An Integrated HPC Performance Tool Suite

Philip J. Mucci

SC'07
Reno, Nevada
The State of Linux Performance Tools

• Linux kernel has no code to support profiling in production (non-root user) environments.*
  - Despite highly stable kernel patches being available for > 10 years on some platforms.
  - Patch deployment complicated for smaller users where support agreements preclude patching of the kernel.

• No major commercial Linux distribution contains anything beyond OProfile and Gprof.*
  - Gprof: requires recompilation, does not support threads
  - Oprofile: requires root privileges to use, does not allow sharing of the PMU resources

* except IA64, which has kernel support and pfmon
The State of Linux Performance Tools

• Vendors have developed some tools, but kept much of the private.
  - This is changing as more adopt standardized kernel support (Perfmon2) and middleware (libpfm and PAPI).

• Numerous quality open-source tools exist but lack real standardization:
  - Build system
  - Documentation
  - Installation
  - Usage semantics
  - Parallel run-time integration and interoperability
  - Distribution (no RPM's, Ebuilds, Debs...)
Our Tools Strategy

• Leverage best-of-breed Open Source tools.
  - Foster relationships with original authors.
  - Propagate changes back to public source trees.
• Provide a drill-down hierarchy driven by needs of the customers and the AE's.
• Uniform user interface and semantics.
  - Observe linux standards. (LSB and beyond)
• Develop value added extensions.
• Guarantee full interoperability.
• Contract expertise where appropriate.
Productizing Open Source Tools

• The model works really well for
  - Innovation
  - Evolution
  - Distribution
  - Support*
  - Standardization*

• The bad news?
  - Specialization
  - Documentation
  - Verification
  - Integration

• Focus resources where needed, drive that ‘last mile’.
Basic Needs of Users

• Application and System Characterization
  - Overall evaluation of performance
  - Identify specific components of interest
  - Black box monitoring

• Analysis and Optimization
  - Improve over baseline performance through changes to the code
  - Repeated experimentation and optimization passes
  - Analyze at full optimization levels

• Performance Development
  - Integration of robust performance evaluation into a code's lifecycle
  - Regular performance regression testing
Performance Criteria for the Tools Suite

• Quick and easy characterization of:
  - Hardware utilization (on and off-core)
  - Memory
  - I/O
  - Communication
  - Thread/Task load balance

• Detailed analysis using sampling
• Work on unmodified codes whenever possible
• Simple instrumentation
• Adv. instrumentation and tracing
• Trace-based visualization
• 'Expert-level' control of the PMU and perfmon2
SiCortex Performance Tools Stack

- Perfmon2 – Kernel support
- Libpfm – Counter programming
- Monitor – Library interposition
- Pfmon – Low level access
- Papi – Portable PMU infrastructure
- GPTL – Simple instrumentation
- MpiP – MPI profiling
- HPCToolkit – Statistical profiling
- TAU – Tuning and Analysis Utilities
- Vampir – Scalable trace visualization
- Epilog (Kojak) – Trace generation
Performance Experiment Tools

• Set of commands that provide the interface to the underlying performance monitoring tools.
  - All are based on Monitor and PAPI

• papiex, mpiex, ioex, hpceex, gptlex, tauex
  - Easy to use as /bin/time, on unmodified executables
  - Generate concise textual output where appropriate
  - Take the same arguments, except for tool-specific options
  - Standardized installation and documentation
Additional Software

• EPILOG
  - Trace library from the KOJAK suite

• OProfile
  - Ported to use the Perfmon2 kernel infrastructure

• Other quality software not included:
  - OpenSpeedShop: LANL
  - PerfSuite: NCSA
  - ParaVer: BSC
  - EXPERT/CUBE from KOJAK: Juelich
  - DynInstAPI: Wisconsin
  - Others...
Summary

• ~1.5 man-years of effort has produced a leading tool suite where none existed.
  - Open Source can truly mean standing on the shoulders of giants.

• Continued success and R.O.I gained by following through on the strategy.
  - Integration and cooperation lowers support cost
Acknowledgements

• Center for Information Services and HPC, Technische Universität Dresden, Germany.
• ParaTools, Inc.
• Innovative Computing Laboratory, University of Tennessee, Knoxville.
• Lawrence Livermore National Laboratory.
• HiPerSoft, Rice University.
• National Center for Atmospheric Research.
• Stefane Eranian of HP Laboratories.
• Tushar Mohan, Jim Rosinski, Peter Watkins of SiCortex.