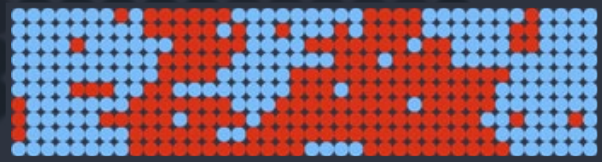

2nd INTERNATIONAL SYMPOSIUM

COMPUTER SIMULATIONS ON GPU

May 27–29, 2013

Freudenstadt, Germany

BOOK OF ABSTRACTS

Further information
http://simgpu2013.complexity-coventry.org

Funded by
VolkswagenStiftung

Organisation supported by
SFB 716

Preface

The second symposium on "Computer Simulations on Graphics Processing Units", SimGPU 2013 in Freudenstadt, is a follow-up event to the successful SimGPU 2011 symposium in Mainz. These symposia intend to bring together experts in the quickly moving field of GPU computing, ranging from applications to specialists in high-performance computing and representatives from computer industry, discussing their experience in designing and using accelerators like graphics processing units for pushing the limits of computationally tractable problems.

The symposium is focused on simulational methods in physics and the computational and algorithmic aspects in computer science and computer industry. Subjects covered include

- High Performance Computing on Modern Architectures
- Applications in Computer and Natural Sciences
- Molecular Simulation
- Computational Fluid Dynamics
- Classical Spin Systems and Spin Dynamics

The symposium starts on Monday, May 27 and finishes on Wednesday, May 29. The symposium is held at Waldhotel Zollernblick, located in Freudenstadt in the beautiful Black Forest region, Germany. All participants are lodged in the facilities of the hotel, allowing for an exchange of ideas even at the after hours.

We gratefully acknowledge financial support of the Symposium by the Volkswagen Foundation under grant No. 85785 and organizational support from the DFG SFB 716 "Dynamic simulation of systems with large particle numbers".

Stuttgart/Mainz/Conventry, May 2013

Axel Arnold
Peter Virnau
Martin Weigel

Contents

Monday, May 27

11:45 – 12:45		**– Lunch –**
12:50 – 13:00		**– Opening remarks –**

Session 1: High Performance Computing on Modern Architectures I (Chair: Martin Weigel)

13:00 – 13:30	**Axel Köhler**	GPU Acceleration Benefits for Science
13:35 – 14:05	**Andre Heidekrüger**	AMD, Germany, TBA
14:10 – 14:40		**– Coffee Break –**

Session 2: Applications in Computer and Natural Sciences (Chair: Axel Arnold)

14:40 – 15:10	**Tobias Kramer**	GPU computing in physics and chemistry: from workstations to smartphones
15:15 – 15:30	**Jeffrey Kelling**	Performing kinetic lattice Monte-Carlo simulations of far-from-equilibrium processes on GPUs
15:35 – 16:05	**Andrei Alexandru**	Efficient implementation of the overlap operator on multi-GPUs
16:10 – 16:40	**Bertil Schmidt**	Algorithms and Tools for Bioinformatics on GPUs
16:45 – 18:30		**– Time for discussions –**
19:00		**– Dinner at hotel –**

Tuesday, May 28

Session 3: High Performance Computing on Modern Architectures II (Chair: Tobias Kramer)

9:00 – 9:30	**Thomas Horn**	Programming OpenACC with PGI Compilers
9:35 – 10:05	**Christian Plessl**	Parallelism and Customization: Trends in Computer Architectures for Accelerating Scientific Computing
10:10 – 10:40		**– Coffee Break –** 

Session 4: Molecular Simulation I (Chair: Peter Virnau)

10:40 – 11:10	**Joshua Anderson**	Efficient techniques for massively parallel many-particle simulations on GPUs
11:15 – 11:30	**Nikita Tretyakov**	Directed motion of droplets on asymmetrically structured vibrating substrates: a Molecular Dynamics study
11:35 – 12:05	**Jan Meinke**	Proteins, Snakes, and Games of Chance
12:10 – 13:10		**– Lunch –**
13:10 – 15:00		**– Poster Session –**
15:00 – 15:30		**– Coffee Break –** 

Session 5: Molecular Simulation II (Chair: Joshua Anderson)

15:30 – 16:00	**Marco Werner**	The bond fluctuation model on graphics processing units: applications and new perspectives
16:05 – 16:20	**Ron Dockhorn**	GPU implementations of the bond fluctuation model: Technical Details
16:25 – 16:40	**Phillip Schierz**	A GPU accelerated Metropolis-Hastings algorithm for particle simulations

16:45 – 17:05	**Alexander Vondrous**	N-dimensional domain decomposition and its effects on Speedup and Efficiency regarding scalability
19:00		**– Departure Conference Dinner –**
22:00		**– Return to Hotel –**

Wednesday, May 29

Session 6: Computational Fluid Dynamics (Chair: Axel Arnold)

