



MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG



COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Workshop

on

Power-Aware COmputing (PACO 2015)

July 06-07, 2015

Max Planck Institute for Dynamics of Complex Technical Systems
Magdeburg

SPONSORED BY THE



Federal Ministry
of Education
and Research



Contents

Program	1
Collection of Abstracts	5
Invited Talks	7
Contributed Talks	13
List of Participants	33
Additional Information	35

Program

Monday, July 06

- 14:15 - 14:30 Opening
- 14:30 - 15:30 **Georg Hager** p.8
White-box modeling for performance and energy: useful patterns for resource optimization
- 15:30 - 16:00 Coffee Break
- 16:00 - 16:30 **Ernesto Dufrechou** p.14
Efficient and power-aware band linear systems solver in hybrid CPU-GPU platforms
- 16:30 - 17:00 **José I. Aliaga** p.16
Adapting Concurrency Throttling and Voltage-Frequency Scaling for Dense Eigensolvers
- 17:00 - 17:30 **René Milk** p.19
Efficiently Scalable Multiscale Methods using DUNE
- 17:30 - 18:00 **Christian Himpe** p.20
Zero-Copy Parallelized Empirical Gramians
- 18:00 - 18:30 **Sponsor Talk: Thomas Blum (MEGWARE)** p.21
Enhanced Power Monitoring with Megware SlideSX
- 20:00 Conference Dinner: Los Amigos
-

Tuesday, July 07

- 09:30 - 10:30 **Markus Geveler** p.10
Realization of a low energy HPC platform powered by renewables as part of a student project - A case study: technical, numerical and implementation aspects
- 10:30 - 11:00 Coffee Break
- 11:00 - 11:30 **Alfredo Remón** p.22
Trading Off Performance for Energy in Sign Function Resolution
- 11:30 - 12:00 **Martin Köhler** p.24
Effects of dynamic frequency scaling of Nvidia GPUs during the computation of the generalized matrix sign function
- 12:00 - 12:30 **Kai Diethelm** p.26
Using the Periscope Tuning Framework for Energy Optimization
- 12:30 - 13:00 **Maria Barreda** p.29
An Integrated Framework for Power-Performance Analysis of Parallel Scientific Workloads
- 13:00 - 14:30 Lunch Break
- 14:30 - 15:30 **Enrique S. Quintana-Ortí** p.11
Saving energy in sparse and dense linear algebra operations
- 15:30 - 16:00 Closing and Farewell Coffee
-

Collection of Abstracts

Invited Talks

White-box modeling for performance and energy: useful patterns for resource optimization

Georg Hager¹

Ayesha Afzal²

A realistic upper limit for the performance of a code on a particular computer hardware may be called its *light speed*. Light speed allows a well-defined answer to the question whether an implementation of an algorithm is “good enough.” A model leading to an accurate light speed estimate requires thorough code analysis, knowledge of computer architecture, and experience on how software interacts with hardware. The notion of light speed depends very much on the machine model underlying the hardware model; if the machine model misses an important performance-limiting detail, one might arrive at the (false) conclusion that light speed is not attained by the code at hand, while it actually is. Which hardware features should be included to arrive at a good balance between simplicity and predictive power is a crucial question, and this talk tries to give useful answers to it. Two pivotal concepts are the cornerstones of the modeling process: *bottlenecks* and *performance patterns*. A bottleneck is a hardware feature that limits the performance of a program. A performance pattern is a performance-limiting motif in which one or more bottlenecks (or a complete lack thereof) may be present. Identifying a performance pattern via observable behavior is the first step towards building a useful performance model.

In complex cases it may not be possible to establish a model at all. If a model can be built, one can gain a deeper understanding of the interactions between software and hardware. If the model works, i.e., if its predictions can be validated by measurements, this is an indication that it describes certain aspects of this interaction accurately. If the model does not work, it must be refined, leading to more insights. A working model can help with predicting the possible gain of code optimizations. Changing the program code may require adjustments in the model, or even building a completely new model when the underlying algorithm was changed.

When quantitative insight into the performance aspects of an implementation has been gained, one can proceed to include energy aspects in the modeling process. To lowest order, the energy used for performing some computation is proportional to the wall-clock time required. Starting from this assertion, together with some simplifying assumptions about scalability behavior and the dependence of power dissipation on clock speed and the number of cores used, one can construct a simple chip-level power/performance model that yields surprisingly deep insights into the energy aspects of computation. The talk presents examples that reveal the interplay between clock speed dependence and scaling behavior, and gives hints as to how one may exploit the full potential for saving energy with minimal concessions regarding performance.

¹Erlangen Regional Computing Center (RRZE), University of Erlangen-Nuremberg, Martenstr. 1, 91058 Erlangen, Germany, georg.hager@fau.de

²Erlangen Regional Computing Center (RRZE), University of Erlangen-Nuremberg, Martenstr. 1, 91058 Erlangen, Germany, ayesha.afzal@fau.de

References

- [1] G. Hager, J. Treibig, J. Habich and G. Wellein. *Exploring performance and power properties of modern multicore chips via simple machine models*. Concurrency Computat.: Pract. Exper. (2013), <http://dx.doi.org/10.1002/cpe.3180>.
 - [2] M. Wittmann, G. Hager, T. Zeiser, J. Treibig, and G. Wellein. *Chip-level and multi-node analysis of energy-optimized lattice Boltzmann CFD simulations*. Concurrency Computat.: Pract. Exper. (2015), <http://dx.doi.org/10.1002/cpe.3489>.
-

Realization of a low energy HPC platform powered by renewables - A case study: technical, numerical and implementation aspects

Markus Geveler¹

Stefan Turek²

We present our approach of integrating low power, unconventional computational hardware and renewables into a modern, green, photovoltaic-driven HPC-facility alongside with specially tailored high-end numerics and CFD simulation software. In this talk, we concentrate on performance engineering for hardware-, numerical- and energy efficiency targeting next generation mobile NVIDIA Tegra K1 processors, which integrate high-end ARM cores with a programmable Kepler GPU on one low-power SoC. In addition, we cover other aspects of the system design like energy supply and climate control.

¹Institut für Angewandte Mathematik (LS3), TU Dortmund, Vogelpothsweg 87, 44227 Dortmund, Germany,
markus.geveler@math.tu-dortmund.de

²Institut für Angewandte Mathematik (LS3), TU Dortmund, Vogelpothsweg 87, 44227 Dortmund, Germany,
ture@featflow.de

Saving Energy in Sparse Linear Algebra Operations

Enrique S. Quintana-Ort¹

Summary

Recent breakthroughs in scientific research heavily rely on complex simulations carried out in large-scale supercomputers. A concern in this scenario is the power draft and energy spent in these facilities, which is rapidly becoming a major constraint.

In this work, we provide an overview of energy-aware scientific computing, for the particular domain of sparse linear algebra, by analyzing the energy efficiency of a broad collection of hardware architectures, and exposing algorithmic and implementation changes that yield energy savings for sparse linear system solvers while maintaining their performance.

Introduction

The High Performance Conjugate Gradients (HPCG) is a sparse linear solver that has been recently proposed as a benchmark with the specific purpose of exercising computational units and producing data access patterns that mimic those present in an ample set of important HPC applications. This alternative to the reference LINPACK benchmark is crucial because such metrics may guide computer architecture designers, e.g. from AMD, ARM, IBM, Intel and NVIDIA, to invest in future hardware features and components with a real impact on the performance and energy efficiency of these applications.

This work investigates the following questions around the CG method:

- Characterizing the power/energy efficiency of CG on state-of-the-art architectures [1]. We will study the runtime and energy efficiency for both out-of-the-box codes, relying exclusively on compiler optimizations, as well as implementations manually optimized, for a variety of architectures that range from general-purpose and digital signal multicore processors to manycore graphics processing units (GPUs), representative of current multithreaded systems.

- Evaluating and modeling the effect of complex preconditioners on power/energy consumption [2].

