

Gemeinsamer Abschlussbericht
des BMBF-Verbundprojekts

SKALB

**Lattice-Boltzmann-Methoden für
skalierbare Multi-Physik-Anwendungen**

J. HABICH, G. WELLEIN, M. WITTMANN T. ZEISER

Regionales Rechenzentrum Erlangen (RRZE)

Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

C. FEICHTINGER, K. PICKL, U. RÜDE, F. SCHORNBAUM

Lehrstuhl für Systemsimulation (LSS)

Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

M. GEVELER, S. TUREK

Lehrstuhl für Angewandte Mathematik & Numerik (TUDo)

Technische Universität Dortmund

M. KRAFCZYK, K. KUCHER, M. SCHÖNHERR

Institut für rechnergestützte Modellierung im Bauingenieurwesen (iRMB)

Technische Universität Braunschweig

U. KÜSTER, M. RESCH

Höchstleistungsrechenzentrum Stuttgart (HLRS)

Universität Stuttgart

F. PLATTE

IANUS GmbH, Dortmund (IANUS)

Das diesem Bericht zugrundeliegende Vorhaben *SKALB* wurde mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) unter dem Förderkennzeichen 01IH08003 im Rahmen des ersten Calls „HPC-Software für skalierbare Parallelrechner“ von Anfang 2009 bis Ende 2011 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Inhaltsverzeichnis

1	Kurze Darstellung des Projekts	1
1.1	Aufgabenstellung	2
1.2	Randbedingungen und Voraussetzungen	4
1.3	Planung und Ablauf des Vorhabens	4
1.4	Wissenschaftlicher und technischer Stand, an den angeknüpft wurde	5
1.4.1	Ausgangssituation VirtualFluids	6
1.4.2	Ausgangssituation waLBerla	6
1.4.3	Ausgangssituation ILBDC	7
1.4.4	Ausgangssituation FEAT* und FEAST*	8
1.5	Zusammenarbeit mit anderen Stellen	9
2	Eingehende Darstellung des Projekts	11
2.1	Verwendung der Zuwendung und erzielte Ergebnisse	11
2.1.1	AP1: Portierung und Optimierung von LB-Anwendungen auf massiv parallele HPC-Systeme	11
2.1.2	AP2: Weiterentwicklung von LB-Methoden für praktische Anwendungen auf hochskalierenden Systemen	16
2.1.3	AP3: Verbesserte numerische Ansätze für LB-Methoden	21
2.1.4	AP4: Hardwarenahe Implementierung für Nicht-Standardprozessoren	23
2.1.5	AP5: Benchmarking und Showcases	25
2.1.6	Erzielte Fortschritte in den weiterentwickelten LB-Codes	35
2.2	wichtigste Positionen des zahlenmäßigen Nachweises	40
2.2.1	FAU / RRZE	40
2.2.2	FAU / LSS	41
2.2.3	iRMB	41
2.2.4	TUDo	42
2.2.5	HLRS	43
2.2.6	IANUS	43
2.3	Notwendigkeit und Angemessenheit der geleisteten Arbeit	44
2.4	Voraussichtlicher Nutzen, insbesondere Verwertbarkeit der Ergebnisse im Sinne des fortgeschriebenen Verwertungsplans	46
2.5	Während der Durchführung des Vorhabens bekannt gewordene Fortschritte bei anderen Stellen	47
2.6	Erfolgte oder geplante Veröffentlichungen der Ergebnisse	47
2.6.1	Veröffentlichungen FAU	47
2.6.2	Veröffentlichungen TUDo	50
2.6.3	Veröffentlichungen iRMB	53

2.6.4	Veröffentlichungen HLRS	54
2.6.5	Vorträge FAU	54
2.6.6	Vorträge TUDo	60
2.6.7	Vorträge iRMB	62
2.6.8	Vorträge HLRS	62
2.6.9	Vorträge IANUS	63
2.7	Sonstige Referenzen	64

Kapitel 1

Kurze Darstellung des Projekts

Die Modellierung und Simulation strömungsmechanischer Prozesse hat in den letzten Jahrzehnten große Fortschritte gemacht. Triebfedern waren gleichermaßen rasante Entwicklungen in der Rechnertechnologie sowie substanzielle Neuerungen im methodisch-algorithmischen Bereich. Die Simulation hat sich daher im Bereich der Strömungsmechanik neben Experiment und Theorie etabliert. Vielfach muss dabei aber noch immer auf vereinfachte Modelle mit problemspezifisch angepassten Korrelationsparametern zurückgegriffen werden. Dies ermöglicht typischerweise qualitativ aussagekräftige CFD-Simulationen (Computational Fluid Dynamics). Ein verlässliches Scale-Up vom Labor- zum Produktionsmaßstab, im Zuge einer numerischen Prozessoptimierung, kann selbst bei Standardprozessen häufig nicht erfolgen.

Weit verbreitete kommerzielle CFD-Werkzeuge besitzen oft weder die numerische Leistungsfähigkeit moderner Methoden der Angewandten Mathematik und des Wissenschaftlichen Rechnens, noch sind sie für hochskalierbare Rechnersysteme mit potentiell heterogener Architektur ausgelegt. Ein weiteres Problem sind Lizenzkostenmodelle, deren Kostenfunktionen mit dem Parallelisierungsgrad ansteigen.

Die Verfügbarkeit validierter, kostengünstiger, hochskalierbarer CFD-Software, welche modernste Rechnersysteme effizient nutzen kann und gleichzeitig fortschrittliche numerische Methoden implementiert, ist daher von grundlegender Bedeutung, um die weiter ansteigende Rechenleistung für die numerische Strömungsmechanik nutzbar zu machen. Das vorliegende Projekt SKALB adressiert in diesem Zusammenhang die Lattice-Boltzmann-(LB)-Verfahren [Hän04], welche sich in den letzten 15 Jahren zu einer vielversprechenden methodischen Alternative zu konventionellen Lösungsverfahren für die gängigen Navier-Stokes-Gleichungen entwickelt haben. Neben den klassischen CFD-Bereichen, wie etwa der Automobilbranche, haben sie inzwischen auch Eingang in die Verfahrenstechnik, das Bauingenieurwesen, die Biomedizin und viele weitere Gebiete gefunden.

Im Rahmen von SKALB wurde das Zusammenspiel von numerischen Methoden, Datenstrukturen und möglichen Implementierungsalternativen für LB-Verfahren im Hinblick auf künftige hochparallele, heterogene Rechnerarchitekturen untersucht. Frühzeitige Umsetzung auf verfügbare „High-End“-Rechnersysteme und Nachhaltigkeit der entwickelten Softwarestrukturen, breite methodische Abdeckung sowie Validierung der Software insbesondere vor dem Hintergrund einer industriellen Nutzung waren die zentralen Aspekte des Projekts.

Die Programmpakete der beteiligten Partner konnten dabei auf ein nachhaltiges und hocheffizientes Niveau gebracht werden, das den beteiligten Gruppen die Möglichkeit eröffnet, ihre international führende Stellung ausbauen zu können – selbst unter Berücksichtigung der vielen Unbekannten hinsichtlich der weiteren Hardwareentwicklung. Darüber hinaus konnten zahlreiche breit nutzbare Erkenntnisse zur Entwicklung zukunftssicherer und hardwareeffizienter Softwarestrukturen für hochparallele, heterogene Hardwarearchitekturen gewonnen werden. Im Rahmen eines engen industriellen Austausches, der vom KMU-Partner kontinuierlich koordiniert und vorangetrieben wurde, konnte auch das wirtschaftliche Potential der SKALB-Entwicklungen demonstriert werden. Im Rahmen von zahlreichen wissenschaftlichen Vorträgen und Publikationen wurden die Projektergebnisse kontinuierlich an Dritte weitergegeben und stießen auf großes internationales Echo. Die SKALB-Ergebnisse haben das Potential, die weiteren Entwicklungen in der gesamten LB-Community hinsichtlich Methodik und Hardwareeffizienz nachhaltig zu beeinflussen.

1.1 Aufgabenstellung

Zentrale Aufgabe des Projekts SKALB war es, Lattice-Boltzmann-Löser vor dem Hintergrund des stattfindenden Wechsels hin zu homogenen und heterogenen Mehr- und Vielkernarchitekturen methodisch und technisch weiterzuentwickeln. Damit sollte ein Beitrag zum nachhaltigen Einsatz von LB-Verfahren in „Virtuellen Strömungslaboren“ geleistet werden, in denen Modellbildung, Simulation, Optimierung und experimentelle Untersuchung iterativ gekoppelt werden können. Zentral für das Projekt war die interdisziplinäre Zusammenarbeit von Wissenschaftlern aus den Ingenieurwissenschaften, der angewandten Informatik und Mathematik sowie Experten der Rechenzentren.

Lattice-Boltzmann-Verfahren zählen zu den *Emerging Technologies*, deren Entwicklung noch immer dynamisch voranschreitet. Daher war es nicht das Ziel von SKALB einen einzigen gruppenüberspannenden Applikationscode zu erstellen. Die in den Gruppen existierenden LB-Programmsysteme sollten vor dem Hintergrund ihrer jeweiligen Anwendungsbereiche weiterentwickelt werden und damit auch langfristig als hocheffiziente Tools zur Verfügung stehen. Gleichzeitig wurde der hohe Aufwand vermieden, der mit einer Neuentwicklung von Grund auf verbunden ist. Die abgedeckten Themengebiete umfassten dabei:

- Der ILBDC-Code (RRZE) ist auf die Simulation von laminaren Strömungen in extrem komplexen Geometrien mit geringem Hohlraumanteil (z.B. poröse Medien oder Katalysatorschüttungen in chemischen Reaktoren) spezialisiert.
- waLBerla (LSS) ist ein Software-Framework für massive parallele Multi-Physik-Anwendungen basierend auf der Lattice-Boltzmann-Methode, das durch sein systematisches Design sowohl flexible, anwendbar und erweiterbar ist als auch effizient und skalierbar.

- VirtualFluids (iRMB) ist ein adaptiver, hierarchischer Fluidlöser basierend auf der Lattice-Boltzmann-Methode, der den vollen Simulationszyklus von der Gittergenerierung bis zur Datenpersistierung abdeckt und eine hochskalierende Parallelisierung auf Multi- und Many-Core-Architekturen erlaubt.
- *FEAST*(TUDo) ist ein Finite Element Framework für massiv parallele Multi-Physik-Anwendungen und enthält neben seinen numerisch hochgradig robusten Komponenten auch Backends für Hardwarebeschleunigung u.a. mit GPGPUs.

Über dieses Anwendungsspektrum hinweg besitzen LB-Verfahren einen gemeinsamen methodischen Kern, ähnliche algorithmische Grundmuster sowie Datenstrukturen und beruhen auf sehr ähnlichen numerischen Kernroutinen. Zahlreiche Projektaufgaben waren daher als stark integrierendes Element angelegt und wurden gruppenübergreifend durchgeführt:

- Portierung und Optimierung für verfügbare High-End-Systeme und Bereitstellung von Benchmarkkonfigurationen für Rechenzentren der Gauß-Allianz und das Gauß-Center for Supercomputing (GCS).
- Entwicklung skalierbarer und effizienter LB-spezifischer Gebietszerlegungsansätze und Lastbalanzierungsmethoden.
- Etablierung eines strukturierten Performance-Engineering-Ansatzes.
- Evaluierung neuer Hardwarekonzepte und (prototypische) Implementierung von LB-Verfahren mit besonderem Schwerpunkt auf parallelen GPGPU-Clustern.
- Nutzbarkeit neuer Programmiermodelle wie hybrides MPI/OpenMP oder PGAS-Sprachen.
- Methodische Weiterentwicklung der LB-Verfahren hinsichtlich lokaler Verfeinerung, Randbedingungen, Kollisionsmodelle.
- Evaluierung der Möglichkeiten eines FEM-Ansatzes für LB-Verfahren.
- Optimiertes Workflowmanagement bezüglich Pre-/Postprocessing Computational Steering und Checkpoint-Restart Mechanismen.
- Benchmarking und Validierung industrierelevanter Strömungsprobleme.

Aufgrund der regulären Grundstruktur von LB-Verfahren wird im akademischen Umfeld erwartet, dass viele der in SKALB gewonnenen Erkenntnisse auch auf andere numerische Verfahren übertragbar sind, insbesondere wenn sie mit einem expliziten Zeitschritt auf Normzellen arbeiten und nur nächste oder übernächste Nachbarn bei der Kommunikation berücksichtigen.

1.2 Randbedingungen und Voraussetzungen

Bei der methodischen Entwicklung und der ingenieurmäßigen Anwendung von LB-Verfahren ist Deutschland neben den USA und Frankreich (letztere speziell für methodische Fragestellungen) federführend. Allgemein ist jedoch zu beobachten, dass in „schnellen“ und „skalierbaren“ LB-Lösern derzeit nur „einfache Physik“, „einfache Numerik“ und „einfache Gitterstrukturen“ implementiert sind – mit entsprechenden Einschränkungen an die zu modellierende Komplexität. Bei Forschungscodes, die sowohl in physikalischer als auch in numerischer Hinsicht deutlich weiter entwickelt sind, werden HPC-Aspekte dagegen weitgehend vernachlässigt.

Das interdisziplinäre Softwareprojekt SKALB, das zwischen akademischer Forschung und industrieller Anwendung anzusiedeln ist, sollte laufende methodische Forschungsarbeiten im LB-Umfeld mit neuesten Techniken des High-Performance-Computing zusammenbringen.

Da zu Beginn des SKALB-Projekts bei allen universitären Projektpartnern bereits langjährig entwickelte LB-Simulationscodes vorlagen, die auf spezifische Anwendungsgebiete optimiert sind, war es von Anfang an nicht geplant, im Rahmen des Projekts SKALB eine gemeinsame Codebasis zu schaffen. Vielmehr sollten die bestehenden Simulationsprogramme so weiterentwickelt werden, dass sie in ihrem jeweiligen Anwendungsbereich eine Spitzenposition einnehmen, also eine wesentlich höhere Effizienz als verfügbare kommerzielle und akademische CFD-Software aufweisen, andererseits aber auch die notwendige Flexibilität und Robustheit hinsichtlich industrieller Aufgabenstellungen besitzen.

1.3 Planung und Ablauf des Vorhabens

Das Projekt SKALB wurde vom 1.1.2009 bis zum 31.12.2011 durch das BMBF gefördert. Die Arbeiten gliederten sich in fünf Arbeitspakete:

- AP1: Portierung und Optimierung von LB-Applikationen auf massiv parallele HPC-Systeme (**RRZE**, LSS, iRMB, TUD_o, HLRS)
- AP2: Weiterentwicklung von LB-Methoden für praktische Anwendungen auf hochskalierenden Systemen (RRZE, **LSS**, iRMB, TUD_o, HLRS)
- AP3: Verbesserte numerische Ansätze für LB-Methoden (LSS, iRMB, **TUD_o**)
- AP4: Hardwarenahe Implementierung für Nicht-Standard-Prozessoren (RRZE, LSS, **iRMB**, TUD_o)
- AP5: Benchmarking und Showcases (RRZE, LSS, iRMB, TUD_o, **IANUS**)

Die Gesamtkoordination des Projekts SKALB lag bei Prof. Wellein vom RRZE an der FAU. An den meisten Arbeitspaketen waren alle universitären Projektpartner beteiligt. Der für ein Arbeitspaket hauptverantwortliche Projektpartner ist in der obigen Liste hervorgehoben.

Das Ziel der Arbeitspakete AP1 und AP4 war es, numerische „Building Blocks“ zu erstellen und auszutauschen; ferner sollten Kernelroutinen zum systematischen Test von Implementierungsalternativen und Skalierbarkeitseffekten entwickelt und verwendet werden. In den Arbeitspaketen AP2 und AP3 sollten existierende Codes methodisch und algorithmisch unter konsequenter Berücksichtigung hoher Parallelisierungsgrade weiterentwickelt werden. Die industrielle Relevanz der im Rahmen dieses Projekts entwickelten Lösungen sollte schließlich in Arbeitspaket AP5 unter Federführung des KMU-Projektpartners IANUS GmbH nachgewiesen werden.

Die Vernetzung und Koordination innerhalb des Projekts erfolgte über gemeinsame Projekttreffen, Telefon- und Videokonferenzen sowie Mailinglisten, Wikis und einen BSCW-Dokumentenserver, aber auch durch koordinierte Konferenzteilnahmen.

Die Aussendarstellung erfolgte über den Webauftritt www.skalb.de, der die wesentlichen Ziele und Arbeitsschritte des Projekts darstellt und in weiten Teilen zweisprachig (deutsch/englisch) ist.

1.4 Wissenschaftlicher und technischer Stand, an den angeknüpft wurde

Durch die amerikanische Firma EXA Corp. wird das kommerzielle Produkt „PowerFlow“ vertrieben, das insbesondere in der Automobilindustrie verbreitet eingesetzt wird. Dieser LB-basierte Löser zeichnet sich vor allem durch ein sehr anwenderfreundliches und automatisches Pre- und Postprocessing aus. Details der methodischen, numerischen und algorithmischen Ansätze sind jedoch unbekannt.

Im Rahmen des englischen Reality-Grid-Projekts konnte eine Gruppe um Prof. Coveney sehr erfolgreich und medienwirksam hochskalierende LB-Simulationen („TeraGyroid“) demonstrieren und hierbei auch ein rudimentäres Computational Steering ermöglichen. Wie auch bei einigen hochskalierenden Show-Cases der Antragsteller des vorliegenden Projekts hat sich die Gruppe um Prof. Coveney auf einfache Ansätze konzentriert und durch die bloße Menge an Rechenleistung einen Rekord erzielt.

Palabos¹ und das OpenLB-Projekt² sind eine Gemeinschaftsarbeit mehrerer internationaler Forschungsgruppen (v.a. Uni Genf, Uni Karlsruhe und Tufts University). Palabos und OpenLB sind als modulares, erweiterbares und plattformunabhängiges System für LB-Simulationen konzipiert. Sie stellen eine gemeinsame Codebasis für Forscher dar,

¹<http://www.palabos.org>

²<http://www.openlb.org>

die Forschungsergebnisse vergleichen und austauschen wollen und bietet einige Werkzeuge zur Vor- und Nachbehandlung von Simulationsdaten. Dabei spielt die hochskalierende Parallelisierung und Performance-Optimierung für verschiedene Architekturen eine untergeordnete Rolle.

Im Rahmen eines früheren BMBF-Projekts („HPSC“) wurde die Parallelisierung und Optimierung des am Fraunhofer-Instituts für Wirtschafts- und Technomathematik in Kaiserslautern entwickelten LB-Lösers „ParPac“ vorangetrieben, so dass für die dort anvisierten Anwendungsbereiche (z.B. Gieß- und Füllprozesse) ein marktreifer deutscher Produktionscode zur Verfügung steht.

Grundlage für das Projekt SKALB waren die vier LB-Löser der Projektpartner, die im Folgenden genauer beschrieben sind.

1.4.1 Ausgangssituation VirtualFluids

Bei *VirtualFluids* des iRMB handelt es sich um einen plattformunabhängigen 2D/3D LB-Code auf hybriden Gittern (hierarchisch/blockstrukturiert), die durch Quad-/Octrees mit uniformen Matrizen als Blättern und einem lokalen Gitterlevelunterschied von eins realisiert sind. Der Code ist in C++ geschrieben, umfasste vor Projektbeginn 2 135 Dateien und bestand aus 300 000 Code-Zeilen und 100 000 Kommentarzeilen. Der Designansatz verfolgt folgende Konzepte:

- Größtmögliche Generalisierung/Abstrahierung der Algorithmen/Pakete/Module u.a. durch generische Programmierung (Gitter-/Strömungsanpassung mittels Strategien: durch Verwendung von Model-Traits gelten die Strategien immer für alle Modelle, wodurch eine minimale Code-Redundanz erreicht wird).
- Verallgemeinertes Kommunikationskonzept zur Minimierung von Redundanz und Parallelisierungsaufwand bzgl. neuer numerischer Kernel (Transmitter-Connector-Konzept).
- Effizientes Mapping von Geometrien auf das Berechnungsgitter mittels Raytracing-Methoden und optimierter Verschneidungsalgorithmen.
- Austauschbare Kommunikationsmodule (MPI, RCF und JADE) für die parallele Berechnung.
- Adaptive Gitteranpassung.

1.4.2 Ausgangssituation waLBerla

Das waLBerla-Framework (**w**idely **a**pplicable **l**attice **B**oltzmann flow solver from **E**rlangen) des LSS stellte bereits zu Projektbeginn einen parallelen 3D-LB-Löser mit flexibel anpassbarer Funktionalität für verschiedene strömungsmechanische Anwendungen dar [FGD⁺07]. Es stützt sich dabei auf das D3Q19-Diskretisierungsschema,

nutzt verschiedene Kollisionsmodelle (BGK, TRT, MRT) und unterstützt vielfältige Randbedingungen, zeit- und ortsabhängige Einflussbedingungen, Beschleunigung und Druckrandbedingungen. Der C++-Code besitzt eine Programmierer-Dokumentation und ist plattformunabhängig. Die Flexibilität des Codes bezieht sich auf verschiedene Gesichtspunkte:

- Flexible Integration weiterer Anwendungen/Funktionalitäten:
Durch ein Konzept zur Minimierung der Code-Redundanz, wie auch generische und objektorientierte Design-Patterns wird die einfache Integration und Parallelisierung neuer Funktionalität ermöglicht.
- Flexible Spezialisierung von Funktionalität zur Effizienzsteigerung:
Komplexe Funktionalität wird nur dort genutzt, wo sie auch benötigt wird. Durch generische Datenstrukturen, die das kartesische LB-Gitter in hierarchische blockstrukturierte Unterregionen (Blöcke) zerlegen, können räumlich limitierte Gebiete mit verschiedener Funktionalität ausgestattet werden. Dies hat entscheidenden Einfluss auf die Recheneffizienz, da algorithmischer Overhead weitgehend vermieden werden kann.
- Flexibler Datenaustausch:
Die Blöcke dienen als Grundstruktur neben der gebietsweisen Anpassung an die jeweilige Funktionalität auch zum parallelen Datenaustausch und als geplante Schnittstelle für heterogene Computerarchitekturen. Der gleiche Ansatz wird auch von TUDo verwendet. Bisher ist es nur möglich, uniforme Blöcke ohne Gitterverfeinerung und mit statischer Funktionalität zu erzeugen, eine entsprechende Erweiterung basierend auf der Vorarbeit des iRMB ist geplant. Auch die für die Parallelisierung nötige Verteilung der Blöcke auf Prozesse erfolgt derzeit statisch. Da die Implementierung der MPI-Parallelisierung es erlaubt, gitterbasierte Daten beliebigen Typs und beliebiger Größe zu versenden, können nicht nur anwendungsspezifische LB-Daten versandt werden, sondern auch ganze Blöcke, um eine dynamische Lastverteilung oder dynamische Veränderung anwendungsspezifischer Funktion zu realisieren. Die optimale Größe der Blöcke dafür soll in Zusammenarbeit mit dem HLRS ermittelt werden (siehe Arbeitspunkt I.c).

Mit der bereits realisierten Funktionalität war es möglich, Strömungen in porösen Medien, Strömungsprobleme mit freien Oberflächen, Diffusionsprobleme, partikelbeladene Strömungen (Fluid-Struktur-Interaktion für bewegte Starrkörper), Brownsche Molekularbewegung (fluctuating LB) und Strömungen in Blutadern zu simulieren.

1.4.3 Ausgangssituation ILBDC

Die beiden Rechenzentren, HLRS und RRZE, haben sich vor einigen Jahren dem *International Lattice Boltzmann Development Consortium* (ILBDC) angeschlossen und sich darin speziell mit Performanceaspekten und der Parallelisierung beschäftigt. Der ab 2002 von Grund auf neu entwickelte ILBDC-Code wurde bereits in diversen Forschungsprojekten (z.B. den FP6-Vorhaben @neurIST und COAST) eingesetzt.

Der ILBDC-Code basiert auf dem D3Q19-Modell in unterschiedlichen Ausprägungen (LBGK, TRT und MRT). Erweiterungen für nicht-newtonsche Fluide (Carreau-Yasuda Model), Turbulenzmodellierung (Smagorinsky LES-Ansatz), Energietransport, Diffusion und geometrische Randbedingungen zweiter Ordnung sind produktionsreif vorhanden oder in Arbeit. Lokale Gitterverfeinerung ist dagegen noch in der Planungsphase.

