Gaining Performance Insights Through Interactive Visualization

VI-HPS 10th Anniversary Workshop

Martin Schulz
schulzm@llnl.gov

http://scalability.llnl.gov/

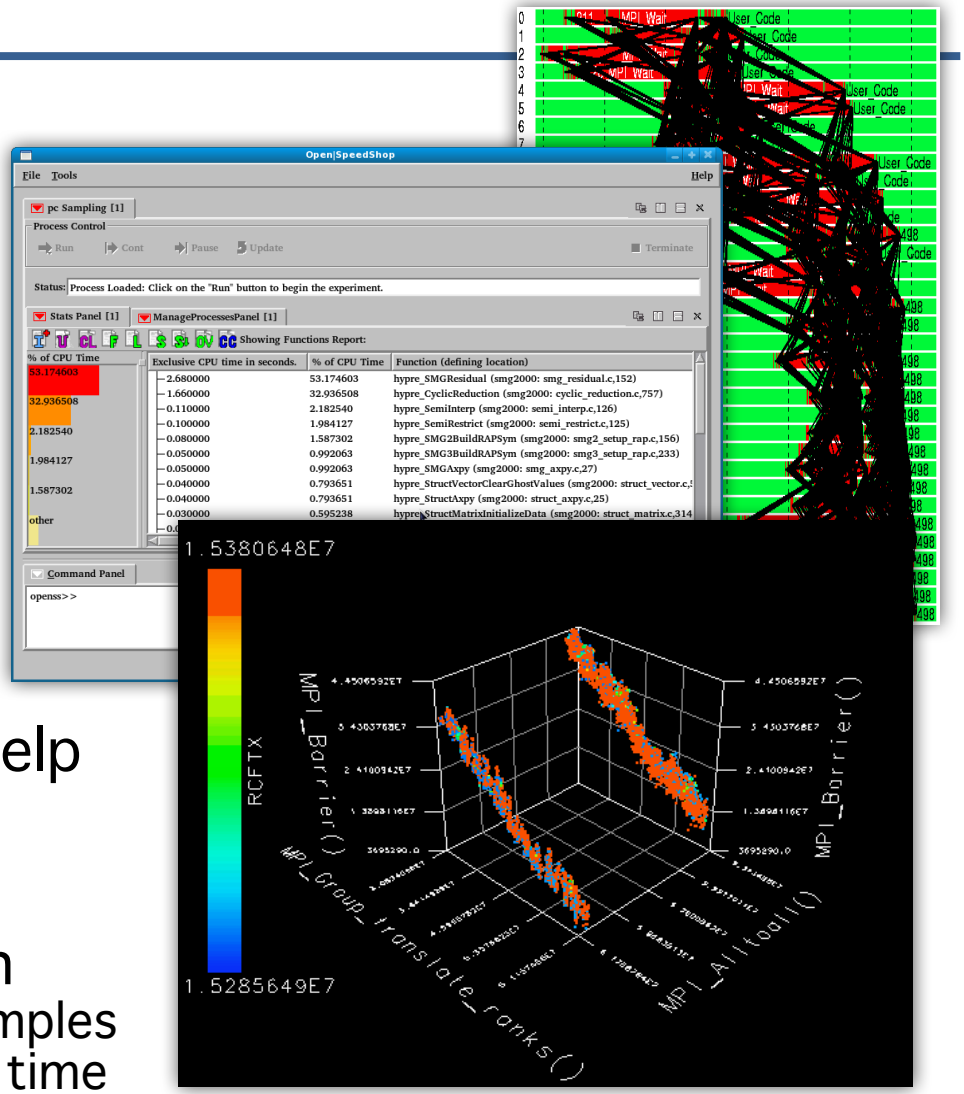
LLNL-PRES-733709

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

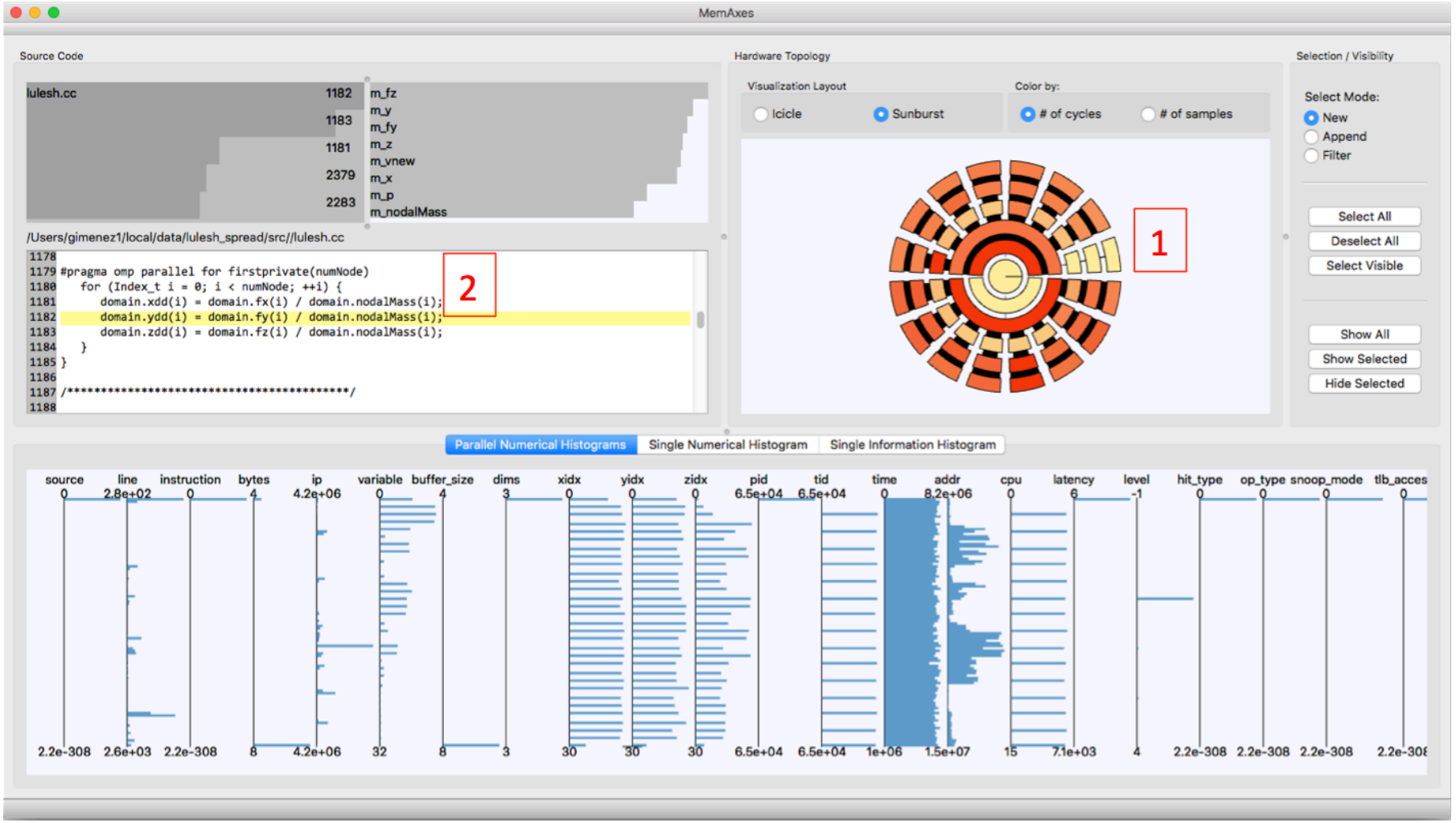
Lawrence Livermore National Laboratory

Development Tools are Critical for Exascale

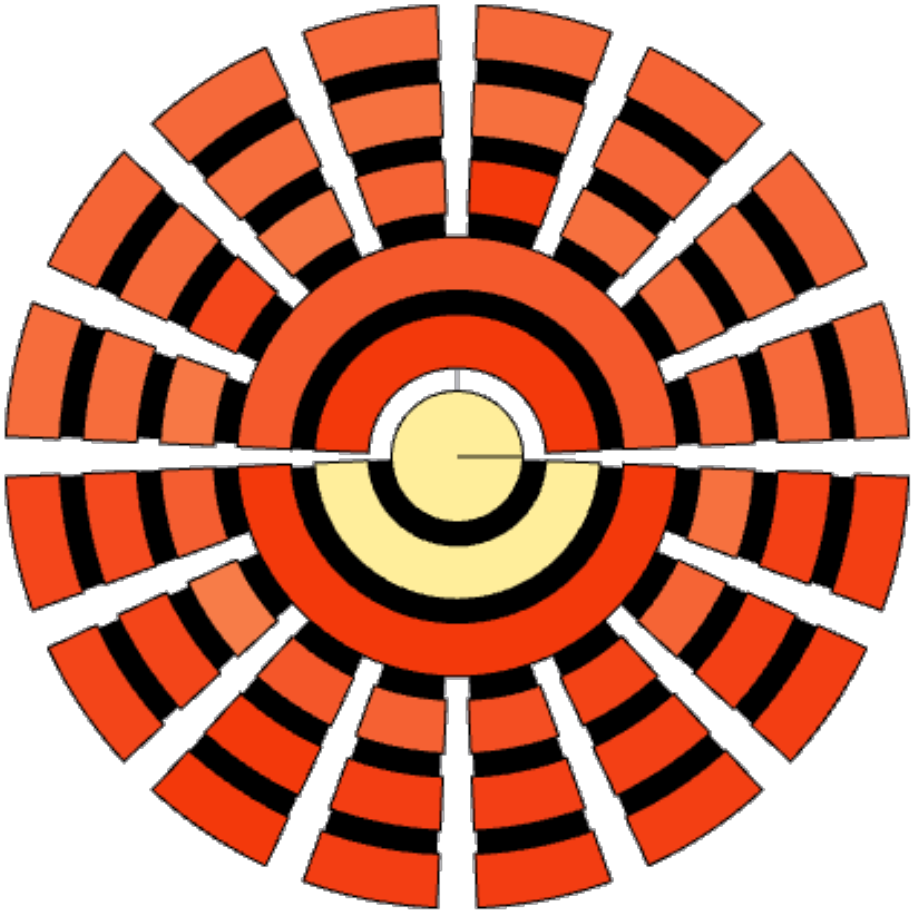
- LLNL Contributions to VI-HPS
  - OpenlSpeedShop
  - mpiP
  - STAT
  - MUST (with RWTH)
  - CBTF/launchMon/P^{n}MPI
  - Soon: Caliper
- Performance tools can collect vast amounts of data
  - But: how to interpret them?
- Interactive visualization can help
  - Intuitive exploration of data
  - Data reduction
- Two examples on data motion
  - MemAxes: Display of memory samples
  - Ravel: Display of traces in virtual time



