Performance Optimization

Philip J. Mucci
Minimal Metrics

phil@minimalmetrics.com

04/17/2012

Overview

- Background
- Customers
- Observations
  - Expertise
  - Programming models
  - Cloud, Virtualization and HPC
  - Tooling
  - I/O
  - Challenges for modeling

About Me

- Consulting since 1997
  - Software and hardware performance in HPC
  - HPC system software design
  - Parallel algo. opt.
- MS from UTK at ICL under Jack Dongarra
  - PVM, PAPI
- Research Consultant since 1998.
- Software architect, BD and app. Engineering at SiCortex.
- Founded Samara Technology Group in 2008.
  - Application and hardware performance experts for hire.
- Founded Minimal Metrics in 2012.

About Minimal Metrics

- A deep network of *the experts*.
- Evaluation, optimization and software engineering.
- Architectural evaluation.
- Moving into small-scale strategic consulting.
  - Logistic and process optimization.
  - Data collection and analytics and forecasting.
- Cofounded with Tushar Mohan

Reasons For Services

Our favorites:

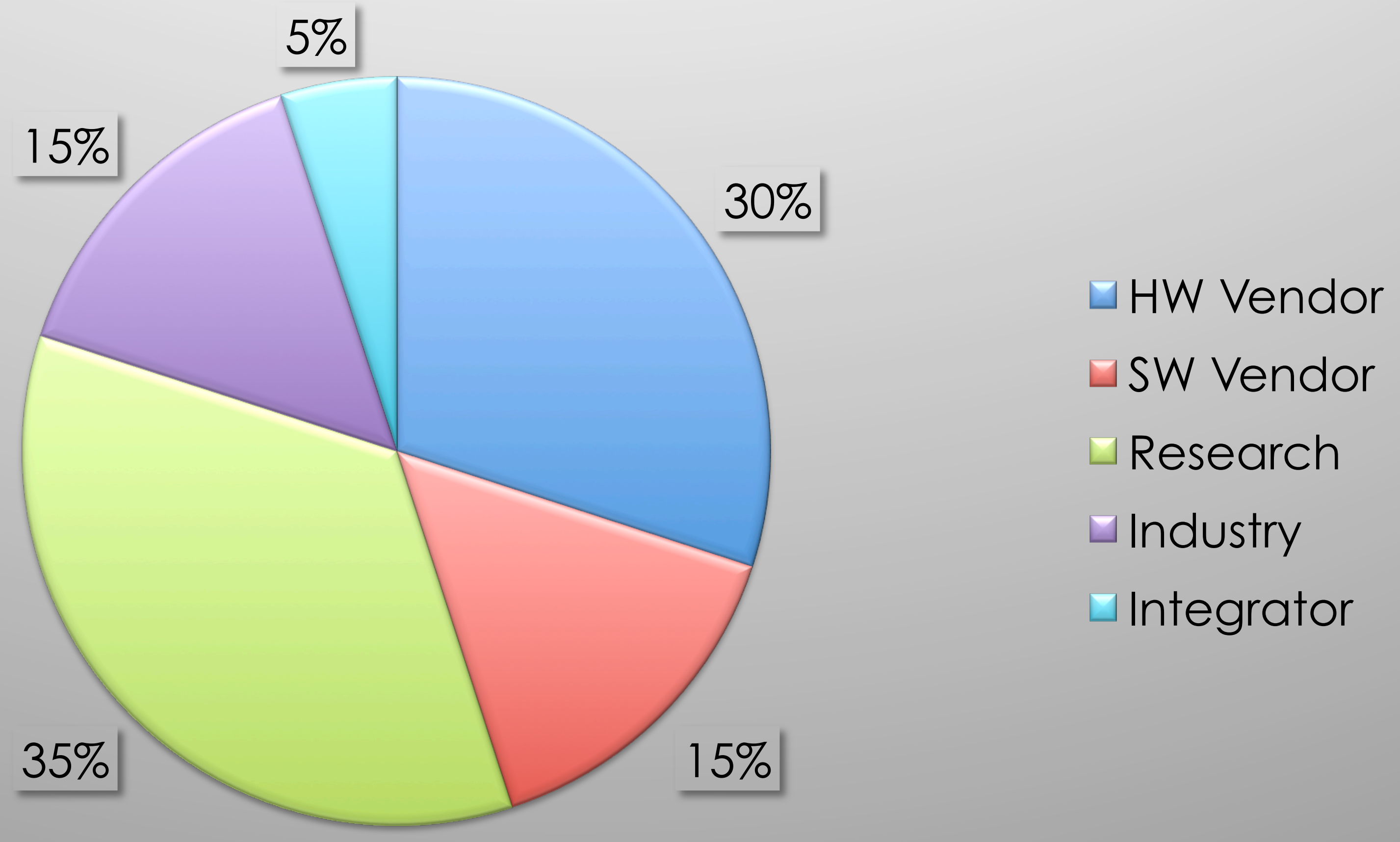
- We want to *understand* the performance of ______.
  - "Predictive vs. Reactive"
- We want to implement ______ using the most experienced talent [that we can't hire].

