differentialgleichungen werden die stationären Zustandswahrscheinlichkeiten des modifizierten Diffusionsprozesses ermittelt, welche anschließend über die Methode $P(N(t)=n)=\rho(n, t)$ für

$n \geq 1$, welche sich als das Verfahren mit den besten Ergebnissen erwies, diskretisiert.

Für die mittlere Anzahl an Kunden im Prozess N ergibt sich dann

$$E[N] = \rho \left(1 + \frac{\rho SCV[I] + SCV[S]}{2(1 - \rho)} \right).$$

Die Diffusionsapproximation mittels Fokker-Planck-Gleichung für allgemeine Einbedienersysteme mit Gruppenbedienung ($GI/GI^{(b,b)}/1$) wurde

von Zisgen in [7] und [8] entwickelt. Die Idee besteht darin, dass der Rand für die Diffusionsapproximation von $x=0$ auf $x=b-1$ gesetzt wird und sich entsprechend die Rand- und Nebenbedingungen verändern. Der Ansatz ist naheliegend, denn wenn Stapel der Größe b vom Bediener bearbeitet werden und die Stapelgröße nicht ausreichend ist, also gleich oder kleiner als $b-1$ ist. Die Fokker-Planck-Gleichung ändert sich gegenüber (1) im letzten Term und hat die Form:

$$\frac{d}{dt} f(x, t) = \frac{\alpha}{2} \cdot \frac{d^2}{dx^2} f(x, t) - \beta \cdot \frac{d}{dx} f(x, t) + \Lambda R \delta(x - b) \quad (2)$$

mit der Randbedingung

$$\begin{aligned} \frac{dR(t)}{dt} &= \left(\beta \frac{d}{dx} f(x, t) - \frac{\alpha}{2} f(x, t) \right) \Big|_{x=b-1} - \Lambda R(t), \\ f(b-1, t) &= 0 \quad \text{für alle } t > 0, \\ 1 &= \int_{b-1}^\infty f(x, t) dx + R(t). \end{aligned}$$

Aus der Lösung der Differentialgleichung resultiert für das $GI/GI^{(b,b)}/1$ -System die mittlere Zeit zur Komplettierung eines Stapels von

$$\Lambda = \frac{b-1}{2} E[I]$$

und die mittlere Anzahl an Kunden im System von

$$E[N] = \frac{b-1}{2} + \rho \left(\frac{\tilde{\rho}\hat{\rho}}{1-\hat{\rho}} + \tilde{\rho}b - \frac{b-1}{2} \right) \quad (3)$$

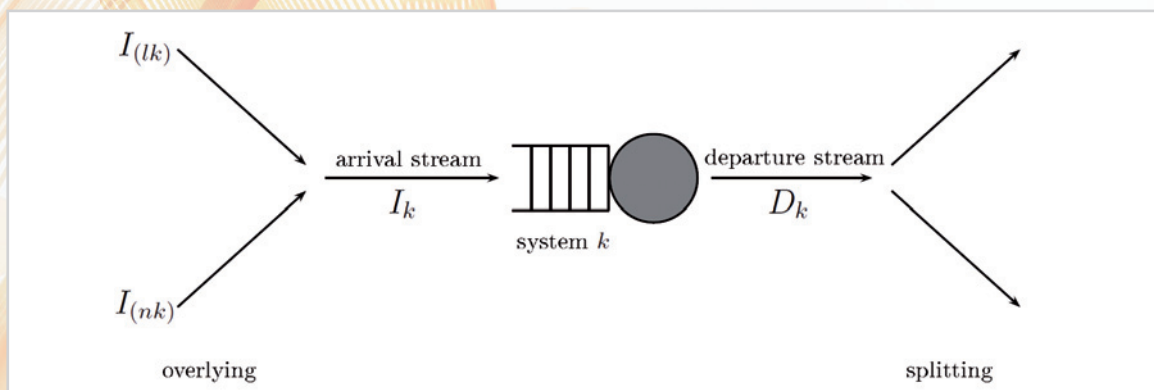


Abbildung 10: Dekompositionsmethode

Figure 10: Decomposition method

and β are determined by the difference or sum of the means and the variances of the associate renewal process for arrival and departure stream or service time of the counting process N by means of the central limit theorem for renewal processes.

The boundary and side conditions are described by

$$\begin{aligned} \frac{dR(t)}{dt} &= \left(\beta \frac{d}{dx} f(x, t) - \frac{\alpha}{2} f(x, t) \right) \Big|_{x=0} - \Lambda R(t), \\ f(0, t) &= 0 \quad \text{for all } t > 0, \\ 1 &= \int_0^{\infty} f(x, t) dx + R(t). \end{aligned}$$

By solving the differential equations the steady state probabilities of the modified diffusion process have been determined, which are then discretized then using the method $P(N(t)=n)=p(n, t)$ with $n \geq 1$, that provides the best results.

To obtain the average number of jobs in the process N , then

$$E[N] = \rho \left(1 + \frac{\rho SCV[I] + SCV[S]}{2(1-\rho)} \right).$$

The diffusion approximation by means of Fokker-Planck equation for general single server queueing systems with batch service ($GI/GI^{(b,b)}/1$) was developed by Zisgen in [7] and [8]. The idea is that the boundary of the diffusion approximation will be set from $x=0$ to $x=b-1$. This effects, that the boundary and side conditions analogously change. The approach is obvious, because if batches of size b are served and the current batch size is not sufficient then the size of the current batch is equal to $b-1$ or smaller. The Fokker-Planck equation changes with respect to (1) in the last term and has the form:

$$\frac{d}{dt} f(x, t) = \frac{\alpha}{2} \cdot \frac{d^2}{dx^2} f(x, t) - \beta \cdot \frac{d}{dx} f(x, t) + \Lambda R \delta(x-b) \quad (2)$$

with boundary and side conditions

$$\begin{aligned} \frac{dR(t)}{dt} &= \left(\beta \frac{d}{dx} f(x, t) - \frac{\alpha}{2} f(x, t) \right) \Big|_{x=b-1} - \Lambda R(t), \\ f(b-1, t) &= 0 \quad \text{for all } t > 0, \\ 1 &= \int_{b-1}^{\infty} f(x, t) dx + R(t). \end{aligned}$$

From the solution of the differential equation results for the $GI/GI^{(b,b)}/1$ system the average time to complete a batch is

$$\Lambda = \frac{b-1}{2} E[I]$$

and the average number of jobs in the system will be determined by

$$E[N] = \frac{b-1}{2} + \rho \left(\frac{\tilde{\rho} \hat{\rho}}{1-\tilde{\rho}} + \tilde{\rho} b - \frac{b-1}{2} \right) \quad (3)$$

specifying $\hat{\rho}$ by

$$\tilde{\rho} = \frac{1}{\gamma} \cdot \frac{2\Lambda}{\gamma\alpha} R(\hat{\rho}^{b-1} - \hat{\rho})$$

with $\gamma=2\beta/\alpha$ and $\hat{\rho}=\exp(\gamma)$.

The solution of the differential equation (2) can also be used for the $GI^X/GI^{(b,b)}/1$ system, which now also includes batch arrival, what was also investigated by Zisgen. The average time a job spends at the boundary is described by

$$\Lambda = \frac{E[X]}{b} E[I],$$

where $E[X]$ is the average batch size of incoming batches. The average number of jobs in the system corresponds to (3).

In addition to this procedure in [9] an approach for a more refined diffusion approximation using reflected stochastic differential equations has been developed. This approach leads to equivalent results in relation to the studies of the queueing system $GI/GI^{(b,b)}/1$ by Zisgen [8]. The advantage of this method is that the queueing process is partitioned, which allows to find a reason for the reflection by the lost service capacity in process downtime. Also the pathwise stochastic analysis is transparent and intuitive.

Allen/Cunneen formula

The initial approximation formula by Allen/Cunneen [p. 221, 1] determines approximately the performance parameter of the mean total time of the jobs in a general multi-server queueing system ($GI/GI/c$) and has the form

Wobei

$$\tilde{\rho} = \frac{1}{\gamma} \cdot \frac{2\Lambda}{\gamma\alpha} R(\hat{\rho}^{b-1} - \hat{\rho})$$

mit $\gamma = 2\beta/\alpha$ und $\hat{\rho} = \exp(\gamma)$ ist.

Die Lösung der Differentialgleichung (5) kann ebenfalls für das $G^X/G^{(b,b)}/1$ -System, das nun zusätzlich Gruppenankünfte beinhaltet, genutzt werden, was ebenfalls von Zisgen untersucht wurde. Für die mittlere Verweildauer am Rand ergibt sich

$$\Lambda = \frac{E[X]}{b} E[I],$$

wobei $E[X]$ die mittlere Gruppengröße der eintreffenden Kundengruppen ist. Die mittlere Anzahl Kunden im System entspricht (8).

Zusätzlich zu diesem Vorgehen wurde in [9] ein Ansatz für eine verfeinerte Diffusionsapproximation über die Formulierung reflektierter stochastischer Differentialgleichungen entwickelt. Der Ansatz liefert gleichwertige Ergebnisse in Bezug auf die Untersuchung des Wartesystems $GI/G^{(b,b)}/1$ von Zisgen [8]. Der Vorteil dieser Methode ist, dass der Warteschlangenprozess zerlegt wird, was die Begründung der Reflektion durch die verlorene Bedienkapazität bei Stillstand des Prozesses ermöglicht. Die pfadweise Betrachtung ist stochastisch transparenter und intuitiver.

Allen/Cunneen Formel

Die ursprüngliche Approximationsformel von Allen/Cunneen [S. 221, 1] ermittelt approximativ die Leistungsgröße der mittleren Verweilzeit der Aufträge in einem allgemeinen Mehrbedienersystem ($GI/GI/c$) und hat die Form

$$E[T] \approx \frac{\rho_c E[S]}{1 - \rho} \cdot \frac{SCV[I] + SCV[S]}{2c},$$

wobei ρ_c mit der Erlang-C-Formel bestimmt wird und die Wahrscheinlichkeit angibt, dass alle Bediener beschäftigt sind. ρ beschreibt die Auslastung des Systems, $E[S]$ ist der Erwartungswert der Verteilung der Bedienzeiten und $SCV[I]$ und $SCV[S]$ sind jeweils die quadrierten Variationskoeff-

izienten der Zwischenankunfts- und Bedienzeiten. Mit Korrekturterm g_{KLB} von Krämer/Langenbach-Belz [S. 271, 2] lautet die Approximationsformel für die mittlere Verweilzeit

$$E[T] \approx \frac{\rho_c E[S]}{1 - \rho} \cdot \frac{SCV[I] + SCV[S]}{2c} g_{KLB},$$

wobei der Korrekturterm durch

$$g_{KLB} = \begin{cases} \exp\left(-\frac{3}{2} \cdot \frac{1-\rho}{\rho_c} \cdot \frac{(1-SCV[I])^2}{SCV[I]+SCV[S]}\right), & \text{für } SCV[I] \leq 1 \\ \exp\left(-(1-\rho) \frac{SCV[I]-1}{SCV[I]+4SCV[S]}\right), & \text{für } SCV[I] > 1 \end{cases}$$

beschrieben ist. Dieser berücksichtigt die Stärke des normierten Maßes der Streuung um den Erwartungswert der Zwischenankunftszeiten. Die Approximation mittels der Allen/Cunneen Formel mit dem Korrekturterm g_{KLB} liefert sehr gute Ergebnisse, wenn $0 < SCV[I] \leq 1.25$ und $0 < SCV[S] \leq 1.25$ gilt [2].

Für allgemeine Wartesysteme mit Gruppenankünften und -bedienung ($G^X/GI^{(b,b)}/c$) wurde die Allen/Cunneen Formel mit Korrekturterm von Hanschke in [10] modifiziert:

$$E[T] \approx \frac{\rho_c E[S]}{1 - \rho} \cdot \frac{SCV[I^*] + SCV[S]}{2c} g_{KLB}$$

mit

$$g_{KLB} = \begin{cases} \exp\left(-\frac{3}{2} \cdot \frac{1-\rho}{\rho_c} \cdot \frac{(1-SCV[I^*])^2}{SCV[I^*]+SCV[S]}\right), & \text{für } SCV[I^*] \leq 1 \\ \exp\left(-(1-\rho) \frac{SCV[I^*]-1}{SCV[I^*]+4SCV[S]}\right), & \text{für } SCV[I^*] > 1. \end{cases}$$

Der Bediener verarbeitet die Aufträge in Stapeln der Größe b , was bedeutet, dass Stapel in dieser Größe erzeugt werden müssen (Abbildung 11). Dies geschieht in der Batching-Area, die dem eigentlichem Wartesystem vorgeschaltet ist (vergleiche Abbildung 5).

Die Grundidee der Modifikation besteht darin, dass die Stapel der Größe b im Wartesystem als einzelne Kunden aufgefasst werden und somit die Allen/Cunneen Formel weiterhin verwendet werden kann. Jedoch wird zuvor der ursprüngliche Eingangsstrom $(I_n)_{n \in \mathbb{N}}$ der die Zwischenankunftszeiten von Kundengruppen beschreibt, mittels des Grenzwertsatzes für Erneuerungsprozesse in einen Eingangsstrom $(I_n^*)_{n \in \mathbb{N}}$ von Stapeln der Größe b umgewandelt. Für den

$$E[T] \approx \frac{p_c E[S]}{1 - \rho} \cdot \frac{SCV[I] + SCV[S]}{2c},$$

where p_c is determined by the Erlang-C formula and represents the probability that all servers are busy. ρ describes the utilisation of the queueing system, $E[S]$ is the mean of the distribution of service times and $SCV[I]$ and $SCV[S]$ are the squared coefficients of variation of the interarrival and service times. Extending the approximation formula with correction term g_{KLB} of Krämer/Langenbach-Belz [p. 271, 2], the formula turns into

$$E[T] \approx \frac{p_c E[S]}{1 - \rho} \cdot \frac{SCV[I] + SCV[S]}{2c} g_{KLB},$$

with the correction term that is described by

$$g_{KLB} = \begin{cases} \exp\left(-\frac{3}{2} \cdot \frac{1-\rho}{p_c} \cdot \frac{(1-SCV[I])^2}{SCV[I]+SCV[S]}\right) & , \text{ if } SCV[I] \leq 1 \\ \exp\left(-\frac{1-\rho}{SCV[I]+4SCV[S]}\right) & , \text{ if } SCV[I] > 1 \end{cases}$$

The correction term considers the extent of the normalized parameter of spread around the mean of the interarrival times.

The approximation of the average total time a job spends in a queueing system according to the Allen/Cunneen formula with the correction term g_{KLB} provides very good results if $0 < SCV[I] \leq 1.25$ and $0 < SCV[S] \leq 1.25$ gilt [2] [2].

The formula by Allen/Cunneen with the correction term was modified by Hanschke in [10] for general multi-server systems with batch arrivals and service ($G^X/G^{(b,b)}/c$):

$$E[T] \approx \frac{p_c E[S]}{1 - \rho} \cdot \frac{SCV[I^*] + SCV[S]}{2c} g_{KLB}$$

with

$$g_{KLB} = \begin{cases} \exp\left(-\frac{3}{2} \cdot \frac{1-\rho}{p_c} \cdot \frac{(1-SCV[I^*])^2}{SCV[I^*]+SCV[S]}\right) & , \text{ if } SCV[I^*] \leq 1 \\ \exp\left(-\frac{1-\rho}{SCV[I^*]+4SCV[S]}\right) & , \text{ if } SCV[I^*] > 1 \end{cases}$$

The server processes the batches of size b , which means, a batch must be generated of size b (figure 11). The batches will be generated in the batching area being passed before entering the queueing system (see figure 5).

The basic idea of the modification is that the batches of size b can be taken as single jobs in the queueing system and thus the Allen/Cunneen formula can still be used. However first the original arrival stream $(I_n)_{n \in \mathbb{N}}$ describing the interarrival times of groups, has to be converted into an arrival stream $(I_n^*)_{n \in \mathbb{N}}$ of batches of size b by means of the central limit theorem for renewal processes. For the squared coefficient of variation of the modified arrival stream one gets

$$SCV[I^*] \approx \frac{E[X]}{b} (SCV[X] + SCV[I]),$$

where X , being a discrete non-negative random variable, describes the size of arriving groups of jobs.

General multi-server queueing systems with batch arrival and service with heterogeneous jobs of type j , $G^X/G^{(b,b)}/c$, were analysed in [11]. In the batching area now batches are generated being type specific with size b^j , which again requires a modification of the Allen/Cunneen formula. An important assumption that is made in this model is that the interarrival and service times of two different types of jobs are stochastically independent. To modify the Allen/Cunneen formula the basic idea of Hanschke [10] has been taken. The modification of arrival streams with specific type of jobs cause to overlay processes that are represented by Σ . Since the renewal processes $(I_n)_{n \in \mathbb{N}}$ and $(S_n)_{n \in \mathbb{N}}$ have the iid characteristic (*independent and identically distributed*), the overlay processes $(I_n^\Sigma)_{n \in \mathbb{N}}$ and $(S_n^\Sigma)_{n \in \mathbb{N}}$ consequently have also this characteristic. Using again the central limit theorem for renewal processes modifies the Allen/Cunneen formula, which now turns into

$$E[T] \approx \frac{p_c E[S^\Sigma]}{1 - \rho^\Sigma} \cdot \frac{SCV[I^{*\Sigma}] + SCV[S^\Sigma]}{2c} g_{KLB},$$

where

$$g_{KLB} = \begin{cases} \exp\left(-\frac{3}{2} \cdot \frac{1-\rho}{p_c} \cdot \frac{(1-SCV[I^{*\Sigma}])^2}{SCV[I^{*\Sigma}]+SCV[S^\Sigma]}\right) & , \text{ if } SCV[I^{*\Sigma}] \leq 1 \\ \exp\left(-\frac{1-\rho}{SCV[I^{*\Sigma}]+4SCV[S^\Sigma]}\right) & , \text{ if } SCV[I^{*\Sigma}] > 1 \end{cases}$$

The squared coefficient of variation for the modified arrival stream $(I_n^{*\Sigma})_{n \in \mathbb{N}}$ is described by

quadratierten Variationskoeffizienten des modifizierten Eingangstroms ergibt sich

$$SCV[I^*] \approx \frac{E[X]}{b} (SCV[X] + SCV[I]),$$

wobei X , eine diskrete nicht-negative Zufallsgröße, die Anzahl an Kunden der eintreffenden Kundengruppen beschreibt.

Allgemeine Wartesysteme mit Gruppenankünften und -bedienung mit heterogenen Kunden, $G^k/G^{(b,b)}/c$ mit heterogenen Kunden vom Typ j , wurden in [11] untersucht. In der Batching-Area werden nun Kundentyp-spezifische Stapel b^i erzeugt, was wiederum eine Modifikation der Allen/Cunneen Formel erfordert. Eine wesentliche Annahme, die in diesem Modell getroffen wird, ist, dass die Zwischenankunfts- und Bedienzeiten zweier unterschiedlicher Kundentypen stochastisch unabhängig sind. Es wurde die Grundidee von Hanschke [10] aufgegriffen. Die Modifikation der Kundentyp-spezifischen Eingangsströme führt zu Überlagerungsprozessen, die mit Σ gekennzeichnet sind. Da die Erneuerungsprozesse $(l_{n'}^{\Sigma})_{n' \in \mathbb{N}}$ und $(S_{n'}^{\Sigma})_{n' \in \mathbb{N}}$ die i.i.d. Eigenschaft (*independent and identically distributed*) aufweisen, besitzen folglich die Überlagerungsprozesse $(I_{n'}^{\Sigma})_{n' \in \mathbb{N}}$ und $(S_{n'}^{\Sigma})_{n' \in \mathbb{N}}$ ebenfalls diese Eigenschaft. Über den zentralen Grenzwertsatz für Erneuerungsprozesse erfolgt die Modifikation der Allen/Cunneen Formel, welche nun die Form

$$E[T] \approx \frac{\rho_c E[S^{\Sigma}]}{1 - \rho^{\Sigma}} \cdot \frac{SCV[I^{*\Sigma}] + SCV[S^{\Sigma}]}{2c} g_{KLB},$$

hat mit

$$g_{KLB} = \begin{cases} \exp\left(-\frac{3}{2} \cdot \frac{1-\rho}{\rho_c} \cdot \frac{(1-SCV[I^{*\Sigma}])^2}{SCV[I^{*\Sigma}] + SCV[S^{\Sigma}]}\right) & \text{für } SCV[I^{*\Sigma}] \leq 1 \\ \exp\left(-\frac{1-\rho}{SCV[I^{*\Sigma}] + 4SCV[S^{\Sigma}]}\right) & \text{für } SCV[I^{*\Sigma}] > 1. \end{cases}$$

Der quadrierte Variationskoeffizient für den modifizierten Eingangstrom $(I_{n'}^{\Sigma})_{n' \in \mathbb{N}}$ ergibt sich durch

$$SCV[I^{*\Sigma}] \approx \sum_{j=1}^k \frac{\lambda^{*j}}{\lambda^{*\Sigma}} SCV[I^{*j}],$$

mit

$$SCV[I^{*j}] \approx \frac{E[X^j]}{b^j} (SCV[X^j] + SCV[I^j]).$$

Dekomposition

Bisher gibt es nur wenige Arbeiten zur Dekompositionsmethode, die sich speziell mit der Dekomposition von allgemeinen Wartesystemen des Typ $G^k/G^{(b,b)}/c$ beschäftigen. Zisgen ging von der ursprünglichen Dekompositionsmethode aus [S. 479 ff., 2],[12] und entwickelte diese in [7] weiter. Die Grundannahme der Dekompositionsmethode ist, dass Wartesysteme in einem Netz als stochastisch unabhängig betrachtet werden können. Da weiterhin von einer hohen Auslastung ausgegangen wird, die in der Praxis häufig im Fokus steht, können die Abgangsströme der Wartesysteme als Erneuerungsprozesse aufgefasst werden. Im ersten Schritt der Dekompositionsmethode werden die Abgangsströme der einzelnen Wartesysteme approximativ bestimmt, wobei die Annahme getroffen wird, dass die Zwischenankunfts- und Bedienzeiten stochastisch unabhängig sind:

$$D_{(i)}^b = \begin{cases} S_{(i)} & \text{, mit Wahrscheinlichkeit } \rho_{(i)} \\ A_{(i)} + S_{(i)} & \text{, mit Wahrscheinlichkeit } 1 - \rho_{(i)}. \end{cases}$$

$D_{(i)}^b$ beschreibt die näherungsweise Zwischenabgangszeiten am Wartesystem i im Netz und $A_{(i)}$ ist die Zeit, die benötigt wird, um einen Stapel zu komplettieren, was $I_{(i)}^*$ des i -ten Wartesystems entspricht (vergleiche Abbildung 11 und [13]). Mit der 1. Waldschen Gleichung ergeben sich die ersten zwei Momente des approximierten Erneuerungsprozesses:

$$\begin{aligned} E[D_{(i)}^b] &= E[T_{(i)}]E[I_{(i)}^X] = E[A_{(i)}] \\ \text{Var}[D_{(i)}^b] &= E[S_{(i)}^2] + 2(1 - \rho_{(i)})E[S_{(i)}]E[T_{(i)}]E[I_{(i)}^X] \\ &\quad + (1 - \rho_{(i)})(E[T_{(i)}]\text{Var}[I_{(i)}^X] + E[T_{(i)}^2]E[I_{(i)}^X]^2). \end{aligned}$$

Über die ersten beiden Momente lässt sich dann der quadrierte Variationskoeffizient berechnen

$$SCV[D_{(i)}^b] = \rho_{(i)}^2 SCV[S_{(i)}] + (1 - \rho_{(i)}) \left(\rho_{(i)} + \frac{SCV[I_{(i)}^X]}{E[T_{(i)}]} + SCV[T_{(i)}] \right),$$

welcher im letzten Schritt benötigt wird.

Im nächsten Schritt erfolgt die Aufspaltung der ermittelten Abgangsströme gemäß den Übergangswahrscheinlichkeiten im Netz. Ausgehend vom Wartesystem i und dessen nachfolgenden Wartesystemen k kann die Aufspaltung des Abgangsstroms von i als Bernoulli-Experiment aufgefasst werden, wobei die Zufallsgröße M , welche die Wahrscheinlichkeit repräsentiert, dass

$$SCV[I^{*\Sigma}] \approx \sum_{j=1}^k \frac{\lambda^{*j}}{\lambda^{*\Sigma}} SCV[I^{*j}],$$

with

$$SCV[I^{*j}] \approx \frac{E[X^j]}{b^j} (SCV[X^j] + SCV[I^j]).$$

Decomposition

So far there have been few publications on the decomposition method that deal specifically with general multi-server queueing systems $GI^X/GI^{(b,b)}/c$. Zisgen used first approaches of decomposition methods [p. 479 et seq., 2] and [12] and extended the method for $GI^X/GI^{(b,b)}/c$ in [7]. The basic assumption of the decomposition method is that queueing systems in a network can be considered as stochastically independent. Since high utilization of the servers, which often has its focus in practise, is still expected the departure process of queueing systems treated as a renewal process. In the first part of the decomposition method, the departure process of the individual queueing systems is determined approximately. The assumption is used that the interarrival and service times are stochastically independent:

$$D_{(i)}^b = \begin{cases} S_{(i)} & , \text{ with probability } \rho_{(i)} \\ A_{(i)} + S_{(i)} & , \text{ with probability } 1 - \rho_{(i)}. \end{cases}$$

$D_{(i)}^b$ describes the approximate interdeparture times at the queueing system i in the network and $A_{(i)}$ represents the time it takes to complete a batch of the i -th queueing system, which corresponds to $I_{(i)}^*$ (see figure 11 and [13]). With the 1st Wald's equation the first and second moment of the approximated renewal process results:

$$\begin{aligned} E[D_{(i)}^b] &= E[T_{(i)}]E[I_{(i)}^X] = E[A_{(i)}] \\ Var[D_{(i)}^b] &= E[S_{(i)}^2] + 2(1 - \rho_{(i)})E[S_{(i)}]E[T_{(i)}]E[I_{(i)}^X] \\ &\quad + (1 - \rho_{(i)})(E[T_{(i)}]Var[I_{(i)}^X] + E[T_{(i)}^2]E[I_{(i)}^X]^2). \end{aligned}$$

Then with the first and second moments the squared coefficient of variation can be calculated:

$$SCV[D_{(i)}^b] = \rho_{(i)}^2 SCV[S_{(i)}] + (1 - \rho_{(i)}) \left(\rho_{(i)} + \frac{SCV[I_{(i)}^X]}{E[T_{(i)}]} + SCV[T_{(i)}] \right),$$

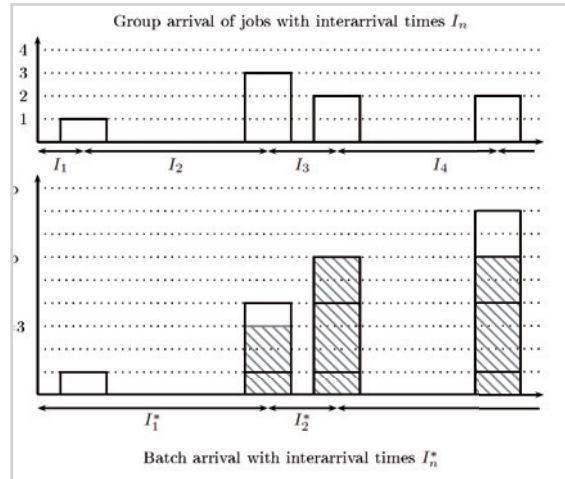


Abbildung 11: Batch-Processing

Figure 11: Batch processing

which is required in the last part of the method. The next step is to split the determined departure streams according to the transition probabilities in the network. Based on queueing system i and its subsequent queueing systems k the splitting of the departure stream of i can be considered as a Bernoulli experiment with the random variable M representing the probability, that the M -th batch sends to station k , is geometrically distributed. The first and second moment of the interarrival times at the queueing system k is calculated by

$$\begin{aligned} E[I_{(ik)}^X] &= \frac{1}{\underbrace{P_{(ik)}}_{E[M]}} E[T_{(i)}]E[I_{(i)}^X] \\ Var[E_{(ik)}^X] &= \frac{1}{\underbrace{P_{(ik)}}_{E[M]}} Var[D_{(i)}^b] + \frac{1 - P_{(ik)}}{\underbrace{P_{(ik)}^2}_{Var[M]}} E[T_{(i)}]^2 E[I_{(i)}^X]^2. \end{aligned} \tag{4}$$

Since the queueing system k can have several arrival streams in a network, the final part deals with overlaying of the arrival streams, which is composed of the departure streams of the predecessor queueing systems. The mean $E[I_{(k)}^X]$ is obtained from the sum of the weighted transition probabilities of departure streams of the predecessor in (6):

$$E[I_{(k)}^X] = \sum_{i=1}^K \frac{1}{P_{(ik)}} E[T_{(i)}]E[I_{(i)}^X].$$

der M -te Stapel zu Station k wechselt, geometrisch verteilt ist. Für die mittlere Zwischenankunftszeit am Wartesystem k ergibt sich für die ersten zwei Momente

$$E[I_{(ik)}^X] = \frac{1}{\underbrace{P_{(ik)}}_{E[M]}} E[T_{(i)}] E[I_{(i)}^X] \quad (4)$$

$$Var[E_{(ik)}^X] = \frac{1}{\underbrace{P_{(ik)}}_{E[M]}} Var[D_{(i)}^b] + \frac{1 - P_{(ik)}}{\underbrace{P_{(ik)}^2}_{Var[M]}} E[T_{(i)}]^2 E[I_{(i)}^X]^2.$$

Da das Wartesystem k in einem Netz mehrere Eingangsströme besitzen kann, besteht der letzte Schritt in der Überlagerung der Eingangsströme, welche sich aus den ermittelten Abgangsströmen der Vorgänger-Wartesysteme zusammensetzen. Der Erwartungswert $E[I_{(k)}^X]$ ergibt sich aus der Summe der mit den Übergangswahrscheinlichkeiten gewichteten Abgangsströme der Vorgänger aus (6):

$$E[I_{(k)}^X] = \sum_{i=1}^K \frac{1}{P_{(ik)}} E[T_{(i)}] E[I_{(i)}^X].$$

Entsprechend ist dann die Ankunftsrate $\lambda_{(k)}^X = 1 / E[I_{(k)}^X]$. Die Varianz wird über den zentralen Grenzwertsatz ermittelt und dadurch, dass angenommen wird, dass die Abgangsströme unabhängig sind, können die Varianzen bzw. die quadrierten Variationskoeffizienten des Überlagerungsprozesses aufsummiert werden

$$SCV[I_{(k)}^X] = \frac{1}{\lambda_{(k)}^X} \sum_{i=1}^K \frac{(P_{(ik)} SCV[D_{(i)}^b] + 1 - P_{(ik)}) P_{(ik)}}{E[T_{(i)}] E[I_{(i)}^X]}.$$

Die Bestimmung der mittleren Anzahl an Aufträgen im Warteschlangennetz erfolgt über

$$E[N] = \sum_{k=1}^K E[N_k].$$

Die Erwartungswerte für die Anzahl an Aufträgen in einem Wartesystem k können über die Allen/Cunneen Formel oder die Diffusionsapproximation gewonnen und aufsummiert werden.