Bei den Datenstrukturen wurde der strukturierte Ansatz vollständig aufgegeben, indem nur noch Fluidzellen und deren Nachbarschaftsbeziehung abgespeichert werden. In dieser Hinsicht unterscheidet sich der ILBDC-Code grundsätzlich von den patchbasierten LB-Implementierungen, die von den anderen Gruppen in SKALB eingebracht werden. Bei hochkomplexen Geometrien bedeutet der ILBDC-Ansatz natürlich eine signifikante Speicherersparnis, die allerdings mit indirekten Feldzugriffen im Kernel erkauft wird. Das Datenlayout kann je nach Rechenarchitektur gewählt werden, um beispielsweise auf Datenlayoutebene bereits eine Cacheoptimierung zu erreichen [WZDH06]. Durch den „unstrukturierten“ aber dennoch auf kartesischen Gittern basierenden Ansatz ist eine sehr flexible Gebietsaufteilung möglich [Zei08], die jedoch nur mit zusätzlichem Aufwand dynamisch verändert werden kann.

1.4.4 Ausgangssituation FEAT* und FEAST*

Die numerischen und algorithmischen Vorarbeiten der AG Turek (TUDo) liegen schwerpunktmäßig auf der Lösung der Navier-Stokes-Gleichungen und sind im OpenSource-Strömungscode *FEATFLOW* zusammengefasst. Sie zielen auf sehr leistungsstarke hierarchische Mehrgitterlöser und adaptive FEM-Diskretisierungen für komplexe Fluide ab und folgen modernen „Pressure Schur Complement“ Ansätzen [Tur99]. In einem weiten Bereich konnten so Konfigurationen, die prototypisch für industrielle Probleme sind, mit hoher numerischer Effizienz und Robustheit erfolgreich gelöst werden.

Analog zu *FEATFLOW* wurde im Vorfeld prototypisch der LB-Code *FEATLBM* entwickelt, der ebenfalls auf der Finite-Element Basisbibliothek *FEAT* basiert. Mit der Hilfe dieses Codes wurde algorithmische Grundlagenforschung zur Behandlung der LB-Gleichung betrieben, insbesondere mit modernen numerischen Verfahren für partielle Differentialgleichungen (vollimplizite Zeitschrittverfahren hoher Ordnung, unstrukturierte Gitter, Krylow-Raum- und Mehrgitterverfahren, Newton-Löser zur Behandlung der Nichtlinearitäten etc.).

Parallel zu diesen eher mathematisch orientierten Codes wurde das HPC-Paket *FEAST* entwickelt [Bec07], das die Flexibilität, numerische Leistungsfähigkeit und Robustheit von *FEAT* und *FEAT*-basierten Applikationen um Techniken der hardwareorientierten, massiv parallelen Numerik [TBK06] erweitert. Als Lösungsverfahren kommt hierbei ein hocheffizienter ScaRC-Löser [KT98, Kil01] als verallgemeinertes paralleles Mehrgitter-/ Gebietszerlegungsverfahren zum Einsatz.

Mit dieser Ausgangssituation war es ein wesentliches Ziel innerhalb von SKALB, diese numerischen und HPC-Komponenten als Module für ein neues OpenSource-Software-Framework, das sowohl die numerischen Features von *FEAT* als auch die HPC-Fähigkeiten des ersten *FEAST*-Paketes hat, zusammen zu führen. Des Weiteren sollte dieses neue Framework als Komponenten weiterentwickelte Versionen von *FEATFLOW* und *FEATLBM* enthalten. Dieses neue Softwarepaket trägt den tradierten Namen *FEAST* und beinhaltet dementsprechend die Module *FEASTFLOW* und *FEATLBM*, vergleiche Abschnitt 2.1.6.

1.5 Zusammenarbeit mit anderen Stellen

Die assoziierten Partner Intel, Cray und IBM sowie zusätzlich Fujitsu haben die Projektarbeit durch Bereitstellung von Early-Access-Testsystemen am RRZE sowie durch technischen Experten nachhaltig gefördert. NLE-IT hat bis zur Auflösung des Forschungslabors in St. Augustin die Implementierung skalierbarer LB-Algorithmen aktiv begleitet.

Die BASF AG hat einen industrierelevanten Anwendungsbenchmark aus dem Bereich der chemischen Verfahrenstechnik gestellt, der in AP5 von allen beteiligten Projektpartnern intensiv bearbeitet wurde. Der von der Sulzer Chemtech AG vorgeschlagene Benchmarkcase konnte nicht untersucht werden, da für die dabei zu simulierende Mehrphasenströmung umfangreiche physikalische Modellentwicklungen nötig gewesen wären, die nicht Gegenstand des Projekts waren.

Es wurde von Seiten der AG Turek (TUDo) mit dem SINTEF in Oslo erfolgreich an einer gemeinsamen Codeschnittstelle zum Vergleich von LB-Codes gearbeitet.

Zum Intel Exascale Laboratory unter der Leitung von Prof. Jalby an der Universität Versailles Saint-Quentin-en-Yvelines bei Paris konnte ebenso ein intensiver Kontakt aufgebaut werden, wie zu Prof. Aoki vom Tokyo Institute of Technology, über den auch Zugang zum japanischen Tsubame-2.0 Supercomputer erhalten werden konnte. Jan Treibig und Markus Wittmann vom RRZE haben mehrmonatige Forschungsaufenthalte in Versailles bei Prof. Jalby verbracht und Christian Feichtinger (LSS) wird in Kürze eine PostDoc-Stelle bei Prof. Aoki antreten.

Im Rahmen der Gauß-Allianz stand das RRZE in regelmäßigem Kontakt mit anderen Gruppen bundesweit.

Ergebnisse aus dem Projekt SKALB sind in Blockkurse eingeflossen, die das RRZE regelmäßig zusammen mit dem Leibniz Rechenzentrum in München und dem Höchstleistungsrechenzentrum in Stuttgart durchführt.

Kapitel 2

Eingehende Darstellung des Projekts

2.1 Verwendung der Zuwendung und erzielte Ergebnisse

Die Zuwendung wurde verwendet, um Doktoranden und PostDocs in den beteiligten Gruppen zu finanzieren, welche die LB-Codes der jeweiligen Gruppen im Sinne des BMBF-Calls für hochskalierende HPC-Systeme weiterentwickelt haben. Die durchgeführten Arbeiten gliedern sich in fünf Arbeitspakete, die von der Portierung und Optimierung der Codes (AP1) über algorithmische (AP2) und methodische Weiterentwicklungen (AP3) sowie die Nutzung alternativer Hardware (insbesondere GPGPUs, AP4) bis hin zur exemplarischen Anwendung auf industrierelevante Anwendungen (AP5) reichen. In den folgenden Unterabschnitten wird auf die durchgeführten Arbeiten der einzelnen Arbeitspunkte genauer eingegangen, wobei jeweils auch die ausführende Gruppe angegeben ist.

Aufgrund des theoretischen und ergebnismäßigen Umfangs der einzelnen Arbeiten werden in den folgenden Abschnitten bei der einzelne Arbeiten und Ergebnisse exemplarisch herausgegriffen und werden näher ausgeführt. Weitergehende Informationen finden sich in den projektbezogenen Publikationen der Projektpartner, auf die häufig verweisen wird.

2.1.1 AP1: Portierung und Optimierung von LB-Anwendungen auf massiv parallele HPC-Systeme

Portierung und Optimierung

Ziel von AP1 war es zunächst, die verwendeten Codes auf die unterschiedlichen zur Verfügung stehenden HPC-Systeme zu portieren und dort die Performance der in das Projekt eingebrachten LB-Applikationen zu optimieren. Neben lokalen HPC-Clustern aller beteiligten Einrichtungen und den Tier-1 Systemen an den Bundeshöchstleistungszentren in Jülich, München und Stuttgart standen über die europäischen Projekte DEISA und PRACE zusätzliche Ressourcen bei CINECA in Italien, am EPCC in England sowie beim Barcelona Supercomputing Center bereit. Hervorzuheben ist, dass auch hochskalierende Systeme in den USA (z.B. am National Energy Research

Scientific Computing Center (NERSC) in Berkeley) sowie eines der größten GPGPU-Systeme (Tsubame 2.0 am Tokyo Institute of Technology) für Testläufe genutzt werden konnten.

Nach der Portierung wurden die unterschiedlichen Systeme sowohl für Skalierungs- und Benchmarkuntersuchungen aber auch im Rahmen komplementärer Projekte für Produktionsläufe verwendet. Als eines von nur acht Teams wurde der LSS im Februar 2010 mit waLBerla zum „Extreme Scaling Workshop“ nach Jülich eingeladen und hatte so die Möglichkeit, Skalierungsmessungen und Programmoptimierungen auf der gesamten Jugene (eine IBM Blue Gene/P mit knapp 300.000 Cores) durchzuführen [9]. Durch die Vorarbeiten im Rahmen von SKALB ist davon auszugehen, dass der derzeit schnellste Rechner Deutschlands, SuperMUC am LRZ, innerhalb kürzester Zeit auch produktiv genutzt werden kann.

Die Portierung von *FEAST* durch TUDo auf die Cell-BE Zielplattform nahm eine Sonderstellung ein: Während andere Portierungen lediglich minimale Maßnahmen erforderten, war die Entwicklung einer Laufzeitbibliothek zur Nutzung der SPEs der Cell-BE erforderlich. Zudem mussten von IBM extern bereitgestellte Cell-Blades verwendet werden. Leider hat IBM während der Projektlaufzeit bekanntgegeben, dass die Hardware der Cell-BE nicht weiterentwickelt wird, so die Aktivitäten in diesem Bereich reduziert wurden. Im Rahmen der Forschung im Bereich *Unconventional HPC und Energieeffizientes Rechnen* hat TUDo dafür zuletzt einen LB-Code der Arbeitsgruppe auf einen prototypischen Cluster des Barcelona Supercomputing Center portiert, der auf low-power ARM Kernen basiert.

Implementierungs- und Parallelisierungskonzepte, Performance-Engineering und Performancemodelle

Nach den Portierungen galt es, in einem zweiten Schritt, anhand von überschaubaren Benchmark-Kernen neue Implementierungs- und Parallelisierungskonzepte zu evaluieren und durch einen strukturierten Ansatzes für das Performance-Engineering über Performancemodelle die zu erwartende Performance abzuschätzen und das Optimierungspotential einzuschätzen.

Das RRZE hat sich auf die Aspekte Performancemodelle und Performance-Engineering [10, 25] sowie die Implementierung (einfacher) Benchmarkkernel mit SSE/AVX-Intrinsics für x86_64-CPU's [10] konzentriert. Wavefront- und Multi-Halo-Parallelisierungsansätze wurden als spezielle Ansätze für Multi-Core-Implementierungen untersucht [12, 21, 23, 24]. Mit optimierten Benchmarkkernen konnte annähernd die durch die Performancemodelle theoretisch vorhergesagte Performance erreicht werden. Die Benchmarkkernel wurden anschließend weiterentwickelt und beispielsweise als Teil der SPEC-OpenMP-Suite submittiert oder sind in die Benchmarksuiten zur Beschaffung von HPC-Systemen am RRZE und dem LRZ eingeflossen. Die Benchmarkkernel wurden aber auch in Produktionscodes übernommen und erreichen sowohl im ILBDC als auch waLBerla annähernd die gleiche Performance

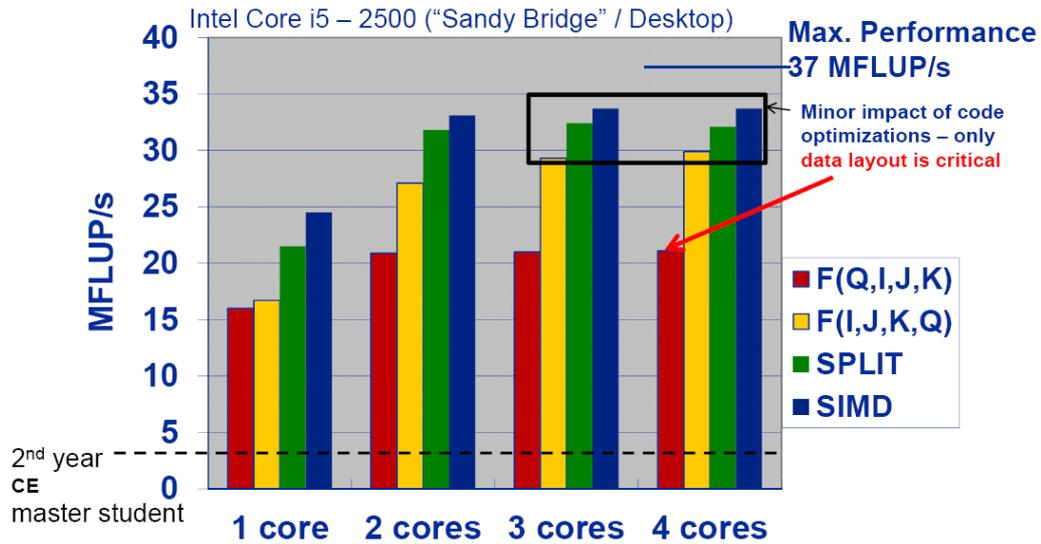
Double precision (DP): Lid driven cavity (230³)

Abbildung 2.1: Erzielte Performance in Abhängigkeit von der gewählten Implementierung und der Zahl der verwendeten Kerne für Berechnungen in doppelter Genauigkeit.

wie in den eigenständigen Benchmarkkernen [5, 10, 22]. Ausgangspunkt für jegliche Optimierung muss auch im Exascale-Zeitalter weiterhin der einzelne Kern und einzelne Rechenknoten sein. Abbildung 2.1 und 2.2 zeigen daher die auf einem Single-Socket-System erzielte Performance in Abhängigkeit von der gewählten Implementierung und der Zahl der verwendeten Cores. Mit optimierten Implementierungen (SIMD bzw. teilweise auch SPLIT) wird dabei bereits mit nur einem Teil der verfügbaren Kerne im Knoten eine Sättigung der Performance erreicht, wobei die erzielte Leistung nahe an die aufgrund der aggregierten Speicherbandbreite vorgegebenen maximale Performance heranreicht [10, 25]. Die schlechte Skalierung innerhalb eines Knotens ist somit kein Zeichen dafür, dass schlecht implementiert wurde, sondern für genau das Gegenteil und die Unausgeglichenheit der Hardware mit hoher Rechenleistung aber geringer Speicherbandbreite. In der Praxis kann die Tatsache, dass bereits wenige Kerne die maximale Anwendungsleistung auf dem Knoten erreichen, ausgenutzt werden, indem man nicht benötigte Kerne in einen Schlafmodus versetzt und somit den Stromverbrauch reduziert.

Vom LSS wurde ein Performancemodell für den Kommunikationsaufwand für Standard- und Nicht-Standard-Architekturen erstellt. Hiermit konnte unter anderem die zu erwartende Performance auf Tsubame 2.0 vorhergesagt und mit der gemessenen Performance verglichen werden [5]. Auftretende signifikante Abweichungen bei den ersten Messungen konnten auf Hardwareprobleme zurückgeführt werden, haben aber auch die Notwendigkeit der Überlappung von Kommunikation und Berechnung deutlich gemacht [5], siehe auch Abbildung 2.3. Ferner konnte der Overhead bei hybrider und heterogener Ausführung evaluiert werden. Es hat sich gezeigt, dass es lohnenswert ist,

Single precision (SP): Lid driven cavity (230³)

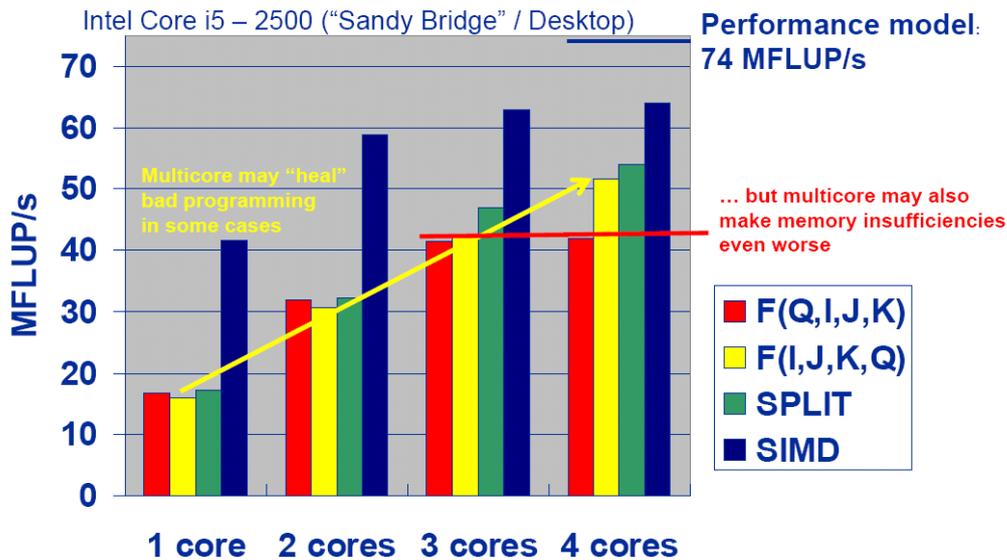


Abbildung 2.2: Erzielte Performance in Abhängigkeit von der gewählten Implementierung und der Zahl der verwendeten Kerne für Berechnungen in einfacher Genauigkeit.

GPGPUs nicht nur im „Accelerator Mode“ zu betreiben, sondern die Rechenleistung der CPU-Kerne im Knoten auch mitzubeneutzen.

Durch Kernel-Prototypen für parallele Programmiermodelle wurden von TUDO insbesondere im Bereich von Basiskernen (SIMD/Multi-Core) [33, 37, 48], aber auch bei Gebietszerlegungs- und parallelen Mehrgitterverfahren weitreichende Ergebnisse erzielt [34–36, 39, 40, 49, 50]. Basierend sowohl auf OpenMP, als auch auf PThreads wurden LBM- und FEM-Kernel portiert, optimiert und evaluiert [33, 37, 48]. Ferner wurden die in *FEAST* eingesetzten SBBLAS-Bibliotheken für effiziente numerische lineare Algebra vor dem Hintergrund des Einsatzes Finiter Elemente höherer Ordnung sowie nicht konformer Finite Element Räume erweitert und evaluiert.

Von TUDO wurden außerdem in den Bereichen hybride Ausführung, Laufzeitumgebungen und Softwaretechnik substantielle Fortschritte erzielt: [33–37, 48]

- Es wurden sehr erfolgreiche Softwarebibliotheken für die hybride Ausführung (MPI in Verbindung mit OpenMP/PThreads oder GPGPU (beispielsweise CUDA oder OpenCL)) für *FEAST* entwickelt.
- Laufzeitumgebungen für die Speicherkonsistenz, das Job-Scheduling, RTTI für die Cell-BE, und Speichertransfers bei Einsatz von GPGPUs und hybrider Ausführung wurden entwickelt und finden inzwischen Einsatz in *FEAST*.
- Für geometrische Mehrgittermethoden wurden neuartige Funktor-basierte Operatorketten entwickelt, die innerhalb des Lösungsvorgangs Rekursionen minimieren sollen.

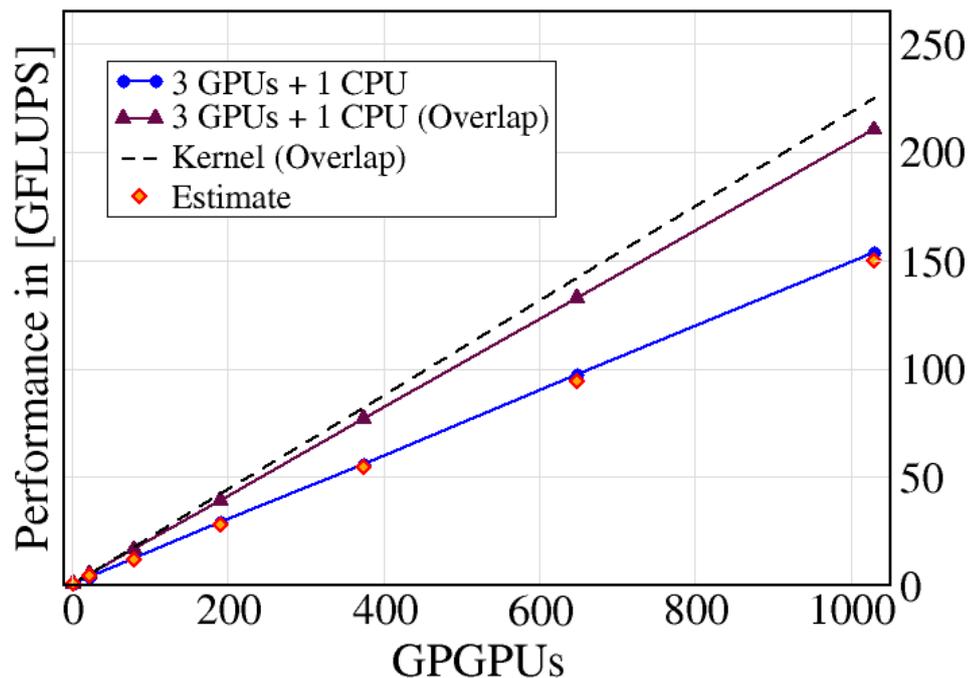


Abbildung 2.3: Heterogene Weak-Scaling-Performance von waLBerla auf bis zu 1029 GPGPUs des Tsubame 2.0-Clusters. Die blaue Kurve enthält keine Überlappung von Berechnung und Kommunikation. Durch explizite Überlappung wird eine signifikante Performancesteigerung erreicht (magenta Kurve), die sehr nahe an die extrapolierte Kernel-Performance einer GPGPU herankommt (gestrichelte schwarze Kurve).

- Für die numerische lineare Algebra (vormals SBBLAS) von *FEAST* wurde ein Frontend entwickelt, das die Applikationsentwicklung/ -anpassung insofern erleichtern soll, als dass es die Vorzüge von C++ Operator-Overloading mit performanter kernelbasierter Ausführung kombinieren kann (hardware-orientierte Expression Templates).

Die Nutzung hybrider Programmiermodelle (insbesondere OpenMP+MPI aber auch CPU+GPGPU) hat sich bei allen SKALB-Gruppen insgesamt bewährt. PGAS-Sprachen wie UPC oder Co-Array Fortran stecken dagegen noch in den Kinderschuhen und sind (noch) keine Alternative zu MPI. Insbesondere ist ein MPI-artiger Programmieransatz mit expliziter Gebietszerlegung und HALO-Austausch nötig, um mit PGAS-Sprachen eine mit MPI vergleichbare Performance zu erzielen [19, 63].

Die sogenannte „Adaptive Data and Communication Library“ (ADCL), die von der Universität Houston und dem HLRS entwickelt wird und zur Laufzeit die optimale Wahl des zu verwendenden MPI-Kommunikationsschemas treffen soll, wurde am HLRS evaluiert. Da für die untersuchte Lattice-Boltzmann-Applikation kein signifikanter Ge-

winn festgestellt werden konnte, wurden diese Arbeiten im Rahmen von SKALB nicht weiter verfolgt.

2.1.2 AP2: Weiterentwicklung von LB-Methoden für praktische Anwendungen auf hochskalierenden Systemen

Datenstrukturen

Die Performance von LB-Anwendungen ist meist durch die Speicherbandbreite beschränkt. Grundsätzlich sind verschiedene Datenstrukturen denkbar und in der Literatur auch veröffentlicht, um einerseits den Gesamtdatenumfang zu minimieren und andererseits die Anzahl der Speicherzugriffe zu minimieren. In diesem Kontext wurde vom iRMB ein neues Verfahren entwickelt, welches als „Esoteric Twist“ bezeichnet wird. Dieses ermöglicht die Verwendung von einem Datensatz für die Speicherung der Verteilungen (in den meisten Implementationen werden zwei Felder verwendet, was zu einer Verdoppelung des Speicherbedarfs der Simulation führt). Hinzu kommt die Vereinigung des Kollisions- und Propagationsschritts. Insgesamt wird die Menge der Speicherzugriffe pro Berechnungsschritt signifikant reduziert. Ein entscheidender Vorteil von Esoteric Twist liegt darin, dass man nur einen Verteilungssatz verwendet und die Gitterknoten trotzdem unabhängig voneinander (parallel) bearbeitet werden können. Weiterhin wurde vom iRMB eine kompakte Randbedingung zweiter Ordnung entwickelt, die für das „Esoteric Twist“-Verfahren besonders geeignet ist. Die Randbedingung ist gut für parallele Anwendungen geeignet, weil sie ein sehr lokales Verfahren darstellt, das keine Informationen von Nachbarknoten benötigt.

