

A unified Energy Footprint for Simulation Software

H. Anzt, A. Beglarian, S. Chilingaryan, A. Ferrone,
V. Heuveline, A. Kopmann

No. 2012-04

Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)





Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)
ISSN 2191-0693
No. 2012-04

Impressum

Karlsruhe Institute of Technology (KIT)
Engineering Mathematics and Computing Lab (EMCL)

Fritz-Erler-Str. 23, building 01.86
76133 Karlsruhe
Germany

KIT – University of the State of Baden Wuerttemberg and
National Laboratory of the Helmholtz Association

Published on the Internet under the following Creative Commons License:
<http://creativecommons.org/licenses/by-nc-nd/3.0/de> .



www.emcl.kit.edu

A unified Energy Footprint for Simulation Software

Hartwig Anzt · Armen Beglarian ·
Suren Chilingaryan · Andrew Ferrone ·
Vincent Heuveline · Andreas Kopmann

Received: date / Accepted: date

Abstract The focus in High-Performance Computing increasingly turns to energy efficiency. Therefore the pure concentration on floating point operations and runtime performance is no longer sufficient. In terms of hardware, this change of paradigm has already taken place: The GREEN500 list as counterpart to the runtime performance oriented TOP500 list has been established. The new metrics take runtime and energy consumption into account. Nevertheless, all these developments consider hardware only - still an inadequate situation to face the challenges of Energy-Efficient Exascale Computing. The necessity of optimizing simulation software with respect to power and energy draft demands for detailed profiling of the power consumption during the calculations and a norm quantifying the respective efficiency. In this paper we propose a unified energy footprint for simulation software that enables a fast comparison between different models, implementations and hardware configurations, respectively. By way of example we provide the footprints for the tomographic reconstruction code PyHST optimized for CPU and GPU operation as well as the operational numerical weather prediction model COSMO. We then discuss the power and energy profiles and investigate the effects of scaling with respect to hardware resources and simulation parameters.

Keywords Benchmarking

The authors want to thank Peter Schöck, Frederic Hupbauer and Alexander Biewer for their support in employing the measurement setup. Furthermore, thanks go to KIT for funding the project in the framework of a Startup-Project.

Karlsruhe Institute of Technology, Karlsruhe, Germany.
E-mail: {hartwig.anzt, armen.beglarian, suren.chilingaryan, andrew.ferrone, vincent.heuveline, andreas.kopmann }@kit.edu

1 Introduction

In the Exaflop Era, the energy consumption might become the limiting factor for further development in simulation based research. Already today, the running energy costs of computing centers often exceed the acquisition costs after few years, and the carbon dioxide footprint becomes a serious factor when arguing for larger computing facilities [5]. Besides, it may no longer be obvious whether running computer simulations is always more resource efficient than conducting the experiments in reality.

Although many hardware developers have realized the importance of the issue, and put strong efforts in developing power-efficient devices and energy-saving mechanisms, most scientific simulation software is still optimized for runtime performance only [7]. A paradigm change is necessary to meet the challenge of energy-efficient computing. In this context, an essential first step is the detailed power and energy analysis of current simulation software enabling the fast comparison between different simulation models, implementations and hardware configurations. Furthermore, it simplifies the optimization with respect to metrics combining energy consumption, runtime performance and acquisition costs [1].

We target the challenge of quantifying the energy efficiency of simulation software by proposing an energy footprint that provides not only information about the power draft over the runtime, the total energy consumption of one typical simulation run, but also the scaling with respect to hardware resources and simulation parameters like additional cores or extended simulation time.

