
Performance Tools Working Group
August 1-2, 2007
Washington, DC

Chairs:

Dan Reed, Renaissance Computing Institute (RENCI)

Bernd Mohr, Research Centre Juelich



Thank You

- ➔ Thanks to the group for valuable insights
 - lots of interchange and good ideas
- ➔ We have tried to capture the key ideas
 - any errors are Bernd/Dan's



Our Charge: Performance Tools

➔ Topics

- analysis, modeling and optimization
- interactive and automatic approaches
- data management and instrumentation
- hardware and OS support
- visualization and presentation
- etc

➔ Current status

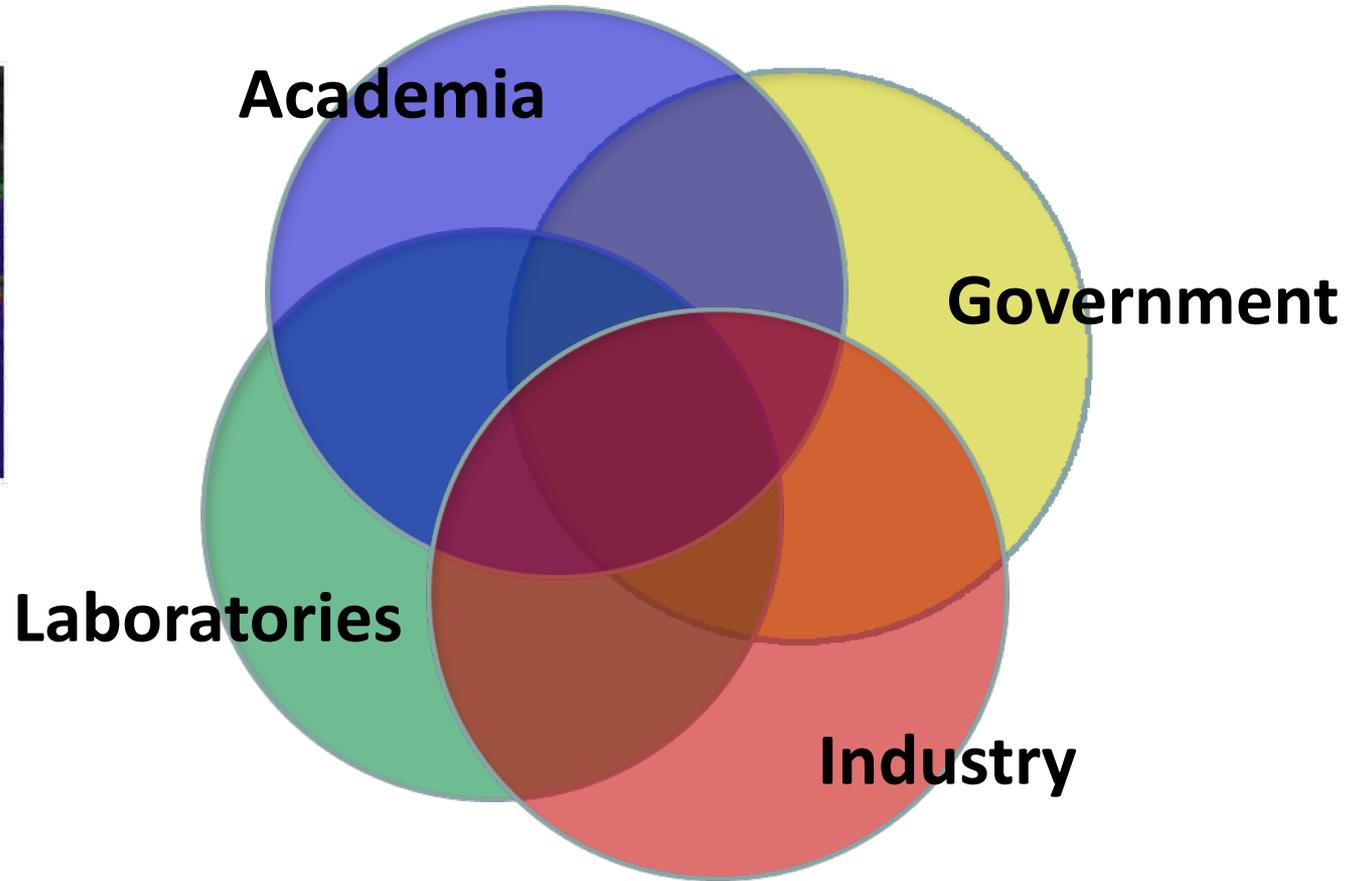
➔ Petascale requirements

➔ Findings

➔ Recommendations

- ordered priority list
- challenge type
 - technical, funding
 - policy, training
- impact
 - high, medium, low
- Probability (risk)
 - high, medium, low