Forschungsaufgaben

Die Forschungsaufgabe besteht nun darin, die Dekompositionsmethode für ein Netz von Wartesystemen vom Typ $G^X/G^{(b,b)}/c$ mit heterogenen Kunden weiterzuentwickeln. Dazu werden Erkenntnisse, die bereits in [11] gewonnen wurden und die Dekompositionsmethode für $G^X/G^{(b,b)}/c$ mit homogenen Kunden, genutzt.

Im ersten Schritt werden Simulationsmodelle aufgebaut und Experimente durchgeführt. Anschließend wird die Dekompositionsmethode für den Multi-Class-Fall entwickelt. Der aus der Simulation entstandene Datensatz bildet dann die Grundlage für die Bewertung der approximativ berechneten Ergebnisse mit Hilfe der Diffusionsapproximation und Allen/Cunneen Formel für den Mehrbedienerfall, wobei die Diffusionsapproximation für den Mehrbedienerfall mit Batch-Processing und unterschiedlichen Kundentypen noch entwickelt werden muss.

Zusätzlich wird nach Vereinfachungen bzgl. des Batch-Processing gesucht. Eventuell lassen sich faktorielle Übereinstimmungen zwischen $M^X/G^{(b,b)}/c$ - und $G^X/G^{(b,b)}/c$ -Systemen identifizieren, die zukünftige Leistungsgrößenberechnungen vereinfachen.

Implementierung

Am Ende des Projekts soll ein Tool für die warteschlangentheoretische Optimierung von Produktionsprozessen stehen. Denkbar ist eine Client-Server-Architektur, die entsprechende Schnittstellen aufweist für den benötigten Input, um eine Leistungsgrößenbewertung vorzunehmen. Da der Anwendungsbereich speziell in großen Unternehmen liegt, ist es sinnvoll, dass mehrere Anwender gleichzeitig auf die zur Verfügung gestellte Toolbox zugreifen können. Da insbesondere die Performance ausschlaggebend ist, um möglichst schnell eine Entscheidungsgrundlage zu generieren, sind Ansätze wie paralleles Rechnen und der Einsatz eines Frameworks für Netzflüsse, um Warteschlangennetze optimal abzubilden, vorstellbar.

Hence the arrival rate is given by $\lambda_{(k)}^X = 1/E[I_{(k)}^X]$. The variance is determined by the central limit theorem and thereby it is assumed, that the departure streams are independent, the variance or the squared coefficient of variation of the overlaying process can be summed

$$SCV[I_{(k)}^X] = \frac{1}{\lambda_{(k)}^X} \sum_{i=1}^K \frac{(P_{(ik)} SCV[D_{(i)}^b] + 1 - P_{(ik)}) P_{(ik)}}{E[T_{(i)}] E[I_{(i)}^X]}.$$

The average number of jobs in the queueing network is

$$E[N] = \sum_{k=1}^K E[N_k].$$

The average of the number of jobs in a queueing system k can be proceeded by the Allen/Cunneen formula or the diffusion approximation and then summed.

Research tasks

The research task is to develop the decomposition method for a network of multi-server systems $G^X/G^{(b,b)}/c$ with heterogeneous jobs, using knowledge, which was already gained in [11] and the approach of decomposition method for $G^X/G^{(b,b)}/c$ with homogeneous jobs.

In the first step, simulation models will be constructed and experiments performed. Then the decomposition method for multi-class case will be developed. The results from the simulation data then form the basis for the evaluation of the approximate results calculated by using the diffusion approximation and the Allen/Cunneen formula for the multi-user case. The diffusion approximation for the multi-server case with batch processing and heterogeneous jobs has yet to be developed.

In addition it will be searched for simplification in relating to batch processing. Maybe it is possible to find factorial matches between $M^X/G^{(b,b)}/c$ and $G^X/G^{(b,b)}/c$ systems that simplify future performance parameters calculations.

Implementation

At the end of the project a new tool for the queueing optimisation of production processes should be created. Conceivable is a client-server architecture with appropriate interfaces for the required input data to calculate performance parameters of queueing networks. Since the scope of application is arranged especially in large companies, it makes sense that several users can simultaneously access the provided toolbox. Especially since the performance of the tool is a crucial factor in order to quickly generate a foundation for decisions, approaches like parallel computing and the use of a framework for network flows to optimally reproduce queueing networks could be applied. Furthermore it is necessary to show the results graphically to obtain useful information for the user to allow him a more rapid interpretation. There are many potential classification algorithms, which could classify the performance parameters of queueing systems and identify critical areas.

Literatur // References

- [1] Arnold O. Allen. Probability, Statistics and Queueing Theory: With Computer Science Application, Academic Press, Florida, 1. Edition, 1978.
- [2] Gunter Bolch, Stefan Greiner, Hermann de Meer, and Kishor Shridharbhai Trivedi. Queueing Networks and Markov Chains - Modeling and Performance Evaluation with Computer Science Applications. Wiley, New York, 2. Edition, 2006.
- [3] Donald Gross and Carl M. Harris. Fundamentals of Queueing Theory. John Wiley & Sons, Incorporated, New York, 3rd Edition, 1998.
- [3] John D. C. Little. A proof for the queueing formula: $L = \lambda w$. Operations Research, 9(3):383-387, 1961.
- [4] Andrzej Duda. Transient Diffusion Approximation for Some Queueing Systems. In Proceedings of the 1983 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '83,

60

Des Weiteren ist insbesondere eine graphische Darstellung der berechneten Ergebnisse nützlich, um die gewonnenen Informationen für den Nutzer zugänglicher zu gestalten und ihm eine schnellere Interpretation erlauben. Vorstellbar sind Klassifikationsalgorithmen, die die Leistungsgrößenbewertungen der Bedienstationen klassifizieren und kritische Bereiche aufzeigen.

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

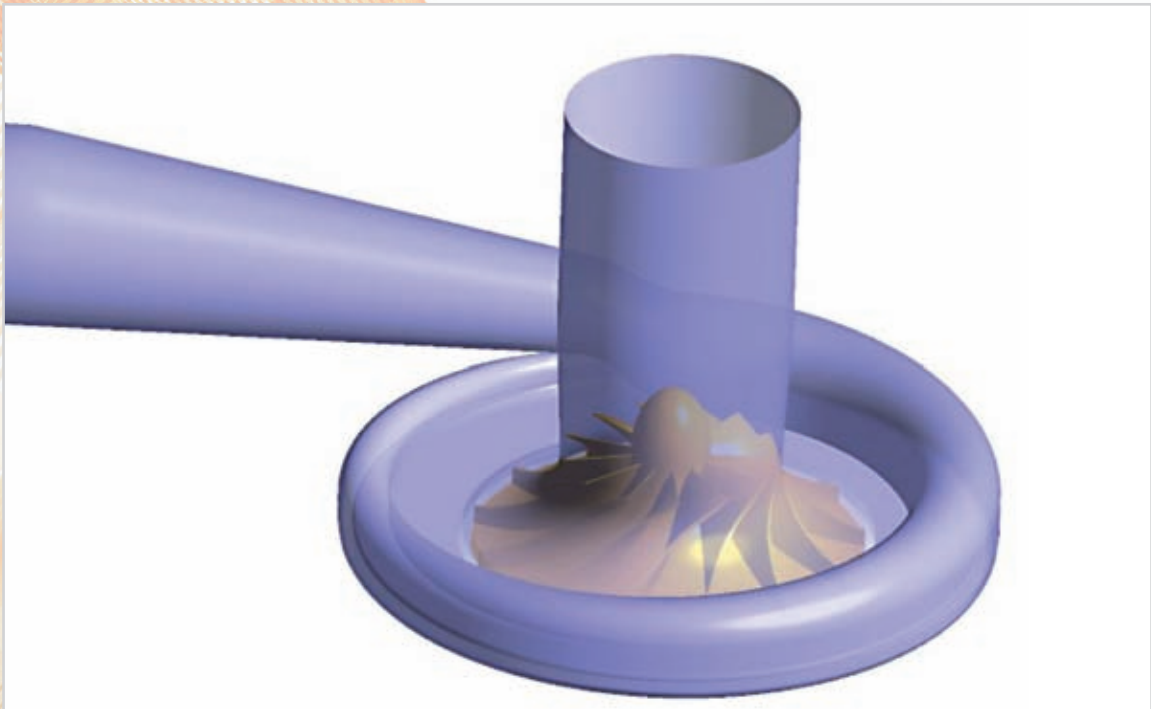
- pages 118-128, New York, NY, USA, 1983. ACM.
- [5] Erol Gelenbe. On approximate computer system models. *J. ACM*, 22(2):261-269, April 1975.
 - [6] Horst Zisgen. Warteschlangennetzwerke mit Gruppenbedienung. TU Clausthal, 1999. Dissertation.
 - [7] Horst Zisgen. A diffusion approximation for batch queues of type $G^X/G^{(b,b)}/1$. *Performance Evaluation*, 66(9/10):557-563, 2009.
 - [8] Lars Dohse. Diffusionsapproximation von Warteschlangensystemen mit Gruppenbedienung. TU Clausthal, 2005. Dissertation.
 - [9] Thomas Hanschke. Approximations for the mean queue length of the $GI^X/G^{(b,b)}/c$ queue. *Oper. Res. Lett.*, 34(2):205-213, 2006.
 - [10] Tobias Gröger. Warteschlangensysteme mit Gruppenankünften, Gruppenbedienung und heterogenen Kunden. TU Clausthal, 2007. Diplomarbeit.
 - [11] Ward Whitt. The queueing network analyzer. *Bell System Technical Journal*, 62(9):2779-2815, 1983.
 - [12] Thomas Hanschke and Horst Zisgen. Queueing networks with batch service. *European Journal of Industrial Engineering*, 5(3):313-326, 2011.

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 für Simulationsanwendungen installiert. Die größten Parallelrechner in Deutschland haben nur einen Bruchteil der Leistung dieser Rechner und sind an den drei Standorten München (LRZ und RZG), Stuttgart (HLRS) und Jülich (FZ) konzentriert. In Niedersachsen stehen für große Simulationsaufgaben der HLRN II (Norddeutscher Verbund für Hoch- und Höchstleistungsrechnen) und einige kleinere Anlagen an universitären Rechenzentren zur Verfügung. Um noch mehr Rechenleistung zu erhalten, ist es notwendig, die numerischen Probleme auf die bereits existierende Hardware-Infrastruktur an den Universitäten und ihre Rechenzentren zu verteilen. Man spricht dabei von verteilter Simulation. Leider ist dabei der Aufwand für die Erstellung der Software und der

Middleware, die zwischen den Anwendungen und den Betriebssystemen sitzt, 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 dem Aufkommen des Grid Computing 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.



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 for simulation applications. The most powerful parallel computers in Germany together have only a fraction of the power of 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 some smaller systems in university computer centers available for large simulation tasks. To get even more computing power, it is necessary to split the numerical problems so they can be calculated on the existing hardware infrastructure at universities and their data centers. Unfor-

tunately 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. With the advent of Grid Computing is becoming apparent, that the cost of the distribution of simulation applications comes on existing PCs and workstations in an area where it will be interesting for 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.

Eine cloud-basierte Software für verteilte Simulation

Teilprojekt – Leistungsfähigkeit der Cloud bei verteilter Simulation

Harald Richter, Alexander Keidel, Roman Ledyayev

Einleitung

Cloud Computing hat sich zum neuen, allgegenwärtigen Vorbild für Rechen- und Speicherdienste entwickelt. Die Gründe dafür sind die Kostenabrechnung nach tatsächlich verbrauchten Ressourcen und die intrinsische „Dehnbarkeit“, d.h. Anpassbarkeit an Benutzerwünsche. Firmen, Institute und Einzelpersonen können davon profitieren, indem sie ihre Rechen- und Speicherbedarfe von kommerziellen Cloud-Anbietern (CSPs) durchführen lassen, denn deren Dienste erstrecken sich mittlerweile von einfachen Daten-Backups bis hin zu kompletten virtuellen Rechenzentren. Diese Vorteile machen Cloud Computing auch für Wissenschaftler attraktiv, denn sie müssen dadurch nicht mehr eine eigene IT-Infrastruktur vorhalten und betreiben, sondern können diese an CSPs auslagern, die die Wissenschaftler-IT als „virtualisierte IT“ betreiben.

Allerdings wird die Situation komplizierter, wenn es um Hoch- und Höchstleistungsrechnen (HPC) geht, wie es beispielsweise bei Simulationen oder bei der Analyse großer Datenmengen der Fall ist. Dies gilt, sobald diese Anwendungen auf einer Cloud ausgeführt werden sollen. Die Gründe dafür sind, dass HPC-Programme hoch effizient und skalierbar bleiben müssen, auch für den Fall, dass sie von mehreren oder sogar vielen virtuellen Cores, Prozessoren oder Servern parallelisiert ausgeführt werden. Leider ist dies in Standard-Clouds nicht gegeben, wie verschiedene Messungen gezeigt haben. Beispielsweise stellt das US Department of Energy (DoE), das für das Hoch- und Höchstleistungsrechnen in den U.S.A. zuständig ist, die Nützlichkeit von Clouds für HPC

[1] grundsätzlich in Frage [1]. Andere Autoren, die dieselbe Meinung haben, sind [4], [5] und [6]. Wir glauben deshalb, dass erhebliche Forschungsanstrengungen notwendig sind, um die Ausführungseffizienz von Clouds bei HPC zu verbessern. In diesem Beitrag werden diverse Gründe für Cloud-Ineffizienzen bei der Verwendung von OpenStack als Cloud-Betriebssystem und von OpenFoam als Beispiel-HPC-Code angeführt. Es werden Vorschläge gemacht, wie man diese Probleme lösen bzw. umgehen kann. Die Ergebnisse können auf andere Cloud-Betriebssysteme und andere HPC-Codes übertragen werden.

Dieser Beitrag ist wie folgt aufgebaut: In Kapitel 2 wird der Stand der Technik rezensiert. Kapitel 3 beschreibt unser Projekt und welche Ausrüstung und Werkzeuge wir dafür verwendet haben. In Kapitel 4 werden die durchgeführten Messungen und Ergebnisse präsentiert und diskutiert. Der Artikel endet mit einem Ergebniskapitel, gefolgt von einem Ausblick und einer Literaturliste.

Stand der Technik

Es wurden von uns verschiedene wissenschaftliche HPC-Projekte studiert, die sich mit Optimierungen und Scheduling in Clouds beschäftigt haben. Außerdem wurde eine gründliche Literaturstudie durchgeführt, um herauszufinden, welche Cloud-Eigenschaften verbessert werden sollten, um Clouds HPC-fähig zu machen. Auch die Autoren dieses Beitrags haben bereits über die Leistungsfähigkeit von Clouds in [12] geforscht. Die Fachaufsätze, die hier rezensiert werden sollen, sind [4]-[11] (in der Reihenfolge ihrer

A Cloud-Based Software Infrastructure for Distributed-Simulation

Sub Project – Cloud Performance for Distributed-Simulation

Harald Richter, Alexander Keidel, Roman Ledyayev

Introduction

Cloud computing has become the new ubiquitous computing and storage paradigm, because of its pay-as-you-go accounting, its inherent elasticity, flexibility and user customising. Companies, institutions and individuals can profit from this new paradigm by off-loading computing and storage to commercial cloud service providers (CSPs), because CSP services range from simple data backups to entire virtual data centres. These advantages make cloud computing also attractive for scientists, because researchers do not need any more to provide and maintain their own IT-infrastructure, but can out-source their IT to CSPs, which is then hosted as ‘virtual IT’.

However, things are getting more intricate for the case of High-Performance Computing (HPC), such as simulations and big data analysis, which are intended to be executed by a cloud. The reasons for that are that HPC codes require high efficiency and scalable speed-up in case of parallel execution on multiple virtual cores, virtual CPUs and virtual servers. Unfortunately, this is not the case in standard clouds as various measurements have shown. For example, the US Dep. of Energy (DoE), which is responsible for HPC in the USA, is questioning the usefulness of clouds for HPC in general in [1]. Other authors with the same opinion are [4], [5] and [6], for example. We believe therefore, that significant research effort is needed to improve execution-efficiency and speed-up for HPC.

In this contribution, multiple reasons for cloud inefficiencies are presented for the case of

OpenStack [2] as cloud operating system and for OpenFoam [3] as HPC example code. Suggestions are made how to solve or circumvent them. The results can be extrapolated to other cloud OSes and other HPC codes. The contribution is organized as follows: In chapter 2, the state-of-the-art is reviewed. Chapter 3 describes our project and what equipment and tools we were using. In chapter 4, the conducted measurements and findings are presented and discussed. The paper ends with a conclusion, followed by an outlook and a reference list.

State-of-the-Art

Several HPC-related scientific projects dedicated to cloud optimization and scheduling were studied by us, and a thorough literature search was conducted to determine which characteristics of the cloud should be improved to make it HPC-capable. Also the authors of this paper have worked previously on cloud performance issues in [12]. The papers, that are reviewed here are [4]-[11] (in rank of importance as we felt it). In [4] and [5], it is stated that the virtualization overhead, the poor interconnect and communication performance clouds usually have, the HPC-agnostic cloud schedulers, and the cloud’s heterogeneity and multi-tenancy, which creates cross-application interference, are some but not all bottlenecks that prevent from HPC in clouds.

In order to address these challenges, the ‘Nova’ scheduler of OpenStack was made HPC-aware by the authors of [4] and [5], which resulted in performance improvements of up to 45%. This

Bedeutung). In [4] und [5] wird festgestellt, dass der Verwaltungszusatzaufwand aufgrund der Virtualisierung, die schlechte Leistungsfähigkeit beim Verbindungsnetzwerk und bei der Kommunikation, die Clouds üblicherweise haben, die HPC-Ignoranz der Cloud Scheduler, die Heterogenität und die gleichzeitige Vielfach-Vermietung derselben Cloud-Ressourcen diejenigen Engpässe sind, die HPC in Clouds verhindern. Um dies anzugehen, haben die Autoren von [4] und [5] den „Nova“ Scheduler von OpenStack mit dem Wissen über die HPC-Anwendungen „angereichert“, was in einer Leistungssteigerung von 45% resultierte. Das ist ein bemerkenswertes Ergebnis. In Artikel [6] wird die Idee einer verbesserten Platzierung virtueller Maschinen (VMs) durch den Nova-Scheduler vertieft. Die Autoren haben Nova modifiziert, um ihn über die zu Grunde liegende Cloud Hardware, die Topologie des Verbindungsnetzwerkes, die Gruppierung von Cloud-Ressourcen und über Störungen zwischen Jobs in Form von sog. „lauten Nachbarn“ informieren zu können. In [7] wird eine Übersicht über Scheduling-Verfahren gegeben, die auch für Clouds relevant ist. In [8] werden Inter-Cloud Meta-Scheduler diskutiert, die die Systemdynamik, die Interoperabilität und die Heterogenität verschiedener Clouds berücksichtigen. Die Absicht dabei ist, die Eigenschaften eines gegebenen HPC-Codes herauszufinden und daraus ein Modell abzuleiten, das die Ressourcenanforderungen dieses HPC-Jobs in sog. kooperativen E-Science-Infrastrukturen widerspiegelt. Artikel [9] berichtet über ein verteiltes Job-Managementsystem, das Millionen kleiner HPC-Jobs verwalten kann. Dieses System zielt ab auf die großen kommerziellen CSPs, wie z.B. auf Amazon. Der Fokus liegt dabei auf hohem Durchsatz und guter Auslastung, was genau das ist, was CSPs benötigen. In [10] wird die Bezeichnung „HPC-as-a-Service“ als ein neuer CSP-Dienst verwendet. Das dort beschriebene Projekt versucht, die Lücke zu schließen zwischen dem, was ein CSP in jedem Moment seinen Kunden als Rechenressourcen zu einem bestimmten Preis anbieten kann, und dem, was die Kunden in diesem Augenblick haben wollen. Es wird erklärt, dass beide Seiten (CSP und Kunden) große Streubreiten und Inhomogenitäten aufweisen. Es wird weiterhin aufgezeigt, dass eine multi-kriterielle Optimierung der Cloud-Ressourcen erforderlich ist. Die Autoren verwenden dafür etwas, was sie als „gemischte Ganzzahl-Linearanprogrammierung“ bezeichnen, sowie ein sto-

chastisches Optimierungsmodell für eine effiziente Verwendung gemeinsamer HPC-Ressourcen zur Bereitstellung von Diensten. Ihr Fokus liegt auf dem Kosten/Nutzen-Verhältnis von Cloud-Ressourcen.

In [11] wird berichtet, dass traditionelle Hypervisoren nicht gleichermaßen gut die Last verschiedener Jobs handhaben können, die entweder eine hohe oder eine niedrige Rechenleistung benötigen. Der Grund dafür liegt darin, dass die Inter-VM-Kommunikation im selben HPC-Job dadurch herabgesetzt wird, dass kommunizierende VMs im Moment der Kommunikation descheduled werden können. Die Autoren schlagen vor, nicht eine ganze Cloud zu schedulen, sondern das Scheduling nur innerhalb von isolierten Teilmengen von Cloud-Ressourcen durchzuführen, um die gegenseitige Beeinflussung von Jobs zu begrenzen. Sie verwenden dazu ein Vorhersagemodell samt Software-Implementierung desselben, um eine Prognose darüber abzugeben, welche VM mit welcher anderen in Zukunft kommunizieren wird, um ein descheduling zum falschen Zeitpunkt zu vermeiden. Außerdem verschieben sie kommunikationsintensive Gruppen von VMs auf andere Teile der Cloud-Ressourcen, um solche HPC-Codes effizienter in der Cloud zu machen. Das wird durch einen Scheduler bewerkstelligt, der Kenntnis von den IO-Aktivitäten, d.h. von den Kommunikationsbeziehungen der VMs untereinander hat.

Die Artikel, die bislang rezensiert wurden, haben substantiell zu HPC-effizienteren Clouds beigetragen. Allerdings, haben nicht wenige Artikel keine reale Cloud sondern einen Simulator benutzt, oder sie haben keine echten HPC-Codes eingesetzt, sondern synthetische Lastgeneratoren. Aus unserer Sicht war es deshalb nicht immer klar, wie realistisch die erzielten Ergebnisse sind. Deshalb haben wir die Richtung verfolgt, echte Hardware einzusetzen und ein weit verbreitetes HPC-Programmpaket zu benutzen, um damit Messungen auf einer Standard-Cloud durchzuführen, in der Hoffnung, dadurch zuverlässigere und realistischere Ergebnisse zu erzielen.

Projektbeschreibung

Für das Projekt haben wir eine eigene Cloud aufgebaut, OpenStack [2] installiert und verwen-

is a remarkable result. In [6], the idea of better placement of virtual machines (VMs) by an improved Nova scheduler is explained in more detail. The authors modified Nova to make it know about the cloud's underlying hardware, about the interconnect topology, the grouping of resources and about inter-job interferences ('noisy neighbours'). In [7], a monograph on scheduling approaches is given, which is also relevant for clouds. In [8], inter-cloud meta-schedulers are discussed, that consider cloud system dynamics, interoperability and heterogeneity issues. The intention is to elicit the characteristics of a given HPC code and to produce from that information a model that reflects the resource requirements of that HPC job in so-called 'cooperative e-science infrastructures'. [9] reports about a distributed job management system that can support millions of small HPC jobs. This system aims to the big commercial cloud service providers such as Amazon, for example. The focus lies on high throughput and good utilization, which is exactly what CSPs need. In [10], the term 'HPC-as-a-Service' is used as a new offer from CSPs. The project tries to bridge the gap between what a CSP can offer as compute resources to his clients in a specific moment and for a specific price and what clients want to have in that moment. It is explained that both sides (clients and CSPs) exhibit big variances and heterogeneities. It is furthermore pinpointed that a multi-criteria optimization of cloud resources is needed. The authors used for that purpose 'mixed integer linear-programming' and a 'stochastic optimization model for efficient HPC resource sharing' for their service provisioning. Their focus lies on the cost-benefit of cloud resources. In [11], it is reported that traditional Hypervisors cannot handle adequately heterogeneous workloads from high and low performance compute jobs at the same time. The reasons for that is that inter-VM communication inside of the same HPC job is degraded by the fact that VMs are descheduled in the very moment of their communication. The authors suggest to schedule HPC jobs not on a cloud-wide basis, but only inside isolated subsets of cloud resources to limit inter-job interferences. The authors are using a predictor model and a software implementation for a prognosis which VM will communicate which other to avoid descheduling at the wrong point in time. Furthermore, they migrate a communication

intensive group of VMs to another resource subset to make IO-dominant HPC more effective. This is accomplished by a scheduler that is aware of the IO activities of VMs, i.e. of their ongoing communication relationships.

The papers, which we reviewed so far, have contributed significantly to HPC-efficient clouds. However, not few papers were using not a real cloud but some simulators, or they have not used real HPC codes but synthetic load generators. From our point of view, it was not always clear how realistic the achieved results are. Therefore, we followed the path of real hardware executing a widely-used HPC package, and to make with this package real measurements on a standard cloud in the hope to achieve more realistic results.

Project Description

For the project, we built an own cloud, installed OpenStack [2] and used OpenFoam [3] as HPC benchmark, which is based on the MPI parallelization standard [13]. Furthermore, shell scripts were written to automate the OpenFoam benchmarks by running them with various parameters and set-ups.

Easier Possibilities

However, before we started to establish an own cloud, we have investigated the subsequent easier possibilities: 1.) installing a cloud on a set of VMs (nested virtualization), 2.) using a cloud simulator, 3.) doing all measurements in a commercial cloud or on a University cloud in a computing centre. All three options were evaluated, and it is explained in the following why we dismissed them.

Nested Virtualization

We found out by our measurements, that already a single virtualization that is not nested, decreases HPC speedup and efficiency significantly, unless the latest available hardware accelerators for virtualized computation and communication are engaged. Literature studies revealed also that nested virtualization is never practiced for HPC.

den OpenFoam [3] als Benchmark. Außerdem wurden Skripte geschrieben, um OpenFoam in diversen Parameter-Konfigurationen automatisiert auszuführen.

Einfachere Möglichkeiten

Bevor wir mit dem Bau der eigenen Cloud begannen, haben wir die nachfolgenden einfacheren Möglichkeiten untersucht: 1.) Installation des Cloud OS auf virtuellen Maschinen anstelle von realer Hardware (sog. doppelte Virtualisierung), 2.) Verwendung eines Cloud-Simulators oder 3.) Durchführung der Messungen auf einer bestehenden Universitäts- oder kommerziellen Cloud. Alle drei Möglichkeiten wurden evaluiert. Im folgenden wird erläutert, warum wir keine davon verwendet haben.

Doppelte Virtualisierung

Wir haben durch Messungen ermittelt, dass bereits eine einfache Virtualisierung, die nicht geschachtelt ist, die Beschleunigung und Effizienz von HPC-Programmen erheblich schmälert, außer wenn man die neuesten Hardware-Beschleuniger für virtualisiertes Rechnen und virtuelle Kommunikation verwendet. Literaturstudien zeigen außerdem, dass doppelte Virtualisierung im HPC-Bereich nicht praktiziert wird.

Cloud-Simulatoren

Wir begannen intensiv CloudSim [14] aus der Menge leicht erhältlicher Open-Source Cloud-Simulatoren zu analysieren. Die Alternativen, die wir davor betrachtet und wieder verworfen hatten, waren GreenCloud [15], iCanCloud [16] und eine verbesserte Version des sog. MaGateSim-Simulators, die in [18] beschrieben wird. GreenCloud und MaGateSim sind für Energie-sparendes Cloud Computing gemacht, was nicht in unserem Fokus lag. Sie sind auch zu begrenzt für unsere Leistungsanalysen. iCanCloud ist hilfreich, um den Zielkonflikt zwischen Leistungsvermögen einer Cloud und dazu notwendige Kosten zu bewerten, was für uns ebenfalls nicht relevant war. Ein genauerer Blick auf CloudSim zeigte uns

allerdings, dass es nur eine kleinere Zahl von Java-Klassen gibt, die auch keine Inter-Prozessor- oder Inter-Server-Kommunikation unterstützen. CloudSim ist zu eingeschränkt für das, was wir brauchen. Zusätzlich fanden wir den Artikel [17], in dem festgestellt wird, dass die Ergebnisse von CloudSim nicht realistisch seien. Deswegen haben die Autoren von [17] einen Nachfolger mit dem Namen „NetworkCloudSim“ entwickelt, der Kommunikation auf Basis eines einfachen Bandbreite/Latenz-Modells unterstützt. Daraus schlossen wir, dass von allen Simulatoren höchstens NetworkCloudSim für uns relevant sein könnte.

Grundsätzlich ist es auch möglich von NetworkCloudSim zu profitieren, indem man Modelle von Cloud-Anwendungen damit simulativ exploriert. Diese Modelle werden typischerweise dadurch definiert, dass man Job-Ausführungszeiten und die Kommunikation zwischen den Prozessen eines parallelen Jobs vorab abschätzt. Allerdings konnten wir die Behauptung der Autoren, dass NetworkCloudSim eine präzise Bewertung von Scheduling-Verfahren in wissenschaftlichen und MPI-basierten Anwendungen erlaubt -einschließlich der Kommunikationsinfrastrukturen in Rechenzentren- nicht nachvollziehen. Das Problem mit dieser Behauptung war, dass die die Autoren keinerlei Zahlen oder Beispiele von echten Messungen geliefert haben, um damit die vielen Parameter von NetworkCloudSim zu eichen. Außerdem kam hinzu, dass Hardware-Beschleuniger, die z.B. in [19] beschrieben sind, wie die Single Root IO-Virtualization (SR-IOV) [19], [20], sich für eine effiziente virtualisierte Kommunikation inkl. des Rechnens als unverzichtbar herausgestellt haben. Unglücklicherweise sind solche Beschleuniger nicht in NetworkCloudSim enthalten. Daraus schlossen wir, dass es auf diese Weise nicht möglich ist, die Kommunikationsstruktur einer HPC-Anwendung oder eines Rechenzentrums zu modellieren, weil NetworkCloudSim keinerlei Modell für die Inter-VM-Kommunikation bei OpenStack oder bei KVM enthält. Unsere diesbezügliche Literaturstudie zeigte außerdem, dass bisher niemand versucht hat, OpenStack, KVM oder SR-IOV in NetworkCloudSim zu simulieren. Das bedeutete für uns, dass es nicht möglich ist, realistische Werte für die Konfigurationsparameter von NetworkCloudSim zu ermitteln.