Sowohl waLBerla als auch VirtualFluids verwenden Blöcke, Untergebiete der Simulationsregion, als grundlegende Datenstruktur zur Gebietszerlegung. Es kann ein Block pro Prozess verwendet werden oder für Lastbalancierungszwecke mehrere Blöcke pro Prozess (Abb. 2.4). Darüber hinaus bietet die Block-Datenstruktur weitere Funktionalität, z.B. die lokale Anpassung der Simulationsdaten an unterschiedliche Architekturen, Speicherlayouts und physikalische Funktionalität. In VirtualFluids werden Blöcke in sogenannten Patches organisiert (Abb. 2.5). Unterschiedliche Adaptivitätslevel werden durch die hierarchische Anordnung der Patches ermöglicht. waLBerla organisiert die Blöcke in einer Octree-ähnlichen Datenstruktur (Abb. 2.6), um Adaptivität zu unterstützen. Weiterhin können Blöcke in waLBerla in nicht-uniforme Sub-Blöcke unterteilt werden, um heterogene Rechenknoten abzubilden. Hierbei entspricht jeder Sub-Block einer heterogenen Hardwareeinheit, z.B. ein Rechenknoten mit zwei CPU-Sockeln und drei GPGPUs allokiert fünf Sub-Blöcke pro regulärem Block, und ermöglicht die statische Lastbalancierung trotz der unterschiedlichen Rechenleistung der heterogenen Hardwareeinheiten. Des Weiteren wurden Datenstrukturen für massive parallele dreidimensionale Simulationen auf GPGPU-Clustern entwickelt. Hierzu werden Puffer auf Seiten der GPGPUs verwendet, um die Daten zunächst zu aggregieren und dann mit Hilfe von einem einzigem PCI-Express-Transfer zum Host zu kopiert. Diese Datenstrukturen ermöglichen massive parallele adaptive dynamisch lastbalancierte Simula-

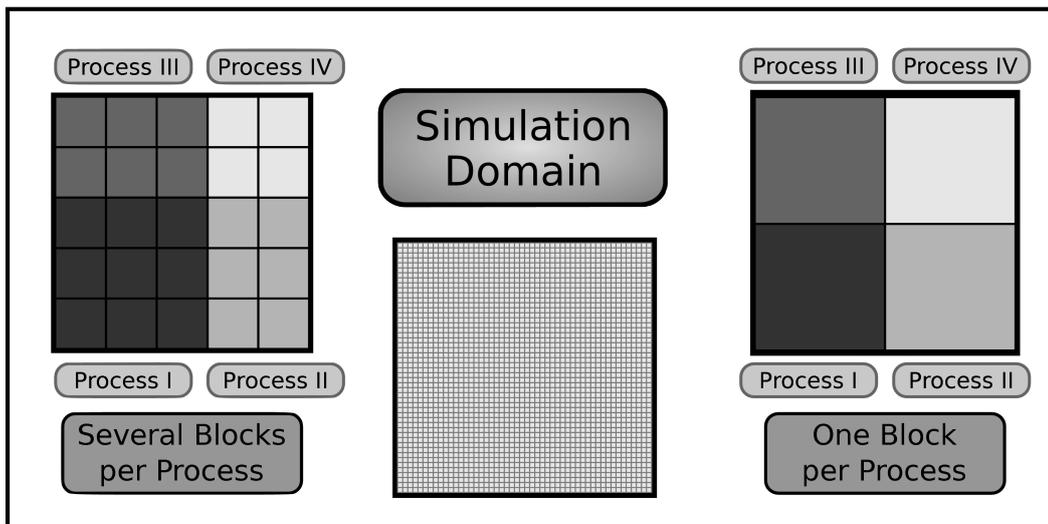


Abbildung 2.4: Gebietszerlegung in Blöcke.

tionen auf Standard- wie auch auf Nicht-Standard-Architekturen, das Überlappen von Berechnung mit Kommunikation und MPI-, hybride oder heterogene Parallelisierungsansätze.

Gebietszerlegung und dynamische/adaptive Lastbalanzierung

Beim ILBDC erfolgt der Zugriff auf die Zellen indirekt über eine Adjazenzliste (Abbildung 2.7). Lediglich Fluidzellen werden allokiert und in einem 1-D Array gespeichert. Durch den *Präprozessor* wird diese Liste auf skalierbare Weise parallel erzeugt, was vollständig entkoppelt von der späteren Strömungssimulation erfolgt, wobei allerdings die Reihenfolge der Fluidzellen die Performance der Strömungssimulation beeinflusst. Eine Art Cacheblocking kann bereits im Präprozessor-Schritt erfolgen, ohne dass Änderungen im Strömungslöser nötig sind [28]. Zur Anordnung der Zellen im Array wurden diskrete raumfüllende Kurven, wie die Z-Kurve, aber auch die lexikographische Sortierung mit Blocking untersucht (Abbildung 2.8), wobei letztere Variante die besten Ergebnisse erzielt. Jedoch ist der richtige Blockingfaktor a priori schwer zu bestimmen und wird deshalb per „Autotuning“ ermittelt [26]. Aufgrund der gewählten Datenstruktur und der indirekten Adressierung der Fluidzellen über die Adjazenzliste ist eine dynamische Anpassung zur Laufzeit nicht leicht möglich.

In VirtualFluids wurde eine serviceorientierte Master-Slave-Architektur durch ein neues Parallelisierungskonzept basierend auf einem dezentralisierten Verfahren inklusive einer Multi-Core-Unterstützung implementiert. Ein MPI-Prozess wird dabei als multithreaded betrachtet, wobei die Anzahl der Threads der Anzahl der Prozessorkerne entspricht. Ein Patch wird im ersten Schritt entsprechend der Anzahl der MPI-Prozesse zerlegt. Dann werden die Blöcke, die zu einem MPI-Prozess gehören, gleichmäßig zwischen Berechnungsthreads verteilt. Ein dedizierter Thread pro CPU-Knoten ist für die

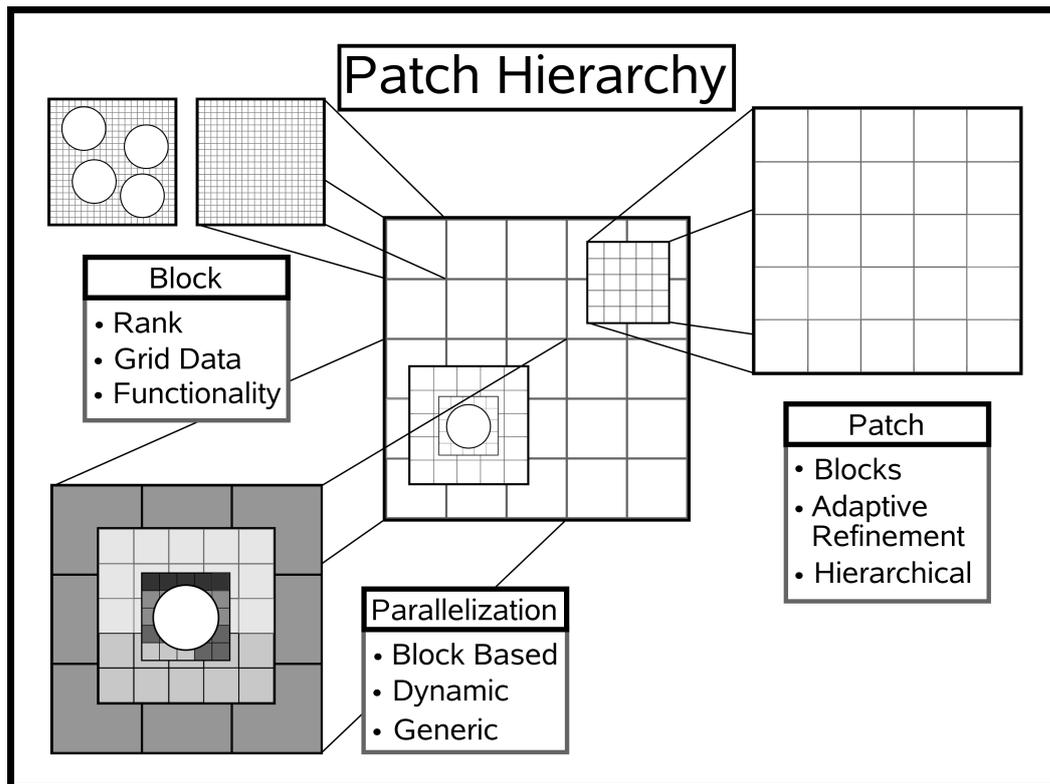


Abbildung 2.5: Datenstrukturen für hochparallele, dynamisch-adaptive Simulationen basierend auf Blöcken. Blöcke speichern Simulationsdaten und auch Zusatzinformationen für die Parallelisierung. Für adaptive Simulationen werden Blöcke in Patches zerlegt, welche hierarchisch angeordnet sind. Jede Patchebene entspricht einem Adaptivitätslevel. Blöcke eines Patches können auf mehrere MPI-Prozesse aufgeteilt werden.

MPI-Kommunikation zuständig. Ferner wurde die Bibliothek Zoltan integriert, welche die Partitionierung und dynamische Lastbalanzierung realisiert. Das ermöglicht die Verwendung verschiedener Partitionierer, z.B. für geometriebasierte oder graphen-, hypergraphenbasierte Gebietszerlegung, sowie Migration der Daten nach der Partitionierung. Ein weiterer Vorteil von Zoltan ist die Unterstützung paralleler Partitionierung, welche die Basis für die Verfahren der dezentralisierten Parallelisierung darstellt. Für die Zerlegung auf Interthreadebene wurde eine schnelle und effektive Alternative der graphenbasierten Partitionierung entwickelt, die auf einem Priority Queue Algorithmus basiert. Die Blöcke haben verschiedene Gewichtungen, die der Anzahl von inneren und äußeren Kommunikationskanälen entspricht. Nach diesen Gewichtungen werden die Blöcke möglichst gleichmäßig zwischen Threads verteilt.

Zur dynamische Lastbalancierung wurde sowohl in VirtualFluids als auch waLBerla eine Methode für massiv parallele und heterogene Umgebung implementiert, die auf einem erweiterten verteilten Diffusionsalgorithmus basiert. Die Initialzerlegung wird

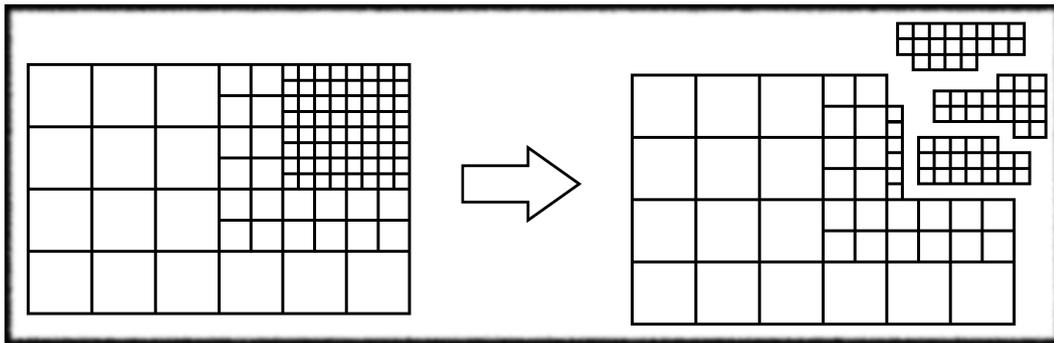


Abbildung 2.6: Gebietszerlegung in waLberla mittels Octree-Datenstrukturen. Jeder Block enthält gleich viele Zellen. Zur dynamischen Lastbalanzierung werden Blöcke je nach ihrer Last auf die MPI-Prozesse verteilt. In der Abbildung ist dies schematisch für vier MPI-Prozesse gezeigt.

mittels einer statischen Partitionierung im Präprozessor durchgeführt, um bessere Bedingungen für eine schnellere Konvergenz der Lastverteilung, zu erhalten. Zur Simulationszeit wird die Last für jeden Prozess ausgerechnet. Die überbelasteten Prozesse schicken einen Teil ihrer Last zu weniger belasteten Nachbarprozessen. Bei der Lastverteilung wird berücksichtigt, dass die Kontaktoberflächen zwischen den Nachbarprozessen immer minimal bleiben.

Ein wesentlicher Bestandteil der an der TUDo entwickelten Ergebnisse bezieht sich auf die Diskretisierung und die Lastbalancierung von *FEAST* vor und während des (parallelen) Lösungsvorgangs. Dabei ist ein auf einer Mischung aus Gebietszerlegung und Mehrgitterverfahren basierendes System zugrundegelegt, wobei die einzelnen Teilgebiete zu Clustern (so genannter Matrixpatches) zusammengefasst werden können. Schwierigkeiten dabei sind die lokal durchaus sehr unterschiedlichen numerischen Erfordernisse, die ein hohes Maß an Flexibilität verlangen. Beispielsweise müssen strukturierte und (im Fall von LBM insbesondere ungewöhnliche) unstrukturierte Gitter zuzüglich entsprechender Adaptivitätsansätze und eine Vielzahl von Datenstruktur-Operationen unterstützt werden. Gleichzeitig müssen statisches und dynamisches Loadbalancing unterstützt werden. Entsprechende Datenstrukturen wurden im Rahmen von SKALB entwickelt.

Preprocessing und Visualisierung

Am HLRS wurde ein SMP-paralleler Voxelizer entwickelt, der neben der Diskretisierung eines Objekts durch Voxel auch die sogenannten „q-Werte“ zur besseren Approximation des Randes [BFL01] berechnet. Zudem wurde ein Ansatz verfolgt, der auf cad2octree und OpenCascade basiert. Cad2octree ist hervorgegangen aus einer Diplomarbeit an der Universität Stuttgart (<http://cad2octree.berlios.de/>), und OpenCascade (<http://www.opencascade.org/>) ist OpenSource. Der Ansatz basiert

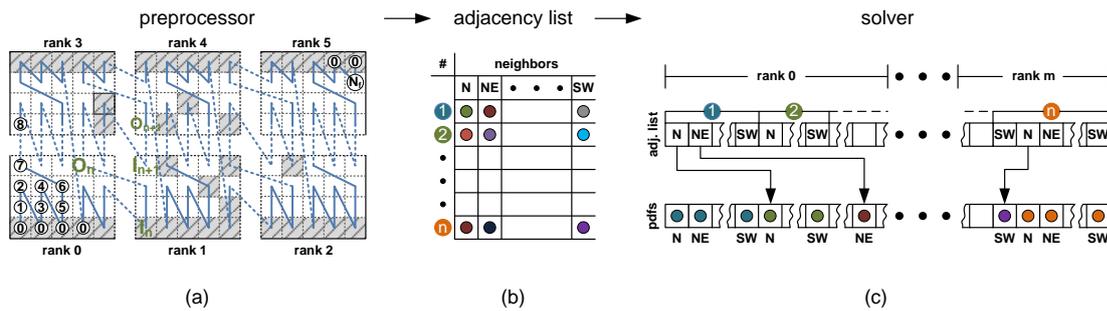


Abbildung 2.7: Durch den Präprozessor wird das Simulationsgebiet geometrisch zerlegt und die resultierenden Partitionen werden Prozessen zugewiesen (a). In diesem Fall wird eine lexikographische Sortierung mit Blockingfaktor 3 verwendet, um die Adjazenzliste (b) der Fluidzellen (weiße Zellen) aufzustellen. Dazu erhalten die Fluidzellen einen global eindeutigen und kontinuierlichen Index, der die Feststoffzellen (dunkle Zellen mit Index 0) ausläßt. Vom Strömungslöser wird diese Liste in gleich große Stücke zerlegt, so dass jeder Prozess gleich viele Fluidzellen besitzt (c). [26]

auf einer hierarchischer Diskretisierung und wurde in einem hybrid-parallelen Programm realisiert, womit mehrere zehn Millionen Voxel verarbeitet werden können. Die voxelisierten Daten wurden von diversen Gruppen erfolgreich für die Simulation der BASF-Geometrie aus AP5 verwendet.

Der Aspekt der Visualisierung wurde primär ebenfalls am HLRS bearbeitet. Zum einen wurde das Computational-Steering-Framework *Steereo*¹ im Rahmen von SKALB u.a. um ein Fortran-Interface erweitert und die parallele Skalierbarkeit konnte verbessert werden. Von den LB-Codes aus SKALB wurde der ILBDC-Solver erfolgreich in das Steereo-Framework integriert. Zum anderen wurde die am HLRS entwickelte und von Visenso GmbH vertriebene Visualisierungssoftware COVISE angepasst, da der in AP5 bearbeitete BASF-Showcase aufgrund seiner Größe zunächst nicht visualisiert werden konnte. Die Daten werden nun auf einem vereinfachten Gitter dargestellt, welches das Datenaufkommen soweit reduziert, dass mehrere Partitionen und mehrere Zeitschritte visualisiert werden können.

Von TUDo wurde ein auf OpenGL- und Qt-basierender Visualisierer zur 3D-Datenvisualisierung und Echtzeitdarstellung von Zeitschrittverfahren entwickelt, der in Lehrveranstaltungen verwendet wird. Ansonsten haben die meisten Gruppen die Anbindung ihrer LB-Solver an die OpenSource-Visualisierungssoftware ParaView verbessert.

¹<http://steereo.hlrs.de/>

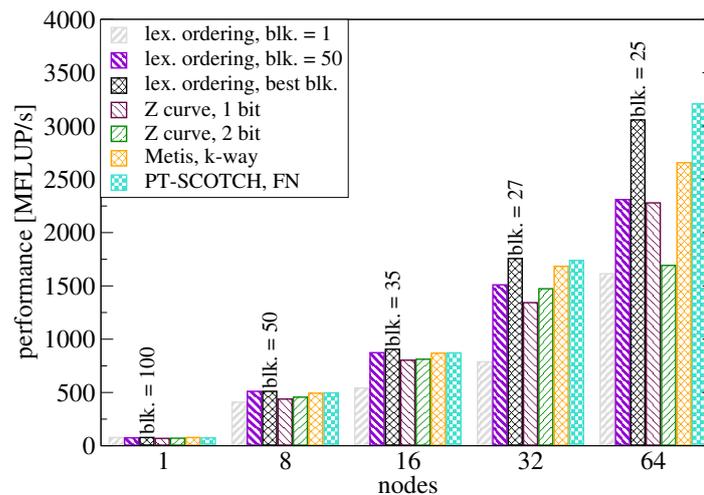


Abbildung 2.8: Strong-Scaling eines Festbettreaktors der Größe $500 \times 100 \times 100$ Zellen mit ungefähr $2,1 \times 10^6$ Fluidzellen auf 1 – 64 Westmere-Rechenknoten (12 MPI-Prozesse pro Knoten). Als Nummerierungschemata wurde eine lexikographische Sortierung mit Blocking und die diskrete Variante der raumfüllenden Z-Kurve (1 und 2 Bit) verwendet, sowie die Graphpartitionierer METIS und PT-SCOTCH [26]

2.1.3 AP3: Verbesserte numerische Ansätze für LB-Methoden

AP3 wurde primär von TUDo und iRMB bearbeitet. Einzelne Ergebnisse (z.B. hinsichtlich Adaptivität) wurden insbesondere vom LSS verwendet und prototypisch in waLberla umgesetzt.

Zentral für die Arbeiten der TUDo in diesem Arbeitspaket war die Auslotung der Möglichkeiten der Anbindung der LBM an modernste Methoden der numerischen Mathematik (z.B. hierarchische Mehrgitterlöser auf adaptiven Rechengittern). Dazu muss man die LB-Gleichung als PDE auffassen und dann entsprechend (ohne die der klassischen LBM zugrunde liegende Partikel-Gasdynamik) in Ort und Zeit diskretisieren. Dazu wurde ein vollimpliziter Ansatz entwickelt (*FEASTLBM*) [41, 42]. Die Short-Characteristic Upwind Diskretisierung von 1. und 2. Ordnung wurde in einer GEF-Formulierung der Lattice-Boltzmann-Gleichung (diskretes Geschwindigkeitsmodell) eingesetzt und effiziente Lösungsmethoden wurden in diesem Rahmen untersucht. Speziell wurde ein Mehrgitter-Löser entwickelt, der einen effizienten monolithischen Zugang für stationäre Probleme ermöglicht. In FeatLBM wurden effiziente und stabile Zeitdiskretisierungen bis zu zweiter Ordnung für zeitabhängige Probleme eingesetzt und Resultate für den instationären Benchmark (Umströmter Zylinder) erreicht, wobei Referenzwerte für Drag und Lift mit grosser Zeitschrittwahl ($\Delta t = 0.01$) akkurat reproduziert wurden.

Wesentliche Komponenten impliziter Löserpipelines sind geometrische Mehrgittermethoden und starke Glätter für unstrukturierte Gitter, die auch mit moderner Beschleunigung

nigerhardware (wie GPGPUs) verträglich sein müssen: Als grundlegende Komponente des Lösungsvorgangs in *FEAST* (und damit für *FEASTLBM*) wurden die auf Finite Elemente Methoden zugeschnittenen Mehrgitterlöser für unstrukturierte Gitter (FE-gMG) insbesondere im Hinblick auf moderne Beschleunigerhardware wie GPGPUs neu konzipiert. In diesem Teilprojekt vereinen sich drei wesentliche Aspekte: (1) Die numerische Effizienz von Mehrgittermethoden selbst und starker Glättungsoperatoren, (2) die Hardwareeffizienz angepasster Komponenten (s.u.) und (3) die Flexibilität unstrukturierter Gitter [34–36].

Innerhalb dieser Mehrgitteroperatoren kommen hochoptimierte Basiskomponenten zum Einsatz. In diesem Fall sind dies effiziente SparseMatrix-Vector Produkte (SpMV) für FE-gMG. Alle Komponenten der oben beschriebenen Mehrgitteroperatoren konnten als Kette von SpMV ausgedrückt werden. Diese grundlegende Operation wurde im Hinblick auf die Hardwareeffizienz (CPU/SSE und Multi-Core sowie insbesondere GPGPU) auf den neuesten Stand der Technik gebracht [34–36].

Eingebettet werden die lokalen Mehrgittermethoden in das *FEAST*-Paket über den parallelen ScaRC (scalable recursive clustering) Löser. Dieser parallele lineare Lösungsprozess in *FEAST*, für den die weiter oben beschriebenen FE-gMG lokale Löser sind und der die beschriebenen Datenstrukturen und Lastverteilungskomponenten verwendet, wurde ebenfalls weiterentwickelt [49, 50].

Zusätzlich zu den alternativen numerischen Zugängen und den zugehörigen Operatoren und Datenstrukturen ist ein wesentlicher Teil der Projektarbeit in die Entwicklung von Hardware-orientierten Standard-LBM und Anwendungen für komplexe gekoppelte Systeme geflossen. Es wurde ein komplettes Framework für Standard-LBM entwickelt, auf alle betrachteten Hardwareplattformen portiert, optimiert und evaluiert [33, 37, 48]. Neben einfachen Flachwassersimulationen ist dieses – in ein Modul von *FEAST* integrierte – Framework auch in der Lage, schwierige gekoppelte Physiksimulationen durchzuführen.

Ebenfalls zentral für *FEAST* sind die verschiedenen Adaptivitätsansätze, die im Rahmen von SKALB hinzugefügt und evaluiert wurden. Hier wurden zwei verschiedene Ansätze umgesetzt. In 2D wurde die Gitterdeformation kombiniert mit konformen, lokalen Verfeinerungen (h-Adaptivität). In 3D verwenden wir reine Gitterdeformation zur Konzentration der Gitterpunkte unter Beibehaltung der logischen Struktur des Gitters (r-Adaptivität). Die logische Struktur des Gitters (Nachbarschaften zwischen Knoten) wird beibehalten, lediglich beispielsweise um die Hülle des Modells werden die Knoten zusammengezogen, um die Genauigkeit und die Auflösung zu erhöhen. Es wurden bei beiden eingesetzten Techniken – Gitterdeformation kombiniert mit konformen, lokalen Verfeinerungen in 2D bzw. reine Gitterdeformation in 3D – wesentliche Fortschritte erzielt. Es konnte beispielsweise gezeigt werden, wie und bis zu welchem Grad die numerischen Ergebnisse von Simulationen mit konformen Verfeinerungen des Gitters verbessert werden können. Es konnte insbesondere gezeigt werden, dass bei manchen Problemen bei nicht ausreichender Auflösung bzw. ohne adaptive Verfeinerung der Fehler signifikant steigt (beispielsweise durch Nachweis des Ausbleibens von Diffusion, vergleiche insbesondere Zwischenbericht 3).