We will investigate the benefits that an energy-aware implementation of the *runtime* in charge of the concurrent execution of ILUPACK —a sophisticated preconditioned iterative solver for sparse linear systems— produces on the time-power-energy balance of the application. Furthermore, we will propose several simple yet accurate power models that capture the power variations that result from the introduction of the energy-aware strategies, connecting this effect with the processor C-states, as well as the impact of the P-states into ILUPACK's runtime.

¹Dpto. de Ingeniería y Ciencia de Computadores, Universidad Jaume I, 12.071–Castellón, Spain, quintana@uji.es

- Energy saving techniques for hybrid CPU-GPU platforms [3].
We will introduce a systematic methodology to derive fused versions some of the most popular iterative solvers (with and without preconditioning) for sparse linear systems. These versions attain remarkable energy savings when executed in blocking mode and, in general, they match the performance of an execution of the same versions when executed in the performance-active but power-hungrier polling mode.

References

- [1] José I. Aliaga, Hartwig Anzt, Maribel Castillo, Juan C. Fernández, Germán León, Joaquín Pérez, and Enrique S. Quintana-Ortí. Unveiling the performance-energy trade-off in iterative linear system solvers for multithreaded processors. *Concurrency and Computation: Practice and Experience*, 27(4):885–904, 2015.
 - [2] José I. Aliaga, Maria Barreda, Manuel F. Dolz, Alberto F. Martín, Rafael Mayo, and Enrique S. Quintana-Ortí. Assessing the impact of the CPU power-saving modes on the task-parallel solution of sparse linear systems. *Cluster Computing*, 17(4):1335–1348, 2014.
 - [3] José I. Aliaga, Joaquín Pérez, and Enrique S. Quintana-Ortí. Systematic fusion of CUDA kernels for iterative sparse linear system solvers. In *Proceedings of the 21th International Euro-Par Conference*, Lecture Notes in Computer Science. Springer, 2015. To appear.
-

Contributed Talks

Efficient and power-aware band linear systems solver in hybrid CPU-GPU platforms

Ernesto Dufrechou¹

Pablo Ezzatti²
Alfredo Remón⁴

Enrique S. Quintana-Ortí³

Linear systems with band coefficient matrix appear in a large variety of applications [5], including finite element analysis in structural mechanics, domain decomposition methods for partial differential equations in civil engineering, and as part of matrix equations solvers in control and systems theory. Exploiting the structure of the matrix in these problems yields huge savings, both in number of computations and storage space. For this reason the LAPACK library [2, 4] includes a band system solver, which is an efficient means to solve this type of systems on multicore platforms, provided a (multi-threaded) implementation of BLAS is available.

In the last decade, hybrid computer platforms consisting of multicore processors and GPUs (graphics processing units) have evolved to become common in many application areas with high computational requirements, but also in end-user workstations and relatively low cost servers. This wide use is motivated by their high throughput, their low cost and their remarkably low flops/watt ratio. Thus, GPUs conform a number of platforms in the TOP 500 list [6], but also in the Green 500 list [1].

In this work we perform a multi-criteria analysis of a set of hybrid CPU-GPU routines presented in a previous work [3], to accelerate the solution of band linear systems. These routines leverage the large-scale parallelism available in hybrid CPU-GPU platforms by offloading the computationally expensive operations to the GPU.

Our study address (computational) performance and energy efficiency, measuring both execution time and energy consumption on a platform equipped with an Intel Core i7-4770 processor (Haswell) and a Nvidia K40 GPU, using the built-in power meters of both devices to measure the energy consumed during the experiments.

Table 0.1 shows the results of the solution of five band linear systems with dimensions n between 25,600 and 76,800 and a bandwidth of 1%, 2% and 4% of the problem size, using our hybrid implementation and the band solver included in the Intel MKL library. For the smaller instances, the MKL version yields better runtimes than the hybrid one, although the latter remains competitive. For $n > 38,400$ and a bandwidth of 2% the hybrid version outperforms the MKL-based version with regard to execution time, reaching an improvement of $8\times$ for the largest case. Regarding the energy requirements of both versions, the hybrid implementations becomes convenient with $n > 64,000$ for a

¹Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay, 11300, edufrechou@fing.edu.uy

²Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay, 11300, pezzatti@fing.edu.uy

³Departamento de Ingeniería y Ciencia de Computadores, Universidad Jaume I, Castellón, Spain, 12071, quintana@icc.uji.es

⁴Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany, 39106, remon@mpi-magdeburg.mpg.de

Matrix Dimension	Bandwidth $k_b = k_u = k_l$	MKL			hybrid	
		time	E_{cpu}	$E_{cpu+gpu}$	time	$E_{cpu+gpu}$
25,600	1%	0.260	12.5	29.7	0.814	96.3
	2%	0.511	29.8	63.6	1.025	127.7
	4%	2.072	130.1	261.7	1.397	184.4
38,400	1%	0.669	33.9	78.1	1.404	164.6
	2%	2.216	126.4	272.7	1.792	228.6
	4%	7.347	452.3	920.3	2.680	366.7
51,200	1%	1.421	77.0	171.0	2.061	244.0
	2%	4.361	283.6	571.6	2.771	352.2
	4%	16.132	1046.5	2076.9	4.541	651.2
64,000	1%	2.223	128.5	275.4	2.807	336.3
	2%	8.751	558.2	1135.9	3.968	523.9
	4%	32.021	2032.8	4088.8	7.028	1058.8
76,800	1%	3.448	225.9	453.6	3.618	440.5
	2%	15.477	937.4	1926.4	5.393	730.6
	4%	85.106	3999.5	9523.4	10.431	1635.9

Table 0.1: Execution time (in seconds) and energy consumption (in joules) of the band linear system solvers.

bandwidth of 2% and with $n > 38,400$ for a bandwidth of 4%. In general, the hybrid version has a better energy performance (flops/watt), so it becomes more convenient as the problem size increases. Additionally, if the power spent by the GPU when it stays idle is taken into account for the MKL based version, the benefits of using the hybrid routines when executing on a CPU-GPU platform become evident.

References

- [1] The Green500 list, 2015. Available at <http://www.green500.org>.
- [2] E. Anderson, Z. Bai, J. Demmel, J. E. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. E. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, 1992.
- [3] Peter Benner, Ernesto Dufrechou, Pablo Ezzatti, Pablo Igounet, Enrique S. Quintana-Ortí, and Alfredo Remón. Accelerating band linear algebra operations on gpus with application in model reduction. In *Computational Science and Its Applications ICCSA 2014*, volume 8584 of *Lecture Notes in Computer Science*, pages 386–400. Springer International Publishing, 2014.
- [4] Jeremy Du Croz, Peter Mayes, and Giuseppe Radicati. Factorization of band matrices using level 3 BLAS. LAPACK Working Note 21, Technical Report CS-90-109, University of Tennessee, July 1990.
- [5] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 3rd edition, 1996.
- [6] The Top500 list. Available at <http://www.top500.org>, 2015.

Adapting Concurrency Throttling and Voltage-Frequency Scaling for Dense Eigensolvers

José I. Aliaga¹

María Barreda¹, M. Asunción Castaño¹, Enrique S. Quintana-Ortí¹

Manuel F. Dolz²

Introduction

The crucial role that dense linear algebra (DLA) operations play in many scientific and engineering applications has motivated, over the past decades, the development of highly tuned implementations of BLAS (*Basic Linear Algebra Subprograms*) [1] and LAPACK (*Linear Algebra PACKage*) [2] as, e.g., those included in Intel's MKL, AMD's ACML and IBM's ESSL. Unfortunately, in an era where power has become the key factor that constrains both the design and performance of current computer architectures, the kernels and routines in these libraries are largely optimized for raw performance, either being completely oblivious of the energy they consume or operating under the assumption that tuning for performance is equivalent to optimizing energy.

