A Case for More Intuitive Performance Analysis

Martin Schulz

Lawrence Livermore National Laboratory

Salishan ◆ April 25–28, 2011

LLNL-PRES-481656

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Complexity is on the Rise

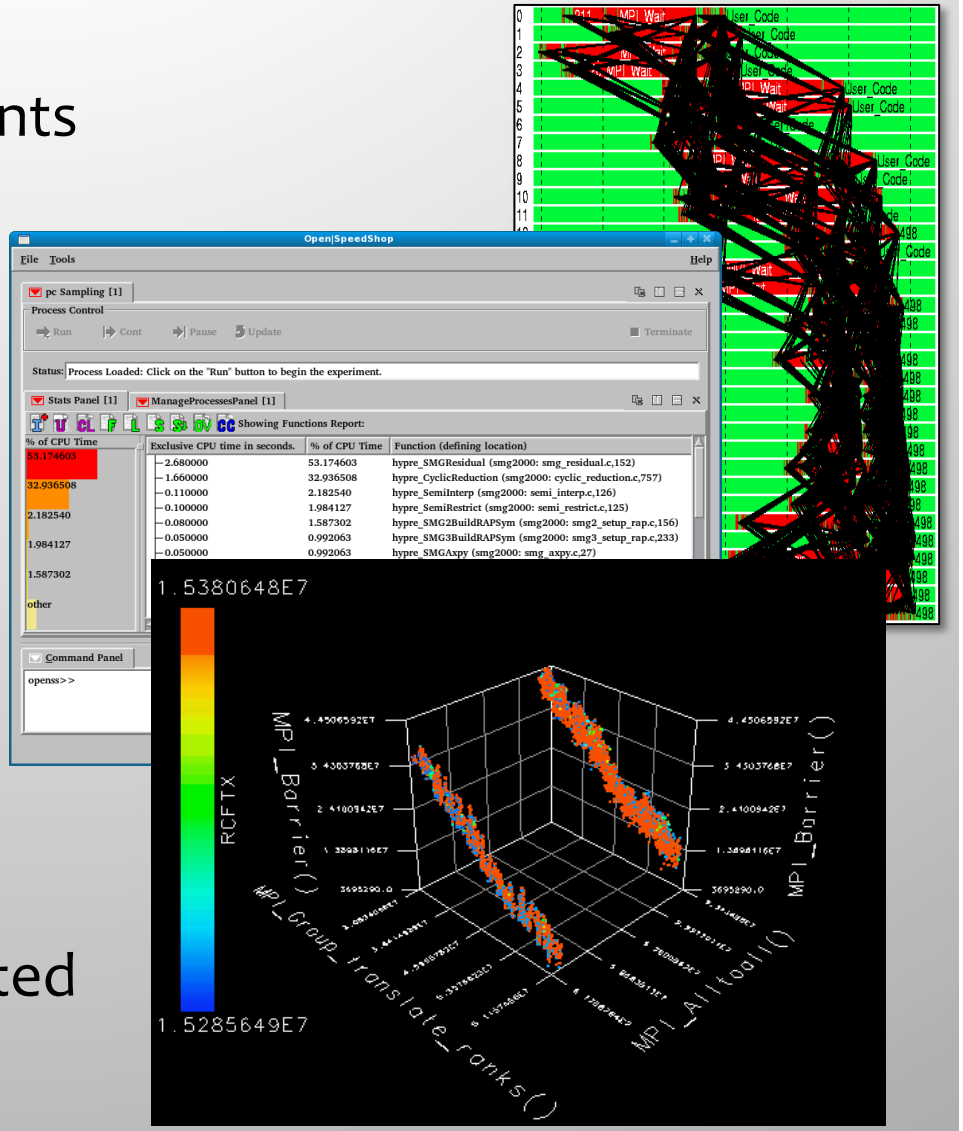
- Architectures are getting more complex
  - Huge process and/or thread counts
  - High dimensional network topologies
- More constraints
  - Less memory per core
  - Power limitations
- Applications are getting more complex
  - Multiphysics/Multiscale codes
  - Integration of UQ
- Complex interactions that are not well understood

Complexity Directly Impacts Programmers

- Programmers need a new way to program at Exascale
  - … and need to do so efficiently
- It will be a challenge to achieve efficiency
  - Load balance will be key at billions of threads
  - Manual adaptation to architectures/swim lanes may be necessary
  - Memory and network architectures will require layout optimizations
- Definition of efficiency needs to be revisited
  - Heterogeneous systems/nodes/chips/units/…
  - Multiple optimization targets (power vs. reliability vs. memory vs. speed)
  - Self-adaptive systems at all layers
  - Baselines are no longer obvious
- Too many factors are not accessible -> “Hidden Costs”
  - Costs that we cannot measure (yet)
  - Costs that we can measure, but we can not interpret (yet)