"The Cleaner" – La Femme Nikita, 1990

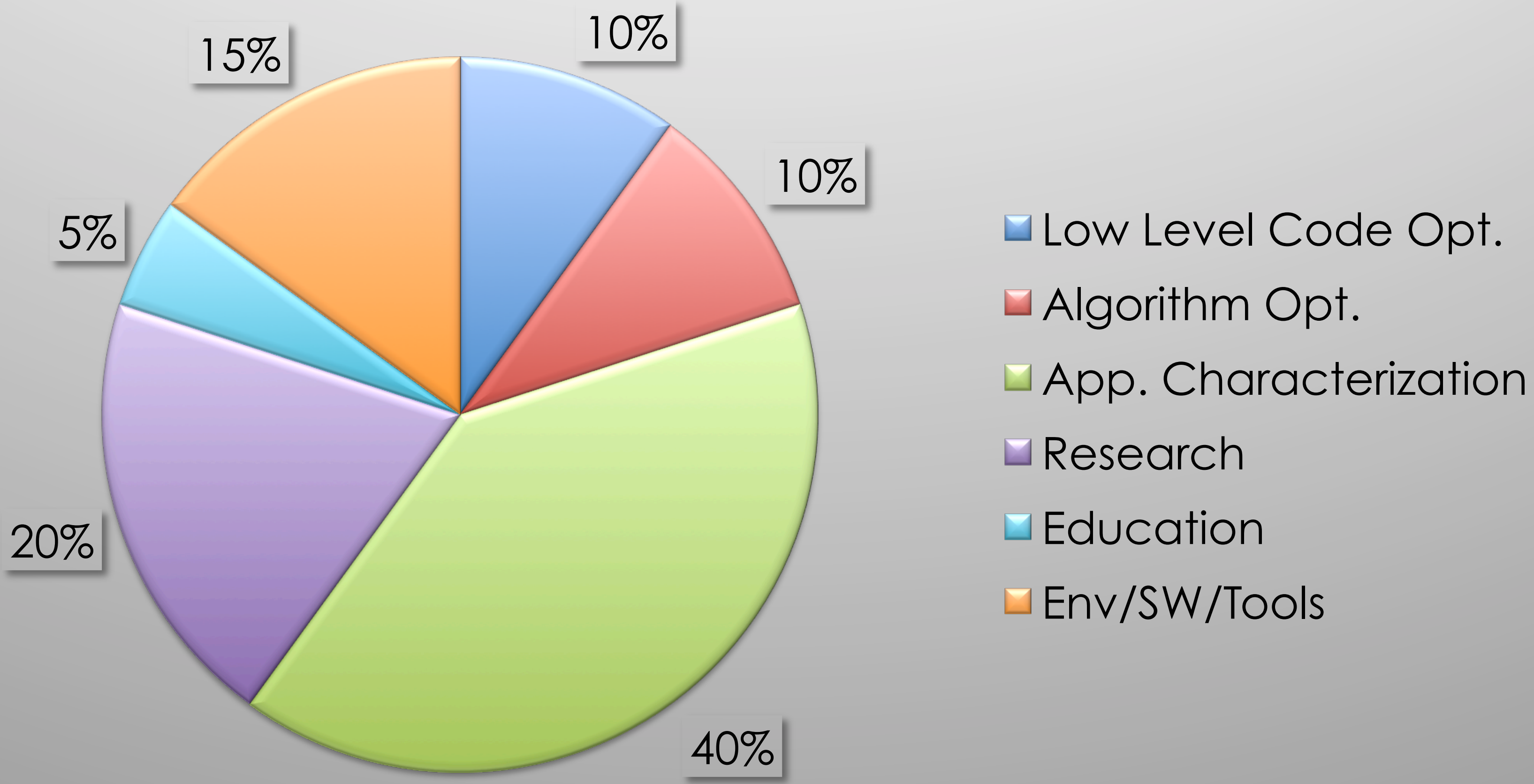
Some are preventable:

- We implemented our own ______ solver.
- We wrote this using ______ for ease of maintenance.
- We decided that ______ was the right technology to use.

Services by Customer



Services by Type



Expertise

- Lack throughout industry
  - Concentrated in specific verticals, with technology vendors and in research.
- Hiring for a critical task can be impossible.
  - Outsourcing is viable and cost effective.
- Educational curriculum catching up, but latency is long.

Sample qualifications from a job posting:

- Minimum 10 years related experience in a large scale R&D HPC environment.
- Expert knowledge using parallel programming techniques (e.g. MPI, OpenMP, pthreads), parallel programming languages (e.g., C, C++, F90) and scientific simulation and/or data analysis.
- Experience with parallel file systems, common data formats like NetCDF and HDF5, high-performance networking and storage systems.

Parallel Programming Models

- We often propose (limited) library and directive-based programming.
    - Code can be easily reduced, verified and retargeted.
- Low-level technology adopted relatively quickly.
    - A bit of buyers remorse.
- Many abstractions come at a cost:
    - But good compute is available through native methods.
    - Robust data movement remains a bottleneck.
        - Between SW and HW components.
    - Limited tooling support
        - Analysis pipeline can require code transformation.

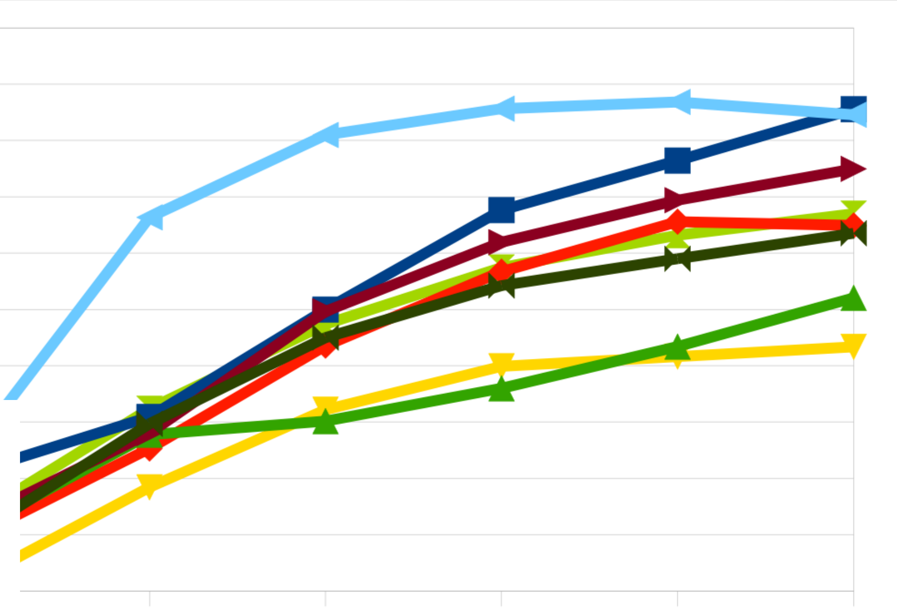
Cloud Technology, HPC & Grid Reflux

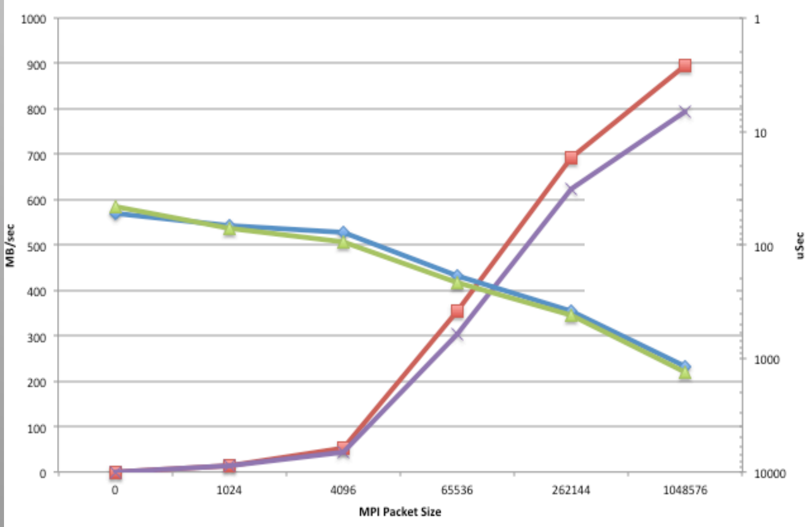
- Options exist for turnkey HPC cloud-based environments.
  - Yet plenty of integration work remains for HPC.
- Single node performance near parity, including decent I/O.
- Communication's (and thus parallel) performance getting there:
  - Per-core network bandwidth is limited.
  - Lack of low-latency, high-bandwidth comm. capability through the VM.

% MPI Time

Optimization and Virtualization

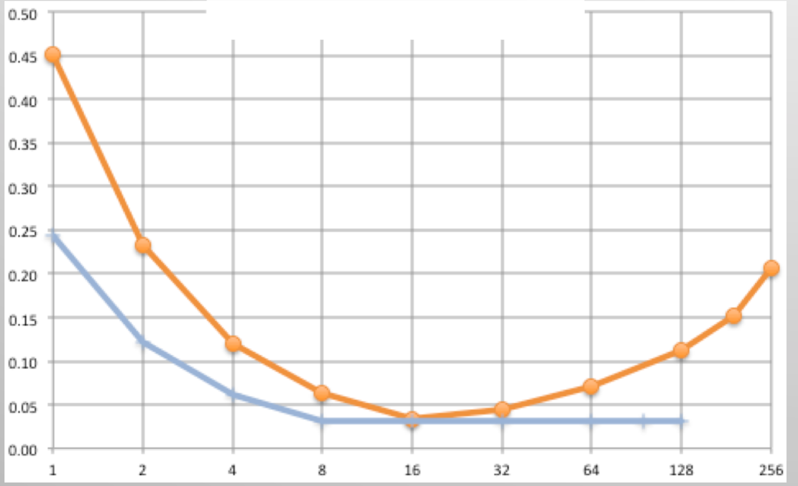
- Largely environmental.
  - OS and software stack
  - I/O and MPI
- Intra-node MPI still quite good.
- Not so with off-node

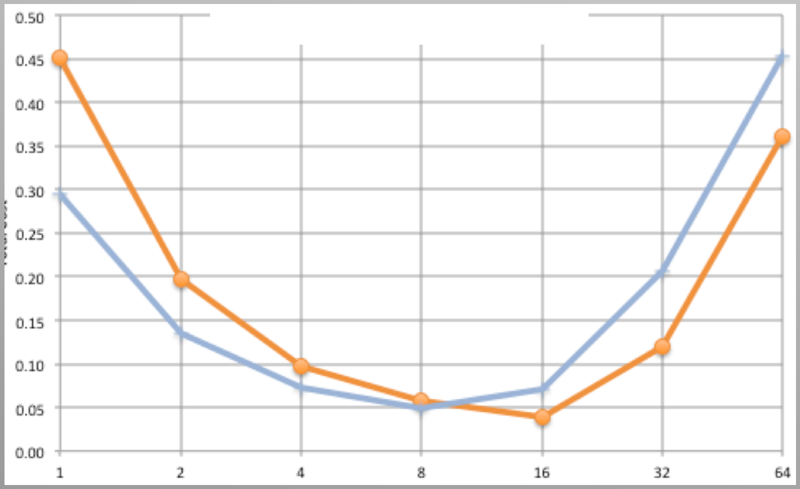




- Scaling is lost due to high MPI latencies for un-accelerated comm. in VM's for MPI