Cloud Simulators

From the set of easily available open-source cloud-simulators, we started with a closer look to CloudSim [14]. The alternatives we have considered before and dismissed were GreenCloud [15], iCanCloud [16] and an improved version to the MaGateSim simulator, which is described in [18]. GreenCloud and MaGateSim are for energy-saving cloud-computing, which is not in our focus. They are too limited for our performance analyses. iCanCloud helps to predict the trade-off between cost and performance, which is also not relevant for us. A closer look into CloudSim revealed, however, that it contains merely a smaller set of Java classes that do not include inter-processor or inter-server communication. CloudSim alone is too restricted for what we need. Additionally, we found the paper [17], which stated that the results from CloudSim are not realistic. Because of that, the authors of [17] created a successor called 'NetworkCloudSim', which includes communication in a simple bandwidth/latency model. We concluded, from all network simulators only NetworkCloudSim might have relevance for us.

It is in principle possible to profit from NetworkCloudSim by a simulative exploration of models of cloud applications. These models are typically defined in terms of estimated job duration and communication between a job's parallel tasks. However, the claim of the authors that NetworkCloudSim allows for precise evaluation of scheduling algorithms in scienti□c, MPI-based applications, including the modelling of a data centre's interconnection network could not be verified by us. The problem with this claim was that the authors did not provide any figures or examples from real measurements to gauge NetworkCloudSim parameters. Furthermore, hardware accelerators, which are described in [19], such as Single Root IO-Virtualization (SR-IOV) [19], [20] for example, are nowadays indispensable for an efficient virtualized computation and communication. Unfortunately, they are not contained in NetworkCloudSim. We concluded from that, that it is not possible to model the interconnect-performance of an HPC application or a data centre, because NetworkCloudSim does not contain a model for OpenStack's and KVM's inter-VM communication

via such hardware accelerators. Finally, due to our literature search, nobody has modelled, so far, OpenStack, KVM or SR-IOV in NetworkCloudSim. This meant for us that it is not possible to obtain realistic values for the set-up parameters NetworkCloudSim needs.

Existing Commercial or University Clouds
Quickly, we learned that it is not possible to change on-the-fly the interconnect structures or to add contemporary hardware accelerators cards to a commercial or University cloud, because this disturbs their productive operation.

Vote for Own Real Cloud

We decided that all three options discussed above, although simple at first sight, are not viable. We expected either results, which may be not realistic enough in case of simulations, or poor performance and non-deterministic behaviour in case of nested virtualization, or we learned they are not feasible. Because of that, performance estimations by simulations were disregarded by us for reasons of quality and validity, as well as execution-time guesses by means of a cloud that runs on a set of VMs. Because of that, we decided to build an own cloud.

Our Project Cloud

Our cloud consists of 17 used servers from Dell and Sun with a total number of 76 Cores, 292 GB RAM and 19 TB as Disk Storage. The servers are coupled by 17 Infiniband network interface cards of 40 Gbit/s each and a 40 Gbit/s Infiniband switch. The interfaces are of type Mellanox MHQH19B-XTR and are using QSFP copper cables, as well as the switch itself, which is of type Mellanox Infiniscale IS5023. The switch has 18 ports and an ultra-low port-to-port latency of 100 ns only. In parallel to that high-speed network, a standard communication system was installed, that comprises 17 Ethernet cards of 1 Gbits/s each and a 24 port 1 Gbits/s Ethernet switch to allow for performance comparisons between the two couplings. The host OS for the cloud is Ubuntu Server 12.04 LTS with the OpenStack 'Grizzly' release. The available Guest OSes are

Existierende kommerzielle oder universitäre Clouds

Wir lernten schnell, dass es bei kommerziellen oder universitären Clouds nicht möglich ist, im laufenden Betrieb, schnell die Verbindungsstrukturen zu ändern oder aktuelle Hardware-Beschleunigerkarten hinzuzufügen, weil solche Aktionen die produktiven Abläufe stören.

Entscheidung für eine eigene Cloud

Wir entschieden, dass alle drei zuvor diskutierten Optionen nicht gangbar sind, obwohl sie auf den ersten Blick einfach erschienen waren. Entweder sind die zu erwartenden Resultate im Falle von Simulationen nicht realistisch genug, oder sie weisen wie im Falle der doppelten Virtualisierung zu schlechte Leistungsdaten sowie Indeterminismus auf, oder sie sind schlicht undurchführbar. Deswegen beschlossen wir, eine eigene Cloud aufzubauen.

Unsere Projekt-Cloud

Unsere Cloud umfasst 17 gebrauchte Server von Dell und Sun mit insgesamt 76 Cores, 292 GB RAM und 19 TB Plattenspeicher. Die Server sind über 17 Infiniband Netzwerkkarten von je 40 Gbit/s und einem 40 Gbit/s Infiniband Switch gekoppelt. Die Karten sind vom Typ Mellanox MHQH19B-XTR und benutzen QSFP-Kupferkabel, genau wie der Switch, der vom Typ Mellanox Infiniscale IS5023 ist. Der Switch hat 18 Ports mit einer sehr niedrigen Port-zu-Port-Latenz von nur 100 ns. Parallel zu diesem Hochgeschwindigkeitsnetzwerk wurde ein Standardkommunikationssystem installiert, das 17 Ethernetkarten zu je 1 Gb/s und einen Ethernet Switch mit 24 Ports von 1Gb/s umfasst, um Leistungsvergleiche durchführen zu können. Das Host-Betriebssystem für die Cloud ist Ubuntu Server 12.04 LTS mit der OpenStack 'Grizzly'-Version. Die verfügbaren Gast-Betriebssysteme sind Scientific Linux 6.5, Ubuntu 14.04.1 Server LTS, und Ubuntu 12.04 Server LTS, alle für 64 Bit.

OpenFoam

OpenFoam ist ein paralleler HPC-Code, der primär dafür entwickelt wurde, die Laplace -und Navier-Stokes-Gleichungen numerisch zu lösen, d.h. er dient der Berechnung von laminaren und turbulenten Strömungen von komprimierbaren und unkomprimierbaren Fluiden (Gasen und Flüssigkeiten). Er hat zusätzliche Gleichungslöser für allgemeine Teilchenflüsse, Verbrennungsvorgänge, Moleküldynamik, Wärmeleitung, für elektromagnetische Probleme, starre elastische Körper und andere Zwecke. Die parallele Ausführung von OpenFoam basiert auf Open MPI. Bevor wir mit OpenFoam begannen, haben wir eine Befragung unter Benutzern von OpenFoam durchgeführt, um heraus zu finden, was sie genau machen und welche Erwartungen sie an die Ausführung in einer Cloud hegen. Anhand dieser Befragung, haben wir verstanden, was die Probleme der Anwender sind, und wo unser Ausgangspunkt ist. Anhand der Messungen, die nachfolgend präsentiert werden, konnten wir auch die Meinung der Benutzer bestätigen, dass OpenFoam und OpenStack, bis jetzt jedenfalls, keine gute Kombination sind.

Leistungstests

Anfangs konfigurierten wir OpenFoam so, dass das „Dambruch“-Beispiel ausgeführt wurde, das dem Release 2.2.1 beigefügt ist, weil dieses Beispiel gut dokumentiert ist. Darin gibt es 7700 Gitterpunkte für geometrische Objekte in drei Dimensionen. Eine Sekunde in der Realität wird über 1000 Zeitschritte simuliert. Anschließend haben wir die ursprüngliche Konfiguration modifiziert, um fortgeschrittenere Tests durchzuführen. Die Messungen wurden 50 Mal wiederholt, und der erste Durchlauf wurde gelöscht, um Einschwingvorgänge auszuschließen.

Messergebnisse

Die Programmausführungszeiten aller Testdurchläufe sind in Tabelle 1 gezeigt. Diese Ergebnisse

Scientific Linux 6.5, Ubuntu 14.04.1 Server LTS, and Ubuntu 12.04 Server LTS, all for 64 bits.

OpenFoam

OpenFoam is a parallel HPC code, which was primarily created for the numeric solution of Laplace and Navier-Stokes equations, i.e. for the calculation of laminar and turbulent flows of compressible and incompressible fluids (gases and liquids). It has additional solvers for general particle flows, for combustion, molecular dynamics, heat transfer, electromagnetic problems, solid elastic bodies, and other purposes. The parallel execution of OpenFoam is based on Open MPI. Before we started with OpenFoam, we performed a questionnaire by asking users of OpenFoam what they are exactly doing and what their expectations are when executing the code on a cloud. From that questionnaire, we understood what the concerns of the users are and what our starting point is. With the measurements presented subsequently, we could also confirm the users' opinion that OpenFoam and OpenStack is, so far, no good match.

Performance Tests

Initially, we configured OpenFoam to execute the 'dam-brake' example that comes with the 2.2.1 distribution, because it is well documented. In the example, there are 7700 grid points for geometric objects in three dimensions. One second in reality was simulated by 1000 time steps. After that, we modified the initial configuration to make more advanced tests. The measurements were repeated 50 times, and the first run was deleted in order to exclude transient effects.

Measurement Results

The execution time results of all test set-ups are shown in table 1. The results were post processed by calculating the speed-up and the efficiency. These two metrics are defined by:

Def. 1: Speed-up S is the ratio of the execution times of a sequential code before and after virtualization and parallelization.

Def. 2: Efficiency E is the utilization of n cores and defined by: E=S/n.

Tabelle 1: Messergebnisse für die Set-Ups 1-7.
Table 1: Performance results for the set-ups 1-7.

Set-Up	Wall-Time [s]	Speed Up	Efficiency [%]
1a: 1 core, bare metal	144	1	100
1b: 4 cores, 2 CPUs, 1 server, bare metal	46	3.1	78
2a: 1 core, 1 KVM	180	0.8	80
2b: 4 cores, 2 CPUs, 1 server, 1 KVM	62	2.3	58
3: nested virtualization, 1 core, 2 KVMs	-	-	-
4a: 1 core, 1 KVM, OpenStack	154	0.94	94
4b: 4 cores, 2 CPUs, 1 server, 4 KVMs, OpenStack	60	2.4	60
5: 4 cores, 4 CPUs, 4 servers, 4 Ethernets, 4 KVMs, OpenStack	320	0.45	11
6: 16 cores, 4 CPUs, 4 servers, 4 Ethernets, 16 KVMs, OpenStack,	670	0.21	5
7a: 4 cores, 4 CPUs, 4 SUN servers, 4 Infinibands, 4 KVMs, OpenStack	237	0.61	15
7b: 16 cores, 4 CPUs, 4 SUN servers, 4 Infinibands, 16 KVMs, OpenStack	998	0.14	4

wurden nachbearbeitet, indem die Beschleunigung und die Effizienz berechnet wurden. Beide Maße wurden folgendermaßen definiert:

Def. 1: Die Beschleunigung S ist das Verhältnis der Ausführungszeiten eines sequentiellen Programms vor bzw. nach Virtualisierung und Parallelisierung.

Def. 2: Die Effizienz E ist der Ausnutzungsgrad von n Cores und definiert als: $E=S/n$

Bewertung der Messergebnisse

Anhand von Set-Up 1a wurde eine Ausführungszeit von 144 s als Referenzzeit für alle nachfolgenden Messungen ermittelt. Dieser rel. geringe Wert weist bereits darauf hin, dass die Problemgröße, gemessen an üblichen HPC-Standards, zu klein ist, aber so ist das „Dammbruch“-Beispiel. Anhand von Set-Up 1b kann man sehen, dass eine Parallelisierung von OpenFoam nützlich ist, sofern die Programmausführung im selben Server erfolgt. Allerdings ist die Effizienz bei 4 Cores um 22 Prozentpunkte auf 78% abgefallen. Auf einem Supercomputer oder Parallelrechner sollte OpenFoam laut Manual bis ca. 1000 Cores skalieren. Ein Abfall um 22 Punkte bereits bei 4 Cores ist ein Indiz dafür, dass die Kommunikationszeit in der Cloud im Vergleich zur Rechenzeit nicht vernachlässigt werden kann. Die Messung bestätigt, dass die Problemgröße zu klein für gute Skalierbarkeit in der Cloud ist. Die Messung 2a zeigt, dass eine Virtualisierung einen Effizienzverlust von 20 Prozentpunkten zur Folge hat. Dies kann dadurch begründet werden, dass nur der AMD Vi [19] Hardware-Beschleuniger verwendet wurde, aber nicht das neuere Intel VT-x [19]. Set-Up 2b zeigt, dass die gleichzeitige Anwendung von Virtualisierung und Parallelisierung die Effizienz um 42 Prozentpunkte schmälert, was durch die Addition der Zahlen von 1b und 2a bereits erwartet werden konnte. Set-Up 3 konnte nicht ausgeführt werden, weil die VM, die von KVM innerhalb einer anderen VM gemäß doppelter Virtualisierung, erzeugt wurde, nicht in der Lage war, ihr Gast-Betriebssystem auszuführen. Die Ursache dafür ist unbekannt. Set-Up 4a zeigt dass die Anwesenheit eines Cloud-Betriebssystems einen Effizienzverlust von 40 Punkten für den parallelen Code bewirkt, was für HPC nicht tragbar ist. Dies kann dadurch

erklärt werden, dass bei den Messungen kein Hardware-Beschleuniger für die Kommunikation vorhanden war. Dieser Grund ist, neben anderen, auch bei Set-Up 5 dafür verantwortlich, dass die Effizienz auf nur noch 11% absinkt, sobald der Code auf 4 Cores verteilt wird, die in 4 Servern untergebracht sind. Ein zweiter Grund ist, dass bei der Messung nur das 1 Gbit/s-Ethernet verwendet wurde, was für HPC-Parallelrechner viel zu langsam ist. Als dritten Grund haben wir den Cloud Overhead identifizieren können, der durch die Verwendung von zwei IP-Headern, sowie eines weiteren Headers für das sog. GRE-Protokoll [22] entsteht. Für den Einsatz von GRE ist der „Neutron“-Netzwerkdienst von OpenStack verantwortlich, für die doppelten IP Header (einer für das Host-Betriebssystem und einer für das Gast-Betriebssystem) sind der Hypervisor und das Cloud OS verantwortlich. Der vierte Grund ist nach unseren Erkenntnissen der, dass Neutron nicht L2 Switching sondern L3 Routing einsetzt, um zwischen Servern im selben Gehäuse Daten austauschen. Alle vier Gründe zusammen sind für den geschilderten Effizienzeinbruch auf 11% bei nur 4 Cores auf 4 Servern verantwortlich. Dies bedeutet, dass OpenFoam auf diese Weise nicht skalierbar ist. In Set-Up 6 verschärft sich die Situation auf 5% Effizienz, sobald der Code auf 16 Cores in 4 Servern verteilt wird. Zusätzlich zu den bereits in Set-Up 5 erwähnten 4 Gründen haben wir hier noch einen fünften Grund für diesen Abfall auf 5% identifizieren können: berechnete Zwischenergebnisse müssen in jedem Server auf eine einzige Netzwerkkarte (NIC) gemultiplext werden, um als Eingabe für andere Server zu dienen. Das Multiplexen ist jedoch ein Engpass, weil es vom Hypervisor vorgenommen wird, weil kein Hardware Multiplexer wie z.B. SR-IOV [19], [21] verfügbar war.

Die größte Überraschung war allerdings, als wir in den Set-Ups 7a und 7b auf ein Kommunikationssystem umstellten, das die 40-fache Bandbreite von Ethernet hat und weniger als 10% seiner Latenz. Trotz dieser überlegenen Zahlen bei Bandbreite und Latenz konnte bei der Messung 7a keine bedeutende Verbesserung im Vergleich zu Set-Up 5 gemessen werden, was ein bemerkenswertes Resultat ist. Es bedeutet, dass es in einer Cloud nicht ausreicht, das bestehende Standard-Kommunikationssystem durch eine Hochtechno-

Evaluation of Measurement Results

According to set-up 1a, the reference execution time for all subsequent measurements could be determined as 144 s. This value indicates that the problem size compared to usual HPC standards is too small. From set-up 1b, it can be seen that an OpenFoam parallelization is beneficial if the code execution takes place in the same server. However, efficiency already drops by 22 percentage points to 78% with 4 cores only. On a supercomputer or a parallel computer, OpenFoam should scale well until about 1000 cores, according to its manual. A drop of 22 points already at 4 cores is a sign that the communication time cannot be neglected in the cloud compared to the computation time. It confirms also that the problem size is too small for good scalability. From set-up 2a, it can be seen that virtualization causes the efficiency to drop by 20 percentage points. This can be explained by the fact that only AMD Vi [19] is used as hardware accelerator, but not the newer Intel VT-x [19]. Set-up 2b shows that the simultaneous application of virtualization and parallelization reduces efficiency by 42 percentage points, which could be expected already by adding the figures 1b and 2a. Set-up 3 was not possible to conduct, because the VM that was created by KVM inside of another VM -according to nested virtualization- was not able to run its guest OS. The reason for this is unknown.

Set-up 4a shows that the cloud OS incurs an overhead such that the efficiency drops by 6 percentage points which is low. However, Set-up 4b shows that the cloud OS causes an efficiency drop by 40 percentage points for the parallel code, which is not tolerable for HPC. This is explained by the fact that no hardware accelerator for virtualized communication was engaged by us. In Set-up 5, this reason is also responsible, among others, for the drastic efficiency drop of down to 11%, in case of code distribution over 4 cores residing on 4 servers. A second reason for the drastic drop is the fact that 1 Gbit/s Ethernet was used, which is never the case in parallel computers for HPC, because it is too slow. In addition to that, we have found as third reason the overheads of two IP headers and one additional GRE [22] header. The GRE header is introduced by the 'Neutron' network service of OpenStack. The

two IP Headers (one for the host OS and one for the Guest OS) are caused by the Hypervisor and the Cloud OS. A fourth reason according to our findings is that Neutron does not use L2 switching but L3 routing to exchange data between servers in the same rack. All four reasons together are responsible for the described efficiency drop of down to 11% with 4 cores only, which means that OpenFoam is not scalable this way. In set-up 6, the situation escalates even to 5% efficiency, when the code is executed in parallel on 16 cores from 4 servers. In addition to the four reasons mentioned in set-up 5, a fifth reason could be identified by us for the drop down to 5%: intermediate results must be multiplexed onto the same physical Ethernet network interface card (NIC) in order to be useful as input for other servers. However, multiplexing is a bottleneck, because it is accomplished by the Hypervisor in software, if no hardware multiplexer such as SR-IOV [19], [20] is available, which was true for our case.

The biggest surprise, however, created set-ups 7a and 7b when we moved to a communication system that has 40 times the bandwidth of Ethernet and less than 10% of its latency. Despite this superior figures for bandwidth and latency, no significant improvement could be measured in set-up 7a compared to set-up 5, which is a remarkable result. This means that it is not sufficient for a cloud to replace its existing standard communication system by a high performance technology, in order to achieve high performance computing. Obviously, several other factors must be improved as well. One of them is to allow Open MPI to access the Infiniband cards directly and not via a virtual Ethernet NIC (so-called virtual Ethernet over Infiniband). Set-up 7b confirms this insight, because the efficiency is with 4% similar bad as in set-up 6.

Summary of Findings and Discussion

Our findings are that parallelization can compensate, at least in OpenFoam, the efficiency losses, caused by virtualization and by the cloud OS as long as the code is executed on the same server. However, if the code is distributed onto multiple servers, efficiency drops to a value which cannot be compensated any more by the engaged cores.

logie zu ersetzen, um dadurch Hochleistungsrechnen zu erzielen. Offenbar müssen mehrere andere Faktoren ebenfalls verbessert werden. Einer dieser Faktoren ist, Open MPI den Direktzugriff auf die Infinibandkarten zu erlauben, und nicht über eine virtuelle Ethernet NIC zu gehen (sog. Ethernet über Infiniband). Set-Up 7b bestätigt diese Erkenntnis, weil die dortigen 4% ähnlich schlecht sind wie die 6% von Set-Up 6.

Zusammenfassung der Erkenntnisse und Diskussion

Unsere Ergebnisse zeigen, dass eine Code-Parallelisierung, zumindest im Falle von Open-Foam, die Verluste kompensieren konnte, die die Virtualisierung und das Cloud-Betriebssystem verursachen, sofern der Code auf demselben Server ausgeführt wird. Wenn allerdings der Code auf mehrere Server verteilt wird, dann war es den beteiligten Cores nicht mehr möglich den Effizienzverlust auszugleichen. Die Gründe dafür sind:

- Es wurde bei den Messungen kein Hardware-Beschleuniger für die virtualisierte Kommunikation benutzt
- Das Multiplexen von Datenpaketen wurde vom Hypervisor vorgenommen und nicht von einer NIC, wie es bei SR-IOV NIC der Fall gewesen wäre
- Ein Direktzugriff der HPC-Anwendung auf den Infiniband-Gerätetreiber war nicht möglich, so wie es bei Intel VT-d [21] gemacht wird. Stattdessen wurde die Datenübertragung mittels virtuellem Ethernet über Infiniband durchgeführt.
- L3 Routing anstelle von L2 Switching wurde verwendet, mit 2 IP Headern und einem Header für das GRE-Protokoll als Overhead

Die beste Beschleunigung wurde für die Fälle erzielt, bei denen Cores innerhalb derselben CPU oder mindestens desselben Servers Daten austauschen. Dies wird dadurch erklärt, dass Cores in derselben CPU über einen gemeinsamen on-chip Level 1 oder 2 Cache verbunden sind und nicht über Netzwerkkarten, was viel schneller ist. Cores

im selben Server, die auf verschiedenen CPUs verteilt sind, sind immerhin noch über den gemeinsamen Hauptspeicher verbunden, der off-chip ist. Weil Open MPI immer die schnellste verfügbare Bittransferschicht benützt, schaltet es automatisch um von IP über NICs, wie im verteilten Fall, nach Kommunikation über gemeinsamen Speicher, wie im Fall der Einzel-CPU oder des Einzel-Servers. Die letzte Erkenntnis war, dass ein Ersetzen eines Standard 1 Gbit/s-Ethernets durch ein schnelles 40 Gbit/s Infiniband nicht ausreichend ist, um eine übliche Cloud in ein HPC-fähiges System zu verwandeln.

Schlussfolgerungen

Durch die Leistungstests, die bislang von uns durchgeführt wurden, und anhand unserer Literaturstudien konnte gezeigt werden, dass Standard-Clouds nicht für HPC geeignet sind. Die Gründe dafür sind die schlechte Leistungsfähigkeit, die das Verbindungsnetzwerk und die Ein-/Ausgabe haben, die bestehenden HPC-ignorierenden Cloud-Scheduler, die intrinsische Heterogenität von Cloud-Hardware und Anwender-Software und das gleichzeitige Vermieten ein- und derselben Cloud-Ressourcen durch den CSP, das sog. „laute Nachbarn“ verursacht. Wir fanden außerdem heraus, dass aktuelle Hardware-Beschleuniger zur virtualisierten Kommunikation für HPC unverzichtbar sind, weil sonst das Multiplexen von Datenpaketen vom Hypervisor in Software vollzogen wird und nicht durch die Netzwerkkarte selbst. Außerdem haben ohne SR-IOV- oder Intel VT-d-Beschleuniger weder das Cloud- noch das Gast-Betriebssystem effizienten Zugang zu den Infiniband-Gerätetreibern. Dann werden Datenübertragungen auf langsame Weise mittels virtuellem Ethernet über Infiniband abgewickelt. Außerdem wird Layer 3 Routing anstelle des schnelleren Layer 2 Switchings verwendet, zusammen mit zwei IP Headern und einem GRE Header. In Folge dessen ist selbst ein Infiniband-Netzwerk mit hoher Bandbreite und geringer Latenz nicht viel schneller als Standard-Ethernet. Schließlich sollte auch Open MPI wegen seiner umfassenden und automatischen Unterstützung aller Arten von Kommunikation verwendet werden. Mit dieser Liste haben wir die meisten, aber nicht alle Engpässe aufgelistet, die eine effiziente Ausführung von HPC-Codes in Clouds bislang verhindern.

The reasons are:

- no hardware accelerators for virtualized communication were used
- data packet multiplexing was done by the Hypervisor in software and not by the network interface card in hardware, as SR-IOV does, for example
- no direct access for the HPC application to the Infiniband device drivers was possible, as it is the case with Intel VT-d [19], [21]. Instead data transmission took place via virtual Ethernet over Infiniband.
- layer 3 routing instead of layer 2 switching was engaged, together with two IP headers and one header from the additional GRE protocol [22] of Neutron

The best speedup is obtained for the cases that cores of the same CPU or the same server are communicating with each other. This is explained by the fact that cores in the same CPU are coupled via on-chip level 1 or level 2 caches and not via network interface cards, which is much faster. Cores in the same server, which reside on different CPUs, are coupled via off-chip shared-memory, which is still better than a NIC coupling. Since Open MPI uses always the fastest bit transfer layer available, it switches automatically from IP via NICs as in the distributed case to shared memory communication as in the single CPU or single server case. Finally, a simple replacement of a 1 Gbit/s Ethernet by a 40 Gbit/s Infiniband is also not sufficient to turn a standard cloud into a system that is HPC capable.

Conclusions

By the performance tests conducted so far by us and by literature studies it could be demonstrated that standard clouds are not suited for HPC. The reasons are the poor interconnect and I/O performance a standard cloud has, the existing HPC-ignoring cloud schedulers, the inherent heterogeneity of cloud hardware and user software, and the multi-tenancy applied by the CSPs, which creates „noisy neighbours“. We found out

that contemporary hardware accelerators for virtualized communication are indispensable for HPC efficiency, because otherwise data-packet multiplexing is accomplished in software by the Hypervisor and not by the network interface card itself. Additionally, without a SR-IOV or Intel VT-d accelerator neither the cloud OS nor the guest OS have an efficient access to the Infiniband device drivers, and data transmission is accomplished slowly via virtual Ethernet over Infiniband. Additionally, layer 3 routing instead of layer 2 switching is employed by the Hypervisor, together with two IP headers and one additional GRE header. As a consequence, even high-speed low-latency Infiniband is not much faster than standard Ethernet. Finally, Open MPI should be used, because of its comprehensive and automatic support of all kinds of inter process communication. With this list, we enumerated most but not all of the bottlenecks that hamper efficient execution of HPC codes in a cloud.

Outlook

Our working hypothesis is that it is possible to convert any standard cloud into a parallel system that allows for HPC, provided that a set of improvements are done in the cloud. The suggested improvements are:

1. replacement of Standard Ethernet for inter-server communication by fast Infiniband of at least 10 Gbit/s (better 40 Gbit/s) that is driven by 8-lane PCIe interfaces as minimum
2. replacement of layer 3 routing between VMs by layer 2 switching to avoid the overhead of double IP packets in host OS and guest OS and to avoid the GRE tunnel overhead between guest OS and host OS
3. replacement of virtual Ethernet over Infiniband by allowing a direct access for Open MPI to the Infiniband device drivers via Intel VT-d
4. engagement of a suited combination of hardware accelerators for virtualized computation and communication, that are localized in the CPUs, in the PCIe peripherals and on the server boards to minimize Hypervisor interceptions
5. configuration and activation of a proper set of BIOS-, Hypervisor- and Cloud OS options

Ausblick

Es ist unsere Arbeitshypothese, dass es möglich ist, jede Standard-Cloud in einen Parallelrechner für HPC zu verwandeln, sofern in der Cloud bestimmte Verbesserungen durchgeführt werden. Diese Verbesserungen sind:

1. Ersatz von Standard-Ethernet für die Inter-Server-Kommunikation durch ein schnelles Infinibandnetzwerk von mindesten 10 Gbit/s (besser 40 Gbit/s), das mindestens von achtfach PCIe-Schnittstellen gespeist wird
2. Ersatz von Layer 3-Routing durch Layer 2-Switching um den Zusatzaufwand von 2 IP-Paketen und einem GRE-Tunnel zwischen Guest OS und Host OS zu vermeiden.
3. Ersatz von virtuellem Ethernet über Infiniband durch einen Direktzugriff von Open MPI auf die Infiniband-Gerätetreiber so wie bei Intel VT-d
4. Einsatz einer geeigneten Kombination von Hardware-basierten Beschleunigern für virtualisiertes Rechnen und für virtualisierter Kommunikation. Diese sollten in den Prozessoren, den PCIe-Peripheriegeräten und in den Server-Platinen lokalisiert sind, um Eingriffe des Hypervisors zu minimieren.
5. Konfigurieren und Aktivieren der zuständigen BIOS-, Hypervisor- und Cloud OS-Optionen und Flags, um die zuvor erwähnten Beschleuniger voll ausnutzen zu können
6. Vermeidung der „Überbuchung“ von Speicher und Cores durch zu viel Virtualisierung. Die Zahl der gestarteten VMs sollte begrenzt und überwacht werden.
7. Vermeidung von exzessivem Paging in Guest OS und Host OS durch eine passende Zuordnung von gestarteten Jobs zu verfügbaren Speicherressourcen
8. Vermeidung von doppelter Virtualisierung. VMs innerhalb von VMs geht bei HPC nicht.
9. Wegfall aller sonstiger Middleware für die Inter-Core, Inter-CPU und Inter-Server-Kommunikation und Ersatz durch Open MPI wegen seiner Effizienz, der automatischen Wahl des besten verfügbaren Kommunikationspfades und seiner Zurückhaltung bei der Verwendung von TCP/IP
10. Ersatz der bisherigen Guest OS und Host OS Scheduler durch einen Ansatz, der Prioritäten vergibt und sich der Cloud-Hardware, ihrer Verbindungstopologie und des Problems der „lau-

ten Nachbarn“ bewusst ist und verschiedene Job-Klassen unterstützt (von low performance computing bis high performance computing)

11. Hinzufügen von „Gang Scheduling“ in den Cloud OS Scheduler, um miteinander kommunizierende VMs simultan zu auf physikalische Cores zu schedulen
12. Vermeidung von Warten auf Inter-VM-Kommunikation und Ein-/Ausgabe durch eine Vorabreservierung von IO-Ressourcen bereits in den Anwender-Jobs mit Hilfe des Cloud OS Schedulers und einer Jobablauf-Beschreibungssprache
13. Erweiterung des Cloud OS Schedulers durch ein Leistungslieferungsmodell der Cloud und ein Leistungsverbrauchsmodell seiner Kunden zur vorausschauenden Ressourcenplanung. Diese Angebot-/Nachfragemodelle sollten in der Cloud über die Verbindungstopologie, die Server Hardware und die Anwendungs-Software informiert sein und so bessere Scheduling-Entscheidungen ermöglichen.
14. Vermeidung des Effekts der „lauten Nachbarn“ durch reinen Stapelbetrieb mit Hilfe des Cloud OS Schedulers. Dadurch sollte eine exklusive Zuordnung von Cores zu Jobs erfolgen, ohne die Cores im Zeitscheibenverfahren zwischen Benutzern und Jobs aufteilen zu müssen. Das Zeitscheibenverfahren sollte auf eine Teilmenge von Cores beschränkt bleiben, da es die Inter-VM-Kommunikation stört.
15. Erhöhen der Problemgrößen auf mindestens 100 000 Gitterpunkte, um das Verhältnis von Rechnen und Datenaustauschen zu verbessern

Andernfalls ist unserer Meinung nach ist eine Cloud nur dann HPC-geeignet, wenn das Cloud-Betriebssystem auf einem vorhandenen Parallelrechner installiert wird. Ein Cluster aus Standard-Servern bleibt ohne die Anwendung einiger oder aller der vorgeschlagenen Verbesserungen HPC-ungeeignet.