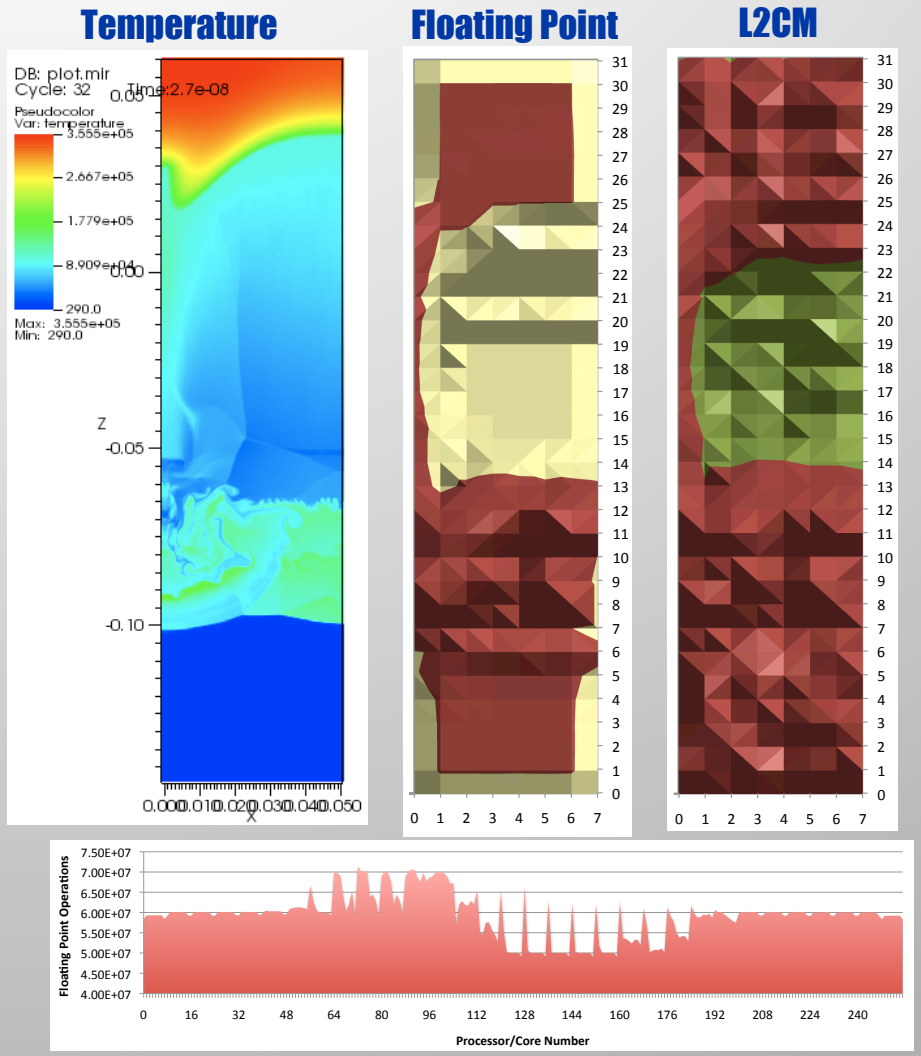
Existing Tools Fail to Provide the Right Perspective

- Large variety of measurements
  - Sampling/Tracing
  - Timings/Counters
- Attribution to source code
  - Static: Debug information
  - Dynamic: Stack traces
- Limited perspectives
  - Typically MPI rank space
  - Aggregated (phase based) or timeline diagrams
- Some tools allow sophisticated multi-metric visualizations



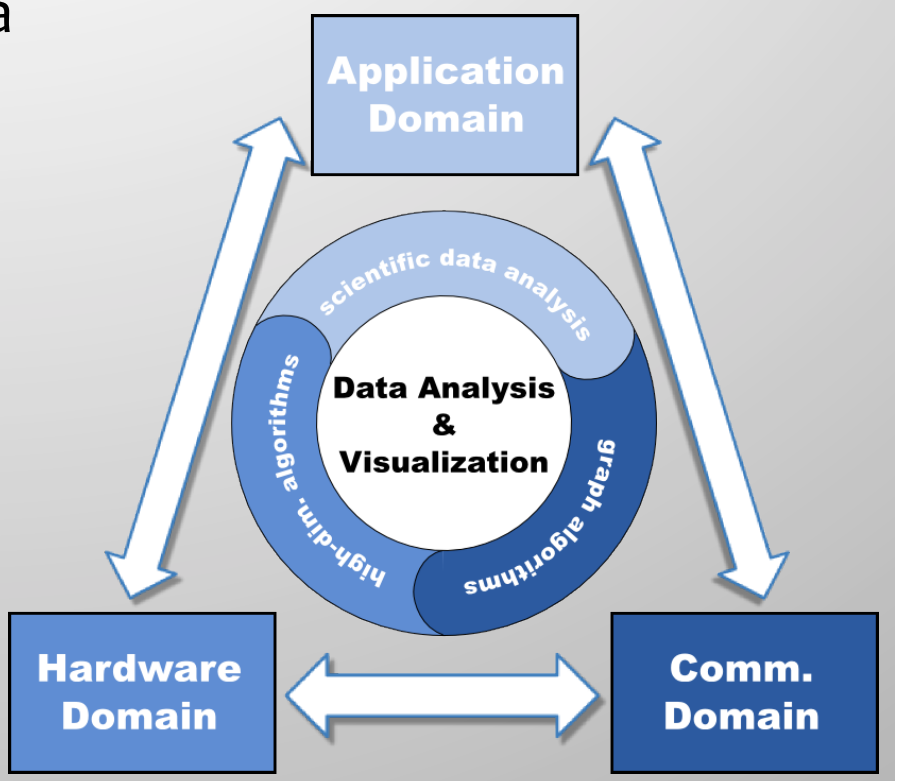
Taking the Application Developer's Perspective

- Example: 256 core run of a CFD application
  - Floating point operations
- Application developers think in the app domain
- Simple step:
  - Map floating point ops onto the application domain
  - Similar L2 cache misses
- Clear correlations
  - Explains performance
  - Helps establish a baseline