2 Hybrid Hardware Platform and Measurement Setup

The hardware system we use for the power and energy experiments is a heterogeneous GPU-accelerated multicore system. It is equipped with two Intel XEON E5540 @ 2.53GHz and 192 GB of main memory [10]. The system is furthermore accelerated by 4 Fermi C2070 (each with 14 Multiprocessors x 32 CUDA cores @1.15GHz, 6 GB memory). While the CPU interconnection is handled by QPI (up to 16 GB/s), always two GPUs are connected to one CPU through a PCI-Express bus with 16 lanes and up to 8 GB/s throughput [10]. The measurement setup for power analysis is inspired by the layout employed at the High Performance Computing & Architectures research group at the University of Jaume I, Spain [2]. External powermeters monitor voltages and currents in all lines powering the chipset, the hard-disks and the GPUs. Also the power provided to the GPUs via the PCI-express and the power consumption of the GPU fans is considered. The sketch in Figure 1 shows the measurement setup. The measurement system is based on an NI-9188 CDAQ 8-slot Ethernet chassis from National Instrument designed for remote or distributed sensors and electrical measurements. Eight analog input modules NI-9205 which features 32 single-ended or 16 differential analog inputs, 16-bit resolution, and a maximum sampling rate of 250 kS/s have been installed. Each channel has a programmable input ranges of 200 mV, 1, 5,

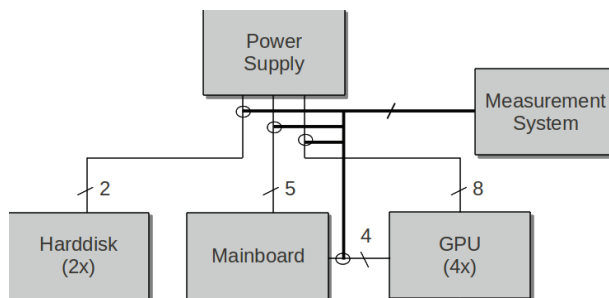


Fig. 1: Embedded Power Measurement System for harddisks, mainboard and GPUs. All three power sources of the GPUs are recorded.

and 10 V. The voltages are connected either directly to the NI-9205 modules or via attenuator, if input values exceed the 10 V limit. For current measurements compact Hall-sensor modules HX 10-P from LEM with linearity $\pm 1\%$ of nominal are used.

CPU Power Analysis Before running scientific application, we evaluate power dissipation caused by the CPU cores. The benchmark coreburn [8] has been used to simulate a high CPU load on an increasing number of cores. Power demand is rising from 119.45 W in idle state to 197.75 W when both CPUs are running under full load. The curve is not linear, but we can estimate an increase of the power demand of about 6.5 Watts per activated core. The power consumption of coreburn is used as reference in the energy footprint.

3 A Unified Energy Footprint for Simulation Software

To compress as much information as possible in an energy footprint it is necessary to identify the significant parameters that enable estimations when using the software on a different system, with a different configuration or applied to a different data set. Since the performance information for a typical model run implies a certain hardware configuration, it is essential to include power parameters characterizing the hardware platform. Optimizing with respect to multi-parameter metrics taking power, energy and runtime into account also requires the information about runtime performance and average power demand including peaks. The challenge is to pack all this information into just a few lines and figures. In experiments we identified the following parameters as necessary:

1. Short Information about the used Hardware Platform, the Simulation Model and a typical application/hardware configuration.
2. Power profile for the typical simulation run including the total runtime, average power demand of the distinct hardware components, the total energy draft and split to the distinct components (chipset includes the CPUs).

Providing the total and the net energy consumption provides insight into the energy profile of the hardware system when running idle.

3. The scaling with respect to used hardware resources enables to estimate the total energy demand when running the application with a different hardware configuration. Including the energy consumption when running the system idle or using coreburn for the same time provides information about the runtime performance behavior when scaling on the hardware, and the respective efficiency. I.e. the model efficiency may decrease with increasing resources as the memory /communication becomes the bottleneck. This can then be identified by a lower slope of the energy curve in comparison to the curves for running idle/coreburn.
4. The scaling with respect to simulation parameters enables to estimate the energy cost when running the simulation on different data sets.

