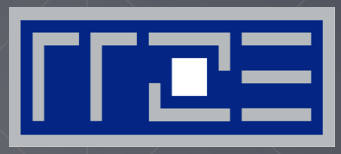
ERLANGEN REGIONAL COMPUTING CENTER



Efficient numerical simulation on multicore processors (MuCoSim) SS 2018

Prof. Gerhard Wellein, Dr. G. Hager, Dr. J. Eitzinger

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

http://moodle.rrze.uni-erlangen.de/course/view.php?id=376
(see also univis)

Mission

We
care
about
performance!

- Optimization & Parallelization on all modern compute architectures

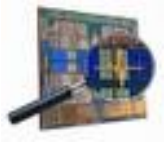
→Understand interaction between code & hardware

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

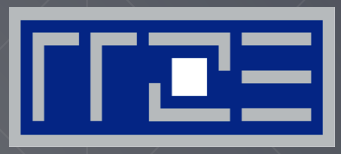
- We collaborate with many users doing numerical simulation, e.g.:
  - Prof. Rüde: waLBerla / efficient C++; hardware analysis: Prof. Fey
  - Chemistry, Physics Engineering
  - Sparse Linear Solvers: Theoretical Physics; DLR; Tokyo Univ.; USI Lugano
  - Code analysis/generation: H. Köstler / S. Hack
  - ….
- We operate the compute resources at FAU

- Our group:
  - 6 senior scientists (incl. RRZE) (GW/GH/TZ/MM/JE/MW)
  - 4 PhD students (FS/TR/JH/CLA/DE)
  - 2 Master/Bachelor students (MH / JL)

- RRZE Testcluster („Playground“)
  - 14-core Haswell / 22-core Broadwell / 20-core Skylake / 24-core AMD EPYC
  - 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 nodes with 2 x NVIDIA K20 + 8 nodes with 2 x Intel Xeon Phi / KNC**
  - **728 nodes Intel Broadwell (2x10 cores / node) → 14.560 cores**

- 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 XC40 (7.4 PFLOP/s; 185,000 cores)
  - CSCS Lugano: CRAY XC50 (4,500 nodes with 1 CPU+1NVIDIA P100)
  - Oakforrest-PACS (Japan): Intel KNL (8,000 nodes with KNL only)



MuCoSim SS 2018

Format

„Old“ talks/projects from previous semester;

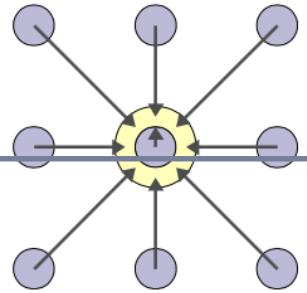
New projects:

- MiniApps from Mantevo

Projects from previous semesters

- L. Liebischer & C. Dev: ECM Modelling of Skylake
  2nd talk from WS17/8: ??.??.????
- J. Baumgärtner: Box stencils & Semi stencil optimization
  2 talks: ??.??.???? and ??.??.????
- T. Habermann: Basic stencils on NVIDIA GPGPUs
  2 talks: ??.??.???? and ??.??.????
- Other talks:
  - N. Dommaraju: Coarray Fortran Implementation of Himeno benchmark

Stencil topics – Box stencils & Semi-stencil



```
do k=1,kmax
  do j=1,jmax
    y(j,k) = x(j,k)+c10*x(j-1,k) +  c20*x(j+1,k) &
                   +c01*x(j,k-1) +  c02*x(j,k+1) &
                   +c11*x(j-1,k-1)+ c21*x(j+1,k-1)&
                   +c21*x(j-1,k+1)+ c22*x(j+1,k+1)
  enddo
enddo
```

Tasks:

- Implement OpenMP paralllel box stencil in 2D & 3D
- Roofline/ECM; modelling – spatial blocking?!
- Apply semi stencil optimization

Julia Baumgärtner

Advisor: GW / JH / CLA

ECM & performance of standard stencils on AMD Epyc and Intel Skylake SP (2 students)

Stengel et al., *Quantifying performance bottlenecks of stencil computations using the Execution-Cache-Memory model*. Proc. ICS15. DOI: 10.1145/2751205.2751240

Malas et al., *Multicore-optimized wavefront diamond blocking for optimizing stencil updates*. SIAM SISC 2015. DOI: 10.1137/140991133

Tasks:

- Implement a OpenMP parallel benchmark framework which tests all the stencils in the 2 papers
- Comprehensive benchmarking – architecture, scalability, problem size
- ECM/Roofline model – Validation using HW performance counter
- Spatial blocking/ Modelling & Validation

Chaitanya Dev & Lukas Liebischer

Advisor: GH / GW

Performance of standard stencils on NVIDIA GPU

Stengel et al., *Quantifying performance bottlenecks of stencil computations using the Execution-Cache-Memory model*. Proc. ICS15. DOI: 10.1145/2751205.2751240

Malas et al., *Multicore-optimized wavefront diamond blocking for optimizing stencil updates*. SIAM SISC 2015. DOI: 10.1137/140991133

Tasks:

- Implement selected stencils on NVIDIA GPU
- Comprehensive benchmarking
- Check layer conditions using HW performance counter
- Spatial blocking

Tim Habermann

Advisor: DE

Performance analysis of HPCG GS Smoother

The HPCG benchmark is an effort to accompany the LINPACK benchmark for characterizing HPC systems.
One of its main components is a Gauss-Seidel (GS) smoother.

