Tools for visualizing communication, network traffic, and job placement

Abhinav Bhatele
Center for Applied Scientific Computing

Petascale Tools Workshop ◆ August 04, 2014

LLNL: Peer-Timo Bremer, Todd Gamblin, Katherine E. Isaacs, Steven H. Langer, Martin Schulz

Davis: Dylan Wang, Dipak Ghosal

Illinois: Nikhil Jain, Laxmikant V. Kale

Utah: Aaditya G. Landge, Joshua A. Levine, Valerio Pascucci

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

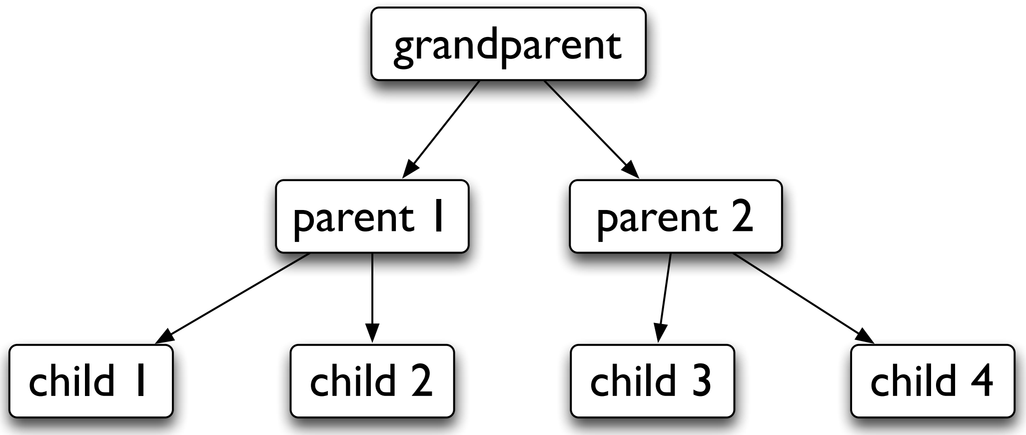
Lawrence Livermore National Laboratory, P. O. Box 808, Livermore, CA 94551

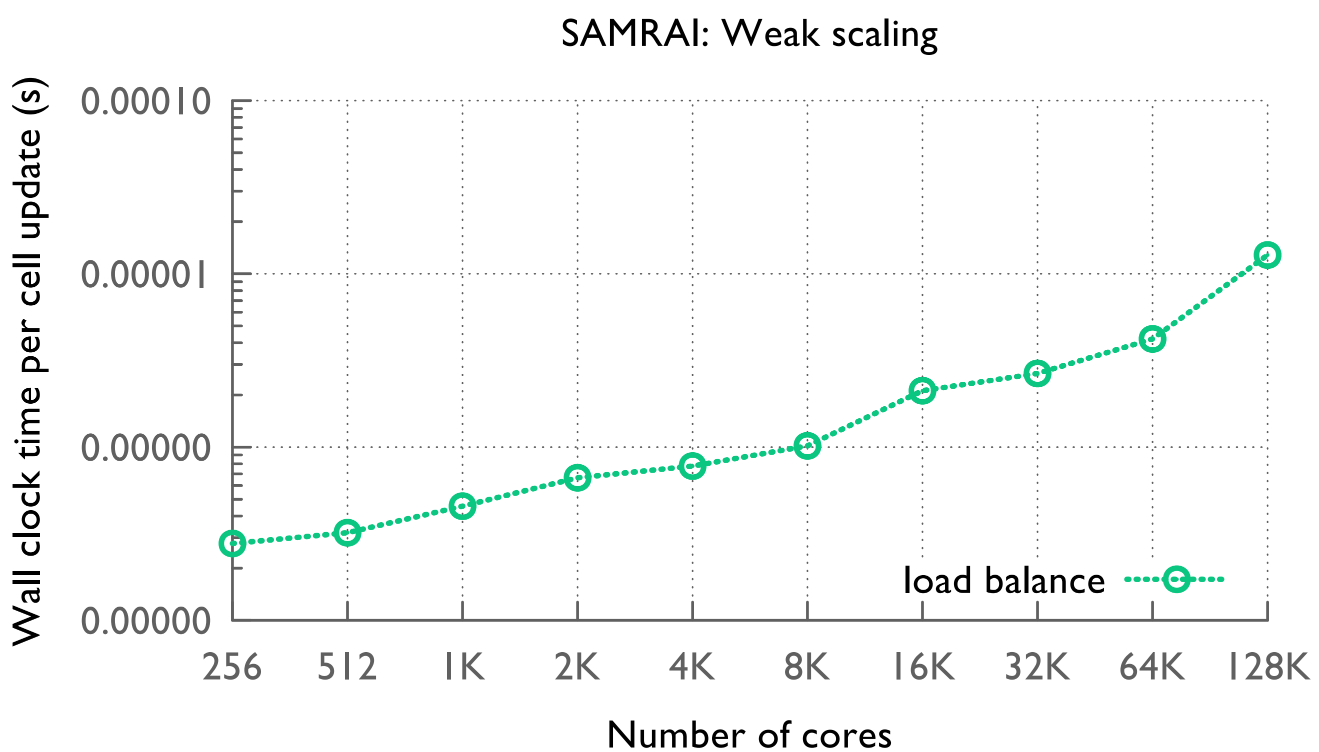
Performance analysis at extreme scale

- Large number of processes in an execution
  - Large amounts of data - impossible to analyze manually
- Complex architectures and adaptive applications
  - Make attribution of problems to the real cause difficult
- Traditional performance analysis tools leave a lot to the user

Load balancing in SAMRAI

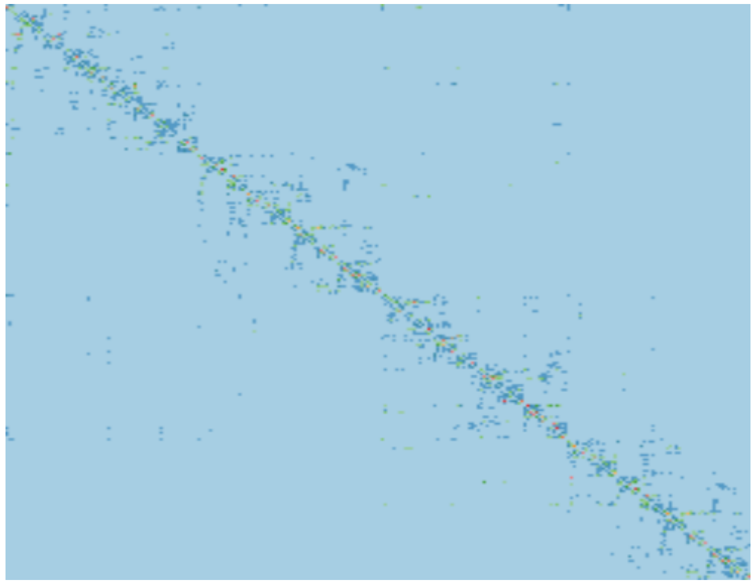
- Phase in which load balancing decisions are made
- Three sub-phases:
  - Phase 1: Load distribution
  - Phase 2: Mapping generation
  - Phase 3: Overlap update

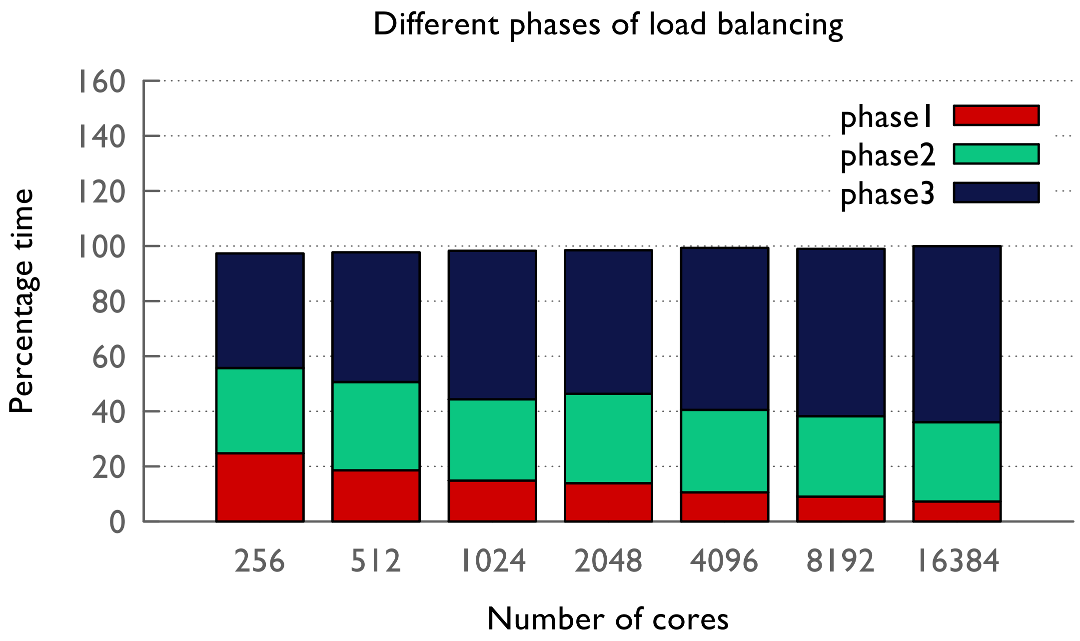


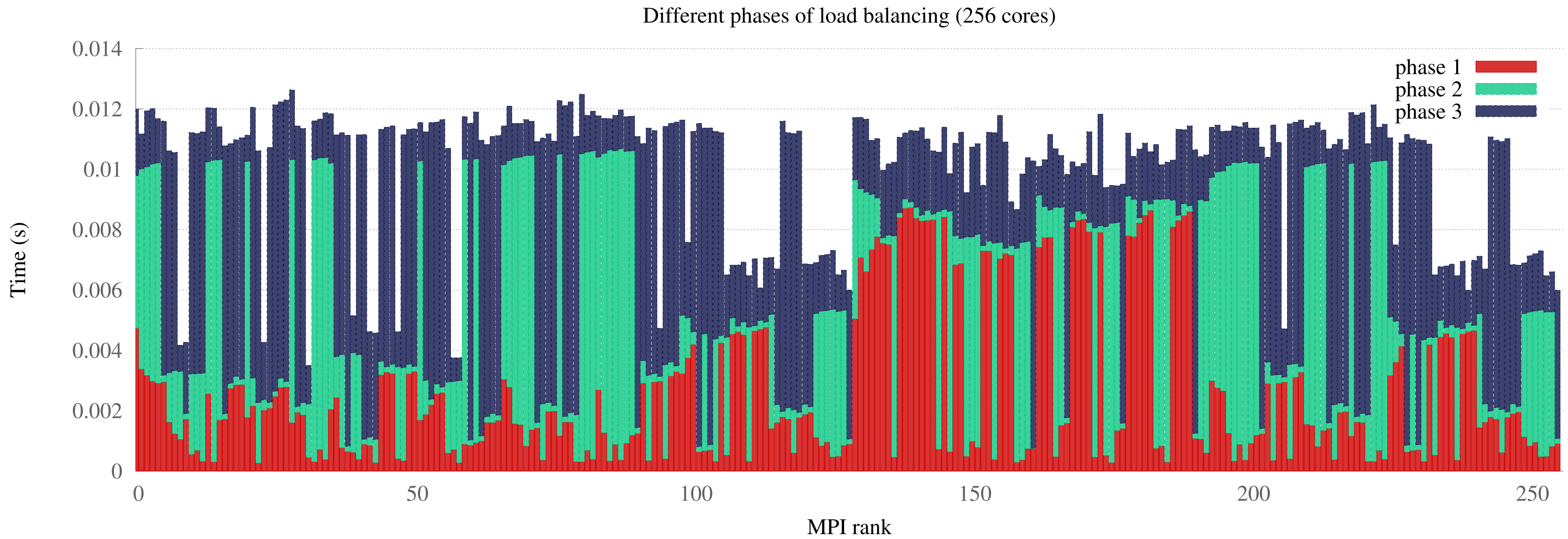


Abhinav Bhatele et al. Novel views of performance data to analyze large-scale adaptive applications. In Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC '12. November 2012. LLNL-CONF-554552.