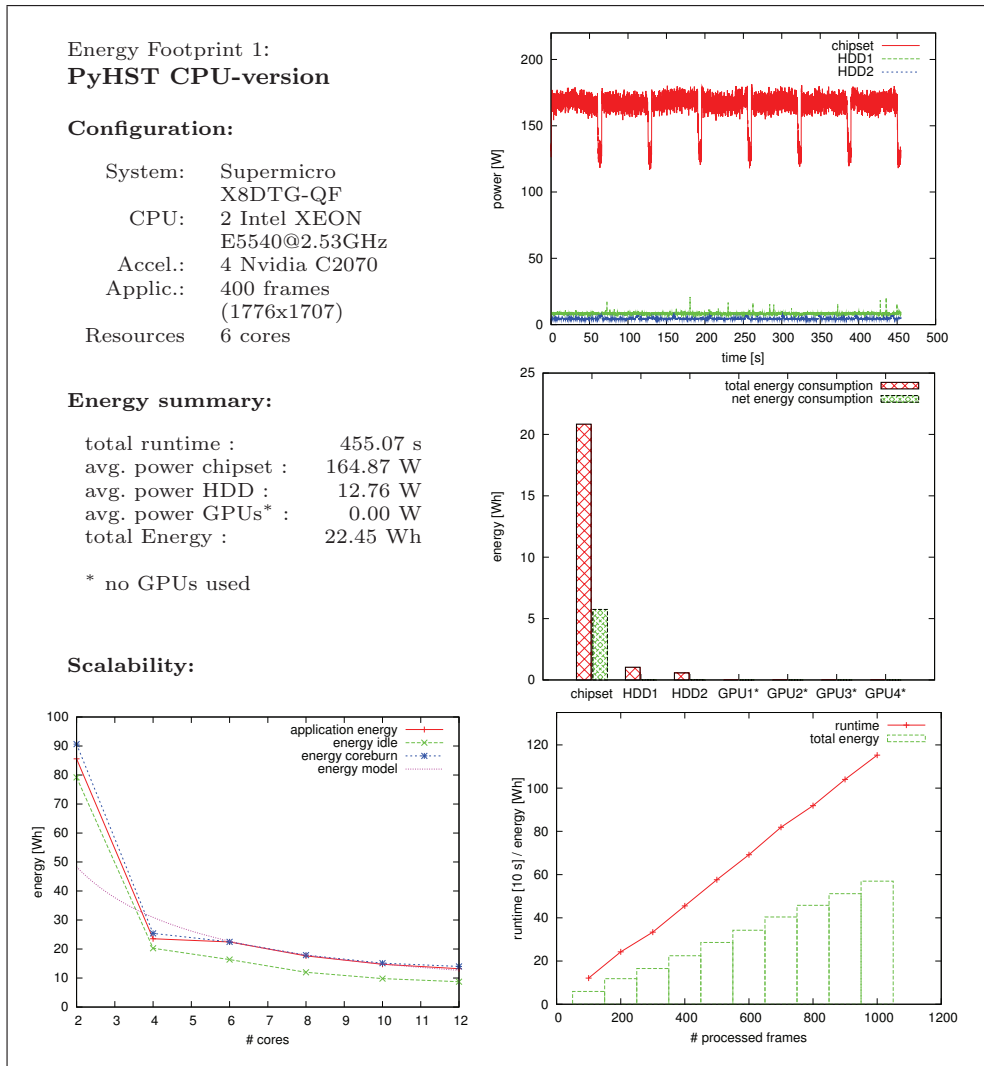
Although we know that there exist parameters like accuracy, memory demand and scaling w.r.t. different hardware communication interfaces that are also important when providing a comprehensive footprint, we neglect these to keep the information as compact as possible.

4 Comparing Power Consumption of CPU and GPU

The scientific program PyHST has been selected as a test case for analyzing the power consumption of compute intensive applications. It is intended to reconstruct 3D images of objects from multiple projections the using the Filtered Back Projection (FBP) algorithm. PyHST has been developed at the ESRF [6], and is currently used as standard application for X-ray tomography at several synchrotron facilities in Europe. The operations splits roughly in four major steps, loading the projections, filtering, back projections and saving the resulting 3D image back to disk. For the application highly-optimized CPU and GPU versions are available. The computation time with a single GPU could be reduced by a factor of 10 compared to the optimized CPU version [3]. The task scales linear with the number of processors and thus more GPUs will even increase this figure.

