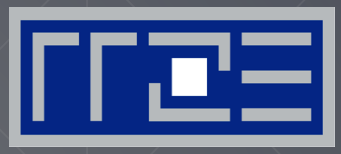
ERLANGEN REGIONAL
COMPUTING CENTER



Efficient numerical simulation on multicore processors (MuCoSim) WS 2016/7

Prof. Gerhard Wellein, Dr. G. Hager

Department für Informatik & HPC Services
Regionales Rechenzentrum Erlangen (RRZE)

http://moodle.rrze.uni-erlangen.de/course/view.php?id=349
(see also univis)

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Mission

We
care
about
Performance!

- Optimization & Parallelization on all modern compute architectures

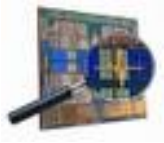
→Understand interaction between code & hardware

This is ours!

→Performance modelling: Roofline model & ECM model

→Performance tools:

likwid – lightweight performance tools
(https://github.com/RRZE-HPC/likwid)



kerncraft - Loop Kernel Analysis & Performance Modeling Toolkit
(https://github.com/RRZE-HPC/kerncraft)

- GHOST: General, Hybrid and Optimized Sparse Toolkit
(https://bitbucket.org/essex/ghost)
- ILBDC: Lattice Boltzmann Method based CFD solver

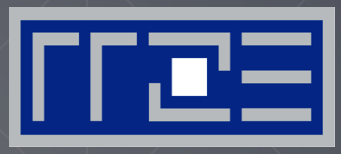
- We do performance optimization, performance modeling, parallelization for
  - Multi-core CPUs: core, socket, node and large scale 100,000+ cores
  - GPGPUs: single devices and cluster
  - Many-core CPUs: Intel Xeon Phi

- We collaborate with many users doing numerical simulation, e.g.:
  - Prof. Rüde: waLBerla / efficient C++; Prof. Fey; Prof. Teich
  - Chemistry, Physics
  - Engineering: Prof. Schwieger, PD Dr. S. Becker
  - Medical Image Reconstruction: Prof. Hornegger
  - Sparse Linear Solvers: Theoretical Physics, DLR
  - ….
- We operate the compute resources at FAU
- Our group:
  - 6 senior scientists (incl. RRZE) (GW/GH/TZ/MM/JE/MW)
  - 4 PhD students (FS/MK/TR/JH)
  - 2 Master student (DE/CA)

- RRZE Testcluster („Playground“)
  - 10-core Ivy Bridge – 14-core Haswell - 22-core Broadwell
  - nVIDIA & AMD GPUs & Intel Xeon Phi KNL

- RRZE production machines (Infiniband/Ominpath Interconnect):
  - **544 nodes Intel Ivy Bridge (2x 10 cores/node)** → **10.880 cores**
    - + 8 x (2 x Intel Ivy Bridge 10 core + 2 x NVIDIA K20 GPGPU)
    - + 8 x (2 x Intel Ivy Bridge 10 core + 2 x Intel Xeon Phi)
  - **740 nodes Intel „Broadwell“ (2x10 cores / node)** → **14.800 cores**
    - Will be released soo….

- Access to external machines
  - NIC Jülich: IBM BlueGene/Q 450.000+ cores (6 PFLOP/s)
  - LRZ Garching: Intel Cluster ((3.2+3.6) PFLOP/s; 230,000 cores)
  - HLR Stuttgart: CRAY XC 40 (7.4 PFLOP/s; 185,000 cores)
  - CSCS Lugano: CRAY XC30 (5,272 nodes with 1 CPU/1GPGPU each)



MuCoSim WS 2016/7

„old“ talks from last semester:

New projects: „Journal Club“

Projects from SS 2016 – 2nd talks from last semester

M. Hußnätter: Performance analysis of a list-based lattice-Boltzmann code

F. Hofmann: Analysis of Mantevo's HPCCG benchmark

D. Zint: Performance Analysis of the Molecular Dynamics SPEC Code

A. Bundle: Smith Waterman algorithm - Performance Analysis

T. Klein: Analysis of the 363.swim benchmark

G. Chotalia: Analysis of the Mantevo MiniMD benchmark

D. Arcrossito: Performance Analysis of SPEC's 350.md Benchmark on Xeon/Phi

Journal club - What to do?

Goal: A critical review of important papers in the field of High Performance Computing

1. Read the paper(s) assigned to you thoroughly
2. Discuss open questions with your advisor.
3. In collaboration with your advisor, perform programming and/or measurements of your own (depends on the topic)
4. Prepare slides with a summary of your paper(s) and experiments (one or two talks <30min each)
   - Concentrate on main insight, not on every little bit of data
   - If the paper contains BS, say so (and provide proof!)
5. Every seminar attendee should be able to understand the main findings of the paper(s) you present

Roofline and its roots (taken)

Roofline is a “simple” bottleneck-based performance model for loops on multicore CPUs

- D. Callahan et al., Estimating Interlock and Improving Balance for Pipelined Architectures. Journal of Parallel and Distributed Computing archive Volume 5 Issue 4, August 1988, Pages 334–358, DOI: 10.1016/0743-7315(88)90002-0
- Hockney, Roger W. and I. J. Curington. “f1/2: a Parameter to Characterize Memory and Communication Bottlenecks.” *Parallel Computing* 10 (1989): 277-286, DOI: 10.1016/0167-8191(89)90100-2
- S. Williams et al.: Roofline: an insightful visual performance model for multicore architectures. Communications of the ACM Volume 52 Issue 4, April 2009 Pages 65-76, DOI: 10.1145/1498765.1498785
- S. Williams et al.: Optimization of sparse matrix–vector multiplication on emerging multicore platforms. Parallel Computing 35 (2009) 178–194. DOI: 10.1016/j.parco.2008.12.006

Stencil optimizations

Stencil algorithms appear frequently in scientific computing. Optimization of those algorithms is the subject of intense research.