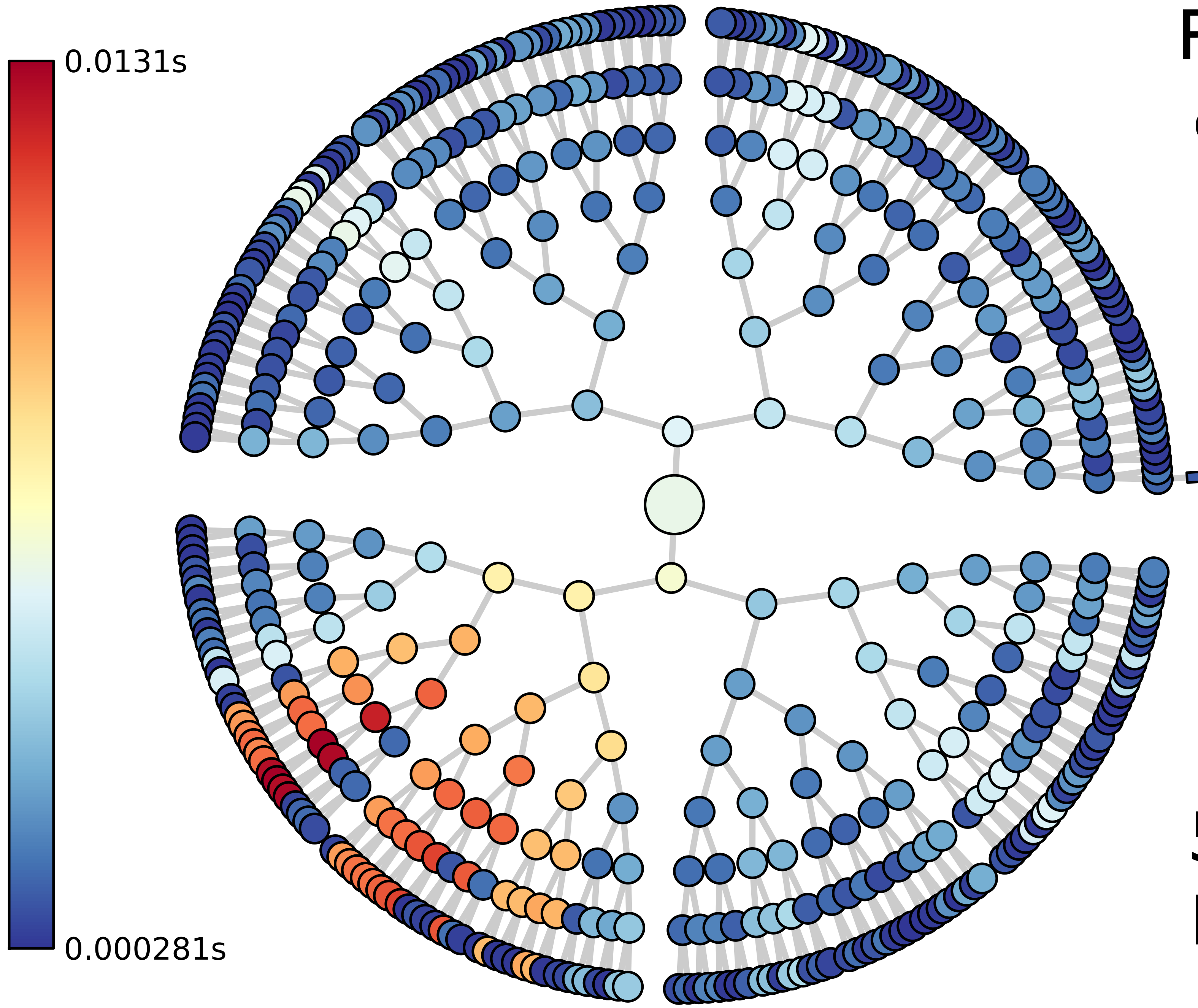
Traditional performance analysis







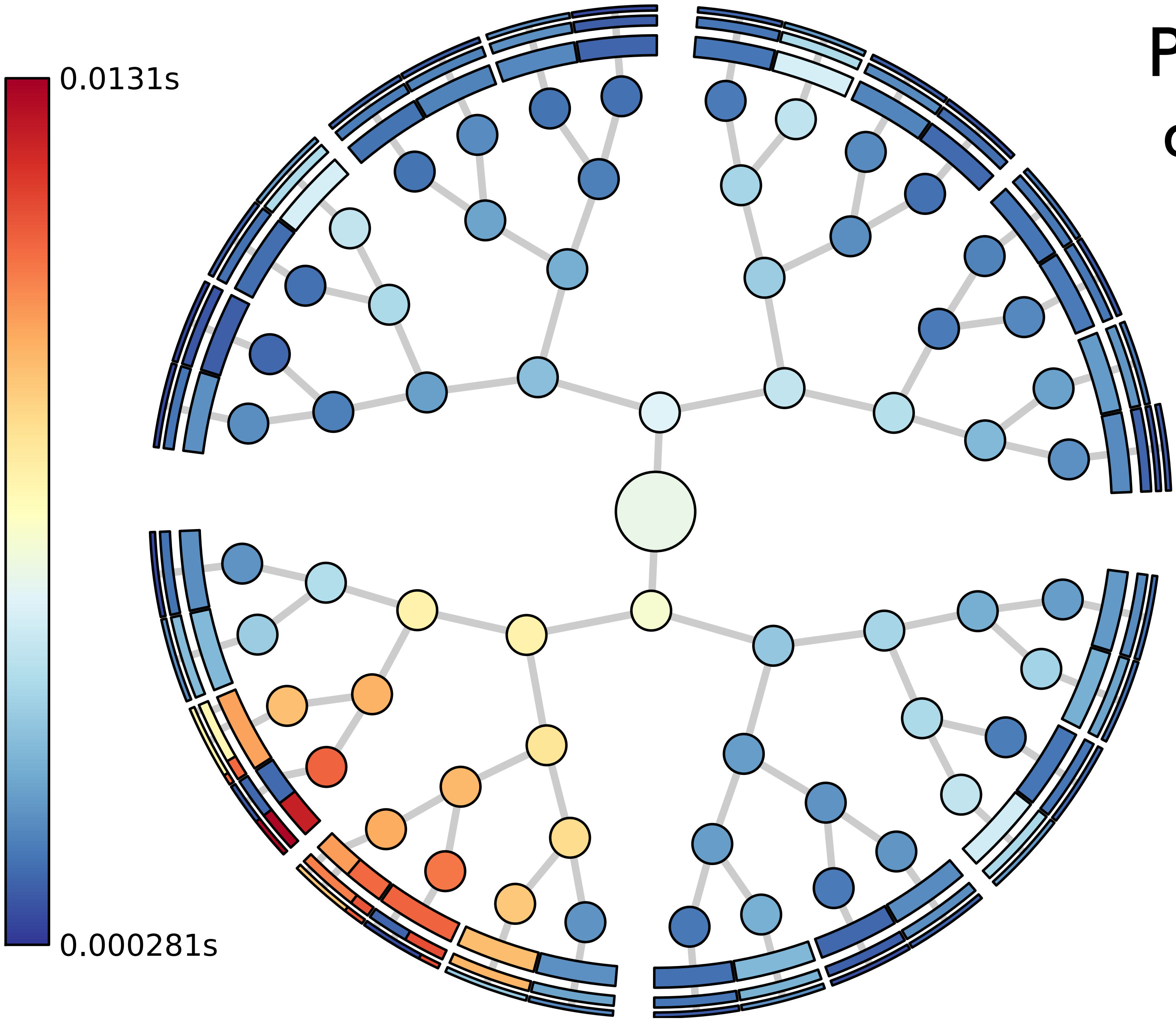
Projections on the communication domain



Phase 1 (load distribution) timing data

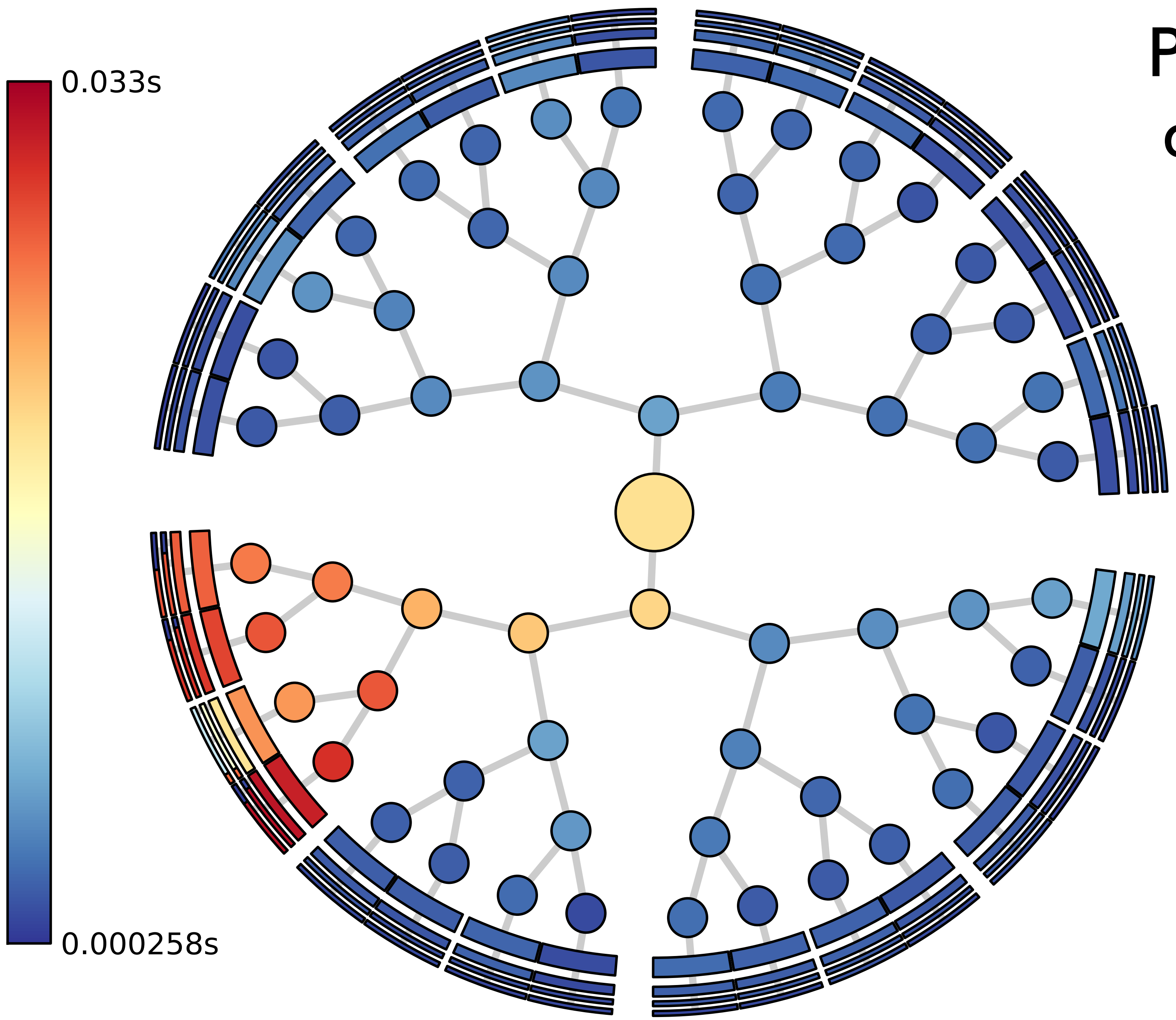
512 cores of Blue Gene/P

Scalable view of the comm. graph



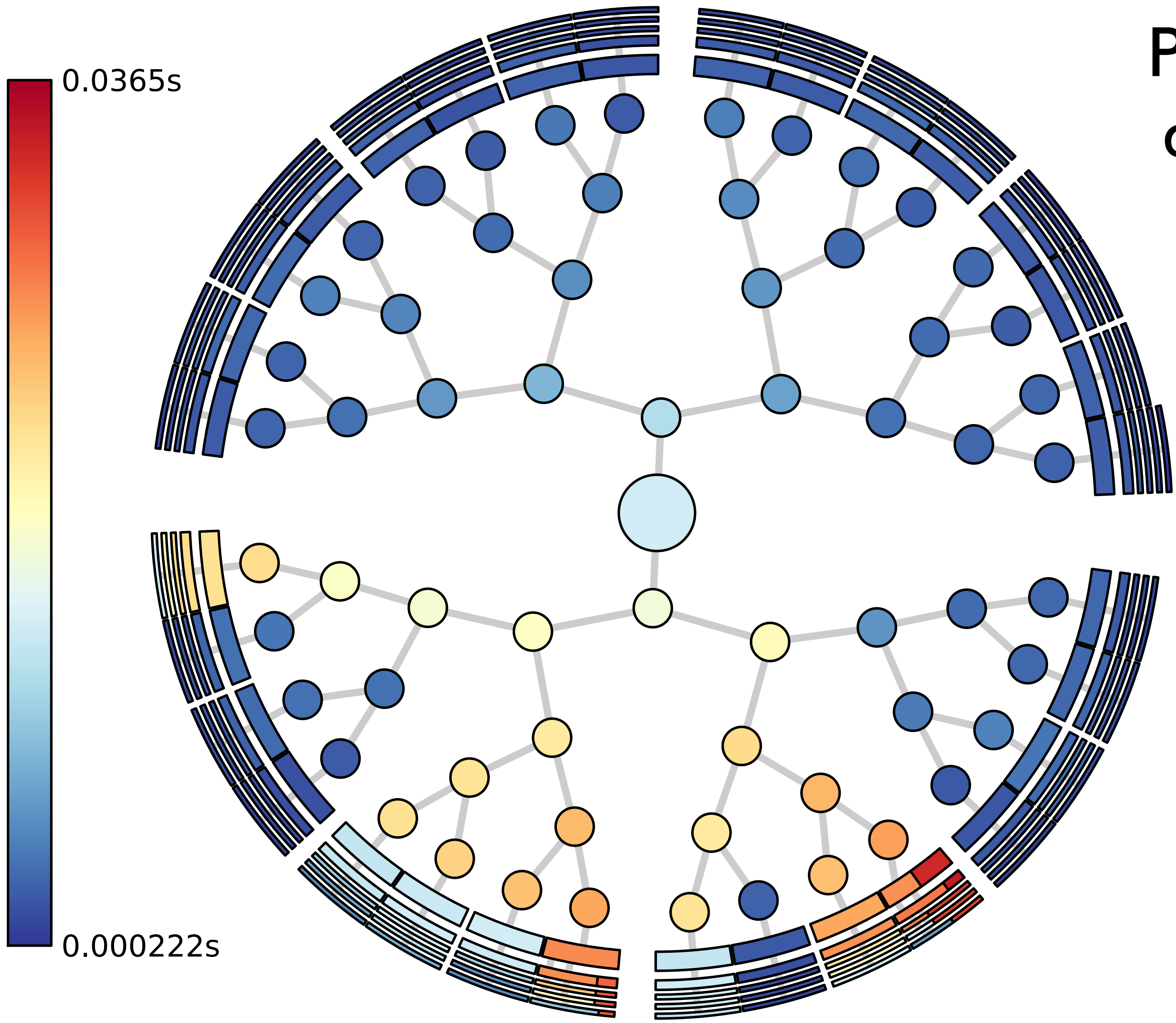
Phase 1 (load distribution) timing data

Scalable view of the comm. graph



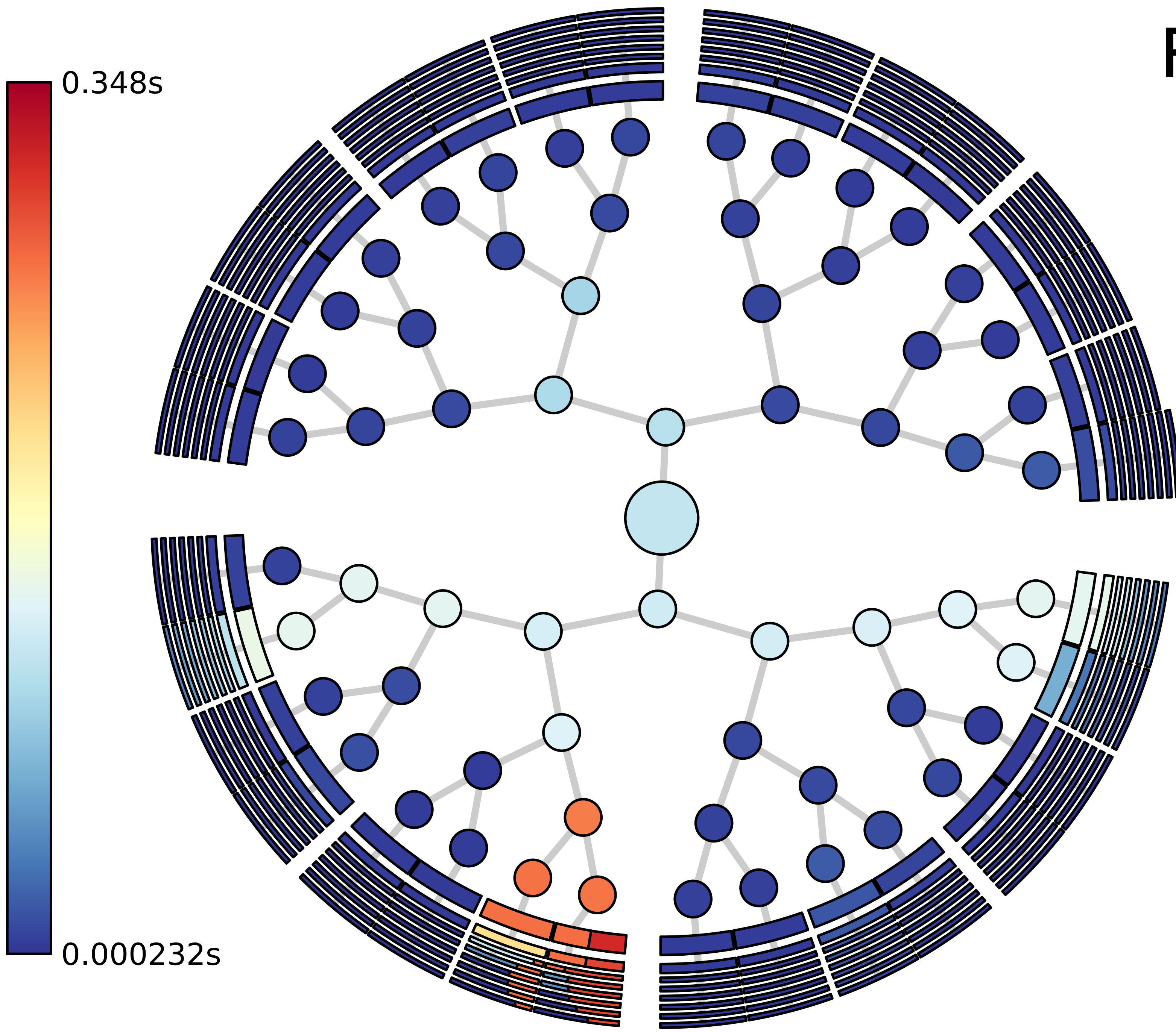
Phase 1 (load distribution) timing data

Scalable view of the comm. graph



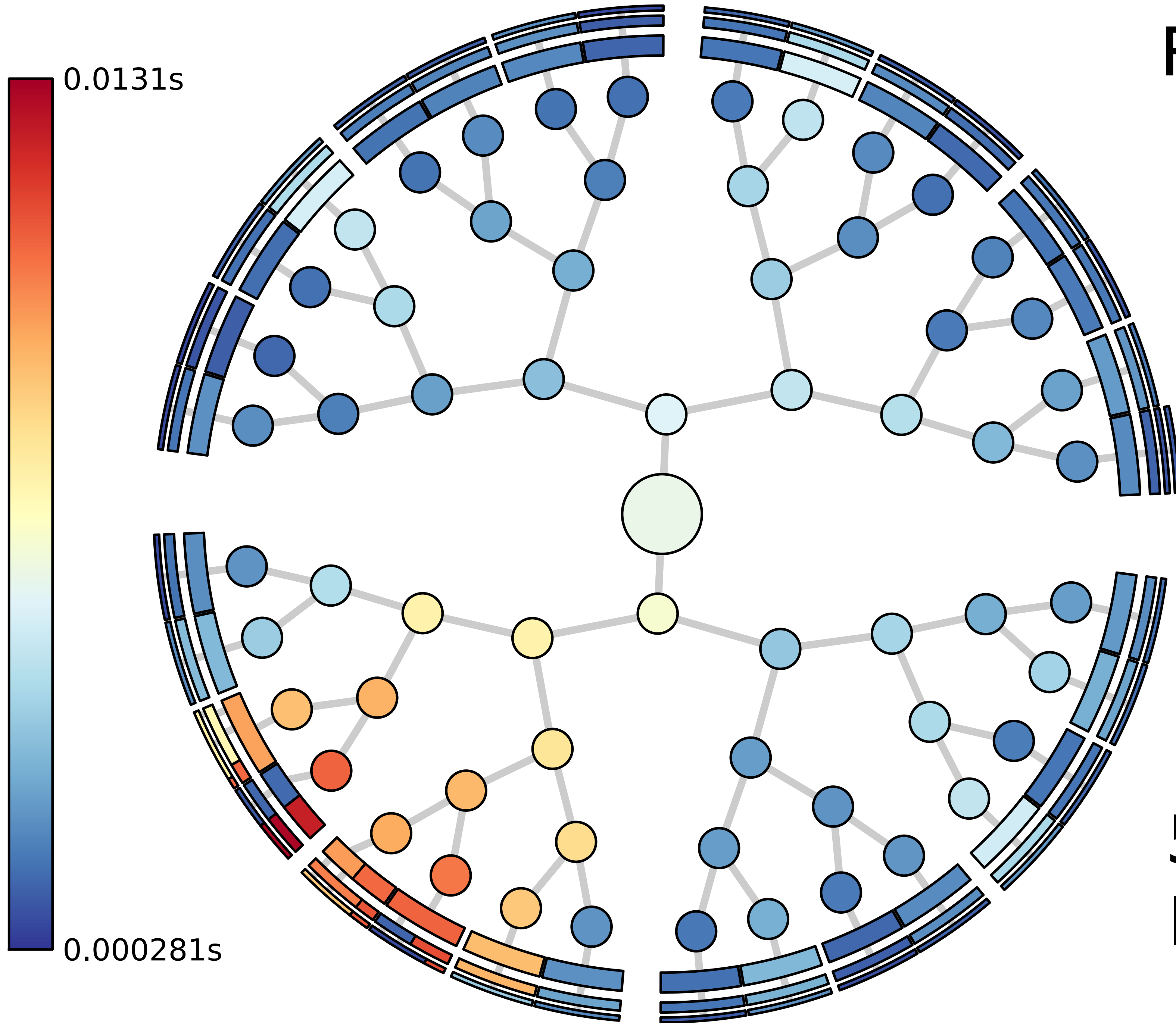
Phase 1 (load distribution) timing data

Scalable view of the comm. graph



Phase 1 (load distribution) timing data

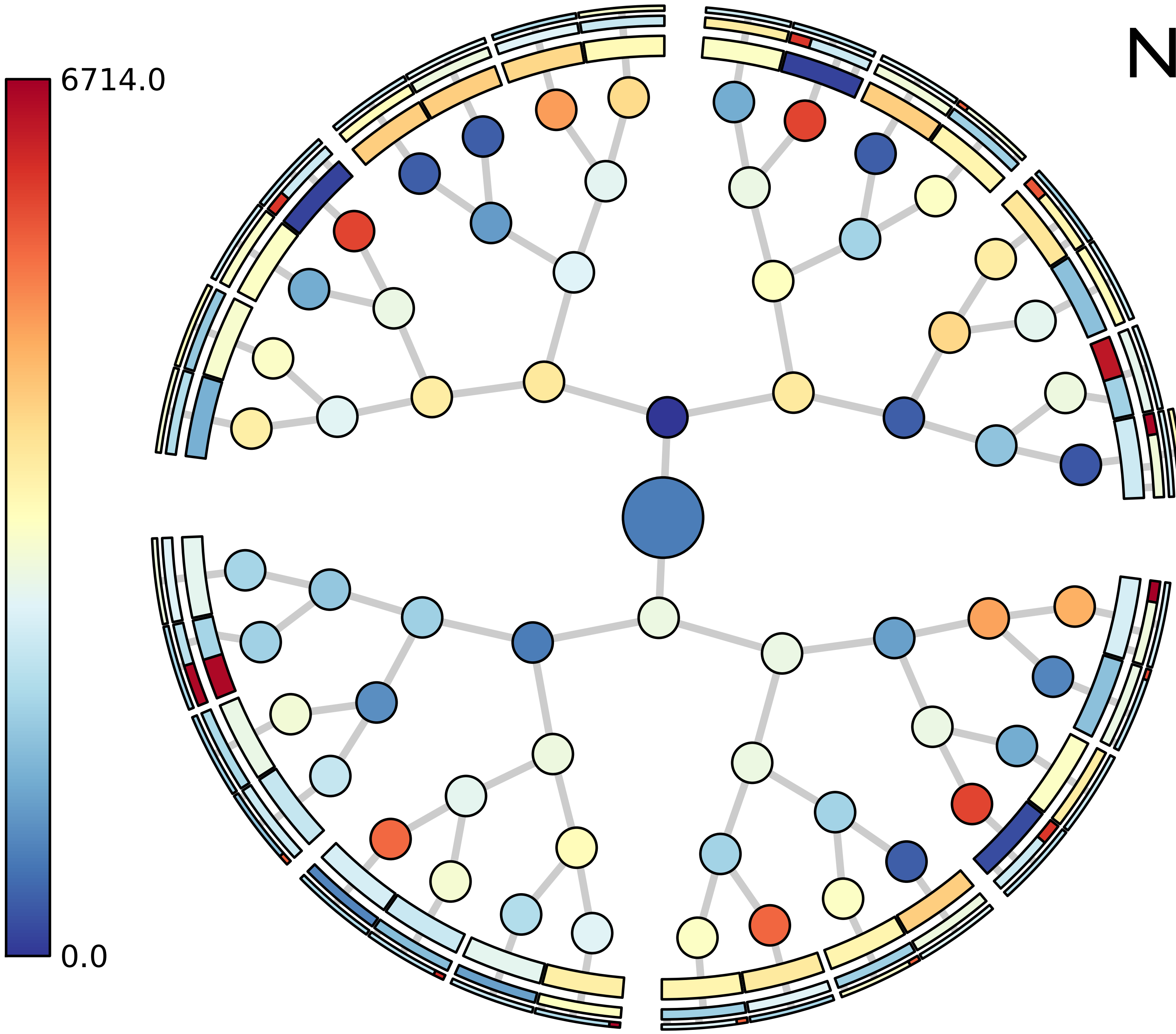
Phase 1 timings for each processor