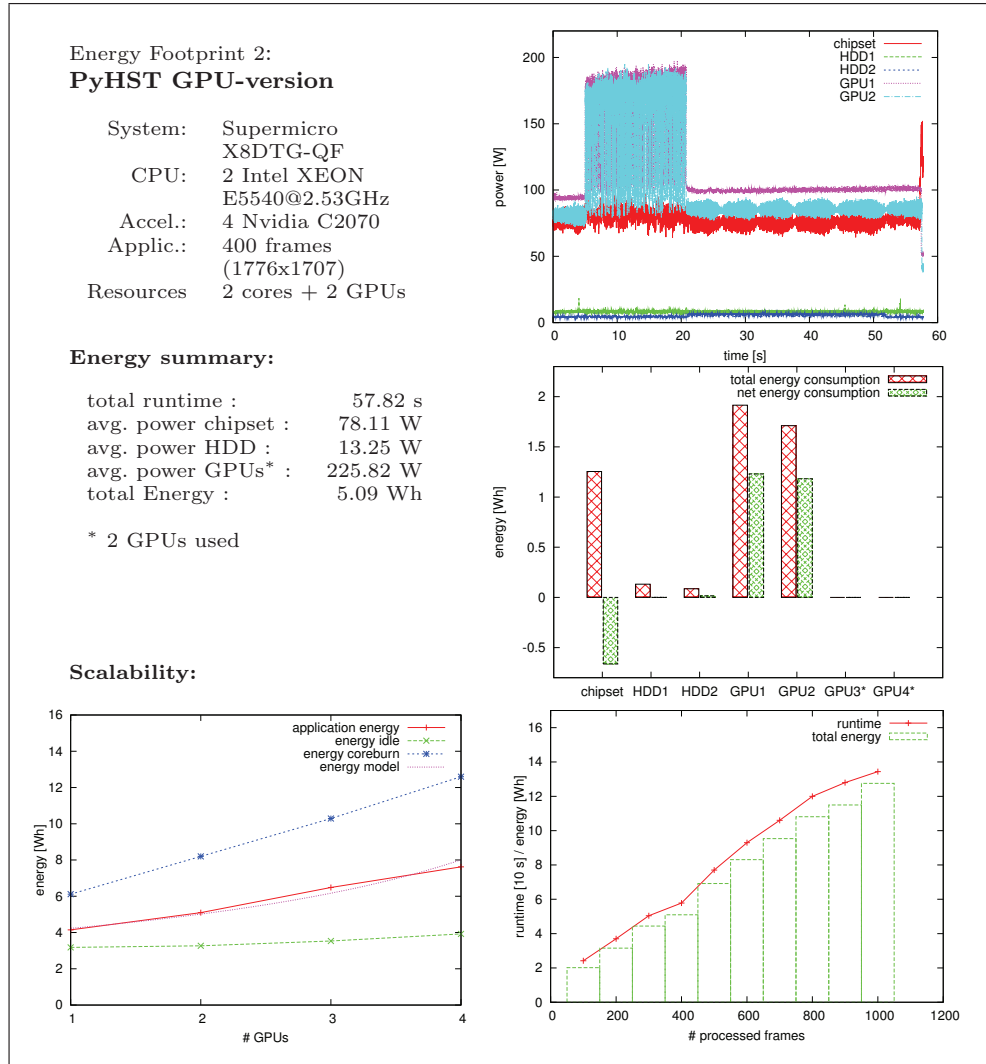
Footprint 1 provides information about the energy usage of the PyHST CPU version. The segments reconstructing the information and the read/write phases can clearly be distinguished. Due to the memory usage, the overall energy consumption exceeds the coreburn energy demand when running for the same time. While the scaling with respect to processed frames is almost linear, the scaling with respect to hardware resources shows some specific pattern. This is due to the algorithm design, which uses always one core for I/O operations, and the remaining cores for rendering. The fact that the application energy almost equals the coreburn energy shows the high computational intensity. While this is especially true for high core numbers, the I/O-phases are the only limitations.

