A Brief History of VI-HPS

Felix Wolf
TU Darmstadt
About 12 years ago...

- Major investments in supercomputing hardware
  - Commodity clusters and massively parallel processors
  - Unprecedented levels of “peak” performance

- Little investment in software
  - Software taken for granted
  - Reliance on growing HPC open-source community

- Parallelism accessible via low-level interfaces
  - Hard to learn and use
  - Performance behavior hard to understand and predict
Increasing awareness that hardware is not everything...

- Focus shifts to **productivity**
  - Metric: time to solution
- Beyond hardware
  - People
  - Software environment

**VI-HPS**

- Modeling
- Coding
- Debugging
- Optimization
- Execution
- Interpretation
- Refinement
Examples of productivity-related initiatives

<table>
<thead>
<tr>
<th>Since 1990s</th>
<th>DoD HPC Modernization Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>1998 – 2004</td>
<td>EU Working Group APART</td>
</tr>
<tr>
<td>2002 – 2010</td>
<td>DARPA High Productivity Computing Systems Program</td>
</tr>
</tbody>
</table>

- More to come such as
  - BMBF HPC program
  - DOE CScADS
  - EU FETHPC
  - PASC (CH)
  - JST CREST
  - DOE IDEAS
2006: Call for Helmholtz Virtual Institutes

- Funding instrument of the President of the Helmholtz Association of German Research Centers
  - Supplemented with up to €300K per year for three years
  - Main goal: expanding networks with universities
- Unite national and international competencies and resources
- Have their own management structure
- Qualify young researchers
Objectives

- Modeling
- Coding
- Debugging
- Optimization
- Execution
- Interpretation
Objectives

- Develop advanced HPC programming tools to improve the code quality
  - Remove errors
  - Increase performance
- Accelerate the development process
  - Make error detection and performance optimization not only better but also faster
- Tools must be easy to use
- Offer training & support
Proposal accepted end of 2006

Scalable & intuitive programming tools for performance analysis and error detection

Areas of expertise

- Trace generation
- Trace visualization
- Automatic trace analysis
- Error detection
- Performance optimization
- Hardware counter access
- End-user analysis of profile data

Applications

- Physics & chemistry
- Engineering
- Environmental sciences
- Biology & biomaterial sciences
- Medicine

Architectures

- Massively parallel
- Shared memory
- Clusters
- Vector
- Multicore

PROF. DR. FELIX WOLF, TU DARMSTADT
Inauguration – Jülich, July 4th 2007

Keynote **Doug Post**

*The Opportunities and Challenges of Computational Science and Engineering*

Chief Scientist, United States Department of Defense High Performance Computing Modernization Program
Mission components

Development of HPC programming tools

Integration of theses tools

Training

Academic workshops
## Tools

<table>
<thead>
<tr>
<th>Single Node Performance</th>
<th>Parallel Performance</th>
<th>Debugging &amp; Correctness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callgrind</td>
<td>Allinea MAP</td>
<td>Allinea DDT</td>
</tr>
<tr>
<td></td>
<td>Allinea Performance Reports</td>
<td>Archer</td>
</tr>
<tr>
<td></td>
<td>Dimemas</td>
<td>MUST</td>
</tr>
<tr>
<td></td>
<td>MAQAO</td>
<td>STAT</td>
</tr>
<tr>
<td></td>
<td>Open</td>
<td>SpeedShop</td>
</tr>
<tr>
<td>Modeling</td>
<td>Measurement</td>
<td>Visualization</td>
</tr>
<tr>
<td>Extra-P</td>
<td>PAPI</td>
<td>Cube</td>
</tr>
<tr>
<td></td>
<td>Score-P</td>
<td></td>
</tr>
</tbody>
</table>
Performance optimization examples