- Frigo et al.: Cache Oblivious Stencil Computations. Proceedings of the 19th annual international conference on Supercomputing, Pages 361-366. DOI: 10.1145/1088149.1088197
- D. Wonnacott: Achieving Scalable Locality with Time Skewing. International Journal of Parallel Programming, Vol. 30, No. 3, June 2002. DOI: 10.1023/A:1015460304860

Stencil DSLs (2 students)

Domain-specific languages (DSLs) are regarded by many as a solution to the optimization problem with stencil algorithms.

- Y. Tang et al.: The Pochoir Stencil Compiler. SPAA '11 Proceedings of the twenty-third annual ACM symposium on Parallelism in algorithms and architectures, p. 117-128. DOI: 10.1145/1989493.1989508
- Y. Tang et al.: Coding Stencil Computations Using the Pochoir Stencil-Specification Language. USENIX HotPar'11 May 26–27, 2011, Berkeley, California, USA
- U. Bondhugula et al.: Automatic Transformations for Communication-Minimized Parallelization and Locality Optimization in the Polyhedral Model International Conference on Compiler Construction (ETAPS CC), Apr 2008, Budapest, Hungary.
- U. Bondhugula et al.: A Practical Automatic Polyhedral Parallelizer and Locality Optimizer. ACM SIGPLAN Programming Languages Design and Implementation (PLDI), Jun 2008, Tucson, Arizona.

Energy modeling

Energy consumption of computer systems is a major concern. One approach is to set up analytic models about energy consumption of basic operations (data transfer, flops).

- J. Choi et al.: A roofline model of energy. Proc. IPDPS 2013, DOI: 10.1109/IPDPS.2013.77
- J. Choi et al.: Algorithmic Time, Energy, and Power on Candidate HPC Compute Building Blocks. Proc. IPDPS 2014, DOI: 10.1109/IPDPS.2014.54

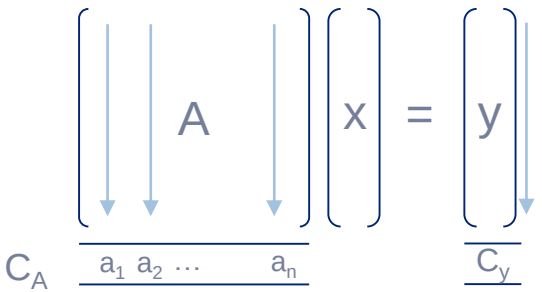
Modeling by fitting and machine learning

Analytic modeling can be complemented by machine learning approaches that automatically predict metrics (runtime, energy consumption) based on code features or hardware counter measurements.

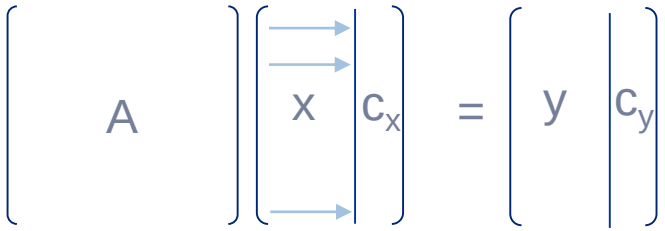
- A. Tiwari et al.: Modeling Power and Energy Usage of HPC Kernels. Proc. IPDPS 2012 Workshops, DOI: 10.1109/IPDPSW.2012.121
- J. Peraza et al.: Understanding the Performance of Stencil Computations on Intel's Xeon Phi. Proc. Cluster 2013, DOI: 10.1109/CLUSTER.2013.6702651
- A. Calotoiu et al.: Using Automated Performance Modeling to Find Scalability Bugs in Complex Codes. Proc. SC13, DOI: 10.1145/2503210.2503277

Soft-error detection methods for sparse linear algebra using checksum based techniques:

- Checksum based fault tolerance:



$C_A * x = C_y$



$C_y^i = \sum y^i$

- K.H. Huang, J. A. Abraham, *"Algorithm-Based Fault Tolerance for Matrix Operations"*, IEEE Transactions on Computers, 1984., (dense MMM)

- Longxiang Chen, Dingwen Tao, Panruo Wu and Zizhong Chen, *"Extending checksum-based ABFT to tolerate soft errors online in iterative methods"*, ICPADS 2014.

Co-Design

Co-Design means matching the requirements of algorithms with the resources that are provided by a computer architecture. These guys provide a mathematical method for doing this.

- K. Czechowski et al.: A theoretical framework for algorithm-architecture co-design. Proc. IPDPS 2013, DOI: 10.1109/IPDPS.2013.99

Benchmarking & best practices

This paper proposes twelve best practices for measuring and presenting performance data. It can be seen as a complement to the “Fooling the Masses” motif.

- T. Hoefler et al.: Scientific Benchmarking of Parallel Computing Systems. Proc. SC15, DOI: 10.1145/2807591.2807644

The “Unum” number format

“Unum” was proposed as a solution to the ubiquitous accuracy problem with floating-point representations.

- J. L. Gustafson: The End of Error: Unum Computing.
- J. L. Gustafson: A radical approach to computation with real numbers. DOI: 10.14529/jsfi160203

- See also http://johngustafson.net/unums.html