Literaturliste // References

- [1] U.S. Department of Energy, Office of Advanced Scientific Computing Research (ASCR), *The Magellan Report on Cloud Computing for Science*, December, 2011
- [2] <http://www.openstack.org/>

- and flags to fully exploit the aforementioned accelerators
6. avoidance of memory and core over committing by too much virtualization. The number of VM launches should be limited and carefully monitored.
 7. avoidance of excessive paging in guest OS and host OSes by proper allocation of started jobs to memory resources
 8. avoidance of nested virtualization. VMs inside of VMs are a no-go for HPC.
 9. replacement of other middleware for inter-core-, inter-CPU- and inter-server communication by Open MPI because of its efficiency, automatic selection of the best-available physical-transport layer and its reluctance to use TCP/IP
 10. replacement of the existing Guest OS und Host OS Schedulers by an approach that allows for priorities and that is aware of the cloud hardware, its network topology and the problem noisy neighbours and that dif-

Projektdaten

Das Projekt wird seit Juli 2013 vom SWZ mit insgesamt 0,5 TV-L E13 Stelle am Standort Clausthal gefördert. Beteiligte Wissenschaftler sind:



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



Roman Ledyayev, M.Sc.
Arbeitsgruppe Technische Informatik und Rechnersysteme
Institut für Informatik, Technische Universität Clausthal



Alexander Keidel, M.Sc.
Arbeitsgruppe Technische Informatik und Rechnersysteme
Institut für Informatik, Technische Universität Clausthal

- [3] <http://www.openfoam.com/>
- [4] A. Gupta, *Techniques For Efficient High Performance Computing In The Cloud*, Dissertation, University of Illinois at Urbana-Champaign, 2014
- [5] A. Gupta, L. V. Kale, *Towards Efficient Mapping, Scheduling, and Execution of HPC applications*, 27th IEEE International Symposium on Parallel & Distributed Processing Workshop, May 20-24, Boston, USA, 2013
- [6] A. Gupta, L. V. Kale, D. Milojicic, P. Faraboschi, S. M. Balle, *HPC-Aware VM Placement in Infrastructure Clouds*, IEEE International Conference on Cloud Engineering, March 25-28, San Francisco, USA, 2013
- [7] Stelios Sotiriadis, Nik Bessis, Fatos Xhafa, Nick Antonopoulos, *From meta-computing to interoperable infrastructures: A review of meta-schedulers for HPC, grid and cloud*, 26th IEEE International Conference on Advanced Information Networking and Applications, March 26-29, Fukuoka, Japan, 2012
- [8] N. Bessis, S. Sotiriadis, V. Cristea, F. Pop, *Modelling Requirements for Enabling Meta-Scheduling*, Third International Conference on Intelligent Networking and Collaborative Systems, 30 Nov - 02 Dec., Fukuoka Institute of Technology, Fukuoka, Japan, 2011
- [9] Iman Sadooghi, Sandeep Palur, Ajay Anthony, Isha Kapur, Karthik Belagodu, Pan-kaj Purandare, Kiran, Ramamurty, Ke Wang, Ioan Raicu, *Achieving Efficient Distributed Scheduling with Message Queues in the Cloud for Many-Task Computing and High-Performance Computing*, 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, May 26-29, Chicago, USA, 2014
- [10] Han Zhao, Xiaolin Li, *Designing Flexible Resource Rental Models for Implementing, HPC-as-a-Service in Cloud*, 26th IEEE International Parallel and Distributed Processing Symposium Workshops & PhD Forum, DOI 10.1109/IPDPSW.2012.324, 21-25 May, Shanghai, China, 2012
- [11] Yanyan Hu, Xiang Long, Jiong Zhang, *Enhance Virtualized HPC System Based on I/O Behavior Perception and Asymmetric Scheduling*, 14th IEEE International Conference on High Performance Computing and Communications, 25-27 Jun, Liverpool, UK, 2012.
- [12] 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.
- [13] <http://www.mcs.anl.gov/research/projects/mpi/>
- [14] <http://www.cloudbus.org/cloudsim/>
- [15] <http://www.opensourceforu.com/2015/01/getting-started-greencloud-simulator/>
- [16] <http://www.arcos.inf.uc3m.es/~icancloud/Home.html>
- [17] S. K. Garg, R. Buyya, *NetworkCloudSim: Modelling Parallel Applications in Cloud Simulations*, Proc. Fourth IEEE International Conference on Utility and Cloud Computing, 5-8 Dec., Melbourne, Australia, 2011
- [18] C. Thiam, G. Da Costa, J.-M. Pierson, *Cooperative Scheduling Anti-load balancing Algorithm for Cloud : CSAAC*, IEEE International Conference on Cloud Computing Technology and Science, 2-5 Dec., Bristol, UK, 2013
- [19] 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.
- [20] <https://msdn.microsoft.com/en-us/library/windows/hardware/hh440148%28v=vs.85%29.aspx>
- [21] <http://www.intel.de/content/www/de/de/intelligent-systems/intel-technology/vt-directed-I/O-spec.html>, 2013, abgerufen am 24.07.2014.
- [22] http://de.wikipedia.org/wiki/Generic_Routing_Encapsulation_Protocol

ferentiates jobs into classes ranging from low performance computing to high performance computing.

11. Adding to the cloud OS scheduler a “gang scheduling” feature, that simultaneously allocates communicating sets of tasks to cores for better inter-process communication
12. avoidance of waiting for IO and communication by allowing for an advanced reservation of IO resources already in the jobs via the cloud OS scheduler and a job-workflow description-language
13. enhancement of the cloud OS scheduler by a performance delivery model of the cloud and a resource consumption model of its clients for predictive resource planning. These demand/offer models should allow for interconnect topology-awareness, server hardware-awareness and application-awareness and contribute to more profound scheduling decisions.
14. avoidance of the „noisy-neighbour“ effect by batch processing in the cloud. The cloud OS scheduler should make an exclusive allocation of cores to jobs without any time sharing of cores between jobs and users. The time sharing in the cloud should be restricted to a subset of cores, because it disturbs inter-VM communication.
15. incrementing of the problem sizes to at least 100 000 grid points to make the ratio between computation and communication better

Our opinion is that otherwise a cloud is only HPC-capable if the cloud operation system is installed on an existing a high-performance parallel computer, but not on a cluster of standard servers.

Eine cloud-basierte Software für verteilte Simulation

Teilprojekt - Schnittstellenentwicklung und Zuverlässigkeitstechnik

Jens Grabowski, Fabian Glaser, Xiaowei Wang

Der Begriff der Cloud Computing-Dienste deckt den gesamten Bereich von einfachen Datensicherungen bis hin zu der Bereitstellung von Computing-Clustern oder ganzen Rechenzentren in einer Remote-Umgebung ab. Unternehmen können Cloud-Computing zu nutzen, um Spitzenlasten in ihrer IT-Infrastruktur abzufangen, indem sie die benötigten Ressourcen auf Abruf von Cloud-Computing-Anbieter mieten. Die Flexibilität, Wirtschaftlichkeit und Benutzerfreundlichkeit machen Cloud-Dienste auch zu einem attraktiven Modell für rechenintensive wissenschaftliche Fragestellungen. Einzelnen Forschungsgruppen müssen keine eigene IT-Infrastruktur vorhalten und warten, sondern können Cloud-Computing-Dienste verwenden, um den eigentlichen Fragestellungen nach zu gehen.

Für Cloud-Systeme existieren verschiedene Implementierungen und Service-Modelle, für die es eine vom National Institutes für Standards und Technologie (NIST) erstellte Klassifikation gibt (siehe [1]). Infrastructure as a Service (IaaS) stellt die niedrigste Abstraktionsebene dar. Hier kann der Benutzer (virtualisierte) Ressourcen wie Rechenleistung, Arbeitsspeicher oder Datenpeicher direkt erwerben, ist in der Lage, seinen eigenen Software-Stack auf den Systemen einzusetzen und ist für die Wartung seiner Software selbst verantwortlich. Ein Beispiel für IaaS sind die meisten die von der Amazon Web Services (AWS) angebotenen Dienstleistungen [2]. Platform as a Service (PaaS) beschreibt ein Service-Angebot, das es dem Nutzer ermöglicht, auf einer vorgegebenen Plattform eigene Applikationen zu entwickeln und zu implementieren. Dies bedeutet, dass die Benutzer keine eigenen Ressourcen in die Verwaltung der Infrastruktur investieren müssen. Die Google App Engine [3] und Microsoft Windows Azure [4] sind

klassische Beispiele für PaaS. Als höchste Abstraktionsebene bietet Software as a Service (SaaS) vollständige Anwendungen als Dienstleistungen über das Internet an, in der Regel über einen Standard-Web-Browser. Wie auch schon bei PaaS, hat der Nutzer keine Kontrolle über die Infrastruktur selbst. Ein Beispiel für SaaS sind die Google Docs [5], in denen eine Gruppe von Benutzern zusammenarbeiten kann, um Texte online bearbeiten und teilen zu können. All diese verschiedenen Service-Modelle können auf verschiedenen Cloud-Bereitstellungsmodellen eingesetzt werden. Die Cloud-Bereitstellungsmodelle unterscheiden sich danach, von wem sie angeboten und von wem sie genutzt werden. Private Clouds werden von einem einzelnen Unternehmen intern bereitgestellt und genutzt. Im Gegensatz dazu stehen Public Clouds (meist gegen eine Gebühr) der Öffentlichkeit zur Verfügung. Community Clouds sind Angebote, die von mehreren Organisationen gemeinsam genutzt werden und oftmals auf besondere Aufgaben ausgerichtet sind. Hybrid Clouds schließlich stellen eine Mischung der verschiedenen beschriebenen Cloud-Bereitstellungsmodelle dar.

In unserem Projekt implementieren wir eine Cloud-Plattform, die in Form eines Platform as a Service Systems auf einfache Weise Dienste bereitstellt, die speziell auf wissenschaftliche Simulationen ausgerichtet sind. Abbildung 1 zeigt, wie sich dieser Service in den oben beschriebenen Cloud-Stack eingliedert.

In dem Projekt werden drei verschiedene Aspekte einer Simulation Platform as a Service Cloud betrachte: die Zuverlässigkeit von Cloud-Anwendungen, die Benutzerfreundlichkeit von Cloud-Plattformen für einzelne Wissenschaftler und die

A Cloud-Based Software Infrastructure for Distributed-Simulation

Sub Project – Interface Development and Reliability Engineering

Jens Grabowski, Fabian Glaser, Xiaowei Wang

Cloud computing services range from simple data backup to the possibility of deploying entire compute clusters or data centers in a remote environment. Companies can utilize cloud computing to serve peak-loads on their IT infrastructure by automatically renting the needed resources on-demand from cloud computing providers. The flexibility, cost-efficiency and user-friendliness of cloud services make it also an attractive model to address computational challenges in the scientific community. Individual research groups do not need to provide and maintain IT-infrastructure on their own, but instead rely on cloud-computing services to satisfy their needs.

Different deployment and service models exist for cloud systems, they are commonly classified according to the definition of the National Institutes for Standards and Technology (NIST) [1].

Infrastructure as a Service (IaaS) provides the lowest level of abstraction. Here, the user can acquire (virtualized) resources such as processing power, memory or storage directly, is able to deploy his own software stack on top of the resources, and is responsible for keeping the software up-to-date. One example for IaaS are most of the services offered by the Amazon Web Services (AWS) [2]. Platform as a Service (PaaS) describes a service level, which offers the users a platform to develop and deploy their own applications. Hereby, the users are liberated from managing the infrastructure themselves. The Google App Engine [3] and Microsoft's Windows Azure [4] are well-known examples of PaaS. At the highest level of abstraction, Software as a Service (SaaS) offers full grown applications as services over the internet, typically accessible through a standard web-browser. Like in PaaS, the users have no control over the infra-

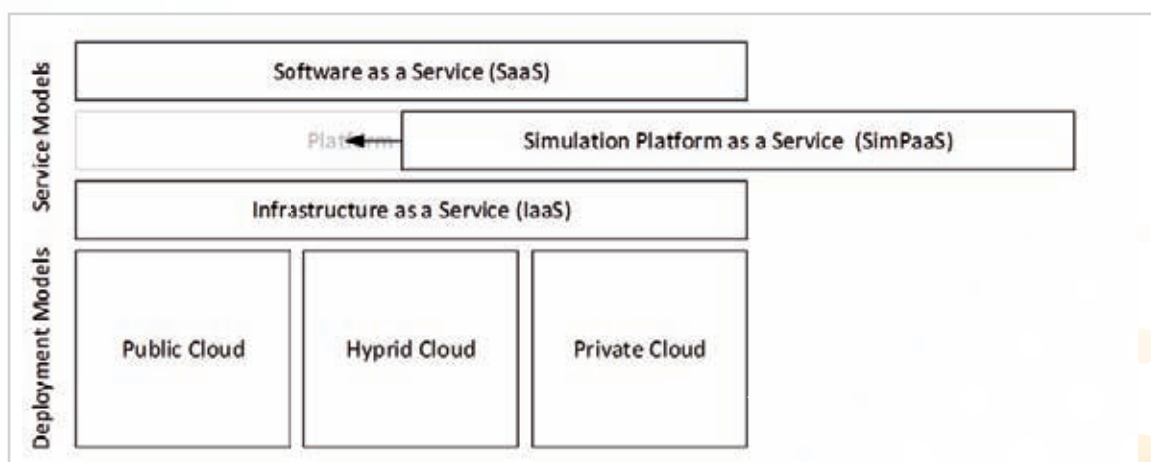


Abbildung 1: Positionierung von SimPaaS im Cloud-Stack.

Figure 1: Positioning SimPaaS inside the cloud stack.

optimierte Nutzung von Cloud-Ressourcen. Die ersten beiden Aspekte werden von Wissenschaftlern aus dem Bereich der Softwaretechnik für Verteilte Systeme an der Universität Göttingen unter Leitung von Herrn Prof. Grabowski untersucht, der dritte Aspekt wird von Wissenschaftlern aus dem Bereich der Technischen Informatik der TU Clausthal unter Leitung von Herrn Prof Richter untersucht. In diesem Teilabschnitt werden die Arbeiten, die an der Universität Göttingen ausgeführt werden, dargestellt.

Zuverlässigkeitstechnik

Neben dem schnellen Wachstum des Cloud Computing, hat die Zuverlässigkeit der Cloud-Anwendungen noch nicht das von den Nutzern gewünschte Niveau erreicht. Da unzuverlässige Cloud-Dienste zu Einnahmeausfällen und Datenverlusten führen können, bedarf die Bewertung und Verbesserung der Zuverlässigkeit der Cloud-Systeme und der darauf laufenden Anwendungen signifikanter Aufmerksamkeit von Forschung und Industrie [2] [3]. Aufgrund des tiefgehenden Abhängigkeiten [4] der Cloud-Objekte wie physischer Server, virtueller Maschinen (VMs), Plattformen, Dienstleistungen und Management-Software etc. in den verschiedenen Schichten: Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) und physische Infrastruktur, ist es eine Aufgabe auf System-Level, die Zuverlässigkeit der Cloud-Anwendungen zu beurteilen, da die Zuverlässigkeit der Objekte in den oberen Schichten abhängig von der Zuverlässigkeit der Objekte auf den unteren Schicht ist. Die hierarchische Abhängigkeit zwischen Cloud-Objekte erschwert es, die Ursache eines Ausfalls zu bestimmen, d.h. den Ort zu bestimmen, an dem Maßnahmen zur Verbesserung der Zuverlässigkeit ansetzen sollten.

Stand der Technik

Zheng et al. [2] haben ein FTCloud genanntes Framework entworfen, welches es ermöglicht, die in Bezug auf die Abhängigkeitsbeziehungen und Verwendungshäufigkeiten relevanten Komponenten auszuwählen, um die optimale Fehlertoleranz-Strategie für komponentenbasierte

Cloud-Anwendungen zu bestimmen. Allerdings werden bei FTCloud die Hardware-Komponenten nicht mit berücksichtigt. Dai et al. [3] unterteilen Cloud-Service Ausfälle in Fehler während der Anmeldephase und Fehler während der Ausführung. In diesem Kontext setzte er Markov-Modell und Methoden der Graphentheorie ein, um die Zuverlässigkeit von Cloud-Services zu analysieren. Thanakornworakij et al. [4] betrachten ein Zuverlässigkeitsmodell für High-Performance Computing Anwendungen unter Berücksichtigung der Verbindung von Software-Fehlern und Hardware-Ausfälle. Allerdings wird in keiner der genannten Arbeiten auf die Struktur der Anwendung eingegangen. Tamura et al. [5] betrachten ein Zuverlässigkeitsmodell für Open-Source-Cloud-Software, in dem der Fokus auf der Veränderung der Betriebsumgebungen liegt. Allerdings erlaubt dieser Ansatz weder eine Beurteilung der Zuverlässigkeit der Cloud-Systeme noch der Zuverlässigkeit der Cloud-Anwendungen.

Unser Ansatz

Angesichts der bestehenden Arbeiten und ihrer Einschränkungen, erarbeiten wir einen Rahmen für die Bewertung und Analyse der Zuverlässigkeit von Cloud-Anwendungen. Das Framework nutzt einen gestuften Abhängigkeitsgraph (Layered Dependency Graph, LDG), der es erlaubt, Abhängigkeiten zwischen zusammengehörigen Cloud-Objekte (physischen Servern, virtuellen Maschinen und Service-Instanzen) zu modellieren. Ein LDG enthält von unten nach oben die folgenden drei Schichten: physische Server-Schicht, VM-Schicht und Anwendungsschicht. Cloud-Anwendungen sind in der Anwendungsschicht modelliert.

Außerdem wird ein Verfahren zur Zuverlässigkeitsbewertung, welches auf dem gestuften Abhängigkeitsgraph basiert, vorgestellt. Um die Auswirkungen der Abhängigkeiten auf die Zuverlässigkeit eines Objekts zu analysieren, nehmen wir an, dass die Zuverlässigkeit eines Cloud-Objektes durch die eigene Zuverlässigkeit (innere Sicherheit) und die Zuverlässigkeit der Objekte, auf die es aufbaut, bestimmt wird. Die interne Sicherheit von Cloud-Objekte wird mit einem exponentiellen Zuverlässigkeitsmodell geschätzt. Auf Basis der

structure itself. One example for SaaS is Google Docs [5], where a group of users can collaborate to edit and share text documents online. 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 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 build around special requirements. Hybrid clouds form a mixture of the introduced cloud deployment models.

In our project we implement a specialized cloud platform that eases and optimizes cloud utilization for scientific simulations in the Platform as a Service layer. Figure 1 depicts how our service fits into the cloud stack described above.

In the project, we target three different aspects of a Simulation Platform as a Service cloud, which are: Reliability of cloud applications, the ease of use of cloud platforms for individual scientists, and the optimized usage of cloud resources. The first two aspects are investigated by researchers from the Software Engineering for Distributed Systems Group at the University of Göttingen, headed by Prof. Grabowski, while the third aspect is targeted by researchers from the Computer Engineering Group of the TU Clausthal, led by Prof. Richter. The remainder of this chapter will introduce the work conducted in Göttingen, while the work of Prof. Richter's group is presented in the next chapter in further detail.

Reliability Engineering

Beside the rapid growth of cloud computing, the reliability of cloud applications is still on the road to satisfy cloud-users. As unreliable cloud services may lead to revenue and data loss, the assessment and improvement of cloud system and applications' reliability attract significant attention of both academia and industry [2][3]. However, the deep dependency stack [4] of cloud objects, such as physical servers, virtual machines (VMs), platforms, services and management software etc., in different layers: Software as a Service (SaaS),

Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and physical infrastructure, makes it a system-level task to assess the reliability of cloud applications, since the reliability of objects in upper layers is dependent on the reliability of objects in lower layers. The hierarchical dependency among cloud objects makes it tough to find out root causes of failures, i.e., where to put efforts to improve the reliability.

State of the art

Zheng et al. [2] propose a framework called FTCloud to choose significant components according to their invocation relationships and frequencies to determine the optimal fault tolerance strategy for component-based cloud applications. But FTCloud does not take hardware components into consideration. Dai et al. [3] divide the cloud service failures into request stage failures and execution stage failures, and employ Markov model and graph theory to analyze the reliability of cloud services. Thanakornworakij et al. [4] propose a reliability model for high performance computing applications considering the correlation of software failures and hardware failures. However, neither of above work considers the structure of the application. Tamura et al. [5] propose a reliability model for open source cloud software focusing on the operational environment fluctuation. But it is assessing the reliability of cloud systems rather than the reliability of cloud applications.

Our Approach

Given the existing works and its shortcomings, we propose a framework to assess and analyze the reliability of cloud applications. The framework utilizes a layered dependency graph (LDG) to model dependencies between related cloud objects (physical servers, VMs and service instances). One LDG contains three layers from bottom to top: physical server layer, VM layer and service instance layer. Cloud applications are modeled in the service instances layer.

Furthermore, a reliability assessment method is proposed based on the layered dependency

internen Sicherheit der einzelnen Cloud-Objekte und der Struktur der Anwendung kann die Zuverlässigkeit der gesamten Cloud-Anwendungen geschätzt werden.

Eine erste Fallstudie zur Überprüfung des erarbeiteten Rahmens wurde in Java auf der Basis einer Cloudify PaaS [5] Cloud umgesetzt und auf dem einem experimentellen Openstack [6] System getestet. Der Rahmen besteht aus drei Komponenten: einem Monitoring-System, ein System zur Abhängigkeitsanalyse und ein Zuverlässigkeitsanalysator. Das System zur Abhängigkeitsanalyse speichert die Abhängigkeitsinformationen aus Cloudify und Openstack, um den gestuften Abhängigkeitsgraphen zu erstellen. Cloudify wird eingesetzt, um Dienstinstanzen und VMs zu überwachen und Ganglia wird verwendet, um die physischen Server zu überwachen. Der Zuverlässigkeitsanalysator ist für die Beurteilung der Zuverlässigkeit der Anwendung und der Objekte in Bezug auf die vom Monitoring-System und dem LDG ermittelten Fehlerdaten verantwortlich.

Ausblick

Wie die vorläufige Fallstudie zeigt, lässt sich das vorgestellte Framework zur Bewertung der Zuverlässigkeit von Cloud-Objekte einsetzen. Das vorläufige Experiment enthält jedoch keine VM-Ausfälle oder Ausfälle physischer Server. Daher ist die Entwicklung eines Fehler-Injektors in das Framework geplant. Durch das künstliche Hervorrufen eines Fehlermodus für ein Cloud-Objekt, soll das Framework mit mehr statistischen Daten und auf einer komplexeren Struktur überprüft werden. Weitere Details zu dem Modell und der Fallstudie sind in [6] dargestellt.

Schnittstellenentwicklung

Der Abstraktionsgrad und die Flexibilität der Cloud-Dienste führen dazu, dass sich einzelne Wissenschaftler nicht mehr mit der Bereitstellung und der Verwaltung der für sie notwendigen Infrastruktur befassen müssen. Wir untersuchen, wie eine Cloud-basierte Software-Infrastruktur Wissenschaftler während des gesamten Simulationslebenszyklus von Modellierung über Implementie-

rung, Durchführung und Überwachung bis hin zur Erfassung und Darstellung der Simulationsergebnisse unterstützen kann.

In der Modellierungsphase werden Software-Engineering-Techniken, die aus der modellgetriebenen Entwicklung (MDD) bekannt sind, verwendet. Grafische Modellierungssprachen, wie z.B. die Unified Modeling Language (UML) [7], können erweitert werden, um den Modellierungsprozess für Simulationen oder domänenspezifische Sprachen (DSL), die auf die Modellierung von Cloud-basierten Simulationen abziehen, zu unterstützen.

Stand der Technik

Es gibt verschiedene Forschungsansätze, die sich mit der Anwendung von MDD-Techniken auf Cloud Computing und Simulation befassen. Wir unterscheiden zwischen Forschung, die auf die Modellierung von Simulationsanwendungen abzielt, Forschung in der MDD auf den gesamten Entwicklungsprozess von Cloud-Anwendungen angewandt wird und Werkzeugen und Techniken, die Modelle verwenden, um die Bereitstellung von Anwendungen in Cloud-Umgebungen zu automatisieren. ODIM [7] und onto-UML [8] sind Ansätze, um Modellierungssprachen insbesondere für die Entwicklung von Simulationen zu definieren. ODIM stellt ein Modell für agentenbasierte Simulation dar, während onto-UML auf diskrete Ereignis-Simulationen abzielt. MODAClouds [9] und PaaSage [10] gehören zur zweiten Kategorie und fokussieren sich auf die Entwicklung von Verfahren für die Cloud-Anwendungen. TOSCA [11] und OCCI [12] sind Beispiele aus der dritten Kategorie. Das erste System ist ein vielversprechender Kandidat für ein standardisiertes Modell zur Koordination von Cloud-Anwendung und das zweite System, welches ursprünglich als standardisiertes API für Clouds entwickelt wurde, definiert ein Modell für die Bereitstellung von Anwendungen auf IaaS-Clouds. Cloud-Koordination bedeutet hierbei, der koordinierte Start von Ressourcen in Cloud-Umgebungen.

Die oben beschriebenen Ansätze konzentrieren sich entweder auf die Entwicklung von Simulationen und Anwendung, wobei bei den Cloud-spezifischen Lösungen Interoperabilität und

graph. To analyze the impact of dependencies to the reliability of an object, we assume that the reliability of a cloud object is determined by the reliability of itself (inner reliability) and the reliability of objects on which it depends. The inner reliability of cloud objects is estimated with the exponential reliability model. With the inner reliability of cloud objects and the structure of the application, the reliability of cloud applications can be estimated.

An initial case study is conducted for verifying the proposed framework, which is implemented with Java language based on a Cloudify PaaS [5] cloud built on top of an experimental OpenStack [6] cloud. The framework contains three components: a monitor, a dependency analyzer and a reliability analyzer. The dependency analyzer collects the dependency information from Cloudify and OpenStack to create the layered dependency graph. Cloudify is employed to monitor service instances and VMs, and Ganglia is used to monitor physical servers. The reliability analyzer is responsible for assessing the reliability of the application and objects according to the field failure data obtained from the monitor and the LDG obtained from the dependency analyzer.

Outlook

As the preliminary case study shows, the framework can assess the reliability of cloud objects and applications. However, the preliminary experiment shows no VM or physical server failures. We are going to integrate a fault injector into our framework in the future. By setting the failure mode of cloud objects, our framework will be validated with more usage information and on more complex structures. Further details on the model and the case study are presented in [6].

Interface Development

The abstraction level and flexibility of clouds offers the chance to free the individual scientists from the burden of infrastructure provisioning and management. We are investigating how a cloud-based software infrastructure can support scientists throughout the whole simulation life-cycle,

which involves modeling, deployment, execution, monitoring, and collection and representation of the simulation results.

In the modeling phase, software engineering techniques known from model-driven development (MDD) are utilized. Graphical modeling languages, e.g., the Unified Modeling Language (UML) [7] can be extended to support the modeling process for simulations or domain-specific languages (DSL) targeting the modeling of cloud-based simulation can be developed.

State of the art

Different projects and research on how to apply MDD techniques to cloud computing and simulation exist. We differentiate between research that targets the modelling of simulation applications, research that applies MDD to the whole development process of cloud applications and tools and techniques that utilize models to automate the deployment of applications in cloud environments. ODIM [7] and onto-UML [8] are approaches to define modelling languages especially targeted for developing simulations. The former provides a model for agent-based simulation while the later targets Discrete-Event-Simulation. MODAClouds [9] and PaaSage [10] fall in the second category and focus on the development procedure of applications for the cloud from scratch, especially targeting a cloud provider agnostic design. TOSCA [11] and OCCI [12] are examples from the third category. The first one is a promising candidate for a standardized model for cloud application orchestration and the second, while it was originally developed as a standardized API for clouds, defines a model for application deployments on IaaS clouds. Cloud orchestration hereby refers to the orchestrated launch of resources in cloud environments.

The approaches described above either focus on the development of simulations and application from scratch, whereby the cloud specific solutions focus on the interoperability and cloud provider agnostic design, or they focus on deployment, whereby the cloud characteristics scalability and elasticity are not targeted. In the scope of the SWZ, we deal with already existing simulation

Unabhängigkeit von bestimmten Cloud-Anbieter im Vordergrund stehen, oder sie konzentrieren sich auf den Einsatz der Cloud-Lösungen, wobei jedoch die Merkmale Skalierbarkeit und Elastizität nicht direkt berücksichtigt werden. Im Rahmen des SWZ beschäftigen wir uns mit bereits bestehenden Simulationsanwendungen und fokussieren uns auf die Modellierung ihres Einsatzes in Cloud-Umgebungen, wobei Skalierbarkeit und Elastizität transparent genutzt werden sollen. Ein solches

Modell kann verwendet werden, um eine Simulationsanwendung mit geringem Aufwand in der Cloud anzubieten und die Wissenschaftler von den internen Aspekten der Cloud abzuschirmen.