MemAxes (v2): Visualizing Memory Traffic



Case Study 1: Memory Access in XSBench (Monto Carlo Proxy App)



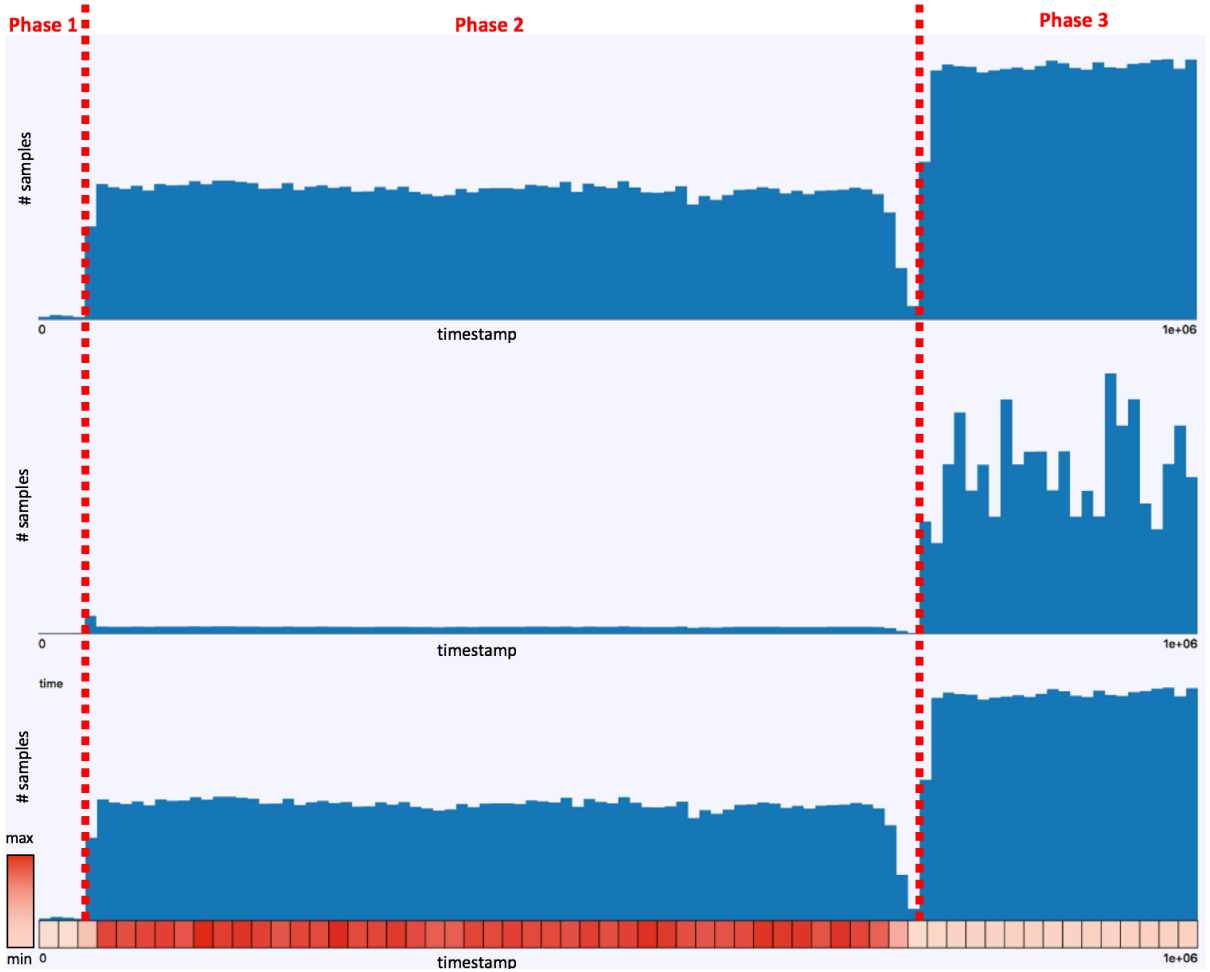
- Observation 1:
  - Uniforum memory access from all cores
- Observation 2:
  - All data accessed on a single NUMA core
- Phase description helps further investigate the results
  - Three main phases
  - Different characteristics

Case Study 1: Memory Access in XSBench (Monto Carlo Proxy App)