A New Generation of Tools to Expose Hidden Costs

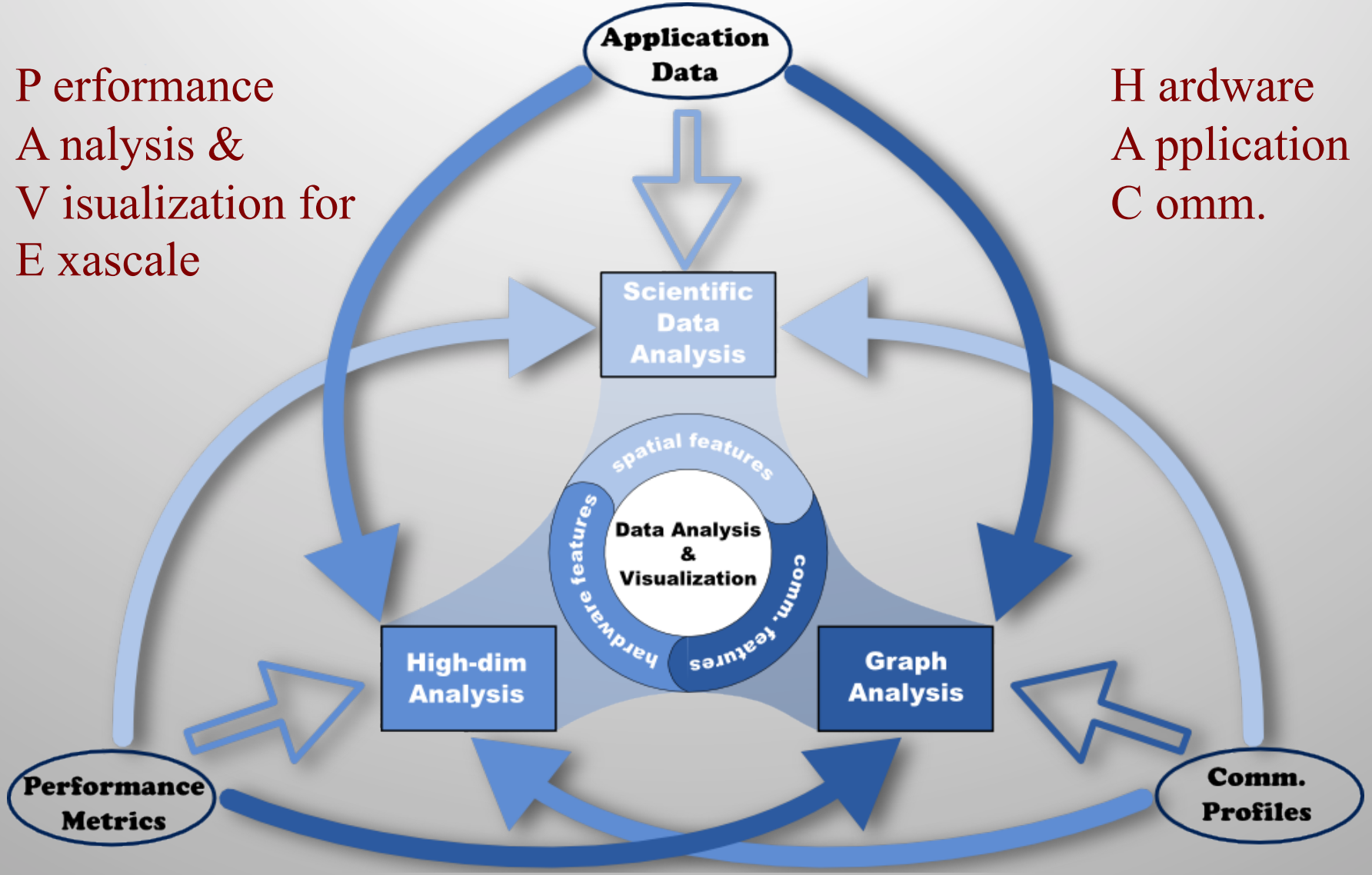
- Tools must consider several domains / perspectives on data
  - Application domain
  - Hardware domain
  - Communication domain
- Visualize in the domains
- Inter domain mappings
  - Enable new perspectives
  - Analysis across domains
  - Use data analysis techniques
- Forming a bridge between Performance Analysis and Data Analysis/Visualization



The PAVE/HAC Model

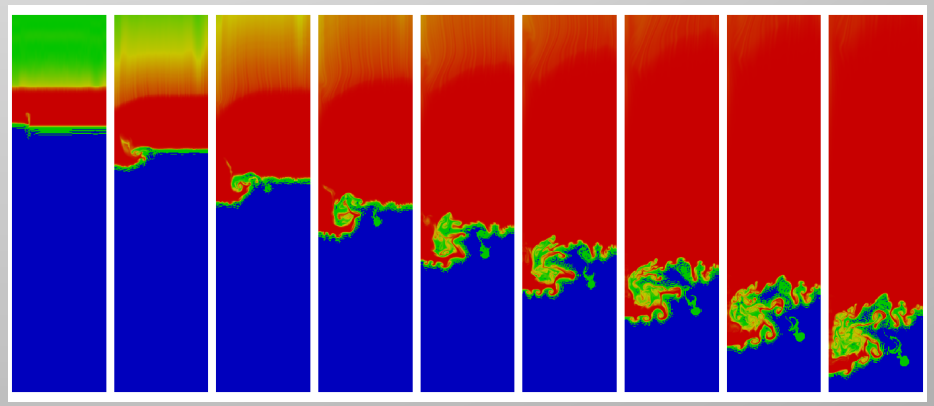
P erformance
A nalysis &
V isualization for
E xascale

H ardware
A pplication
C omm.