Unser Ansatz

Die Bereitstellung von (Simulations-) Anwendungen in einer Cloud ist keine triviale Aufgabe: Eine

Projektdaten

Das Projekt wird seit Juli 2013 vom SWZ mit insgesamt 0,5 TV-L E13 Stelle am Standort Göttingen gefördert. Beteiligte Wissenschaftler sind:



Prof. Dr. Jens Grabowski
Arbeitsgruppe Softwaretech-
nik für Verteilte Systeme
Institut für Informatik,
Universität Göttingen



Fabian Glaser, M.Sc.
Arbeitsgruppe Softwaretech-
nik für Verteilte Systeme
Institut für Informatik,
Universität Göttingen



Xiaowei Wang, M.Sc.
Arbeitsgruppe Softwaretech-
nik für Verteilte Systeme
Institut für Informatik,
Universität Göttingen

applications and focus on modelling their deployment in cloud environments such that scalability and elasticity can be utilized transparently. Such a model can be used to easily deploy a simulation application in the cloud and shield the scientist from cloud internals.

Our Approach

Deploying (simulation) applications in clouds is not a trivial task: A suitable (virtualized) infrastructure needs to be selected and deployed, the deployed virtual machines need to be configured to meet the target application requirements and the application itself need to be installed and configured. A cloud deployment model hence need 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 application from scratch.

Due to the complexity described above, we are building on existing open-source solutions to model simulation applications and to automate their deployments and adaption. On the upmost level we are modelling applications with help of the aforementioned upcoming cloud orchestration standard TOSCA, which can be extended with help of plugins to define new functionality. A modified version of Cloudify is used to automatically deploy the modelled infrastructure. The configuration management tool Ansible [8] is used to enforce the required software configuration on the launched infrastructure. Thereby, we are especially interested in providing an easy to use scalability mechanism that can be used by non-computer scientists to adapt the performance of the deployed simulation application to the requirements. The goal is to provide a prototypical web-service, where the scientist can select from a list of simulation application models which is then automatically deployed on the cloud infrastructure. The scale of the provided infrastructure

is hereby determined by parameters that can be set at deployment-time.

Outlook

To further foster the proliferation of ease to use cloud tools and cloud infrastructure for simulations, we are collaborating with the “Gesellschaft für wissenschaftliche Datenverarbeitung Göttingen” (GWDG), who serves as the IT-infrastructure provider for the University and the Max-Planck Society in Göttingen. The GWDG has developed a solution for running simulations from engineering in the cloud, which is called Cloud4E [13]. Cloud4E provides mechanisms to wrap existing simulation applications with a thin adapter that enables them to be run and steered as services in a remote cloud environment. While our focus is on the modelling and deployment of cloud applications, Cloud4E focuses on the infrastructure to communicate with simulations run in a cloud. Therefore the approaches are complementary and efforts are planned to integrate both.

Collaborating Partners

To be able to develop solutions that are suitable for scientists from heterogeneous scientific domains, our research is driven by three different use cases, which represent heterogeneous requirements. Prof. Quadt research group (University of Göttingen) is providing a use case from the domain of high energy physics, Prof. Schöbels research group (University of Göttingen) from optimization of public transport networks, and the research group of Prof. Brenner (TU Clausthal) is providing a use case from material science.

Literatur // References

- [1] P. Mell, T. Grace, “The NIST definition of cloud computing.” 2011.
- [2] „Amazon Web Services (AWS) – Cloud Computing-Services“ [Online]. Available: aws.amazon.com/. [Accessed March 2015].
- [3] „App Engine – Google Cloud Platform“ [Online]. <https://cloud.google.com/appengine/>. [Accessed March 2015].

geeignete (virtualisierte) Infrastruktur muss dafür ausgewählt und aufgesetzt werden, die eingesetzten virtuellen Maschinen müssen konfiguriert werden, um die Anforderungen der Zielanwendungen zu erfüllen, und die Anwendungen selbst müssen installiert und konfiguriert werden. Ein Cloud-Bereitstellungsmodell muss daher die Hardware-Konfiguration, die Software-Konfiguration und eine Abbildung der Anwendungskonfiguration auf die Hardware-Komponenten enthalten. Um im vollen Umfang von der Skalierbarkeit und Elastizität des Cloud-Systems profitieren zu können, müssen auch Informationen darüber mit einbezogen werden, wie die Hard- und Softwarekomponenten so skaliert werden können, dass die Simulationsimplementierung leicht an veränderte Leistungsanforderungen angepasst werden kann, ohne dass die Anwendung dafür von Grund auf neu implementiert werden muss.

Aufgrund dieser Komplexität entwickeln wird auf Basis bestehender Open-Source-Software Lösungen zur Modellierung von Simulationsanwendungen und ihrer automatischen Bereitstellung und Anpassung. Auf der obersten Ebene modellieren wir Anwendungen mit Hilfe des erwähnten Cloud-Koordinationsstandards TOSCA, der mit Hilfe von Plug-Ins erweitert werden kann, um neue Funktionen zu definieren. Eine angepasste Version des Cloudify wird verwendet, um die modellierte Infrastruktur automatisch auszurollen. Das Konfigurationsmanagementwerkzeug Ansible [8] wird verwendet, um die erforderliche Software-Konfiguration auf der Start-Infrastruktur einzusetzen. Dabei ist es von besonderem Interesse, einfache, skalierbare Mechanismen, die durch Nicht-Informatiker verwendet werden können, um die Leistungsfähigkeit der eingesetzten Simulationsanwendung an die Anforderungen anzupassen, anzubieten. Ziel ist es, einen prototypischen Web-Dienst anzubieten, bei dem Wissenschaftler aus einer Liste von Simulationsanwendung Modellen auswählen können, die dann automatisch auf der Cloud-Infrastruktur zur Verfügung gestellt werden. Das Art der angebotenen Infrastruktur wird hierbei

durch Parameter, die vor der Bereitstellung eingestellt werden können, bestimmt.

Ausblick

Zur Förderung der weiteren Verbreitung leicht einzusetzenden Cloud-Werkzeugen und Cloud-Infrastrukturen für die Simulationen, arbeiten wir derzeit mit der „Gesellschaft für wissenschaftliche Datenverarbeitung Göttingen“ (GWDG) zusammen, die als IT-Infrastruktur-Anbieter für die Universität und die Max-Planck-Gesellschaft in Göttingen dient. Die GWDG hat eine Lösung für die Durchführung von Simulationen im Bereich der Ingenieurwissenschaften in der Cloud, die Cloud4E [13], entwickelt. Cloud4E stellt Mechanismen bereit, um bestehende Simulationsanwendungen mit einem leichtgewichtigen Rahmen zu versehen, der es ermöglicht, sie als Service in einer Remote-Cloud-Umgebung auszuführen und zu steuern. Während unser Schwerpunkt auf der Modellierung und Implementierung von Cloud-Anwendungen liegt, konzentriert sich Cloud4E auf die Infrastruktur, um mit den Simulationsläufen in der Cloud zu kommunizieren. Daher ergänzen sich die beiden Ansätze und es sind Maßnahmen geplant, um sie zu integrieren.

Kooperationspartner

Um Lösungen, die für Wissenschaftler aus verschiedenen Bereichen stammen, entwickeln zu können, betrachten wir in unserer Forschung drei verschiedene Anwendungsfälle, um diesen heterogenen Anforderungen gerecht zu werden. Die Arbeitsgruppe von Prof. Quadt (Universität Göttingen) liefert einen Anwendungsfall aus dem Bereich der Hochenergiephysik dar, Prof. Schöbels (Universität Göttingen) Arbeitsgruppe befasst sich mit der Optimierung öffentlicher Verkehrsnetze und die Arbeitsgruppe von Prof. Brenner (TU Clausthal) stellt einen Anwendungsfall aus den Materialwissenschaften bereit.

- [4] „Microsoft Azure: Cloud Computing Platform & Services“ [Online]. <http://azure.microsoft.com/>. [Accessed March 2015].
- [5] „Cloud Orchestration & Cloud Automation the DevOps Way“. [Online]. <http://getcloudify.org/>. [Accessed March 2015].
- [6] „OpenStack Open Source Cloud Computing Software“. [Online]. <https://www.openstack.org/>. [Accessed March 2015].
- [7] „Unified Modeling Language (UML)“. [Online]. <http://www.uml.org>. [Accessed March 2015].
- [8] „Ansible is Simple IT Automation“. [Online]. <http://www.ansible.com/>. [Accessed March 2015].
- [9] Z. Zheng, T. Zhou, M. Lyu and I. King, „Component Ranking for Fault-Tolerant Cloud Application“ IEEE Transactions on Service Computing, pp. 1-12, 2011.
- [10] Y. Dai, B. Yang, J. Dongarra and G. Zhang, „Cloud Service Reliability: Modeling and Analysis“ in Proceedings of the 15th IEEE Pacific Rim International Symposium on Dependable Computing, 2009.
- [11] T. Thanakornworakij, R. Nassar, C. Leangsuksun and M. Pūn, „A Reliability Model for Cloud Computing for High Performance Computing Applications“ in Euro-Par 2012: Parallel Processing Workshops, 2013.
- [12] Y. Tamura, M. Kawakami and S. Yamada, „Reliability modeling and analysis for open source cloud computing“ in Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 2013.
- [13] X. Wang, and J. Grabowski, „A Reliability Assessment Framework for Cloud Applications“, The Sixth International Conference on Cloud Computing, GRIDs, and Virtualization , IARIA XPS Press, 2015.
- [14] J. N. Duarte and J. de Lara, „ODIM: A model-driven approach to agent-based simulation“ in Proceedings of the 23rd European Conference in Modelling and Simulation (ECMS 2009), 2009.
- [15] G. Guizzarardi and G. Wagner, „Conceptual simulation modeling with onto-UML“ in WSC ,12 Proceedings of the Winter Simulation Conference , 2012.
- [16] „MODAClouds“ [Online]. Available: <http://www.modaclouds.eu/> [Accessed November 2014].
- [17] „PaaSage: Model-based Cloud Platform Upperware“ [Online]. Available: www.paasage.eu. [Accessed November 2014].
- [18] „Topology and Orchestration Specification for Cloud Applications Version 1.0“ [Online]. Available: <http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html>. [Accessed March 2015].
- [19] „Open Cloud Computing Interface | Open Standard | Open Community“ [Online]. Available: <http://occi-wg.org/>. [Accessed March 2015].
- [20] „Cloud4E | Trusted Cloud Computing for Engineering“ [Online]. Available: www.cloud4e.de. [Accessed March 2015].

Simulationsbasierte Qualitätssicherung für Software Systeme

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

Aus der Sicht eines für ein Software-Projekt verantwortlichen Projektmanagers ist es wichtig, verschiedene Umsetzungsstrategien zur Verbesserung des Entwicklungsprozesses in Betracht zu ziehen. Verschiedene Faktoren beeinflussen den Prozess der Softwareerstellung und die resultierende Produktqualität. Der Manager muss festlegen, wie viele Ressourcen (einschließlich der Entwickler) eingesetzt werden sollen, welchen technischen Grundlagen zu folgen ist und wie viel Zeit für die verschiedenen Entwicklungsphasen (wie z.B. Entwurf, Entwicklung und Tests) eingeplant werden soll. Die Entscheidungen über die möglichen Handlungsalternativen im Rahmen eines Projekts müssen dabei in Übereinstimmung mit den angestrebten Ergebnissen und Anforderungen stehen. Unser Ziel ist es, ein Werkzeug zur Simulation von Software-Entwicklungsprozessen, das Entscheidungshilfen für Manager bietet, zu erstellen. Dieses Werkzeug soll es ermöglichen, Alternativen und Lösungen zu simulieren, um so die jeweils beste Alternative, die die meisten möglichen Probleme bereits im Vorherein ausschließt, zu identifizieren und die zukünftigen Vorgehensweisen entsprechend ausrichten zu können. In der Simulation sollen z.B. die Möglichkeiten, mehr Zeit für Softwaretests einzuplanen oder Teile der Softwarearchitektur oder der Parameter zu verändern und damit die Arbeit in dem Entwicklerteam entsprechend zu verändern und dadurch verschiedene Entwicklungsstrategien zu ermöglichen berücksichtigt werden. Um solch ein Werkzeug zu erstellen, wird ein Simulationsmodell auf Basis der Erkenntnisse, die sich im Rahmen früherer Softwareprojekte ergeben haben, entworfen. Beispielsweise ist bekannt, dass Komponenten, die häufig geändert werden und viele Schnittstellen besitzen, dazu neigen, fehleranfälliger als andere zu sein. Diese Komponenten lassen sich durch eine Analyse der Veränderungen in der Software und dem Quellcode erkennen, so dass geeignete Qualitätssicherungsmaßnahmen vorgenommen

werden können. Der Nachteil dieses heuristischen Ansatzes besteht jedoch darin, dass sich die jeweilige Wirkung der durchgeführten Aktion nur schwer abschätzen lässt. Dieser Nachteil lässt sich durch die Verwendung von Simulationsmethoden beheben, die es ermöglichen, die Auswirkungen von Veränderungen in der Softwareentwicklung vorherzusagen. Basierend auf Open-Source-Software-Archiven, werden Simulationsmodelle erstellt und validiert. Durch die Änderung der Simulationsparameter können Faktoren wie Managemententscheidungen, alternative Softwarearchitekturen und Zusammenstellungen von Entwicklertrams modelliert werden und ihre Auswirkungen auf die Evolution einer Software verfolgt werden. Diese konzentriert sich vor allem auf Aspekte der Qualitätssicherung, zum Beispiel Fehlerverteilungen und Wartbarkeit. Die gewonnenen Ergebnisse werden durch bedingte Zufallsfelder aus und vergleichen unsere Vorgehensweise mit anderen Ansätzen zur Messung der Qualitätssicherung.

Software Evolution

Mit der Einführung des Begriffs der *Software Evolution* und der Veröffentlichung von *Lehmans Gesetzen der Software Evolution* [1] begann die Erforschung von sich ständig ändernder Software. Diese Änderungen werden durch die sich ständig verändernden Entwicklungsumgebungen, an sich die Software anpassen muss, verursacht. Einen großen Vorteil bei der Untersuchung der Software Evolution stellen dabei die Verfügbarkeit von großen Open-Source-Projekten sowie die zugehörigen Versionsverwaltungssysteme und Fehler-Tracking-Systeme und Mailing-Listen dar.

Das Durchsuchen von Software-Archiven ermöglicht es, einen Einblick in die Entwicklungsprozesse zu gewinnen und die Ursachen für Codeänderungen sowie das Verhalten der Software-Entwickler

Simulation-Based Quality Assurance for Software Systems

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

From the point of view of a project manager in charge of a software project, it is important to consider different implementation strategies pushing and improving the development process. Various factors influence the software building process and the resulting quality of the created product. The manager has to plan resources (including developers), which technical principles to follow, and how much time to invest in different development phases such as design, implementation and tests. Decisions on the possible alternative courses of action during a project must be in line with the desired outcomes and demands.

We aim to create a tool for the simulation of software development processes that provides decision-making support for managers. Alternatives and solutions can be simulated which allows the identification of the best alternative on the prevention of possible problems by changing the course of action. The simulation shall account for, e.g., spending more time on testing, changing parts of the software architecture or parameters that influence the work inside the developer team and with it provides different development strategies. To create such a tool, we build a simulation model based on experiences extracted from actual and past software projects. For example, it is known, that modules which are changed often and have many interfaces tend to be more error-prone than others. Through the analysis of the changes in software and source code, it is possible to identify these and to take suitable quality assurance actions. But, with this heuristic approach, it is difficult to appreciate the effect and intensity of the performed action. To overcome these weaknesses, it is a valuable approach to use simulation techniques to predict the resulting changes to the software development process. Based on open source software archives, simulation models are built and validated. If the retrieved simulation parameters are changed, factors, such as man-

agement decisions, alternative software architecture and development team constellation, can be modeled and their influence on software evolution can be traced. This focuses especially on aspects of quality assurance, e.g., bug distribution and maintainability. The gained results are evaluated by conditional random fields and our approach is compared to other approaches for measuring quality assurance.

Software Evolution

With the introduction of the term *Software Evolution* and the release of *Lehman's Laws of Software Evolution* [1] the exploration of continuously changing software began. This change is caused by the ever changing development environment which the software has to adapt to. The availability of large open-source projects as well as the appropriate software version, history of version management systems, issue tracking systems and mailing lists is a big advantage of studying software evolution.

Through mining software archives, we can learn about development processes and the triggers of code changes as well as about the behavior of software developers. Based on the results of the mining process, we mimic the observed behavior of developers in our simulation. For the prediction of the evolution of the software, it is common to use machine learning and statistical approaches. As a result of this project, prediction rules will be established which in combination with the behavior of involved developers build the foundation for our simulation process.

Mining software repositories

A huge amount of software repositories, bug tracking systems, mailing list and forums is free

zu verstehen. Basierend auf den Ergebnissen dieser Datenerhebung ist es uns möglich, das Verhalten von Entwicklern in unserer Simulation nachzubilden. Zur Vorhersage der Evolution von Software werden üblicher Weise maschinelles Lernen und statistische Verfahren eingesetzt. Als Ergebnis dieses Projekts werden Vorhersageregeln aufgestellt, die in Verbindung mit dem Verhalten der beteiligten Entwickler die Grundlage für unsere Simulationsprozess darstellen.

Datenerhebung in Software-Archiven

Eine große Menge von Software-Archiven, Fehler-Tracking-Systemen, Mailing-Listen und Nutzerforen sind über das Internet frei zugänglich, zum Beispiel das PROMISE-Repository [2]. Der erste Schritt zur Untersuchung eines Software-Archivs besteht in der Auswahl der darin zu betrachtenden Software-Projekte. Dabei werden die Parameter Vollständigkeit, Bekanntheit, Projektgröße und Projekttyp berücksichtigt. Im Anschluss daran müssen die in dem Archiv gespeicherten Informationen extrahiert werden. Zu diesem Zweck existieren es einige Tools, wie z.B. *cvsanaly* [3], die das Archiv analysieren und die extrahierten Daten in einer Datenbank speichern. Auf dieselbe Weise kann auch auf Fehler-Tracking-Systeme zugegriffen werden. Ein bisher nicht zufriedenstellend gelöstes Problem im Bereich der Software-Archiv Datenerhebung besteht darin, die Fehler-Informationen mit den Daten aus der Versionsverwaltung in Beziehung zu setzen. Momentan wird diese meist per Textsuche in den Commit-Kommentaren durchgeführt. Weitere Herausforderungen Untersuchung von Software-Archiven stellt die Berücksichtigung von Namenskonventionen unter anderem bei Benutzernamen und E-Mail-Adressen dar.

Sobald die Daten gesammelt und vorbereitet sind, können sie weitergehend analysiert werden. Da wir uns vor allem für Statistiken und Muster, mit denen sich der Software-Erstellungsprozess charakterisieren lässt, interessieren, setzen wir hierfür Tools wie *R* [4] und *Weka* [5] ein. Um unsere momentanen Forschungsfragen zu untersuchen, verwenden wir Ansätze aus dem statistischen und maschinellen Lernen, um Regeln aufzustellen und Beobachtungen belegen zu können, die später

dazu dienen sollen, die Parameter für die Simulation realitätsnah festlegen zu können.

Agentenbasierte Simulation

Agentenbasierte Modellierung und Simulation (ABMS) beschäftigt sich mit Modellierung von Systemen auf der Basis von interagierenden Individuen, d.h. Agenten [6]. Die Vorteile dieses Ansatzes bestehen darin, dass er sehr flexibel ist und es uns ermöglicht, Systeme, deren Komplexität kontinuierlich zunimmt, zu modellieren: Das Wachstum der Komplexität kann implizit durch die Agenten modelliert werden. Das Verhalten wird als eine Reihe von Aktivitäten dargestellt, die die Agenten ausführen, was Änderungen der Eigenschaften der Artefakte des Systems zur Folge hat. Die Artefakte stellen dabei passive Elemente des Systems, wie z.B. Quellcode-Stücke, dar.

Die Entstehung einer Software kann als der Versuch verstanden werden, (wechselnde) äußere Anforderungen zu erfüllen und gleichzeitig die innere Komplexität des Systems zu beherrschen. Daher liegt es nahe, Software Evolution vom Startpunkt des menschlichen Verhaltens aus zu modellieren. Aus diesem Grund schlagen wir ein agentenbasiertes Modell vor, um bestimmte Aspekte der sich entwickelnden Software zu simulieren. Ein großer Vorteil der agentenbasierten Simulation besteht darin, dass man relativ komplexe Prozesse durch einfache Regeln beschreiben kann. Daher ist es möglich, den Einfluss von kleinen Änderungen der Regeln auf den gesamten Systemzustand zu analysieren. Da der Begriff *Agent* in der Literatur nicht einheitlich definiert ist, definieren wir die folgende Terminologie.

Ein Agent besteht aus seiner Menge von Attributen und Verhaltensweisen. Er existiert in einer Umgebung, beispielsweise einem Softwareprojekt, wirkt autonom und ist in der Lage, mit anderen Agenten aus einer Umwelt zu interagieren. Da eine agentenbasierte Simulation als rundenbasiert angesehen werden kann, kann ein Agent jede Runde eine Reihe von gewünschten Aktionen ausführen. Welche Aktion als nächstes ausgeführt wird, hängt von der lokalen Umgebung und dem aktuellen Status des Agenten, ab. Eine Aktion kann den Zustand ausführenden Agenten

accessible via the internet, e.g., the PROMISE repository [2]. The first step in analyzing software repositories is to choose eligible software projects. For this, we examined the data available allowing for completeness, noisiness, size, and project type. After this the facts stored in the repository need to be extracted. For this purpose some tools exist, e.g., *cvsanaly* [3], which parses the repository and stores the extracted data in a database. The same way, bug data can be accessed. One not satisfactory solved problem in the domain of software repository mining is how to link the bug data to the version control data, which means to link the bugs to the software entity. This is done by text search in commit comments in most cases. Other challenges in the mining of software repositories are, e.g., naming conventions like usernames and email aliases.

Once the data is collected and prepared, it is ready for analysis. Since we are interested in statistics and patterns describing the software development process, we use tools like *R* [4] and *Weka* [5]. According to our current question under investigation, we use statistical and machine learning approaches to specify rules and observations which later on help to estimate simulation parameters close to reality.

Agent-based Simulation

Agent-based Modeling and Simulation (ABMS) deals with modeling systems based on interacting individuals, i.e. agents [6]. Advantages are that this approach is very flexible and allows us to model increasingly complex systems: The growth of complexity may be modeled implicitly by the agents. The behavior is modeled as a series of activities performed by the agents, which cause changes of the attributes of the artifacts on which they are working. The artifacts represent passive entities belonging to the software system such as code pieces.

Because software evolution can be understood as a way of humans struggling with the (changing) requirements from the outside and the complexity of the system on the inside, it seems natural to model software evolution from the starting point of human behavior. For that reason, we suggest

an agent-based model to simulate certain aspects of evolving software. One major benefit of agent-based simulation is that you can describe rather complex processes by simple rules. Therefore, it is feasible to analyze the influence of small rule changes to the entire system state. Since the term *agent* is not well defined in the literature, we define the following terminology.

An agent contains a set of attributes and behaviors. It lives in an environment, for example a software project, acts autonomous, and is able to interact with other agents and the environment. Since an agent based simulation can be seen as round based, an agent can execute a set of desired actions each round. Which action will be executed next depends on the agent's local environment and its current state. An action can change the state of the executing agent as well as the state of the other agents in the environment. We distinguish between active and passive agents. The active ones, representing developers, are mainly responsible for executing actions that can change the state of the passive agents. The passive agents represent pieces of software, e.g., modules or requirements. In this way, software evolution can be simulated over the time. Further details on our simulation model and all contained entities are described in the section *Approach*.

Conditional Random Fields

We address the problem of evaluating quality of software systems as a node labeling problem for undirected graphs, the so-called *neighborhood graph*, using the model class of conditional random fields (CRFs) [7]. Here, the nodes are the software modules, where two nodes are connected by an edge, if there is an immediate semantic dependency between the two modules. CRFs are conditional models that specify the probability of node labeling given observed data that in turn are formed by the knowledge on the problem under study. In our case the observations are determined by the agent-based simulation. CRFs are maximum entropy models. The defining feature of a CRF consists in a single exponential model for the probability of the entire node labeling.

sowie den Zustand der anderen Agenten in seiner Umwelt ändern. Wir unterscheiden zwischen aktiven und passiven Agenten. Die aktiven Agenten repräsentieren die Entwickler und sind hauptsächlich für das Ausführen von Aktionen, die den Zustand der passiven Agenten ändern können, verantwortlich. Die passiven Agenten stellen Teile der Software, z.B. Module oder Anforderungen, dar. Auf diese Weise können Software-Entwicklung über die Zeit hinweg simuliert werden. Weitere Informationen zu unserem Simulationsmodell und alle enthaltenen Elementen werden im Abschnitt *Vorgehensweise* beschrieben.

Bedingte Zufallsfelder

Wir betrachten das Problem der Bewertung der Qualität von Softwaresystemen als ein Knotenbewertungsproblem in einem ungerichteten Graphen, dem sogenannten *Nachbarschaftsgraph*. Dabei werden die Modellklassen der bedingten Zufallsfelder (CRFs) [7] eingesetzt. Hier sind die Knoten Software-Module, wobei zwei Knoten durch eine Kante verbunden werden, wenn es eine unmittelbare semantische Abhängigkeit zwischen den beiden Modulen gibt. CRFs sind bedingte Modelle, die die Wahrscheinlichkeit für eine Knotenbewertung angeben, unter Einbeziehung der Beobachtungen und des Wissens über das zu untersuchende Problem. Im vorliegenden Fall werden die Beobachtungen durch die agentenbasierte Simulation festgelegt. CRFs sind Maximum-Entropie-Modelle. Das bestimmende Merkmal einer CRF besteht in einem einzelnen exponentiellen Modell für die Wahrscheinlichkeit für die initiale Knotenbewertung. Lafferty et al. [7] waren die ersten, die bedingte Zufallsfelder für das Problem der Bewertung von Sequenzierungsdaten in der Bioinformatik verwendet haben. Li et al. [8] haben den den CRFs zugrundeliegenden Nachbarschaftsgraphen verwendet, um Protein-Protein-Interaktionen vorherzusagen. Dong et al. [9] verwendeten die von ihnen paarweise bedingte Zufallsfelder (pCRFs) genannten Methoden für dasselbe Problem, in dem sie eine Heuristik entwickelten, die eine Kennzeichnung der maximalen a posteriori Wahrscheinlichkeit ermöglicht. Paarweise bedingte Zufallsfeldern berücksichtigen Knoten und Kanten nur in Form von Cliques, während allgemeine

CRFs von jeder Clique im Nachbarschaftsgraphen abhängen. Unseres Wissens sind wir die ersten, die paarweise CRFs im Rahmen der Qualitätssicherung von Software-Systemen anzuwenden.

Vorgehensweise

Unser allgemeiner Ansatz ist in Abbildung 1 dargestellt. Die Rechtecke stellen die Aktivitäten und die Ellipsen stellen Ein- und Ausgänge der Aktivitäten dar. Wir beginnen mit einer Fragestellung in Bezug auf die Struktur der Software-Entwicklung. Dies können organisatorische Fragen sein wie: „Was ist eine geeignete Anzahl an Tester in dem Team?“ sowie Code-spezifische Fragen wie „Wie entwickelt sich die Projektgröße gemessen in der Anzahl an Dateien, über die Zeit?“. Auf dieser Basis erstellen wir ein für die zu untersuchenden Forschungsfragen geeignetes Modell zur agentenbasierten Simulation. Das bedeutet, dass wir die Einflussfaktoren identifizieren und sie in den Kontext des zu modellierenden Prozesses stellen. Der nächste Schritt beinhaltet die Auswahl der Open-Source-Software-Projekte und die Erhebung der Daten über sie, um genügend Wissen zu erhalten, um das Modell mit Daten zu befüllen. Das Ergebnis dieser Aufgabe dient uns zur Schätzung der Simulationsparameter. Zusammen mit dem angegebenen Simulationsmodell werden die geschätzten Parameter in der Simulation verarbeitet. Auf diese Weise werden die empirischen Daten mit den Simulationsergebnissen verglichen. Im Anschluss daran bewerten wir unsere Ergebnisse und damit auch die auf Basis der Simulation vorgeschlagene Antwort auf die initiale Fragestellung.

Wir verwenden Repast Symphony [10] als Simulations-Framework. Es erlaubt, agentenbasierte Modelle zu erstellen, sie auszuführen und die Simulationsergebnisse mit integrierten Werkzeugen wie R oder Weka zu untersuchen. In Abbildung 3 ist die Repast Symphony Umgebung dargestellt.

Als erstes haben wir mit der Simulation von *Systemwachstumstrends* begonnen. Dabei mussten nur die Elemente Entwickler und Artifact berücksichtigt werden (siehe Abbildung 2). Das Wachstum eines Systems wird dabei in der Anzahl an Dateien gemessen.

Lafferty et al. [7] were the first who utilized conditional random fields to the problem of labeling sequence data in Bioinformatics. Li et al. [8] used CRFs the underlying neighborhood graph structure to predict protein-protein-interaction sites. Dong et al. [9] made what they called pair-wise conditional random fields (pCRFs) algorithmically tractable for the same problem by devising a heuristic that approximates a labeling of maximum posterior probability. Pair-wise conditional random fields take only nodes and edges as cliques into account, while general CRFs depend on any clique of the neighborhood graph. To the best of our knowledge, we are the first who apply pairwise CRFs in the context of software system quality assurance.

Approach

Our overall approach is depicted in Figure 1. The rectangles represent activities and the ellipses represent input and outcomes of the activities. We start with a research question concerning the evolution of software development. This can be organizational questions such as “What is a suitable amount of testers in the team?” as well as code specific questions like “How does the project size in number of files evolve over the time?”. Suitable for the specific research question we then create an agent based simulation model, which is compatible to the research question under inves-

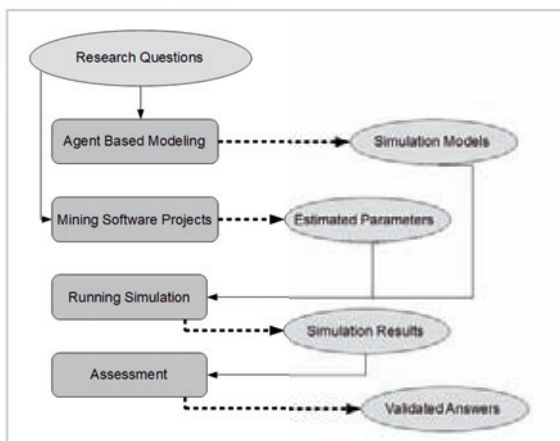


Abbildung 1: Allgemeine Vorgehensweise.