9:00 – 9:30	**Harald Köstler**	Lattice Boltzmann Simulations on Heterogeneous CPU-GPU Clusters
9:35 – 10:05	**Alan Gray**	Scaling Soft Matter Physics to Thousands of GPUs in Parallel
10:10 – 10:25	**Gyorgy Tegze**	A GPU cluster optimized multigrid scheme for computing unsteady incompressible fluid flow
10:30 – 11:00		**– Coffee Break –**

Session 7: Classical Spin Systems and Spin Dynamics (Chair: Andrei Alexandru)

11:00 – 11:30	**Benjamin Block**	Determination of line tension in the 3d Ising model using GPUs
11:35 – 11:50	**Stefan Gerlach**	Ultrafast spin dynamics simulations using OpenCL
11:55 – 12:00		**– Closing remarks –**
12:00 – 13:00		**– Lunch –**
13:00		**– Departure –**

GPU Acceleration Benefits for Science

Axel Köhler
Senior Solution Architect HPC, NVIDIA, Germany
akoehler@nvidia.com

Monday, May 27, 13:00 – 13:30

Computational researchers, scientists and engineers are rapidly shifting to computing solutions running on GPUs as this offers significant advantages in performance and energy efficiency. This presentation will give a short overview about the latest Kepler GPU architecture and will show how features like Hyper-Q, GPU-aware MPI and GPUDirect RDMA improve the performance, scalability and GPU utilization of scientific applications. With multi-GPU nodes and large-scale GPU clusters the optimal mapping and scheduling of GPU resources dependent on the hardware topology plays an important role. Mechanisms for this mapping and the integration into MPI applications will also be covered in the talk. In addition an update about NVIDIA's parallel computing platform, the CUDA Compiler SDK and future developments will be presented.

GPU computing in physics and chemistry: from workstations to smartphones

Tobias Kramer
Humboldt Universität zu Berlin, Germany
tobias.kramer@physik.hu-berlin.de

Monday, May 27, 14:40 – 15:10

New supercomputers increasingly boost their performance by GPUs, but most physics codes are not ready to take advantage of massively parallel processors on a single chip. Also an automatic code conversion does not achieve the highest possible performance. This requires to hand code GPU kernels and to understand details of the GPU hardware and software stack. In addition GPUs have arrived around us in commodity hardware, including smartphones with surprisingly good numerical performance.
To write GPU OpenCL code which runs efficiently on a wide variety of hardware platforms requires some design considerations, but is possible as demonstrated in this talk for physics applications ranging from condensed matter theory to energy transfer in photosynthetic bacteria.
Code examples and references can be found at http://quantumdynamics.de

Performing kinetic lattice Monte-Carlo simulations of far-from-equilibrium processes on GPUs

Jeffrey Kelling
Helmholtz Zentrum Dresden-Rossendorf, Germany
j.kelling@hzdr.de

Monday, May 27, 15:15 – 15:30

Kinetic Metropolis lattice Monte-Carlo (KLMC) simulations can be applied to study systems without strong external driving forces, for example the evolution of nanostructures. An algorithm allowing nearest neighbor interaction in thermally activated processes has been implemented for GPUs. External driving forces can be modeled by adding appropriate rules to the cellular automaton. Ion-beam induced mixing, which is governed by comparatively long-ranged interactions, leads to far-from-equilibrium processes (defect relaxation) producing disordered structures. A plain implementation of long-range interactions reduces the performance on GPUs substantially. In this talk we present an efficient implementation of long-range ballistic displacements in GPU-KLMC.

The code to be presented is based on the KLMC implementation presented in: J. Kelling, G. Ódor, M. F. Nagy, H. Schulz, and K. Heinig. *Comparison of different parallel implementations of the 2+1-dimensional KPZ model and the 3- dimensional KMC model* The European Physical Journal - Special Topics, **210**:175-187, 2012

Efficient implementation of the overlap operator on multi-GPUs

Andrei Alexandru
George Washington University, Washington, USA
andrei.alexan@gmail.com

Monday, May 27, 15:35 – 16:05

Lattice QCD calculations were one of the first applications to show the potential of GPUs in the area of high performance computing. Our interest is to find ways to effectively use GPUs for lattice calculations using the overlap operator. The large memory footprint of these codes requires the use of multiple GPUs in parallel. In this talk I will discuss the methods we used to implement this operator efficiently. We run our codes both on a GPU cluster and a CPU cluster with similar interconnects. We find that to match performance the CPU cluster requires 20-30 times more CPU cores than GPUs.