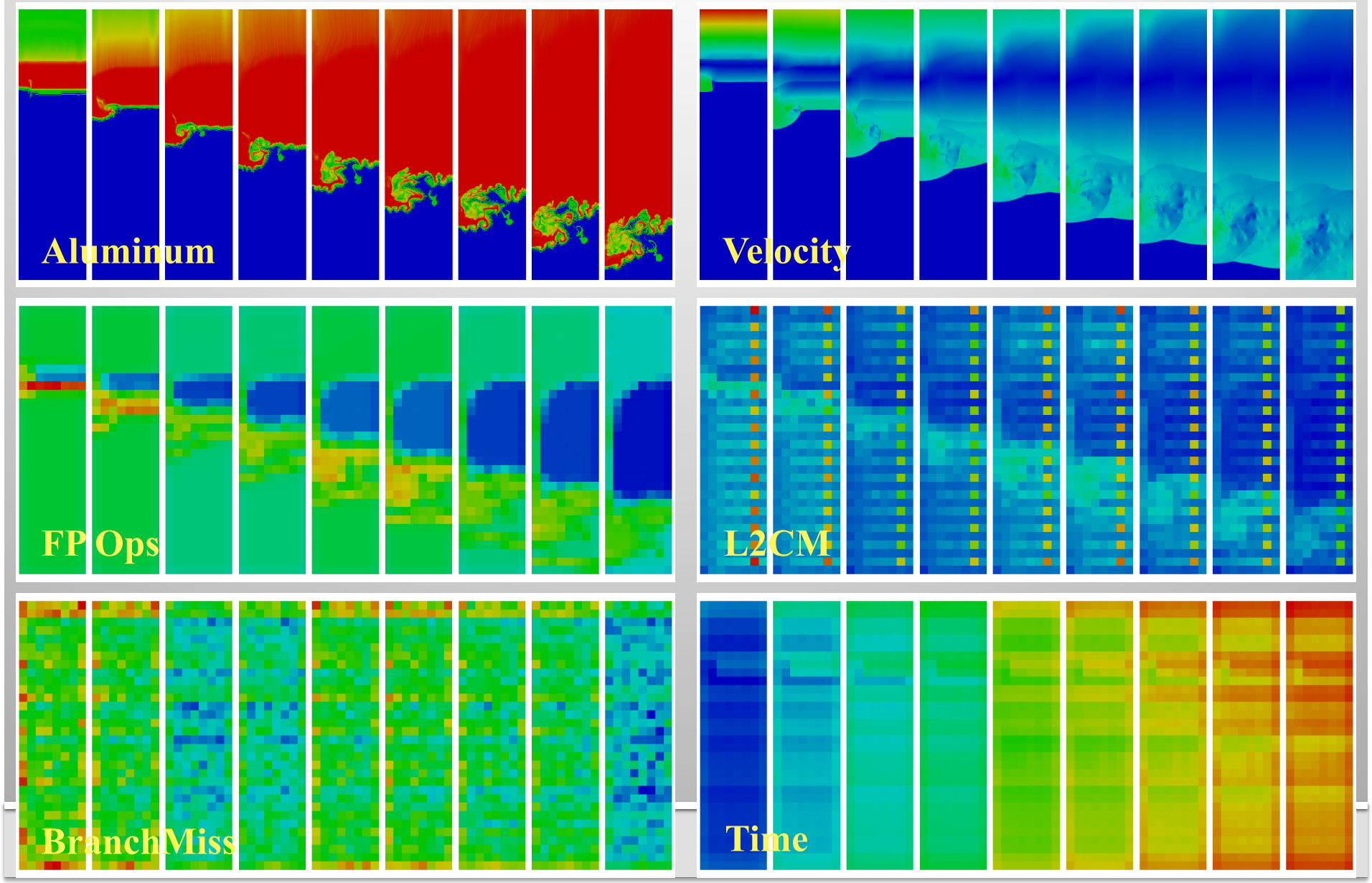


Mapping Measurements into the Application Domain

- Performance measurements acquired in the hardware domain
  - Map into the application's physical space
  - Visualize data in the application domain that is familiar to the developer
  - Ability to use existing and proven visualization techniques
- Requirements
  - Applications need to expose process ID -> grid mapping
  - In some cases we extract this automatically
- Case study
  - CFD application
  - Shock wave caused by Aluminum jet
  - 2D version, 32x8 CPUs
  - 9 time steps

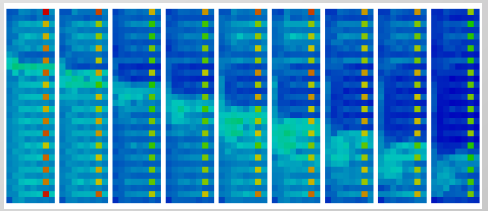


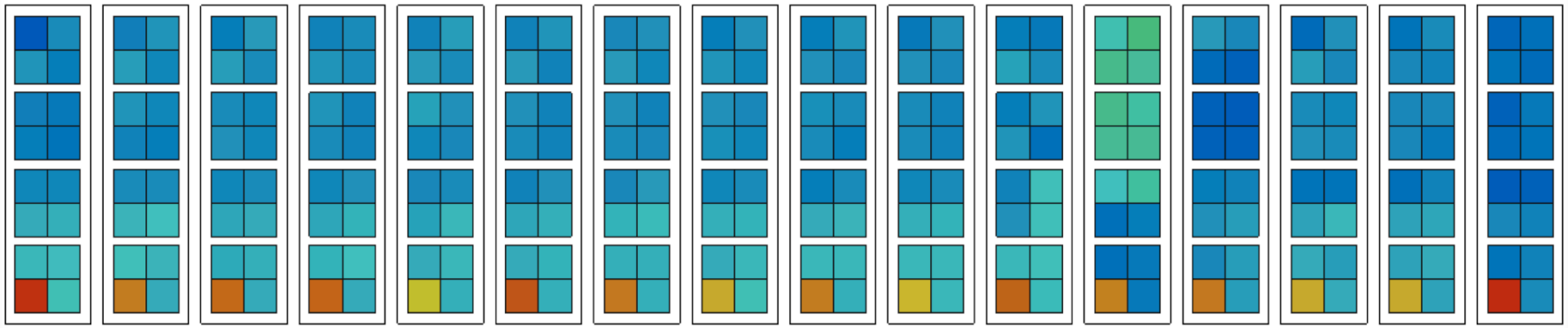
CFD Code, 9 Time Steps, 32x8 Processor Grid



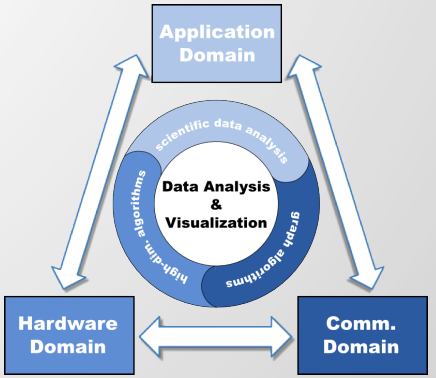
Observations

- Clear correlations for FP Ops and L2 misses
- Branch misses and time show boundary effects
- Time steps get more expensive over time
- Secondary, independent effect for L2 misses
  - Single core per socket creates more cache misses
  - Most likely caused by MPI activity
  - Shows that we need different perspectives to disambiguate causes