Ecosystem Roles and Interactions



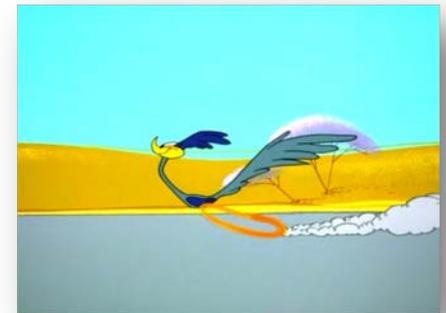
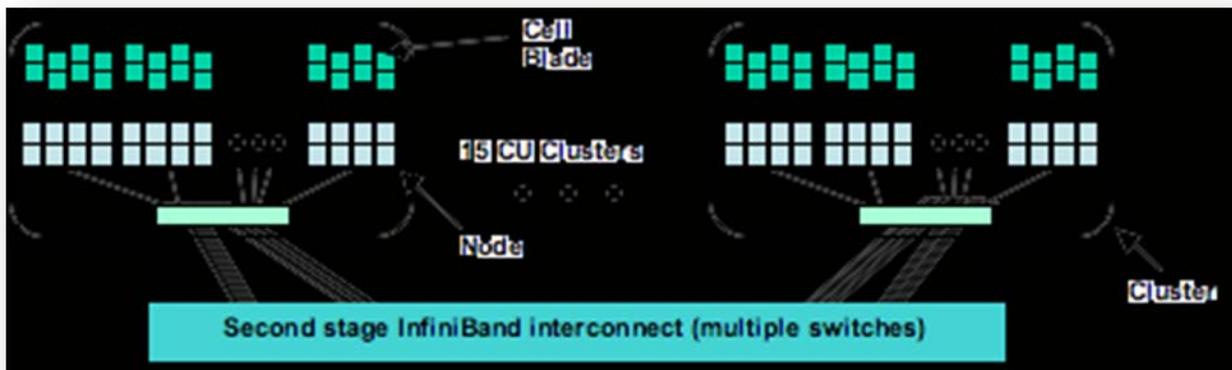
- ➔ Less disjoint than one might think
- small community with deep and long history

Performance Measurement: A Status Report

- ➔ Intelligent use well understood
 - instrumentation, measurement and analysis
- ➔ Instrumentation techniques
 - FORTRAN, C, less C++, ...
 - MPI, user functions/regions, less OpenMP, ...
- ➔ Measurement techniques
 - sampling, profiling and tracing
- ➔ Analysis techniques (weakest of the three)
 - too much data, not enough analysis
 - tools find symptoms, but not root problems

Performance Measurement: A Status Report

- ➔ Effective techniques for homogeneous systems
 - heterogeneous challenges coming
 - multicore, specialized processors, ...
- ➔ Too much concentration on *time as a metric*
 - need more support for memory analysis, ...
- ➔ Largely left to the user
 - true analysis, then optimization/tuning



Performance Modeling: A Status Report

- ➔ Limited breakout group discussion
 - not a reflection of lack of importance
- ➔ Multiple meanings/uses of modeling
 - system characterization
 - application prediction, ...
- ➔ Opinion
 - better than in the measurement community

Qualitative Status Assessment

- ➔ Measurement/analysis: WIP
 - ➔ Modeling: WIP
 - ➔ Optimization: NC
 - ➔ Interactive/manual: WIP
 - ➔ Automatic: WIP/NC
 - ➔ Data management: WIP
 - ➔ Instrumentation: WIP
 - ➔ Hardware and operating system support: WIP
 - ➔ Visualization/presentation: WIP
- ➔ *Legend*
- ➔ *Work in progress (WIP)*
 - ➔ *No clue (NC)*
 - ➔ *In hand (IH)*