Figure 1: Overall approach.

tigation. This means, that we identify influencing factors and set them into context by the modeling process. The next step contains the selection of open source software projects and the mining of them in order to get enough knowledge to fill the model with information. The output of this task serves us as estimation of adequate simulation parameters. Together with the specified simulation model the estimated parameters are processed by running the simulation. This way, empirical data is compared with simulation results. Following, we validate our retrieved results and thereby the proposed answer to our research question under investigation.

We use Repast Symphony [10] as simulation framework. It allows to define agent-based models, execute them, and evaluate the simulated data with integrated tools like R or Weka. Repast Symphony at runtime is depicted in Figure 3.

As a first investigation, we started with the simulation of *system growth* trends. Therefore, we only considered the entities *Developer* and *Artifact* (see Figure 2). The system growth is measured as file count.

We investigated if it is possible to simulate the growth with the developers commit behavior. To parameterize the simulation model, the data retrieved through the mining process was used. In this simulation, we used a fixed number of average developers with average behavior. This means, that each developer spends the same effort. For the number of files, that will be created, updated, or deleted due to a developers commit, we assumed that the number of file changes follow a geometric distribution. From the developers point of view, a success means that there is nothing to do and zero files are updated. The underlying simulation model is depicted in Figure 2. More details about this approach can be found in one of our publications [11]. As a result, it was possible to reproduce the file growth of an open source project with the system under simulation. As reference project K3b [12] was used, more details about the project and the data obtained through mining the repository can be found in the subsections *System Growth*, *Bugs Lifespan*, and *Developer Collaboration*.

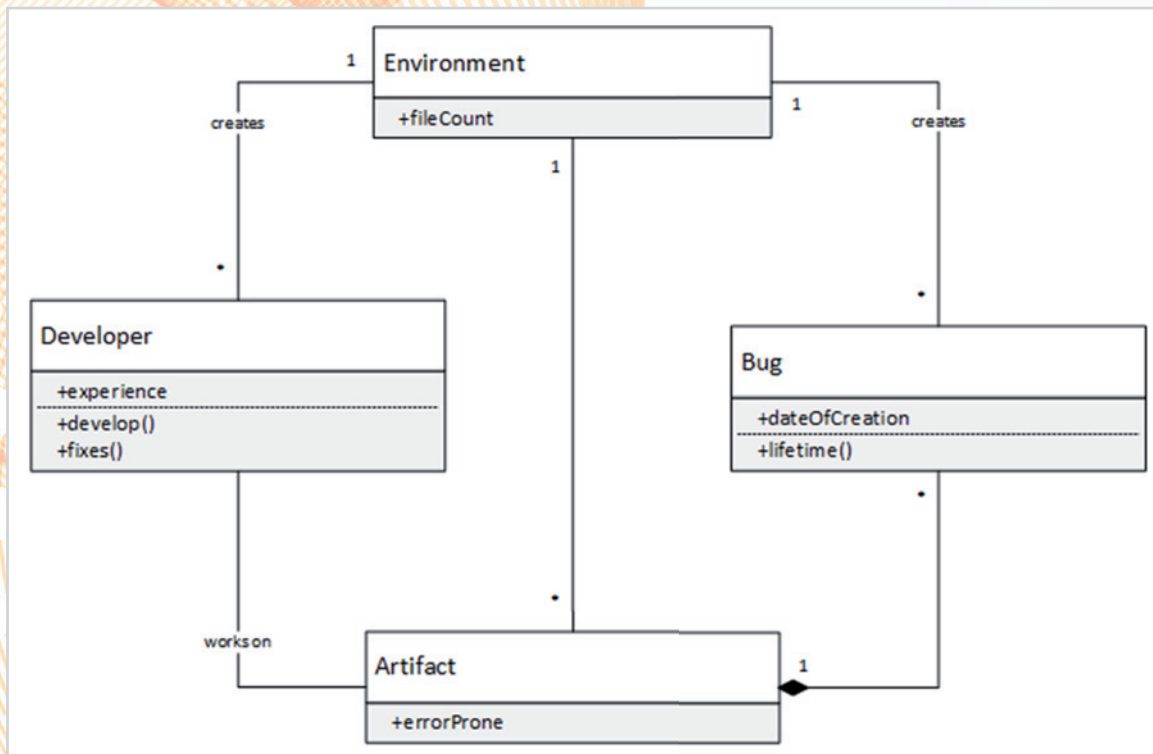


Abbildung 2: Simulationsmodell.

Figure 2: Simulation Model.

Wir untersuchten, ob es möglich ist, das Wachstum auf Basis des Commit-Verhaltens der Entwickler zu simulieren. Um das Simulationsmodell zu parametrieren, wurde die durch den Data-Mining-Prozess gewonnenen Informationen verwendet. In der Simulation verwendeten wir eine feste Durchschnittszahl von Entwicklern mit durchschnittlichem Verhalten. Das bedeutet, dass jeder Entwickler dieselbe Menge an Arbeit investiert. Für die Anzahl der Dateien, auf Basis eines Entwickler-Commits die erstellt, aktualisiert oder gelöscht werden, haben wir angenommen, dass die Anzahl der Dateiänderungen einer geometrischen Verteilung folgt. Aus Sicht eines Entwicklers stellt es einen Erfolg dar, wenn nicht zu tun ist bzw. keine Dateien aktualisiert werden müssen. Das zugrunde liegende Simulationsmodell wird in Abbildung 2 dargestellt. Weitere Infos zu dieser Vorgehensweise können Sie in einem unserer Publikationen [11] finden. Als ein Ergebnis war es mit diesem System möglich, das Wachstum eines

Open-Source-Projektes in der Simulation nachzubilden. Als Referenzprojekt wurde das Programm K3b [12] verwendet; weitere Informationen zu diesem Programm und die durch die Datenerhebung gewonnenen Informationen finden sich in den Unterabschnitten *Systemwachstum*, *Fehlerlebensdauer* und *Entwicklerzusammenarbeit*.

Derzeit arbeiten wir an Modelle mit einer größeren Anzahl an Einheiten, mit spezielleren Einheiten wie verschiedenen Arten von Entwicklern oder anderen Arten von Fehler sowie einer Integration der daraus resultierenden Abhängigkeiten zwischen den Einheiten. Die Abhängigkeiten werden als Netze oder Graphen modelliert und stellen Entwickler-Datei-Netzwerke, Dateikopplungen oder Wechselwirkungen dar. Darüber hinaus sind wir daran interessiert, unsere auf Basis von CRFs entstandenen Simulationsergebnisse weiter zu bewerten. Daher wird einer der Graphen an das CRF übergeben, wo die weitere Beurteilung durchgeführt werden soll.

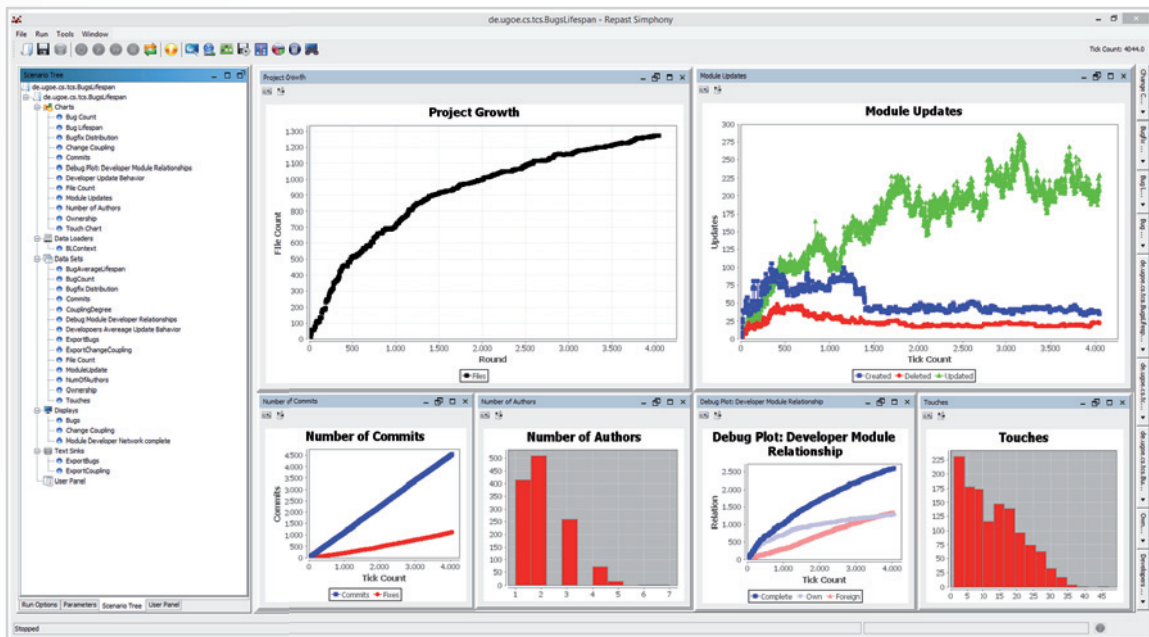


Abbildung 3: Repast Simphony.

Figure 3: Repast Simphony at Runtime.

Currently, we are working on models with more entities, specialized entities such as different types of developers or different kind of bugs, and the integration of sophisticated dependencies between the entities. The dependencies will be modeled as networks or graphs and represent developer-files-networks, file coupling, or change coupling. Furthermore, we are interested in the evaluation of our simulation results by CRFs. Therefore, one of the graphs will be passed to the CRF where the further assessment is performed.

Another point of topical interest is to find a *module-wise classification* rule that classifies one module into *acceptable* and *problematic* ones without knowledge about the neighbor classifications. This module-wise classification will be used for the *quality labeling* done by the CRF.

We give an example for a module-wise classification rule given the simulation model depicted in Figure 2. Additionally to the description of Figure 2, this model contains the entity *bug*. A bug can randomly arise each round and afterwards be assigned to an arbitrary module. Assuming that

the bug is specialized to critical, normal, and minor. As an example a simple rule to classify the module is: If a module contains at least one major bug it is always problematic, if a module does not contain a major bug but more than a determined selected number of normal bugs it is problematic and otherwise it is acceptable.

For building models close to reality, we make use of data retrieved from real projects. Case studies are used to enrich our models with knowledge and for the validation of results by comparing different evolutionary project behavior with our simulation.

We studied a first open source project to show the feasibility of our approach. Therefore, we selected three specific questions concerning software evolution:

- Q1: How does the project size in number of files evolve over the time?
- Q2: What kinds of bugs live how long and how are their lifespans distributed?
- Q3: How active are the developers at different points of time and how balanced is the work?

Ein weiterer Untersuchungsaspekt besteht darin, Klassifikationsregel für die Module aufzustellen, die eine Unterteilung in *akzeptable* und *problematische* Module ermöglicht, ohne dass dafür Wissen über die Nachbarklassifikationen notwendig wäre. Diese modulweise Klassifizierung wird für die *Qualitätskennzeichnung* durch die CRF verwendet.

Im Folgenden stellen wir für das in Abbildung 2 dargestellte Simulationsmodell ein Beispiel für eine modulweise Klassifikationsregel vor. Zusätzlich zu der in Abbildung 2 dargestellten Beschreibung, enthält das Modell die Einheit *Fehler*. Ein Fehler kann zufällig in jeder Runde auftreten und anschließend einem bestimmten Modul zugeordnet werden. Im Folgenden wird angenommen, dass jeder Fehler in eine der Kategorien *kritisch*, *normal*, *geringfügig* fällt. Eine mögliche Klassifikationsregel für ein Modul wäre dann: Ein Modul wird genau dann als *problematisch* eingestuft, wenn es entweder einen *kritischen Fehler* enthält oder aber eine zu große Anzahl an *normalen Fehlern*; andernfalls wird es als *akzeptabel* angesehen.

Um realitätsnahe Modelle zu erstellen, verwenden wir Daten, die wir aus den realen Projekten erhoben haben. Um das Projekt mit weiterem Wissen anzureichern und für die Bewertung der Ergebnisse bzw. dem Vergleich verschiedener Entwicklungsprojekte mit den Simulationsergebnissen, wurden weitere Fallstudien herangezogen. Um die Durchführbarkeit unseres Ansatzes zu belegen, haben wir ein erstes Open-Source-Projekt untersucht. Dabei haben wir drei spezielle Fragen zum Thema der Software Evolution betrachtet:

- Q1: Wie entwickelt sich die Projektgröße gemessen in der Anzahl der Dateien über die Zeit?
- Q2: Welche Arten von Fehlern existierten wie lange und wie ist ihre Lebensdauer verteilt?
- Q3: Wie aktiv waren die Entwickler zu den einzelnen Zeitpunkten und wie ausgewogen ist diese Arbeit verteilt?

Q1: Systemwachstum

Um Q1 zu analysieren, betrachten wir den Wachstumstrends des Systems. Das Wachstum der

Anzahl an Quellcode-Modulen ist üblicherweise sublinear mit einer deutlichen Abnahme hin zum Ende der Entwicklungszeit. Wir beobachteten ähnliche Ergebnisse für K3b [12] und verwenden diese, um ein statistisches Modell für das Wachstum auf Basis der Erstellung, Änderungen und Löschungen von Dateien durch die Entwickler zu erstellen. Da der betrachtete Zeitraum 4044 Tage umfasste, wurden insgesamt 4044 Runden simuliert, wobei eine Runde jeweils einen Tag darstellte. In dieser Simulation verwendeten wir einen durchschnittlichen Entwickler mit einem durchschnittlichen Verhalten. Das bedeutet, dass jeder Entwickler dieselbe Menge an Arbeit in das Projekt investiert. Für die Anzahl der Dateien, die auf Basis eines Entwickler-Commits erstellt, aktualisiert oder gelöscht werden, nehmen wir an, dass die Anzahl der Dateiänderungen einer geometrischen Verteilung folgen. Wir verwenden dabei die Darstellungsform der geometrischen Verteilung, in der die Anzahl an Fehlversuchen vor dem ersten Erfolg gezählt wird, d.h. die Null zulässig ist: $P(action) = (1-p)^{(k-1)}$. Aus Entwicklersicht liegt ein Erfolg vor, wenn er nichts unternehmen muss und keine Dateien zu aktualisieren sind. Die Wahrscheinlichkeit für Änderungen nimmt mit steigender Systemgröße ab. Beim Start der Simulation wird die Wahrscheinlichkeit für Dateiänderungen mit Hilfe von statistisch erhobenen Daten initialisiert. Die Werte für die Anzahl an Datei-erstellungen, Löschungen und Aktualisierungen werden aus der Datenerhebung entnommen und erlauben das Wachstum von K3b zu beschreiben. Abbildung 4 zeigt dieses empirische Wachstum von K3b und in Abbildung 5 wird das Wachstum, das sich aus der Simulation ergibt, dargestellt. Der Vergleich der beiden Graphen zeigt, dass die Simulation das tatsächliche Wachstum von K3b gut nachmodelliert.

Derzeit arbeiten wir an der Bestimmung von Phasen des Software-Entwicklungsprozesses, die jeweils einen bestimmten Einfluss auf das Systemwachstum haben. Die meisten Software-Projekte weist mindestens vier Phasen auf: eine Startphase, eine Hauptaktivitätsphase, eine Supportphase und eine Auslaufphase; innerhalb dieser Phasen existieren möglicherweise Schleifen. Die Verteilung der Art der Änderungen an dem Projekt kann sich dabei von Phase zu Phase unterscheiden.

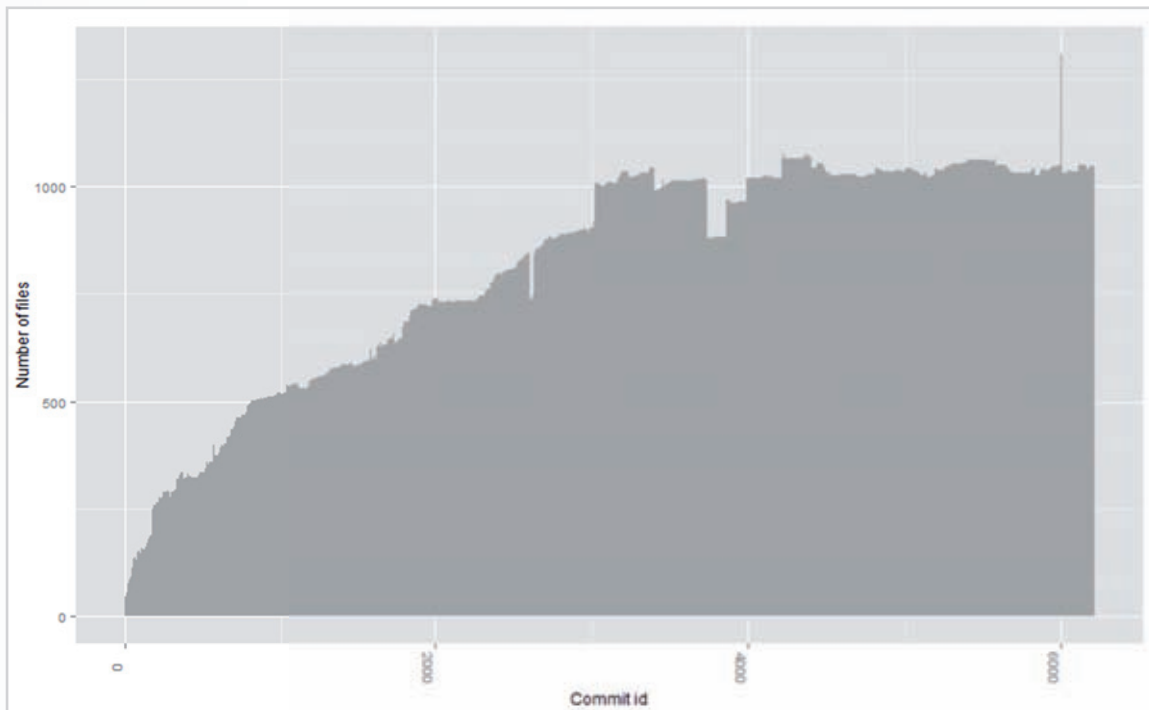


Abbildung 4: Empirisches Systemwachstum von K3b

Figure 4: Empirical System Growth of K3b

Q1: System Growth

For answering Q1, we analyzed system growth trends. The growth in number of source modules usually is sub-linear with a decrease at the end of the development. We observed similar results for K3b [12] and used this to build a statistical model for the growth based on creations, updates and deletions by developers. Since the mined projects lifespan took 4044 days, the simulation runs 4044 rounds whereby one round represents one day. In this simulation we are using the average developer with average behavior. This means, that each developer spend the same effort. For the number of files, that will be created, updated, or deleted due to a developers commit, we assume that the number of file changes follow a geometric distribution. We used the version of the geometric distribution where the number of failures before the first success is counted, i.e., zero is allowed. Therefore, we have $P(\text{action}) = (1-p)^{k-1}$. From developers point of view, a success means that

there is nothing to do and zero files are updated. The likelihood of file changes decreases with the increasing system growth. When starting the simulation the likelihood of file changes is also

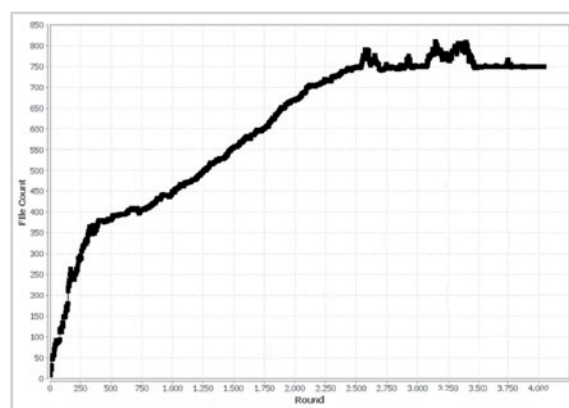


Abbildung 5: Simuliertes Systemwachstum

Figure 5: Simulated System Growth

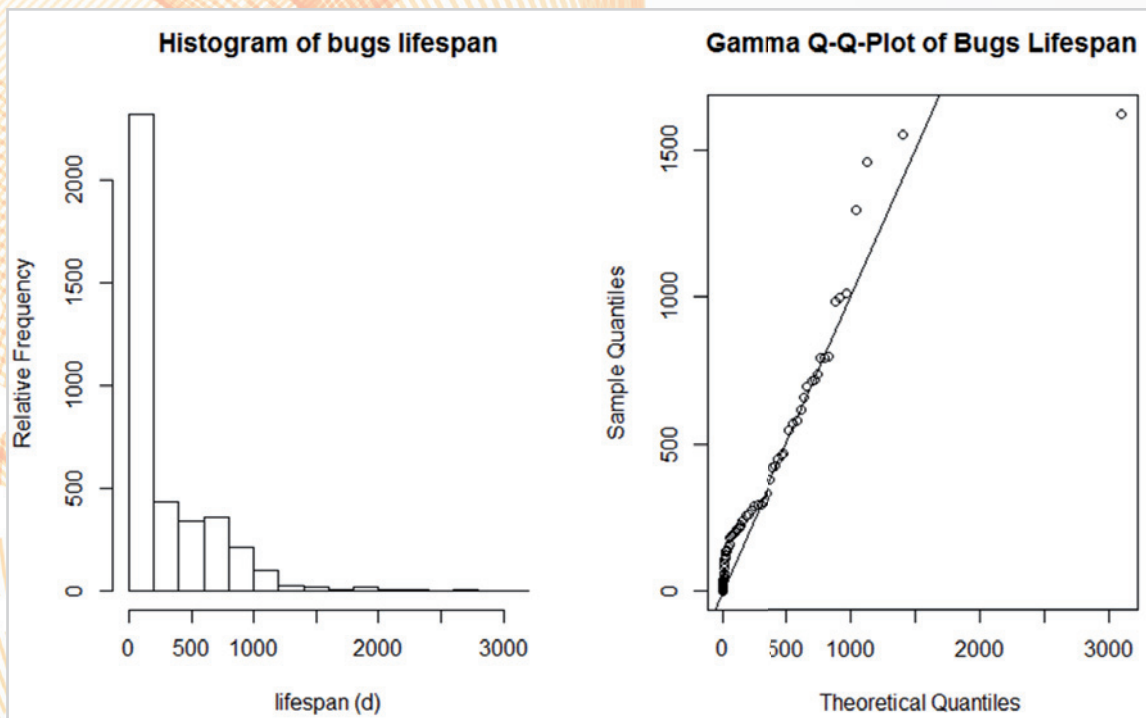


Abbildung 6: Lebensdauer von Fehler in K3b. Quantil-Quantil-Plot für die Gamma-Verteilung der Fehler.

Figure 6: Lifespan of bugs in K3b. Quantile-quantile plot for Gamma distribution of bugs.

Q2: Fehlerlebensdauer

Um Q2 zu beantworten, untersuchten wir die Lebensdauerverteilung von Fehlern und ihre Dichte. Wir definieren die Lebensdauer eines Fehlers als die Zeit zwischen dem Fehlerbericht und dem letzten Zeitpunkt, an dem es Änderungen in Bezug auf diesen Fehler gab. Auf dieser Basis haben wir Statistiken zu den verschiedenen Arten von Fehlern erhoben. Wir beobachteten, dass kritischerere Fehler in der Regel früher befestigt werden. Die Verteilung aller Fehler ist in Abbildung 6 zu sehen. Auf Basis des Histogramms haben wir angenommen, dass die Fehlerlebensdauer näherungsweise Gammaverteilt ist und haben einen Vergleich auf Basis eines Quantil-Quantil-Plot vorgenommen. In dem Simulationsmodell verwenden wir daher die Gamma-Verteilung zur Modellierung der Lebensdauern der Fehler.

Q3: Entwicklerzusammenarbeit

Zur Untersuchung von Q3, interessieren wir uns für das Wie und Warum der Entwicklertätigkeit und die Zusammenarbeit der Entwickler. Dafür wurden Entwickler-Datei-Netze (ohne Dateiabhängigkeiten) für jedes Jahr der Entwicklung aus vorhandenen Protokolldateien extrahiert und die Ergebnisse mit Gephi [13] visualisiert. Die Entwicklung ist in Abbildung 7 dargestellt.

Die Graphen zeigen, dass es einen Hauptentwickler gibt, der auch zugleich der Projektinitiator ist. In den Grafiken ist er stets durch den rot eingefärbten Knoten auf der rechten Seite des Netzes dargestellt. Im Laufe der Jahre erlangte er immer mehr Einfluss und erreichte 2006 einen Spitzenwert. Danach wurde die Arbeit ausgewogener und ein anderer Entwickler (in Blau dargestellt) erschien und übernahm schließlich die zentrale Rolle in dem Projekt als Betreuer. In Bezug dazu

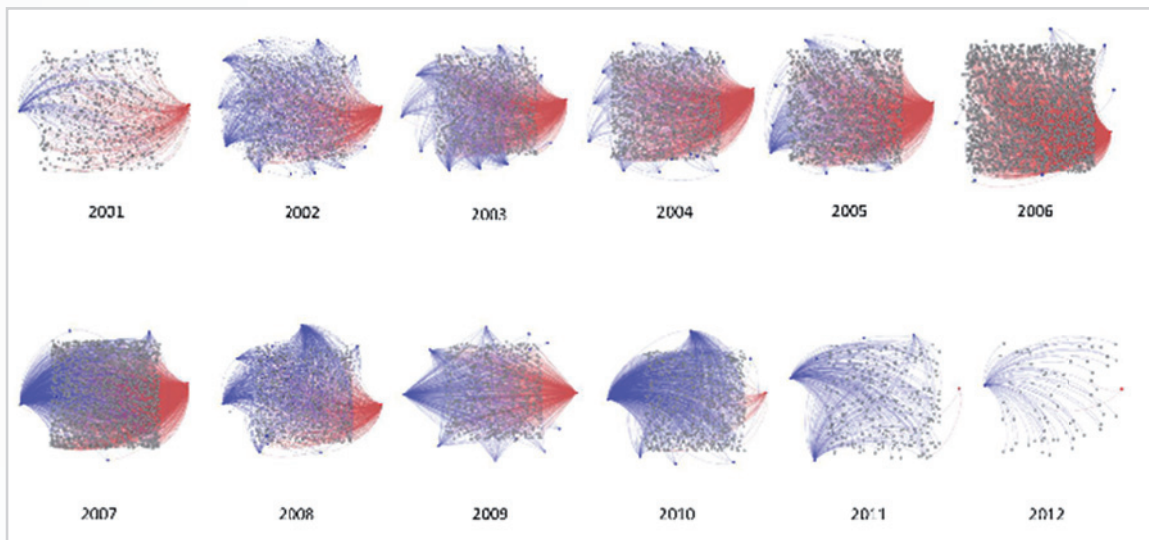


Abbildung 7: Veränderung Entwickler-Dateien-Netzwerke in K3b über die Jahre.

Figure 7: Yearly developer-file networks of K3b.

be initialized with mined parameters. From our data mining, we inferred values for file creation, file deletion, and updates as suitable to describe the growth of K3b. In Figure 4, the empirical file growth of K3b is shown and in Figure 5 the file growth produced by our simulation is pictured. Comparing these one can see that our simulation is able to reproduce the actual file growth of K3b.

We are currently working on defining phases of the software development process influencing the system growth. Most software projects exhibits at least four phases, an initial, changing phase and maintenance phase and phase-out, which may have loops. During these phases the distribution of the kind of changes change as well.

Q2: Bugs Lifespan

To answer Q2, we investigated the lifetime of bugs and their density. We declare a bugs lifetime as the time between the date it was reported and the date it was last modified. From this we gained the statistics for the distinct kinds of bugs. We observe that more critical bugs tend to be fixed sooner. The distribution of all bugs is displayed in

Figure 6. From the histogram, we assume that the lifespan of bugs follow a gamma distribution and compared it by a quantile-quantile-plot. In the simulation model we use this as lifetime attribute for a bug spread by the environment.

Q3: Developer Collaboration

For the investigation of Q3, we are interested in the how and why of the developer activity and collaboration evolution. For this, we created developer-files-networks (without file dependencies) for each year of development extracted from commit log files and visualized the results with Gephi [13]. The evolution of the graph is depicted in Figure 7.

These graphs show that there is one main contributor who is the project creator. In the graphic he poses always the red node on the right side of the network. Over the years he gained more and more influence peaked in 2006. Afterwards the work gets more balanced and another (blue) developer arises and finally inherits the central status in the project as its maintainer. According to this, we also calculate several social network

haben wir auch einige soziale Netzwerk-Metriken berechnet und haben ihre Entwicklung über den Verlauf des Projektes verfolgt. Die Zusammenhangskomponente des Netzes war dabei von besonderem Interesse. In unserem Kontext gibt uns der Zusammenhangskomponente einen Hinweis darauf, wie ausgewogen die Arbeit in dem jeweiligen Jahr verteilt wurde. In 2006 und 2012 war dieser Faktor sehr gering, was bedeutet, dass das Netzwerk in mehrere Cluster hätte aufgeteilt werden können. In diesen beiden Jahren hing die gesamte Arbeit jeweils im Wesentlichen von einem einzigen Entwickler ab. Das am besten ausgewogene Netz im Jahr 2002 hatte hingegen eine starke Zusammenhangskomponente zur Folge.

Ausblick

In unserer initialen Fallstudie konnten wir die generelle Einsatzfähigkeit unseres Ansatzes zur Erhebung von statistischen Daten aus Open-Source-Archiven und deren Verwendung als Eingangsdaten für die Simulation nachweisen. Im Rahmen dieser Studie haben wir ein Simulationsmodell zur Beantwortung von Q1-Q3 entworfen. Wir haben das Wachstum des Projektes simuliert und erfolgreich mit den erhobenen empirischen Daten verglichen (Q1). Darüber hinaus konnten im Rahmen von Q2 Verteilungen zur Modellierung der Fehlerlebensdauer bestimmt werden und die Ergebnisse von Q3 dienen als Grundlage für

Projektdate

Das Projekt wird seit Mai 2013 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 Honsel
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

metrics and trace their evolution over time. Of special interest was the modularity of the networks for us. In our context the modularity factor gives us a hint on how balanced the work in this year was, because a low factor as we derived in 2006 and 2012, means that the network cannot be modularized in clusters. There the work depends too much on a certain developer. The most balanced network in 2002 results the best modularity factor.

Outlook

With the initial case study, we showed the feasibility of our approach in mining open source repositories as simulation input. Within this study, we built a simulation model aimed to answer Q1-Q3. We simulated the system growth of a project and successfully compared this with the empirical behavior (Q1). Furthermore, the investigations of Q2 serves as lifespan attribute of bugs in the model and the results of Q3 serves as a foundation for modeling networks in the current progress of our work.

Most parts of the results can be found in the corresponding publication [11]. In addition, we are working on an example case study for preparing the input data for the labeling by CRFs. For comparison and validation more case studies are planned presently.