Algorithms and Tools for Bioinformatics on GPUs

Bertil Schmidt
University of Mainz, Germany
bertil.schmidt@uni-mainz.de

Monday, May 27, 16:10 – 16:40

Abstract: High-throughput techniques for DNA sequencing have led to a rapid growth in the amount of digital biological data. The current state-of-the-art technology produces 600 billion nucleotides per machine run. Furthermore, the speed and yield of NGS (Next-generation sequencing) instruments continue to increase at a rate beyond Moore's Law, with updates in 2012 enabling 1 trillion nucleotides per run. Correspondingly, sequencing costs (per sequenced nucleotide) continue to fall rapidly, from several billion dollars for the first human genome in 2000 to a forecast US$1000 per genome by the end of 2013. However, to be effective, the usage of NGS for medical treatment will require algorithms and tools for sequence analysis that can scale to billions of short reads. In this talk I will demonstrate how parallel computing platforms based on CUDA-enabled GPUs, multi-core CPUs, and heterogeneous CPU/GPU clusters can be used as efficient computational platforms to design and implement scalable tools for sequence analysis. I will present solutions for classical sequence alignment problems (such as pairwise sequence alignment, BLAST, multiple sequence analysis, motif finding) as well as for NGS algorithms (such as short-read error correction, short-read mapping, short-read assembly, short-read clustering).

Programming OpenACC with PGI Compilers

Thomas Horn
SMB/PGI, Germany
thomas.horn@smb-net.de

Tuesday, May 28, 9:00 – 9:30

OpenACC is a programming standard developed by Cray, CAPS, NVIDIA and PGI to simplify parallel programming of heterogeneous systems. In the beginning developed for CPU/GPU systems it comes now to an open industry standard for heterogeneous systems supporting NVIDIA, AMD and Intel Accelerators. With OpenMP like pragmas the programmer can annotate regions in C, C++ and Fortran code that can be offloaded to an accelerator if any is available. With the OpenACC API, a specification of directives and runtime routines, the programmer can designate loops that should run on an accelerator device and the data that needs to be copied to the device memory. Like in OpenMP one of the goals was to preserve the origin structure of the code and not to be forced to redesign the code for every new accelerator or CPU platform which comes to market.
PGI who developed large parts of the pragma model offers a broad range of different compilers to cope all the different possible development paths around these heterogeneous architectures. From auto parallelization of standard, OpenMP and MPI Code on x86 CPUs to OpenACC and NVIDIA CUDA dialects like CUDA Fortran and the migration back from GPU to CPU code with CUDA C x86 every code path is supported. Future developments will support AMD APUs and Intels MIC accompanied by the steadily implementation of the OpenACC specification for NVIDA GPUs.
Learn about the techniques to accelerate code regions step by step with the OpenACC runtime directives. PGIs implementation of new OpenACC Features and techniques will be shown and discussed.

Parallelism and Customization: Trends in Computer Architectures for Accelerating Scientific Computing

Christian Plessl
University of Paderborn, Germany
christian.plessl@uni-paderborn.de

Tuesday, May 28, 9:35 – 10:05

Parallel computing has been the cornerstone of scientific and high-performance computing (HPC) for decades. In the last two decades parallelism and thus performance was mainly driven by scaling out. That is, an ever larger number of servers have been connected with high-bandwidth low-latency interconnection networks to form distributed memory computer clusters. In recent years, new computer architectures have emerged that apply the principle of parallel computing within a server node by using massively parallel processing at the chip-level. This architectures use a large number of simpler–and in some cases even customizable–computing elements, resulting in a drastic increase in available compute power and energy efficiency per server node.
In this talk I will discuss these developments from a computer architecture point of view. I will highlight the commonalities and differences of the current processor architectures and discuss how these properties affect the programmability and suitability of the technologies for particular classes of applications by means of case studies.

Efficient techniques for massively parallel many-particle simulations on GPUs

Joshua Anderson
University of Michigan, USA
joaander@umich.edu

Tuesday, May 28, 10:40 – 11:10

Monte Carlo and Molecular Dynamics simulations are standard tools for analyzing the thermodynamic and statistical behavior of many-particle systems. One of the earliest computer simulations computed the equation of state of 224 disks. Now, massive parallelism enables routine simulations of millions of particles. In this talk, we describe our novel GPU Monte Carlo algorithm and compare it with HOOMD-blue, our open-source Molecular Dynamics code. Recent improvements to HOOMD-blue make possible parallel multiple GPU simulations on workstations and clusters. Applications include polymer dynamics, granular materials, non-equilibrium systems, and hard particle self-assembly.

Directed motion of droplets on asymmetrically structured vibrating substrates: a Molecular Dynamics study

Nikita Tretyakov
MPI for Polymer Research in Mainz, Germany
tretyakov@mpip-mainz.mpg.de

Tuesday, May 28, 11:15 – 11:30

We study a transport of coarse-grained polymer droplets on asymmetrically structured (saw-tooth shaped) and vibrating substrates by means of Molecular Dynamics. The temperature of the system is controlled by the DPD thermostat. Due to a continuous supply of power by substrate vibrations and the asymmetry of its topography, the droplets are driven in a preferred direction. This directed motion is investigated as a function of the size of the droplets, linear dimension of the substrate corrugation and the period of substrate vibrations. To this end, our main concern consists in resolving two questions: (i) what is the driving mechanism and (ii) what is the character of the flow inside the droplet?
A typical mechanism of driving is the one provided by droplet's contact lines, as the responses of the advancing and receding contact lines on the vibrating asymmetric substrate are not identical. We, moreover, find a range of vibration periods that lead to an additional driving by the contact area of the substrate between the contact lines.
For the character of the droplet motion there are, in general, three possibilities: sliding, rotating and a combination of both. All of them dissipate the input power by different means. We find that for most of the parameters the droplets are sliding, but the linear size of the substrate corrugation may give a rise to an additional rotation.