Website
http://www.hpcg-benchmark.org

Tasks:

- Analyze the HPCG performance on recent Intel architectures
- Parallelize and SIMD vectorize the Gauss-Seidel smoother
- Replace the integrated GS Smoother using the RACE library

?

Srikrishna Jaganathan

Advisor: CA / JE

Node-level performance analysis of Mantevo benchmarks

- Mantevo: Slection of Mini-Apps published by M. Heroux et al. (Sandia, 2009)

- Applications: https://mantevo.org/about/applications/

- Tasks:
  - Install Benchmarks
  - Core-/socket-node level profile of all codes
  - Key Performance metrics for hotspots measured with likwid: Cache/Memory Bandwidth; Core ressources
  - Identify Hotspots which can be modeled with kerncraft

Advisor: JE

Task

You pick one benchmark from **MANTEVO**.

Your task:

- **Analyse** and **understand** what the code is doing
- Aquire a **runtime profile**
- Extract **computational kernels** and instrument them for HPM profiling with likwid-perfctr
- If applicable set up simple **performance model** for some of the computational kernels

Task cont.

- Perform a in-depth **HPM performance profiling** analysis
- Perform **benchmark variations** on some of the following test systems:
  - Intel Xeon IvyBridge-EP, Haswell-EP and Broadwell-EP
  - RRZE Emmy Cluster
  - Intel Xeon Phi or Nvidia TESLA
- Identify **performance bottlenecks** and propose **optimizations**
- Optional: Improve the performance of the benchmark codes

MANTEVO

- **pHPCCG / HPCCG** – Complete node level analysis including modelling
  - 800 lines, C++
  - Solver for unstructured grid problems

- **MiniMD** – Comparison Multi-core and GPGPU
  - App. 3000 lines, C++
  - Extracted from LAMMPS Molecular dynamics application code

- **MiniGhost** – MPI scalability study
  - 4000 lines, Fortran 77/90
  - Bulk Synchronous Parallel (BSP) model

MANTEVO cont.

- **`CloverLeaf3D`** – Node level analysis, OpenCL vs OpenACC
  - App. 9000 lines, Fortran 90
  - 3D Lagrangian-Eulerian hydrodynamics benchmark

- **`CoMD`** – Node level and scalability analysis
  - App. 3000 lines, C
- Mimics the application SPaSM (Scalable Parallel Short-range Molecular dynamics)

- **`miniTri`** – Node level and scalability analysis
  - App. 700/1500 lines, C++
  - Simple, triangle-based data analytics code.

Intel Pin and SDE

- Pin: Dynamic Binary Instrumentation Tool
  - Intercept assembly and insert code
    - Count specific instruction/function calls
    - Transform one instruction to another (AVX load to split SSE loads)
    - https://software.intel.com/en-us/articles/pintool/
- SDE: Software Development Emulator
  - Run code in a virtual architecture
  - Sets up on top of Pin
  - https://software.intel.com/en-us/articles/intel-software-development-emulator/
- Questions for the seminar: How much work is it to use Pin? Where are the limitations ? What's the overhead of running an application with Pin and SDE? How is the accuracy?

Advisor: TR

Tools to use from our group

General

- likwid-perfctr → Thomas Röhl
- kerncraft → Julian Hammer

Sparse solvers:

- CRAFT / PHIST → Faisal Shazad
- GHOST / PHIST → Dominik Ernst

Schedule

	MuCoSim	Intern
10.04.2018	G. Wellein – Intro	
17.04.2018	G. Hager – Performance Models	
24.04.2018	J. Eitzinger	Julian Hammer / Dana A
01.05.2018	*PUBLIC HOLIDAY*	3./4.5: A.: Julian Hornich – K
08.05.2018	T. Röhl / D. Uhl	
15.05.2018		
22.05.2018	*PUBLIC HOLIDAY ER*	24./25.5: A.: Dominik Ernst- K
29.05.2018		
05.06.2018		
12.06.2018		
19.06.2018		
26.06.2018		

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.

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

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

MuCoSim seminar

in

room 2.049