Two crucial factors that control power dissipation and, in consequence, energy consumption of a multithreaded application, running on a multicore processor, are the level of thread parallelism (concurrency throttling) and the core voltage-frequency setting. In this work, we show how to attain an *actual energy-efficient execution* of a key computational routine to tackle eigenproblems in LAPACK, namely the symmetric reduction to tridiagonal form (`dsytrd`). To obtain the *energy savings*, we first analyze the performance and energy efficiency of the two building blocks that govern the performance of `dsytrd`, using the dynamic concurrency throttling (DCT) and the dynamic voltage-frequency scaling (DVFS). Next, we employ the best option of DCT and DVFS for each block to tune the execution of the `dsytrd`.

Energy Savings for Eigenvalue Problems

BLAS is organized into three groups or *levels*, known as BLAS-1, BLAS-2 and BLAS-3, with the kernels in the latter two respectively conducting quadratic and cubic numbers of flops (floating-point arithmetic operations) on a quadratic amount of elements. On current cache-based architectures, tuned implementations of BLAS-3 generally deliver a high GFLOPS (billions of flops/sec.) rate, close to the processor's peak, as they present a ratio of flops to memory operations that grows linearly with the problem size. On the other hand, the kernels in BLAS-2 cannot hide today's high memory latency, due to their

¹Depto. de Ingeniería y Ciencia de Computadores, Universidad Jaime I, 12071-Castellon, Spain, aliaga@uji.es, mvaya@uji.es, castano@uji.es, quintana@uji.es

²Dept. of Informatics, University of Hamburg, 22.527-Hamburg, Germany, manuel.dolz@informatik.uni-hamburg.de

low number of flops per memory access, and in consequence often deliver a low energy efficiency in terms of GFLOPS per watt (GFLOPS/W).

Given a symmetric matrix $A \in \mathbb{R}^{n \times n}$, the standard routine in LAPACK for the reduction to tridiagonal form, `dsytrd`, yields the decomposition $A = Q^T T Q$, where $Q \in \mathbb{R}^{n \times n}$ is orthogonal and $T \in \mathbb{R}^{n \times n}$ is tridiagonal. In case only T is explicitly built, this routine requires $4n^3/3$ flops, roughly performing half of its flops in terms of BLAS-2 (mainly, via kernel `dsymv` for the symmetric matrix-vector product), while the other half is cast as BLAS-3 operations (concretely, via kernel `dsyr2k` for the symmetric rank- $2k$ update). The routine consists of a main loop that processes the input matrix A by blocks (panels) of b columns/rows per iteration. As the factorization proceeds, the symmetric rank- $2k$ updates that are performed in the loop body decrease in the number of columns/rows, from $n - b$ towards 1 in steps of $k = b$ (algorithmic block size) per iteration. For `dsymv`, the progression is from $n - 1$ to 1 in unit steps. In principle, the two BLAS kernels involved in the routine `dsytrd` feature different properties. The first one is characterized as a strongly memory-bound operation, and the second one is in principle a CPU-bound operation, but it becomes memory-bound when the block size is small.

In the experimentation we will illustrate the performance and energy efficiency of the building blocks `dsymv` and `dsyr2k`, under different configurations of DVFS-DCT, taking into account also whether or not the problem data fits into the processor's last level cache. These experiments indicate a clear path to adjust the number of threads and voltage-frequency setting in order to optimize either performance or energy efficiency, depending on the problem size.

Moreover, to obtain an energy-aware execution of the `dsytrd` operation, we also analyze the cost of varying the CPU performance state (P-state), which determines the processor voltage and frequency, and the cost of changing the number of cores. In order to compute the average cost of the variation between any two P-states we use the `cpufrequtils` [3] and `FTaLaT` [4] packages. If some of these costs are high, we should minimize the changes, as otherwise the negative impact on the performance will be large.

Analyzing the results obtained in the previous experimentation, we define and assess three energy-aware strategies. In the first two strategies we execute both kernels (`dsymv` and `dsyr2k`) at the same frequency, choosing a compromise value, and tune only the number of cores/threads. For the frequency, we only distinguish whether or not the problem size fits into the LLC. The difference between both strategies is that we use `cpufrequtils` and `FTaLaT`, respectively, to perform the frequency changes. In the third strategy we adjust the frequency and the number of cores for each kernel and subproblem size. We change the voltage-frequency by modifying the files related to frequencies management using `FTaLaT`, with a cost that is negligible.

References

- [1] Jack J. Dongarra, Jeremy Du Croz, Sven Hammarling and Iain Duff. A Set of Level 3 Basic Linear Algebra Subprograms. *ACM Transactions on Mathematical Software*, Vol.16(1), pages 1-17, 1990.
- [2] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, Jack J. Dongarra, J. Du Croz, S. Hammarling, A. Greenbaum, A. McKenney and D. Sorensen. *LAPACK Users' guide*. SIAM, 3rd edition, 1999.

- [3] How to use cpufrequtils. http://www.thinkwiki.org/wiki/How_to_use_cpufrequtils
 - [4] Abdelhafid Mazouz, Alexandre Laurent, Benoit Pradelle and William Jalby. Evaluation of CPU frequency transition latency. *Computer Science - Research and Development*, Vol.29(3-4), pages 187-195, 2014.
-

Efficiently Scalable Multiscale Methods using DUNE

René Milk¹ Mario Ohlberger²

The modeling and simulation of complex fluid flows, for example in reservoir engineering, give rise to a problem class that is inherently multiscale and requires the solution of demanding partial differential equations (PDEs). In this contribution we introduce a mathematical abstraction of multiscale methods [4], which are able to deal with the difficulties stemming from the numerical approximation of solutions to these PDEs. Based on this unified mathematical abstraction layer, we present a parallelization approach that reflects the different layers of these multiscale methods. We detail our implementation built using the Distributed and Unified Numerics Environment DUNE [1] and the DUNE Generic Discretization Toolbox [3]. As members of the research consortium EXA-DUNE [2], part of the DFG Priority Programme SPP 1648-1 "Software for Exascale Computing", we are concerned with working towards utilizing current and future, potentially highly heterogeneous, peta- and exa-scale computing clusters. We will discuss our findings on pure-MPI versus hybrid MPI/shared memory parallelization strategies regarding efficiency, scalability, time to solution, and power consumption.

References

- [1] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöforn, Ralf Kornhuber, Mario Ohlberger, and Oliver Sander. A generic grid interface for parallel and adaptive scientific computing. Part II: Implementation and Tests in DUNE. *Computing*, 82(2-3):121–138, June 2008.
- [2] P. Bastian, C. Engwer, D. Göddeke, O. Iliev, O. Ippisch, M. Ohlberger, S. Turek, J. Fahlke, S. Kaulmann, S. Müthing, and D. Ribbrock. EXA-DUNE: Flexible PDE Solvers, Numerical Methods and Applications . To appear in: *Proceedings EuroPar 2014, Workshop on Software for Exascale Computing* . Springer, August 2014, Porto, Portugal .
- [3] Felix Schindler, René Milk. DUNE-Gdt. <https://github.com/pymor/dune-gdt/>
- [4] Mario Ohlberger. Error control based model reduction for multiscale problems. In Angela Handlovičová, Zuzana Minarechová, and Daniel Ševčovič, editors, *Algoritmy 2012*, pages 1–10. Slovak University of Technology in Bratislava, April 2012.

¹Institute for Numerical and Applied Mathematics, University of Münster, Einsteinstrasse 62, 48149 Münster,
rene.milk@wwu.de

²Institute for Numerical and Applied Mathematics, University of Münster, Einsteinstrasse 62, 48149 Münster,
mario.ohlberger@wwu.de

Zero-Copy Parallelized Empirical Gramians

Christian Himpe¹ Mario Ohlberger²

Empirical gramian matrices [1, 2] encode information of control systems and can be used for tasks such as model reduction or system identification. The assembly of empirical gramians requires two steps. First, the generation of trajectories using in example Runge-Kutta methods, and second, the assembly of the gramian matrix. While the Runge-Kutta methods are classically a sequential process, the matrix multiplication during the gramian assembly parallelizes very well. Hence, on a heterogeneous CPU / GPU system the workload can be shared accordingly. Yet, since the gramian matrices are assembled from the previously computed trajectories, possibly large amounts of data have to be transferred between the system memory and the accelerators' separated memory. The current generation, but especially the next generation, of accelerated processing units (APU) with integrated CPU and GPU cores, can use the same memory space without copying between pre-set memory regions. In the context of the empirical gramian framework [4], the concept and possibilities of heterogenous uniform memory access (hUMA) [3] for economical APU devices are described.