Comparing with the energy footprint of the GPU version (see Footprint 2), the power profile looks different. After reading data and memory allocation we observe a reconstruction phase handled by the GPU, followed by a



long phase, where the I/O cache is synchronized with the hard drives. This is inefficient because the GPUs are idle during this time. Still the total energy needed for the processing of the same dataset is significantly lower than for the CPU version. Splitting the energy into the hardware resources results in a negative net energy consumption for the CPU, which can be explained by underpowering of the chipset: while the GPU is active, the chipset consumes less power compared to running idle, as the GPU consumes more power via the PCI than the chipset power rise.

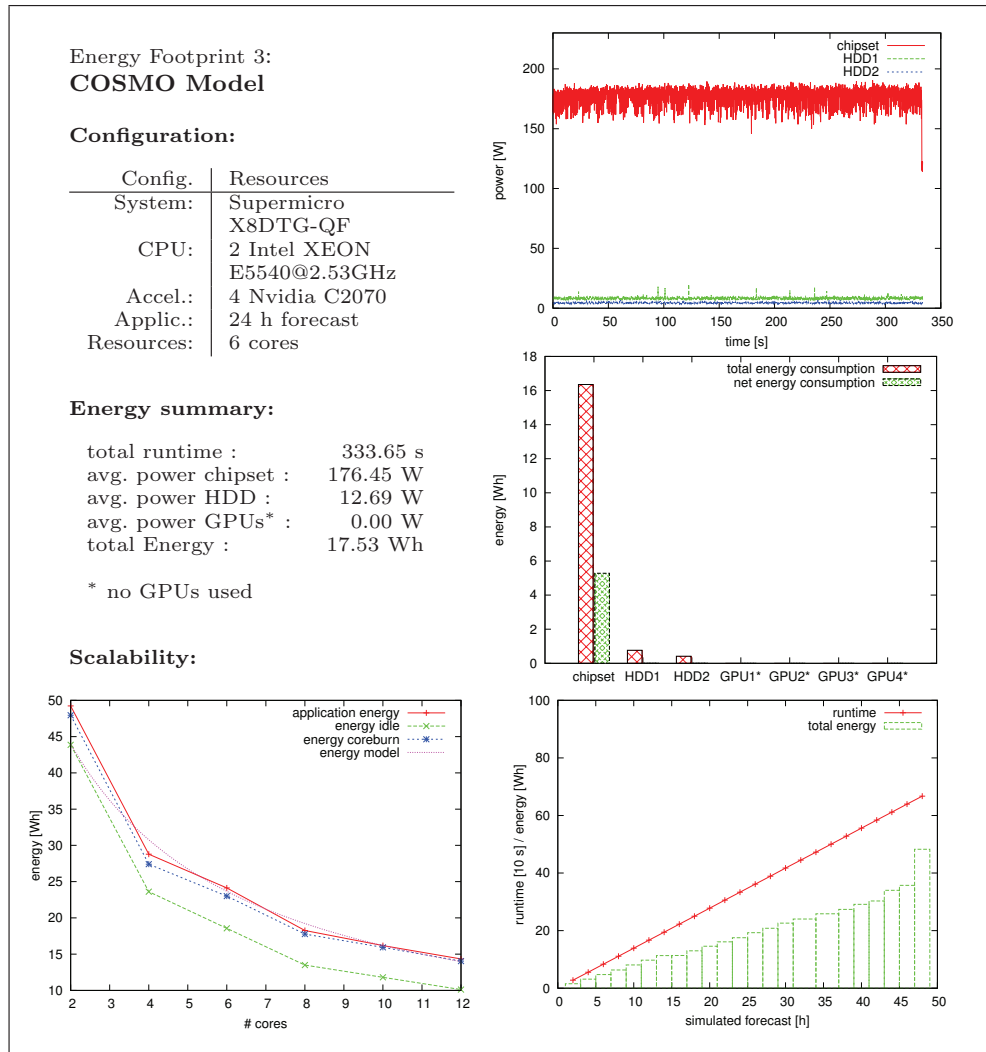
Because of the long idle phases the power consumption of the GPU code increases with the number of devices. The additional power usage in the I/O phase can not be compensated by the acceleration during the reconstruction phase. Hence, we conclude that for the test case of 400 frames, a hardware setup featuring 1 GPU is most energy efficient. Calculating the energy cost



and the number of runs it is possible to estimate whether purchasing one GPU pays off, or whether the CPU-version with the extended runtime is more economical.