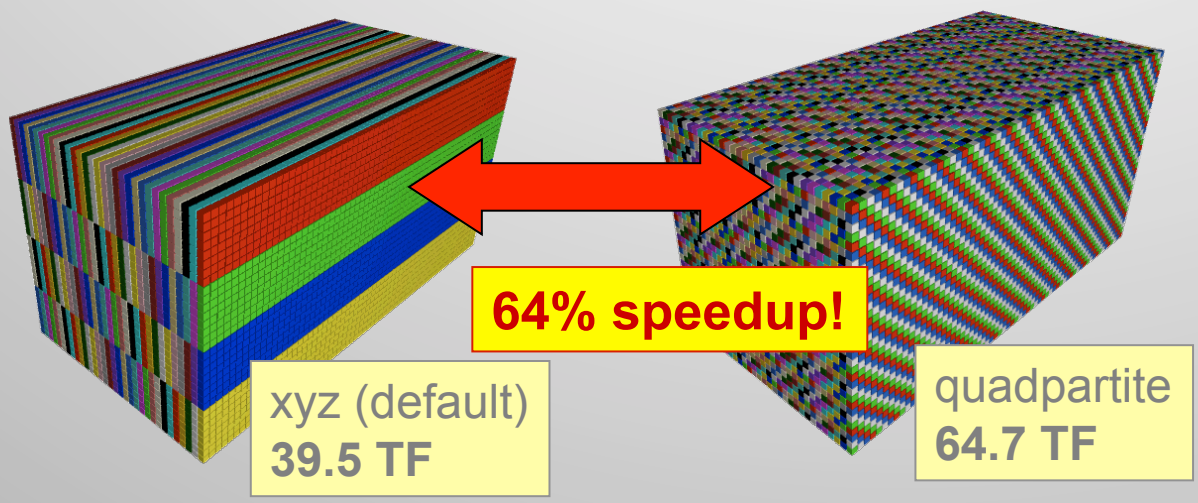


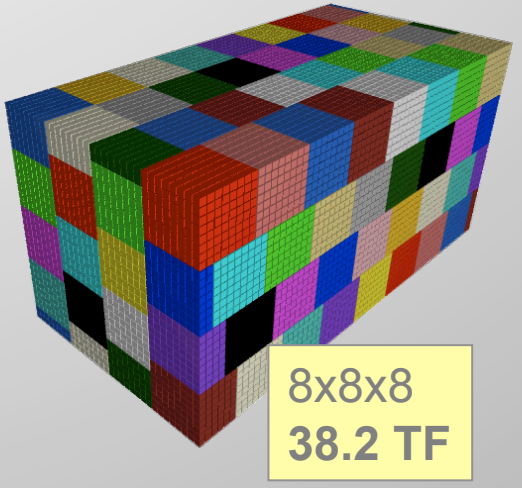


Hardware vs. Communication Domains



- Network topologies getting more complex
  - Interactions with communication topology non-trivial
  - Node placement has huge impact on performance

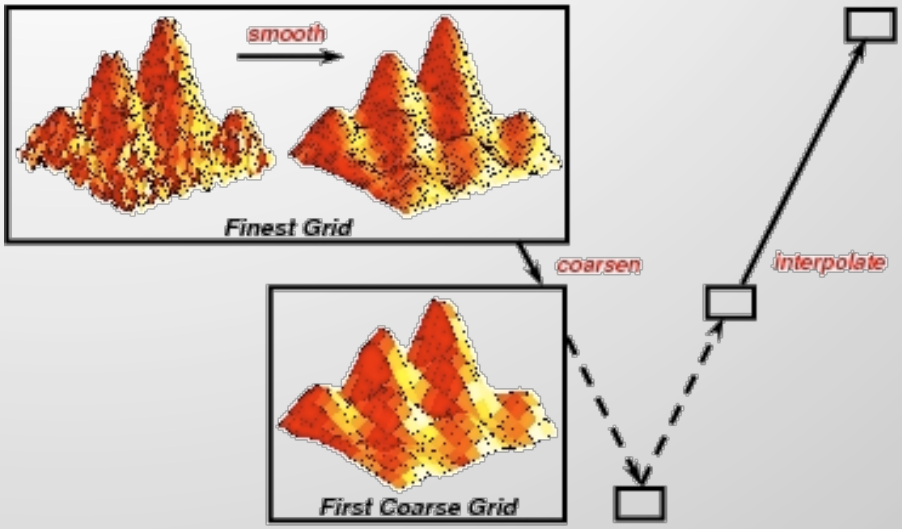




- Exploit network counters available on modern networks
  - Low-level per link with potentially shared packets
  - Hard to visualize

Case Study: Algebraic Multigrid Solver (AMG)

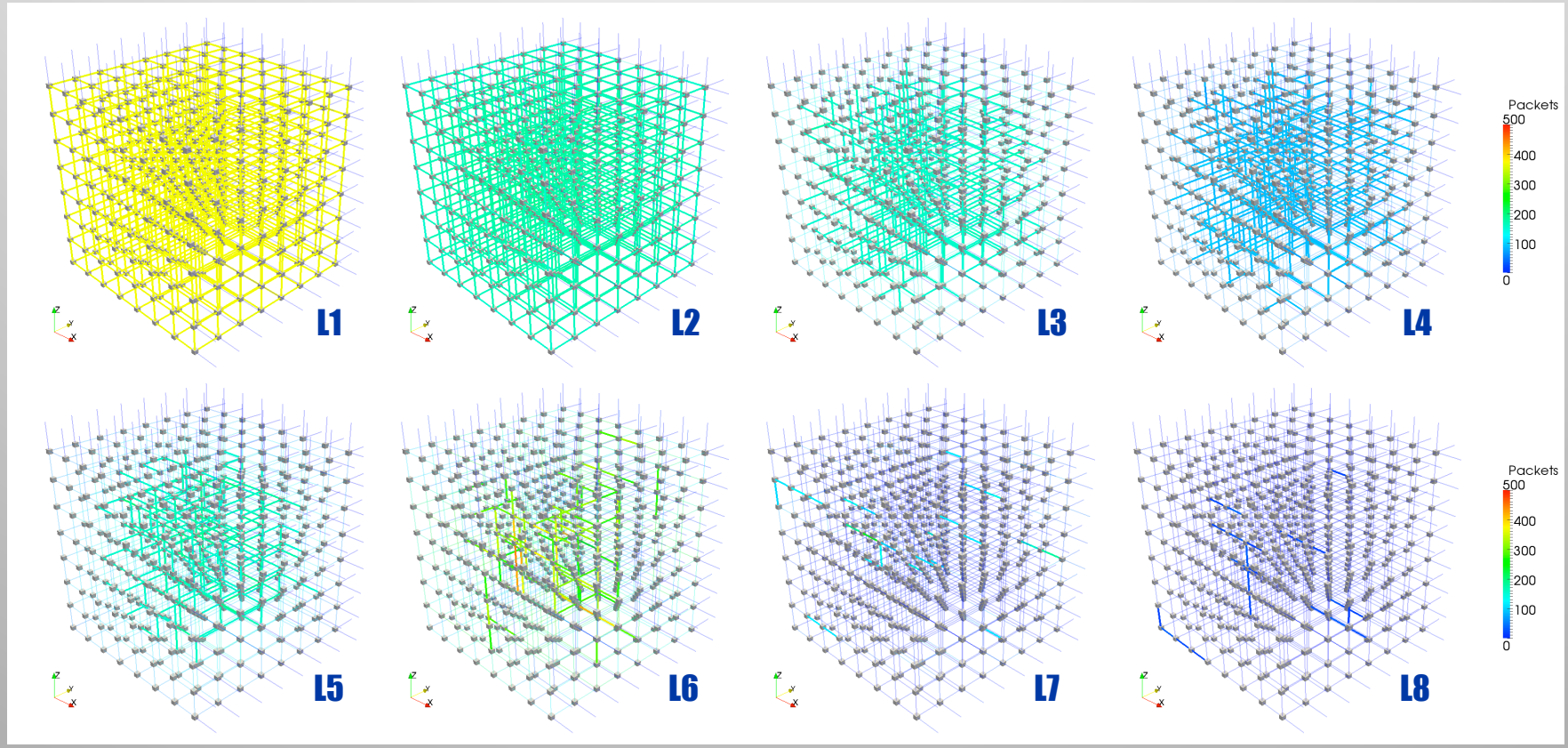
- Essential component for many applications
- Series of V Cycles
  - Coarsening
  - Direct solve
  - Interpolation