Phase 1 (*load distribution*) timing data

512 cores of Blue Gene/P

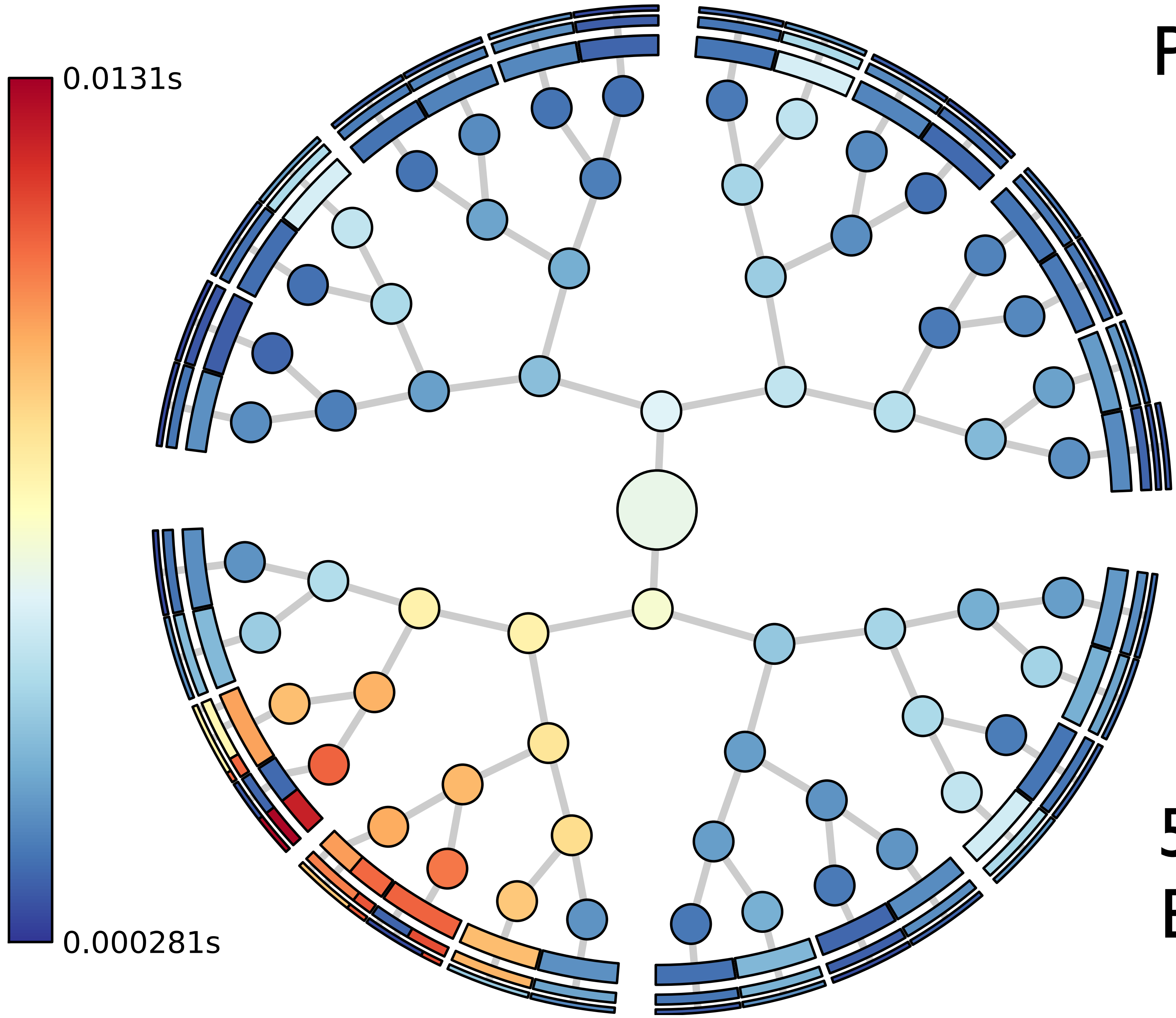
Load on each processor



Number of cells on each processor

512 cores of Blue Gene/P

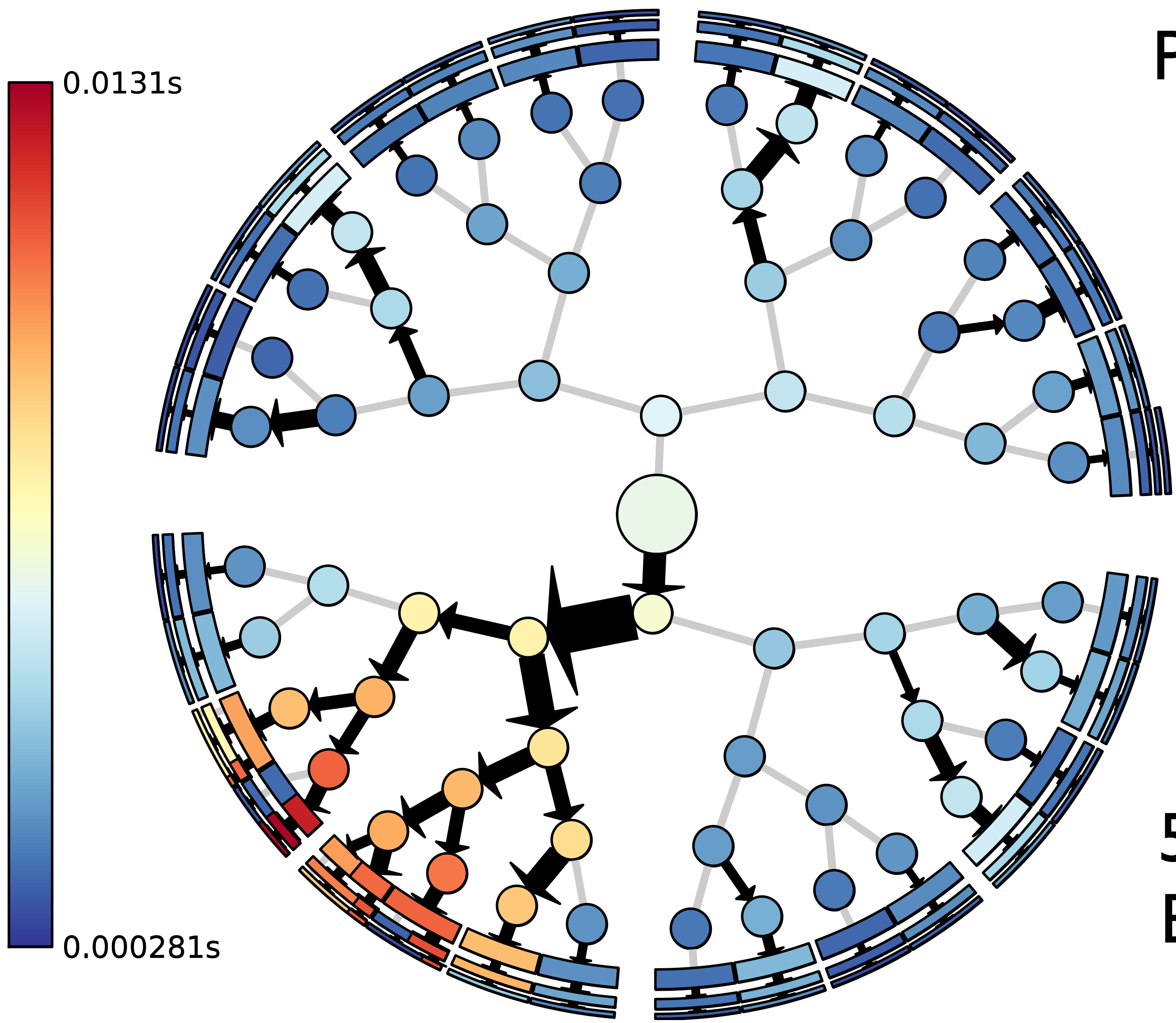
Phase 1 timings for each processor



Phase 1 (*load distribution*) timing data

512 cores of Blue Gene/P

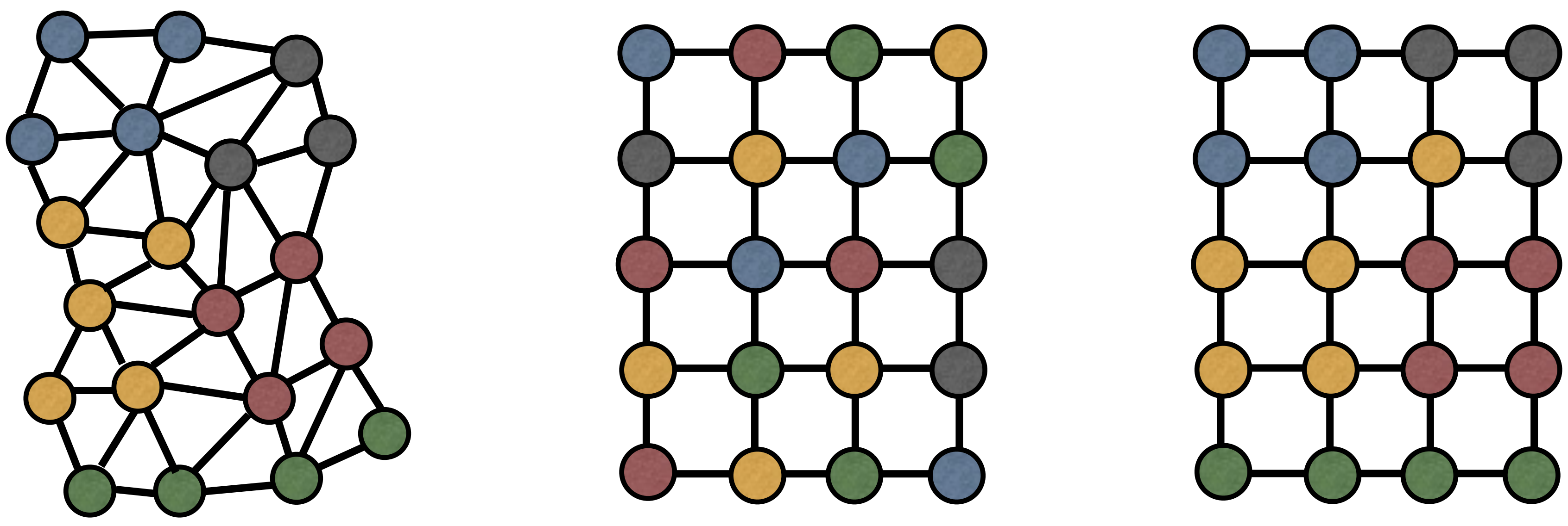
Phase 1 timings for each processor



Phase 1 (*load distribution*) timing data

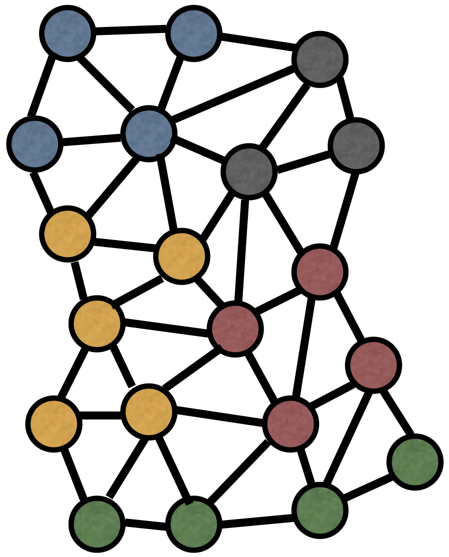
512 cores of Blue Gene/P

TASK MAPPING



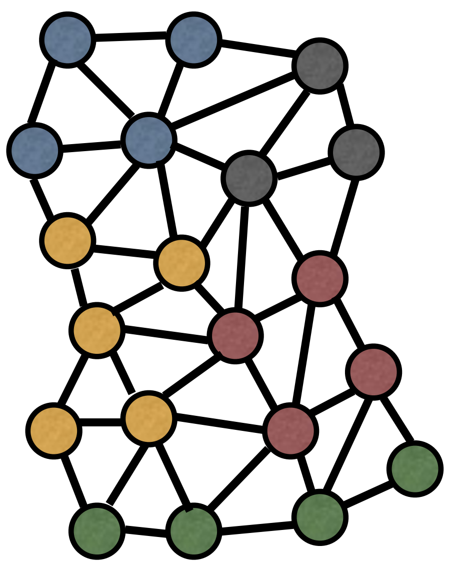
Topology aware task mapping

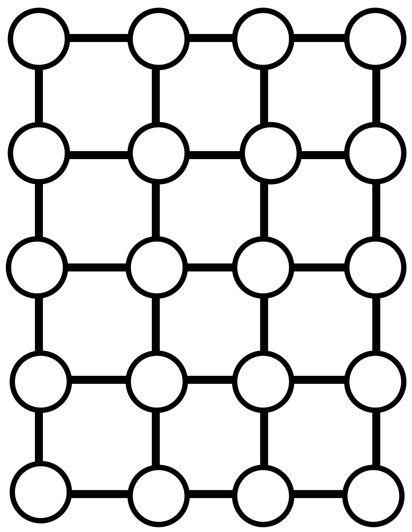
- What is mapping - layout/placement of tasks/processes in an application on the physical interconnect
- Does not require any changes to the application