Datenreduktion Die kontinuierlich steigende Leistungsfähigkeit von Hochleistungsrechnern ermöglicht auch im Bereich numerischer Strömungssimulationen immer größere Systeme und bedingt damit gleichzeitig ein stetiges Wachstum der Ergebnisdatensätze. Schon heute umfasst die Größe dieser Daten leicht mehrere TeraBytes und es ist abzusehen, dass schon bald die PetaByte-Grenze überschritten werden wird. Die Verarbeitung und Analyse dieser kontinuierlich wachsenden Datenmengen stellt sowohl für traditionelle, im Post-Processing-Verfahren arbeitende Simulationssysteme als auch für interaktive Ansätze eine zunehmend schwierigere Aufgabe dar. Ein typischer Ansatz zur Balancierung besteht darin, die Analyse des Ergebnisraums auf einen Ausschnitt zu begrenzen (Cropping) und/oder die Ergebnisse zeitlich und räumlich auszudünnen (Subsampling). Im Rahmen des Projektes wurde vom iRMB ein Lösungsansatz verfolgt, der in erster Linie eine gängige Limitierung sowohl frei verfügbarer als auch kommerzieller Visualisierungsumgebungen adressiert. Viele der durchaus populären Post-Processing-Systeme wie beispielsweise ParaView und AVS stoßen bei der Verarbeitung von Massendaten schnell an Ressourcengrenzen, da sie stets und ausschließlich vollständige Datensätze einlesen und im Speicher vorhalten. Im Projektverlauf wurde vom iRMB ein Format für den Datenaustausch zwischen Simulation und Visualisierungsumgebung auf Basis von HDF5 spezifiziert, mit dessen Hilfe eine a priori-Datenreduktion realisiert wurde, die nicht mehr alle Daten zum Beginn der Visualisierung einliest.

2.1.4 AP4: Hardwarenahe Implementierung für Nicht-Standardprozessoren

In diesem Bereich wurden vorrangig GPGPU-Implementierungen entwickelt. Dabei unterschied sich der Fokus der beteiligten Gruppen zum Teil erheblich. Während TU-Do vorrangig für das Softwareprodukt *FEAST* die FE-Kernel auf die GPGPU portiert und neue FEM-LB-GPU-Kernel entwickelte, hat sich das RRZE auf die Entwicklung von hochoptimierten D3Q19-LBM-GPGPU-Kernen konzentriert [10]. Weiterhin wurde der Einfluss des ECC auf Performance und Datenstrukturen untersucht, da vor allem die LBM äußerst sensibel darauf reagierte. Die RRZE Kernel wurden dem LSS zur Verfügung gestellt und in enger Zusammenarbeit in das Softwareprodukt *waLBerla* integriert. Der LSS hat seinerseits diese Kernel für die Entwicklung massiv paralleler LB-Simulationen mit Hilfe von pure-MPI, hybriden- und heterogenen Parallelisierungsansätzen benutzt. Diese wurden im weiteren Verlauf des Projektes für entsprechende Benchmarks und Performance-Modellierungen von Multi-GPGPU Simulationen auf uniformen Gittern verwendet [5]. Der Fokus des iRMB lag weniger auf der reinen Performance-Optimierung, sondern stärker auf der Weiterentwicklung der Methoden. So wurden ergänzend zum D3Q19-Modell, welches von den Partnern favorisiert wurde, auch ein D2Q9-CLB (cascaded lattice Boltzmann) und ein D3Q27-CLB-Modell entwickelt. Diese ermöglichen Simulationen von turbulenten Strömungen ohne explizite Turbulenzmodellierung, bei denen die laminaren LB-Modelle der Partner nicht mehr eingesetzt werden konnten. Abgesehen davon war es zu Beginn des Projektes bei allen Partnern üblich, zwei volle Verteilungssätze auf den GPGPUs zu verwenden, um einen thread-sicheren Programmablauf zu gewährleisten. Das iRMB entwickelte im

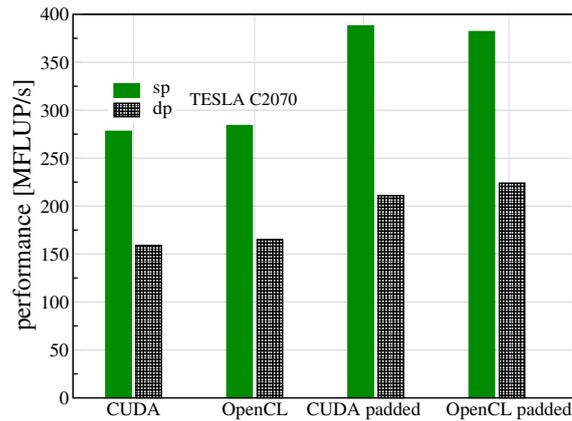


Abbildung 2.9: Performancevergleich zwischen der CUDA- und OpenCL-Implementierung in waLBerla auf einer NVIDIA Tesla C2070 GPU.

Projektverlauf eine neue Datenstruktur bzw. eine neue Form des Speicherzugriffs mit dem Namen „EsoTwist“. Bei vergleichbarer Performance ermöglicht diese eine Halbierung des zu allozierenden Speichers auf der GPGPU, da sie mit einem Satz an Verteilungsfunktionen auskommt. Das RRZE hat diese neue Methode mit einer bereits veröffentlichten Methode von Bailey [BMW⁺09] und einigen anderen teilweise nur für CPUs geeigneten Methoden verglichen [25]. Das iRMB hat für Testfälle mit dünnbestzten Geometriematrizen bzw. mit einem hohen Solid-Anteil eine Implementierung mit indirekter Adressierung auf der GPGPU entwickelt. Diese funktioniert im Vergleich zum Matrix-Code ohne signifikanten Performanceverlust und erzielt (abhängig von der Testfallgeometrie eine erhebliche Ersparnis bzgl. des erforderlichen GPGPU-Speichers. Für realitätsnahe CFD-Testfälle ist Gitterverfeinerung in Ortsbereichen hoher Gradienten notwendig. Während der Projektphase entwickelte das iRMB eine kompakte Interpolationsmethode zweiter Ordnung, die auf GPGPUs mit sehr guter Performance und Ergebnisqualität funktioniert. Diese Eigenschaften waren der Grund, warum diese Methode im weiteren Verlauf sogar für die CPU-Codelinie implementiert wurde. Es entstanden Versionen für das D2Q9-, D3Q19- und das D3Q27-Modell. Selbstverständlich funktioniert die Interpolation auch in Kombination mit dem vorgestellten EsoTwist und der indirekten Adressierung. Analog zu der performanceoptimierten Multi-GPU-Implementierung des LSS wurde auch vom iRMB eine Version mit dem Fokus auf turbulente Probleme implementiert und auf dem während der Projektphase beschafften Cluster „Ludwig“ mit bis zu 96 GPGPUs validiert. Diese Multi-GPGPU-Version wurde erfolgreich für die Simulation des BASF-Benchmark verwendet.

Das RRZE hat OpenCL als alternative Programmierschnittstelle zum CUDA-Framework getestet und konnte damit in waLBerla vergleichbare Performanceergebnisse erzielen (Abb. 2.9).

Implementierungen auf Nicht-Standardprozessoren beschränkten sich im SKALB-Projekt aber nicht auf GPGPU-Entwicklungen. TUDo hat einen FEM-LBM-Code aus *FEAST* sowohl für die ARM-Architektur portiert als auch für den Cell-BE Prozessor implementiert.

2.1.5 AP5: Benchmarking und Showcases

Benchmarks

In enger Zusammenarbeit mit TUDo wurden von IANUS zwei Benchmarks erarbeitet, die gemäß der Vorgaben in AP5 zur Anwendung kommen sollen. Die Anforderungen an die Benchmarks wurden aus zwei unterschiedlichen Sichtweisen heraus definiert. Aus der akademischen Sicht sollten prototypische Benchmarks bzgl. Geometrie und mathematischer Modellierung einfach zu definieren aber komplex in ihrer Ausprägung sein. Aus Sicht der Industrie müssen die aus den Benchmarks erarbeiteten Resultate aber mehr als nur akademische Fragen beantworten und dabei helfen, technische Probleme besser zu verstehen.

Flow around a cylinder in 3D Der *flow around a cylinder* Benchmark stellt ein instationäres Strömungsproblem in einer vergleichsweise einfachen Geometrie dar. Dieser Benchmark beschäftigt sich mit der Umströmung eines zylindrischen Hindernisses in einem Strömungskanal mit rechteckigem Querschnitt. Das fließende Medium ist einphasig und wird durch die inkompressiblen Navier-Stokes-Gleichungen beschrieben. Die Umströmung des Hindernisses führt zu einer instationären Ausprägung der Strömung. Studiert werden kann in diesem Benchmark der Einfluss der Reynoldszahl auf die qualitative und quantitative Ausprägung der Strömung. Als makroskopische Vergleichsgrößen eignen sich z.B. die zeitabhängige Ausprägung von *drag* und *lift* als resultierende Druckkraft auf den Zylinder. Während die Problemstellung für den 2D Fall als gelöst gelten kann, ist die Berechnung des vollen dreidimensionalen Falls immer noch eine Herausforderung. Mit Hilfe von *FEATFLOW* wurden erst kürzlich sehr hochaufgelöste dreidimensionale Berechnungen durchgeführt und gegen Berechnungen mit CFX und OpenFOAM verglichen [BMT11]. Die Ergebnisse des Benchmarks erlauben einen Vergleich der LB-Codes untereinander als auch mit den Ergebnissen kommerzieller Codes.

Die laminare Umströmung eines Zylinders in 3D ist ein einfach zu verstehender Benchmark. Durch Arbeiten, die von TUDo durchgeführt wurden, stehen nun auch Ergebnisse zur Verfügung, die als Referenzlösung angesehen werden können. Diese Berechnungen wurden mit der Q2/P1-Variante von *FEATFLOW* durchgeführt. Berechnet wurden die Beiwerte C_d und C_l (für *drag* und *lift*) während der Umströmung mit variierender Reynoldszahl von 0 bis 100. Die Werte reagieren sehr sensitiv auf z.B. zu geringe Auflösung in Ort und Zeit. Für diesen Benchmark wurden auch Vergleiche mit dem kommerziellen Code CFX und zusätzlich mit OpenFOAM durchgeführt. Dabei

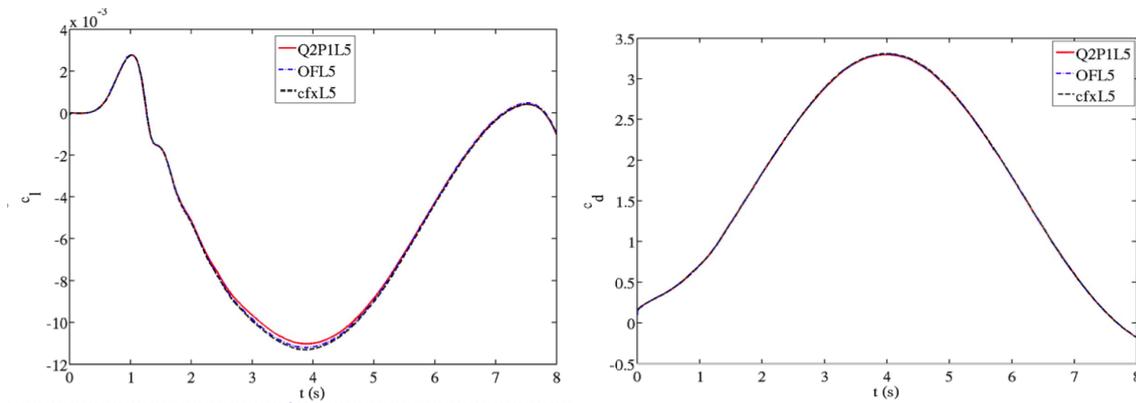


Abbildung 2.10: *Instationäre Entwicklung der Beiwerte. Vergleich verschiedener Codes.*

wurden in allen drei Codes die gleichen Gitter verwendet, die in verschiedenen Feinheiten zur Verfügung standen. Der Benchmark ist in Bezug auf die Geometrie und nötigen physikalischen Parameter exakt definiert. Die nötigen Informationen wurden allen Projektpartnern zur Verfügung gestellt. Die Diagramme in Abbildung 2.10 zeigen die zeitliche Entwicklung der Beiwerte C_d und C_l für die drei Codes.

Monodispers Droplets Der zweite Benchmark ist ebenfalls ein instationäres Problem, welches jedoch zweiphasig ist. Die Problemstellung des Monodispers Droplet Benchmarks betrachtet die Tropfenbildung zweier nicht-mischbarer Flüssigkeiten. Gewählt wurde die Tropfenbildung von Silikon-Öl, welches durch eine Düse in Wasser eingetragen wird. Das Silikon-Öl bildet nach Austritt aus der Düse zunächst einen recht stabilen Pfropfen, der stromabwärts instabil wird, sich einschnürt und letztendlich zu einer Tropfenbildung führt. Die Stoffwerte (Dichte und Viskosität) der beiden Fluide und die Oberflächenspannung sind bekannt. Als gut berechenbare und leicht verständliche makroskopische Vergleichsgrößen eignen sich in diesem Benchmark die Frequenz der Tropfenbildung, die Größe der Tropfen und die Länge des Pfropfens bis zur Einschnürung. Neben einer numerisch berechneten Referenzlösung kann in diesem Fall auch der Vergleich mit experimentellen Werten herangezogen werden, die uns vorliegen.

IANUS hat den Benchmark exakt definiert und den Projektpartnern alle Informationen zur Verfügung gestellt. Abbildung 2.11 zeigt den grafischen Vergleich zwischen Simulation (FEM mit *FEATFLOW*) und Experiment, der eine sehr gute qualitative Übereinstimmung zeigt. Die Abbildungen 2.12 und 2.13 zeigen die Übereinstimmung für verschiedene makroskopische Werte. Berechnet wurden dazu der Volumenanteil der dispersen Phase, die Pfropfenlänge bei der Abschnürung und die Abrissfrequenz für verschiedene Volumenströme.

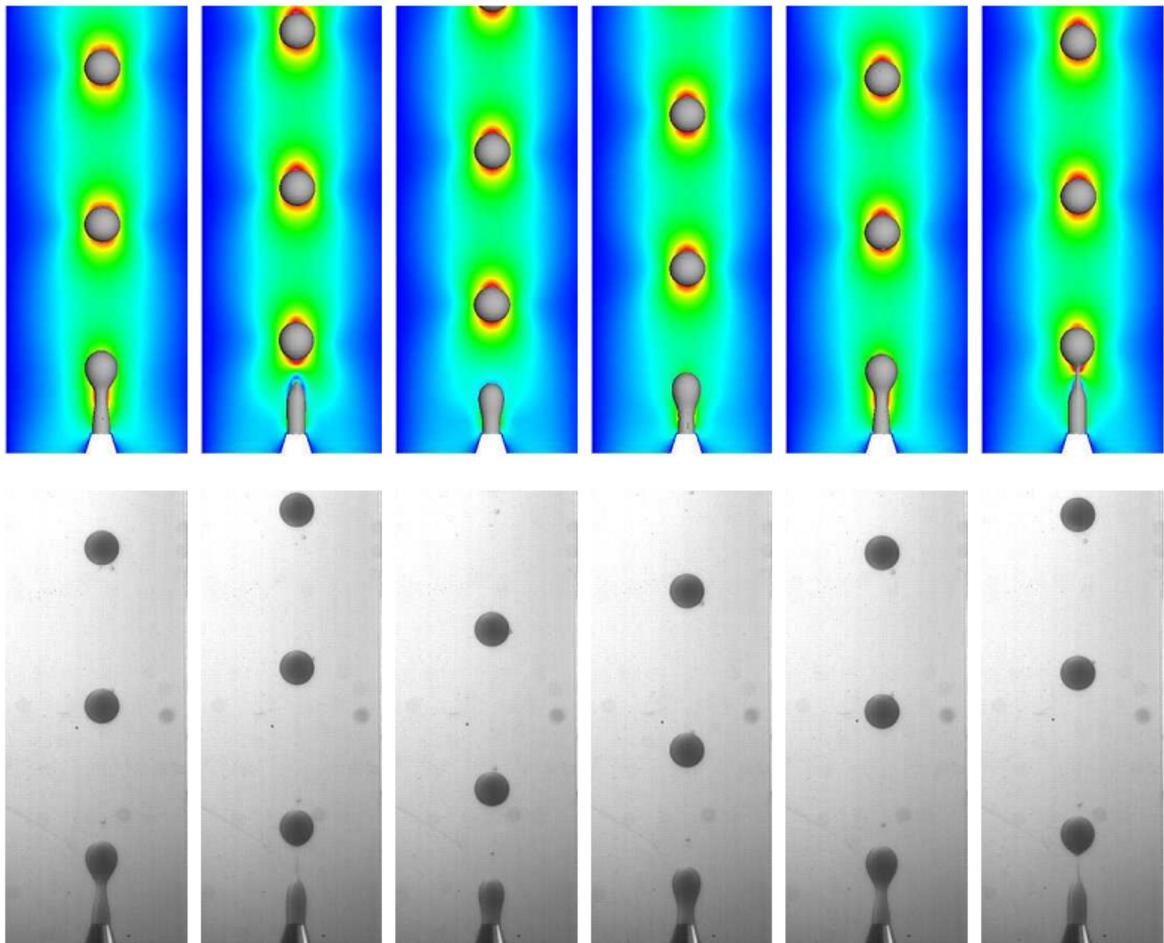


Abbildung 2.11: *Instationäre Tropfenabschnürung. Vergleich zwischen FEM-Simulation (oben) und Experiment (unten) für verschiedene Simulationszeiten.*

Der Benchmark wurde auch mit einem LB-Code am iRMB gerechnet. Abbildung 2.14 zeigt für einen ausgewählten Fall eine qualitativ gute Übereinstimmung mit den FEM-Berechnungen und den experimentellen Ergebnissen. Die Frequenz der Abschnürung und Tropfengröße stimmen sehr gut überein.

Die Ergebnisse zeigen sehr eindrucksvoll, dass LB-Codes sich auch für mehrphasige Problemstellungen eignen. Die Ergebnisse werden wissenschaftlich weiter verfolgt. Die Gruppen in Braunschweig und Dortmund wollen dazu weiterhin eng zusammenarbeiten.

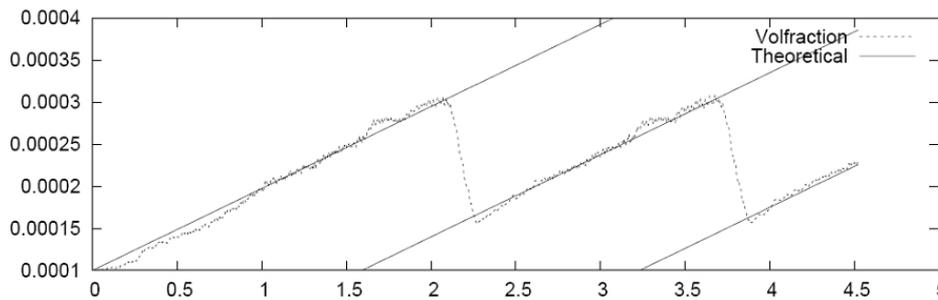


Abbildung 2.12: Berechnete und theoretische Vorhersage des zeitabhängigen Volumenanteils der dispersen Phase.

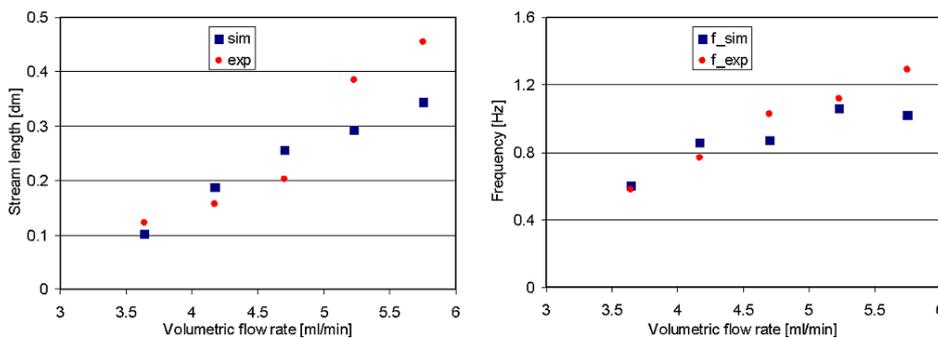


Abbildung 2.13: Gemessene und berechnete Pfropfenlänge (links) und Abrissfrequenz (rechts) für verschiedene Volumenströme.

Showcases

Die verschiedenen Showcases sollten im Gegensatz zu den Benchmarks die Interessen der Industrie besser berücksichtigen. Im nächsten Abschnitt werden die wesentlichen Ergebnisse insbesondere anhand des BASF-Showcases aufgezeigt.

Strömung in einem Reaktionsrohr (BASF) Der Showcase „Reaktionsrohr“ wurde in Zusammenarbeit mit dem assoziierten Partner BASF aufgebaut und definiert. Apparate dieses Typs sind in der chemischen Verfahrenstechnik weit verbreitet. Die zuverlässige Berechnung der Durchströmung insbesondere unter Betrachtung des Wärme- und Stofftransports oder sogar Reaktionen stellt noch immer eine Herausforderung für die Industrie dar. Insbesondere die Vorhersage von Hot-Spots ist dabei ein zentraler Aspekt, da die Temperaturspitzen zu einer Verringerung der Produktqualität führen können. Ziel ist es, durch Kombination von (einer reduzierten Anzahl an) Experimenten mit genaueren Vorhersagen durch Simulationen die Sicherheitsaufschläge in den Prozessen verringern zu können. Damit gelingt eine Erhöhung der chemischen Umsätze und Selektivitäten bei einem geringeren Einsatz an Material und Energie. Die

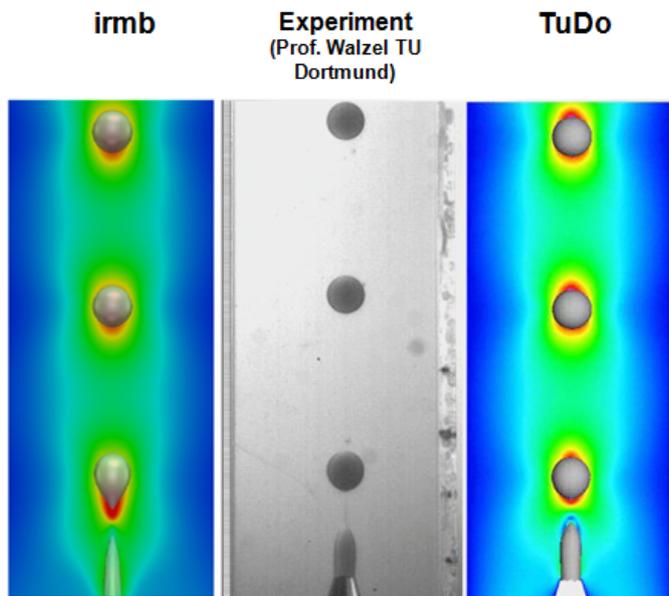


Abbildung 2.14: Vergleich der Ergebnisse des LB-Codes (links) mit einem Experiment (Mitte) und FEM-Berechnung (rechts).

Optimierung der chemischen Prozesse in Richtung höhere Produktqualität bei geringerem Ressourceneinsatz stellt für die deutschen Firmen der chemischen Industrie den Schlüssel im Wettbewerb mit insbesondere asiatischen Mitbewerbern dar.

Herr Dr. Winkler von der BASF hat verschiedene Geometriebeschreibungen von dem Rohr und der darin enthaltenen Schüttung in Form von STL-Dateien bereitgestellt. IANUS hat diese Dateien und alle nötigen Betriebsparameter und Stoffdaten über die Kommunikationsplattform an die einzelnen Partner verteilt. Der gängige Zugang ist, die STL-Daten zu voxeln. Die BASF hat selbst Simulationen mit (teilweise angepassten) kommerziellen Codes durchgeführt und kann die Berechnungen mit den Ergebnissen umfangreicher Versuchsreihen im eigenen Technikum vergleichen. Die BASF besitzt auf Basis dieser Aktivitäten Ergebnisse, die nach ihrer Aussage Referenzcharakter haben. Diese Daten eignen sich damit sehr gut zum Vergleich der Genauigkeit und der Performance der innerhalb von SKALB entwickelten Ansätzen untereinander und mit den BASF-Ergebnissen selbst. Die Daten wurden jedoch bewusst von der BASF bis zum Ende des Projekts zurück gehalten, um einen möglichst unbeeinflussten Einstieg in diesen Showcase zu gewährleisten.