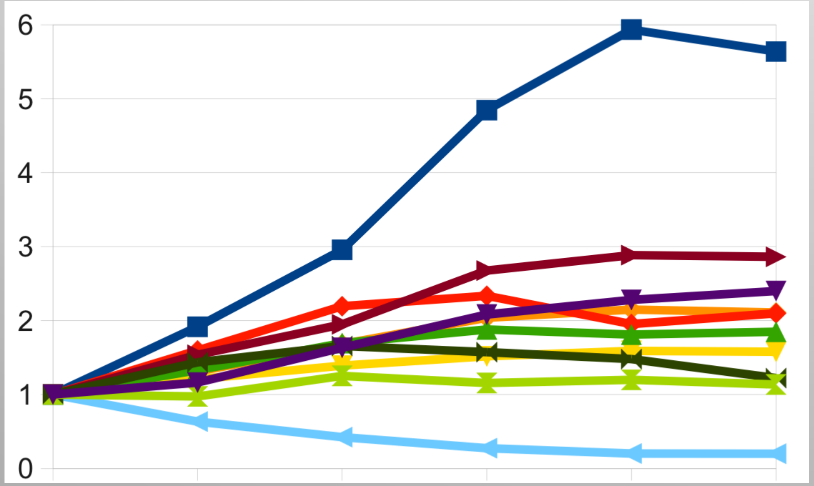
Economics of HPC in the Cloud





- Pricing requires very good scaling to be cost effective.
- Scaling's worth is related to the importance of the problem.