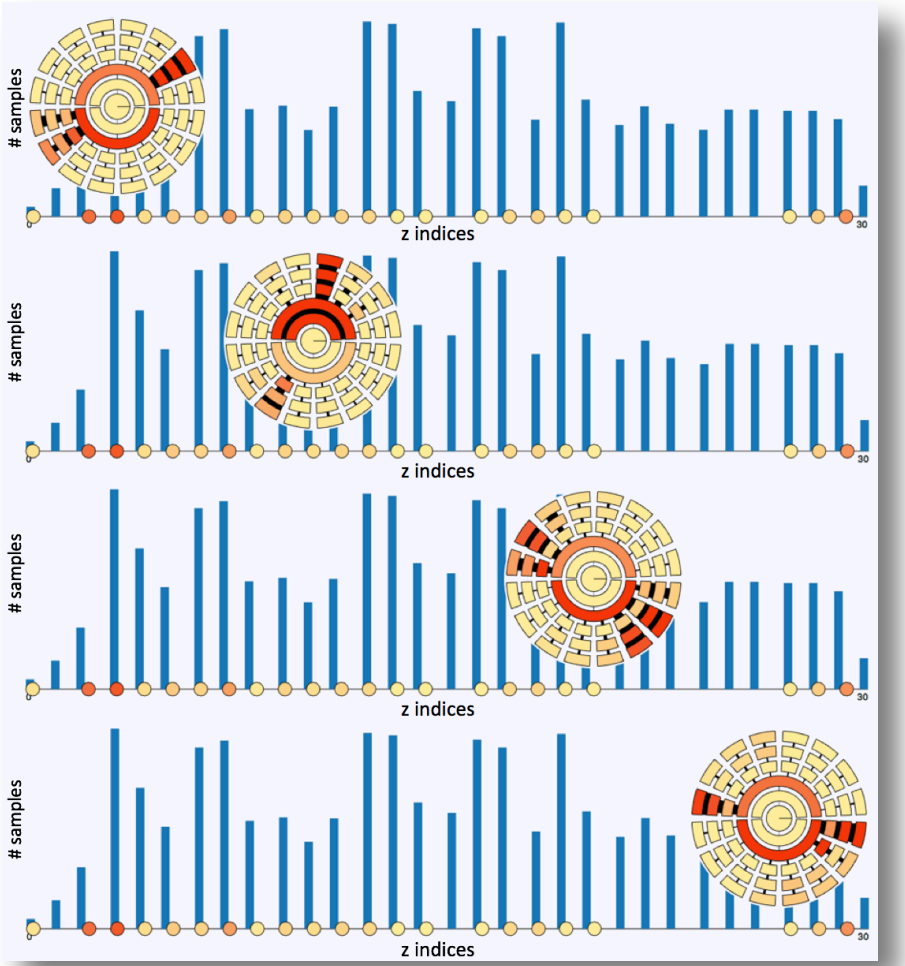
Time

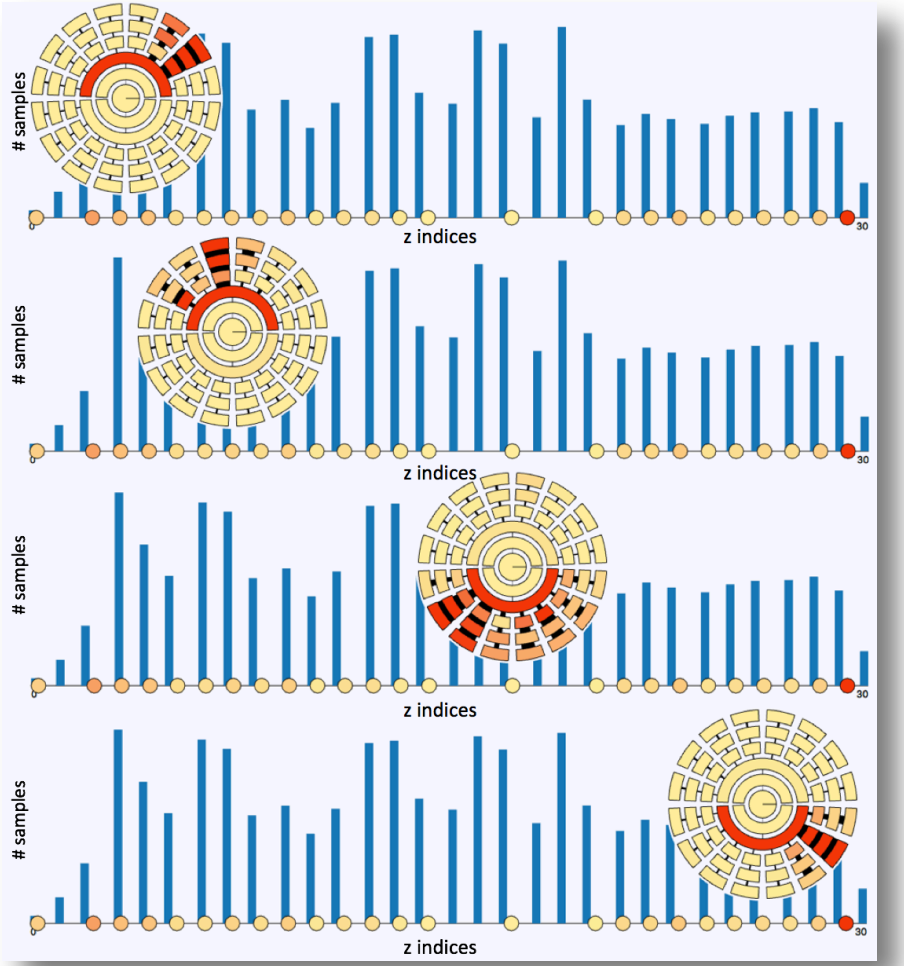
NUMA accesses

L2 accesses



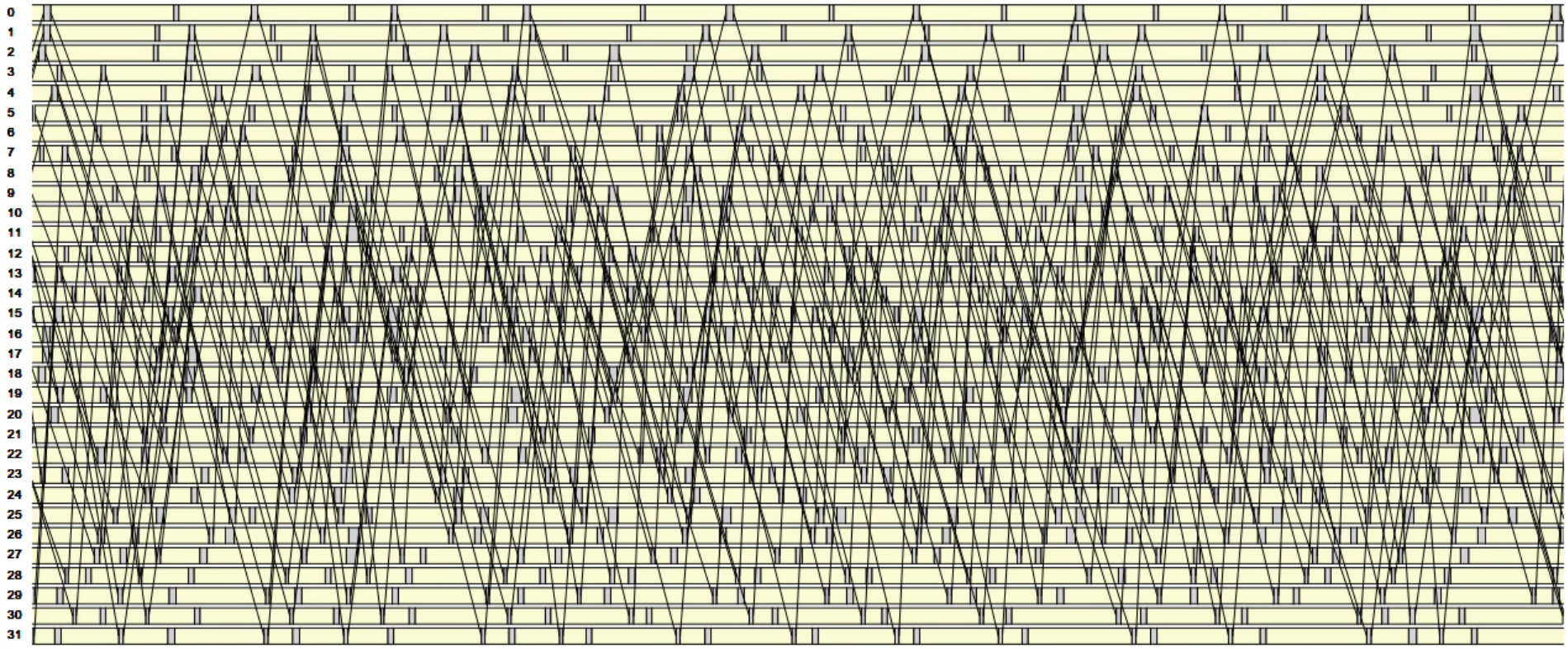
Case Study 2: Locality in Lulesh Applications (Shock Hydro Proxy)





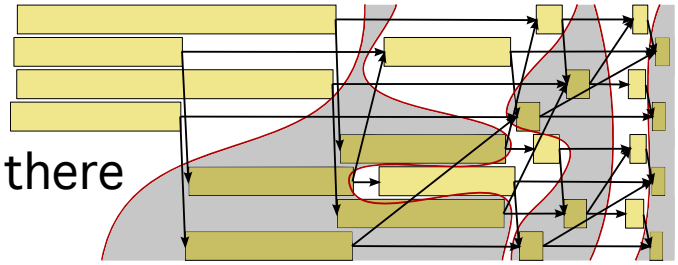
Ravel: Making Message Traces Readable

- Trace visualization is a helpful tool to show message details