Neben Berechnungen mit zwei Codes des iRBM und Codes von LSS und RRZE liegen zusätzlich auch Ergebnisse auf Basis von *FEATFLOW* vor, die mit einer Q2/P1-Variante erzielt wurden. Die komplexe innere Struktur der Geometrie wurde dabei mit Hilfe der Fictitious Boundary Methode (FBM) abgebildet und gegen die Ergebnisse

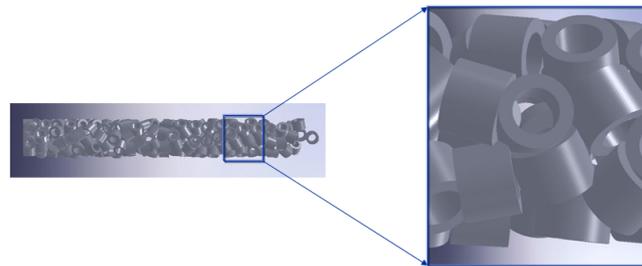


Abbildung 2.15: Das Festbett in dem Reaktionsrohr ist aus zylindrischen Formkörpern aufgebaut.

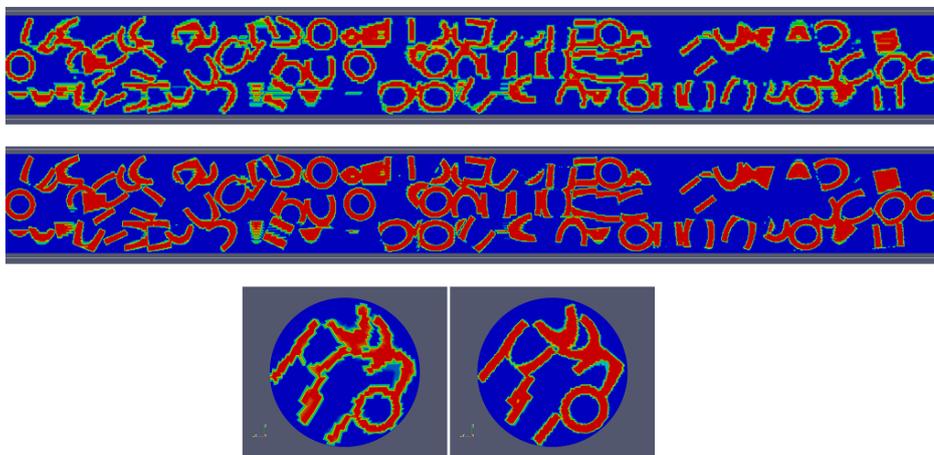


Abbildung 2.16: Örtliche Auflösung durch den Fictitious Boundary Ansatz im Längs- und Querschnitt.

der Voxelisierung getestet. Um die Rechenzeit zu minimieren, wurden die stationären Grenzwerte, die auf einem groben Gitter berechnet werden konnten, auf ein feineres Gitter interpoliert und die Berechnung dort fortgesetzt. Diese Rechnungen wurden auf drei Knoten des LiDo (Linux Cluster der TU-Dortmund) mit insgesamt 48 CPUs gerechnet (Intel Xeon E7340). Die Arbeiten zu dem Benchmark konzentrieren sich zu Anfang auf den Bereich kleinerer Geschwindigkeiten (Leerrohrgeschwindigkeiten $< 1,0 \text{ m/s}$). Die Berechnungen für höhere Leerrohrgeschwindigkeiten bedürfen geeigneter Turbulenzmodelle, die nicht in allen Codes implementiert sind. Abbildung 2.15 zeigt die komplexe innere Struktur des Festbettes. Die Geometrie liegt als STL-Datei vor. Aus der Abbildung 2.16 lässt sich erkennen, wie diese Formkörper für ein gröberes und ein feineres Gitter geometrisch aufgelöst werden. In den Abbildungen 2.17 und 2.18 sind die Geschwindigkeits- und Druckverläufe für eine Leerrohrgeschwindigkeit von $v=0,1 \text{ m/s}$ dargestellt.

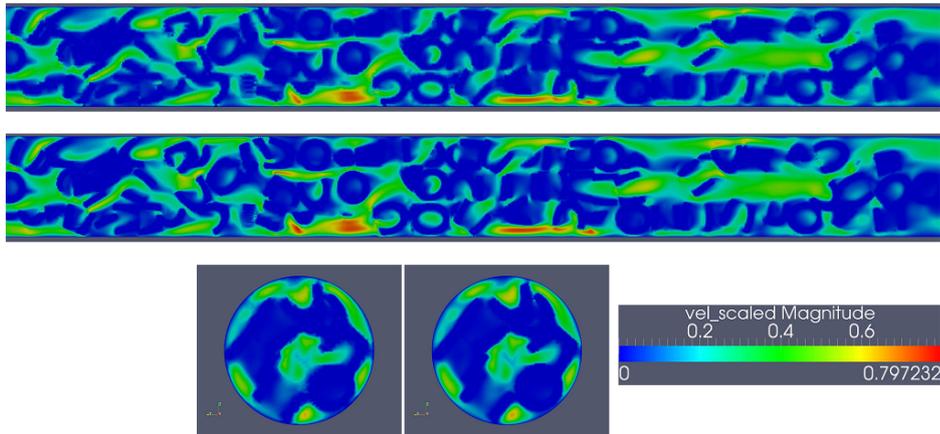


Abbildung 2.17: Geschwindigkeitsverteilung im Festbett für $v=0,1$ m/s.

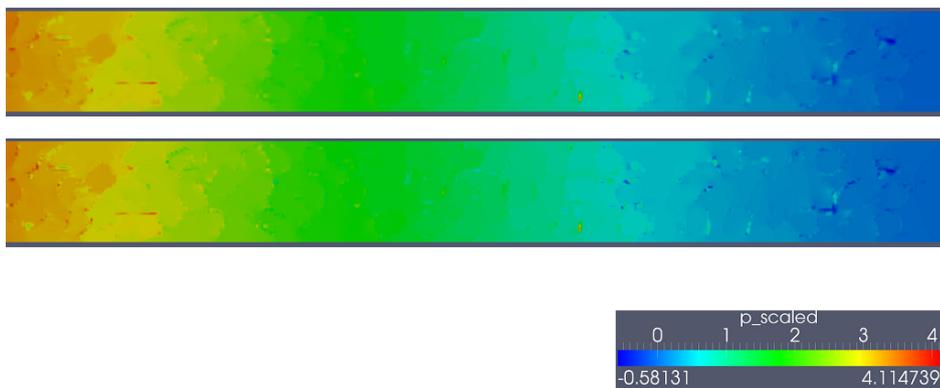


Abbildung 2.18: Druckverteilung im Festbett für $v=0,1$ m/s.

Abbildung 2.19 zeigt das Ergebnis der Voxelisierung durch das iRMB anhand von Schnittdarstellungen. Vergleichbare und auch identische Voxelgitter wurden bei den Berechnungen von LSS und RRZE verwendet.

Die folgende Abbildung 2.20 zeigt, dass vergleichbare Auflösungen bei der FBM und den LB-Ansätzen verwendet wurden. Dargestellt ist eine ausgewählte Schnittebene durch die Schüttung mit klar erkennbaren Strukturen der Füllkörper. Die FBM-Auflösung bildet die geometrischen Details zwar etwas schlechter ab als die verwendete Voxelabbildung, dafür wird bei *FEATFLOW* durch den Q2/P1 Ansatz eine höhere Approximationsgüte erzielt als bei linearen FEM-Ansätzen. Insgesamt sind die Ergebnisse vergleichbar.

In Abbildung 2.21 wird anhand von Schnittdarstellungen der Feldgröße Geschwindigkeit (dargestellt als VectorMagnitude) verdeutlicht, dass die Berechnungen zu vergleichbar guten Ergebnissen führen. Zonen hoher und geringer Strömung werden qualitativ gleich berechnet. Die verschiedenen Größenordnungen der dargestellten Werte-

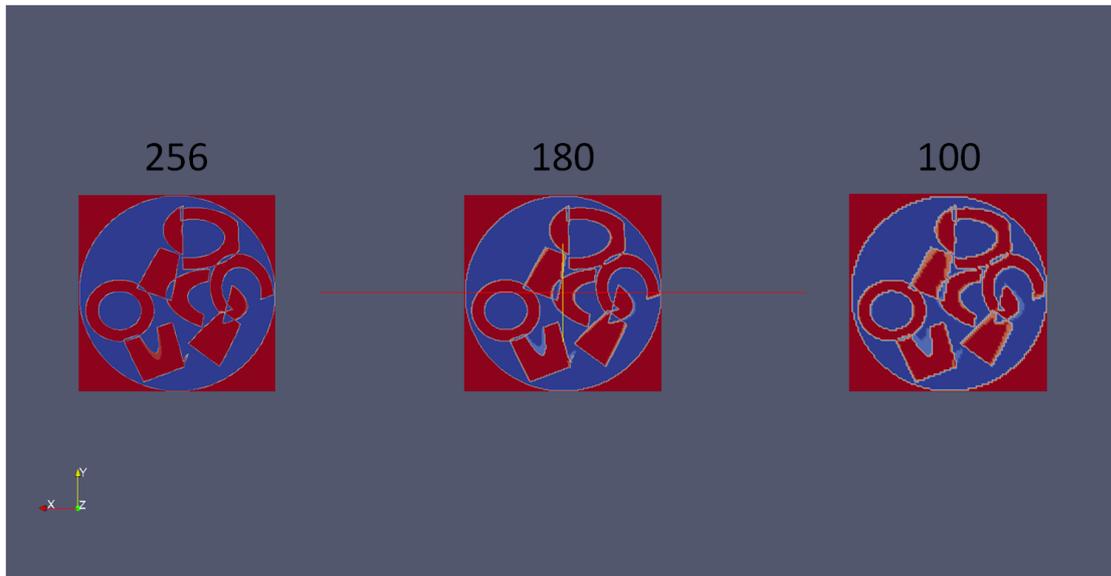


Abbildung 2.19: Unterschiedlich feine Auflösungen beim Voxelisieren.

bereiche stammen aus unterschiedlichen Längeneinheiten (Meter, Dezimeter etc.), die in den drei Codes angesetzt wurden.

Die BASF hat Leerrohrgeschwindigkeiten von mehr als 1 m/s als besonders industrie-relevant ausgewiesen. Bei zunehmenden Leerrohrgeschwindigkeiten wird die Strömung jedoch zunächst instationär und dann turbulent. Durch diesen Umstand bedingt wird der Vergleich der Codes untereinander und mit den Ergebnissen der BASF schwieriger. Dies liegt vor allem daran, dass nicht alle Codes implementierte Turbulenzmodelle aufweisen bzw. unterschiedliche Turbulenzmodelle bieten. Für die industrierelevanten Berechnungen mit Leerrohrgeschwindigkeiten von mehr als 1 m/s wurde ein iRMB-Code verwendet, in dem auch ein Turbulenzmodell (Cascaded LBM, Smagorinski LES)

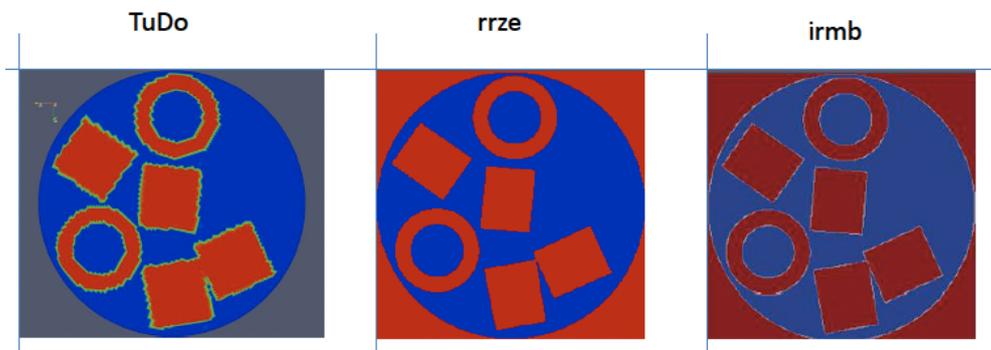


Abbildung 2.20: Vergleich der Auflösung bei der FBM (links) und der Voxelisierung (mitte und rechts).

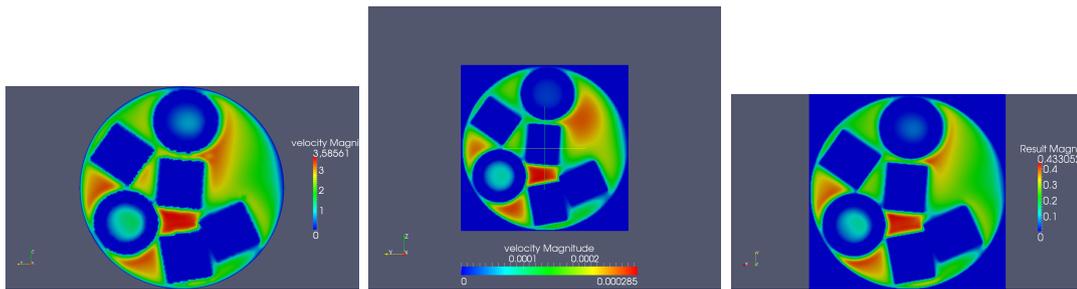


Abbildung 2.21: Vergleich der Berechnungsergebnisse anhand von drei Schnitten. Dargestellt ist die Geschwindigkeit als VectorMagnitude (Reihenfolge wie in Abbildung 2.20).

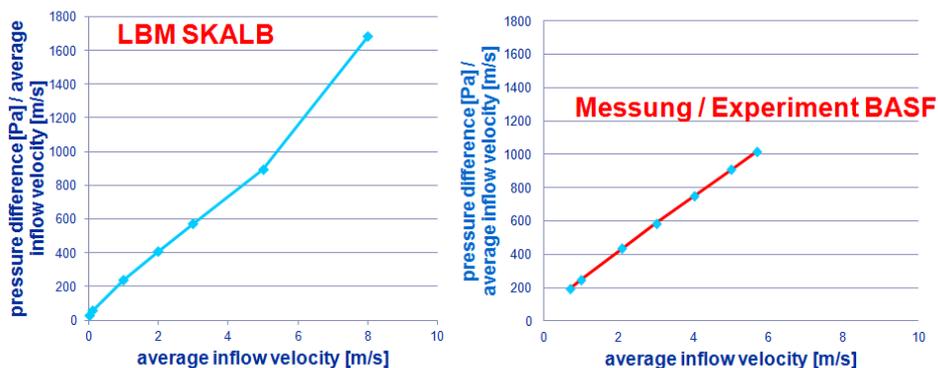


Abbildung 2.22: Vergleich der Ergebnisse für den Druckverlust bei höheren Leerrohrgeschwindigkeiten.

implementiert ist. Auf 12 GPGPUs mit Infiniband-Netzwerk wurden dabei ungefähr 300 MNUPs/GPU erzielt. Die Gitterauflösung lag bei 256x265x3900 Knoten und ist damit sehr fein. Abbildung 2.22 zeigt, dass mit dem Code aus Braunschweig für Leerrohrgeschwindigkeiten bis 5 m/s die Ergebnisse fast exakt abgebildet werden.

Auf Basis der Ergebnisse aus Braunschweig liegen die Kosten für eine Berechnung bei ca. nur 4 bis 5 Euro (Annahme 0,32 Euro pro Knotenstunde und 3 Euro Fixkosten). Als Ergebnis kann festgehalten werden, dass sich LB-Methoden für industrierelevante Fragestellungen eignen. Insgesamt lässt sich feststellen, dass die LB-Codes aus SKALB (VirtualFluids, ILBDC und waLBerla) alle sehr gut skalieren. Anhand des BASF-Showcases konnte nachgewiesen werden, dass die hydrodynamische Berechnung in sehr komplexen 3D-Geometrien schnell und günstig durchgeführt werden kann.

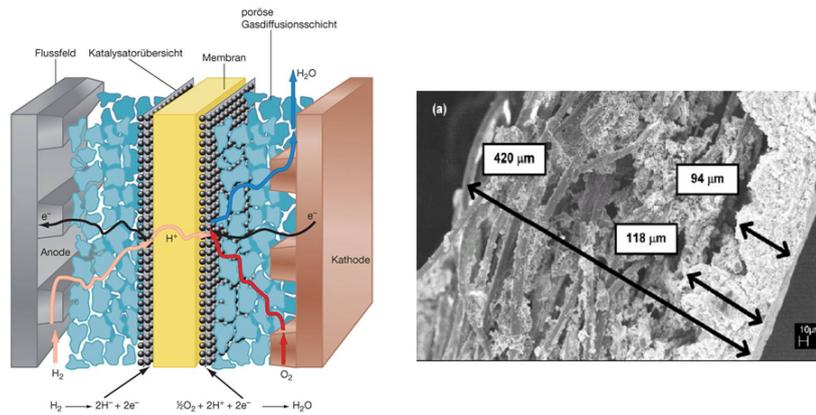


Abbildung 2.23: Aufbau einer Brennstoffzelle mit Gasdiffusionsschicht (GDS).

Simulation des Flüssigwassers im Gasdiffusionsmedium der Brennstoffzelle (DECODE) Der LSS hat Simulationen zum Verhalten des Flüssigwassers im Gasdiffusionsmedium der Brennstoffzelle durchgeführt. Diese Arbeiten wurden innerhalb des EU-Projekts DECODE durchgeführt (www.decode-project.eu). Durch den Verlust von Hydrophobizität in der Diffusions- und Reaktionsschicht sinkt die elektrochemische Leistung der Zelle. Es gibt zahlreiche Anforderungen an derartige Simulationen. Die komplexe Geometrie erfordert eine feine Auflösung in Ort und Zeit. Simulationszeiten in der Größenordnung von 10 Sekunden führen schon zu Simulationslaufzeiten von vielen Tagen. Bisher skalierten die Rechnungen nur bis 64 Prozessen (globale MPI-Kommunikation). Die Parallelisierungsoptimierung in Bezug auf den Kernel und Restart, die in SKALB erarbeitet wurden, bringt einen Performance-Vorteil von 50–80% in Strong-Scaling-Experimenten für die Brennstoffzellegeometrien. Die Wassertransportsimulationen in der Gasdiffusionsschicht (GDS) von Brennstoffzellen wird dadurch möglich. In Abbildung 2.23 ist der beispielhafte schematische Aufbau einer Brennstoffzelle zu erkennen. In Abbildung 2.24 ist die Wasserteilung auf Basis einer Wassertransportsimulation dargestellt.

Strömung in einer Trennkolonnen (SULZER) Dieser Showcase wurde mit Sulzer erarbeitet. In einer Trennkolonnen mit Boden bildet sich eine komplexe mehrphasige Strömung aus. Ziel war die Vorhersage der Grenzen sogenannter Strömungsregimes. Die Kenntnis darüber, wann die Böden z.B. leerlaufen oder die Gasströmung in ungewünschter Weise Flüssigkeit mitreißt, sind wichtige Informationen bei der Auslegung und dem Betrieb dieser Apparate.

Der Showcase der Trennkolonnen ist nach Einschätzung aller Projektpartner zur Zeit nicht befriedigend zu lösen. Dies hängt in erster Linie aber damit zusammen, dass für derart komplexe Strömungsregimes kaum zuverlässige Modelle zur Verfügung stehen. Dieser Showcase wurde daher innerhalb des SKALB-Projekts nicht weiter verfolgt. Sulzer wurde darüber informiert.

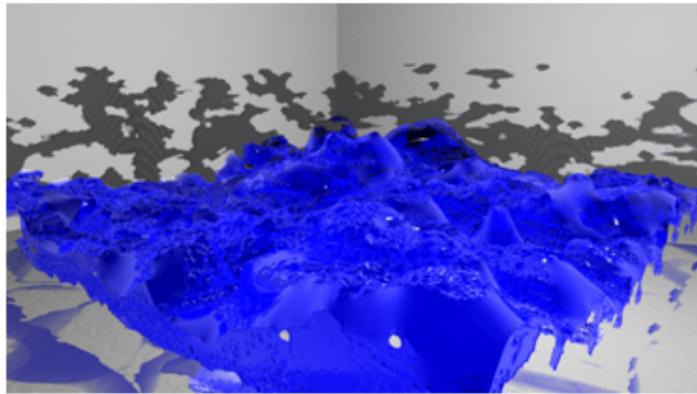


Abbildung 2.24: Verteilung der Flüssigkeit in einer GDS auf Basis einer Wassertransportsimulation.

2.1.6 Erzielte Fortschritte in den weiterentwickelten LB-Codes

Fortschritt VirtualFluids

- Unterstützung von Multi-Core- und Many-Core-(GPGPU)-Hardware
- neue kompakte Interpolationsverfahren zweiter Ordnung für Gitterverfeinerung und Haftrandbedingungen
- adaptive, dynamische, dezentralisierter Lastbalancierung
- Implementierung des D3Q27-Modells und Weiterentwicklung sowie Validierung des Cascaded-Lattice-Boltzmann-Verfahrens für turbulente Strömungen

Fortschritt waLBerla

- hardware-nahe effiziente Kernel
 - effiziente Kernel für LBM und partikuläre Strömungen auf CPUs und GPGPUs
 - CPU-Kernel nutzt SSE und NT-Stores
 - Sowohl CPU als auch GPGPU-LB-Kernel erreichen ungefähr 75% der vorhergesagten Lightspeed-Performance
- Restart
 - Unterstützung von Checkpoint-Restart für Simulationen von freien Oberflächen und partikulären Strömungen
- massiv parallele Simulation

- massiv parallele Simulationen von partikulären Strömungen und freien Oberflächen mit bis zu ca. 300.000 Cores auf Jugene und über 1.000 GPGPUs auf Tsubame 2.0
- Unterstützung von heterogenen CPU-GPGPU-Clustern für LB-Simulation und partikuläre Strömungen
- pure-MPI, hybride, und heterogene Parallelisierungen
- Überlappen von Arbeit und Kommunikation
- verbesserte Skalierbarkeit von waLBerla durch Optimierung des parallelen I/O und weiteren Optimierungen mit Hilfe der Analyse des parallelen Kommunikationsaufwandes
- statisch lastbalancierte heterogene Simulation auf CPU und GPGPU
 - heterogene Simulationen von LB und partikulären Strömungen
 - statische Lastbalancierung durch nicht uniforme Blöcke
 - Kapselung der Heterogenität innerhalb eines MPI-Prozesses und Beschreibung der Hardware pro Rechenknoten
- Metainformationen
 - Beschreibung aller Kernel durch Metainformationen
 - Mit Hilfe der Metainformationen können spezifische Kernel zur Laufzeit ausgewählt werden, um somit unterschiedliche Hardware, physikalische Modelle, und dynamischer Simulationen zu unterstützen
- Software Qualität
 - verbesserte Wartbarkeit und Erweiterbarkeit durch Minimierung der physikalischen Abhängigkeit der einzelnen Komponenten von waLBerla
- erweiterte Simulationsanwendungen und Nutzerschaft
 - Die Erweiterungen an waLBerla ermöglichen eine größere Anzahl an Simulationsanwendungen für einen erweiterten Nutzerkreis.
 - Simulationsanwendungen sind u.a. die Simulation von Schmelzprozessen, Proteinschäumen, Nano-Fluiden, elektro-osmotischen Strömungen, Mikroschwimmer mit Eigenantreib, Strömungen in porösen Medien und Fluidisierungen.