XNS CFD
RWTH Aachen
4-5x faster

QMC=CHEM Quantum chemistry
IRSAMC
> 3x faster

MAGMAfill CFD
MAGMASOFT® GmbH
25% faster

Yales2 CFD
CORIA
up to 2,8x faster

INDEED FEM
GNS mbH
30-40% faster

Polaris - molecular dynamics
CEA
50-70% faster

Illumination - particle simulation
Queen’s University
2x faster

AVBP CFD
CERFACS
~ 10% faster
## R&D projects

... with at least 2 partners from VI-HPS

<table>
<thead>
<tr>
<th>Project</th>
<th>Agency</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SILC</td>
<td>BMBF</td>
<td>2009 - 2011</td>
</tr>
<tr>
<td>PRIMA</td>
<td>DOE</td>
<td>2009 - 2013</td>
</tr>
<tr>
<td>TEXT</td>
<td>EU</td>
<td>2010 - 2012</td>
</tr>
<tr>
<td>H4H</td>
<td>ITEA</td>
<td>2010 - 2013</td>
</tr>
<tr>
<td>HOPSA</td>
<td>EU</td>
<td>2011 - 2013</td>
</tr>
<tr>
<td>ECS</td>
<td>G8</td>
<td>2011 - 2014</td>
</tr>
<tr>
<td>LMAC</td>
<td>BMBF</td>
<td>2011 - 2014</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project</th>
<th>Agency</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELP</td>
<td>BMBF</td>
<td>2013 – 2016</td>
</tr>
<tr>
<td>Catwalk</td>
<td>DFG</td>
<td>2013 - 2017</td>
</tr>
<tr>
<td>Score-E</td>
<td>BMBF</td>
<td>2013 - 2016</td>
</tr>
<tr>
<td>PRIMA-X</td>
<td>DOE</td>
<td>2014 - 2017</td>
</tr>
<tr>
<td>POP</td>
<td>EU</td>
<td>2015 - 2018</td>
</tr>
<tr>
<td>ProPE</td>
<td>DFG</td>
<td>2017 - 2019</td>
</tr>
</tbody>
</table>
Technologies and their integration

- KCACHEGRIND
- LWM2 / MAP / MPIP / O|SS / MAQAO / PR
- Automatic profile & trace analysis
- Visual trace analysis
- Hardware monitoring
- Debugging, error & anomaly detection
- Execution
- Optimization
- PAPI
- MUST / ARCHER
- DDT
- STAT
- SYSMON / SPINDLE / SIONLIB / OPENMPI
- TAU
- SCORE-P
- PERISCOPE
- SCALASCA
- VAMPIR
- PARAVER
- PTF / RUBIK / MAQAO
Score-P

- Vampir
- Scalasca
- CUBE
- TAU
- TAUdb
- Periscope

Event traces (OTF2)

Call-path profiles (CUBE4, TAU)

Hardware counter (PAPI, rusage, PERF, plugins)

Online interface

Instrumentation wrapper

Process-level parallelism (MPI, SHMEM)

Thread-level parallelism (OpenMP, Pthreads)

Accelerator-based parallelism (CUDA, OpenCL, OpenACC)

Source code instrumentation (Compiler, PDT, User)

Sampling interrupts (PAPI, PERF)

Application
MAQAO

MAQAO (Modular Assembly Quality Analyzer and Optimizer) is a performance analysis and optimization tool suite targeting binary applications (no recompilation). The main goal of MAQAO is to analyze binary codes and provide application developers with synthetic reports in order to help them optimize their code. The tool mixes both dynamic and static analyses based on the ability to reconstruct high level structures such as functions and loops from a binary application.

Another key feature of MAQAO is its extensibility. Users can easily write their own plugins thanks to an embedded scripting language (Lua) and an instrumentation language (MIL). These allow fast prototyping of new MAQAO tools.

Typical questions MAQAO helps to answer

- What is the time breakdown between I/O, MPI, OpenMP, PThreads in my application?
- Which functions and loops are the most time consuming?
- Are all my function hotspots consuming the same amount of time across all the processes/threads (load balancing)?
- How can I optimize a loop? Which performance factor may I gain?

Workflow

The first step consists in pinpointing the most time consuming functions and loops thanks to the profiler module. It provides users with a global understanding of where optimization efforts should be directed to. The Code Quality Analyzer module can then help users optimizing hot loops (innermost). Given a loop it generates a set of reports describing potential issues, an estimation of the potential gain and hints on how to fix them.