References

- [1] S. Lall, J.E. Marsden and S. Glavaski. Empirical model reduction of controlled nonlinear systems. *Proceedings of the 14th IFAC Congress*, F:473–478, 1999.
- [2] C. Himpe and M. Ohlberger. Cross-Gramian Based Combined State and Parameter Reduction for Large-Scale Control Systems. *Mathematical Problems in Engineering*, 2014:1–13, 2014.
- [3] P. Rogers, J. Macri and S. Marikovic. AMD heterogeneous Uniform Memory Access. 2013.
- [4] C. Himpe. emgr - Empirical Gramian Framework. <http://gramian.de>, 2015.

¹Institute for Computational and Applied Mathematics, University of Münster, Einsteinstrasse 62, 48149 Münster,
christian.himpe@uni-muenster.de

²Institute for Computational and Applied Mathematics, University of Münster, Einsteinstrasse 62, 48149 Münster,
mario.ohlberger@uni-muenster.de

Enhanced Power Monitoring with Megware SlideSX

Thomas Blum¹

During the past years the measurement of the power consumption on the AC side within server and HPC environments has been getting an important factor to foster energy efficient systems in general. With the MEGWARE SlideSX® computing platform we introduce a way to measure the power on the DC side for every single compute node with a very fine grained resolution. This talk will introduce the computing platform including the measurement and monitoring facilities that comes built into the system and gives an overview about how this information can be accessed and further used and which functionalities are under development.

¹MEGWARE Computer GmbH
Vertrieb und Service
Nordstraße 19
09247 Chemnitz-Röhrsdorf, Germany,
thomas.blum@megware.com

Trading Off Performance for Energy in Sign Function Resolution

Alfredo Remón¹

Peter Benner²

Juan P. Silva³

Pablo Ezzatti⁴

Enrique S. Quintana-Ortí⁵

Consider a matrix $A \in \mathbb{R}^{n \times n}$ with no eigenvalues on the imaginary axis, and let

$$A = T^{-1} \begin{pmatrix} J_- & 0 \\ 0 & J_+ \end{pmatrix} T, \quad (0.1)$$

be its Jordan decomposition, where the eigenvalues of $J_- \in \mathbb{R}^{j \times j} / J_+ \in \mathbb{R}^{(n-j) \times (n-j)}$ all have negative/positive real parts [5]. The *matrix sign function* of A is then defined as

$$\text{sign}(A) = T^{-1} \begin{pmatrix} -I_j & 0 \\ 0 & I_{n-j} \end{pmatrix} T, \quad (0.2)$$

where I denotes the identity matrix of the order indicated by the subscript. The matrix sign function is a useful numerical tool for the solution of control theory problems (model reduction, optimal control) [6], the bottleneck computation in many lattice quantum chromodynamics computations and dense linear algebra computations (block diagonalization, eigenspectrum separation) [5]. Large-scale problems as those arising, e.g., in control theory often involve matrices of dimension $n \rightarrow O(10,000 - 100,000)$.

There are simple iterative schemes for the computation of the sign function. Among these, the Newton iteration, given by

$$\begin{aligned} A_0 &:= A, \\ A_{k+1} &:= \frac{1}{2}(A_k + A_k^{-1}), \quad k = 0, 1, 2, \dots, \end{aligned} \quad (0.3)$$

is specially appealing for its simplicity, efficiency, parallel performance, and asymptotic quadratic convergence. However, even if A is sparse, $\{A_k\}_{k=1,2,\dots}$ in general are full dense matrices and, thus, the scheme in (0.3) roughly requires $2n^3$ floating-point arithmetic operations (flops) per iteration.

General-purpose multicore architectures and graphics processor units (GPUs) dominate today's landscape of high performance computing (HPC), offering unprecedented levels of raw performance when aggregated to build the systems of the Top500 list [2].

¹Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany, 39106, remon@mpi-magdeburg.mpg.de

²Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany, 39106, benner@mpi-magdeburg.mpg.de

³Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay, 11300, jpsilva@fing.edu.uy

⁴Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay, 11300, pezzatti@fing.edu.uy

⁵Departamento de Ingeniería y Ciencia de Computadores, Universidad Jaume I, Castellón, Spain, 12071, quintana@icc.uji.es

While the performance-power trade-off of HPC platforms has also enjoyed considerable advances in the past few years [1]—mostly due to the deployment of heterogeneous platforms equipped with hardware accelerators (e.g., NVIDIA and AMD graphics processors, Intel Xeon Phi) or the adoption of low-power multicore processors (IBM PowerPC A2, ARM chips, etc.)—much remains to be done from the perspective of energy efficiency. In particular, power consumption has been identified as a key challenge that will have to be confronted to render Exascale systems feasible by 2020 [3, 4]. Even if the current progress pace of the performance-power ratio can be maintained (a factor of about $5\times$ in the last 5 years [1]), the ambitious goal of yielding a sustained ExaFLOPS (i.e., 10^{18} floating-point arithmetic operations, or flops, per second) for 20–40 MWatts by the end of this decade will be clearly exceeded.

In recent years, a number of HPC prototypes have proposed the use of low-power technology, initially designed for mobile appliances like smart phones and tablets, to deliver high MFLOPS/Watt rates. Following this trend, in this paper we investigate the performance, power and energy consumption of two low-power architectures, concretely an Intel Atom and a hybrid system composed of a multicore ARM processor and an NVIDIA GPU, and a general-purpose multicore processor to address the sign resolution.

The routines for the Xeon and Atom processors heavily rely on the matrix-matrix product kernel in Intel MKL. The hybrid implementation platform makes intensive use of the kernels in CUBLAS and the legacy implementation of BLAS parallelized with OpenMP.

We evaluate the performance and power-energy consumption to solve the matrix sign function using the three target platforms, i.e. Xeon, Atom and GPU+ARM. Concretely, the runtime to obtain the matrix sign function for four different problem dimensions, 256, 2,048, 5,120 and 8,192.

The experimental evaluation shows that the highest energy consumption is required by Atom. Despite its low average power consumption, the large computational time leads to the worst results in terms of energy for this platform. Thus, the energy consumed by the Xeon is $4\times$ lower for the largest problem tackled. On the other hand the lowest energy consumption is obtained for ARM+GPU platform, which requires $2\times$ and $8\times$ less energy than Xeon and Atom respectively. This is explained by the favorable performance-power ratio of the ARM+GPU platform.

References

- [1] The Green500 list, 2015. Available at <http://www.green500.org>.
- [2] The top500 list, 2015. Available at <http://www.top500.org>.
- [3] Steve Ashby *et al.* The opportunities and challenges of Exascale computing. Summary Report of the Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee, November 2010.
- [4] J. Dongarra and *et al.* The international ExaScale software project roadmap. *Int. J. of High Performance Computing & Applications*, 25(1):3–60, 2011.
- [5] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 3rd edition, 1996.
- [6] P.H. Petkov, N.D. Christov, and M.M. Konstantinov. *Computational Methods for Linear Control Systems*. Hertfordshire, UK, 1991.