Fortschritt ILBDC

- skalierbarer, lokalitätsoptimierender Präprozessor und Partitionierer
 - Die Zahl der Prozessoren, die für das Präprocessing genutzt werden, ist unabhängig von der Zahl der Prozessoren, die für die anschließende Strömungssimulation genutzt werden.
 - Eine vom Präprozessor aufbereitete Geometriedatei (Partitionierungsdatei) kann anschließend zur Strömungssimulation auf beliebig vielen Prozessoren verwendet werden.
 - Die Knoten des Strömungsgebiets werden so sortiert, dass sie für den Strömungslöser lokalitätsoptimiert vorliegen, d.h. dass im Strömungslöser eine gute Cache-Ausnutzung erfolgt.
 - Zur Aufbereitung und Sortierung der Knoten des Strömungslösers können unterschiedliche raumfüllende Kurven oder Blocking-Ansätze genutzt werden.
- SSE-optimierte Zwei-Gitter-Implementierung des TRT-Kollisionsmodells
 - Nutzung von NT-Stores über algorithmische Änderungen (Pull-Scheme, Split-Loops) und Compiler-Direktiven, so dass auf aktuellen Dual-Socket-Computeknoten mit Intel Westmere-Prozessoren 75% der theoretisch durch das Performancemodell vorhergesagten Spitzenleistung erreicht wird.
- Speicheroptimierte Ein-Gitter-Implementierung des Propagationsschritts auf Basis des AA-Patterns
 - Reduktion des Speicherbedarfs um 40%.
 - Elimination des Write-Allocates (RFO) durch Lesen und Schreiben auf identische Array-Elemente.
 - In Verbindung mit dem SSE-optimierten TRT-Kollisionsmodell werden auf aktuellen Dual-Socket-Computeknoten mit Intel Westmere-Prozessoren 90% der theoretisch durch das Performancemodell vorhergesagten Spitzenleistung erreicht.
- MPI-IO basierter Checkpoint-Restart-Mechanismus
 - Optimale Ausnutzung der Bandbreite von parallelen Dateisystemen.
 - Vermeidung von einzelnen Checkpoint-Dateien pro MPI-Prozess.
 - Restart-Möglichkeit mit einer unterschiedlichen Prozessorzahl.

Fortschritt *FEAST*

- Optimierung der ursprünglichen Implementierung des voll impliziten Ansatzes (*FEASTLBM*)
 - Verbesserung der Stabilität: Obwohl das Finite Differenzen Upwinding von zweiter Ordnung schon für grobe Auflösung des unstrukturierten, lokal verfeinerten Gitters das Strömungsprofil in den meisten Fällen akkurat wieder geben kann, konnte die Stabilität des Verfahrens noch verbessert werden durch den Einsatz des MRT-Modells und des Crank-Nicholson-Verfahrens
 - Die linearen Systeme können mit dem alternativen *generalized equilibrium formulation* Ansatz auch ohne Einsatz von Mehrgitterverfahren gelöst werden.
 - Entwicklung eines speziellen Transport-Vorkonditionierers der in Kombination mit dem BICG-Stab Verfahren bereits hohe Effizienz erreicht.
 - Zeitdiskretisierung hoher Ordnung ist nun möglich
 - Direkt stationäre Lösung ist nun möglich
 - Newton- und Fixpunktiterationen sind nun möglich
 - Die Implementierung des semi-impliziten Ansatzes war erfolgreich
 - Weiterentwicklung mit Finiten Elementen hoher Ordnung und Discontinuous Galerkin
- Implementierung von Finite Elemente Mehrgittermethoden für unstrukturierte Gitter
 - Der FE-gMG wurde an *FEAST* und ScarC erfolgreich angeschlossen
 - Die Datenstrukturen wurden um neue Sparsematrix Formate (CSR, COO, ELLPACK, ELLPACK-R und ELLR-T) für verschiedene Hardwarearchitekturen erweitert
 - Die SparseBandedBLAS von *FEAST* wurde um optimierte Kernel für die SparseMatrix-Vector Multiplikation (SpMV) erweitert: Insbesondere die Vektorisierung (SSE) und Multi-Core bzw. GPGPU-Kernel wurden hier entworfen und optimiert
 - neue Glättungsoperatoren für geometrische Mehrgittermethoden und unstrukturierte Gitter basierend auf Sparse Approximate Inverse Techniken (SPAI, SAINV) wurden hinzugefügt
 - Matrix-basierter Gittertransfer wurde hinzugefügt (Assemblierung)
 - optimierte Assemblierungsroutinen für GPGPUs wurden hinzugefügt
- Allgemeine Verbesserungen des Frameworks:

- Die ursprüngliche Laufzeitumgebung wurde für die Speicherkonsistenzverwaltung bei hybrider Rechnung erweitert
- Die ursprüngliche Laufzeitumgebung wurde für die Multi-GPU-Berechnungen erweitert
- Die Mehrgitterlöser wurden um eine neue funktorbasierte Zyklussteuerung zur Umgehung von Rekursionsoverhead während der Rechnung erweitert
- Die SparseBandedBLAS (SBBLAS) von *FEAST* wurde um ein Frontend mit effizientem Operator- Overloading (basierend auf Hardware-orientierten Expression Templates) erweitert
- Ein 3D Visualisierungstool auf Basis von OpenGL und Qt wurde entwickelt und angebunden
- Neue Datenformate für parallele Gittergeneration / -partitionierung wurden entwickelt und angebunden
- Gitter, Loadbalancing und parallele Löser
 - Die ursprünglichen Prototypen für statisches / dynamisches Loadbalancing wurden verbessert
 - Die numerischen Performancemodelle konnten durch Erkenntnisse im Bereich der starken Glätter verbessert werden, was sich insbesondere auch auf die Entwicklung der Loadbalancer-Logik ausgewirkt hat
 - Der Top-level Löser (ScaRC) und zugehörige Datenstrukturen konnten durch Verwendung neuer Kommunikationsroutinen stark verbessert werden
 - Unstrukturierte (Teil-)Gittern sind verbessert worden und nun auch auf GPGPUs möglich
 - Das Zusammenfassen von Gitterpatches für gemeinsame Matrixassemblierung wurde hinzugefügt
 - Neue Methoden Adaptiver Verfeinerung wurden hinzugefügt
- Standard-LB-Löser
 - Eine Bibliothek mit effizienten 2D LB-Operatoren, die auf jeder zu evaluierenden Hardware läuft ((Multi-)GPU- (CUDA und OpenCL), generische CPU-, SSE-, Multi-Core PThreads-, Cell-Backends und hybride Ausführung) wurde entwickelt und hinzugefügt.
 - Einfache Operatoren wurden um komplexe (Multi-)Physik-Kernel und deren hardwarenahe Optimierungen erweitert: Zentral hierbei sind die folgenden Features: Fluid-Struktur Interaktion, komplexe Kraftterme und zugehörige Datenstrukturen, Kopplung verschiedener Gleichungen (Flachwassergleichungen mit Konvektions- Diffusionsgleichung), Momentum-Exchange, komplexe Gitter, dynamische Gitter.

2.2 wichtigste Positionen des zahlenmäßigen Nachweises

Bei allen am Projekt beteiligten Gruppen sind die Personalkosten für die Projektmitarbeiter der mit Abstand größte Posten. Darüberhinaus standen den einzelnen Gruppen nur noch Reisemittel zur Verfügung. Die Projektbearbeiter und die wichtigsten aus Projektmitteln finanzierten Konferenzteilnahmen sind im Folgenden, nach Gruppen aufgeschlüsselt, gelistet. Eine vollständige Liste der Veröffentlichungen und gehaltenen Vorträge findet sich in Abschnitt 2.6.

2.2.1 FAU / RRZE

Projektbearbeiter (84 FTE)

- J. Habich: 36 FTE (Entwicklung LB-Kernel, GPGPU-Implementierung CUDA)
- M. Wittmann: 30 FTE (skalierbare Partitionierung, ILBDC-Solver)
- J. Treibig: 10 FTE (Performancemodellierung)
- M. Kreutzer: 4 FTE (GPGPU-Implementierung CUDA und OpenCL)
- G. Schubert, F. Shazad, T. Zeiser: 5 FTE (Programmiermodelle, ILBDC-Solver, Checkpoint-Restart)

Verwendung der Reisemittel (größte Posten)

- International Conference on Parallel, Distributed and Grid Computing for Engineering (PARENG); Pecs (Ungarn), 2009: Vortrag Habich [84], [13]
- Parallel Computational Fluid Dynamics (ParCFD); Moffett Field, CA (USA), 2009: Vortrag Wellein [81]
- IEEE International Parallel and Distributed Processing Symposium (IPDPS); Rom (Italien), 2009; Vortrag Zeiser [124], [28, 29]
- International Conference for Mesoscopic Methods in Engineering and Science (ICMMES); Guangzhou (China), 2009: Vortrag Zeiser [82, 116]
- IEEE International Parallel and Distributed Processing Symposium (IPDPS); Atlanta (USA), 2010: Vortrag Wittmann [113], [23, 24]
- ECCOMAS CFD; Lisabon (Portugal), 2010: Vortrag Habich [90]
- SIAM CSE; Reno (USA), 2011: Vorträge Habich und Wellein [76, 104, 106]
- Parallel Computational Fluid Dynamics (ParCFD); Barcelona (Spanien), 2011: Vorträge Habich, Wellein, Wittmann und Zeiser [89, 105, 115], [26]
- International Conference for Mesoscopic Methods in Engineering and Science (ICMMES); Lyon (Frankreich), 2011: Vortrag Zeiser [126], [25]

2.2.2 FAU / LSS

Projektbearbeiter (69 FTE)

- C. Feichtinger: 21 FTE (waLBerla, Datenstrukturen, Parallelisierung, GPGPU, Performance-Modellierung, Partikuläre Strömungen)
- K. Pickl: 18 FTE (waLBerla, Datenstrukturen, Parallele Auswertungen, Partikuläre Strömungen, Datenreduktion)
- D. Bartuschat: 11 FTE (waLBerla, Parallelisierung, Datenreduktion, Partikuläre Strömungen, Checkpoint-Restart)
- F. Schornbaum: 9 FTE (Datenstrukturen, Dynamische Lastbalancierung, Adaptivität, Checkpoint-Restart)
- T. Prelik: 8 FTE (Partikuläre Strömungen, Parallelisierung)
- C. Mihoubi: 1 FTE (Komplexe Geometrien, Visualisierung)

Verwendung der Reisemittel (größte Posten)

- ISC; Hamburg, 2009: Vortrag/Poster Iglberger
- International Conference for Mesoscopic Methods in Engineering and Science (ICMMES); Guangzhou (China), 2009: Vortrag Feichtinger [70]
- Vorträge auf einer Reise nach China/Australien, 2010: Rüde [99, 100]
- SIAM CSE; Reno (USA), 2011: Vortrag Feichtinger [68]
- Parallel Computational Fluid Dynamics (ParCFD); Barcelona (Spanien), 2011: Vortrag Feichtinger [69]
- International Conference for Mesoscopic Methods in Engineering and Science (ICMMES); Lyon (Frankreich), 2011: Vortrag Pickl [95]

2.2.3 iRMB

Projektbearbeiter (72 FTE)

- M. Schönherr: 36 FTE (GPGPU-Implementierung mit CUDA, Gitterverfeinerung und Kernel-Weiterentwicklung)
- K. Kucher: 36 FTE (Weiterentwicklung VirtualFluids, Lastbalancierung, Entwicklung Patchcode)

Verwendung der Reisemittel (größte Posten)

- ECCOMAS CFD; Lissabon (Portugal), 2010: Vortrag Schönherr
- Parallel Computational Fluid Dynamics (ParCFD); Barcelona (Spanien), 2011: Vortrag Schönherr [151]
- International Conference for Mesoscopic Methods in Engineering and Science (ICMMES); Edmonton (Canada), 2010: Vortrag Schönherr [150]
- Akademiekonferenz für junge Wissenschaftler: Facing the Multicore-Challenge; Heidelberg, 2010: Vorträge Kucher und Schönherr

2.2.4 TUDo

Projektbearbeiter (108 FTE)

- M. Geveler: 21 FTE (UcHPC, *FEAST*, HO_rNET, hardwareorientierte LBM)
- T. Hübner: 18 FTE (numerische Alternativen zu LBM, *FEATLBM*, *FEAST*)
- H. Wobker: 18 FTE (*FEAST*)
- S. Buijssen: 16 FTE (*FEASTFLOW*, *FEAST*)
- J. Acker: 15 FTE (*FEAST*)
- V. Mattick: 8,5 FTE (dynamische Lastverteilung)
- M. Köster, R. Mahmood, J.-L. Gonzalez, E. Yaqub: 11,5 FTE (*FEAST*)

Verwendung der Reisemittel (größte Posten)

- Akademiekonferenz für junge Wissenschaftler: Facing the Multicore-Challenge; Heidelberg, 2010: Vortrag Geveler [134], [33]
- International Conference for Computer Science; Amsterdam (Niederlande), 2010: Vortrag Geveler [145], [48]
- SIAM Conference on Mathematical and Computational Issues in the Geosciences; Long Beach, CA (USA), 2011: Vortrag Geveler [130]
- Conference on Parallel, Distributed, Grid and Clouded Computing for Engineering (PARENG); Ajaccio (Frankreich), 2011: Vortrag Geveler [135], [34]
- Parallel Computational Fluid Dynamics (ParCFD); Barcelona (Spanien), 2011: Vortrag Geveler [136], [35, 36]
- International Conference for Mesoscopic Methods in Engineering and Science (ICMMES); Lyon (Frankreich), 2011: Vortrag Hübner

- Mathematical Methods and Models in Biosciences (BIOMATH); Sofia (Bulgarien), 2011: Vortrag Strehl
- AGU Fall Meeting; San Francisco, CA (USA), 2011: Vortrag Götdeke [139]

2.2.5 HLRS

Projektbearbeiter (37 FTE)

- M. Hasert: 10 FTE (Programmiermodelle, CoArray-Fortran, LBM-Kernel)
- D. Jenz, A. Chut, B. Czink: 27 FTE (Erweiterungen Stereo und Covise, Voxelisierer)

Verwendung der Reisemittel (größte Posten) Reisemittel wurden nur für die Projekttreffen in Braunschweig, Dortmund und Erlangen sowie die Teilnahme am Statusworkshop der Gauß-Allianz in Darmstadt verwendet.

2.2.6 IANUS

Projektbearbeiter (geplant 18+18 FTE) Wie in den Zwischenberichten angegeben, wurden innerhalb der Förderphase nicht die vollen Projektmittel abgerufen, da seitens IANUS zu Anfang kein voller Projektmitarbeiter gefunden werden konnte. Die Arbeiten wurden auf verschiedene Mitarbeiter verteilt. Beteiligt wurden die Mitarbeiter Maria Mierkova Szekelyova, Miriam Platte, Konstantinos Nalpantidis und Frank Platte. Die genaue Abrechnung findet sich in den Formblätter.

Projektseitig wurden weniger Showcases angegangen, als ursprünglich geplant war. Die Fragestellungen von Sulzer waren z.B. aufgrund fehlender physikalischer Beschreibungsmodelle mit den verfügbaren Codes nicht berechenbar.

Verwendung der Reisemittel (größte Posten)

- diverse SKALB-interne Projekttreffen in Braunschweig und Erlangen
- Status-Workshops von BMBF und Gauß-Allianz: Vorträge Platte [156, 157]

2.3 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Am **RRZE/FAU** hat ein Doktorand (J. Habich) durchgängig an der Implementierung von hochoptimierten CPU- und GPGPU-Kernen gearbeitet und die Übernahme in das waLBerla-Framework des LSS unterstützt. Der zweite Doktorand (M. Wittmann) hat sich mit der Optimierung des ILBDC-Codes sowie Möglichkeiten der skalierbaren Gebietspartitionierung bei gleichzeitig optimierten Datenzugriff („Cache-Optimierung“) im Rechenanteil des LB-Lösers beschäftigt. Spezifische Aufgaben (z.B. detaillierte Performancemodellierung und Entwicklung eines strukturierten Performance-Engineering-Ansatzes) wurden durch erfahrene Mitarbeiter des RRZE (z.B. J. Treibig, G. Schubert, T. Zeiser) unterstützt. Die Dissemination der Ergebnisse mit über 60 Vorträgen und über 20 Publikationen hat nicht zu unterschätzende Zeit in Anspruch genommen, aber für große Sichtbarkeit in der Scientific Community und Verbreitung der Erkenntnisse gesorgt.

Am **LSS/FAU** hat ein Doktorand (C. Feichtinger) an der Weiterentwicklung der Datenstrukturen und der Parallelisierung von waLBerla gearbeitet, um massiv parallele, hybride, und statisch lastbalancierte heterogene Simulation auf CPU- und GPGPU-Clustern zu unterstützen. Weiterhin wurde die Performance von parallelen Simulation auf GPGPU-Clustern modelliert und vorhergesagt. Ein weiterer Doktorand (F. Schornbaum) hat an der Entwicklung von Datenstrukturen und dynamischen Lastbalancierungsalgorithmen für massive parallele adaptive Simulationen gearbeitet. Weitere Aufgaben (Erweiterung der Datenstrukturen und Parallelisierung für Datenreduktion, Restart, Geometrie, und Show- und Testcases) wurden von weiteren Doktoranden (K. Pickl, D. Bartuschat, T. Preklik, S. Mihoubi) übernommen.

Die Arbeit der **TUDO** war durchweg durch eine enge interdisziplinäre Zusammenarbeit aus den Mathematikern, Informatikern und Ingenieuren des Instituts gekennzeichnet, wobei die Spezialisten in den einzelnen (numerischen, technischen oder hard-/softwaretechnischen) Themengebieten für einzelne Teilprojekte die Führung übernahmen und eventuell dort die Juniorwissenschaftler anleiteten. Details zu den einzelnen Zuständigkeiten der Mitarbeiter können in Kapitel 2.2.4 der Personalaufstellung entnommen werden. Dabei haben die einzelnen Schlüssel die folgende Bedeutung²:

- *FEAST*: Softwareentwicklung /-weiterentwicklung und Implementierung der einzelnen Komponenten und Applikationen, aber auch theoretische Grundlagenarbeit der Basisparadigmen von *FEAST* wie beispielsweise das Löserkonzept (ScaRC)
- UcHPC: Entwicklung von Laufzeitumgebungen und Kernen für unkonventionelle Hardware (GPGPU, Cell, ARM)
- numerische Alternativen zu LBM: Entwicklung des in Abschnitt monolithischen numerischen Ansatzes

²Dieselben Schlüssel wurden auch in den Zwischenberichten verwendet

- *FEASTLBM*: Implementierung der LB-Kernel, Schwerpunkt numerische Evaluation
- *FEASTFLOW*: Implementierung und Weiterentwicklung von Applikationen für *FEAST*, Benchmarking
- HONEI/HORNET: Implementierung und Entwicklung der numerischen Basis-komponenten, insbesondere numerische lineare Algebra (Hardware-Backends und Frontends) (vormals SBBLAS)
- hardwareorientierte LBM: Entwicklung und Implementierung von standard LB-Methoden zur Evaluierung und Grundlagenforschung mit Applikationen für gekoppelte Systeme

Am **iRMB** hat ein Doktorand (K. Kucher) durchgängig an der Weiterentwicklung des VirtualFluids-Frameworks gearbeitet. Dabei wurde alternativ zur vorhandenen Lastbalancierung gemäß Master-Slave-Prinzip eine adaptive, dynamische, dezentralisierte Lastbalancierung implementiert. Zusätzlich wurde gemeinsam mit dem zweiten Doktoranden (M. Schönherr) ein D3Q27-CLB-Modell sowohl für die CPU als auch für die GPGPU implementiert. M. Schönherr hat sich vorrangig mit GPGPU-Implementierungen beschäftigt. Abgesehen vom besonders für turbulente Strömungen geeigneten LB-Modell wurden in der Zeit eine kompakte Gitterverfeinerung zweiter Ordnung und hoch skalierende Parallelisierungen sowohl im CPU- als auch GPGPU-Bereich implementiert. Zusätzlich wurde für das D3Q27-Modell eine Implementierung entwickelt, die auf GPGPUs mit einer indirekten Adressierung funktioniert. Unterstützung speziell im Bereich der Theorie erhielten die Doktoranden von erfahrenen Mitarbeitern (M. Geier, M. Stiebler und M. Krafczyk). Einige der Weiterentwicklungen waren essentiell, um z.B. den BASF-Benchmark erfolgreich simulieren zu können.

Am **HLRS** kam es während der Projektlaufzeit zu massiven Personalfuktuationen und damit verbundenen Zeitverlusten durch die Einarbeitung neuer Mitarbeiter. Herr Hasert ist nach weniger als einem Jahr Projektlaufzeit mit Frau Prof. Roller vom HLRS an die German Research School for Simulation Sciences gGmbH nach Aachen gewechselt. Dort hat er die am HLRS begonnenen Arbeiten als assoziierter Partner ohne förderung aus SKALB weitergeführt und auch an einigen der SKALB-Meetings teilgenommen. Ein geeigneter Nachfolger für Herrn Hasert konnte am HLRS in vertretbarer Zeit nicht gefunden werden, so dass in Abstimmung mit dem Projektträger und dem Projektkoordinator Arbeiten und Mannmonate aus AP1 und AP2 vom HLRS an die FAU verlagert wurden. Die Arbeiten im Pre- und Postprocessing konnten am HLRS erfolgreich durchgeführt werden, wenngleich es auch in diesem Teilbereich zu mehrmaligem Personalwechsel gekommen ist, da Mitarbeiter das HLRS kurzfristig verlassen haben.

Aufgabe der **IANUS GmbH** war die Formulierung geeigneter Showcases und Benchmarks sowie die industriegerechte Aufbereitung und Darstellung der Berechnungsergebnisse. Die industrierelevanten Showcases wurden zusammen mit den assoziierten Industriepartnern BASF und Sulzer definiert und den Konsortialpartner als Blindtest zur Verfügung gestellt. Die Bewertung der erzielten Ergebnisse wurde zusammen mit

BASF vorgenommen. Die zwei Benchmarks der Zylinderumströmung und der Tropfenabschnürung dienten dazu, die Genauigkeit und Performance der LB-Codes gegen bekannte experimentelle Lösungen und/oder Lösungen anderer CFD-Codes zu testen. Die Showcases haben industrierelevante Problemstellungen adressiert. Der Arbeitsaufwand für die beiden Aufgabenbereiche von IANUS war voll gerechtfertigt.

2.4 Voraussichtlicher Nutzen, insbesondere Verwertbarkeit der Ergebnisse im Sinne des fortgeschriebenen Verwertungsplans

Das SKALB-Projekt war durch die Dominanz der universitären Partner sehr wissenschaftlich ausgerichtet. Anschlußprojekte ergeben sich daher v.a. auch wieder im wissenschaftlichen Bereich sowie der Lehre. Für die Firma IANUS GmbH aber auch die universitären Partner bieten die weiterentwickelten LB-Codes und die gewonnenen Erkenntnisse hinsichtlich der Implementierung und Optimierung von hochperformanten wissenschaftlichen Simulationscodes aber auch nicht unerhebliches Potential für Bearbeitung von industriellen Auftragsprojekten.

Anfang 2012 erfolgte durch das iRMB die Gründung eines Technologietransferzentrums „Interaktive Strömungssimulation“ (iSSIM) im Rahmen der iTUBS GmbH. Dasierend auf den in SKALB erzielten Ergebnissen wird der LB-Löser VirtualFluids dort für Industrieprojekte eingesetzt. Zurzeit wird ein größeres Projekt im Bereich CFD-Optimierung eines Kühlcontainers bearbeitet. Ein weiteres Projekt mit der Firma Siemens (Generator Kühlung) ist in Vorbereitung. Es ist zu erwarten, dass mittelfristig eine eigenständige Ausgründung als GmbH erfolgt, wenn sich die Auftragslage mit Industrieprojekten konsolidiert.

Eine Studie der Bayerischen Eliteakademie hinsichtlich einer Kommerzialisierung des waLberla-Frameworks hat das Potential für spezielle Anwendungen im Bereich der medizinischen Physik untersucht. Die Ergebnisse sind grundsätzlich positiv ausgefallen, jedoch fehlt derzeit eine geeignete Gruppe von Unternehmensgründern, welche die Kommerzialisierung umsetzen könnten.

Weiterhin ist basierend auf den Ergebnissen aus SKALB ein weiteres BMBF-Projekt erfolgreich beantragt worden (FeTOL), an dem auch die Projektpartner aus Erlangen und Stuttgart beteiligt sind.

Detailliert wird im Erfolgskontrollbericht der einzelnen geförderten Partner auf den fortgeschriebenen Verwertungsplan eingegangen.

2.5 Während der Durchführung des Vorhabens bekannt gewordene Fortschritte bei anderen Stellen

Alle Projektpartner haben während der Projektlaufzeit thematisch relevante Konferenzen aus den unterschiedlichen für das Projekt relevanten Themenbereiche (Hardware, Computer Science, Engineering) besucht und die Veröffentlichungen in der wissenschaftlichen Literatur intensiv verfolgt. Insbesondere in den USA und China sind unterschiedliche Gruppen auf ähnlichen Gebieten aktiv. Die im Projekt SKALB erzielten Ergebnisse sind aber weltweit wegweisend und durch die Vielzahl an Publikationen und (eingeladenen) Vorträge belegt.