Topology aware task mapping

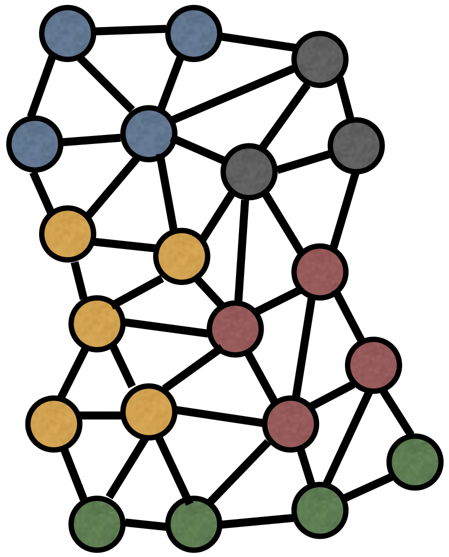
- What is mapping - layout/placement of tasks/processes in an application on the physical interconnect
- Does not require any changes to the application

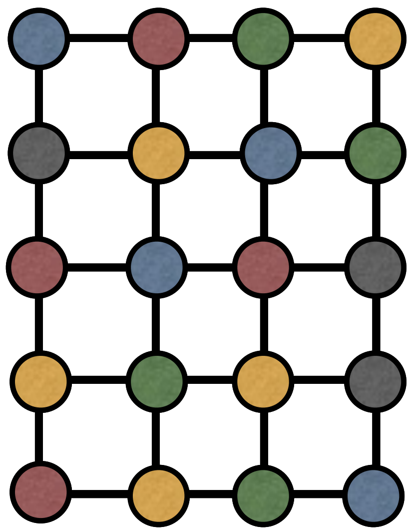




Topology aware task mapping

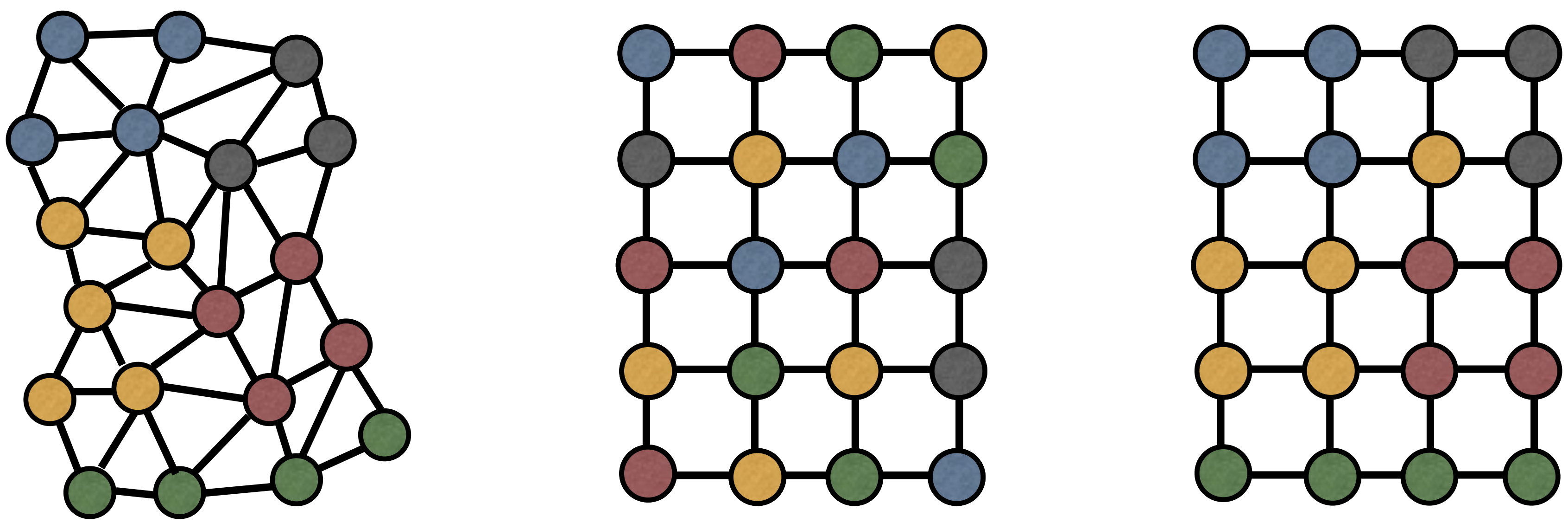
- What is mapping - layout/placement of tasks/processes in an application on the physical interconnect
- Does not require any changes to the application





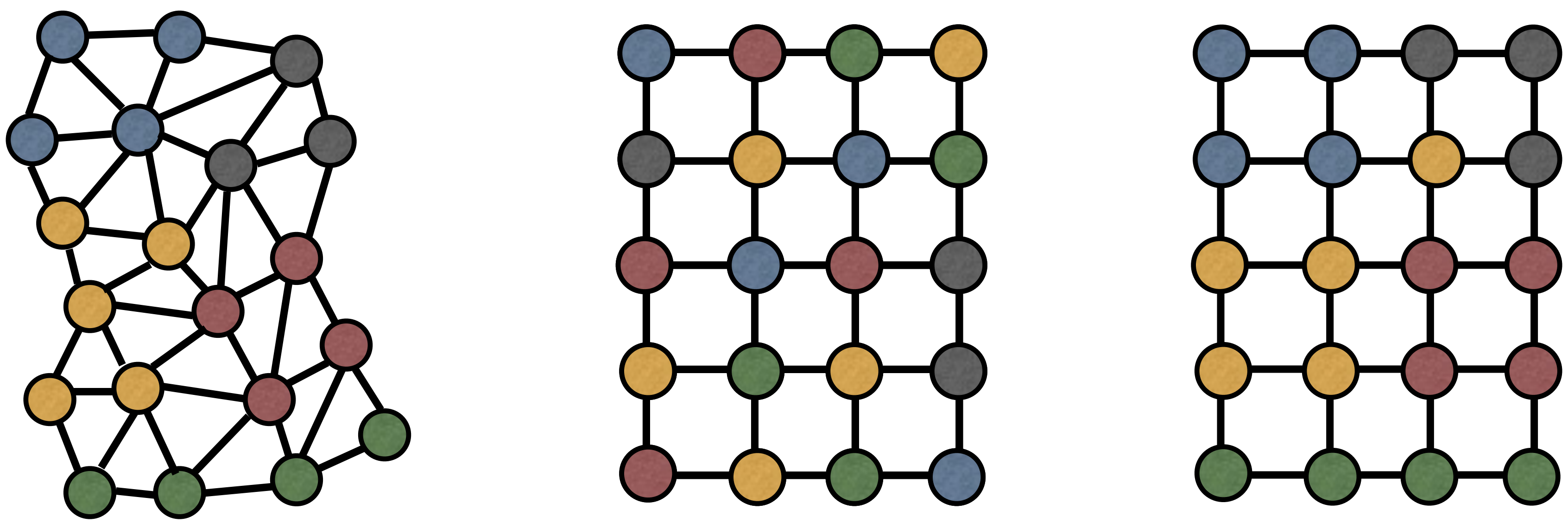
Topology aware task mapping

- What is mapping - layout/placement of tasks/processes in an application on the physical interconnect
- Does not require any changes to the application



Topology aware task mapping

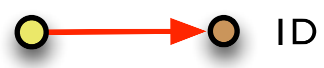
- What is mapping - layout/placement of tasks/processes in an application on the physical interconnect
- Does not require any changes to the application



- Goals:
  - Balance computational load
  - Minimize contention (optimize latency or bandwidth)

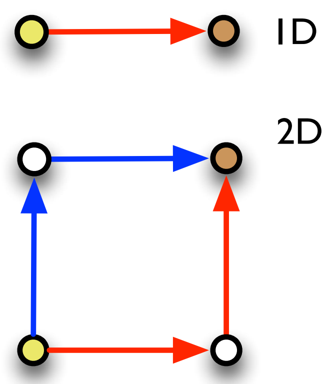
Maximize bandwidth?

- Traditionally, research has focused on bringing tasks closer to reduce the number of hops
  - Minimizes latency, but more importantly link contention
- For applications that send large messages this might not be optimal



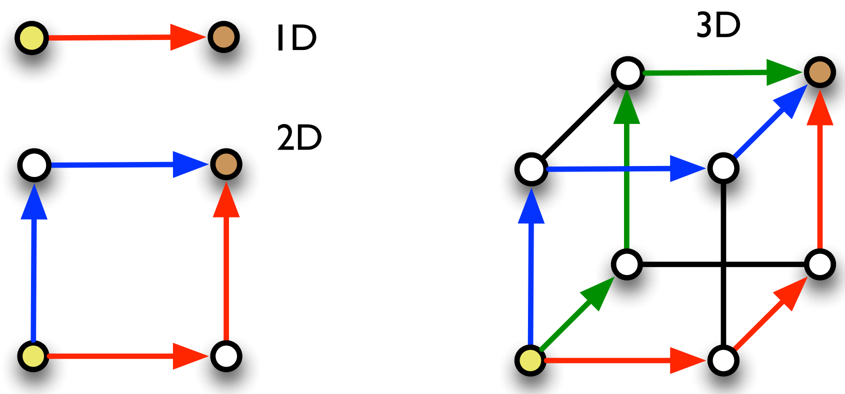
Maximize bandwidth?

- Traditionally, research has focused on bringing tasks closer to reduce the number of hops
  - Minimizes latency, but more importantly link contention
- For applications that send large messages this might not be optimal



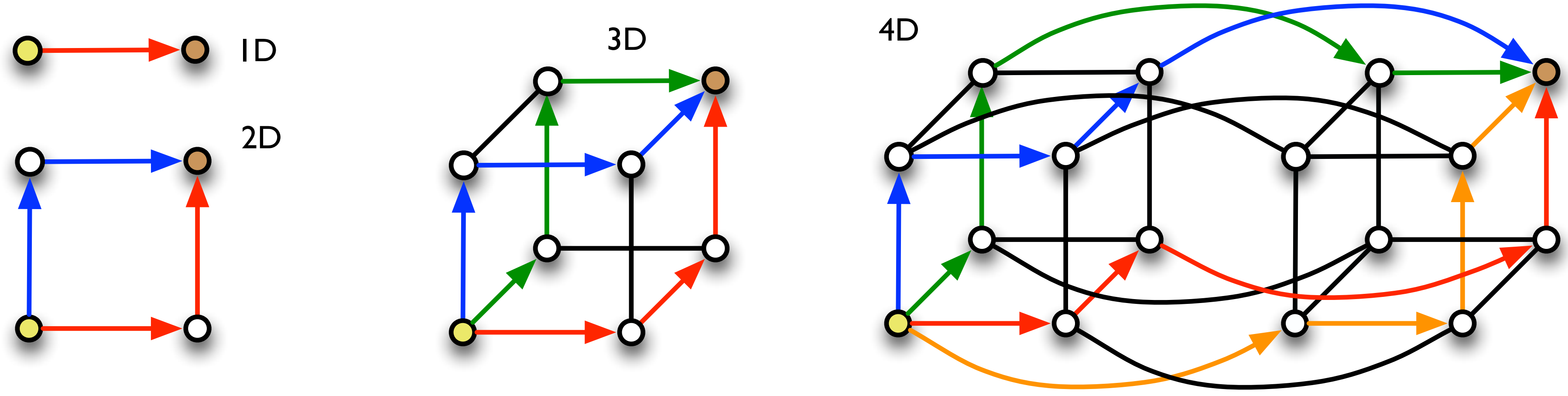
Maximize bandwidth?

- Traditionally, research has focused on bringing tasks closer to reduce the number of hops
  - Minimizes latency, but more importantly link contention
- For applications that send large messages this might not be optimal



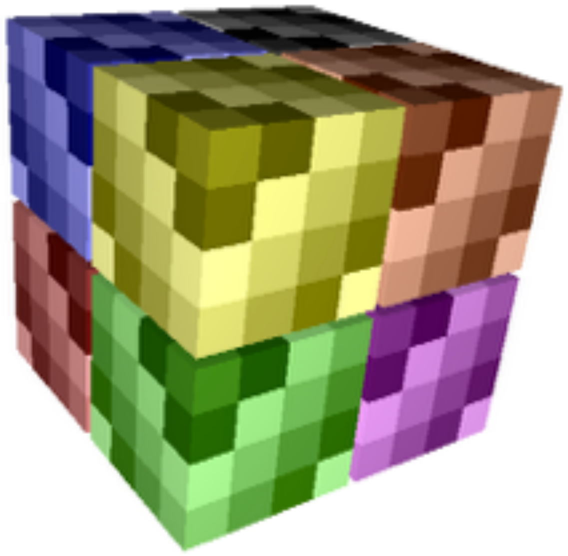
Maximize bandwidth?