  - But: we need new techniques to unravel this hairball
  - Applicable to MPI and task-based traces (e.g., Charm++)

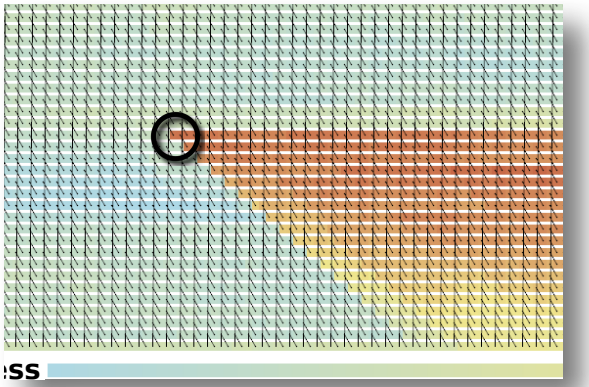


Extracting Logical Time Order

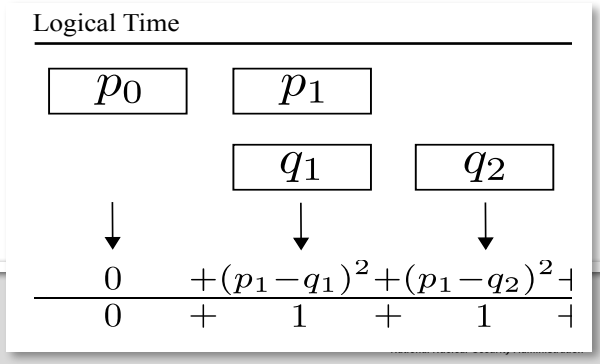
- Step 1: Identifying time slices
  - Concept of connected components
  - Start with send/recv pairs and grow from there
  - Heuristics on when to stop growing