Proteins, Snakes, and Games of Chance

Jan Meinke
Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany
j.meinke@fz-juelich.de

Tuesday, May 28, 11:35 – 12:05

Proteins are the workhorses of the cell. They transport nutrients, fight diseases, catalyze reactions, and help build the cell. Their unique shapes are encoded in their specific amino-acid sequence. How the sequence of amino acids controls a protein's shape, however, is poorly understood.
The process by which a protein moves from an extended chain of amino acids to a compact three-dimensional structure is called folding. Exploring the thermodynamics of protein folding using simulations is computationally expensive due to the vast conformational space and the computational cost of the energy function.
In this talk I'm showing how I am using Python and OpenCL to iteratively move SMMP, a Monte Carlo code for protein simulations, to take advantage of accelerators such as GPUs and Intel's Xeon Phi.

The bond fluctuation model on graphics processing units: applications and new perspectives

Marco Werner, Christoph Jentzsch, Ron Dockhorn, Jens-Uwe Sommer
Leibniz-Institut für Polymerforschung Dresden e.V., Germany
werner-marco@ipfdd.de

Tuesday, May 28, 15:30 – 16:00

Recent implementations of the bond fluctuation model (BFM) [1] on graphics processing units have provided a high performance to study static and dynamic properties of polymer structures on a coarse grained level [2]. Applied to some current focus topics such as the dynamics in polymer melts, and polymer brush dynamics and the swelling of polymer networks, we demonstrate that lattice Monte-Carlo simulations on GPU can give access to considerably larger time- and length scales. We discuss strategies to face conceptional and technical challenges for parallel BFM simulations with local Monte-Carlo moves only. Here, parallel interfering moves, which would lead to collisions, have to be prevented. We present in more detail a parallel version of a new bond fluctuation model based on an alternative lattice geometry, which allows for independent monomer moves without collisions: This is achieved by defining four subsets of monomers, in which intrinsically monomer moves can be performed independently. Characteristic features of this model and optimization techniques using CUDA will be presented and compared with the classic BFM.

1. H.-P. Deutsch and K. Binder, J. Chem. Phys. **94**, 2294 (1991)

2. S. Nedelcu, M. Werner, M. Lang, J.-U. Sommer, J. Comp. Phys. **231**, 2811-2824 (2012)

GPU implementations of the bond fluctuation model: Technical Details

Ron Dockhorn
Leibniz-Institut für Polymerforschung Dresden e.V., Germany
dockhorn@ipfdd.de

Tuesday, May 28, 16:05 – 16:20

see M. Werner et al, "The bond fluctuation model on graphics processing units: applications and new perspectives"

A GPU accelerated Metropolis-Hastings algorithm for particle simulations

Phillip Schierz, W. Janke
Universitiy of Leipzig, Germany
Philipp.Schierz@itp.uni-leipzig.de

Tuesday, May 28, 16:25 – 16:40

The Metropolis-Hastings algorithm offers a possibility for doing statistical particle simulations. But unlike in the Molecular Dynamics approach the algorithm itself can not be parallelised easily. In the Metropolis-Hastings method a simulation step is usually done for a subset of particles and is depending on the previous moves.
So far the attempts for parallelization are mostly to do independent statistical simulations on several cores to obtain a large amount of statistical data. But usually there is no attempt for particle simulations to accelerate single steps of the algorithm (apart from partial energy summation).
In this talk a method will be presented to use a computation algorithm that is doing the necessary calculations by a team up of GPU and CPU. The GPU is preparing several Metropolis-Hastings steps by a precalculation for the energies. Therefore the computational afford for the statistical step on the CPU is reduced to a small fraction. The implementation of the GPU code is done via OPEN CL.
This method should of course be parallelizable again by many CPU-GPU groups to produce the statistic in parallel for independent start situations and will therefore multiply the already obtained acceleration by an additional factor.