- Traditionally, research has focused on bringing tasks closer to reduce the number of hops
  - Minimizes latency, but more importantly link contention
- For applications that send large messages this might not be optimal



Rubik

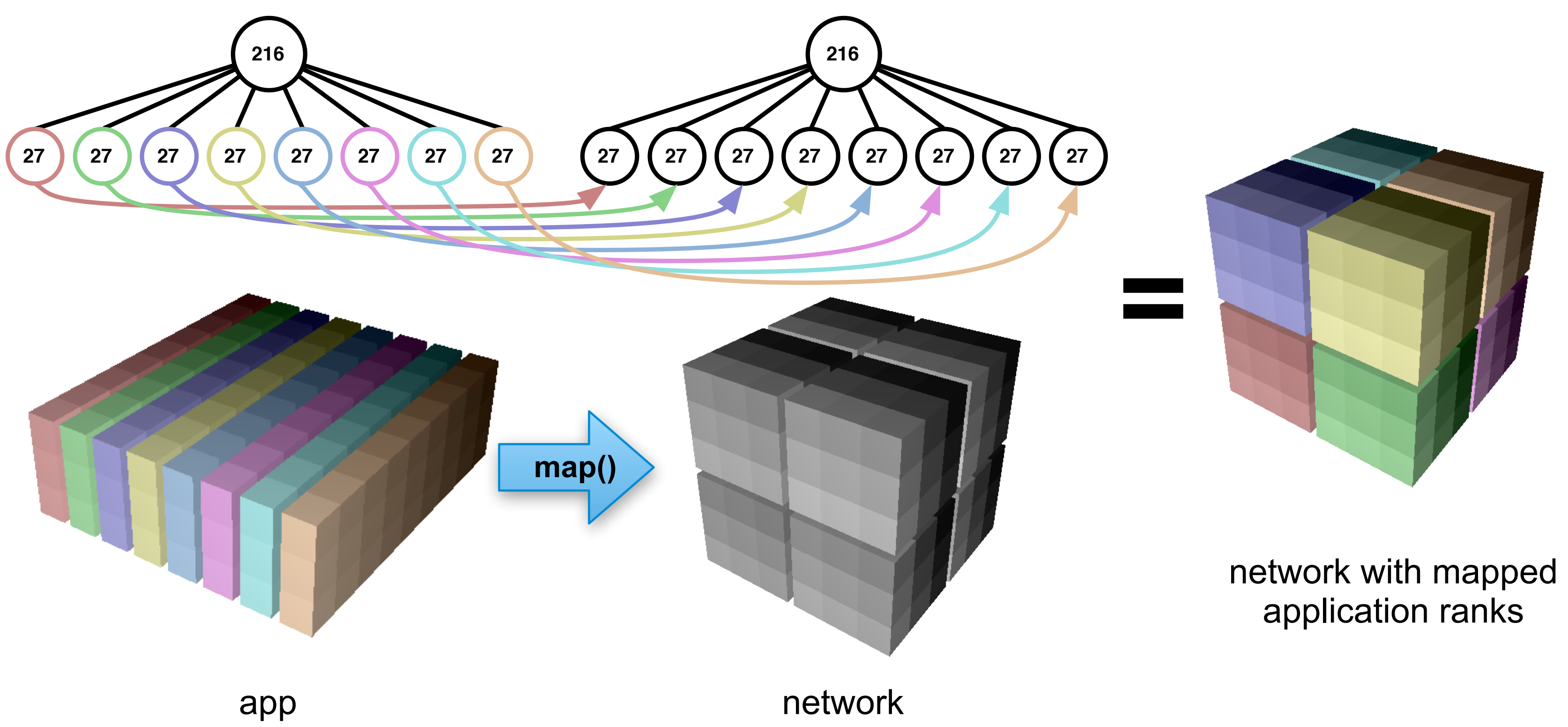
- We have developed a mapping tool focusing on:
  - structured applications that are bandwidth-bound, use collectives over sub-communicators
  - built-in operations that can increase effective bandwidth on torus networks based on heuristics
- Input:
  - Application topology with subsets identified
  - Processor topology
  - Set of operations to perform



- Output: map file for job launcher

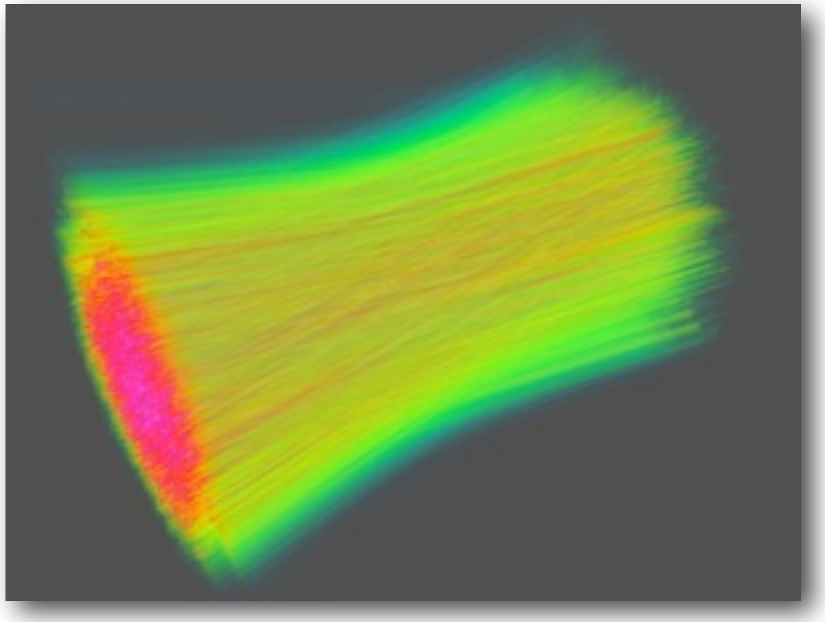
Application example

```
app = box([9,3,8]) # Create app partition tree of 27-task planes
app.tile([9,3,1])

network = box([6,6,6]) # Create network partition tree of 27-processor cubes
network.tile([3,3,3])

network.map(app)  # Map task planes into cubes
```



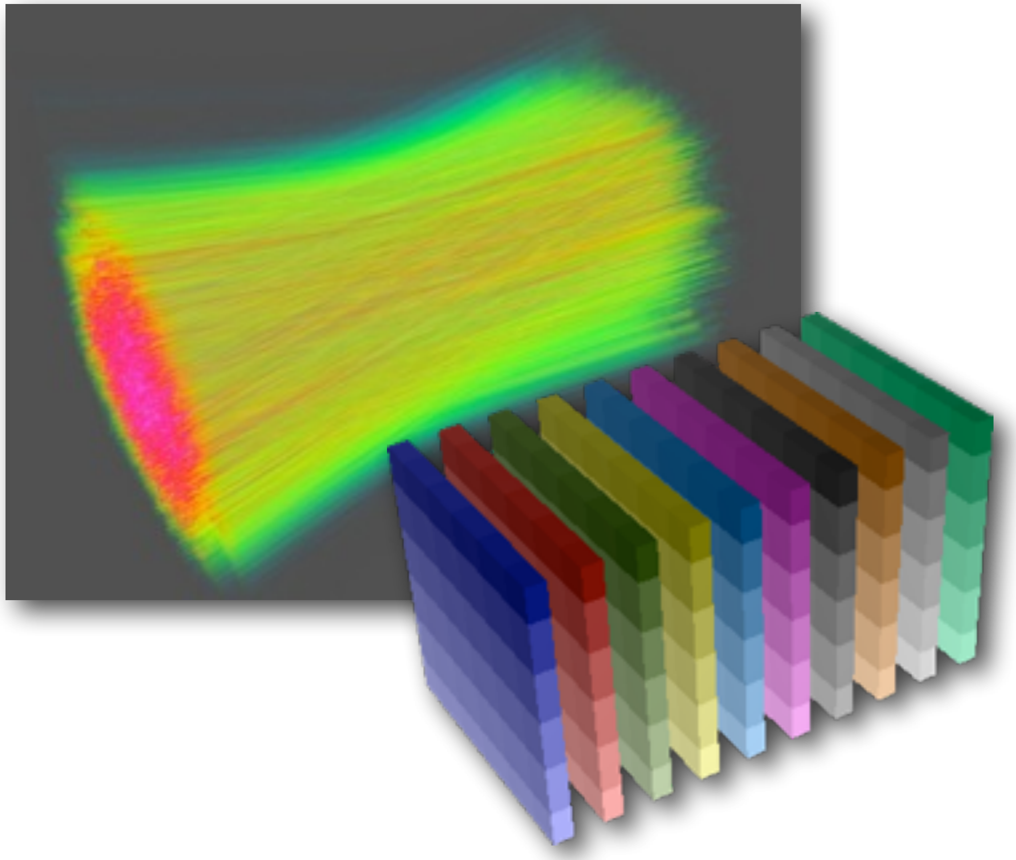
Mapping pF3D

- A laser-plasma interaction code used at the National Ignition Facility (NIF) at LLNL
- Three communication phases over a 3D virtual topology:
  - Wave propagation and coupling: 2D FFTs within XY planes
  - Light advection: Send-recv between consecutive XY planes
  - Hydrodynamic equations: 3D near-neighbor exchange



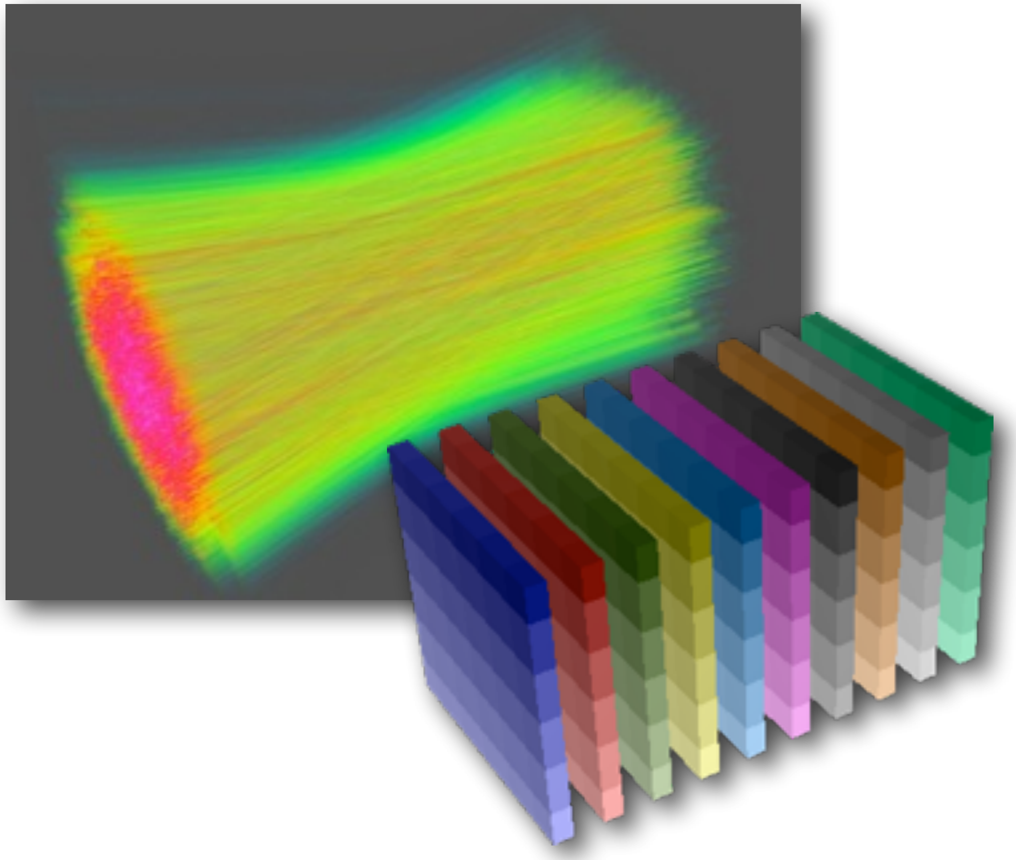
Mapping pF3D

- A laser-plasma interaction code used at the National Ignition Facility (NIF) at LLNL
- Three communication phases over a 3D virtual topology:
  - Wave propagation and coupling: 2D FFTs within XY planes
  - Light advection: Send-recv between consecutive XY planes
  - Hydrodynamic equations: 3D near-neighbor exchange



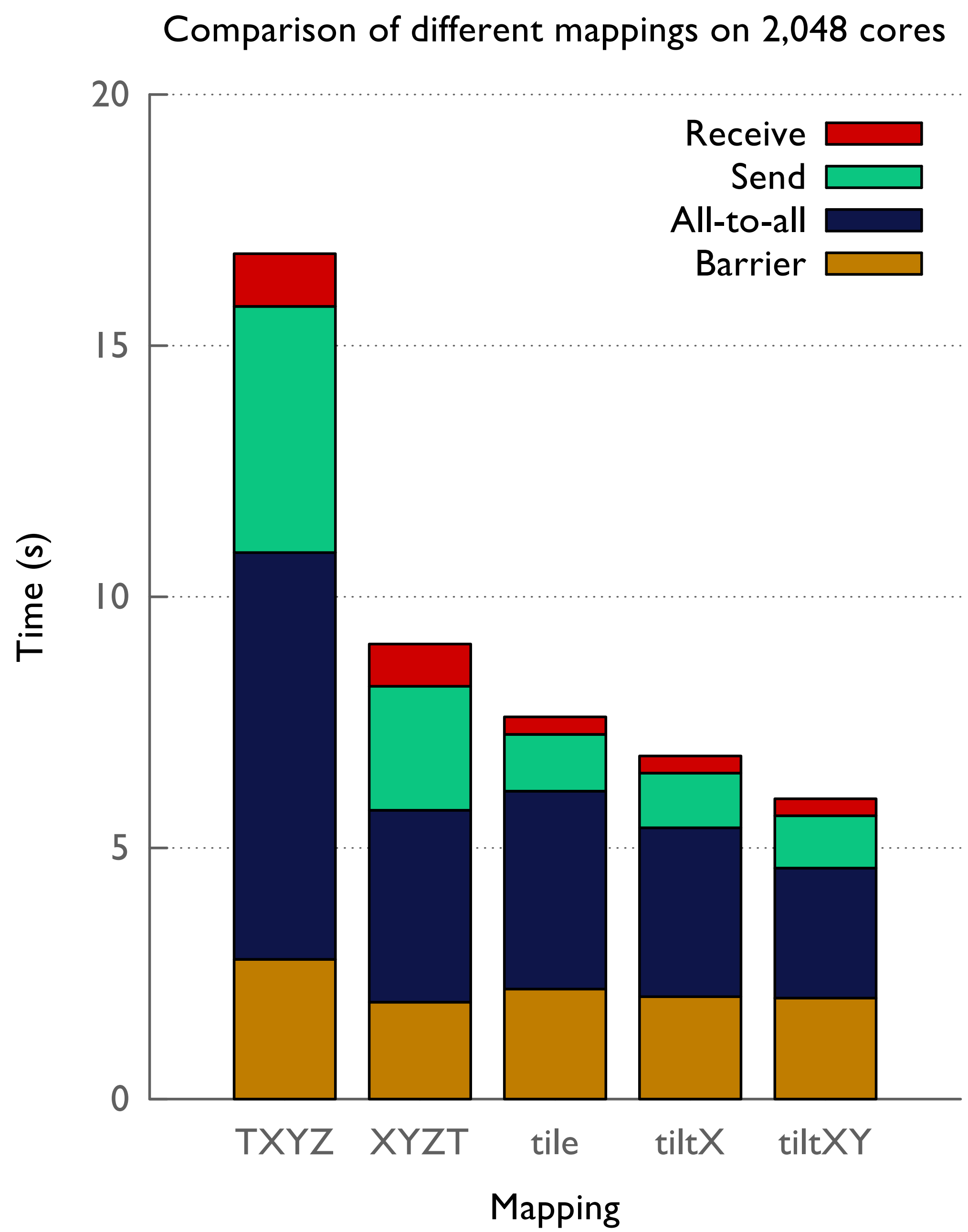
Mapping pF3D

- A laser-plasma interaction code used at the National Ignition Facility (NIF) at LLNL
- Three communication phases over a 3D virtual topology:
  - Wave propagation and coupling: 2D FFTs within XY planes
  - Light advection: Send-recv between consecutive XY planes
  - Hydrodynamic equations: 3D near-neighbor exchange



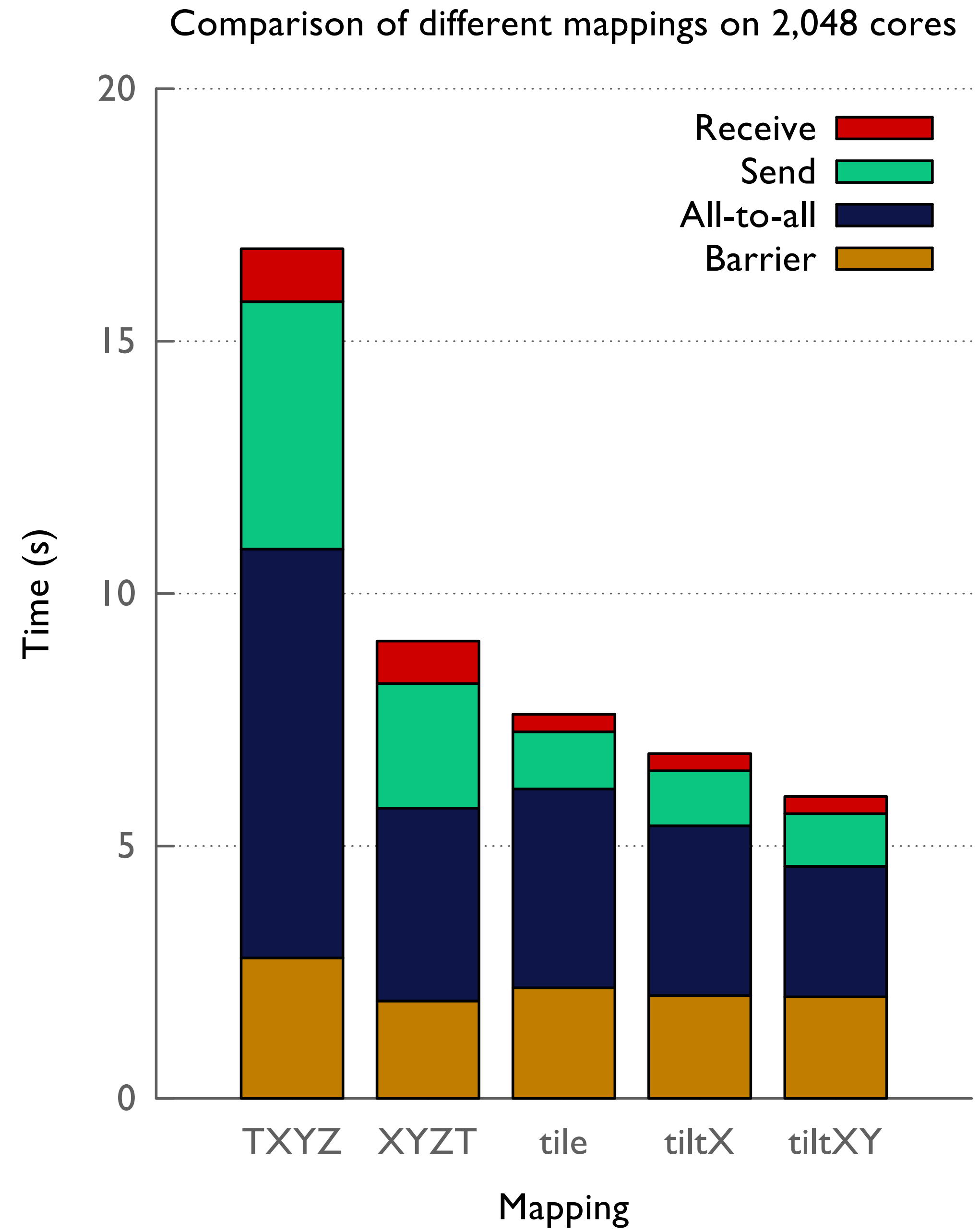
	2048 cores		16384 cores	
MPI call	**Total %**	**MPI %**	**Total %**	**MPI %**
Send	4.90	28.45	23.10	57.21
Alltoall	8.10	46.94	7.30	18.07
Barrier	2.78	16.10	8.13	20.15