We exposed sources and tools available for the knowledge extraction in preparation of simulating software processes. In previous investigations (Q1-Q3), the need for a more fine-grained model of developer behavior exposed. To respect experience, communication among developers and contribution activities, we plan to model developers using Hidden Markov Models (HMMs). The underlying idea is presented in [14].

Next steps are the sophistication of dependency graphs and the preparation of input data for the CRF. Furthermore, we plan to cover more research questions like the simulation of possible software architecture degeneration. Another planned task is to use the CRF as feedback loop for the simulation as an automated process.

Finally, we are interested in the comparison of our approach to other ones. The main part of this work will be covered in 2015.

Literatur // References

- [1] M. M. Lehman. Programs, life cycles, and laws of software evolution. Proceedings of the IEEE, vol. 68, no. 9, Sep. 1980, pp. 1060 – 1076.
- [2] The PROMISE Repository of Software Engineering Databases (2005) by Sayyad, T. J. Menzies
- [3] G. Robles, S. Koch. Remote analysis and measurement of libre software systems by means of the CVSanaly tool. Proceedings of the 2nd ICSE Workshop on Remote Analysis and Measurement of Software Systems (RAMSS), 2004
- [4] RStudio (2012). RStudio: Integrated development environment for R [Computer software]. Boston, MA. Retrieved March 15, 2015. Available from <http://www.rstudio.org/>
- [5] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.
- [6] C.M. Macal, Michael J. North. Tutorial on agent-based modeling and simulation. Proceedings of the 37th Winter Simulation conference, 2005, pp. 2-15.
- [7] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proceedings of the 18th International conference on Machine Learning, 2001, pp. 282-289.
- [8] Ming-Hui Li, Lei Lin, Xiao-Long Wang, and Tao Liu. Protein-protein interaction site prediction based on conditional random fields. In: Bioinformatics 23.5 (2007), pp. 597-604.
- [9] Zhijie Dong, Keyu Wang, Truong Khanh Linh Dang, Mehmet Gültas, Marlon Welter, Torsten Wierschin, Mario Stanke, and Stephan Waack. CRF-based models of protein surfaces improve protein-protein interaction site predictions. In: BMC Bioinformatics 15 (2014), p. 277.

die aktuell in Arbeit befindliche Modellierung des Netzwerks.

Ein Großteil der Ergebnisse ist in [11] publiziert. Darüber hinaus arbeiten wir derzeit an einer Beispielfallstudie, um die Eingangsdaten für die Bewertung durch die CRFs vorzubereiten. Um die Ergebnisse zu bewerten und zu vergleichen, sind mehrere weitere Fallstudien geplant.

Zur Vorbereitung der Simulation von Software-Prozessen haben wir dabei verschiedene Methoden und Werkzeuge zur Wissensextraktion betrachtet. In früheren Untersuchungen (Q1-Q3), zeigte sich die Notwendigkeit einer feinkörnigeren Modellierung des Entwicklerverhaltens. Um die Kommunikation zwischen den Entwicklern und ihre Aktivitäten abbilden zu können, denken wir derzeit

über den Einsatz von Hidden Markov Modellen (HMM) nach. Die dafür zugrunde liegende Idee wird in [14] vorgestellt.

Die nächsten Schritte bestehen in einer Weiterentwicklung des Abhängigkeitsgraphen und in der Vorbereitung der Daten für die CRF. Des Weiteren planen wir weitere Forschungsfragen wie die Simulation von Software-Architektur Veränderungen zu behandeln. Als eine weitere Aufgabe ist vorgesehen, den CRF als Rückkopplungsschleife für die Simulation als ein automatisiertes Verfahren zu verwenden.

Schließlich sind wir daran interessiert, unseren Ansatz mit anderen Vorgehensweisen zu vergleichen. Dies ist im Wesentlichen für das Jahr 2015 vorgesehen.

- [10] Michael J. North, Nicholson T. Collier, Jonathan Ozik, Eric R. Tataro, Charles M. Macal, Mark Bragen, and Pam Sydelko. Complex adaptive systems modeling with Repast Symphony. In: Complex Adaptive Systems Modeling 1.1 (2013), pp. 1-26.
- [11] Verena Honsel, Daniel Honsel, and Jens Grabowski. Software Process Simulation Based on Mining Software Repositories. Proceedings of the Third International Workshop on Software Mining, 2014.
- [12] Sebastian Trueg. K3b-The CD/DVD Kreator for Linux. online. 2011. URL: <http://www.k3b.org/> (visited on 15/03/2015).
- [13] Bastian M., Heymann S., Jacomy M. (2009). Gephi: an open source software for exploring and manipulating networks. International AAAI Conference on Weblogs and Social Media. 2009
- [14] Verena 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 , May 2015

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

Jürgen Dix, Jörg P. Müller

Desim steht für dezentrale Architekturen und Konzepte für die Simulation von Systemen von Systemen. Das Projekt, das zum Forschungsfeld „Verteilte simulation“ des SWZ gehört, stellt eine Zusammenarbeit der beiden Arbeitsgruppen von Prof. Dr. Jürgen Dix und Prof. Dr. Jörg Müller beide vom Institut für Informatik der TU Clausthal dar. Neben Tobias Ahlbrecht (B.Sc.) und Dipl.-Inf. Philipp Kraus aus den jeweiligen Arbeitsgruppen sind die Studenten Niklas Fiekas, Christoph Kuper und Jan Philipp Haltenhof in das Projekt eingebunden und unterstützen es.

Systeme of Systems (SoS) sind Systeme, die wiederum aus mehreren komplexen und heterogenen Systemen bestehen, die auf verschiedene Arten lose gekoppelt sind und miteinander interagieren. Das Verhalten eines solchen Systems kann zwar oft beobachtet werden, aber es ist schwierig, Vorhersagen über sein Verhalten zu treffen. Um das Zusammenspiel der verschiedenen Komponenten und das Verhalten des komplexen Ganzen zu analysieren, ist die Simulation eine sich in natürlicher Weise anbietende praktikable Lösung.

Während sich herkömmliche Simulationskonzepte (d.h. Modelle, Methoden und Plattformen) ungeeignet für die Aufgabe erwiesen haben, erscheint das Gebiet der Multiagentensysteme (MAS) vielversprechend. MAS ermöglichen es, komplexes individuelles Verhalten der kleinsten Einheiten in einer Simulation, der sogenannten Agenten, zu definieren. Bei einer Verkehrssimulation werden in einem konventionellen Ansatz zum Beispiel der Durchsatz und die globalen Eigenschaften der Straßen in Form von Differentialgleichungen modelliert. Dieser Ansatz führt zu effizient lösaren Gleichungen, die aber nur groben Überblick, genannt *makroskopische Sichtweise*, ermöglichen. Um in das Szenario

auf die Ebene eines einzelnen Fahrers hinein zu zoomen (um zu erkennen, welche Ursachen zu dem globalen Systemverhalten geführt haben), ist es notwendig jeden Fahrer individuell zu modellieren und zu simulieren. Diese *mikroskopische Sichtweise* ermöglicht es, den Verlauf der Simulation sehr viel detaillierter analysieren.

Da leicht abstrahierbare Konzepte wie die wissensbasierte Entscheidungsfindung, Planung und Planausführung, sowie die Zusammenarbeit und Kommunikation zwischen autonomen Software-Komponenten zur Verfügung stehen, ist MAS eine attraktive Wahl für diese Art von Fragestellungen. Selbstverständlich bringt dieser Ansatz auch eine Reihe von Anforderungen mit sich. Alle einzelnen Simulationsobjekte müssen konzipiert und auf die effizienteste Art und Weise durchgeführt werden. Die gleichzeitige Abbildung einer großen Anzahl von Software-Agenten kann auch leistungsfähige Systeme an ihre Grenzen zu führen. Somit ist es wichtig, eine Simulationsplattform von Anfang an auf eine möglichst gute Skalierbarkeit hin zu konstruieren. Andere Probleme, wie eine effiziente Speicherverwaltung und der generelle Entwurf der Simulation, müssen ebenfalls berücksichtigt werden.

Momentan existiert eine Reihe von verschiedenen Modellierungsansätzen für MAS. Dies ist ein Grund dafür, warum bisher keine einheitliche hochskalierbare und umfassende Multi-Agenten-Simulationsplattform existiert. Da darüber hinaus die meisten Plattformen auf die Java basieren, ist es schwierig wenn nicht zum Teil unmöglich, eine effiziente Parallelisierung zu erreichen. Andere Plattformen wiederum konzentrieren sich auf die visuelle Modellierung von kleinen Szenarien oder sind ausschließlich auf eine bestimmte Aufgabenstellung spezialisiert.

DeSim: Distributed Architectures and Concepts for the Simulation of Systems of Systems

Jürgen Dix, Jörg P. Müller

DeSim stands for decentralized architectures and concepts for the simulation of Systems of Systems. The project, itself part of the distributed simulation division of the SWZ, is a cooperation between the groups of Prof. Dr. Jürgen Dix and Prof. Dr. Jörg P. Müller from the Institute for Informatics Clausthal University of Technology. In addition Tobias Ahlbrecht (B.Sc.) and Dipl.-Inf. Philipp Kraus from the respective groups are involved in the project and are assisted by our students Niklas Fiekas, Christoph Kuper and Jan Philipp Haltenhof.

Systems of Systems (SoS) are systems which in turn consist of multiple complex and heterogeneous systems that interact in one way or the other, which makes them loosely coupled. Thus, emergent behavior can often be observed, whereas it is rather difficult to predict. To analyse the interplay between the various components and the behavior of the complex whole, simulation is naturally a viable solution.

While traditional simulation concepts (i.e. models, methodologies and platforms) proved to be unfit for the task at hand, the field of Multi-agent Systems (MAS) appeared promising. MAS allow for the specification of complex individual behavior of the smallest entities, called agents, in a simulation. Considering a traffic simulation for example, conventional methods usually model the throughput and global properties of roads with differential equations. While efficiently computable, this only gives a bird's eye view, we will call it *macro view*, of the simulation. In order to zoom into the scenario on the level of a single driver (to see what may have caused the global system behavior), it is necessary to model and run each driver component individually. This *micro view* allows to analyse

the course of the simulation in much more detail. What makes MAS an attractive choice is the availability of readily available abstractions of concepts like knowledge-based decision making, planning and plan execution, as well as cooperation and communication between the autonomous software components.

Of course, this advantage comes at a cost. All of the individual simulation objects have to be designed and executed in the most efficient fashion. Running a vast number of software agents concurrently can bring even the most powerful system to its limits. Thus, it is important to create a simulation platform designed for scalability from the very beginning. Other problems, like efficient memory management and simulation design, have to be considered as well.

The state of the art in MAS reveals many and more different modelling approaches, which is one reason why no sophisticated and highly-scalable multi-agent simulation platform does yet exist. Also, most platforms are programmed in Java, which makes it really difficult, if not impossible, to account for efficient parallelization. Other platforms focus on visual modelling of small-scale scenarios or are too specialized and enforce a certain domain-specific application.

Often, the metamodel, which is the theoretical foundation of each simulation model, is hard-wired to the platform and cannot easily be extended, altered or even exchanged. On the other hand, agent-based concepts are not available or even too restrictive. Also, some platforms are not open enough, which makes it difficult, if not impossible, to extend those or integrate them with other runtime platforms.

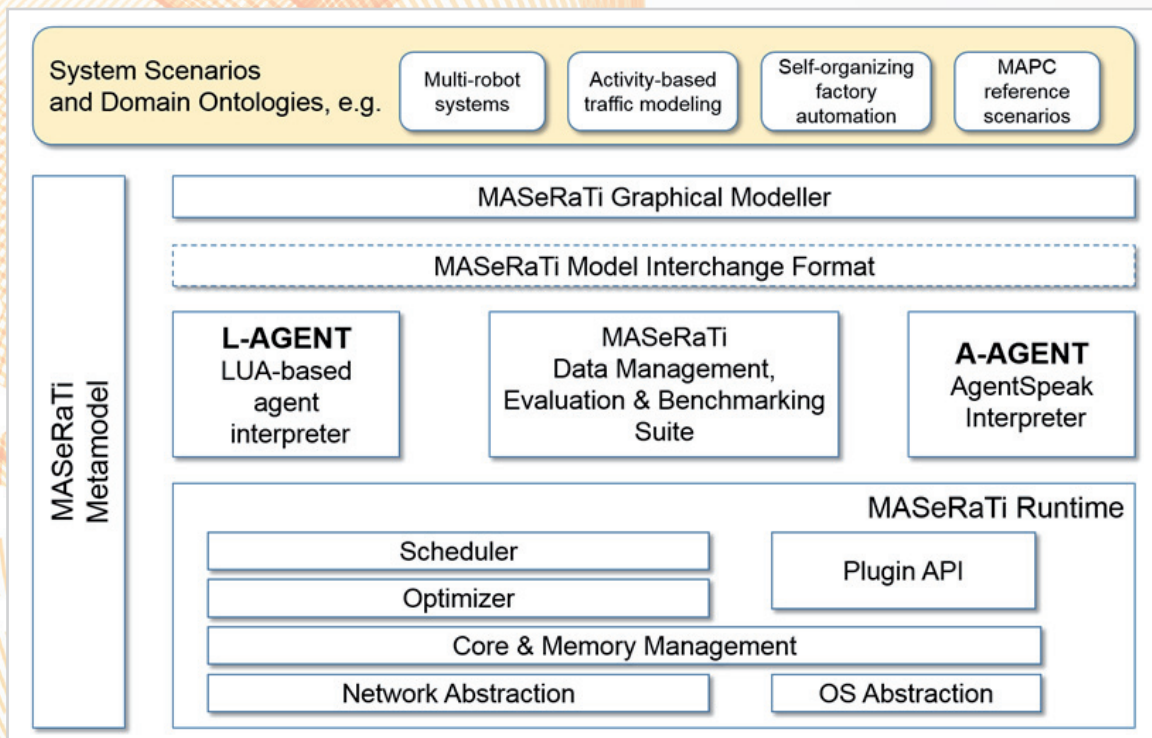


Abbildung 1: MASeRaTi Architektur

Figure 1: MASeRaTi Architecture

Offt ist das Metamodell, das die theoretische Grundlage der einzelnen Simulationsmodell darstellt, fest verdrahtet und kann nicht auf einfache Weise erweitert, verändert oder auch ausgetauscht werden. Auf der anderen Seite sind agentenbasierte Konzepte nicht verfügbar oder zu restriktiv. Auch sind einige Plattformen nicht offen genug, wodurch es schwierig wenn nicht sogar unmöglich ist, diese zu erweitern oder sie in andere Laufzeit Plattformen zu integrieren.

Daher sind die betrachteten Fragestellungen nicht zufriedenstellend durch die verfügbaren Plattformen behandelbar. Unser Ansatz besteht aus einer neuen Multi-Agenten-basierten Laufzeitsimulationsplattform, MASeRaTi, die auf eine hohe Skalierbarkeit bei gleichzeitig einem niedrigen Komplexitätsgrad in der Anwendung abzielt. Diese Plattform kombiniert die klassischen Simulationswerkzeuge (*makroskopische Ansicht*) mit

individuellen Agenten-basierten Simulationen (*mikroskopische Ansicht*). Die Plattform sollte in der Lage sein, große Szenarien effizient parallelisiert sowohl auf Desktop-Rechnern als auch auf Hochleistungs-Computing-Clustern zu simulieren. Die Idee zu MASeRaTi entstand aus zwei voneinander unabhängigen Projekten: *Planets* und *MASSim*, die jeweils auf individuelle Fragestellungen abzielen. In beiden Projekten wurden separate Laufzeitumgebungen entwickelt; eins für die Simulation im städtischen Verkehrsmanagement und eins für die Simulation und den Vergleich verschiedener Agentensysteme. Das erste, *ATSim* genannt, stellt eine Simulationsarchitektur dar, die das kommerzielle Verkehrssimulationsframework *AIMSuN* mit dem Java Agent Development Framework (*JADE*) integriert. Die zweite Plattform, *MASSim*, wurde in Java vor allem als Grundlage für den jährlichen Multi-Agenten Programmierwettbewerb, einem Wettbewerb mit dem Ziel,

Therefore our problems are not satisfactorily addressed by available platforms. Our approach comprises a new multi-agent based runtime simulation platform, *MASeRaTi*, which aims at high scalability while maintaining the lowest possible level of difficulty for the user.

This platform combines the conventional legacy simulation tools (the *macro view*) with individual based agent simulation (the *micro view*). It should be able to efficiently run large scenarios in parallel on standard desktop computers as well as on high-performance computing clusters.

The idea for *MASeRaTi* arose out of two independent projects, *Planets* and *MASSim*, each with its own individual goals. In both projects, separate running systems were developed; one for the simulation of urban traffic management and one for the simulation and comparison of arbitrary agent systems. The former, called *ATSim*, is a simulation architecture integrating the commercial traffic simulation framework *AIMSuN* with the Java Agent Development Framework (*JADE*). The latter platform, *MASSim*, was implemented in Java mainly as a foundation for the annual Multi-Agent Programming Contest, a competition with the goal to evaluate different agent development frameworks and techniques.

MASeRaTi

As elaborated above, the most serious issues of current systems are scalability, memory management, communication and modelling.

To solve the problem of scalability, the question to answer is how to run a huge number of agents (up to millions) efficiently and concurrently. Each agent has its own reasoning cycle, in which it performs many different things, like updating what it knows about its environment, checking and possibly selecting its current goals, devising plans of how to achieve them and of course deciding which plan to follow first. All of these agent cycles have to be executed in each simulation cycle or step, which requires sophisticated scheduling mechanisms. Normally, in current agent systems, each agent has its own command thread. This can become a problem, if the number of agents

constantly goes up, as even the operating system places a rough limit of 5,000 on the number of threads (and thus agents). Regardless, a large number of threads means a significant overhead that has to be processed, forming a great performance issue. As a counter-measure, *MASeRaTi* employs a Micro-Kernel with its own scheduling mechanism, which allows for optimized scheduling which is even improved during the runtime of the simulation to further increase the system's speed and enhance the overall behavior of the platform.

The aspect of communication between agents often forms a serious performance bottleneck. It is usually realized with dedicated and partly cumbersome communication protocols and languages, which aren't necessarily optimized for speed. In *MASeRaTi*, we try to overcome the communication problem using the synchronization functions of the well-known and established Message Passing Interface (*MPI*) for parallel computing.

Agent systems have in common, that all of the agents are potentially active at any given time, while requiring access to any available information to execute their reasoning cycle. This means, it is very difficult to assess in advance which agent needs access to which information at which time. In huge scenarios, this information cannot possibly be kept in the main memory completely and transferring data from and to (hard) disk is a comparatively slow task. Thus, we need an efficient caching mechanism, which, as the scheduling mechanism, optimizes itself during runtime. While all of these issues concern the runtime of a simulation, we should not forget the design-time for it. For this, we need an efficient modelling framework comprising techniques and methodologies, a reliable and time-tested metamodel as well as accompanying tool support.

All of the agents and their environment, as well as the aim of the simulation, i.e. the desired results, have to be specified individually. Without suitable guidelines this process might turn out to be too difficult or even impossible with ever-growing scenarios.

One cannot expect a domain expert to devise the whole scenario and transform it into executable

verschiedene Agenten-Entwicklungs-umgebungen und -Techniken zu bewerten, implementiert.

MASeRaTi

Wie oben ausgeführt stellen die Skalierbarkeit, die Speicherverwaltung, die Kommunikation und die Modellierung die größten Herausforderungen für die aktuellen Systeme dar.

Um das Problem der Skalierbarkeit zu lösen, ist die Frage zu beantworten, wie eine große Anzahl an Agenten (bis zu mehreren Millionen) effizient parallelisiert ausgeführt werden können.

Jeder Agent hat seine eigene Denkansätze, auf deren Basis er verschiedene Handlungen ausführt, wie die Aktualisierung der Informationen über seine Umgebung, die Überprüfen und eventuell die Auswahl der aktuellen Ziele, sowie die Pläne zu deren Erreichung und natürlich die Entscheidung, welche Handlung dafür als nächstes auszuführen ist. All diese Agenten-Arbeitsschritte müssen in jedem Simulationszyklus bzw. Simulationsschritt durchgeführt werden, was komplexe Methoden der Zeitplanung erfordert.

In den aktuellen Agentensystemen besitzt üblicherweise jeder Agent seinen eigenen Befehls-Thread. Dies kann zu Problemen führen, wenn die Anzahl an Agenten ständig steigt, da auch die zugrundeliegenden Betriebssysteme eine obere Grenze von ca. 5.000 Threads (und damit in diesem Fall abbildbaren Agenten) besitzen, die sie gleichzeitig verwalten können. Unabhängig davon stellt eine große Anzahl von Threads einen erheblichen Mehraufwand, der verarbeitet werden muss, dar. Daher führt diese Vorgehensweise zu großen Performance-Problemen. Als Gegenmaßnahme verwendet MASeRaTi daher einen Micro-Kernel, der über eigene Scheduling-Mechanismen verfügt und damit eine optimale Zeitplanung, die während der Simulation laufend weiter verbessert wird, ermöglicht.

Die Kommunikation zwischen Agenten stellt oft einen ernsthaften Performance-Engpass dar. Die Kommunikation wird meistens mit speziellen und teilweise komplexen Protokollen und Sprachen, die nicht notwendigerweise auf Geschwindigkeit

optimiert sind, realisiert. In MASeRaTi versuchen wir, das Kommunikationsproblem mit Hilfe der Synchronisationsfunktionen des für die Zwecke der verteilten Rechnung weit verbreiteten Message Passing Interface (MPI) zu überwinden.

Agent-Systemen ist gemein, dass alle Agenten zu einem bestimmten Zeitpunkt möglicherweise gleichzeitig aktiv sein können und Zugriff auf alle verfügbaren Informationen benötigen, um ihre Denkabläufe auszuführen. Das bedeutet, dass es ist sehr schwierig, im Voraus abzuschätzen, welche Agenten Zugriff auf welche Informationen zu welchem Zeitpunkt benötigt. In großen Szenarien können all diese Informationen möglicherweise nicht vollständig im Hauptspeicher vorgehalten werden und die Übertragung von Daten von und zu einer Speicherauslagerung (wie einer Festplatte) stellt einen vergleichsweise langsamen Ablauf dar. Daher wird ein effizienter Caching-Mechanismus, der sich wie der Planungsmechanismus zur Laufzeit selber optimiert, benötigt. Während sich diese Fragen primär auf die Laufzeit der Simulation beziehen, darf auch der Entwurf des Systems nicht vernachlässigt werden. Dafür wird ein effizienter Rahmen für die Modellierung, die Techniken und Methoden, ein zuverlässiges und bewährtes Metamodell sowie begleitende Tool zur Unterstützung benötigt.

Alle Agenten, ihre Umgebung sowie das Ziel der Simulation, also die gewünschten Auswertungen, müssen individuell festgelegt werden. Ohne geeignete Leitlinien können sich diese Prozesse als schwer oder gar nicht durchführbar für ständig wachsende Szenarien herausstellen.

Von einem Experten für ein Teilsystem kann nicht erwartet werden, dass er das gesamte Szenario entwirft und daraus ausführbaren Code generiert. Daher soll die MASeRaTi Plattform bei ihren Anwendern keine Kenntnisse der erforderlichen Programmierschnittstellen und Datenstrukturen oder dem Prinzip der agentenbasierten Simulation und deren effizienter Parallelisierung voraussetzen.

MASeRaTi Architektur

Wie bereits beschrieben ist die MASeRaTi Architektur von Grund auf auf Skalierbarkeit ausgelegt.

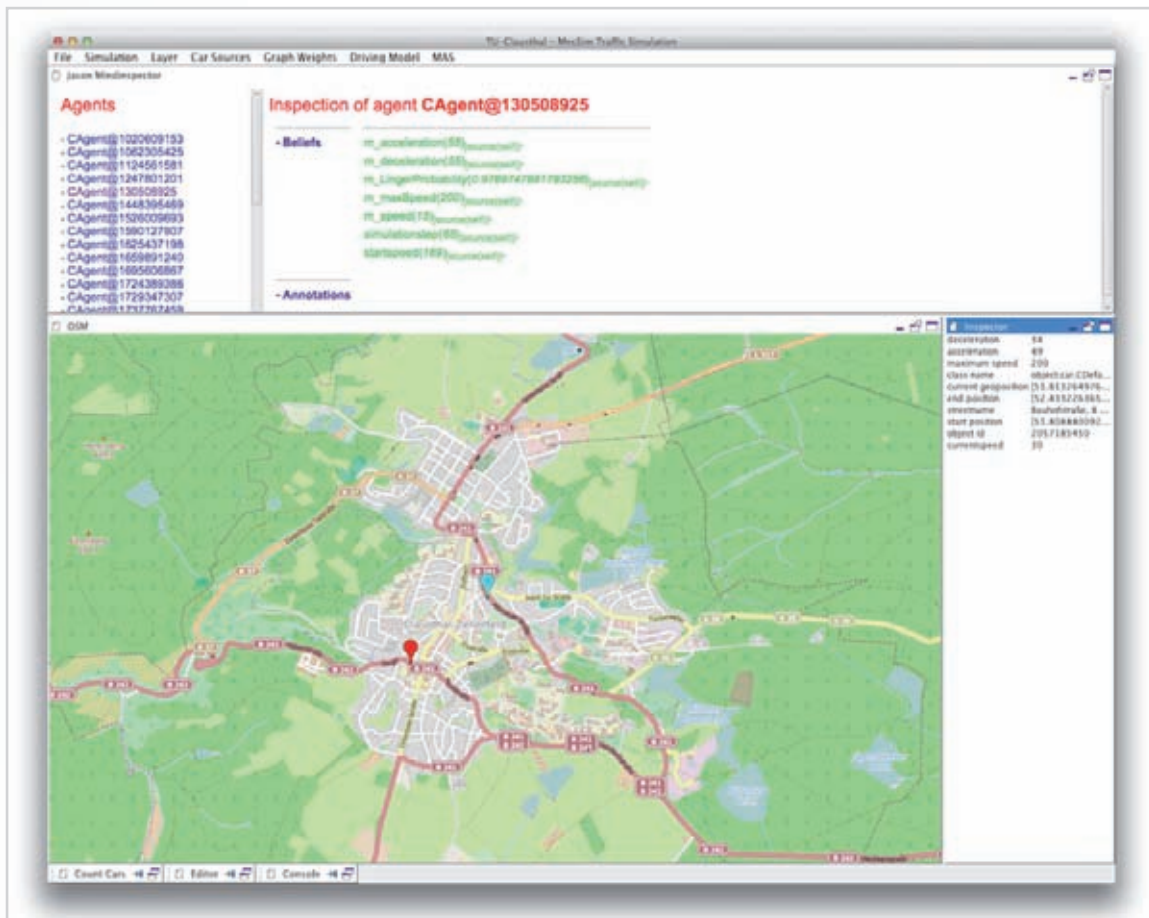


Abbildung 2: MecSim Konzeptbewertung

Figure 2: MecSim Concept Evaluation

source code all by herself. Fundamental knowledge of the required programming interfaces and data structures or even how to efficiently parallelize an agent based simulation, while certainly nice to have, is something we do not want to assume from MASeRaTi's user base.

MASeRaTi Architecture

As mentioned, MASeRaTi's architecture is designed to support scalability from the ground up. It draws inspiration from concepts from game development, especially from games which are distributed over the network to support an incredible number of players. This requires having a

client server architecture which synchronizes its state by sending messages. Of course, we do not have the corresponding machines handy, and our platform has to run on smaller systems down to standard desktop computers for development purposes.

MASeRaTi's architecture can be roughly divided into a number of separate but dependent layers. On the bottom rests the so called Micro Kernel, the technical backbone of the whole system. It is written in C++ to offer the greatest possible performance. It realizes all of the above mentioned parallelization features as well as the scheduling over the network. This scheduler possesses a dedicated thread pool. By constantly measuring

Es werden dabei Konzepte aus der Spieleentwicklung, insbesondere von Spielen, die über das Netzwerk verteilt laufen und große Anzahl von parallel aktiven Spielern unterstützen, übernommen. Dies erfordert eine Client-Server-Architektur, die ihren Zustand durch Senden von Nachrichten synchronisiert. Da uns keine vergleichbaren großen Systeme zur Verfügung stehen, muss unsere Plattform während der Entwicklung auf üblichen Desktop-Rechnern lauffähig sein.

Die MAsEraTi Architektur kann grob in eine Anzahl von getrennten, aber abhängigen Schichten unterteilt werden. Die Basis stellt der sogenannte Micro-Kernel, das technische Rückgrat des gesamten Systems, dar. Es ist in C++ geschrieben, um die größtmögliche Leistung zu bieten. Es verwirklicht alle oben genannten Parallelisierungsfunktionen sowie das Scheduling über das Netzwerk. Dieser Scheduler besitzt einen eigenen Thread-Pool. Durch die ständige Messung der Laufzeit aller Simulationsobjekte pro Simulationsschritt kann die Verteilung der Objekte über die vorhandenen Instanzen (z.B. die CPUs oder Kerne) optimiert werden. Ein Thread-Stealing-Mechanismus wird eingesetzt, was bedeutet, dass alle Kerne Threads (bzw. Simulationsobjekte) aus dem Thread-Pool ziehen, wodurch sichergestellt wird, dass alle Kerne so lange wie möglich beschäftigt bleiben und keine Instanz auf den Abschluss eines anderen Thread warten muss. Dies stellt sicher, dass die Rechenzeit gegenüber einem festen Szenario bestmöglich ist. Das Modell zugrunde liegende Modell kann dabei als stochastischer Prozess ausgedrückt werden. Weitere Funktionen vom Micro-Kernel zur Verfügung gestellte Eigenschaften sind Serialisierungsmechanismen, eine Plug-in-Schnittstelle für einfache Erweiterung der Plattform, wie z.B. um Statistikfunktionen, eine Komponente für die Speicherverwaltung und Abstraktionen der realen Netzwerkschnittstelle und gegenüber dem Betriebssystem. Zusammenfassend stellt der Micro-Kernel einen generischen parallelisierten Simulationskontext, jedoch noch ohne agentenbasierte Funktionen, dar.