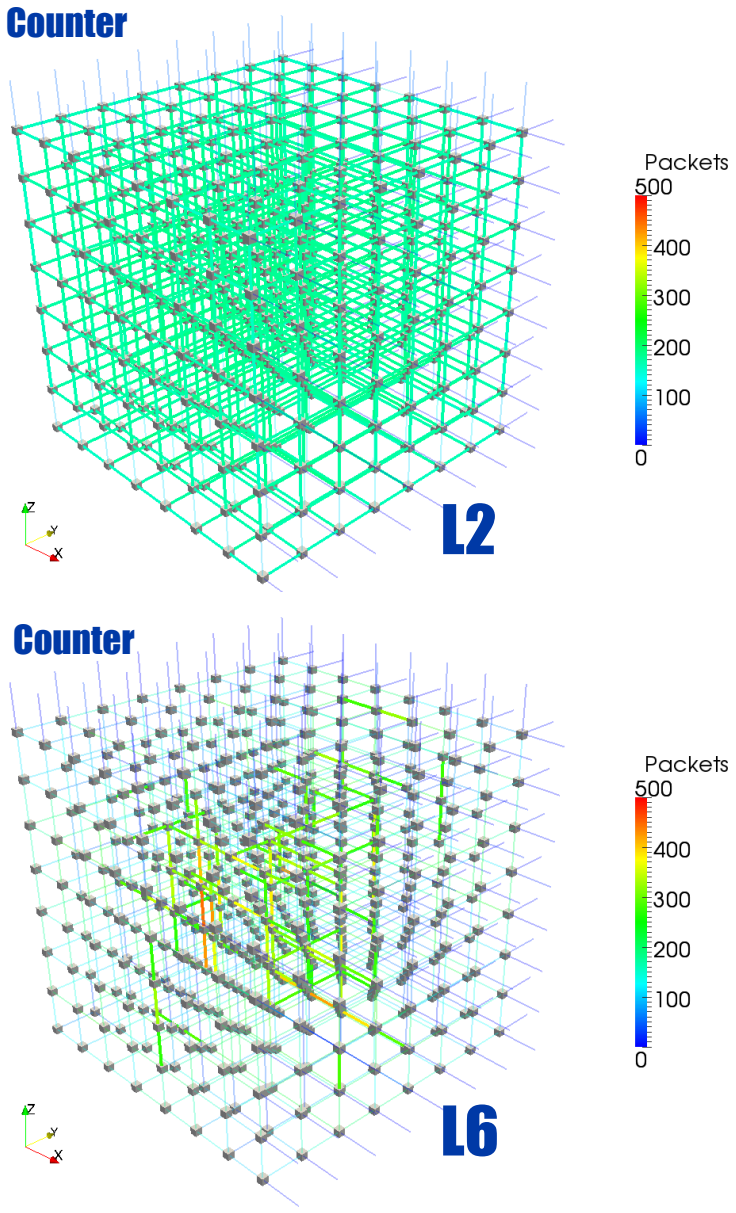
- Communication requirements change between layers
  - Fine layers have nearest neighbor communication
  - Coarse layers have more and less communication partners
  - Potential for link contention
- Experimental setup
  - AMG2006 on BG/P, 512 nodes/tasks
  - Measurements of X+/X-, Y+/Y-, Z+/Z- link activity

AMG on BG/L, 8x8x8 HW Torus, 8x8x8 virtual topology

- Communication counters for all eight levels of AMG
  - Mapped/Aggregated to the edges in a torus display