5 COSMO Model System

The model has initially been developed by the German Weather Service and is their current operational model [4], [9]. We use the limited area atmospheric model COSMO, version 4.21. The dynamics of the model integrate the hydrothermodynamical equations in conservation form using a split explicit thirdorder RungeKutta scheme with a spatial discretization based on a fifthorder upstream advection scheme. In the horizontal the equations are discretized on a rotated Arakawa C grid and in the vertical a terrainfollowing pressurebased hybrid coordinate is used. Vertical turbulent diffusion is treated



with a diagnostic TKE scheme and a Tiedtke scheme is applied to parameterize convection. The radiative transfer scheme was developed by Ritter and Geleyn.

In Footprint 3 we analyze the COSMO Model with respect to its power and energy characteristics. The power profile shows a characteristic pattern of more and less compute intense periods. This is due to the algorithm design, that entails some communication between the computing cores. The disk usage can be neglected in the overall energy draft. Extending the simulation time corresponds to linear increase in runtime and energy demand, with some overhead due to the initialization process. The excellent scaling with respect to hardware resources, at least for the core numbers we considered, shows the high optimization level of the widespread used simulation model: From the idle slope we can approximate a speedup factor of about 1.7 when doubling the core number from 6 to 12. Concerning the energy demand, we reduce the

overall draft by 30% while maintaining the high resource usage. When evaluating a suitable hardware platform, for the typically conducted simulation setup the trade-off between runtime relevance, acquisition costs and power cost has to be considered.

6 Conclusions

We have proposed an energy footprint for scientific simulation software based on an energy measurement system for CPU and GPU applications. The footprint includes detailed information about the power and energy consumption for a typical simulation and hardware setup as well as the scaling with respect to hardware resources and modifications with respect to simulation parameters. Furthermore, we have shown how information can be extracted out of the footprints to evaluate the energy efficiency, and to whether purchasing additional hardware is beneficial by compensating the acquisition costs through reduced energy consumption. This simplifies decisions when acquiring a hardware system for a specific simulation scenario. The unification of such energy information aims for creating a standard that enables the profiling of most relevant simulation software. Creating a database of these energy footprints would provide the possibility of fast comparison between different algorithms on the respective hardware setups.

References

1. Costas Bekas and Alessandro Curioni. A new energy aware performance metric. *Computer Science - Research and Development*, 25:187–195, 2010. 10.1007/s00450-010-0119-z.
2. M. Castillo, M. Dolz, J.C. Fernandez, R. Mayo, E.S. Quintana-Orti, and V. Roca. Evaluation of the Energy Performance of Dense Linear Algebra Kernels on Multi-core and Many-Core Processors. In *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, pages 846–853, may 2011.
3. S. Chilingaryan, A. Mirone, A. Hammersley, C. Ferrero, L. Helfen, A. Kopmann, T. dos Santos Rolo, and P. Vagovic. A GPU-Based Architecture for Real-Time Data Assessment at Synchrotron Experiments. *IEEE Transactions on Nuclear Science*, 58:1447–1455, 2011.
4. G. Doms and U. Schättler. A description of the nonhydrostatic regional model LM. Part I: Dynamics and numerics. *Deutscher Wetterdienst, Offenbach, Germany*, 2002.
5. J. Dongarra *et al.* The international ExaScale software project roadmap. *Int. J. of High Performance Computing & Applications*, 25(1), 2011.
6. A. Hammersley and A. Mirone. *High speed tomography reference manual*. <http://www.esrf.eu/computing/scientific/FIT2D/HST/hst.html>.
7. J. Ou, S. B. Choi, and V. K. Prasanna. Energy-Efficient Hardware/software Co-Synthesis for a Class of Applications on Reconfigurable SoCs. *International Journal of Embedded Systems*, 1:91 – 102, 2005.
8. Robert Redelmeier. <http://www.digipedia.pl/man/doc/view/burnp6.1/>.
9. J. Steppeler, G. Doms, U. Schttler, H. W. Bitzer, A. Gassmann, U. Damrath, and G. Gregoric. Meso-gamma scale forecasts using the nonhydrostatic model LM. *Meteorology and Atmospheric Physics*, 82:75–96, 2003. 10.1007/s00703-001-0592-9.
10. Super Micro Computer, Inc. *Supermicro X8DTG-QF User's Manual*, revision 1.0a edition, 2010.