Simulationsobjekte werden in synchronisierte und nicht synchronisierte Objekte unterteilt. Synchronisierte Objekte sind Objekte, auf die möglicherweise von anderen aus Objekte zugegriffen

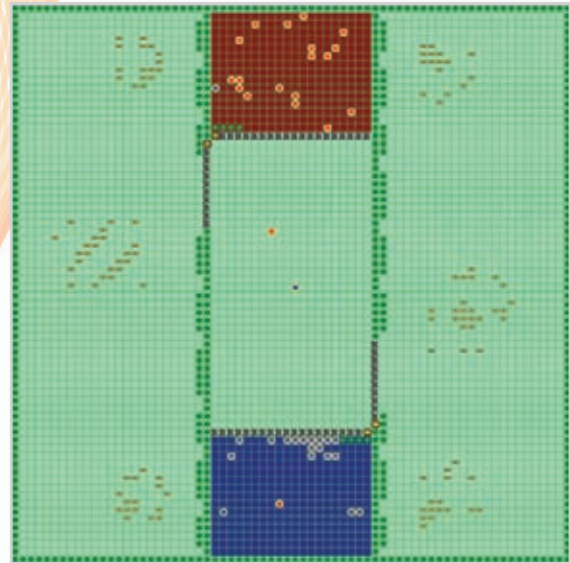


Abbildung 3: Multi-Agent Programmierwettbewerb: Kuh-Szenario

Figure 3: Multi-Agent Programming Contest: Cow Scenario

werden muss und die daher auf jedem Kern vorhanden sein müssen. Die nicht synchronisierten Objekte sind unabhängig und können auf jedem beliebigen Kern bearbeitet werden. Der Vorteil dieser Vorgehensweise besteht darin, dass der zweitgenannte Objekttyp nur auf einem einzigen Kern vorliegen muss und im Rahmen der Parallelisierung nicht verarbeitet werden muss. Dieser gesamte Mechanismus ist mit der Boost MPI Bibliothek in C++ implementiert.

MAsEraTi Metamodell

Um nun die grundlegende parallele Simulations-Engine mit Multi-Agenten-basierten Konzepten anzureichern, wird das MAsEraTi Metamodell, welches senkrecht zu den anderen Ebenen aufgestellt ist, definiert. Das Metamodell verwendet Konzepte der agentenorientierten Paradigmen und baut auf dem eher technischen Micro-Kernel auf. Die Kernobjekte in dem MAsEraTi Metamodell sind Simulation, Welt, Agenten und Artefakt. Das Simulationsobjekt stellt eine Simulation dar und

the runtime of all simulation objects per simulation step, the distribution of these objects over all available instances, i.e. CPUs or cores, can be optimized. A thread-stealing mechanism is employed, which means that each core is picking from the thread-pool the thread (or simulation object) which is most likely to ensure that all cores are kept busy for as long as possible and no instance has to wait for another one to finish its workload for an overly long period of time. This implies the best possible computation time given a fixed scenario. The model behind this can be expressed as a stochastic process.

Further capabilities provided by the Micro Kernel are serialization mechanisms, a plug-in interface for easy extension of the platform, e.g. with statistical functions, a component for memory management and abstractions from the real network interface and operating system. In summary, the Micro Kernel represents a generic parallelized simulation context, yet without agent based features.

Simulation objects are only divided into synchronized and non-synchronized objects. Synchronized objects are those which are possibly accessed by other objects and therefore have to be present on each core. The non-synchronized objects are independent and can live on any core. The advantage of this is that the latter object type has to be processed on one single core only, which makes for the real parallelization. This whole mechanism is implemented with the well-known Boost MPI library for C++.

MASeRaTi Metamodel

Now, to enrich the basic parallel simulation engine with multi-agent based concepts, the MASeRaTi *metamodel*, vertical to the other layers, has to be defined. This metamodel provides concepts from the agent-oriented paradigm and builds upon the rather technical Micro Kernel. The core objects in the MASeRaTi metamodel are Simulation, World, Agent and Artifact. The Simulation object represents of course one simulation and contains all of the other objects on this level. The World object represents one world, or in agent terms an *environment*, in which the agent is located and in which it can act upon to manipu-

late the environment's state. A world can possibly contain many heterogeneous structures, like grids or graphs, which themselves form an environment. Agents may be related to more than one structure, having a different representation in each and being able to perform different actions to alter the structure. This is quite similar to binding mechanisms in modern programming languages.

The term "agent" comprises everything in the simulation that is an autonomous acting entity. These entities can be further divided into (Agent) Body and (Agent) Mind. The Mind is the abstract component representing the disembodied reasoning capacity of an agent. It is where the individual decisions are made based on the current state of the world. This is a rather general concept, however, it does not impose any restrictions on the type of agent used and so, any existing concept of an agent can be implemented for our platform.

Each of these Minds is now associated with at least one Body. The Body is a usually physical representation of an agent in an environment. Thus, it has a number of properties and certain actions available, which can be used to manipulate the environment. This separation of concerns is simultaneously the first step in realizing the above mentioned binding mechanism. Currently, we consider raising the abstraction of an agent so that it cannot only connect to agent bodies, but any other simulation object (even agent minds). This concept is also called a *holonic agent*.

The remaining concept of the metamodel is an Artifact. Contrary to an agent, it represents an entity in the world that is acting non-autonomously. An artifact can be designed to offer certain functionalities to an agent while it processes requests like any conventional software component.

Time has yet to show whether these base concepts are enough to express (almost) any scenario imaginable. While intuitively it seems that these objects are not very restrictive on the scenario structure, it might be that further concepts are required to ease the simulation development. A domain expert designing a scenario would now derive domain-specific classes from the elements of the metamodel, e.g. driver agents and a street graph for a traffic scenario.

enthält alle anderen Objekte auf dieser Ebene. Das Weltobjekt stellt die *Umgebung*, in der sich der Agent befindet und in der er Handlungen vornehmen kann, um den Zustand der Umwelt zu verändern, dar. Eine Umgebung kann dabei viele heterogenen Strukturen, wie Gitter oder Grafiken, die ihrerseits eine Umgebung bilden, enthalten. Agenten können dabei in mehreren Strukturen enthalten sein und in jeder optional eine andere Darstellung besitzen und andere Möglichkeiten zur Veränderung der Struktur besitzen. Dies entspricht weitgehend dem Mechanismus der Bindung in modernen Programmiersprachen. Der Begriff „Agent“ umfasst alles, was in der Simulation eine autonom handelnde Einheit darstellt. Diese Einheiten können weiter in (Agent) Körper und (Agent) Geist unterteilt werden. Der Geist ist die abstrakte Komponente, die das körperlose Denkvermögen eines Agenten darstellt und in dem die individuellen Entscheidungen basierend auf dem aktuellen Zustand der Welt getroffen werden. Dies stellt ein sehr allgemeines Konzept dar, welches daher keine Einschränkungen in Bezug auf den Typ der verwendeten Agenten beinhaltet, und so jedes bestehende Agenten-Konzept für unsere Plattform implementiert werden kann.

Jeder dieser Geiste ist nun mit mindestens einem Körper verbunden. Der Körper stellt dabei üblicher Weise eine physische Repräsentation eines Agenten in einer Umgebung dar. Somit hat er eine Reihe von Eigenschaften und es stehen ihm bestimmte Aktionen zur Verfügung, die verwendet werden können, um die Umwelt zu manipulieren. Diese Trennung in die Bereiche Geist und Körper stellt damit den ersten Schritt der Verwirklichung des oben genannten Bindungsmechanismus dar. Derzeit denken wir über eine Erhöhung des Abstraktionsgrads der Agenten nach, so dass Agenten in Zukunft nicht nur mit Körpern, sondern mit jedem Objekt in der Simulation (auch den Geist-Komponenten anderer Agenten) verbunden werden können. Dieses Konzept wird auch als *Holonic Agenten* bezeichnet.

Das letzte verbleibende Konzept des Metamodells stellen die Artefakte dar. Im Gegensatz zu einem Agenten stellen sie ein Element der Umgebung dar, das nicht autonom handeln wird. Ein Artefakt kann entworfen werden, um Agenten bestimmten

Funktionalitäten zu bieten, während es Anfragen wie jede herkömmliche Softwarekomponente verarbeitet.

Es muss sich noch zeigen, ob diese Grundkonzepte ausreichend sind, um (fast) jedes Szenario abbilden zu können. Während es intuitiv scheint, dass diese Objekte nicht sehr restriktiv auf die Szenario-Struktur wirken, kann es vorkommen, dass weitere Konzepte sind erforderlich, um die Entwicklung der Simulation zu erleichtern. Ein Anwender, der ein Szenario entwerfen soll, kann nur anwendungsspezifische Klassen von den Elementen des Metamodells ableiten, z.B. Fahrer und ein Straßennetz für eine Verkehrssimulation. Außerdem ist das Metamodell auch leicht horizontal erweiterbar, indem einfach neue Objektklassen von dem synchronisierten und dem nicht-synchronisierten Basisobjekt abgeleitet werden. An dieser Stelle verbindet sich das Metamodell auch mit dem Micro-Kernel.

Eine erste Version des Metamodells, Agent Model Layer genannt, wurde mit dem so genannten Kuh-Szenario aus dem Multi-Agent Programmierwettbewerb getestet. Die Umsetzung der Objekttypen von dem Szenario in die Metamodell-Klassen hat sich als leicht umsetzbar herausgestellt, aber es hat sich auch gezeigt, dass noch ein erheblicher Programmieraufwand notwendig war und geeignete Modellierungswerkzeuge erforderlich sind, um die Entwicklung zu beschleunigen und um die Fehlerrate zu reduzieren.

Um die Modellierungsumgebung von der Laufzeitumgebung zu trennen, müssen die Modelle in einem gemeinsamen generischen Datenformat, das durch verschiedene Werkzeuge bearbeitet werden kann, angegeben werden. Wir arbeiten derzeit an einem Format auf Basis der verbreiteten JavaScript Object Notation (JSON), die bereits von einer Vielzahl an Werkzeugen verwendet wird.

Interpreter

Während das Metamodell (laufende Master-Arbeit von T. Ahlbrecht) es uns ermöglicht, die Struktur einer Simulation ähnlich wie den Syntax einer Programmiersprache zu definieren, muss noch das Verhalten der Simulationsobjekte unabhängig

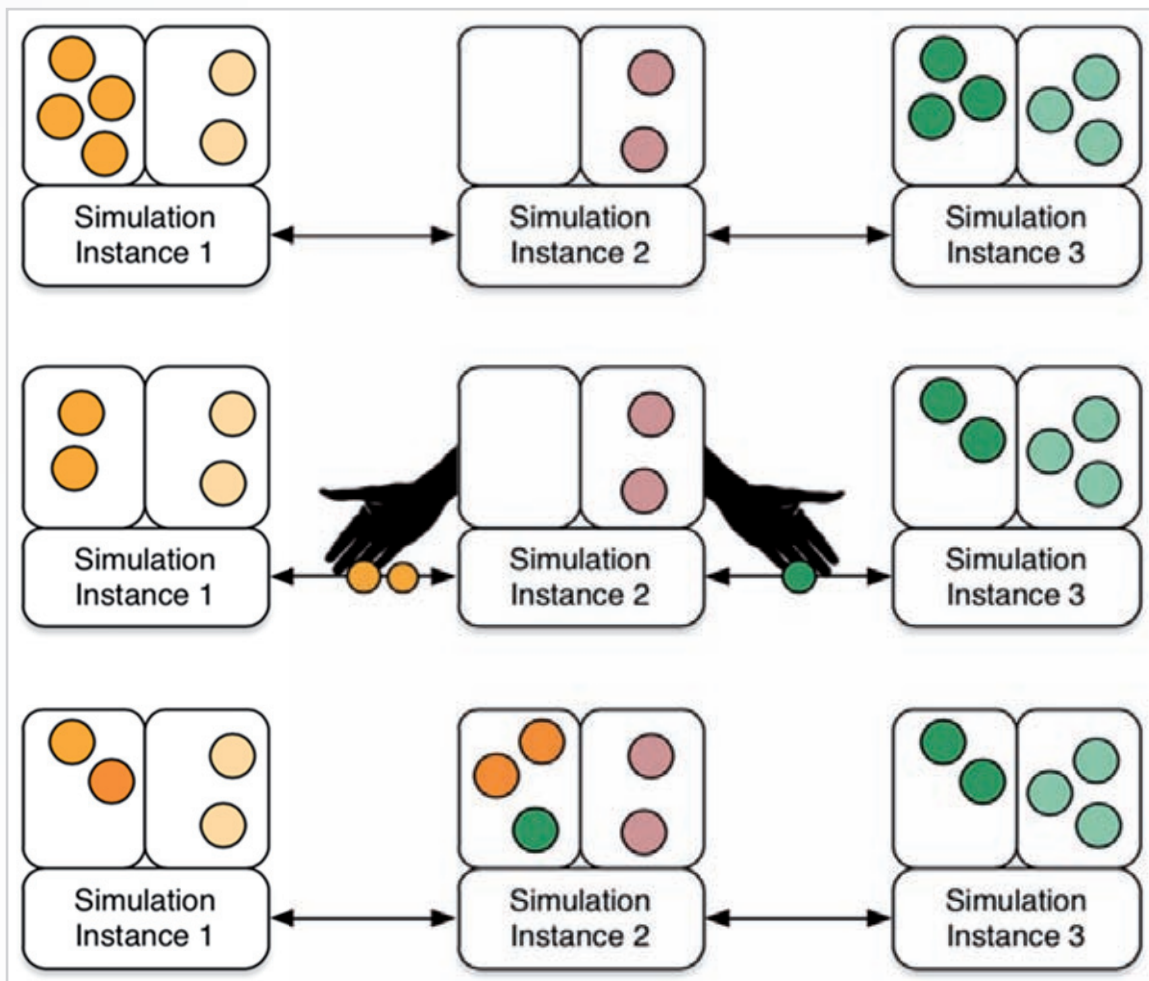


Abbildung 4: Thread-Stealing

Figure 4: Thread stealing

Nevertheless, the metamodel is also easily horizontally extensible by simply deriving new object classes from a synchronous or non-synchronous object. At the same time, this is where the metamodel connects to the Micro Kernel.

An earliest version of the metamodel, then called Agent Model Layer, has been evaluated using the so called Cow Scenario from the Multi-Agent Programming Contest. It was fairly easy to map all object types from that scenario to the metamodel's classes but it showed, that still a considerable

programming effort was necessary and suitable modelling tools are required to speed up development and reduce the error rate.

In order to separate the modelling environment from the runtime environment, the models have to be specified in a common generic data format, which can be edited by different tools. We are currently working towards a format based on the established JavaScript Object Notation (JSON), which can already be used by a wide variety of tools.

von Laufzeitumgebung definiert werden. Um dies zu erreichen, können benutzerdefinierte Interpreter während der Laufzeit über eine Plug-in-Infrastruktur eingesetzt werden. Diese Interpreter definieren die operationale Semantik aller Simulationsobjekte, vor allem der Agenten und wie sie ihre Datenstrukturen (d.h. Überzeugungen, Ziele und Pläne) verwenden, um ihr Verhalten zu bestimmen. Die Eingangsdaten für den Interpreter ergeben sich aus dem Modellrahmen. Dies ermöglicht es, verschiedene Agenten-Konzepte entweder in der gleichen Simulation einzusetzen oder in zwei Simulationen die, die bis auf die verwendeten Agenten-Konzepte identisch sind, zu vergleichen. Gegenwärtig werden zwei unterschiedliche Interpreter in unserem Projekt zum Zwecke der Bewertung der Leistungsfähigkeit der MASERaTi Architektur sowie verschiedener Agenten-Konzepte entwickelt.

Einer dieser Interpreter (Bachelor-Arbeit von C. Kuper) basiert auf der Skriptsprache Lua, um eine Hochleistungs-Ausführung der Simulationsumgebung, ohne dass dafür eine Umgebung in C++ programmieren werden muss, zu ermöglichen. Die Vorteile der Verwendung von Lua bestehen in dem sehr kompakten Interpreter (Lua selbst ist eine interpretierte Sprache), seine Skriptsprachenkonzepte, durch die es ähnlich einfach zu bedienen ist wie Python oder JavaScript, und die Möglichkeit, C++ Komponenten effizient zu integrieren. Zur Abbildung der Entscheidungsprozesse der Agenten verwendet der Interpreter das Answer Set Programming, eine Form der deklarativen Programmierung mit Prolog-artigen Abfrageausdrücken.

Der zweite Interpreter basiert auf der deklarativen Logik-Modellerungssprache *AgentSpeak*. Er soll dem Java-basierten Interpreter der erweiterten Version von *AgentSpeak* (*Jason*) ähneln. Um eine hohe Leistungsfähigkeit zu erzielen, wird er direkt in C++ implementiert. Erste empirische Tests sind vielversprechend: Die gleichzeitige Ausführung von rund 500.000 Agenten war ohne Probleme in einer angemessenen Zeit möglich (vgl. laufende Bachelor-Arbeit von N. Fiekas).

Es muss noch die oberste Komponente der MASERaTi Architektur betrachtet werden. Der Block *System Scenarios and Domain Ontologies*

repräsentiert eine konkrete Ausprägungen des MASERaTi Metamodells. Es enthält alle möglichen domänenspezifische Szenarien, die sich wiederum von dem Metamodell ableiten. Im Rahmen des Projektes wurden bereits eine Anzahl unterschiedlicher Szenarien auf ihre Eignung und spezifischen Anforderungen hin analysiert. Dazu gehören kooperative Verkehrsmodellierung, flexible energieeffiziente Automatisierungssysteme, kooperative Robotersysteme sowie ausgewählte Szenarien aus dem internationalen Multi-Agent Programmierwettbewerb.

Simulationsergebnisse

Im Moment decken wir mit unseren Modellen bereits weite Gebiete der Simulation ab. Komplexe Simulationen liefern selbstverständlich auch komplexe Ergebnisse, daher bleibt noch die Aufgabe, die Ausgaben dahingehend einzuschränken, dass die interessanten Ergebnisse direkt sichtbar werden. Aus diesem Grund stehen in der MASERaTi Architektur Daten-Management, Bewertungen und Benchmarking im Mittelpunkt. Des Weiteren muss noch untersucht werden, wie die Auswertung der Ergebnisse durch das Modellierungs-Framework beeinflusst wird und wie die verschiedenen Datentypen am besten visualisiert werden können, um den Entwickler in die Lage zu versetzen, eine möglichst effiziente Datenanalyse und -Bewertung durchführen zu können. Kriterien und Kennzahlen für die Beurteilung der agentenbasierte Systeme bilden auf jeden Fall einen Ausgangspunkt für diese Fragestellungen (zukünftige Bachelor-Arbeit von J. Haltenhof).

Die MecSim Plattform

Eine wesentliche Rolle bei der Bewertung von geplanten Konzepten spielt die Java-basierte MecSim Plattform. Es handelt sich dabei ebenfalls um eine Eigenentwicklung, so dass neue Ideen schnell umgesetzt werden können. Eine ganze Reihe von Machbarkeitsstudien wurde bereits mit Hilfe dieser flexiblen Plattform durchgeführt. Momentan wird mit der Plattform eine einfache, aber kontinuierlich erweiterte Verkehrssimulation auf der mikroskopischen Eben durchgeführt. Die neuste Ergänzung bestand in der Neu-Implementierung der bereits erwähnten Jason Laufzeitumgebung,

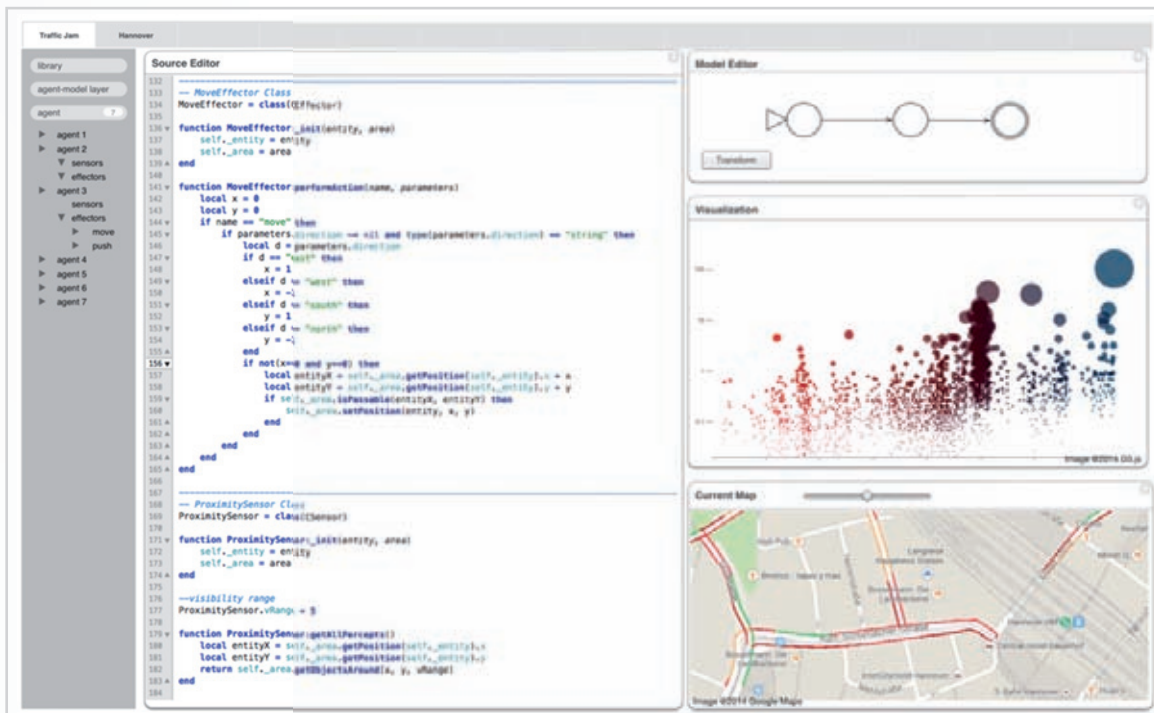


Abbildung 5: Mockup des Modellierwerkzeugs

Figure 5: Modelling Tool Mockup

Interpreters

While the metamodel (upcoming M.Sc. thesis of T. Ahlbrecht) enables us to specify the structure of a simulation, much like syntax does for a programming language, still the behavior of simulation objects has to be defined while remaining decoupled from the runtime environment. To achieve this, custom interpreters can be plugged-in during runtime. These interpreters define the operational semantics of all simulation objects, especially of the agents and how they employ their data structures (i.e. beliefs, goals and plans) to generate their behavior. The input data for the interpreters stems from the modelling framework. This makes it possible to utilize different agent concepts either in the same simulation or in two simulations that are the same but for the used agent concept, which also permits a straightforward comparison. Currently, two different interpreters are being developed in our project for the purpose of evalu-

ating the performance of the MAsRaTi architecture, as well as different agent concepts. One of these interpreters (B.Sc. thesis of C. Kuper) is based on the scripting language Lua to ensure a high-performance execution of simulation objections without requiring a simulation engineer to program in C++. The advantages of using Lua is its very small interpreter (Lua itself is an interpreted language), its scripting language nature, which makes it as easy to grasp as Python or JavaScript, and the possibility to efficiently integrate it with C++ components. For the agent's reasoning the interpreter builds upon Answer Set Programming, a form of declarative programming with Prolog-style query evaluation.

The second interpreter is based on the declarative logic modelling language *AgentSpeak*. It is designed to resemble the Java-based interpreter for an extended version of *AgentSpeak* (*Jason*). To achieve high performance, it is implemented

wodurch einige potenzielle Probleme behoben wurden und mit nur etwa 1.000 Agenten ohne Probleme abgebildet werden können. Dadurch wurde nachgewiesen, dass, obwohl das ursprüngliche Jason einige technische Probleme in Bezug auf die Durchführung von mikroskopischen Verkehrssimulationen aufweist, das generelle Konzept hinter Jason dennoch zweckmäßig ist.

Noch wichtiger ist, dass die Plattform mittlerweile regelmäßig in einigen Bachelor-Lehrveranstaltungen zur Veranschaulichung von Verkehrssimulationen und Agenten-Technologien eingesetzt wird. Wir beabsichtigen, diese Veranstaltungen weiter zu entwickeln und die Plattform intensiv von den Studenten nutzen zu lassen.

Ausblick

Wie beschrieben wurde, ist die Entwicklung der MAsEraTi Architektur und die begleitenden Umsetzung in vollem Gange und erste Auswertungen, insbesondere durch die MecSim Plattform, liefern vielversprechende Ergebnisse. Unser Modellierungs-Framework ist konzeptionell angegangen worden und eine spezifische Sammlung von Werkzeugen, mit denen die beschriebenen Fragestellungen untersucht werden können, ist in Arbeit.

Dadurch kommen wir sukzessive dem Ziel der Modellierung und Simulation von großen Systemen von Systemen näher.

Projektdaten

Das Projekt wird seit Juli 2013 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

directly in C++. First empirical tests look promising: concurrent execution of around 500,000 agents was possible without problems in a reasonable amount of time (upcoming B.Sc. thesis of N. Fiekas).

We still have to tackle the topmost component of the MAsERaTi architecture. The block *System Scenarios and Domain Ontologies* represents concrete instantiations of the MAsERaTi metamodel. It contains all possible domain-specific scenarios which in turn can be derived from the metamodel. In the scope of the project a number of different scenarios have already been analyzed for suitability and specific requirements. These include cooperative traffic modelling, flexible energy-efficient automation systems, cooperative robotic systems and a selected number of scenarios from the international Multi-Agent Programming Contest.

Simulation results

For now, we broadly covered the topic of generating simulation results. Complex simulations of course generate a lot of complex results. It still remains to constrain the results to what we are interested in and how to analyse the remainder of the data. Therefore, in the center of the MAsERaTi architecture are the data management, evaluation & benchmarking suite.

It still has to be evaluated how the analysis of results can be influenced by the modelling framework and how to best visualize the different types of data to enable the simulation developer to perform a most efficient data analysis and verification. Criteria and metrics for the assessment of agent-based systems definitely form a starting point in this case (future BSc thesis of J. Haltenhof).

The MecSim platform

A fundamental role in evaluating our planned concepts is played by the Java-based MecSim platform. It is also being developed in-house, which allows for rapid prototyping of new ideas. Quite a number of proofs of concept have already been conducted on this flexible platform. As of now, it is running a basic but growing traffic scenario on the micro level. The latest addition was a re-implementation of the already mentioned Jason runtime engine, fixing some potential issues and running around 1,000 agents without problems. This was done to prove that, while Jason revealed technical problems running our microscopic traffic simulation, the concepts behind Jason were still useful in this case. More importantly, the platform has been established in a recent BSc course, teaching students about traffic simulation and agent technology. We plan to develop this course further and let the platform extensively used by the students.

Outlook

As we have seen, the development of the MAsERaTi architecture and its accompanying implementation is well underway and first evaluations and proofs of concepts, especially through the MecSim platform, are very promising. Our modelling framework is conceptually approached and a specific toolchain that is up to the task is on the horizon.

Thereby, we are getting ever closer to the goal of large-scale modelling and simulation of *Systems of Systems*.



120

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 Simulationen 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/de/lehre/ringvorlesung



Lecture series „Simulation Sciences“

.....
121
.....

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/de/lehre/ringvorlesung

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 Optimierungsfunktionen 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 Austenitkonditionierung 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 Resource 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 hierarchically 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 trillion 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.)

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 Simulationwissenschaften.

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 defined 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 (binomial, 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 financial 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 competences 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. Benefits 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 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 specific 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 specific 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

.....
129
.....

Vorstand des SWZ // Board of Directors

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

Prof. Dr. Thomas Hanschke
(Vorstandsvorsitzender)
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
Universität Göttingen

Prof. Dr. Dieter Hogrefe
(Stellvertretender Vorstandsvorsitzender)
Arbeitsgruppe Telematik
Institut für Informatik
Universität Göttingen

Dipl.-Math. Fabian Kirchhoff
Arbeitsgruppe Stochastische Optimierung
Institut für Angewandte Stochastik und
Operations Research
Technische Universität Clausthal

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

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

Prof. Dr. Xiaoming Fu

Arbeitsgruppe Computer Networks Group
Institut für Informatik
Universität Göttingen

Prof. Dr. Jens Grabowski

Arbeitsgruppe Softwaretechnik für Verteilte Systeme
Institut für Informatik
Universität Göttingen

Prof. Dr. Thomas Hanschke

Arbeitsgruppe Stochastische Modelle
in den Ingenieurwissenschaften
Institut für Angewandte Stochastik
und Operations Research
Technische Universität Clausthal

Prof. Dr. Dieter Hogrefe

Arbeitsgruppe Telematik
Institut für Informatik
Universität Göttingen

Prof. Dr. Michael Kolonko

Arbeitsgruppe Stochastische Optimierung
Institut für Angewandte Stochastik
und Operations Research
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.-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
Universität Göttingen

Prof. Dr. Stephan Waack

Arbeitsgruppe Theoretische Informatik
und Algorithmische Methoden
Institut für Informatik
Universität Göttingen

Wissenschaftliche Mitarbeiter**Fabian Glaser, M.Sc.**

Arbeitsgruppe Softwaretechnik für
Verteilte Systeme
Institut für Informatik
Universität Göttingen

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

Salke Hartung, M. Sc.
Arbeitsgruppe Telematik
Institut für Informatik
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.-Inf. Daniel Honsel
Arbeitsgruppe Theoretische Informatik
und Algorithmische Methoden
Institut für Informatik
Universität Göttingen

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

Dipl.-Math. Fabian Kirchoff
Arbeitsgruppe Stochastische Optimierung
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

Dr. Somayeh Taheri
Arbeitsgruppe Telematik
Institut für Informatik
Universität Göttingen

Dr. Horst Zisgen
Arbeitsgruppe Stochastische Modelle
in den Ingenieurwissenschaften
Institut für Angewandte Stochastik
und Operations Research
Technische Universität Clausthal



Veröffentlichungen //

Publications

Strukturuntersuchungen zur Entstehung und Fortpflanzung von Verspätungen in Verkehrsnetzen

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 (OASISs)*, 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 Linieneplanung. 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.

Sichere Kommunikation in Internet der Dinge 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. Arumathurai, 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

2015

- X. Wang, "A Reliability Assessment Framework for Cloud Applications", *CLOUD COMPUTING 2015* (submitted)

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

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. Gultas, 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

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.

Postanschrift

Simulationswissenschaftliches Zentrum Clausthal-Göttingen
Prof. Dr. Thomas Hanschke
Institut für Angewandte Stochastik und Operations Research
Erzstraße 1
38678 Clausthal-Zellerfeld
Telefon: (05323) 72-2401/-2567
Fax: (05323) 72-2304

Impressum

Herausgeber: Simulationswissenschaftliches Zentrum Clausthal-Göttingen
Redaktion: Dr. Alexander Herzog, Prof. Dr. Thomas Hanschke
Layout und Satz: Melanie Bruchmann
Bildnachweis: Titelbild – Fotolia.com (© marigold_88)

Hier nicht erwähnte Bilder stammen von den Autoren.



www.simzentrum.de