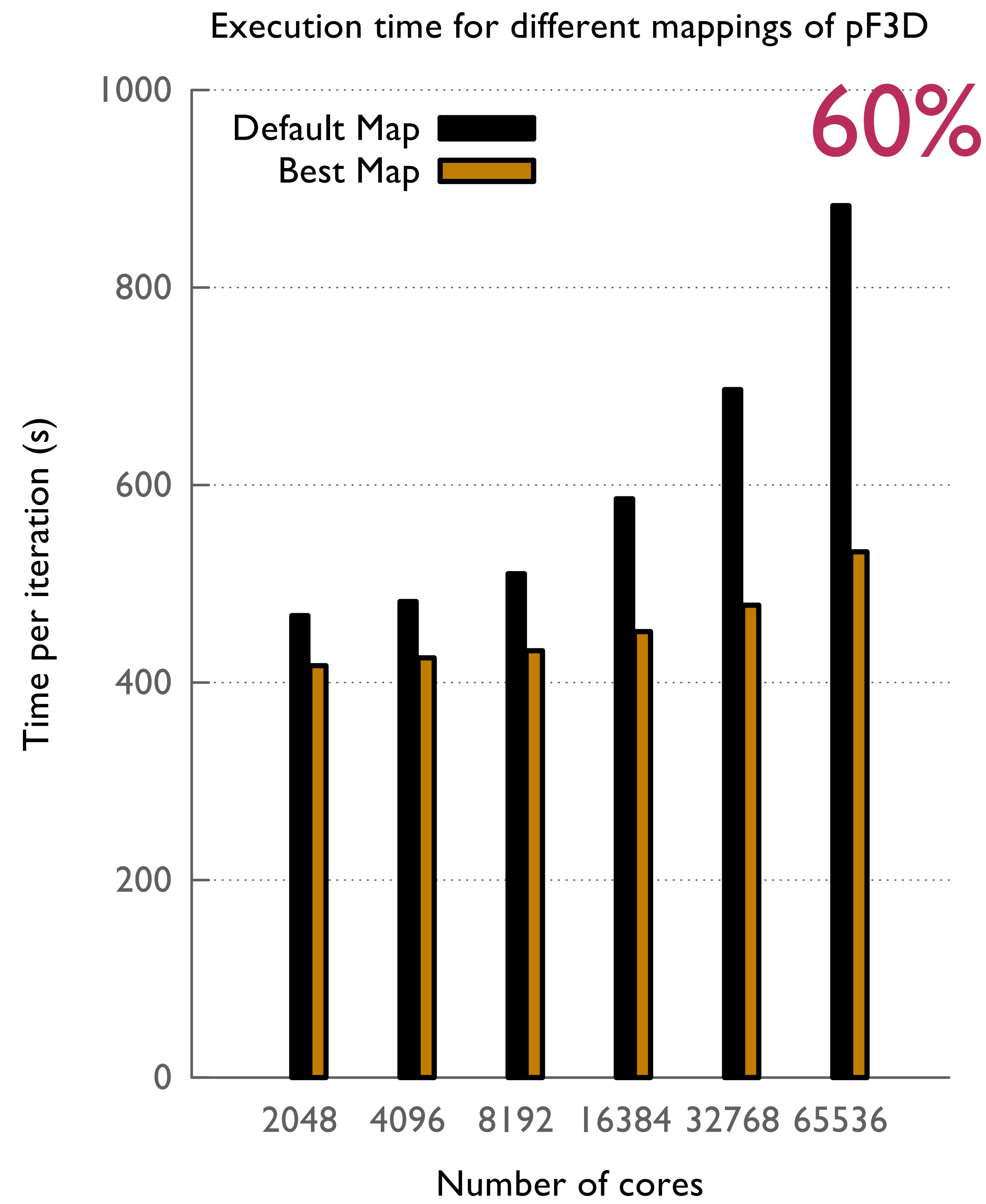
Performance benefits



A. Bhatele et al. Mapping applications with collectives over sub-communicators on torus networks. In *Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '12. November 2012. LLNL-CONF-556491.

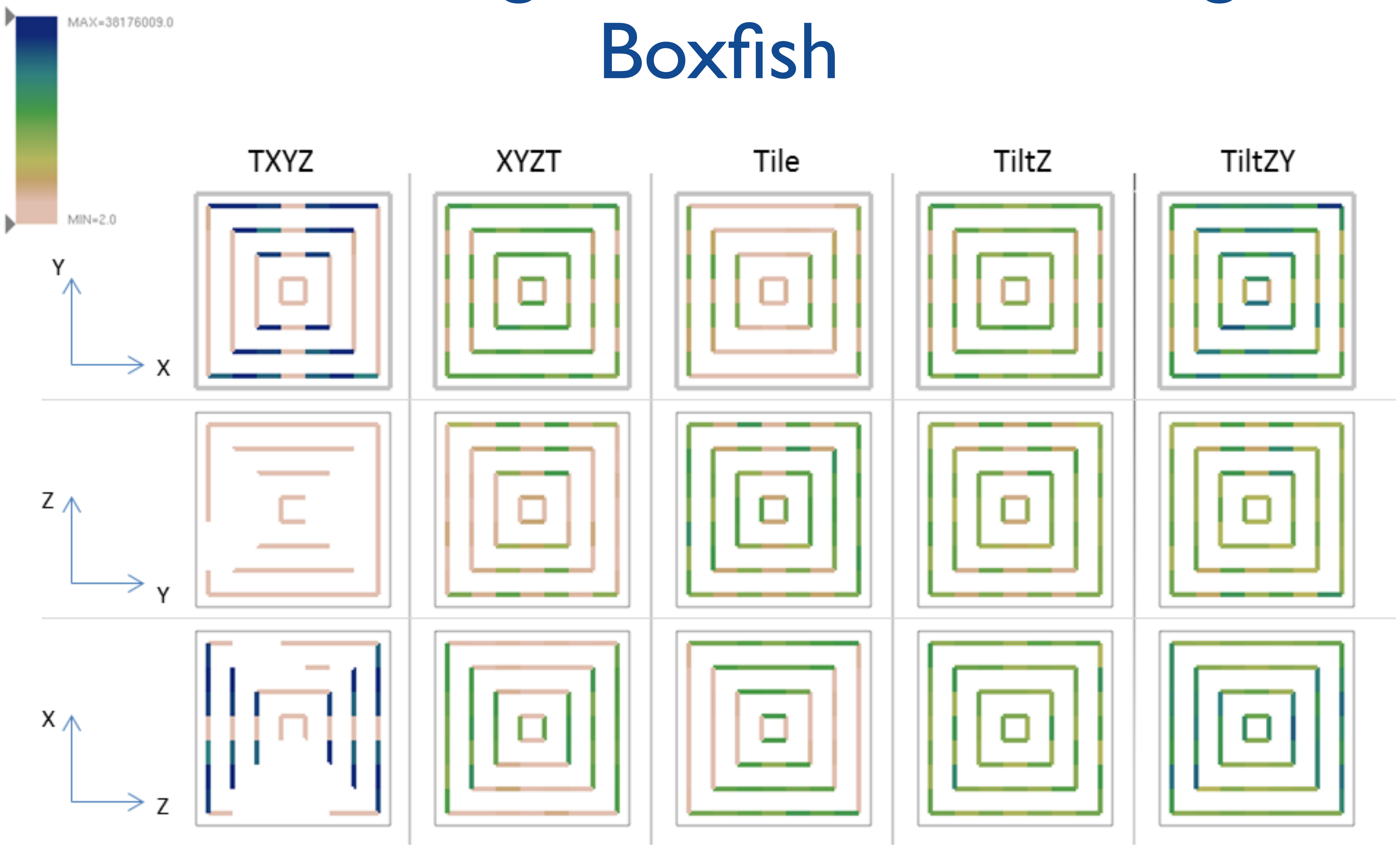
Performance benefits



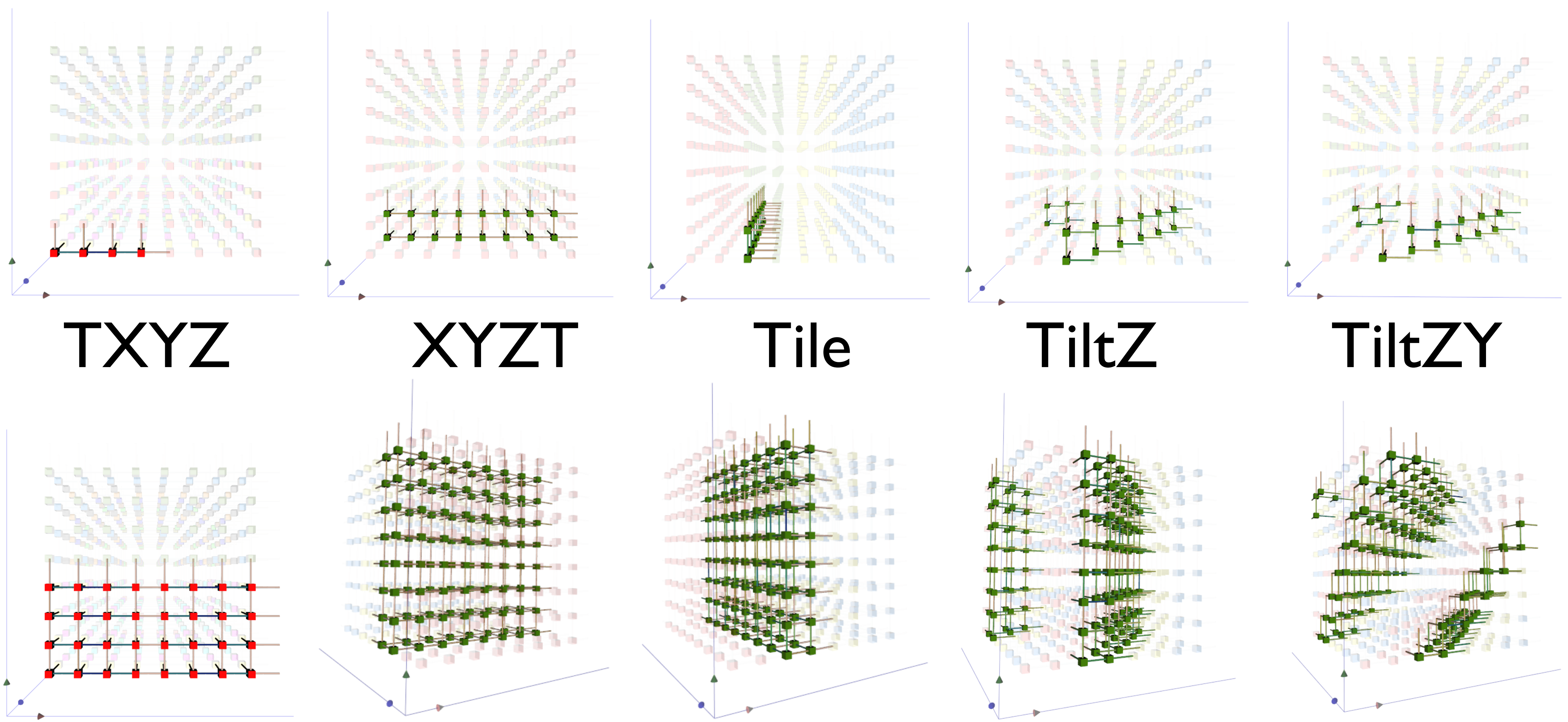


A. Bhatele et al. Mapping applications with collectives over sub-communicators on torus networks. In *Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '12. November 2012. LLNL-CONF-556491.

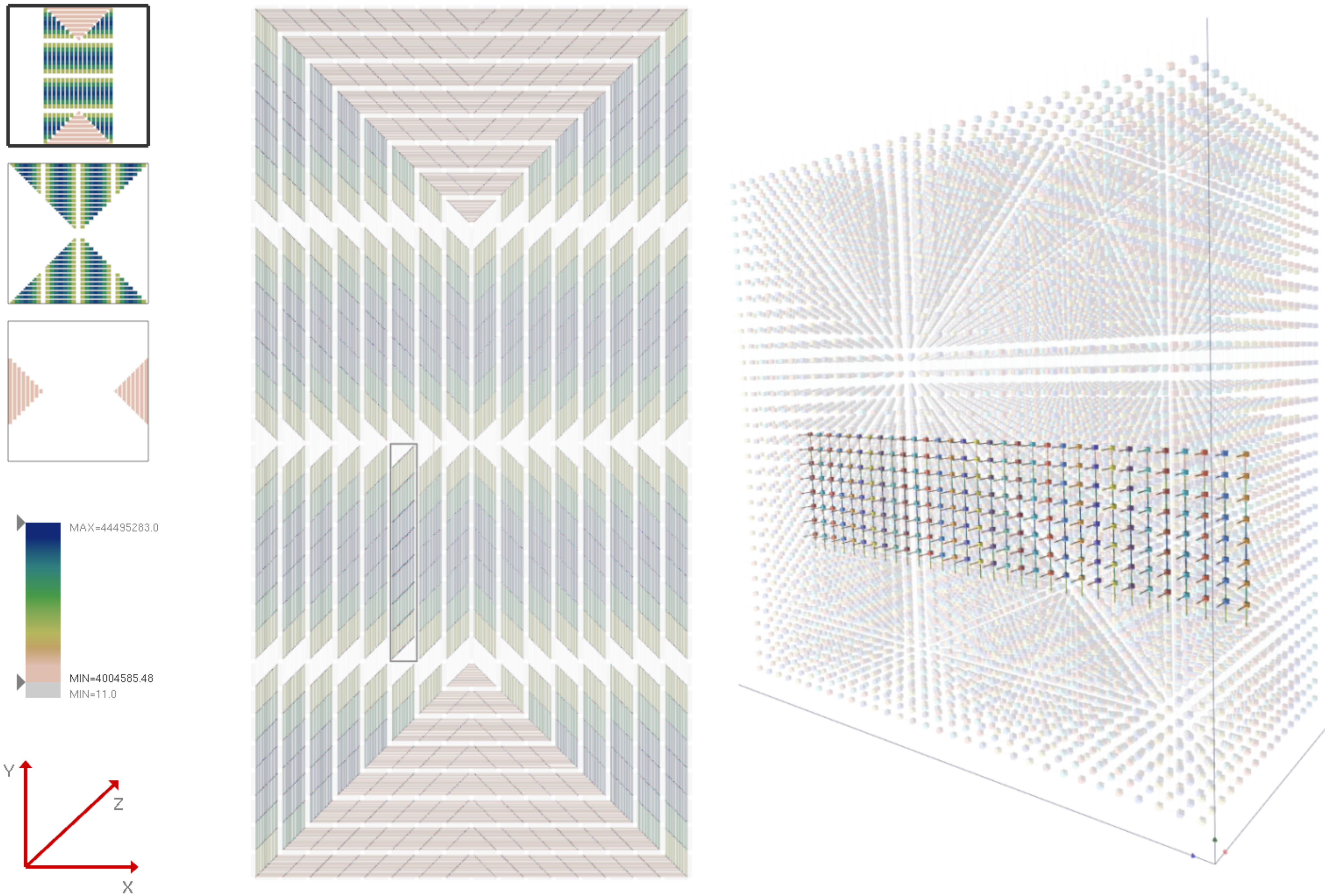
Visualizing network traffic using Boxfish