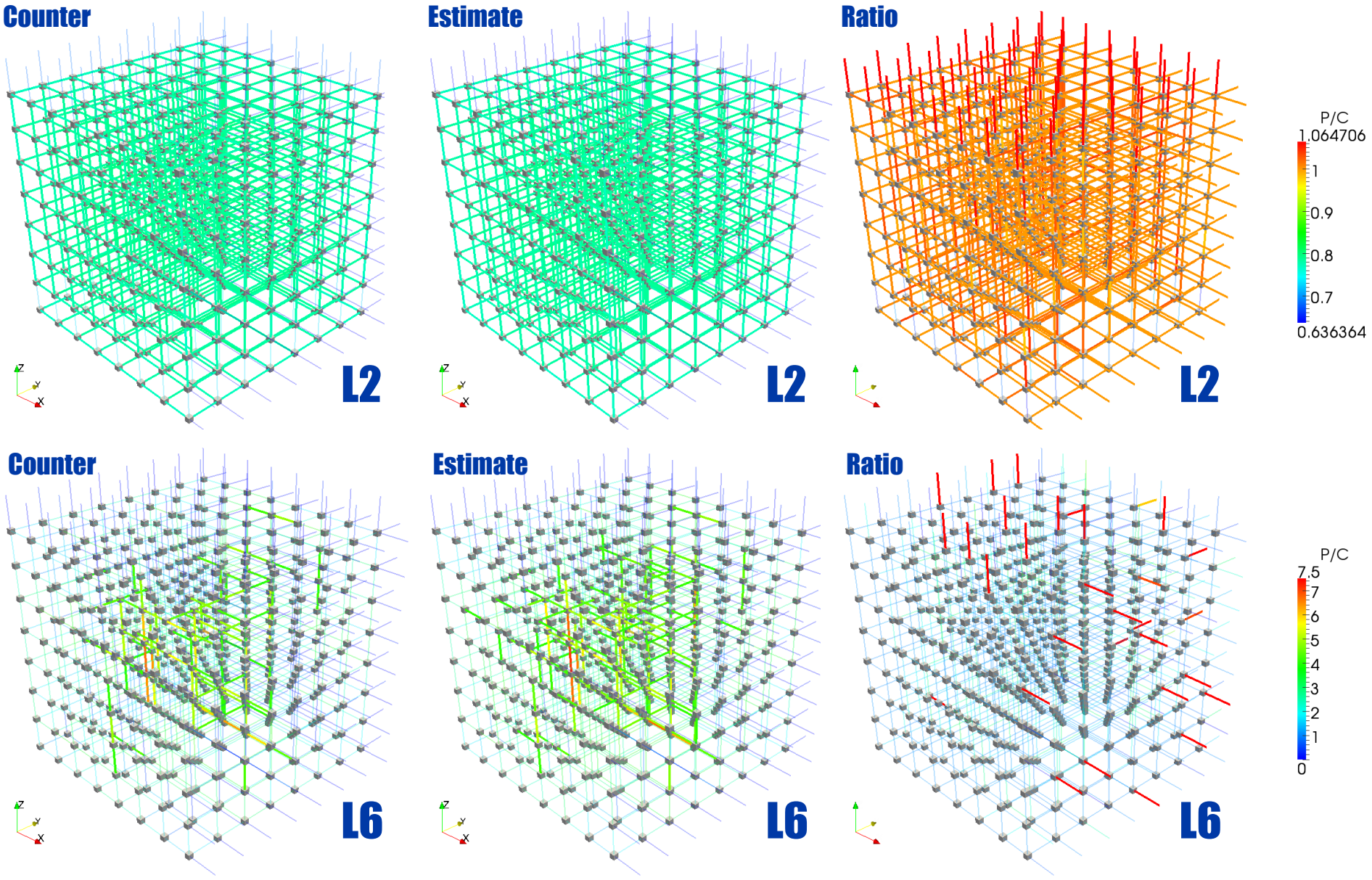


AMG on BG/L, 8x8x8 HW Torus, 8x8x8 virtual topology



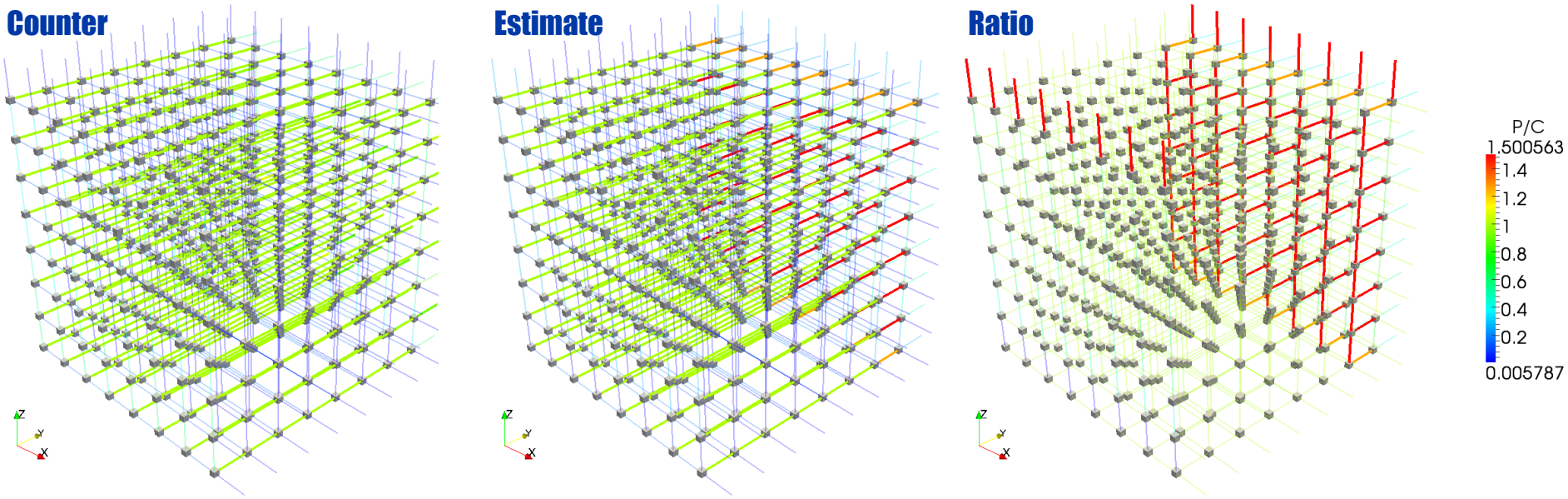
- New visualization that shows the hardware topology (level 2 vs. 6)
  - Finer layers are more global
  - Coarse layers have fewer partners
- How can we interpret the data?
  - Need connection to MPI communication
  - Need a baseline to compare to
- Map Communication to HW domain
  - Gather full MPI communication matrix
  - Emulate each message based on observed patters and aggregate
  - Contrast estimate with measurements
  - Ability to detect hotspots/contention