Effects of dynamic frequency scaling of Nvidia GPUs during the computation of the generalized matrix sign function

Martin Köhler¹ Jens Saak²

The generalized matrix sign function $\text{sign}(A, E)$ is an extension of the common sign function $\text{sign}(A)$ for a single matrix to the case of matrix pencils (A, E) , where A and E are real $n \times n$ matrices. It is used to compute invariant subspaces [1], or approximate generalized eigenvalue problems [2] and to solve matrix equations like generalized Riccati [4] and Lyapunov equations [3]. The computation is commonly performed using a Newton iteration

$$\begin{aligned} A_0 &:= A \\ A_{i+1} &:= \frac{1}{2c_k} (A_i + c_k^2 E A_i^{-1} E) \quad \forall i = 1, \dots, \end{aligned} \quad (0.4)$$

where c_k is a scaling factor intended to accelerate the convergence. Because it is a globally convergent Newton scheme we have $A_i \rightarrow \text{sign}(A, E)$ for $i \rightarrow \infty$. It is easy to see that the iteration only consists of the solution of a linear system followed by a general matrix-matrix product (GEMM) call. Both operations are well suited for accelerator devices like the NVIDIA Tesla K20. Both operations max out the accelerator. In the case of large problems ($n \approx 10\,000$) this computational intensive operations will increase the device's temperature in every iteration. From Ge et. al. [5] it is known that it only takes a relatively short time until the devices starts throttling down its speed to avoid its overheating or to guarantee a proper power supply. This slow down can even result in the situation that the GPU computations are much slower that they could be done by the host CPU.

In our contribution we derive different strategies to handle this problem. We start with a straight-forward implementation using the LU-decomposition from the MAGMA library and the GEMM-operation from CUBLAS requiring at least $\mathcal{O}(3n^2)$ memory on the device. First, we replace the combination of the LU-decomposition and the forward-backward substitution by a Gauss-Jordan scheme to compute $A^{-1}E$ directly without forward-backward substitution. This reduces the number of sweeps the matrix A needs to be transfer between the GPU and its memory, In this way we save for the data transfers. Second, we want reduce the memory footprint on the device to $\mathcal{O}(2n^2)$. On the one hand this reduce the energy consumption of the device for problems of the same size because less memory cells must be refreshed and on the other hand it allows us the solve larger problems on the same device by saving up to $\mathcal{O}(n^2)$ memory on it. This memory footprint reduction is achieved using an asynchronous communication and computation scheme during the computation of

$$A_{i+1} := \frac{1}{2c_k} (A_i + c_k^2 EX),$$

¹Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtor-Str. 1, 39106 Magdeburg, Germany ,
koehlerm@mpi-magdeburg.mpg.de

²Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtor-Str. 1, 39106 Magdeburg, Germany ,
saak@mpi-magdeburg.mpg.de

where $X := A_i^{-1}E$. This schemes increase the communication between the host and the device but also allows us to introduce a fall back strategy if a frequency throttling of the device is detected. Using state information of the device we decide if we are still able to compute the update of A_{i+1} at highest performance on the accelerator device or if we have to move parts of the update to the host CPU in order to let the device cool down again.

References

- [1] Z. BAI AND J. DEMMEL, *Design of a parallel nonsymmetric eigenroutine toolbox, Part II*, tech. rep., Computer Science Division, University of California, Berkeley, CA 94720, 1994.
 - [2] P. BENNER, M. KÖHLER, AND J. SAAK, *Fast approximate solution of the non-symmetric generalized eigenvalue problem on multicore architectures*, in *Parallel Computing: Accelerating Computational Science and Engineering (CSE)*, M. Bader, A. B. H.-J. Bungartz, M. Gerndt, G. R. Joubert, and F. Peters, eds., vol. 25 of *Advances in Parallel Computing*, IOS Press, 2014, pp. 143–152.
 - [3] P. BENNER AND E. S. QUINTANA-ORTÍ, *Solving stable generalized Lyapunov equations with the matrix sign function*, *Numer. Algorithms*, 20 (1999), pp. 75–100.
 - [4] J. D. GARDINER AND A. J. LAUB, *A generalization of the matrix-sign-function solution for algebraic Riccati equations*, *Internat. J. Control*, 44 (1986), pp. 823–832.
 - [5] R. GE, R. VOGT, J. MAJUMDER, A. ALAM, M. BURTSCHER, AND Z. ZONG, *Effects of Dynamic Voltage and Frequency Scaling on a K20 GPU*, in *Parallel Processing (ICPP)*, 2013 42nd International Conference on, Oct 2013, pp. 826–833.
-

Using the Periscope Tuning Framework for Energy Optimization

Kai Diethelm¹

Michael Firbach²

Relevance of this work

Energy cost is nowadays a significant fraction of the total cost of running large HPC applications, besides other operating costs and the investment in hardware. In order to reduce total cost of ownership, compute centers usually have an interest in running as many jobs as possible per unit of time while maintaining moderate energy consumption.

While advances in hardware have accounted for most of the past improvements in energy efficiency, there are also some measures that can be implemented on the software side and take advantage of the current state of the application. Well-known schemes are parallelism capping and voltage/frequency scaling, which in effect limit the amount of compute resources available to the application in situations where those cannot be utilized efficiently. These measures are often under-estimated, although relatively easy to apply even to existing applications. We therefore implemented them in an existing auto-tuning framework and evaluated them using both synthetic and real-world applications.

The Periscope Tuning Framework

Our implementation relies on the Periscope Tuning Framework (PTF). The PTF is a framework that instruments, runs and profiles an HPC application while simultaneously executing a tuning plugin that determines the optimizations to apply to the application.

The tuning plugins follow a specific working cycle during which certain callback methods are invoked. In this cycle, the plugins are free to request *measurements*, like run time or energy consumption, and *tuning actions* based on the measurement results. One such tuning action could for example be to lower the CPU frequency when the plugin detects that an application is I/O bound.

The energy tuning plugins

The purpose of the energy tuning plugins is to minimize the *energy delay product* (EDP) of the running application. The EDP is a (physically implausible) construct that combines energy consumption and application runtime to provide a single metric that can be optimized. It is defined as

¹GNS Gesellschaft für numerische Simulation mbH, Am Gaußberg 2, 38114 Braunschweig, Germany, diethelm@gns-mbh.com

²Informatik 10 — Lehrstuhl für Rechnertechnik und Rechnerorganisation, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany, firbach@in.tum.de

$$\text{EDP} = E \cdot T^w.$$

Both minimal energy consumption (E) and minimal run time (T) are desirable traits in an HPC application and can be weighted against each other using the exponent w on the run time term. The exponent is chosen based on how important run time is regarded in comparison to the energy consumption (a decision to be taken by the compute center based on its corresponding policy), or which value optimizes total cost of ownership best for the specific compute hardware in question. Usually, values between 1 and 3 are chosen, and we refer to the resulting energy-delay-products as EDP1, EDP2 and EDP3 respectively. Three tuning plugins related to energy optimization have been developed so far:

- PCAP: Applies parallelism capping in an OpenMP-based application.
- MPI-Procs: Applies parallelism capping on the inter-node level of the application.
- DVFS: Applies dynamic voltage and frequency scaling to the application.

Evaluation

Evaluation of the plugins was done using both synthetic and real-world applications.

The first synthetic application uses a naive algorithm to determine all prime numbers in a certain range and computes their sum. This application is compute-intensive and scales very well to a high number of processes and threads. We expect that a high number of threads and a relatively high CPU frequency are beneficial to the energy consumption of this application because it allows it to finish more quickly.

The second application works on an array of objects. Each object contains a mutex that has to be locked in order to allow updates on the object. Each object has to be updated for a specific number of times before the application can terminate. Since the number of objects in the array is limited, the application does not scale well with an increasing number of processes and threads. Therefore, we expect a degrading speedup as more compute resources are added and a declining energy efficiency.

The real-world application in our test setup is the commercial special purpose finite element code *Indeed* that has been developed by GNS Gesellschaft für numerische Simulation mbH in order to simulate metal sheet forming processes. *Indeed* is an implicit solver meant to provide highly accurate simulation results. It is a hybrid application that uses a domain decomposition approach to set up a distributed memory computation based on MPI in combination with classical OpenMP-based shared memory techniques for dealing with the computational work on each subdomain (i.e., in each MPI process).