Platform support

Linux clusters (Intel X4 and Xeon Phi)

License

GNU Lesser General Public License (LGPL) v3

Web page

http://www.maqao.org

Contact

support@maqao.org

As presented in Figure 5, MAQAO features high level HTML outputs providing synthetic metrics. Figure 5(a) shows function and loop hotspots by relying on sampling. Figure 5(b) presents related code quality reports showing potential gains and corresponding hints (compiler flags, source code transformations). Figure 5(c) displays the MPI communication topology in an interactive force-driven graph, whereas Figure 5(d) presents communications as a matrix.

Figure 5: HTML outputs for various MAQAO modules.
# Membership

<table>
<thead>
<tr>
<th>Year</th>
<th>Member</th>
</tr>
</thead>
</table>
| 2007 | FZ Jülich  
       | RWTH Aachen  
       | TU Dresden  
       | University of Tennessee |
| 2008 | University of Stuttgart |
| 2009 | TU München |
| 2010 | GRS Aachen  
       | University of Oregon |
| 2011 | University of Versailles Saint-Quentin |
| 2012 | Barcelona Supercomputing Center  
       | Lawrence Livermore National Laboratory |
| 2014 | Allinea (today part of ARM) |
| 2015 | TU Darmstadt (replaces GRS Aachen) |
Virtual Institute – High Productivity Supercomputing

Bylaws

October 17th, 2013

1. Goals and Activities

1.1 The Virtual Institute – High Productivity Supercomputing (VI-HPS) is an international association of programming-tool builders for high-performance computing (HPC). Its goal is to improve the quality and accelerate the development process of complex simulation codes in science and engineering that are being designed to run on highly parallel computer systems.

1.2 To fulfill its goal, VI-HPS pursues the following activities:

1.2.1 Development of portable programming tools that assist programmers in diagnosing programming errors and optimizing the performance of their applications.

1.2.2 Integration of these tools.

1.2.3 Organization of training events designed to teach the application of these tools.

1.2.4 Organization of academic workshops to facilitate the exchange of ideas on tool development and to promote young scientists.
New website online in 2013

www.vi-hps.org
VI-HPS Training & Tuning Workshops

- **Goals**
  - Give an overview of the programming tools suite
  - Explain the functionality of individual tools
  - Teach how to use the tools effectively
  - Offer hands-on experience and expert assistance using tools
  - Receive feedback from users to guide future development
- For best results, bring & analyze/tune your own code(s)!

- **VI-HPS Tuning Workshop series**
  - Multi-day format, (local) HPC system(s), exercises + participant applications

- **VI-HPS Tutorials**
  - Single-day, notebook computer + Linux ISO/OVA, exercises only
  - Half-day, presentations with live demonstrations only
VI-HPS Tuning Workshop series

- 25 workshops
- 11 as part of PRACE
- 19 hosts in 9 countries
- Over 512 participants
- Contributions from all partners
Further training events

- 18 dedicated external training events using local HPC systems
  - CSCS, DiRAC, DKRZ, DLR, ECP, ENSIMAG, ETHZ, IT4I, KAUST, MSU, NCAR, NERSC
  - Over 243 participants, generally 2 to 3 days with hands-on work with own application code
- Numerous conf. tutorials and contributions to seasonal schools
  - Often with hands-on exercises using Live-ISO/OVA or remote HPC
  - ISC-HPC, SC, XSEDE, Cluster, EuroMPI, EuroPar, ICCS, IHPCSS, PRACE
Academic workshops

Target audience includes many young researchers (e.g., PhD students)

PROPER
- @Euro-Par (2008 - 2014)
- Focus on productivity and performance

ESPT
- @SC (since 2012)
- Focus on extreme-scale programming tools

Sponsor of Euro-Par 2013 conference
Outlook

Challenges

- Growing scale and heterogeneity
- New classes of applications
  - Big Data
  - Deep Learning
Thank You