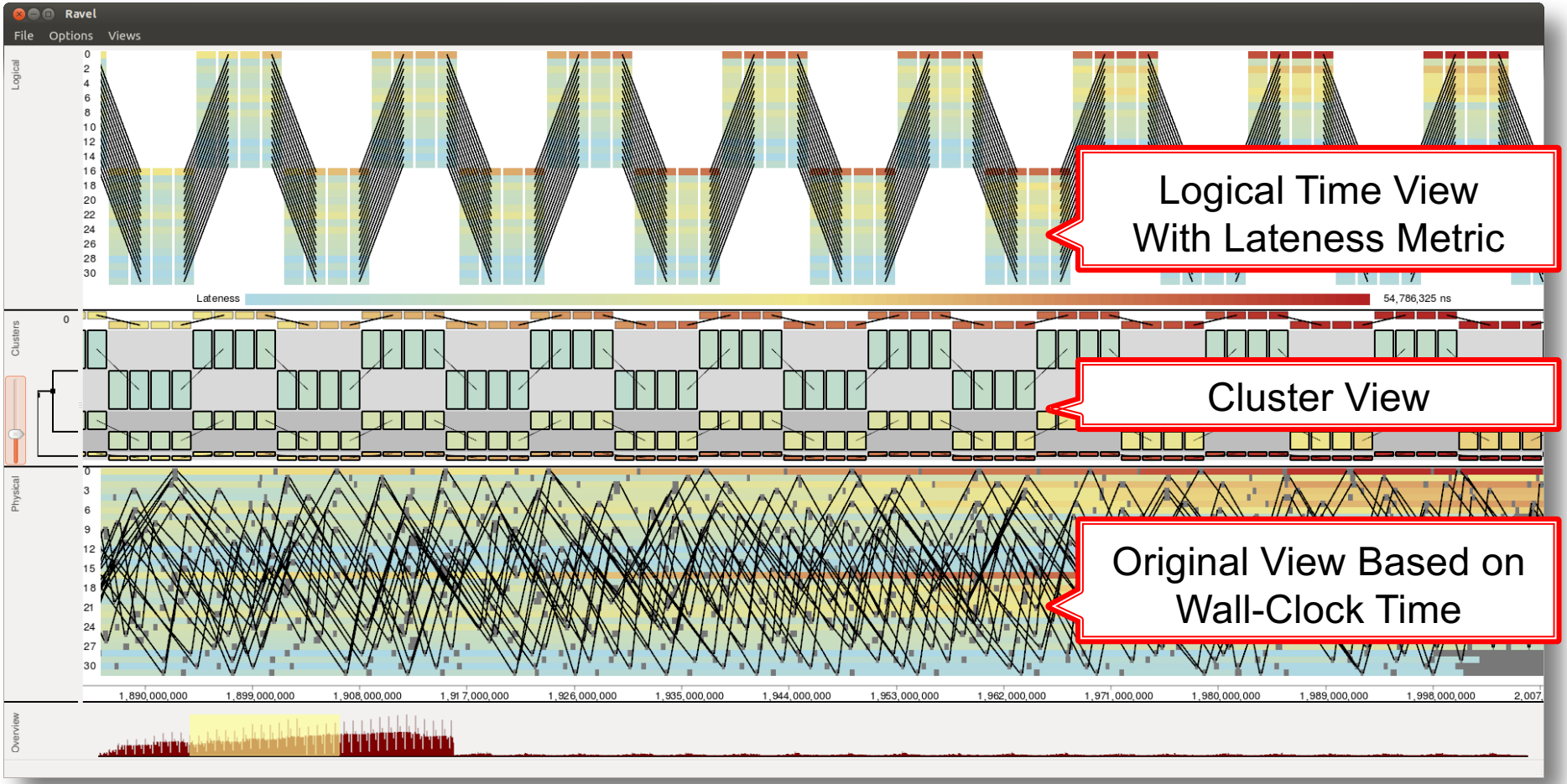
- Step 2: Mapping timing metrics
  - Mapping to virtual time loses physical time
  - Reintroduction of time using lateness metric
    - Time difference to end of aligned phase
    - Shows propagations of delays