Preprint Series of the Engineering Mathematics and Computing Lab

recent issues

- No. 2012-03 Vincent Heuveline, Chandramowli Subramanian: The Coffee-table Book of Pseudospectra
- No. 2012-02 Dominik P.J. Barz, Hendryk Bockelmann, Vincent Heuveline: Electrokinetic optimization of a micromixer for lab-on-chip applications
- No. 2012-01 Sven Janko, Björn Rocker, Martin Schindewolf, Vincent Heuveline, Wolfgang Karl: Software Transactional Memory, OpenMP and Pthread implementations of the Conjugate Gradients Method - a Preliminary Evaluation
- No. 2011-17 Hartwig Anzt, Jack Dongarra, Vincent Heuveline, Piotr Luszczek: GPU-Accelerated Asynchronous Error Correction for Mixed Precision Iterative Refinement
- No. 2011-16 Vincent Heuveline, Sebastian Ritterbusch, Staffan Ronnås: Augmented Reality for Urban Simulation Visualization
- No. 2011-15 Hartwig Anzt, Jack Dongarra, Mark Gates, Stanimire Tomov: Block-asynchronous multigrid smoothers for GPU-accelerated systems
- No. 2011-14 Hartwig Anzt, Jack Dongarra, Vincent Heuveline, Stanimire Tomov: A Block-Asynchronous Relaxation Method for Graphics Processing Units
- No. 2011-13 Vincent Heuveline, Wolfgang Karl, Fabian Nowak, Mareike Schmidtobreck, Florian Wilhelm: Employing a High-Level Language for Porting Numerical Applications to Reconfigurable Hardware
- No. 2011-12 Vincent Heuveline, Gudrun Thäter: Proceedings of the 4th EMCL-Workshop Numerical Simulation, Optimization and High Performance Computing
- No. 2011-11 Thomas Gengenbach, Vincent Heuveline, Mathias J. Krause: Numerical Simulation of the Human Lung: A Two-scale Approach
- No. 2011-10 Vincent Heuveline, Dimitar Lukarski, Fabian Oboril, Mehdi B. Tahoori, Jan-Philipp Weiss: Numerical Defect Correction as an Algorithm-Based Fault Tolerance Technique for Iterative Solvers
- No. 2011-09 Vincent Heuveline, Dimitar Lukarski, Nico Trost, Jan-Philipp Weiss: Parallel Smoothers for Matrix-based Multigrid Methods on Unstructured Meshes Using Multicore CPUs and GPUs
- No. 2011-08 Vincent Heuveline, Dimitar Lukarski, Jan-Philipp Weiss: Enhanced Parallel $ILU(p)$ -based Preconditioners for Multi-core CPUs and GPUs – The $Power(q)$ -pattern Method
- No. 2011-07 Thomas Gengenbach, Vincent Heuveline, Rolf Mayer, Mathias J. Krause, Simon Zimny: A Preprocessing Approach for Innovative Patient-specific Intranasal Flow Simulations
- No. 2011-06 Hartwig Anzt, Maribel Castillo, Juan C. Fernández, Vincent Heuveline, Francisco D. Igual, Rafael Mayo, Enrique S. Quintana-Ortí: Optimization of Power Consumption in the Iterative Solution of Sparse Linear Systems on Graphics Processors