N-dimensional domain decomposition and its effects on Speedup and Efficiency regarding scalability

Alexander Vondrous
AMD, Germany
vondrous@kit.edu

Tuesday, May 28, 16:45 – 17:05

Nowadays the availability of a huge number of processing units, e.g. CPUs or GPUs, connected to each other enables the distribution of the computational problem. With the phase-field method as a representative structured grid application,we show the benefit of three-dimensional domain decomposition over one-dimensional domain decomposition.
Over the last two decades, the phase-field method has evolved to a powerful and important method to describe a broad variety of phenomena, like surface driven processes or coupled physical phenomena, like diffusion processes, fluid flow, mechanical forces or magnetic fields. The challenge to simulate larger domains with more complex models requires an efficient use of high performance clusters, scaleable algorithms and parallel data structures.
We emphasize the efficiency of high performance computing resources by investigating the scaling behavior of one-dimensional domain decomposition and three-dimensional domain decomposition for a finite difference implementation. A performance model for blocking communication and measurements shows that it is necessary to apply n-dimensional domain decomposition for an n-dimensional domain, to efficiently scale on connected computing nodes. Three-dimensional domain decomposition enables the usage of many computing nodes to simulate large domains, which leads to new requirements regarding memory usage, simulation parameter distribution and the parallelization of algorithms. To address these issues, we show parameter caching mechanisms and how to take advantage of spare matrices to reduce memory consumption, computation and communication time.
The performance model shows a good scaling behavior of three dimensional domain decomposition. Measurements with up to 10 000 processing units confirm the expected estimations of the performance model.

Lattice Boltzmann Simulations on Heterogeneous CPU-GPU Clusters

Harald Köstler
FAU Erlangen, Germany
harald.koestler@fau.de

Wednesday, May 29, 9:00 – 9:30

Computational fluid dynamic simulations are in general very compute intensive. Only by parallel simulations on modern architectures and on supercomputers the computational demands of complex simulation tasks can be satisfied. Facing these computational demands GPUs offer high performance, as they provide the high floating point performance and memory to processor chip bandwidth. To successfully utilize GPU clusters for the daily business of a large community, usable software frameworks have to be established on these clusters. The development of such software frameworks is only feasible with maintainable software designs that consider performance as a design objective right from the start. We present software design concepts for efficient and scalable multi-GPU parallelization approaches and introduce these concepts to our software framework waLBerla for multi-physics simulations centered around the lattice Boltzmann method. Our software designs support a pure-MPI and a hybrid parallelization approach capable of heterogeneous simulations using CPUs and GPUs in parallel. Weak and strong scaling performance results obtained on the Tsubame 2.0 cluster for more than 1000 GPUs are presented. With the help of a model the performance results are investigated and analyzed in a detailed and structured performance analysis.
We also outline our latest developments, where we extend the waLBerla to support grid refinement and discuss the influence on the framework structure.

1. J. Habich, C. Feichtinger, H. Köstler, G. Hager, G. Wellein, *Performance Engineering for the Lattice Boltzmann Method on GPGPUs: Architectural Requirements and Performance Results*, Computers & Fluids abs/1112.0850.

2. C. Feichtinger, J. Habich, H. Köstler, G. Hager, U. Rüde, G. Wellein, *A Flexible Patch-Based Lattice Boltzmann Parallelization Approach for Heterogeneous GPU–CPU Clusters*, Parallel Computing **37** (9) (2011) 536–549.

3. C. Feichtinger, S. Donath, H. Köstler, J. Götz, U. Rüde, *WaLBerla: HPC Software Design for Computational Engineering Simulations*, Journal of Computational Science **2** (2) (2011) 105–112.

Scaling Soft Matter Physics to Thousands of GPUs in Parallel

Alan Gray[1], Oliver Henrich[1], Kevin Stratford[1], Alistair Hart[2]
[1] EPCC, University of Edinburgh, UK
[2] Cray Exascale Research Initiative Europe, Edinburgh, UK
a.gray@epcc.ed.ac.uk

Wednesday, May 29, 9:35 – 10:05

Soft matter is all around us, from everyday consumer products and foodstuffs to components within our bodies. Such substances, which we broadly categorise as "complex fluids", exhibit behaviour dependent on the physics of the self-organising structures that form at the microscopic level. The versatile software package Ludwig uses the Lattice Boltzmann method (a popular approach particularly suitable for large-scale parallel implementations) as the basis for hydrodynamic coupling in a number of complex fluid areas: two-component and three-component fluids, colloidal suspensions, and liquid crystals are some examples.
We will describe current work to extend the code to a mixed GPU/MPI implementation. The high performance and large domain sizes reachable with massively-parallel GPU-accelerated supercomputers makes tractable some otherwise inaccessible problems. As well as discussing how to achieve maximal performance on each GPU, we will present a new halo-exchange communication phase for the code, developed to allow efficient parallel scaling to many GPUs in parallel. This includes the combination of CUDA stream functionality with MPI communications allowing the overlapping of separate stages within the communication phase to reduce the overall communication time. We will describe work to enable the advanced functionality required for systems of current research interest, in particular the inclusion of moving particles within liquid crystal simulations. We will present performance results on the Titan machine at Oak Ridge National Laboratory (currently the world's fastest supercomputer), showing that we achieve excellent scaling to at least 8 thousand GPUs in parallel.