Bedingt durch die Einstellung der Weiterentwicklung der NEC Vektorlinie, des SUN Niagara2-Prozessors sowie der Cell-BE-Architektur wurden diese Teilaspekte des Projektplans während der Projektlaufzeit gekürzt bzw. gestrichen und stattdessen die Aktivitäten im Bereich Multi-Core und GPGPU intensiviert. Diese Verschiebung spiegelt auch die weltweiten Forschungstrends wieder.

Das von P. Bailey et al. [BMW⁺09] an der University of Minnesota entwickelte „AA Pattern“ für das Datenlayout und den LB-Propagationsschritt wurde aufgegriffen und in den ILBDC-Code übernommen. Vom Konzept her ist das „AA Pattern“ dem „Esoteric Twist“ des iRMB sehr ähnlich.

Auf der ParCFD-2010 wurde mit den Entwicklern des OpenLB-Community-Codes intensiver Kontakt aufgenommen und Optimierungs- und Implementierungsstrategien ausgetauscht. Der OpenLB-Code ist hinsichtlich seiner HPC-Leistung aber weiterhin signifikant langsamer als die in SKALB weiterentwickelten Codes.

2.6 Erfolgte oder geplante Veröffentlichungen der Ergebnisse

Die folgenden Abschnitte listen, nach Partnern aufgeschlüsselt, alle im Rahmen des Projekts SKALB entstandenen Veröffentlichungen und gehaltenen Vorträge auf. Für die kommenden Monate ist mit weiteren Veröffentlichungen und Vorträgen zu rechnen, in denen SKALB-Ergebnisse wesentliche Bestandteile sind.

2.6.1 Veröffentlichungen FAU

- [1] BERNSDORF, J., T. ZEISER und H. TAKAHARA: *Application of the lattice-Boltzmann method in chemical engineering and medical physics (auf japanisch)*. Japan Society for Computational Engineering and Science (JSCES), Journal “Computational Engineering”, 14(3):2124–2129, 2009.
- [2] DONATH, S., C. FEICHTINGER, T. POHL, J. GÖTZ und U. RÜDE: *A Parallel Free Surface Lattice Boltzmann Method for Large-Scale Applications*. In: BISWAS, R. (Herausgeber): *Parallel CFD 2009, 21st International Conference on*

Parallel Computational Fluid Dynamics, May 18-22, 2009, Moffett Field, CA, USA, Seiten 198–202, May 2009.

- [3] DONATH, S., C. FEICHTINGER, T. POHL, J. GÖTZ und U. RÜDE: *Localized Parallel Algorithm for Bubble Coalescence in Free Surface Lattice-Boltzmann Method*. In: *Lecture Notes in Computer Science, Euro-Par 2009*, Band 5704, Seiten 735–746, 2009.
- [4] DONATH, S., K. MECKE, S. RABHA, V. BUWA und U. RÜDE: *Verification of surface tension in the parallel free surface lattice Boltzmann method in waLBerla*. *Computers & Fluids*, 45(1):177–186, 2011.
- [5] FEICHTINGER, C.: *Entwurf und Performance-Evaluierung eines Software-Frameworks für Multi-Physik-Simulationen auf heterogenen Supercomputern*. Doktorarbeit, Technische Fakultät, Universität Erlangen-Nürnberg, 2012.
- [6] FEICHTINGER, C., S. DONATH, H. KÖSTLER, J. GÖTZ und U. RÜDE: *WaLBerla: HPC software design for computational engineering simulations*. *Journal of Computational Science*, 1(2):105–112, 2011.
- [7] FEICHTINGER, C., J.HABICH, T. ZEISER, G. HAGER und G. WELLEIN: *A Flexible Patch-Based Lattice Boltzmann Parallelization Approach for Heterogeneous GPU–CPU Clusters*. *Parallel Computing*, 37(9):536–549, 2011.
- [8] GÖTZ, J., S. DONATH, C. FEICHTINGER und U. RÜDE: *Single Node Performance and Weak Scaling of waLBerla on Current SuperComputing Platforms*. Technischer Bericht 09–16, University of Erlangen-Nuremberg, Computer Science 10 – Systemsimulation, 2009.
- [9] GÖTZ, J., K. IGLBERGER M. STÜRMER und U. RÜDE: *Direct Numerical Simulation of Particulate Flows on 294912 Processor Cores*. In: *IEEE computer society: 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (Supercomputing 2010, New Orleans)*, 2010.
- [10] HABICH, J.: *Achieving maximum efficiency on contemporary Multi- and Many-Core Cluster with the lattice Boltzmann method through guided performance engineering*. Doktorarbeit, Technische Fakultät, Universität Erlangen-Nürnberg, Einrichtung 2012 geplant.
- [11] HABICH, J., C. FEICHTINGER, H. KÖSTLER, G. HAGER und G. WELLEIN: *Performance engineering for the lattice Boltzmann method on GPGPUs: Architectural requirements and performance results*. *Computers & Fluids*, in press, 2012. DOI:10.1016/j.compfluid.2012.02.013.
- [12] HABICH, J., T. ZEISER, G. HAGER und G. WELLEIN: *Enabling temporal blocking for a lattice Boltzmann flow solver through multicore-aware wavefront*

- parallelization*. In: BISWAS, R. (Herausgeber): *Parallel CFD 2009, 21st International Conference on Parallel Computational Fluid Dynamics, May 18-22, 2009, Moffett Field, CA, USA*, Seiten 178–182, May 2009.
- [13] HABICH, J., T. ZEISER, G. HAGER und G. WELLEIN: *Speeding up a lattice Boltzmann kernel on nVIDIA GPUs*. In: TOPPING, B.H.V. und P. IVÁNYI (Herausgeber): *Proceedings of the First International Conference on Parallel, Distributed and Grid Computing for Engineering*, Kippen, Stirlingshire, United Kingdom, 2009. Civil-Comp Press. DOI:10.4203/ccp.90.17.
- [14] HABICH, J., T. ZEISER, G. HAGER und G. WELLEIN: *Performance analysis and optimization strategies for a D3Q19 lattice Boltzmann kernel on nVIDIA GPUs using CUDA*. *Advances in Engineering Software*, In Press, Corrected Proof, 2011. DOI:10.1016/j.advengsoft.2010.10.007.
- [15] HAGER, G., T. ZEISER und G. WELLEIN: *Data access characteristics and optimizations for Sun UltraSPARC T2 and T2+ systems*. *Parallel Processing Letters*, 18(4):471–490, 2008.
- [16] HALWAI, J.: *Performance Analysis of Hybrid Parallelization Techniques for Large Scale Lattice Boltzmann Simulations on Multicore Platforms*. Master's thesis, University of Erlangen-Nuremberg, Computer Science 10 – Systemsimulation, 2009.
- [17] MASILAMANI, K., S. GANGULY, C. FEICHTINGER und U. RÜDE: *Hybrid lattice-Boltzmann and finite-difference simulation of electroosmotic flow in microchannel*. *Fluid Dynamics Research*, 43(2):1–25, 2011.
- [18] PICKL, K., J. GÖTZ, K. IGLBERGER, J. PANDE, K. MECKE, A.-S. SMITH und U. RÜDE: *All Good Things Come in Threes – Three Beads Learn to Swim With Lattice Boltzmann and a Rigid Body Solver*. *Journal of Computational Science*, in press, 2012. DOI:10.1016/j.jocs.2012.04.009.
- [19] SEMBRITZKI, K.: *Evaluation of the Coarray Fortran Programming Model on the Example of a Lattice Boltzmann Code*. Diplomarbeit, Regionales Rechenzentrum Erlangen, Universität Erlangen-Nürnberg, 2012.
- [20] STENGEL, H.: *Parallel Programming on Hybrid Hardware: Models and Applications*. Masterarbeit, Georg-Simon-Ohm-Hochschule Nürnberg, 2010.
- [21] WELLEIN, G., G. HAGER, T. ZEISER, M. WITTMANN und H. FEHSKE: *Efficient temporal blocking for stencil computations by multicore-aware wavefront parallelization*. In: *33rd IEEE International Computer Software and Applications Conference (COMPSAC 2009)*, Seattle, USA, Juli 2009. DOI:10.1109/COMPSAC.2009.82.
- [22] WITTMANN, M.: *Optimierungen für hoch-skalierende Lattice-Boltzmann Löser*. Doktorarbeit, Technische Fakultät, Universität Erlangen-Nürnberg, laufende Arbeit.

- [23] WITTMANN, M., G. HAGER, J. TREIBIG und G. WELLEIN: *Leveraging shared caches for parallel temporal blocking of stencil codes on multicore processors and clusters*. *Parallel Processing Letters*, 20(4):359–376, 2010.
- [24] WITTMANN, M., G. HAGER und G. WELLEIN: *Multicore-Aware Parallel Temporal Blocking of Stencil Codes for Shared and Distributed Memory*. In: *Proceedings of the Workshop on Large-Scale Parallel Processing (LSPP) 2010, Atlanta, USA*, 2010.
- [25] WITTMANN, M., T. ZEISER, G. HAGER und G. WELLEIN: *Comparison of different propagation steps for lattice Boltzmann methods*. *Computers and Mathematics with Applications*, in press, 2012. DOI:10.1016/j.camwa.2012.05.002.
- [26] WITTMANN, M., T. ZEISER, G. HAGER und G. WELLEIN: *Domain decomposition and locality optimization for large-scale lattice Boltzmann simulations*. *Computers & Fluids*, in press, 2012. DOI:10.1016/j.compfluid.2012.02.007.
- [27] ZEISER, T., G. HAGER und G. WELLEIN: *Vector computers in a world of commodity clusters, massively parallel systems and many-core many-threaded CPUs: recent experience based on advanced lattice Boltzmann flow solvers*. In: NAGEL, W. E., D. B. KRÖNER und M. RESCH (Herausgeber): *High Performance Computing in Science and Engineering '08, Transactions of the High Performance Computing Center, Stuttgart (HLRS) 2008*, Seiten 333–347. Springer, 2008.
- [28] ZEISER, T., G. HAGER und G. WELLEIN: *Benchmark analysis and application results for lattice Boltzmann simulations on NEC SX vector and Intel Nehalem systems*. *Parallel Processing Letters*, 19(4):491–511, 2009.
- [29] ZEISER, T., G. HAGER und G. WELLEIN: *The world's fastest CPU and SMP node: Some performance results from the NEC SX-9*. In: *Proceedings of the 23rd IEEE International Parallel and Distributed Processing Symposium (IPDPS), Roma, 25-29 Mai*, Seiten 1–8, 2009.

2.6.2 Veröffentlichungen TUDo

- [30] BUIJSSEN, SVEN H.M.: *Efficient Multilevel Solvers and High Performance Computing Techniques for the Finite Element Simulation of the Transient, Incompressible Navier–Stokes Equations*. Doktorarbeit, TU Dortmund, Fakultät für Mathematik. in Vorbereitung.
- [31] GEVELER, MARKUS: *Echtzeitfähige Interaktion von Festkörpern mit 2D Lattice–Boltzmann Flachwasserströmungen in 3D Virtual–Reality Anwendungen*. Diplomarbeit, Technische Universität Dortmund, August 2009.
- [32] GEVELER, MARKUS: *Multi-level FEM Simulators for Continuum Mechanics on Future HPC Architectures*. Doktorarbeit, Technische Universität Dortmund, Fakultät für Mathematik, Juni 2013. Provisorischer Titel, in Vorbereitung.

- [33] GEVELER, MARKUS, DIRK RIBBROCK, DOMINIK GÖDDEKE und STEFAN TUREK: *Lattice-Boltzmann Simulation of the Shallow-Water Equations with Fluid-Structure Interaction on Multi- and Manycore Processors*. In: KELLER, RAINER, DAVID KRAMER und JAN-PHILIPP WEISS (Herausgeber): *Facing the Multicore Challenge*, Band 6310 der Reihe *Lecture Notes in Computer Science*, Seiten 92–104. Springer, September 2010.
- [34] GEVELER, MARKUS, DIRK RIBBROCK, DOMINIK GÖDDEKE, PETER ZAJAC und STEFAN TUREK: *Efficient Finite Element Geometric Multigrid Solvers for Unstructured Grids on GPUs*. In: IVÁNYI, PETER und BARRY H.V. TOPPING (Herausgeber): *Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*, Seite 22, April 2011. Young Researcher Best Paper Award.
- [35] GEVELER, MARKUS, DIRK RIBBROCK, DOMINIK GÖDDEKE, PETER ZAJAC und STEFAN TUREK: *Towards a complete FEM-based simulation toolkit on GPUs: Geometric multigrid solvers*. In: *23rd International Conference on Parallel Computational Fluid Dynamics (ParCFD'11)*, Mai 2011.
- [36] GEVELER, MARKUS, DIRK RIBBROCK, DOMINIK GÖDDEKE, PETER ZAJAC und STEFAN TUREK: *Towards a complete FEM-based simulation toolkit on GPUs: Unstructured Grid Finite Element Geometric Multigrid solvers with strong smoothers based on Sparse Approximate Inverses*. *Computers & Fluids*, Januar 2012.
- [37] GEVELER, MARKUS, DIRK RIBBROCK, SVEN MALLACH, DOMINIK GÖDDEKE und STEFAN TUREK: *A Simulation Suite for Lattice-Boltzmann based Real-Time CFD Applications Exploiting Multi-Level Parallelism on modern Multi- and Many-Core Architectures*. *Journal of Computational Science*, 2:113–123, Januar 2011.
- [38] GÖDDEKE, D.: *Fast and Accurate Finite-Element Multigrid Solvers for PDE Simulations on GPU Clusters*. Doktorarbeit, Technische Universität Dortmund, Fakultät für Mathematik, Mai 2010. <http://hdl.handle.net/2003/27243> (<http://www.logos-verlag.de/cgi-bin/buch?isbn=2768>).
- [39] GÖDDEKE, DOMINIK und ROBERT STRZODKA: *Mixed Precision GPU-Multigrid Solvers with Strong Smoothers*. In: KURZAK, JAKUB, DAVID A. BADER und JACK J. DONGARRA (Herausgeber): *Scientific Computing with Multicore and Accelerators*, Kapitel 7, Seiten 131–147. CRC Press, Dezember 2010.
- [40] GÖDDEKE, DOMINIK und ROBERT STRZODKA: *Cyclic Reduction Tridiagonal Solvers on GPUs Applied to Mixed Precision Multigrid*. *IEEE Transactions on Parallel and Distributed Systems*, 22(1):22–32, Januar 2011.
- [41] HÜBNER, T., R. MAHMOOD und S. TUREK: *Fully implicit nonstationary flow simulations with a monolithic off-lattice Boltzmann approach*. Technischer Be-

- richt, Fakultät für Mathematik, TU Dortmund, April 2012. Ergebnisberichte des Instituts für Angewandte Mathematik, Nummer 451.
- [42] HÜBNER, THOMAS: *A Monolithic, Off-Lattice Approach to the Discrete Boltzmann Equation with Fast and Accurate Numerical Methods*. Doktorarbeit, Technische Universität Dortmund, Fakultät für Mathematik, März 2011.
- [43] HÜBNER, THOMAS und STEFAN TUREK: *Efficient monolithic simulation techniques for the stationary Lattice Boltzmann equation on general meshes*. Computing and Visualization in Science 13, 2010.
- [44] KOMATITSCH, DIMITRI, GORDON ERLEBACHER, DOMINIK GÖDDEKE und DAVID MICHÉA: *High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster*. Journal of Computational Physics, 229:7692–7714, Oktober 2010.
- [45] KÜHBACHER, CH.: *Hochauflösende Finite Elemente Methoden mit Anwendung auf die Flachwassergleichungen*. Diplomarbeit, Technische Universität Dortmund, März 2009.
- [46] RIBBROCK, DIRK.: *Entwurf einer Softwarebibliothek zur Entwicklung portabler, hardwareorientierter HPC Anwendungen am Beispiel von Strömungssimulationen mit der Lattice Boltzmann Methode*. Diplomarbeit, Technische Universität Dortmund, August 2009.
- [47] RIBBROCK, DIRK: *Numerical and algorithmic Concepts for FEM Software on Manycore Architectures*. Doktorarbeit, Technische Universität Dortmund, Fakultät für Mathematik, Juni 2013. Provisorischer Titel, in Vorbereitung.
- [48] RIBBROCK, DIRK, MARKUS GEVELER, DOMINIK GÖDDEKE und STEFAN TUREK: *Performance and Accuracy of Lattice-Boltzmann Kernels on Multi- and Manycore Architectures*. In: SLOOT, PETER M.A., G. DICK VAN ALBADA und JACK J. DONGARRA (Herausgeber): *International Conference on Computational Science (ICCS'10)*, Band 1 der Reihe *Procedia Computer Science*, Seiten 239–247, Mai/Juni 2010.
- [49] TUREK, STEFAN, DOMINIK GÖDDEKE, CHRISTIAN BECKER, SVEN H.M. BUIJSSEN und HILMAR WOBKER: *FEAST – Realisation of hardware-oriented Numerics for HPC simulations with Finite Elements*. Concurrency and Computation: Practice and Experience, 22(6):2247–2265, November 2010.
- [50] TUREK, STEFAN, DOMINIK GÖDDEKE, SVEN H.M. BUIJSSEN und HILMAR WOBKER: *Hardware-Oriented Multigrid Finite Element Solvers on GPU-Accelerated Clusters*. In: KURZAK, JAKUB, DAVID A. BADER und JACK J. DONGARRA (Herausgeber): *Scientific Computing with Multicore and Accelerators*, Kapitel 6, Seiten 113–130. CRC Press, Dezember 2010.

- [51] WOBKER, HILMAR: *Efficient Multilevel Solvers and High Performance Computing Techniques for the Finite Element Simulation of Large-Scale Elasticity Problems*. Doktorarbeit, TU Dortmund, Fakultät für Mathematik, März 2010. <http://hdl.handle.net/2003/26998>.

2.6.3 Veröffentlichungen iRMB

- [52] GELLER, S., S. KOLLMANNSSBERGER, M. EL BETTAH, M. KRAFCZYK und D. SCHOLZ: *An Explicit Model for Three-Dimensional Fluid-Structure Interaction using LBM and p-FEM*. In: AL., H.-J. BUNGARTZ (ED.) ET (Herausgeber): *Fluid-structure interaction II*, Seiten 285–325. Springer, 2010.
- [53] JANSSEN, C. und M. KRAFCZYK: *A lattice Boltzmann approach for free-surface-flow simulations on non-uniform block-structured grids*. *Computers & Mathematics with Applications*, 59:341–350, 2010.
- [54] JANSSEN, C. und M. KRAFCZYK: *Free Surface flow simulations on GPGPUs using the LBM*. *Computers and Mathematics with Applications*, 61(12):3549–3563, 2011.
- [55] KRAFCZYK, M., J. TÖLKE, B. AHRENHOLZ, S. BINDICK, S. FREUDIGER, S. GELLER, C. JANSSEN und B. NACHTWEY: *Kinetic Modeling and Simulation of Environmental and Civil Engineering Flow Problems*. In: HIRSCHL, E. und E. KRAUSE (Herausgeber): *100 Volumes of 'Notes on Numerical Fluid Mechanics'*, Seiten 341–350, Kippen, Stirlingshire, United Kingdom, 2009. Springer.
- [56] KRAUSE, T.: *Entwicklung von Algorithmen zur automatisierten Leveldefinition speziell definierter Gitter*. Studienarbeit, iRMB, TU-Braunschweig, 2010.
- [57] LINXWEILER, J., M. KRAFCZYK und J. TÖLKE: *Highly interactive computational steering for coupled 3D flow problems utilizing multiple GPUs*. *Computing and Visualization in Science*, 2011.
- [58] LUO, L.S., M. KRAFCZYK und W. SHYY: *Lattice Boltzmann Method for Computational Fluid Dynamics*. *Encyclopedia of Aerospace Engineering*, Chapter 56:651–660, 2010.
- [59] SCHÖNHERR, M., K. KUCHER, M. GEIER, M. STIEBLER, S. FREUDIGER und M. KRAFCZYK: *Multi-thread implementations of the lattice Boltzmann method on non-uniform grids for CPUs and GPUs*. *Computers and Mathematics with Applications*, 61(12):3730–3743, 2011.
- [60] STIEBLER, M., S. FREUDIGER, M. KRAFCZYK und M. GEIER: *Lattice-Boltzmann large eddy simulation of subcritical flows around a sphere on non-uniform grids*. *Computers and Mathematics with Applications*, 61(12):3475–3484, 2011.

- [61] STIEBLER, M., S. FREUDIGER, M. KRAFCZYK und M. GEIER: *Parallel Lattice-Boltzmann Simulation of Transitional Flow on Non-uniform Grids*. In: KRAUSE, E., Y. SHOKIN, M. RESCH, D. KRÖNER und N. SHOKINA (Herausgeber): *Computational Science and High Performance Computing IV*, Band 115, Seiten 283–295. Springer, 2011.

2.6.4 Veröffentlichungen HLRS

- [62] BERNREUTHER, MARTIN und DOMENIC JENZ: *Steering and coupling parallel numerical codes for nanoflow simulations*. First International Workshop on Computational Engineering - Special Topic Fluid-Structure Interaction, Haus der Bayerischen Landwirtschaft, Herrsching am Ammersee, Oktober 2009.
- [63] HASERT, MANUEL, HARALD KLIMACH und SABINE ROLLER: *CAF versus MPI - applicability of coarray fortran to a flow solver*. In: *Proceedings of the 18th European MPI Users' Group conference on Recent advances in the message passing interface*, EuroMPI'11, Seiten 228–236, Berlin, Heidelberg, 2011. Springer-Verlag.

2.6.5 Vorträge FAU

- [64] DONATH, S.: *A Contact Angle Model for the Parallel Free Surface Lattice Boltzmann Method in waLBerla*. International Conference for Mesoscopic Methods in Engineering and Science 2010, Edmonton, Canada, July 2010.
- [65] DONATH, S.: *Parallel Free/Surface Extension of the Lattice-Boltzmann Method*. Kurzlehrgang NUMET, Numerische Methoden zur Berechnung von Strömungs- und Wärmeübertragungsproblemen, Lehrstuhl für Strömungsmechanik, Erlangen, Germany, Februar 2010.
- [66] DONATH, S., V. BUWA und U. RÜDE: *Experimental Verification of the Parallel Free Surface Lattice Boltzmann Method in waLBerla*. ParCFD 2010, Kaohsiung, Taiwan, Mai 2010.
- [67] DONATH, S. und U. RÜDE: *Capillary Flows with Parallel Free-Surface Lattice Boltzmann Method*. SIAM CSE 2011, Reno, Nevada, USA, March 2011.
- [68] FEICHTINGER, C.: *WaLBerla: Heterogeneous Simulation of Particulate Flows on GPU Clusters*. SIAM CSE 2011, Reno, Nevada, USA, March 2011.
- [69] FEICHTINGER, C., J. HABICH, H. KÖSTLER, U. RÜDE und G. WELLEIN: *WaLBerla: Heterogeneous Simulation of Particulate Flows on GPU Clusters*. ParCDF 2011, special session on LBM, Barcelona, Spain, May 2011.