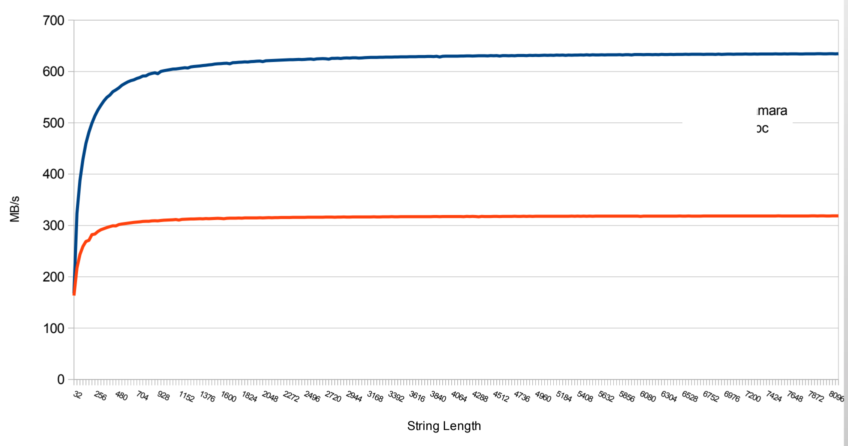
Optimizing ISV Applications

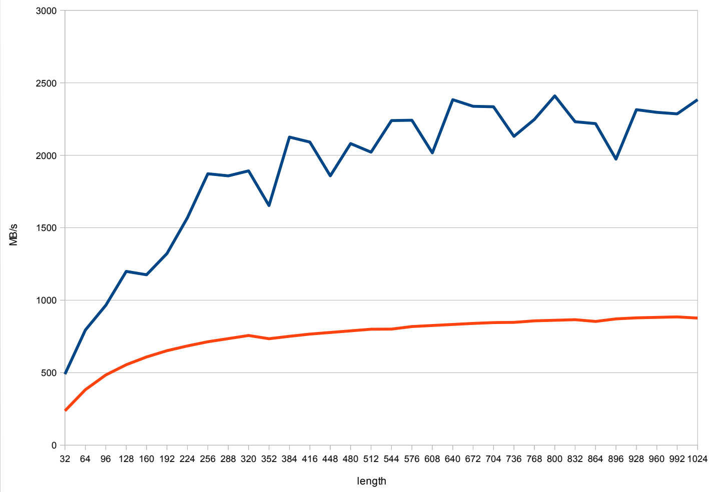


ISV scaling well below state-of-the-art

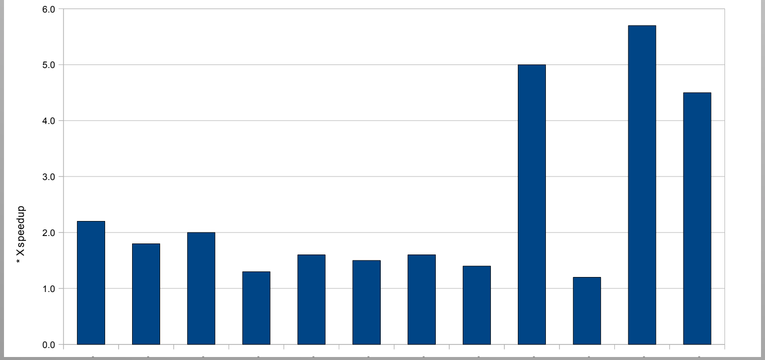
- Code is immutable.
  - And rarely changes.
- System optimization.
  - Uptime
  - System configuration
  - Libraries*
  - Parallel run-time
  - Storage
  - CPU availability
- 80/20 rule.

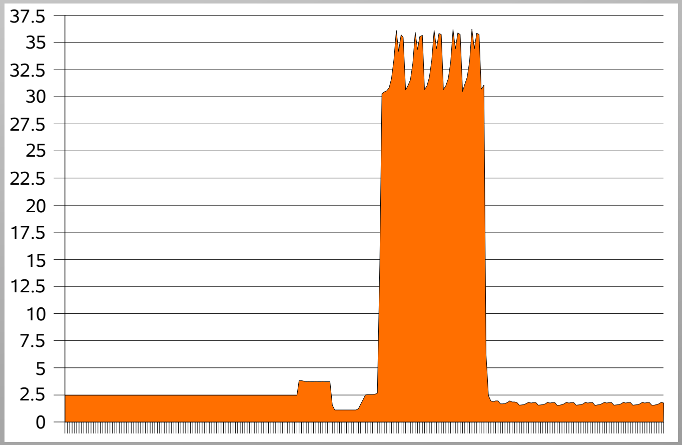
Performance of GNU/Linux





- Ain’t what you think it is for emerging architectures.