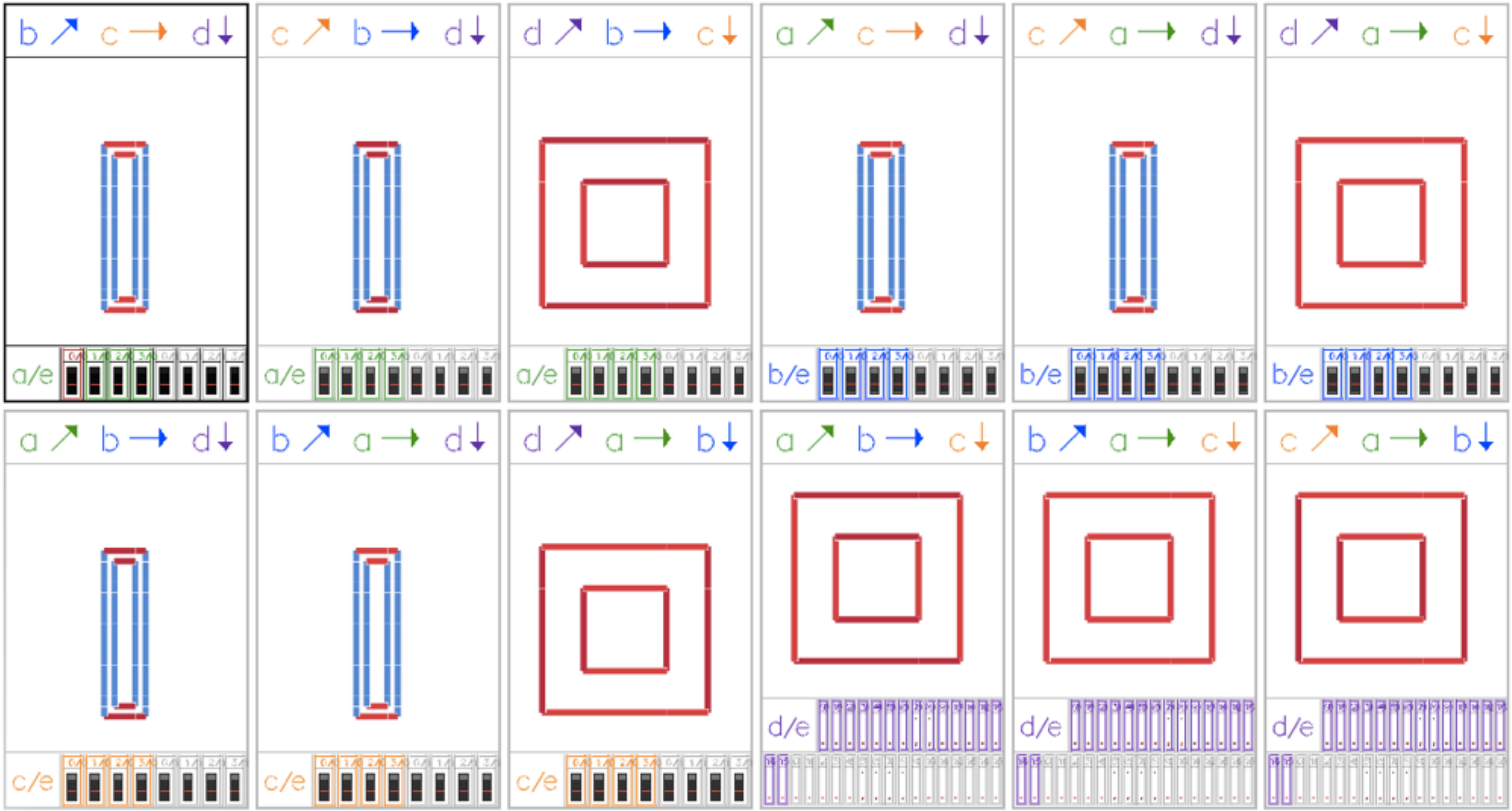
Visualize sub-communicators



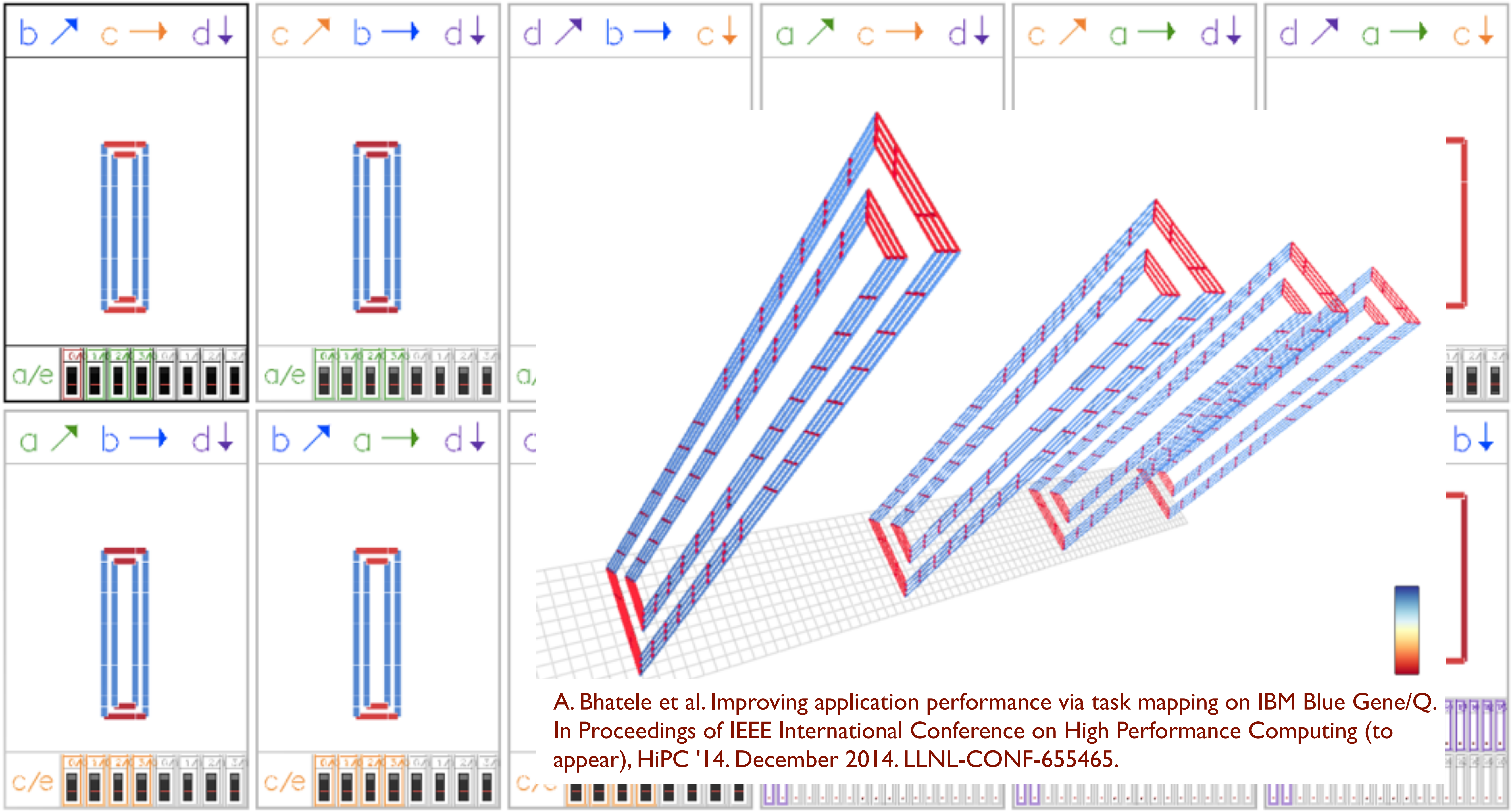
Detailed 2D and 3D views



MILC on Blue Gene/Q

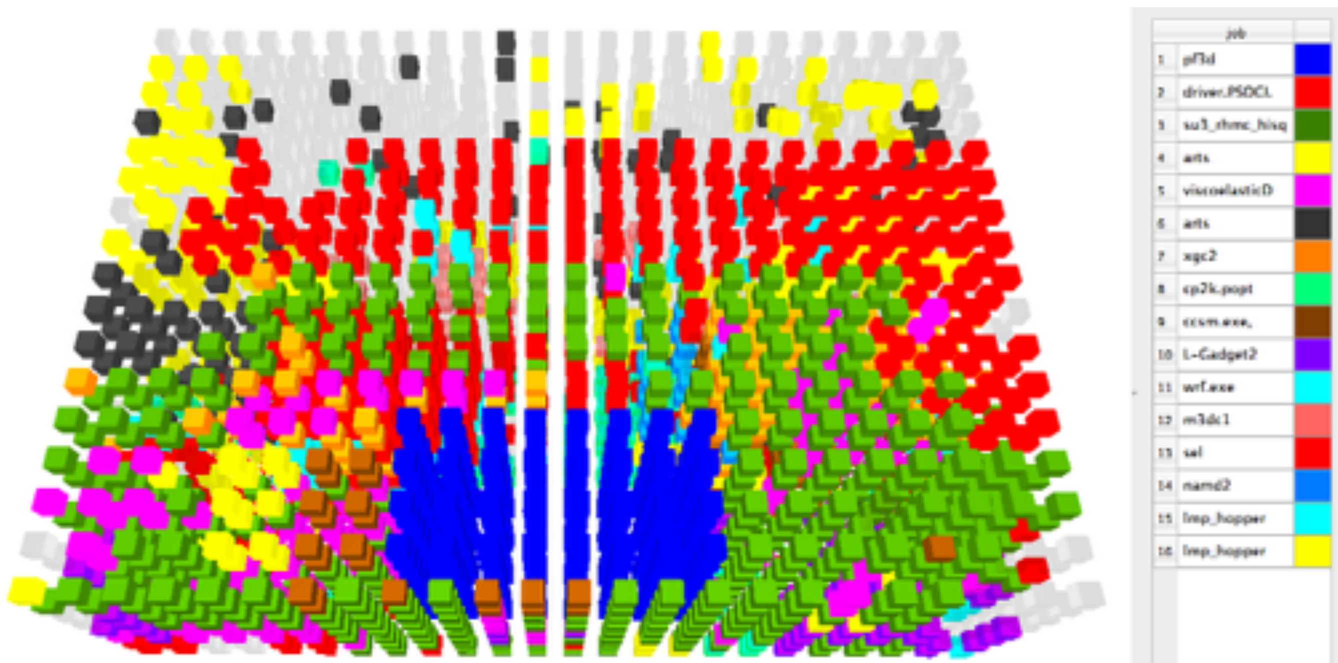


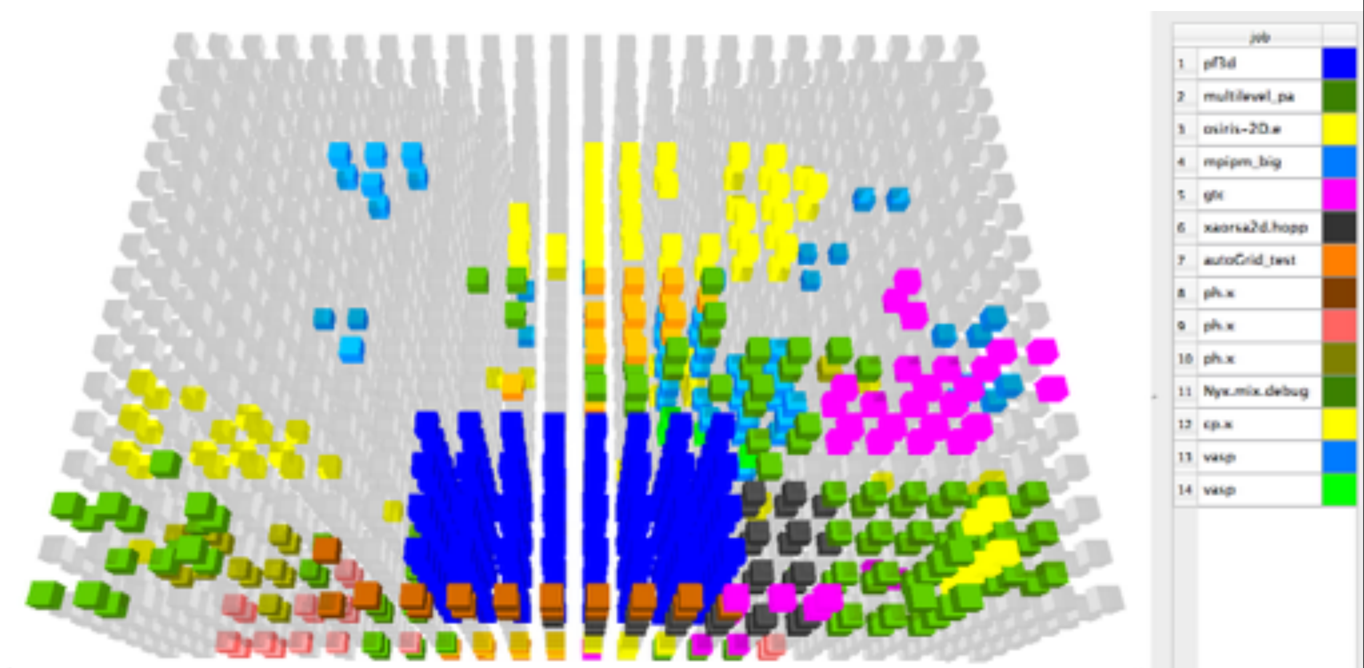
MILC on Blue Gene/Q



A. Bhatele et al. Improving application performance via task mapping on IBM Blue Gene/Q. In Proceedings of IEEE International Conference on High Performance Computing (to appear), HiPC '14. December 2014. LLNL-CONF-655465.