- [70] FEICHTINGER, C., J.GÖTZ, K. IGLBERGER, T. PRECLI, P. NEUMANN, F. ARISTIZABAL und U. RÜDE: *waLBerla: Massively Parallel Lattice Boltzmann Simulations*. 6th International Conference for Mesoscopic Methods in Engineering and Science, Guangzhou City, Guangdong (Canton) Province, China, Juli 2009.
- [71] FEICHTINGER, C., J.GÖTZ, K. IGLBERGER und U. RÜDE: *waLBerla: A Framework for Simulating Complex Flows on 10,000 Cores and Beyond*. PARNUM, Smolenice, Oktober 2009.
- [72] GÖTZ, J., K. IGLBERGER, S. DONATH, C. FEICHTINGER und U. RÜDE: *WaL-Berla: A software framework for CFD applications on 300.000 Compute Cores*. Workshop "Multiscale fluid dynamics with the Lattice Boltzmann Method", Leiden, The Netherlands, Februar 2011.
- [73] GÖTZ, J., K. IGLBERGER und U. RÜDE: *Direct Numerical Simulation of Particulate Flows on 294912 Processor Cores*. SIAM CSE 2011, Reno, Nevada, USA, March 2011.
- [74] HABICH, J.: *Towards multicore-aware wavefront parallelization of a lattice Boltzmann flow solver*. 5th Erlangen International High-End Computing Symposium, Regionales Rechenzentrum, Universität Erlangen-Nürnberg, Juni 2009.
- [75] HABICH, J.: *Optimizations on Highly Parallel On-Chip Architectures: GPUs vs. Multi-Core CPUs (for stencil codes)*. Eingeladener Seminar-Vortrag, iRMB, TU-Braunschweig, July 2010.
- [76] HABICH, J. und C. FEICHTINGER: *Performance Optimizations for Heterogeneous and Hybrid 3D Lattice Boltzmann Simulations on Highly Parallel On-Chip Architectures*. SIAM CSE 2011, Reno, Nevada, USA, March 2011.
- [77] HABICH, J., C. FEICHTINGER, G. HAGER und G. WELLEIN: *Poster: Parallelizing Lattice Boltzmann Simulations on Heterogeneous GPU and CPU Clusters*. 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, New Orleans, Louisiana, USA, November 2010.
- [78] HABICH, J., C. FEICHTINGER und G. WELLEIN: *GPU Optimizations at RRZE*. ZISC GPU Workshop, FAU, Erlangen, April 2011.
- [79] HABICH, J., C. FEICHTINGER und G. WELLEIN: *Hochskalierbarer Lattice Boltzmann Löser für GPGPU Cluster*. High Performance Computing Workshop , Leogang, Austria, Februar 2012.
- [80] HABICH, J., C. FEICHTINGER und G. WELLEIN: *waLBerla: MPI parallele Implementierung eines LBM Löser auf dem Tsubame 2.0 GPU Cluster*. Seminar Talk, Leibniz Rechenzentrum, München, Germany, Februar 2012.

- [81] HABICH, J., T. ZEISER, G. HAGER und G. WELLEIN: *Enabling temporal blocking for a lattice Boltzmann flow solver through multicore-aware wavefront parallelization*. Parallel CFD Conference, Moffett Field, California, USA, Mai 2009.
- [82] HABICH, J., T. ZEISER, G. HAGER und G. WELLEIN: *Multicore-aware wavefront parallelization of a Lattice Boltzmann flow solver*. 6th International Conference for Mesoscopic Methods in Engineering and Science, Guangzhou City, Guangdong (Canton) Province, China, Juli 2009.
- [83] HABICH, J., T. ZEISER, G. HAGER und G. WELLEIN: *Performance Evaluation of Numerical Compute Kernels on GPUs*. First International Workshop on Computational Engineering - Special Topic Fluid-Structure Interaction, Haus der Bayerischen Landwirtschaft, Herrsching am Ammersee, Oktober 2009.
- [84] HABICH, J., T. ZEISER, G. HAGER und G. WELLEIN: *Speeding up a lattice Boltzmann kernel on nVIDIA GPUs*. First International Conference on Parallel, Distributed and Grid Computing for Engineering (PARENG09-S01), Pecs, Hungary, April 2009.
- [85] IGLBERGER, K.: *Direct Numerical Simulation of Particulate Flows on 294912 Processor Cores*. 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, New Orleans, Louisiana, USA, November 2010.
- [86] IGLBERGER, K.: *Direct Numerical Simulation of Particulate Flows on 294912 Processor Cores*. International Workshop on Multiscale Modeling, Simulation and Optimization, Cluster of Excellence Engineering of Advanced Materials, Erlangen, Oktober 2010.
- [87] IGLBERGER, K.: *Towards Direct Numerical Simulation of a Billion Fully Resolved Rigid Bodies Immersed in a Fluid*. CiHPC Competence in High Performance Computing, HPC Status Conference of Gauß-Allianz e.V, Schwetzingen, Germany, Juni 2010.
- [88] J.GÖTZ, U. RÜDE, S. DONATH, C. FEICHTINGER, K. IGLBERGER und F. DESERNO: *Simulation and Animation of Complex Flows on 10,000 Processor Cores*. KONWIHR Results Workshop, Garching, Dezember 2009.
- [89] J.HABICH, CH. FEICHTINGER und G. WELLEIN: *GPGPU implementation of the LBM: Architectural Requirements and Performance Results*. ParCDF 2011, special session on LBM, Barcelona, Spain, May 2011.
- [90] J.HABICH, CH. FEICHTINGER, T. ZEISER, G. HAGER und G. WELLEIN: *Performance Modeling and Optimization for 3D Lattice Boltzmann Simulations on Highly Parallel On-Chip Architectures: GPUs Vs. Multi-Core CPUs*. ECCOMAS CFD, Lisbon, Portugal, Juni 2010.

- [91] J.HABICH, T. ZEISER und G. WELLEIN G. HAGER AND: *Performance Modeling and Multi-Core-aware Optimization for 3D Parallel Lattice Boltzmann Simulations*. Facing the Multicore-Challenge, Heidelberger Akademie der Wissenschaften, Germany, März 2010.
- [92] KÖSTLER, H.: *Numerical Algorithms on Multi-GPU Architectures*. 2nd International Workshops on Advances in Computational Mechanics, Yokohama, Japan, März 2010.
- [93] KÖSTLER, H.: *Numerical Codes on Multi-GPU Architectures*. ASIM Workshop on Trends in Computational Science and Engineering, Garching, Germany, March 2011.
- [94] KÖSTLER, H., C. FEICHTINGER, J. GÖTZ, S. DONATH und U. RÜDE: *HPC Software Design for Computational Engineering Simulations*. SIAM CSE 2011, Reno, Nevada, USA, March 2011.
- [95] PICKL, K.: *Simulation of Self-propelled Micro-devices with a Lattice Boltzmann Solver and a Rigid Body Physics Engine*. International Conference for Mesoscopic Methods in Engineering and Science 2011, Lyon, June 2011.
- [96] RABENSEIFNER, R., G. HAGER und G. JOST: *Hybrid MPI and OpenMP Parallel Programming*. Tutorial, SC09, Portland/OR, USA, November 2009.
- [97] RÜDE, U.: *Massively Parallel Algorithms for Fluid-Structure Interaction with Moving Objects*. FSI-Workshop: Simulation Software for Supercomputers, Herrsching, Oktober 2009.
- [98] RÜDE, U.: *Simulation and Animation of Complex Flows on 10,000 Processor Cores*. Eingeladener Vortrag. SC09, Portland/OR, USA, November 2009.
- [99] RÜDE, U.: *Peta-Scale Computing for the Direct Numerical Simulation of Particle Laden Flows*. International Workshop on Scientific Computing and Nonlinear Partial Differential Equations, Jiuzhaigou National Park, China, Juni 2010.
- [100] RÜDE, U.: *Simulating Complex Flows on 300,000 Compute Cores*. Einladung des State Key Laboratory of Scientific and Engineering Computing (LSEC) der Academy of Sciences, Beijing, China, Mai 2010.
- [101] RÜDE, U.: *Simulation and Animation of Complex Flows Using 294,192 Processor Cores*. International Conference on Simulation Technology, Stuttgart, Germany, June 2011.
- [102] WELLEIN, G.: *Effiziente Nutzung von Hochleistungsrechnern in der Strömungsmechanik*. Kurzlehrgang NUMET Numerische Methoden zur Berechnung von Strömungs- und Wärmeübertragungsproblemen, Lehrstuhl für Strömungsmechanik, Februar 2010.

- [103] WELLEIN, G.: *Supercomputing und Multicore-Systeme*. Ohm-Hochschule Nürnberg, Fachbereich EFI, Juni 2010.
- [104] WELLEIN, G.: *Application Performance in the Multi-Core Era – Lessons to Be Learned for Exascale Computing (Tales from the trenches)*. invited; SIAM CSE 2011, Reno, Nevada, USA, March 2011.
- [105] WELLEIN, G., J. HABICH, G. HAGER und T. ZEISER: *Node-level performance of the lattice Boltzmann method on recent multicore CPUs*. ParCDF 2011, special session on LBM, Barcelona, Spain, May 2011.
- [106] WELLEIN, G., G. HAGER und J. HABICH: *The Lattice Boltzmann Method: Basic Performance Characteristics and Performance Modeling*. SIAM CSE 2011, Reno, Nevada, USA, March 2011.
- [107] WELLEIN, G., G. HAGER, M. WITTMANN, J. HABICH und J. TREIBIG: *Multicore-Aware Parallelization Strategies for Efficient Temporal Blocking*. CiHPC Competence in High Performance Computing, HPC Status Conference of Gauß-Allianz e.V, Schwetzingen, Germany, Juni 2010.
- [108] WELLEIN, G., G. HAGER, T. ZEISER, H. FEHSKE, M. WITTMANN, J. HABICH und J. TREIBIG: *Enabling temporal blocking for stencil computations by multicore-aware wavefront parallelization*. National Energy Research Scientific Computing Center (NERSC), Mai 2009.
- [109] WELLEIN, G., G. HAGER, T. ZEISER und J. HABICH: *Numerical simulation on recent multi-core designs: Some comments on programming effort and performance*. First International Workshop on Computational Engineering - Special Topic Fluid-Structure Interaction, Haus der Bayerischen Landwirtschaft, Herrsching am Ammersee, Oktober 2009.
- [110] WELLEIN, G., G. HAGER, T. ZEISER, M. WITTMANN und H. FEHSKE: *Efficient temporal blocking for stencil computations by multicore-aware wavefront parallelization*. Best paper award. 33rd IEEE International Computer Software and Applications Conference (COMPSAC), Seattle, Juli 2009.
- [111] WELLEIN, G. und T. ZEISER: *Application Performance in the Multi-Core Era – Lessons to Be Learned for Exascale Computing (Tales from the trenches)*. invited; NEC User Group, Prague, June 2011.
- [112] WITTMANN, M.: *A Pipelined, Multicore-aware Approach to Parallel Temporal Blocking of Stencil Codes for Shared and Distributed Memory*. Facing the Multicore-Challenge, Heidelberger Akademie der Wissenschaften, Germany, März 2010.
- [113] WITTMANN, M., G. HAGER und G. WELLEIN: *Multicore-Aware Parallel Temporal Blocking of Stencil Codes for Shared and Distributed Memory*. LSPP Workshop, IPDPS 2010, Atlanta, USA, April 2010.

- [114] WITTMANN, M. und T. ZEISER: *Parttitioning for lattice Boltzmann solver*. LBM Day, ICAMMS, Ruhr-Universität Bochum, Dezember 2010.
- [115] WITTMANN, M. und T. ZEISER: *Domain decomposition and locality optimization for large-scale lattice Boltzmann simulations*. ParCDF 2011, special session on LBM, Barcelona, Spain, May 2011.
- [116] ZEISER, T.: *Efficient implementation of lattice Boltzmann flow solvers*. Eingeladenes Tutorial. 6th International Conference for Mesoscopic Methods in Engineering and Science, Guangzhou City, Guangdong (Canton) Province, China, Juli 2009.
- [117] ZEISER, T.: *High Performance Computing in the Multi-Core Era – Why care about performance*. Eingeladener Vortrag. Richard F. Barry Mathematics & Statistics Colloquium and Computer Science Colloquium, Old Dominion University, Norfolk/VA, USA, Oktober 2009.
- [118] ZEISER, T.: *Performance studies on the NEC SX-9 vector system & Optimization and parallelization of a highly scalable lattice Boltzmann flow solver for porous media flow*. Berkeley Lab - Computing Sciences Seminars, LBNL/NER-SC, Berkeley/CA, USA, November 2009.
- [119] ZEISER, T.: *HPC for LBM – Activities at RRZE and within the BMBF project SKALB*. LBM Day, ICAMMS, Ruhr-Universität Bochum, Dezember 2010.
- [120] ZEISER, T.: *Strömungsberechnung mit Lattice-Boltzmann-Verfahren*. Kurzlehrgang NUMET Numerische Methoden zur Berechnung von Strömungs- und Wärmeübertragungsproblemen, Lehrstuhl für Strömungsmechanik, Februar 2010.
- [121] ZEISER, T.: *Brief introduction to LBM*. ParCDF 2011, special session on LBM, Barcelona, Spain, May 2011.
- [122] ZEISER, T. und G. HAGER: *Thirteen Modern Ways to Fool the Masses with Performance Results on Parallel Computers*. Eingeladener Seminar-Vortrag, iRMB, TU-Braunschweig, Juni 2010.
- [123] ZEISER, T. und G. HAGER: *Thirteen Modern Ways to Fool the Masses with Performance Results on Parallel Computers*. Eingeladener Vortrag auf der NEC-User-Group, Toulouse, Frankreich, Juni 2010.
- [124] ZEISER, T., G. HAGER und G. WELLEIN: *The world's fastest CPU and SMP node: Some performance results from the NEC SX-9*. Workshop on Large-Scale Parallel Processing, IEEE International Parallel and Distributed Processing Symposium, Rome, Italy, Mai 2009.
- [125] ZEISER, T., G. HAGER und G. WELLEIN: *The world's fastest CPU and SMP node: Some performance results from the NEC SX-9*. 12th Results and Review Workshop of the HLRS, Stuttgart, Oktober 2009.

- [126] ZEISER, T., M. WITTMANN, J. HABICH, G. HAGER und G. WELLEIN: *Implementation and optimization strategies for the lattice Boltzmann method*. International Conference for Mesoscopic Methods in Engineering and Science 2011, Lyon, June 2011.

2.6.6 Vorträge TUDo

- [127] BUIJSSEN, SVEN: *Advanced ParaView Visualisation. Tips and tricks beyond GMV-like operations*, Graduate Class Numerical Simulation, 2009.
- [128] BUIJSSEN, SVEN: *Importing FeatFlow / FEAST / SOFAR Visualisation Output into ParaView. Introducing the GMV Reader Plugin*, Graduate Class Numerical Simulation, 2009.
- [129] BUIJSSEN, SVEN: *Introduction to ParaView. Stepping from GMV to ParaView*, Graduate Class Numerical Simulation, 2009.
- [130] GEVELER, M.: *Complex Shallow Water Simulations through Lattice-Boltzmann-based Mesoscopic Numerical Treatment Exploiting Hybrid Compute Nodes*, März 2011. SIAM Conference on Mathematical and Computational Issues in the Geosciences, Long Beach.
- [131] GEVELER, M.: *Advanced, hardware-oriented Shallow Water simulations based on the Lattice-Boltzmann Method*, Juni 2012. accepted in: XIX International Conference on Computational Methods in Water Resources, special session on Heterogeneous Computing for Water Resources, Urbana-Champaign.
- [132] GEVELER, M.: *Hardwareorientierte Numerik für FEM-Frameworks*, April 2012. G2CG, 1st meeting of the German GPU Computing Group.
- [133] GEVELER, M., D. RIBBROCK, D. GÖDDEKE und S. TUREK: *Hardware oriented numerics for real-time CFD applications on parallel hardware architectures*, Oktober 2009. Symposium CFD on Future Architectures - Many-Cores, GPUs, FPGAs, Braunschweig.
- [134] GEVELER, M., D. RIBBROCK, D. GÖDDEKE und S. TUREK: *Lattice-Boltzmann Simulation of the Shallow-Water Equations with Fluid-Structure Interaction on Multi- and Many-core Processors*, März 2010. Facing the Multicore Challenge, Heidelberg.
- [135] GEVELER, M., D. RIBBROCK, D. GÖDDEKE, P. ZAJAC und S. TUREK: *Efficient Finite Element Geometric Multigrid Solvers for Unstructured Grids on Graphics Processing Units*, April 2011. The Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, Ajaccio.

- [136] GEVELER, M., D. RIBBROCK, D. GÖDDEKE, P. ZAJAC und S. TUREK: *Towards a complete FEM-based simulation toolkit on GPUs: Geometric Multigrid solvers*, Mai 2011. 23rd International Conference on Parallel Computational Fluid Dynamics (ParCFD'11), Barcelona.
- [137] GÖDDEKE, DOMINIK: *Fast and Accurate Finite Element Multigrid Solvers for PDE Simulations on GPU Clusters*, April 2011. Kolloquium über Angewandte Mathematik, Georg-August-Universität Göttingen.
- [138] GÖDDEKE, DOMINIK: *Hardware-Oriented Finite Element Multigrid Solvers for PDEs*, März 2011. ASIM Workshop Trends in Computational Science and Engineering, Garching.
- [139] GÖDDEKE, DOMINIK: *High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster*, Dezember 2011. AGU Fall Meeting, special session 'High-Resolution Modeling in the Geosciences Using GPU and Many-Core Architectures', San Francisco, USA.
- [140] GÖDDEKE, DOMINIK: *Mixed-Precision GPU-Multigrid Solvers with Strong Smoothers*, August 2011. ILAS Conference: Pure and Applied Linear Algebra: The new Generation. Young Researchers' mini-symposium 'Parallel Computing in Numerical Linear Algebra', Technische Universität Braunschweig.
- [141] GÖDDEKE, DOMINIK: *Mixed-Precision GPU-Multigrid Solvers with Strong Smoothers and Applications in CFD and CSM*, Juni 2011. SIMTECH 2011 International Conference on Simulation Technology.
- [142] GÖDDEKE, DOMINIK, SVEN H.M. BUIJSSEN, HILMAR WOBKER und STEFAN TUREK: *GPU Cluster Computing for FEM with Applications in CFD and CSM*, Juni 2010. ECCOMAS-CFD 2010, Lisbon, Portugal.
- [143] GÖDDEKE, DOMINIK und ROBERT STRZODKA: *Mixed-Precision GPU-Multigrid Solvers with Strong Smoothers and Applications in CFD and CSM*, September 2011. ENUMATH 2011 Mini-Symposium 'Advanced Numerical Methods on GPUs', Leicester, UK.
- [144] KOMATITSCH, DIMITRI, DOMINIK GÖDDEKE, GORDON ERLEBACHER und DAVID MICHÉA: *Modeling the propagation of elastic waves using spectral elements on a cluster of 192 GPUs*, Mai 2010. ISC, Hamburg.
- [145] RIBBROCK, D., M. GEVELER, D. GÖDDEKE und S. TUREK: *Performance and Accuracy of Lattice-Boltzmann Kernels on Multi- and Manycore Architectures*, Mai 2010. International Conference on Computer Science, Amsterdam.

2.6.7 Vorträge iRMB

- [146] GEIER, M., M. SCHÖNHERR, M. STIEBLER und M. KRAFCZYK: *Compact second-order accurate momentum interpolation for the lattice Boltzmann method in three dimensions*. International Conference for Mesoscopic Methods in Engineering and Science 2010, Edmonton, Canada, Juli 2010.
- [147] KRAFCZYK, M.: *Gitter-Boltzmann-Methoden für gekoppelte Strömungsprobleme*. Eingeladener Vortrag bei der Siemens-AG in München, 2010.
- [148] KRAFCZYK, M.: *LB-methods for engineering CFD: potentials and problems*. Workshop: Multiscale fluid dynamics with the Lattice Boltzmann method, 2010.
- [149] KRAFCZYK, M., B. AHRENHOLZ, S. FREUDIGER, M. GEIER, M. SCHÖNHERR und K. KUCHER: *Massively Parallel Simulation of Three-Dimensional Transient Coupled CFD Problems with Kinetic Methods*. CiHPC Competence in High Performance Computing, HPC Status Conference of Gauß-Allianz e.V, Schwetzingen, Germany, Juni 2010.
- [150] SCHÖNHERR, M., M. GEIER und M. KRAFCZYK: *Non-Uniform LB Simulations on GPUS*. International Conference for Mesoscopic Methods in Engineering and Science 2010, Edmonton, Canada, Juli 2010.
- [151] SCHÖNHERR, M., M. GEIER und M. KRAFCZYK: *3D GPGPU LBM Implementation on Non-Uniform Grids*. ParCDF 2011, special session on LBM, Barcelona, Spain, May 2011.
- [152] UPHOFF, S., M. GEIER, C. JANSSEN und M. KRAFCZYK: *Comparison of Cascaded and MRT Lattice Boltzmann models for low and medium Reynolds number flow*. International Conference for Mesoscopic Methods in Engineering and Science 2010, Edmonton, Canada, Juli 2010.
- [153] WANG, Y., M. KRAFCZYK und B. AHRENHOLZ: *Modeling Evaporation Mechanism with Lattice Boltzmann Multiphase Models*. International Conference for Mesoscopic Methods in Engineering and Science 2010, Edmonton, Canada, Juli 2010.

2.6.8 Vorträge HLRS

- [154] JENZ, D. und M. BERNREUTHER: *The Computational Steering Framework Steereo*. Para 2010, Reykjavik, Juni 2010.
- [155] JENZ, D. und M. BERNREUTHER: *A Framework for Computational Steering*. CiHPC Competence in High Performance Computing, HPC Status Conference of Gauß-Allianz e.V, Schwetzingen, Germany, Juni 2010.

2.6.9 Vorträge IANUS

- [156] PLATTE, F. und G. WELLEIN: *Performante und skalierbare Lattice-Boltzmann-Löser für Strömungsprobleme in Wissenschaft und Technik*. Statusseminar der Projekte aus dem ersten BMBF-HPC-Call, Berlin, Dezember 2010.
- [157] PLATTE, F. und G. WELLEIN: *SKALB: Skalierbare Lattice-Boltzmann-Löser: Von hochskalierenden, heterogenen Parallelrechnern zu industrienahen Simulationen*. HPC-Status-Konferenz der Gauß-Allianz, Darmstadt, Dezember 2011.

2.7 Sonstige Referenzen

- [Bec07] BECKER, C.: *Strategien und Methoden zur Ausnutzung der High-Performance-Ressourcen moderner Rechnerarchitekturen für Finite-Element-Simulationen und ihre Realisierung in FEAST (Finite Element Analysis & Solution Tools)*. Doktorarbeit, Univ. Dortmund, Logos Verlag, Berlin, 2007.
- [BFL01] BOUZIDI, M., M. FIRDAOUSS und P. LALLEMAND: *Momentum Transfer of a Boltzmann-lattice fluid with boundaries*. *Physics of Fluids*, 13(11):3452–3459, 2001.
- [BMT11] BAYRAKTAR, E., O. MIERKA und S. TUREK: *Benchmark Computations of 3D Laminar Flow Around a Cylinder with CFX, OpenFOAM and FeatFlow*. Technischer Bericht, Fakultät für Mathematik, TU Dortmund, September 2011. Ergebnisberichte des Instituts für Angewandte Mathematik, Nummer 433.
- [BMW⁺09] BAILEY, P., J. MYRE, S.D.C. WALSH, D.J. LILJA und M. O. SAAR: *Accelerating Lattice Boltzmann Fluid Flow Simulations Using Graphics Processors*. In: *Proceedings of the 2009 International Conference on Parallel Processing, ICPP*, Seiten 550–557, 2009.
- [FGD⁺07] FEICHTNER, C., J. GÖTZ, S. DONATH, K. IGLBERGER und U. RÜDE: *Concepts of waLBerla Prototype 0.1*. Technischer Bericht 07-10, University of Erlangen-Nuremberg, Computer Science 10 – Systemsimulation, 2007.
- [Hän04] HÄNEL, D.: *Molekulare Gasdynamik: Einführung in die kinetische Theorie der Gase und Lattice-Boltzmann-Methoden*. Springer-Verlag, Berlin, Heidelberg, 2004.
- [Kil01] KILIAN, S.: *ScaRC: Ein verallgemeinertes Gebietszerlegungs-/Mehrgitterkonzept auf Parallelrechnern*. Doktorarbeit, Universität Dortmund, Fachbereich Mathematik, 2001.
- [KT98] KILIAN, S. und S. TUREK: *An Example for Parallel ScaRC and its Application to the Incompressible Navier-Stokes Equations*. In: BOCK, H. G., Y. A. KUZNETSOV, R. GLOWINSKI, J. PERIAUX und R. RANACHER (Herausgeber): *ENUMATH 97: Proceedings of the 2nd European Conference on Numerical Mathematics and Advanced Applications*, Seiten 389–396. World Science Publishing, 1998.
- [TBK06] TUREK, S., C. BECKER und S. KILIAN: *Hardware-oriented Numerics and Concepts for PDE Software*. *Future Generation Computer Systems*, 22(1-2):217–238, 2006.
- [Tur99] TUREK, S.: *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*. Springer, Berlin, 1999.

- [WZDH06] WELLEIN, G., T. ZEISER, S. DONATH und G. HAGER: *On the single processor performance of simple lattice Boltzmann kernels*. *Computers & Fluids*, 35(8-9):910–919, 2006.
- [Zei08] ZEISER, T.: *Simulation von durchströmten Schüttungen auf Hochleistungsrechnern*. Doktorarbeit, Technische Fakultät, Universität Erlangen-Nürnberg, 2008.