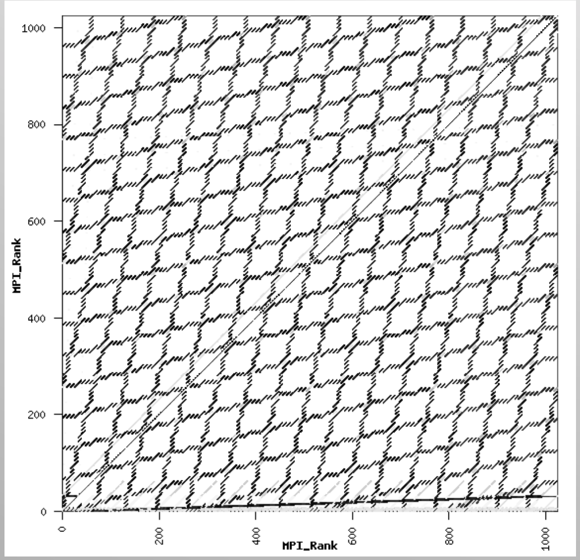




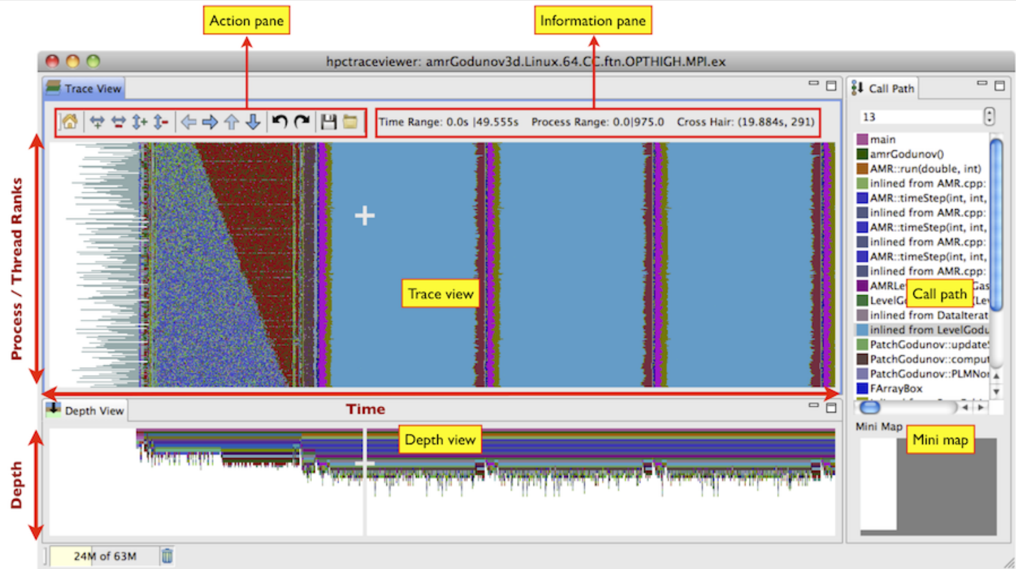
Optimization Tools

- Some excellent commercial and open source tools now available.
  - Some require far more knowledge than others to be effective.
  - Tools for MPI, OpenMP, I/O, GPUs and processors down to the instruction level.
    - Many now include *time* as a dimension of measurement.
  - Focus is on bottom-up view: explain global performance through local observations.
  - Much more robust collection, visualization (and some prediction) capabilities.
- Still lacking full job performance accounting.

Advanced Performance Visualization



IPM
Point to point data flow


hpctraceviewer: amrGodunov3d.Linux.64.CC.ftn.OPTHIGH.MPI.ex

Time Range: 0.0s |49.555s Process Range: 0.0|975.0 Cross Hair: (19.884s, 291)

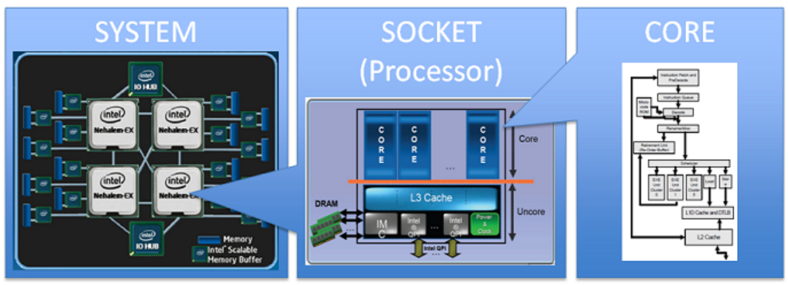
HPCToolkit
Metric vs. Task vs. Function (and Depth)

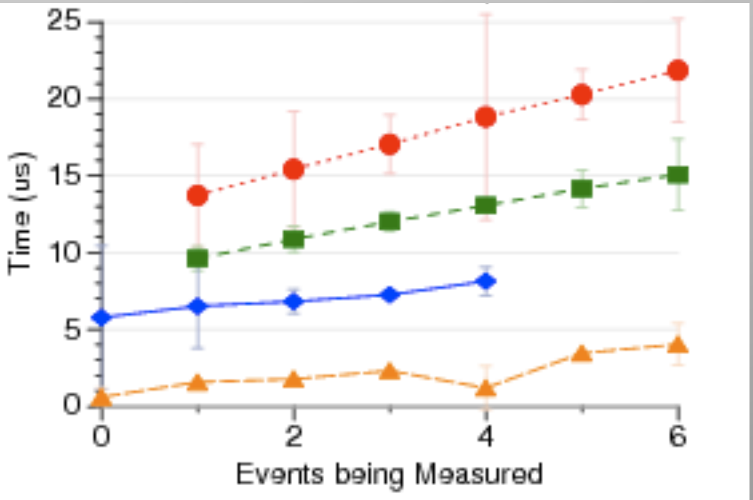
Tools Workflow

- Naïve methods:
  - Instrument and recompile.
    - But changes characteristics of original code.
  - Measure time only.
    - But answers “where”, but not “why” and “by how much”
- Methods now are largely passive and in-situ.
  - Instrumentation is inserted in binary form at run-time.
    - Or by the compiler with knowledge that this code is special.
  - Measure application, operating system and hardware performance events **that are relevant and actionable**.
  - Do so with minimal intrusion.