A GPU cluster optimized multigrid scheme for computing unsteady incompressible fluid flow

Gyorgy Tegze
Wigner Research Centre for Physics, Budapest, Hungary
turpi@tegze.net

Wednesday, May 29, 10:10 – 10:25

A multigrid scheme has been proposed that allows efficient implementation on modern CPUs, many integrated core devices (MIC) and graphics processing units (GPU). It is shown, that wide single instruction multiple data (SIMD) processing engines are used effectively when a deep, 2h coarsened grid hierarchy is replaced with a two level multigrid scheme with 16h-32h coarsening. The coarse graining length can be fitted to the SIMD width to fully utilize modern CPUs and GPUs. Optimal memory transfer also can be ensured, since no strided memory access is required, and restriction step is executed on bigger chunks of data that features optimal caching properties. To counterbalance the inscrease of iteration cycles due to the large coarse-graining length, an interpolated stencil was developed, that minimizing spurious interference between the coarse and the fine scale solutions. The interpolated stencil recovers additive correction multigrid scheme, when used with 2h coarsening. The method is demonstrated on solving the pressure equation for 2D incompressible fluid flow. The benchmark setups cover shear driven laminar flow in a cavity, and direct numerical simulation (DNS) of a turbulent jet. We show, that the scheme also allows efficient use of distributed memory computer clusters via decreasing memory transfer needed between host and compute device and among cluster nodes. The actual implementation optimized for multi-core architectures using the OpenCl programming standard, and MPI standard for communicating in distributed memory system.

Determination of line tension in the 3d Ising model using GPUs

Benjamin Block, Suam Kim, Peter Virnau, Kurt Binder
University of Mainz, Germany
blockb@uni-mainz.de

Wednesday, May 29, 11:00 – 11:30

We have developed methods to simulate the Ising model in 2 and 3 dimensions efficiently on single and multiple GPUs [1]. We apply such an implementation to quantify the effect of line tension on heterogeneous nucleation barriers [2] in detail. In an Ising system with antiperiodic boundary conditions in one dimension, a liquid vapor interface can be stabilized between two walls. When wall fields are applied, this leads to a difference in the Free Energy of the system. This change can be quantified in dependence on the linear dimensions of the simulation box and by varying the size of the box in all dimensions, the contribution of line tension can be extracted.

1. B. Block, EPJ-ST **210**, 147 (2012).

2. D. Winter, P. Virnau, K. Binder, PRL **103**, 225703 (2009).

Ultrafast spin dynamics simulations using OpenCL

Stefan Gerlach
University of Konstanz, Germany
stefan.gerlach@uni-konstanz.de

Wednesday, May 29, 11:35 – 11:50

Ultrafast magnetization dynamics is a fascinating topic in the field of nanotechnology with a broad range of applications not only in the field of storage technologies. Just a few years ago it has been shown experimentally that it is possible to switch the magnetization in thin magnetic films within picoseconds using pure opto-magnetic effects[1]. The behavior of these magnetization processes often depends on several length and time scales which needs to be adressed with multiscale modeling.
Our current work on the understanding of ultrafast spin dynamics is based either on atomistic spin model simulations solving the stochastic Landau-Lifshitz-Gilbert (LLG) equations or on a continuum theory with the Landau-Lifshitz-Bloch (LLB) equation of motion [2] containing exchange, anisotropy and dipole-dipole interaction. With the help of the FFT method we are able to solve these equations numerically for realistic system sizes of up to 10^7 spins [3]. We will show how we use OpenCL to perform these simulations on GPUs and discuss how they compare with more traditional simulations written in C for multicore CPUs.

1. Stanciu, et al., Phys. Rev. Lett. **99**, 047601 (2007)

2. Atxitia, et al., Appl. Phys. Lett. **91**, 232507 (2007)

3. Vahaplar, et al., Phys. Rev. B **85**, 104402 (2012)

Fcc crystal-fluid interfacial free energy in Yukawa systems

Cristian Vasile Achim, A. Mijailovi, R. E. Rozas, J. Horbach, H. Löwen
Heinrich-Heine-University of Duesseldorf, Germany
cristian.v.achim@gmail.com,hlowen@thphy.uni-duesseldorf.de

Poster

We determine the orientation-resolved interfacial free energy between a face-centered-cubic (fcc) crystal and the coexisting fluid for a many-particle system interacting via a Yukawa pair potential. Several different screening strengths were investigated in the framework of Ramakrishnan-Yussouff density functional theory (DFT) of freezing [1]. While the DFT does not predict the correct phase diagram, the meta-stable fcc-fluid transition is in agreement with the Molecular Dynamics simulations. We present the dependence of the coexistence gap and interfacial energy as a function of the screening length.