Conclusion

We can show that by applying simple auto-tuning techniques, the energy efficiency of both synthetic and real-world applications can be increased. In particular, for a typical run

of *Indeed* (which is strongly compute-dominated) our tuning techniques have led to EDP improvements in the range between 10% (for the EDP1 metric) and 30% (for EDP3).

While there are limitations to what auto-tuning can achieve, the little work required to apply these techniques justifies a moderate investment in time to reduce the total cost of ownership for large-scale production runs.

Acknowledgement

The work presented in this talk is part of the project *Score-E* which is supported by the German Ministry for Education and Research (BMBF) under grant No. 01IH13001.

An Integrated Framework for Power-Performance Analysis of Parallel Scientific Workloads

María Barreda¹

Sergio Barrachina, Sandra Catalán, Germán Fabregat, Rafael Mayo,
Enrique S. Quintana-Ortí¹

Manuel F. Dolz²

Introduction

Power consumption has traditionally been a strong constraint for mobile devices due to its impact on battery life. In recent years, the need to reduce energy expenses has also reached the market of desktop and server platforms, which are embracing “greener technologies”; and even in the high performance computing (HPC) arena, the power wall is now recognized as a crucial challenge that the community will have to face.

While system power management has experienced considerable advances during this past period, application software has not benefited from the same degree of attention, in spite of the power harm that an ill-behaving software ingredient can infer. Indeed, tracing the use of power made by scientific applications and workloads is key to detecting energy bottlenecks and understanding power distribution. However, as of today, the number of fully integrated tools for this purpose is insufficient to satisfy a rapidly increasing demand. In this work, we present the PMLIB tool [1, 2] for power-performance analysis of parallel scientific applications.

Overview of PMLib

PMLIB is a framework for power-performance analysis of parallel scientific codes. The framework, with stronger integration and modular design, collects samples of a large number of power sampling devices, including external commercial products, such as APC 8653 PDU and WattsUp? Pro .Net, internal DC powermeters, like a commercial data acquisition system (DAS) from National Instruments (NI) and, alternatively, our own designs that use a microcontroller to sample transducer data.

Calls to the PMLIB application programming interface (API) from the application instruct the tracing server to start/stop collecting the data captured by the powermeters, dump the samples in a given format into a disk file (power trace), query different properties of the powermeters, etc. Upon completion of the application’s execution, the power trace can be inspected, optionally hand-in-hand with a performance trace, using some visualization tool.

¹Depto. de Ingeniería y Ciencia de Computadores, Universidad Jaime I, 12.071–Castellon, Spain, mvaya@uji.es, barrachi@uji.es, catalans@uji.es, fabregat@uji.es, mayo@uji.es, quintana@uji.es

²Dept. of Informatics, University of Hamburg, 22.527–Hamburg, Germany, manuel.dolz@informatik.uni-hamburg.de

A module to detect power-related states

Most current processors adhere now to the the Advanced Configuration and Power Interface (ACPI) specification, which defines an open standard for device configuration and power management from the operating system. For our power monitoring purposes, the ACPI specification defines a series of processor states, collectively known as C-states, that are valid on a per-core basis. Moreover, a core in C0 state can be in one of several performance states, referred to as P-states. These modes are architecture-dependent, though P0 always corresponds to the highest performance state, with P1 to Pn being successively lower performance modes.

Our power framework obtains a trace of the C- and P-states of each core. In order to obtain information on the C-states, a daemon integrated into the power framework reads the corresponding MSR (Model Specific Registers) of the system, for each CPU X and state Y, with a user-configured frequency. Note that the state-recording daemon necessarily has to run on the target application and, thus, it introduces a certain overhead (in terms of execution time as well as power consumption) which, depending on the software that is being monitored, can become nonnegligible. To avoid this effect, the user is advised to experimentally adjust the sampling frequency of this daemon with care.

Automatic Detection of Power Bottlenecks

The visual inspection of performance and power traces is useful but detecting bottlenecks in it sometimes becomes a burden and an error prone process. To facilitate this task, we have elaborated an extension of the PMLIB framework for power-performance analysis that permits a rapid and automatic detection of power sinks during the execution of concurrent scientific workloads [3]. The extension is shaped in the form of a multithreaded Python module that offers high reliability and flexibility, rendering an overall inspection process that introduces low overhead. The detection of power sinks is based on a comparison between the application performance trace and the C-state traces per core. When a core is supposed to be in an “idle” state but the C-state is C0 (i.e, active), the tool detects a power sink. Moreover, the analyzer is flexible, because the task type that correspond to “useful” work can be defined by the user; furthermore, the length of the analysis interval and the divergence (discrepancy) threshold are parameters that can be adjusted by the user to the desired level.

References

- [1] P. Alonso, R. M. Badia, M. Barreda, M. F. Dolz, J. Labarta, R. Mayo and E.S. Quintana-Ortí and R.Reyes. Tools for power and energy analysis of parallel scientific applications. *41st International Conference on Parallel Processing-ICPP*, pages 420–429, 2012.
 - [2] S. Barrachina, M. Barreda, S. Catalán, M. F. Dolz, G. Fabregat, R. Mayo and E.S. Quintana-Ortí. An Integrated Framework for Power-Performance Analysis of Parallel Scientific Workloads. *3rd Int. Conf. on Smart Grids, Green Communications and IT Energy-aware Technologies*, pages 114-119, 2013.
-

- [3] M. Barreda, S. Catalán, M. F. Dolz, R. Mayo, E. S. Quintana-Ortí. Automatic detection of power bottlenecks in parallel scientific applications. *Fourth International Conference on Energy-Aware High Performance Computing, Computer Science Research and Development*, Vol.29(3-4), pages 114-119, 2013.

List of Participants

Name	First Name	Affiliation	Email	Country	Page
Aliaga Estellés	José L.	Universidad Jaime I	aliaga@uji.es	Spain	16
Baran	Björn	Max Planck Institute for Dynamics of Complex Technical Systems	baran@mpi-magdeburg.mpg.de	Germany	
Barreda	María	Universidad Jaime I	mvaya@uji.es	Spain	29
Benner	Peter	Max Planck Institute for Dynamics of Complex Technical Systems	benner@mpi-magdeburg.mpg.de	Germany	
Blum	Thomas	MEGWARE Computer GmbH, Vertrieb und Service	thomas.blum@megware.com	Germany	21
Bollhöfer	Matthias	TU Braunschweig, Inst. Computational Mathematics	m.bollhoefer@tu-bs.de	Germany	
Diethelm	Kai	GNS Gesellschaft für numerische Simulation mbH	diethelm@gns-mbh.com	Germany	26
Dolz	Manuel F.	Universität Hamburg	manuel.dolz@informatik.uni-hamburg.de	Germany	
Dufrechou	Ernesto	Facultad de Ingeniería, Universidad de la República	edufrechou@ing.edu.uy	Uruguay	14
Ezzatti	Pablo	Facultad de Ingeniería, Universidad de la República	pezzatti@ing.edu.uy	Uruguay	
Geveler	Markus	TU Dortmund, Inst. for Applied Mathematics	markus.geveler@math.tu-dortmund.de	Germany	10
Hager	Georg	Erlangen Regional Computing Center (RRZE)	georg.hager@fau.de	Germany	8
Heydemüller	Jörg	MEGWARE Computer GmbH, Vertrieb und Service	joeorg.heydemue1ler@megware.com	Germany	
Himpe	Christian	University of Muenster	christian.himpe@uni-muenster.de	Germany	20
Köhler	Martin	Max Planck Institute for Dynamics of Complex Technical Systems	koehler@mpi-magdeburg.mpg.de	Germany	24
Mena	Hermann	University of Innsbruck	hermann.mena@uibk.ac.at	Austria	
Milk	René	WWU Muenster - Institute for Computational and Applied Mathematics	rene.milk@wwu.de	Germany	19
Mlinaric	Petar	Max Planck Institute for Dynamics of Complex Technical Systems	m1inaric@mpi-magdeburg.mpg.de	Germany	
Penke	Carolin	Max Planck Institute for Dynamics of Complex Technical Systems	carolin.penke@mpi-magdeburg.mpg.de	Germany	
Quintana-Ortí	Enrique S.	Universidad Jaime I	quintana@icc.uji.es	Spain	10
Remón	Alfredo	Max Planck Institute for Dynamics of Complex Technical Systems	remon@mpi-magdeburg.mpg.de	Germany	22
Saak	Jens	Max Planck Institute for Dynamics of Complex Technical Systems	saak@mpi-magdeburg.mpg.de	Germany	