Performance Monitoring

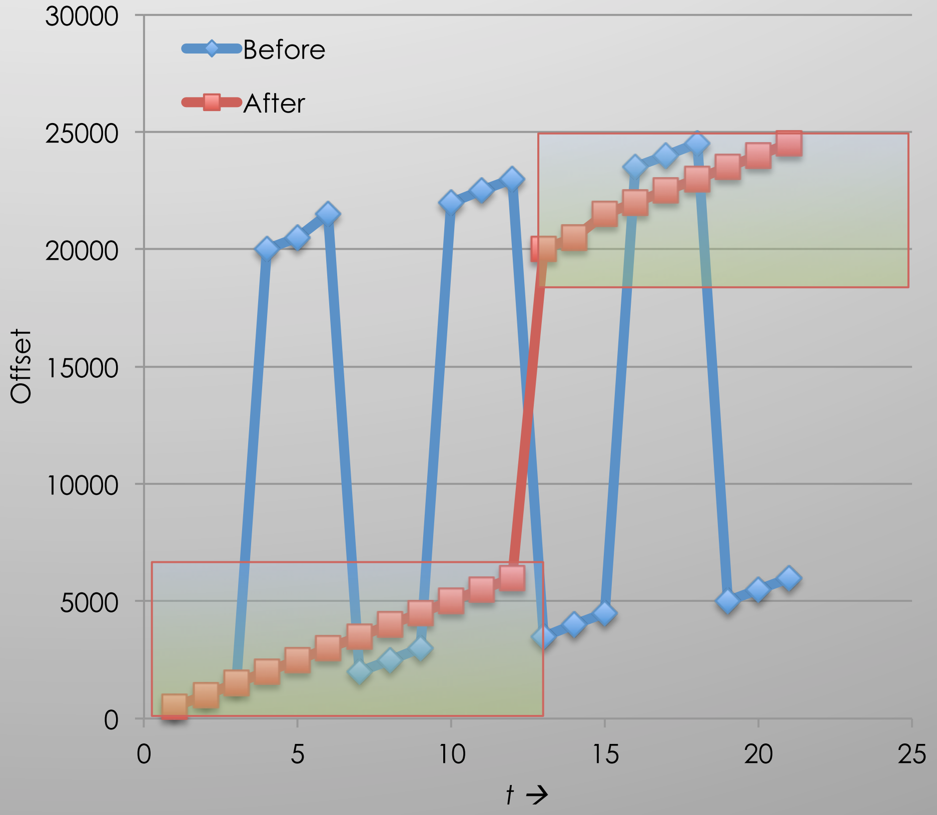
- Hardware PMU's
  - Logic capable of counting and sampling events of interest.
  - Now both on and off-core and in many devices.
- Software
  - System events with significant performance penalties.
- OS support maturing slowly, often regressing.
  - Low-latency, non-privileged access.
- Access often accomplished through PAPI.
  - Only as good as OS support.





Tools for I/O

- I/O bounds abound.
- Lack of bottom-up tooling.
  - System-level tools provide device level statistics.
  - Good for capacity & fault diagnosis, not tuning.



Access reordering and caching

Challenges Related to Modeling

- How will my application run on a new platform?
- Anything other than a kernel is non-trivial.
  - HW monitoring and tools allow us to precisely analyze and predict execution traces, not arbitrary code segments.
- Application performance is now largely data-set dependent.
  - Problems are often irregular and/or sparse.
  - Algorithms may be highly configurable.
  - Convergence criteria may be different.
- The data-set needs to be part of the input vector for any model.
- PMaC @ SDSC reflects the state of the art.

Software Systems

- Software and knowledge are well behind exploiting what the hardware is capable of.
- Quotes from this morning:
  - "Software hurdles are rising to the top for most users"
  - "Software leadership will become the new battleground"
  - "HPC experts often have a narrow view of a new applied user world"
  - "We require ease of everything and just want it to work"
  - "[Engineers become] too hyped about the tools and not about the problem being solved."

Thanks Dad.

- **John Francis Mucci**
  - 5/19/1942 – 2/7/2010
- From Ridgway, PA
  - PhD in High Energy Physics from Carnegie Mellon.
- Career
  - Director GSG @ Digital
  - VP of Sales, Marketing and Technical Research at Thinking Machines
  - Cofounder and CEO of Topical Net, Links2Go, Continuum Software and SiCortex
- Married Patricia A. Mucci in 1967. Two sons, Philip and David.

Thank You

phil@minimalmetrics.com