PeCoH – Performance Conscious HPC

Hinnerk Stüben

FEPA Workshop “Job-specific Performance Monitoring”
Regionales Rechenzentrum Erlangen (RRZE)
20 July 2017

PeCoH is supported by Deutsche Forschungsgemeinschaft (DFG) under grants LU 1335/12-1, OL 241/2-1, RI 1068/7-1.
Applicants

Professors at Department of Informatics at Universität Hamburg:

- Thomas Ludwig, *Scientific Computing*
- Stephan Olbrich, *Scientific Visualization and Parallel Processing*
- Matthias Riebisch, *Software Engineering and Construction Methods*
Partners

- Deutsches Klimarechenzentrum (DKRZ)
- Regionales Rechenzentrum der Universität Hamburg (RRZ)
- Rechenzentrum der Technischen Universität Hamburg (TUHH)
Topics

- development of a cost model
- transfer of HPC know-how
- HPC certification program (German: „HPC-Führerschein“)
- software engineering for HPC
Cost model

• goal
  – raise cost- and performance awareness

• based on resource usage
  – compute node usage
  – disk usage
  – electric power consumption

• give feedback to users
  – automated reports appended to batch log files

• prototype implementation
  – with the Slurm Workload Manager
Transfer of HPC know-how

- Hamburg HPC Competence Center (HHCC) → https://www.hhcc.uni-hamburg.de/
  - knowledge base
  - user support
  - HPC certification
  
- best practices for existing software packages
  - study tuning possibilities (input parameters, environment variables)
  - benchmarking
  - document recommendable settings
### HPC skills (I)

<table>
<thead>
<tr>
<th>Skill Level</th>
<th>HPC Knowledge</th>
<th>Use of the HPC Environment</th>
<th>Performance Engineering</th>
<th>Software Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Parallel Programs</td>
<td>Shared Memory and Distributed Memory Systems, Job Scheduling, File Systems, Network Bandwidth and Latency, Moore’s Law, Amdahl’s Law</td>
<td>Linux Command Line, Shell Scripts, Environment Modules, Workload Managers</td>
<td>Measuring System Performance, Benchmarking, Scaling Studies, Tuning via Runtime Options (e.g. for MPI and OpenMP), Process Mapping to Nodes, CPU Pinning</td>
<td>Automated Testing</td>
</tr>
</tbody>
</table>
### HPC skills (II)

<table>
<thead>
<tr>
<th>Skill Level</th>
<th>HPC Knowledge</th>
<th>Use of the HPC Environment</th>
<th>Performance Engineering</th>
<th>Software Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Parallel Programs (e.g. from Open Source Packages)</td>
<td>Architectures (SMP, NUMA, GPU, Many Core), Hybrid Approaches (e.g. CPU+GPU), Domain Decomposition, Load Balancing</td>
<td>Programming Environments (Compilers, Libraries, Linker, ...)</td>
<td>Package Options, Optimized Libraries, Compiler Options, Profile Guided Optimization</td>
<td>Computational Complexity, Portability for Job Scripts</td>
</tr>
</tbody>
</table>
## HPC skills (III)

<table>
<thead>
<tr>
<th>Skill Level</th>
<th>HPC Knowledge</th>
<th>Use of the HPC Environment</th>
<th>Performance Engineering</th>
<th>Software Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Develop Parallel Programs</strong></td>
<td>Pipelining, Vectorization, Performance-Bounds (CPU, Cache, Memory, I/O, Communication), Overheads for Communication, Synchronization and Redundant Computations, Multi Level Approaches (e.g. MPI+OpenMP)</td>
<td>Programming Environments (Debuggers, IDEs), Version- and Configuration Management-Tools</td>
<td>Profiling, Detecting Performance Bottlenecks, Tuning via Reprogramming (e.g. using Functional Units (Fused-Multiply-Add)), Vectorization, SIMD, GPUs, More Efficient Algorithms</td>
<td>Test-Driven Development, Object Oriented Development, Communication Patterns, Blocking and Non-Blocking I/O, Domain Decomposition Patterns</td>
</tr>
</tbody>
</table>
HPC competences

HPC Certification Program

Performance Engineering
- Measuring System Performance
- Tuning
- Cost Awareness
- Optimization Cycle

Software Engineering
- Efficient Algorithms and Data Structures
- Parallel Programming
- Object-Oriented Development
- Agile Methods

HPC Knowledge
- HPC Certification Program
  - Supercomputer
  - Hardware Architectures
  - Job Scheduling
  - Moore’s Law
  - Performance Frontiers
  - Amdahl’s Law
  - Level
  - Overheads
  - Domain Decomposition
  - Modelling Costs

Use of the HPC Environment
- Operating System
  - Resource Management
  - Programming Environments
  - Version- and Configuration Management
HPC certification program

- analysis and classification of HPC competences
- development of a program to check on these competences
- collection of teaching material
- online examination

→ looking for collaboration partners!

→ https://www.hhcc.uni-hamburg.de
→ Certification
→ subscribe to our mailing list
Software engineering for HPC

Impact of using software engineering on scientific productivity:

- efficient algorithms and data structures
- object oriented development
- agile software development, automated testing / test-driven development
- coding guidelines, refactoring
- version and configuration management
Status

- positions filled (starting March and July 2017)

- HHCC web page established → https://www.hhcc.uni-hamburg.de/

- poster presentation at *ISC High Performance 2017*