Additional Information

On Site

- Room:
All talks will be given in the seminar room *Wiener* on the ground floor of the MPI.
- Coffee breaks:
Coffee, tea, soda, juice, and cookies will be provided during the breaks in front of the neighboring seminar room *Prigogine*.
- Lunch breaks:
Lunch will be served free of charge at the MPI Cafeteria.
- WLAN:
eduroam is available everywhere in the institute building. You may also obtain a guest account for the MPI guest network. For the account, you have to sign at the registration desk.
- Conference dinner:
The conference dinner takes place at the Restaurant “Los Amigos” on Monday evening. The dinner and the drinks on the table upon arrival are kindly sponsored by MegWare.

For Speakers

- Please make sure that your presentation is transferred to the computer (Windows XP, Adobe Reader 11 and PowerPoint 2010) connected to the beamer before your session starts.
- Ask the local organizers if you have any questions.

Local organizers

- Prof. Dr. Peter Benner
- Dr. Jens Saak
- Dr. Alfredo Remón
- Martin Köhler
- Janine Holzmann

Important phone numbers

- Emergency number: **112**
 - MPI reception: **+49 (0)391 611 00**
 - Taxi office: **+49 (0)391 565 650**
-

How to reach the MPI

- By airplane:
The next airports are Berlin-Tegel, Hannover and Halle-Leipzig. All airports have a good train connection to Magdeburg.
- By train/ public transport:
From Magdeburg Hauptbahnhof (central station) you go to *Damaschkeplatz* (tram stop is located behind the main station) or *Alter Markt* (5 minutes walk) and use the MVB to reach the stop *Askanischer Platz* which is next to the MPI. Tram 5 goes directly from *Alter Markt* to *Askanischer Platz*. If you use the tram stop *Damaschkeplatz* you have to change lines. See the MVB maps for more information. There is also a taxi stand in front of the Hauptbahnhof (central station).
- By car:
 - coming from Hannover and Berlin: via highway A2 until the exit 70 (Magdeburg-Zentrum); follow B189/B71 (Magdeburger Ring) in south direction (Halle/Halberstadt) until exit Universität/Zentrum-Nord; turn right on Albert-Vater-Straße, in east direction on Walther-Rathenau-Straße; just before the Jerusalembrücke (bridge over the river Elbe) turn left on Sandtorstraße.
 - coming from Halle: via highway A14 until the exit 5 (Magdeburg-Sudenburg/Magdeburg-Zentrum); follow B189/B71 (Magdeburger Ring) in north direction (Hannover/Berlin) until exit Universität/Zentrum-Nord; turn right on Albert-Vater-Straße, in east direction on Walther-Rathenau-Straße; just before the Jerusalembrücke (bridge over the river Elbe) turn left on Sandtorstraße.

The institute has got a gated Parking area (behind the building in travel direction). An intercom-system connects to the MPI reception desk to open the barrier.

- From Motel One:
Motel One is located at Domplatz close to the Hundertwasser building. The tram stop *Leiterstraße* is in front of that building. You can take tram 5 (direction Messegelände) to reach the stop *Askanischer Platz*. All in all you reach the MPI in around 15 minutes. (Note that tram number 5 is going only every 20 minutes.) Alternatively you can walk along the Elbe and will need about 20 minutes.
- From Hotel Geheimer Rat:
You can take the tram from either *Arndtstr.* or *Alexander-Puschkin-Str.*. From *Arndtstr.* take tram 6 (direction Herrenkrug) and switch to tram 5 at *Alter Markt* (direction Messegelände) or *Jerichower Platz* (direction Klinikum Olvenstedt). From *Alexander-Puschkin-Str.* you can take tram 5 (direction Messegelände) directly at the cost of a small detour.
The Hotel is located 3.5km (roughly 45min) walking distance from the MPI. Check your favorite map provider for directions.

How to reach restaurant “Los Amigos” (conference dinner)

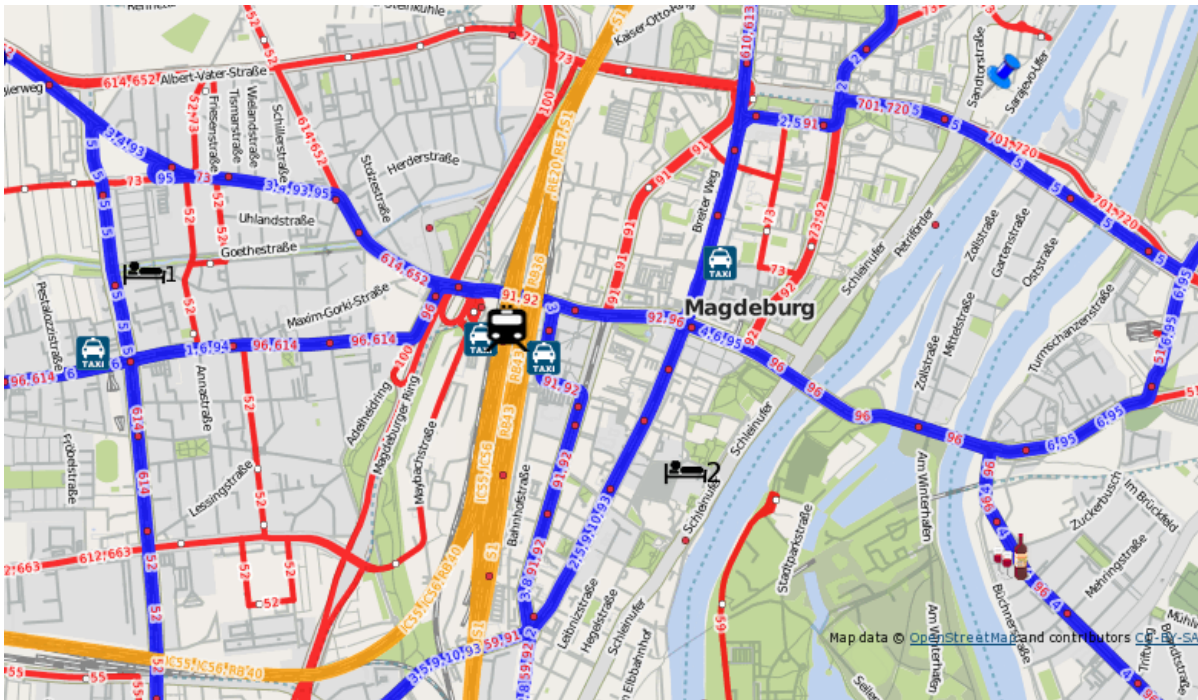
- By tram:
Take tram 5 from *Askanischer Platz* to *Alter Markt* switch to line number 4 in
-

direction "Cracau" to the stop "Am Cracauer Tor". From there follow the road further until you see the restaurant on the right. You can not miss the typical Spanish bull advertisement.

- Taking a walk:

Take a nice 15-20 minutes walk along the river Elbe until you reach the next bridge. Cross the river (both arms) and turn right at "Cracauer Str." Follow the "Cracauer Str." for another few minutes until you see the restaurant on the right.