AMG on BG/L, 8x8x8 HW Torus, 8x8x8 virtual topology



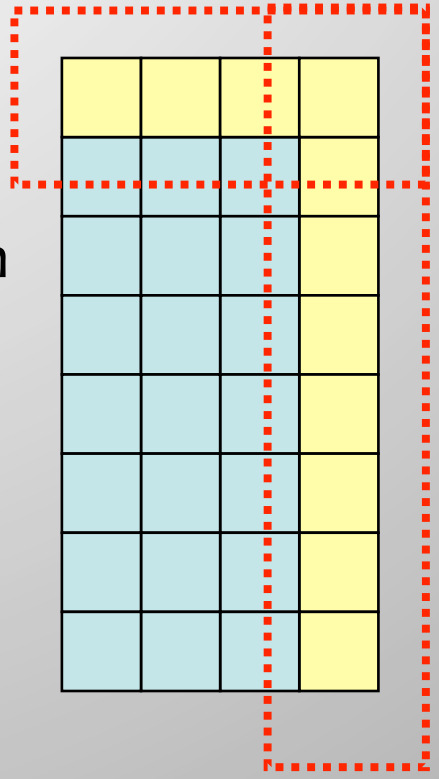
Observation

- Identify communication sparsity
  - Communication displays provide good insight
  - Leverage data analysis and visualization techniques
- Experiments with non optimal decompositions
  - BG/L, 8x8x8 HW Torus, 2x4x64 virtual topology
  - Results show more potential bottlenecks, but ratio is small



Exploiting the HAC Model to Gain New Insights

- Basis for node mapping optimizations
  - Comparison of multiple mappings
  - Communication or I/O traffic
- Mapping communication to application domain
  - Identify communication structure
  - Across library boundaries and communicators
  - Quick understanding of communication properties
- Feature and Anomaly detection
  - Automatic comparison of multiple domains
  - Identify what correlates and what not
  - Helps to establish a baseline

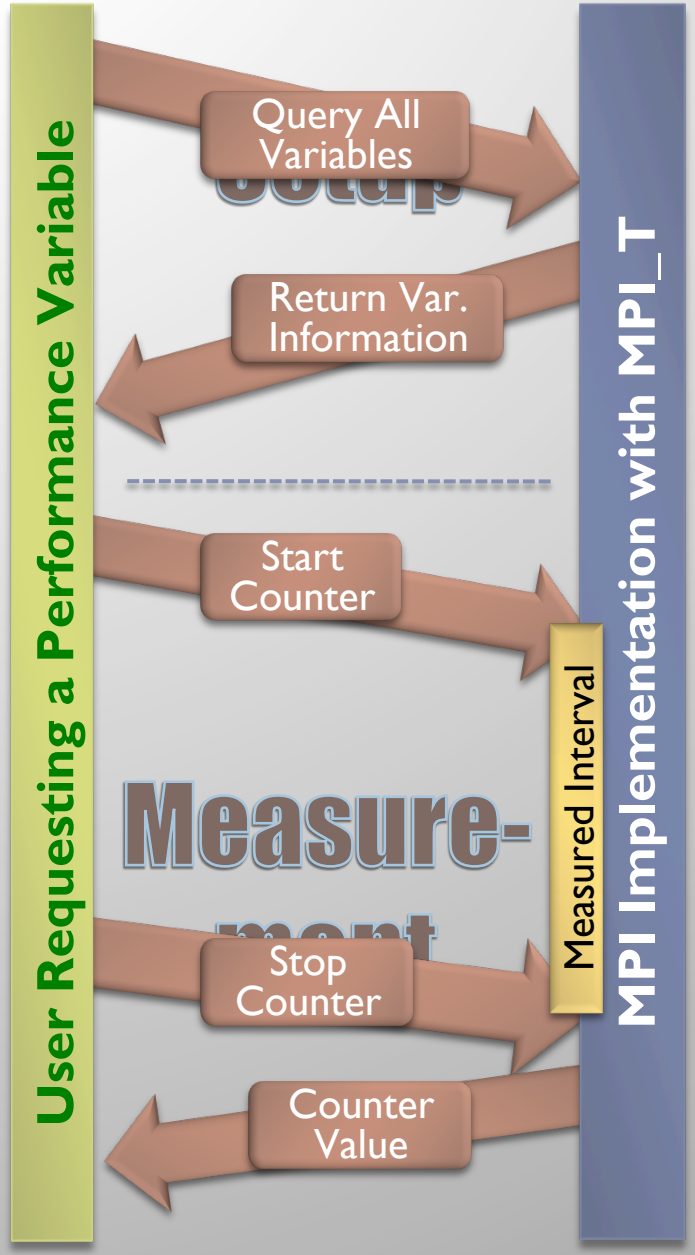


This is Only One Piece of the Puzzle

- Assumes that we can get the data
  - Mapping/Topology information
  - Performance data on various domains
- Applications need to export topology information
  - Standardized/Cross-tool specifications
  - APIs to enable vertical information flow
- New measurements/performance data APIs
  - Gather currently unavailable data
    - Environmental data (power, temperature)
    - Runtime (MPI) internal data
  - Need new interfaces to provide this information
  - System-wide Co-Design, in with PMs, SSW, Arch, I/O, …

Breaking Information Barriers

- Need information flow across layers, components, …
- Example: MPI_T interface
  - Expose hidden costs from MPI
    - Queue lengths
    - Blocking times
  - Query interface to avoid MPI implementation specific rules
  - Proposed for MPI-3
- Need similar APIs for other programming models or libraries
- Portable "standards"!



New Perspectives Lead To New Insights

- New and easy insights by using multiple domains
  - Makes performance results tangible
  - Enables multiple perspectives to disambiguate effects
- PAVE: Performance Analysis & Visualization for Exascale
  - Joint project with Todd Gamblin and Timo Bremer
  - Collaboration with SCI@Utah (Valerio Pascucci and Joshua Levine)
  - Bridge between Performance & Visualization/Analysis
- Only one piece of the puzzle
  - Information interfaces from all system layers
  - Export ability of application specific information
  - Correlate performance between components
- Need holistic approach to expose all hidden costs!