Performance Tool Ecosystems



Petascale Requirements

- ➔ Increased automation
 - anomaly detection
 - correlation and clustering
 - data reduction
- ➔ Abstraction support
 - detail/complexity hiding
- ➔ Runtime adaptation
 - task topologies, ...
- ➔ Heterogeneity
 - programming models: explicit *and* implicit
 - hardware
- ➔ Hierarchy, including sharing



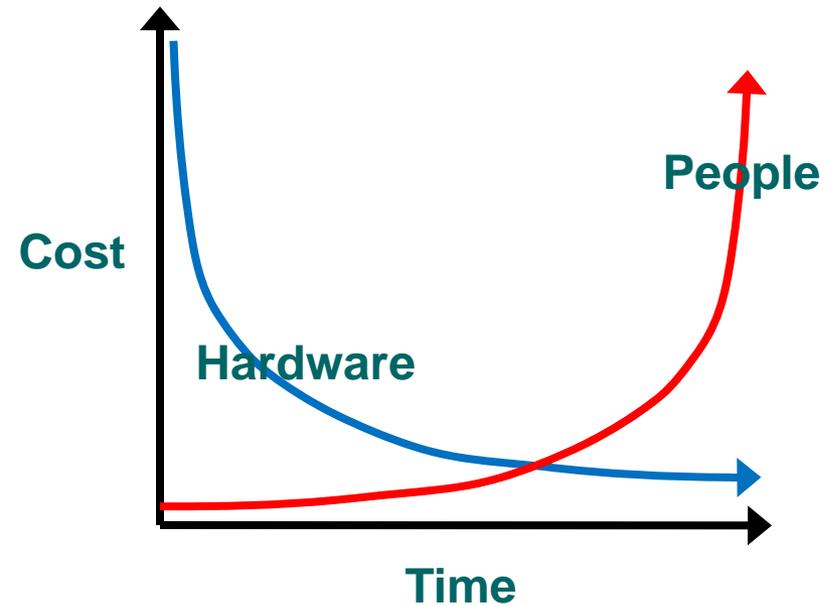
Petascale Requirements

- ➔ Fault tolerance/resilience
- ➔ Education and training
- ➔ Multi-level instrumentation
- ➔ Memory and I/O analysis
- ➔ Performability
 - hybrid/integrated performance and reliability
- ➔ Presentation and insight
 - scalable visualization
- ➔ Performance modeling and prediction
- ➔ Scaling of known methods and techniques
 - million-way parallelism and beyond



Economic Divergence/Optimization

- ➔ \$/teraflop-year
 - declining rapidly
- ➔ \$/developer-year
 - rising rapidly
- ➔ System complexity
 - rising
- ➔ Applications outlive systems
 - by many years
- ➔ Implications ...



Findings

- ➔ Petascale is *not* terascale scaled up
 - higher complexity, heterogeneity
- ➔ Petascale method inadequacies
 - manual methods
 - purely static and offline approaches
- ➔ Manual method needs
 - anomaly detection and optimization
- ➔ Purely static and offline methods
 - complement with online, adaptive methods



Findings

➔ Crucial interactions

- users/staff/developers critical
- education and training
- feedback

➔ Insufficient integration

- among tools
- component reuse

➔ No general pathway for release quality tools

- hardening, documentation, training, support, ...



Recommendations

#	Challenge	Type	Probability (Risk)	Impact
	User engagement and training	Training	High	High
	Additional information sources e.g. I/O, memory	Technical	High	Medium
	Long-term maintenance and support	Funding & policy	High	High
	Funds for technology transfer and deployment	Funding & policy	Medium	Medium
	Application-driven development of tools		Medium	Medium
	Substantial advances in automation of diagnosis, optimization and anomaly detection	Technical	High	High
	Developing live techniques to extend post-mortem	Technical	Medium	Medium

Recommendations (Continued)

#	Challenge	Type	Probability (Risk)	Impact
	Integrated, persistent monitoring components	Technical	Medium	Medium
	Support for multi-component and multi-disciplinary applications	Technical	High	Medium
	Detection of load imbalance	Technical	High	High
	Support for heterogeneous and hierarchical hardware	Technical	High	High
	Support for new and hybrid programming models	Technical, funding, policy & training	Medium	Medium
	Add performance analysis to CS curriculum	Training & policy	Low	Low