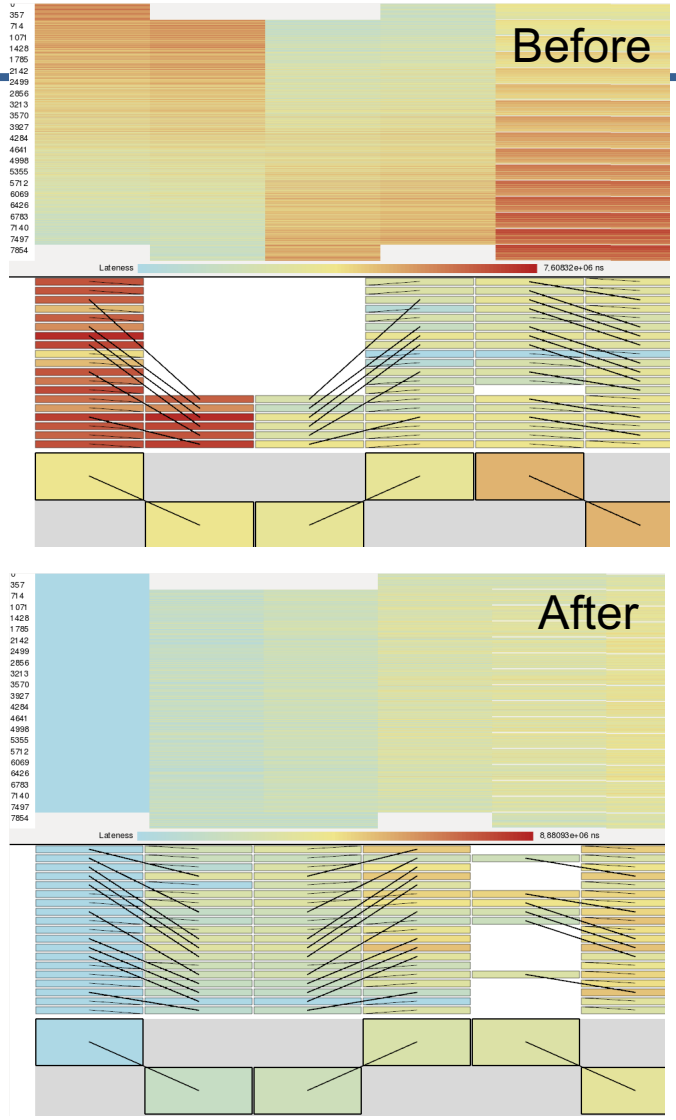
- Step 3: Cross process clustering
  - Aggregate traces with similar lateness
  - Use of representative traces to show data