Keywords: fcc-fluid interface, yukawa particles

1. T. V. Ramakrishnan and M. Yussouff, Phys. Rev. B **19** (1979) 2775.

HOOMD-blue: Massively parallel many-particle simulations on multiple GPUs

Jens Glaser[1], Joshua Anderson[2], Sharon Glotzer[2], David Morse[1]
[1] Dept. of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis
[2] Dept. of Chemical Engineering, University of Michigan, Ann Arbor
jglaser@umn.edu

Poster

We present work for the upcoming 1.0 release of the versatile and easy-to-use open-source molecular dynamics software HOOMD-blue, that enables one to run large-scale simulations on tens to hundreds of GPUs. Because of the high computational throughput of a GPU, it is difficult to prevent communication costs from becoming a bottleneck. By taking advantage of a refined version of Plimpton's communication scheme fully implemented on the GPU and of GPUdirect enabled MPI libraries, we demonstrate excellent strong scaling in simulations that have as few as 20,000 particles per GPU. We will present initial performance data measured on the Keeneland and Titan supercomputers.

Numerical Experiments with DDA C++ code DDscat.C++

Vasyl Choliy
Kyiv Shevchenko University, Ukraine
choliy.vasyl@googlemail.com

Poster

Electromagnetic energy is scattered or absorbed by targets. It is an isolated grain (of arbitrary geometry and possibly with complex refractivity index) or 1-d or 2-d periodic structure of unit cells. According to discrete dipole approximation (DDA), the target is approximated with the array of polarizable particles (dipoles). The theory of DDA was proposed by Purcell and Pennypacker [1] and wes developed by Draine and Flatau in [2], [3]. Extension of the theory to periodic structures was made in [4]. Calculations of the electric and magnetic field near the target was introduced in [5]. All mentioned algorithms were implemented in DDscat and explained for users in [6]. Current version of DDscat is 7.3.0 (the parent code). Parent code User guide [6] is a necessary and beautifull book to start using the code. Our code DDscat.C++ is the DDscat, rewritten in C++. Current version of DDscat.C++ is a clone of the parent code but it contains some C++ specific features to make it easily modifiable and portable. At the beginning the idea was to have a good software for the students to study the photonics and IT in the single package. Step-by-step the code have changed and now we have the code with another design and architecture but mostly with the same functionality. Our code User and programmer guide (http://code.google.com/p/ddscatcpp/) explains mostly differencies between the codes and concentrates upon programming features. The DDscat.C++ users are recommended to read both manuals. Introduction to the code and the experiment with chains of ellipsoids are presented.

1. Purcell E. M., Pennypacker C.R. 1973, Astrophys.J. **186**, 705
2. Draine B. T., Flatau P. J. 1988, Astrophys.J. **333**, 848
3. Draine B. T., Flatau P. J. 1994, J.Opt.Soc.Am.**11**, 1491
4. Draine B. T., Flatau P. J. 2008, J.Opt.Soc.Am.**25**, 2693
5. Flatau P. J., Draine B. T. 2012, Optics Express **20**, 1247
6. Draine B. T., Flatau P. J. 2013, arXiv:1202.3424

Parallel connected component labeling algorithm for 3D domains to detect inclusions during simulation time

Johannes Hötzer, Alexander Vondrous, Michael Selzer, Britta Nestler
Karlsruhe Institute of Technology, Germany
johannes.hoetzer@kit.edu,alexander.vondrous@kit.edu,michael.selzer@kit.edu,britta.nestler@kit.edu

Poster

Nowadays the availability of a huge number of processing units, e.g. CPUs or GPUs, connected to each other enables the distribution of the computational problem. With the phase-field method as a representative structured grid application,we show the benefit of three-dimensional domain decomposition over one-dimensional domain decomposition.
Over the last two decades, the phase-field method has evolved to a powerful and important method to describe a broad variety of phenomena, like surface driven processes or coupled physical phenomena, like diffusion processes, fluid flow, mechanical forces or magnetic fields. The challenge to simulate larger domains with more complex models requires an efficient use of high performance clusters, scaleable algorithms and parallel data structures.
We emphasize the efficiency of high performance computing resources by investigating the scaling behavior of one-dimensional domain decomposition and three-dimensional domain decomposition for a finite difference implementation. A performance model for blocking communication and measurements shows that it is necessary to apply n-dimensional domain decomposition for an n-dimensional domain, to efficiently scale on connected computing nodes. Three-dimensional domain decomposition enables the usage of many computing nodes to simulate large domains, which leads to new requirements regarding memory usage, simulation parameter distribution and the parallelization of algorithms. To address these issues, we show parameter caching mechanisms and how to take advantage of spare matrices to reduce memory consumption, computation and communication time.
The performance model shows a good scaling behavior of three dimensional domain decomposition. Measurements with up to 10 000 processing units confirm the expected estimations of the performance model.

Detailed balance and transition rates in BFM and GPU-accelerated BFM

Sorin Nedelcu
Leipniz-Institut Dresden, Germany
nedelcu@ipfdd.de

Poster

We compute the transition probabilities and the equilibrium distribution of different bond classes in classical bond fluctuation model BFM [1], accelerated method and collision method [2] and show that the condition of detailed balance is fulfilled in all cases. The differences in local details of monomer moves specific to each model lead to different transition rates between bond classes, which are equivalent ways of describing similar long time and length scales polymer melts. Additionally, we estimate how fast a Markov chain approaches equilibrium [3] in the classical algorithm and in the GPU-accelerated methods, and show that when the classical Monte Carlo moves are implemented on the GPU (without acceleration) the local detailed balance is also fulfilled.