City Map



Max Planck Institute for Dynamics of Complex Technical Systems



Restaurant Los Amigos (Conference Dinner)



Hotel Geheimer Rat

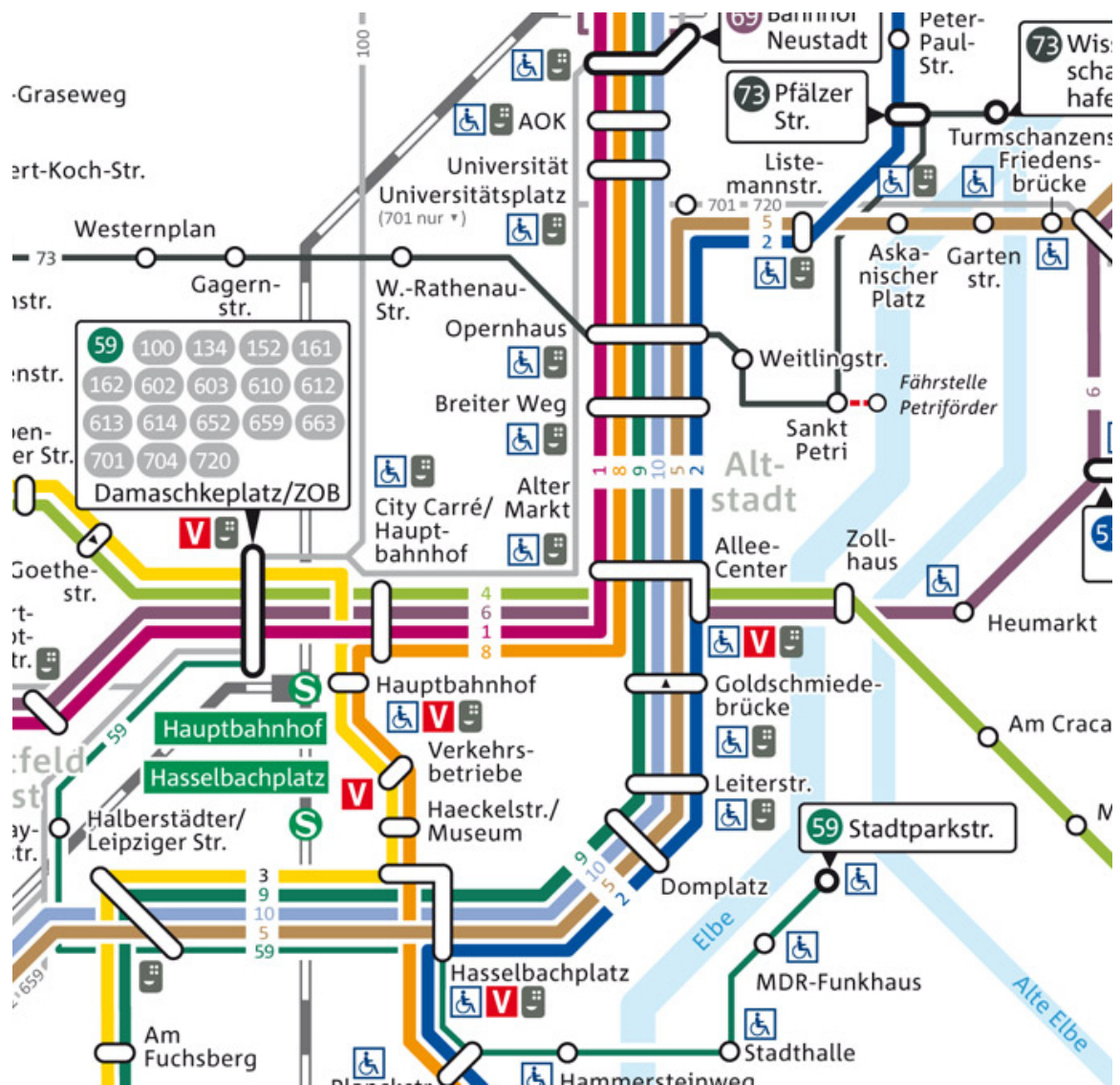


Motel One



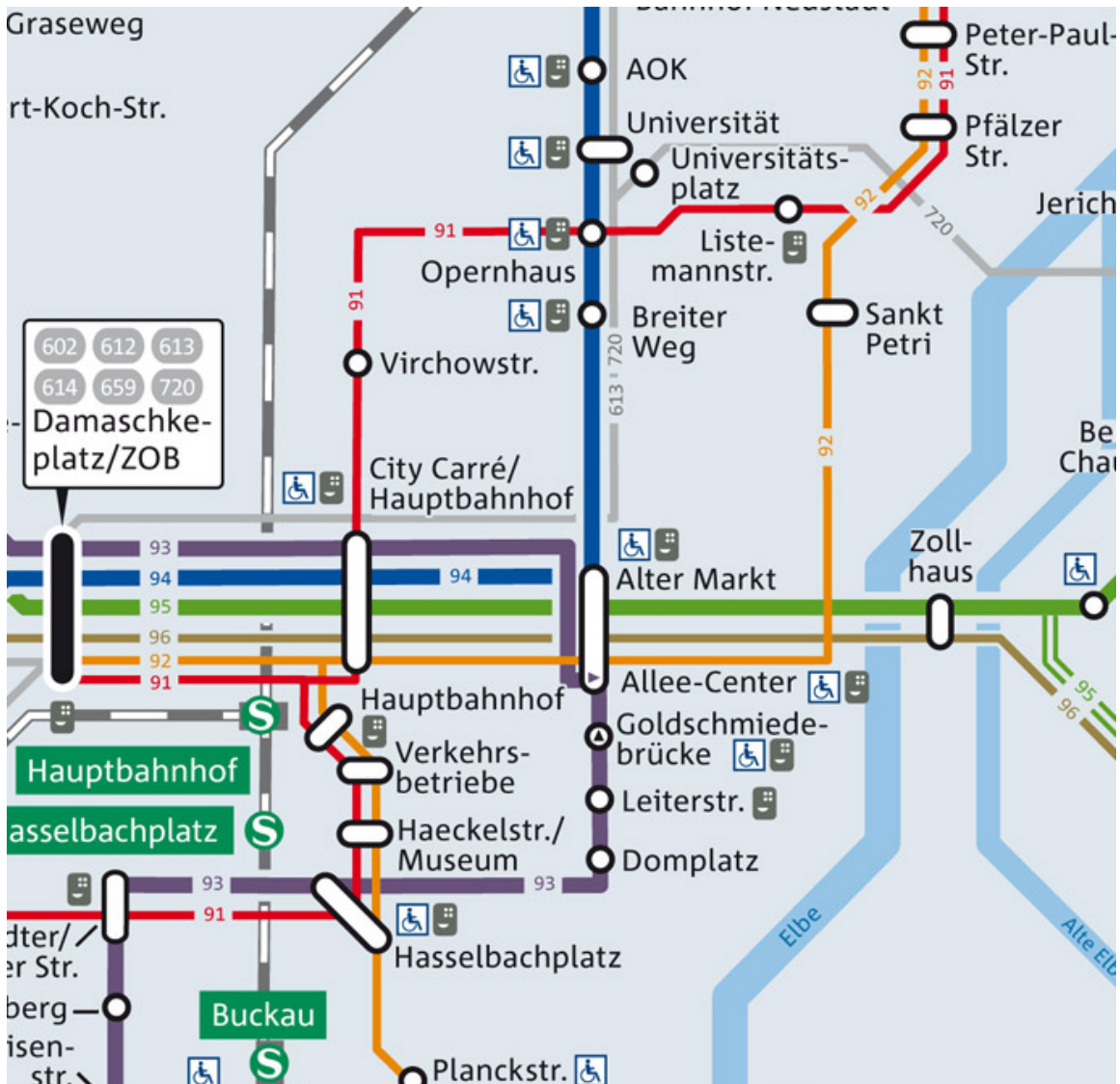
Hauptbahnhof / central station

Excerpt of the MVB map (before 9 pm)



<http://www.mvbnet.de/verkehr/liniennetzplaene/am-tag/>

Excerpt of the MVB map (after 9 pm)



<http://www.mvbnet.de/verkehr/liniennetzplaene/zum-anschlussverkehr/>

Notes