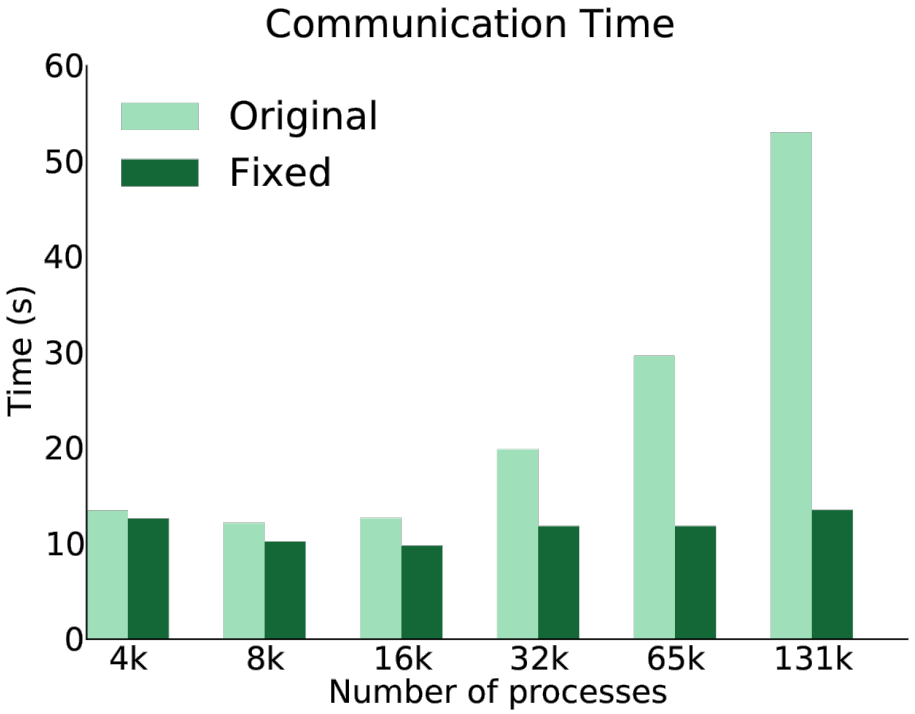


Ravel: Trace Visualization Using Logical Time

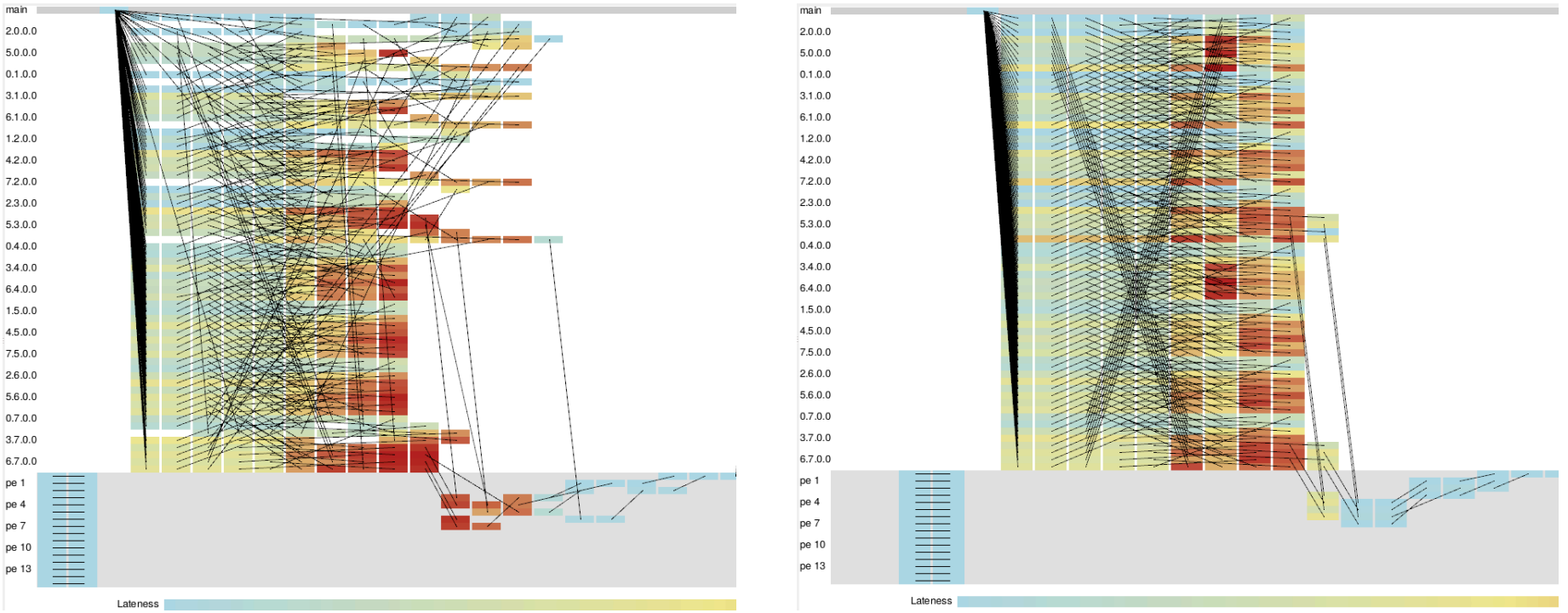


Case Study: Optimizing Communication Patterns





Unraveling Task Based Execution
An Example Based on Charm++



- Visualize tasks and their dependencies
- Left shows mess of tasks *considering* message receive order
- Right shows messages reordered to ignore nondeterminism, colored by lateness.

Conclusions

- Performance visualization can be a helpful approach
  - Interactive exploration of performance data
  - Increases intuition for developers
  - Mappings between domains helps to get new perspectives
  - Attribution and correlation with meta-data essential
- MemAxes shows on node memory access traffic
  - Memory sampling along with sample attributes
  - Display mapped to hardware architecture
- Ravel shows a logical timeline view of message traces
  - Enables new delay metrics
  - Applies to task based models as well
- Must be part of a larger set of efforts
  - Include more metrics (power, environmental, network, …)
  - Implicit and in-situ analysis of performance data
  - Extract the necessary context across the SW stack

The Scalability Team

http://scalability.llnl.gov/

Senior Staff


Abhinav Bhatele


Todd Gamblin


Ignacio Laguna


Kathryn Mohror


Barry Rountree


Martin Schulz

Junior Staff / Postdoc


David Beckingsale


David Boehme

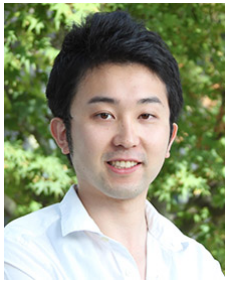

Murali Emani


Nikhil Jain


Harshita Menon


Aniruddha Marathe


Tapasya Patki


Kento Sato

- Performance analysis tools and optimization
- Correctness and debugging (incl. STAT, AutomaDeD, MUST)
- Power-aware and power-limited computing (incl. Adagio, Conductor)
- Resilience and Checkpoint/Restart (incl. SCR, I/O, file systems)

The Scalability Team

http://scalability.llnl.gov/

Senior Staff


Abhinav Bhatele


Todd Gamblin


Ignacio Laguna


Kathryn Mohror


Barry Rountree


Martin Schulz

Junior Staff / Postdoc


David Beckingsale


David Boehme

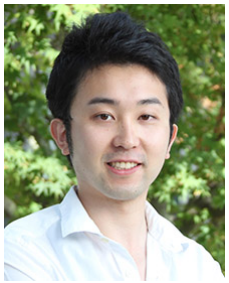

Murali Emani


Nikhil Jain


Harshita Menon


Aniruddha Marathe


Tapasya Patki


Kento Sato

- Performance analysis tools and optimization
- Correctness and debugging (incl. STAT, AutomaDe
- Power-aware and power-limited computing (incl. A
- Resilience and Checkpoint/Restart (incl. SCR, I/O,


Conclusions

http://scalability.llnl.gov/
http://github.com/LLNL/

- Performance visualization can be a helpful approach
  - Interactive exploration of performance data
  - Increases intuition for developers
  - Mappings between domains helps to get new perspectives
  - Attribution and correlation with meta-data essential
- MemAxes shows on node memory access traffic
  - Memory sampling along with sample attributes
  - Display mapped to hardware architecture
- Ravel shows a logical timeline view of message traces
  - Enables new delay metrics
  - Applies to task based models as well
- Must be part of a larger set of efforts
  - Include more metrics (power, environmental, network, …)
  - Implicit and in-situ analysis of performance data
  - Extract the necessary context across the SW stack