Application performance measurement towards Exascale using Score-P and Vampir

Dr. Matthias S. Müller
Email: matthias.mueller@tu-dresden.de
Content

- Overview – Challenges towards Exascale
- Performance Monitoring with Score-P
- Performance Analysis with Vampir
- Conclusions & Wrap-Up
Challenges towards Exascale

Dr. Matthias S. Müller
Email: matthias.mueller@tu-dresden.de
HPC systems evolve; Tools need to adapt:

**Challenges**

- Paradigms
  - Co-Array Fortran
  - Heterogeneous/Hybrid
  - MPI + Accelerator + Threads
  - MPI + CUDA
  - PGAS + CUDA
  - MPI + Threads
  - MPI + PGAS?

**Heterogeneous/Hybrid**

**Fault Tolerance**
Towards Exascale...

**Scalability**

**Usability**
Make all this accessible!

**TOP 500 – Largest Machine**

**Year**

- 2001
- 2002
- 2003
- 2004
- 2005
- 2006
- 2007
- 2008
- 2009
- 2010
- 2011

- #Cores
  - 0
  - 200000
  - 400000
  - 600000
  - 800000

**ZIH**
Center for Information Services &
High Performance Computing

Matthias Müller
Performance Monitoring with Score-P

Dr. Matthias S. Müller
Email: matthias.mueller@tu-dresden.de
Score-P

- Community effort for a common performance tools infrastructure
  - Score-P instrumentation and measurement system
  - Common data formats OTF2 and CUBE4

- Save efforts for new features, scalability, support, training, porting, QM
- Improve tool interoperability and data exchange
- Single learning curve for users, single installation, single measurement run

- Currently works with Vampir, Scalasca, TAU, and Periscope

- Initial funding:
  - SILC and LMAC projects, German BMBF
  - PRIMA project, US DOE

- On-going Open Source project, open for new partners/tools

See http://www.score-p.org/
Score-P – Overview

 Tool ecosystem:

- Vampir
- Scalasca
- TAU
- Periscope
  - VampirTrace
  - EPILOG / CUBE
  - TAU native formats
  - Online measurement

Consequence:
- Each tool requires its own measurement system
- Tool combinations limited
- Higher tool development effort
Score-P – Overview

Tool ecosystem:
- Vampir
- Scalasca
- TAU
- Periscope

Score-P

Goals:
- Single measurement system
- Better learning curve
- More tool integration
- Reduced tool development costs
Tracing – Instrumentation

Data Sources

Source level
- Compiler
- Manual
- OpenMP with Opari
- Tau PDT

Binary level
- Dyninst

Runtime/Library
- MPI Correctness
- Pthreads
- NVIDIA CUDA
- OpenCL
- 3rd party libraries
- External counters
- Plugin counters
- Java tracing

Operating System
- Resource usage
- Memory allocation

Hardware
- Performance counters
- I/O
- CPU ID

MPI Profiling Interface
- MPI

Score-P

Application

1.0 ms

Performance Visualization
Score-P – State

- Preview versions since 2010
- First release December 2011
- Score-P will replace VampirTrace
- Feature migrations are ongoing:
  - CUDA support on its way
  - TAU PDT support available
  - OPARI2 (in Score-P) supersedes OPARI in VT
  - ...
- Vampir Tutorials will use Score-P starting mid 2012
Performance Analysis with Vampir

Dr. Matthias S. Müller
Email: matthias.mueller@tu-dresden.de
Technology

Multi-Core Program

Vampir Trace

Trace File (OTF)

CPU

1.0 ms

Vampir 7

VampirServer

Many-Core Program

CPU

Trace Bundle

CPU
Meeting the Challenges

- HPC systems evolve; tools need adapt:
  - Scalability
  - Paradigms
  - Heterogeneity
  - Fault tolerance
  - Usability

- Several extensions address these challenges
Meeting the Challenges – Long Tracing Runs (1)

- Tracing long running applications is hard
- Case:
  - Long running application (~50min)
  - Non-reproducible bad performance in iterations
  - Coarse-grained traces gave no insight
  - Fine-grained traces would not fit into memory
- Solution:
  - VampirTrace feature “Rewind”
  - Allows to discard parts of the trace buffer
  - Controlled via VampirTrace instrumentation API
Meeting the Challenges – Long Tracing Runs (2)

First and last iterations + 4 “slow” iterations
Meeting the Challenges – Long Tracing Runs (3)

Zoom into a slow iteration
Meeting the Challenges – Long Tracing Runs (4)

All except rank 20 need to wait

Can also be spotted in Radar

Process 20 total CPU cycles suddenly drop for > 6 seconds
Case:
- Particle in cell code on NVIDIA GPGPUs
- Multi-level parallelism: MPI + CUDA + pthreads
- Performance is a black box without tools

Solution:
- VampirTrace support for CUDA
- Close collaboration with NVIDIA
Meeting the Challenges – CUDA, PicOnGPU
Meeting the Challenges – CUDA, PicOnGPU
Meeting the Challenges – CUDA, PicOnGPU
Meeting the Challenges – CUDA, PicOnGPU
Goal:

- Support hybrid system and software architecture at 10 Petascale
- Show MPI and GPGPU programming
- Do full system performance profiling and tracing

Facts:

- Jaguar + Successor
- >220,000 cores
- 200,448 monitored MPI processes
- >20 Tera-bytes of performance data
- 21,515 VampirServer processes
Meeting the Challenges – Collaboration with ORNL
Meeting the Challenges – Collaboration with ORNL
Meeting the Challenges – Collaboration with ORNL

Zoom into Details
Work in Progress

Dr. Matthias S. Müller
Email: matthias.mueller@tu-dresden.de
Vampir Partial Loading

Selectable time interval to load, save analysis resources
Scalasca-Vampir Interoperation

- Scalasca’s CUBE GUI calls Vampir GUI
- Open worst instance of selected performance problem
  - Start Vampir or use existing instance and open trace
  - Zoom to time interval of interest
  - Control via DBus, allow multiple successive operations
- Requires ‘-i’ option for Scalasca (scout –i)
Scalasca-Vampir Interoperation

### Metric tree
- 98.99 Late Sender
- 4573.65 Time
- 4.81e7 Visits
- 1.46 Wait at Barrier
- 0.12 Barrier Completion

### Call tree
- 0.00 setup_btiode
- 0.00 lhsinit
- 0.00 exact_rhs
- 0.00 compute_buffer
- 0.00 adi
- 12.66 copy_faces
- 11.04 x_solve

### System tree
- - generic cluster
  - 13.45 n035
  - 20.85 n036
  - 18.53 n037
  - 10.49 n038

---

**Shows the most severe instance of pattern in trace browser**
Scalasca-Vampir Interoperation
## Roadmap

### June 2012
- Thinned out trace files (removal of similar/redundant tasks)
- Temporal pattern highlighting during tracing
- OpenMP 3.0 task support
- VampirServer: improved remote connections
- ScoreP becomes official measurement system for Vampir
- CUDA 4.1 support with native CUPTI based tracing

### November 2012
- Further MPI performance metrics in performance radar
- Support for HMPP Instrumentation API
- OTF2 extensions
- Temporal pattern based trace compression
- Score-P: Scalability, rewind, selective tracing

### Longer term
- In memory performance event data compression
- Alignment based trace comparison
- Improved correlation of event types.
- Scalability improvements
- Score-P: measurement of system background counters (I/O, energy, …)
- Vampir: Background counter view
- PGAS/SHMEM support