1. H.-P. Deutsch, K. Binder, J. Chem. Phys. **94**, (1991) 2294

2. S. Nedelcu, M. Werner, M. Lang, J.-U. Sommer, J. Comp. Phys. **231**, (2012) 2811-2824

3. C. Moore, S. Mertens, The nature of computation, Oxford, University, Press, **2011**, Ch. 12

GPU Implementation of an Electrokinetics Solver as proposed by Capuani, Pagonabarraga and Frenkel

Georg Rempfer, Axel Arnold, Christiam Holm
University of Stuttgart, ICP, Germany
georg.rempfer@icp.uni-stuttgart.de

Poster

In 2004 Capuani, Pagonabarraga and Frenkel proposed an algorithm to solve the full set of electrokinetic equations [1]. It mainly consists if a lattice-Boltzmann algorithm coupled to a solver for the convection-diffusion equation. They demonstrated that electro-osmotic flow and the sedimentation of charged spheres can be treated with this method. We implement this method to be executed by GPUs as part of ESPResSo [2], the Extensible Simulation Package for Research on Soft Matter Systems.

1. Fabrizio Capuani, Ignacio Pagonabarraga and Daan Frenkel: “Discrete solution of the electrokinetic equations", J. Chem. Phys., **121(2)**, 973-986 (2004).

2. Hans-Jörg Limbach, Axel Arnold, Bernward A. Mann and Christian Holm: “ESPResSo – An Extensible Simulation Package for Research on Soft Matter Systems", Comput. Phys. Commun. **174(9)**, 704-727 (2006)

ESPResSo++: simulation package for condensed soft matter research

Vitaliy Starchenko
University of Coventry, UK
tarasyk3@gmail.com

Poster

ESPResSo++ (the redesigned Extensible Simulation Package for Research on Soft matter systems) is an open-source, flexible, parallelized simulation package for many particle simulations of condensed soft matter systems. The package provides molecular dynamics and Monte Carlo simulations. Beside the standard techniques and methods found in well-established packages, ESPResSo++ includes the Adaptive Resolution Scheme (AdResS) for multiscale simulations of molecular systems where the level of resolution of each molecule can be changed on-the-fly. Currently there are active works on implementation of H-AdResS (extension to AdResS method, which is formulated in terms of a general Hamiltonian for the whole system) and on implementation of Lattice Boltzmann method which will allow to couple standard molecular dynamics and hydrodynamics. Flexibility of ESPResSo++ is based on the integration of Python and C++. Python is used for user interface. It makes easy to use the power of scripting language as well as other python extensions. C++ kernel is highly modular and provides fast calculations. The flexibility of ESPResSo++ allows the study a wide range of systems (typical examples: melts, solutions, brushes, copolymers, blends, liquid crystals, colloids, charged fluids etc). ESPResSo++ can be used by scientists as a research platform for development and testing of their own methods as well.

Spectral methods in PDE solving: a multi-GPU framework

Gyula Toth, G. Tegze
Wigner Research Centre for Physics, Budapest, Hungary
gytoth.risspo@gmail.com

Poster

Studying pattern formation in non-equilibrium systems usually leads to the numerical solution of highly nonlinear higher order partial differential equations. The operator-splitting based pseudo-spectral semi-implicit methods often yield near unconditionally stable, consistent time-step schemes, allowing us to avoid the stability criterium of finite difference schemes. From the computational point of view, moreover, a parallel spectral scheme necessitates only parallel FFT, all other operations can be executed "in place" (i.e. without the need of communication). The PDE solver framework presented here works on both single and multiple GPUs. We discuss the pros and cons of the method on the results obtained for various pattern formation systems and demonstrate the efficiency and limitations of the multi-GPU version.

GPU accelerated spectral analysis of the susceptible-infected susceptible model on Barabási-Albert networks

Domokos Zoltan
Eotvos University Budapest, Hungary

Poster

The susceptible-infected-susceptible (SIS) model is the simplest model for information and epidemic spreading. Our work focuses on the spectral analysis of the SIS process in recursive trees and the opportunities to exploit our NVIDIA graphics processing units' parallel processing capabilities in network growth modeling. We show that although the Barabási-Albert tree (BAT) algorithm in general misfits the SIMT architecture of the GPU, some important calculations can be implemented on it, achieving a reasonable speed-up compared to fully host code. We solve the eigenvalue problem by Lanczos algorithm, calculate multiplicative weights for the graph's edges and generate random numbers on the GPU, while simulating the attachment of new nodes to the network remains to be a task of the CPU. Results for a wide range of network sizes will also be presented, in order to demonstrate the accuracy of our program.