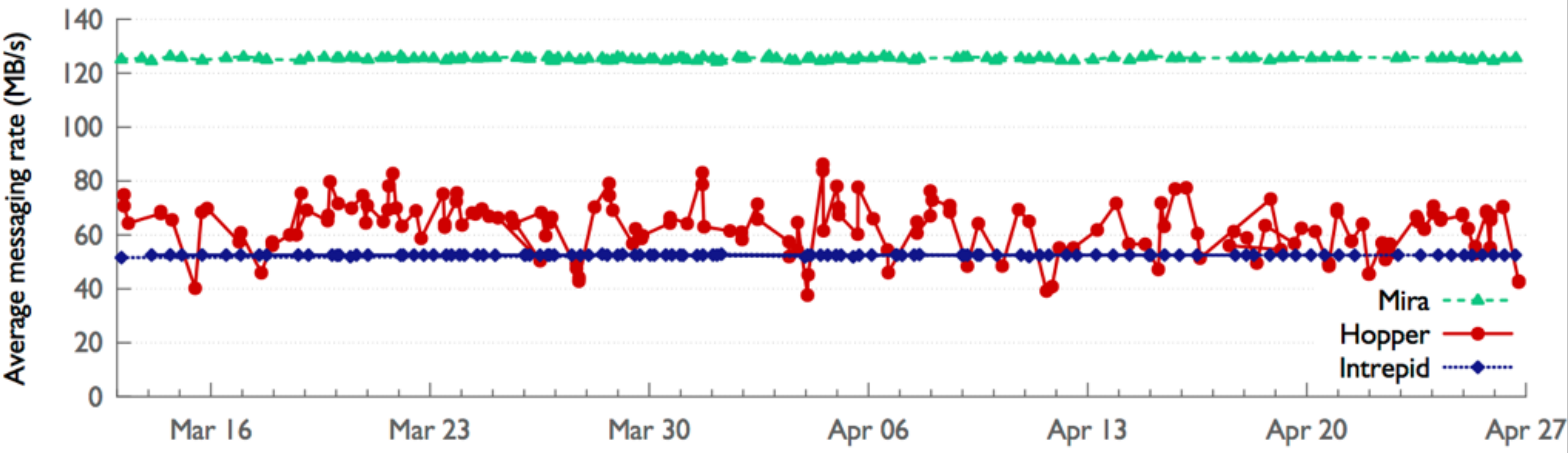
JOB PLACEMENT & ROUTING





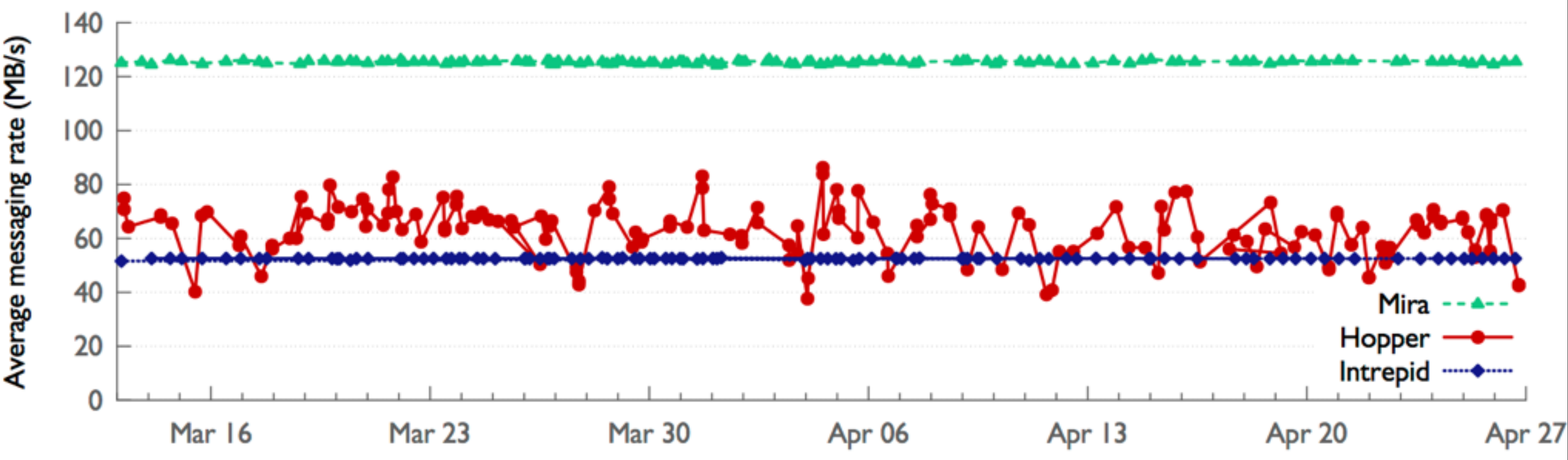
Performance variability

Average messaging rates for batch jobs running a laser-plasma interaction code



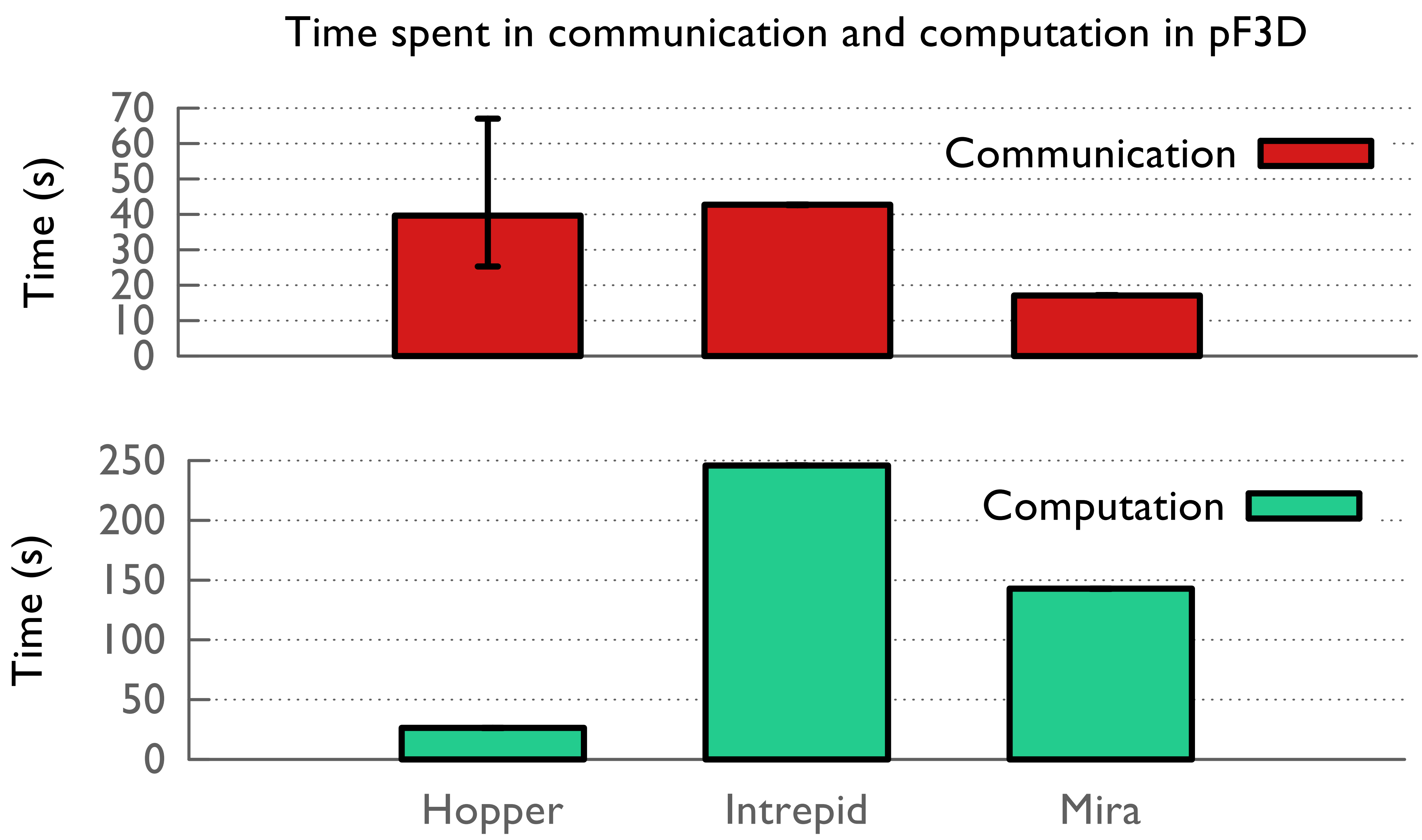
Performance variability

Average messaging rates for batch jobs running a laser-plasma interaction code

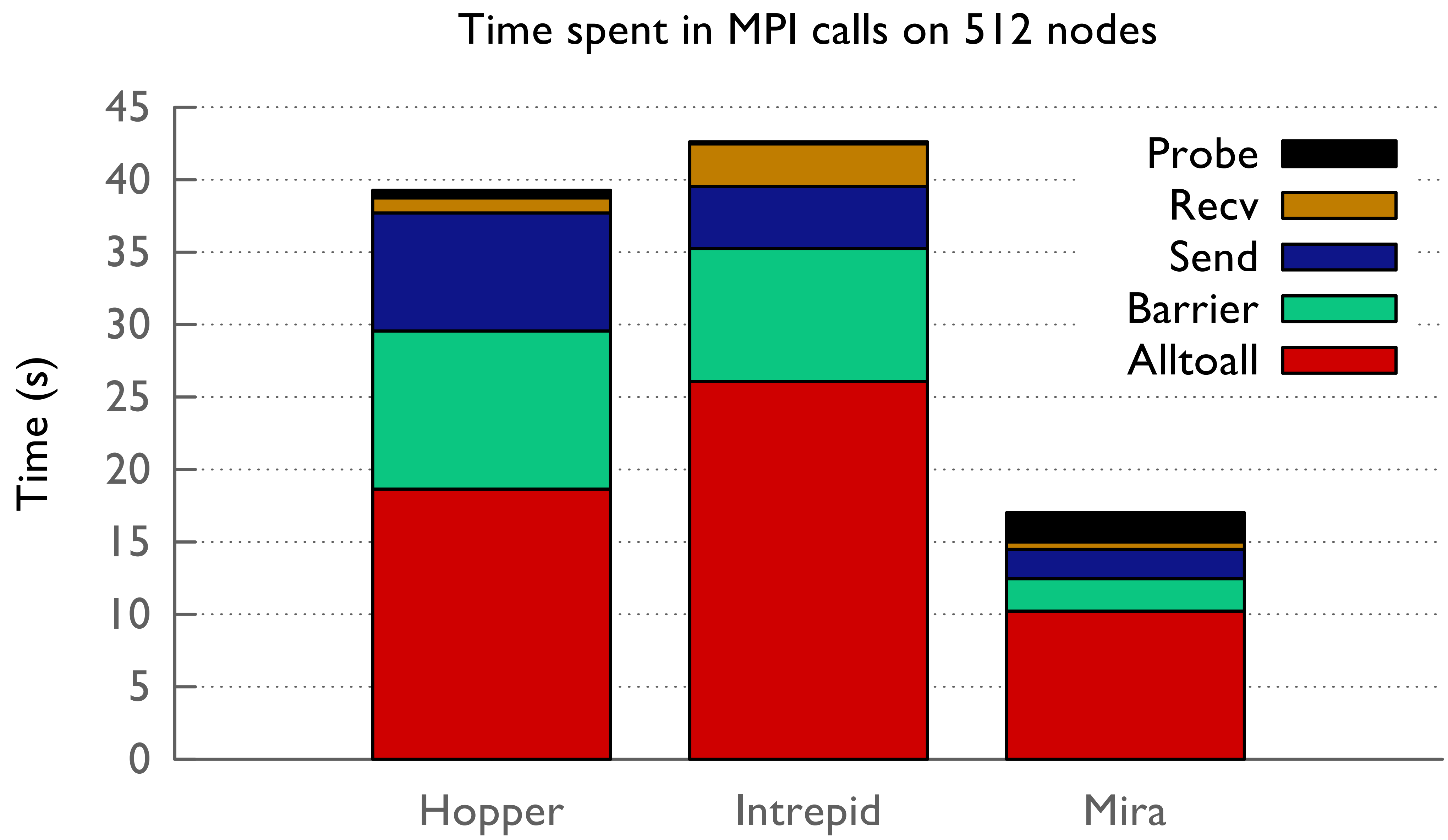


$$\frac{\text{Total number of bytes sent on the network}}{\text{Time spent sending the messages}}$$

pF3D characterization



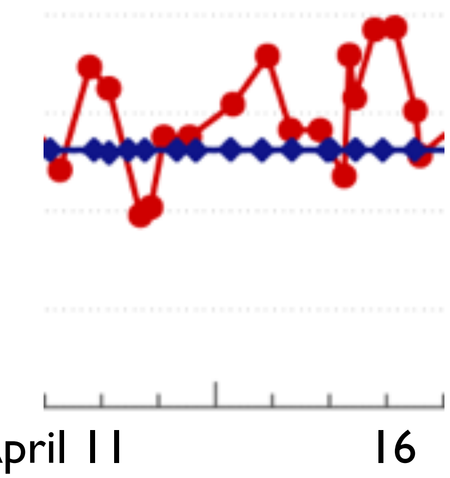
pF3D characterization

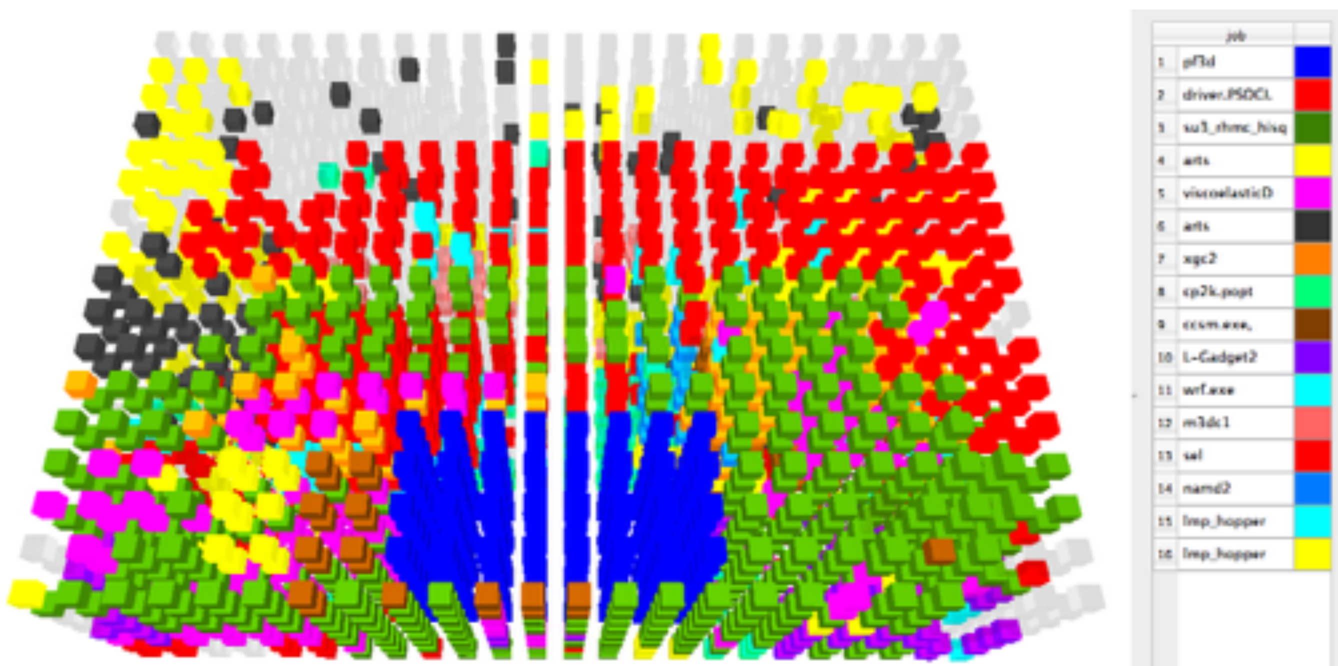


Sources of variability

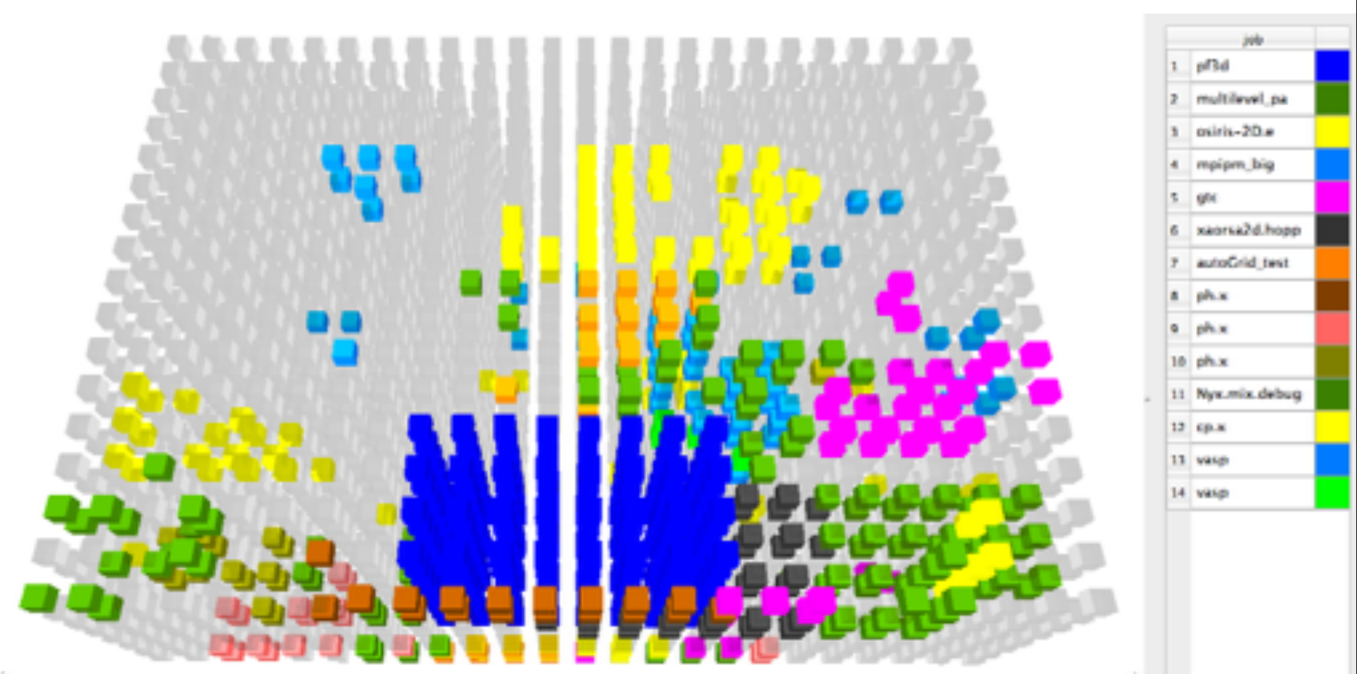
- Operating system noise (OS jitter)
  - OS daemons running on some cores of each node
- Placement/location of the allocated nodes for the job (Allocation shape)
- Contention for shared resources (Inter-job contention)
  - Sharing network links with other jobs

4x8x8-shaped pF3D job





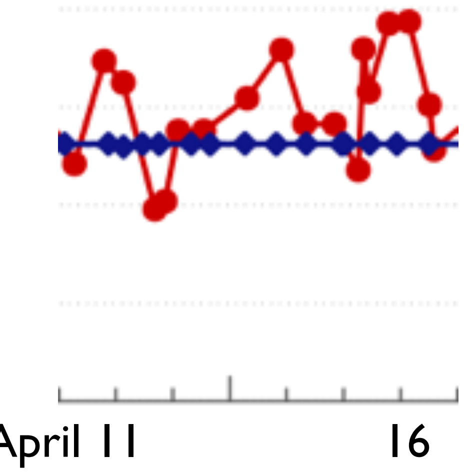
April 11

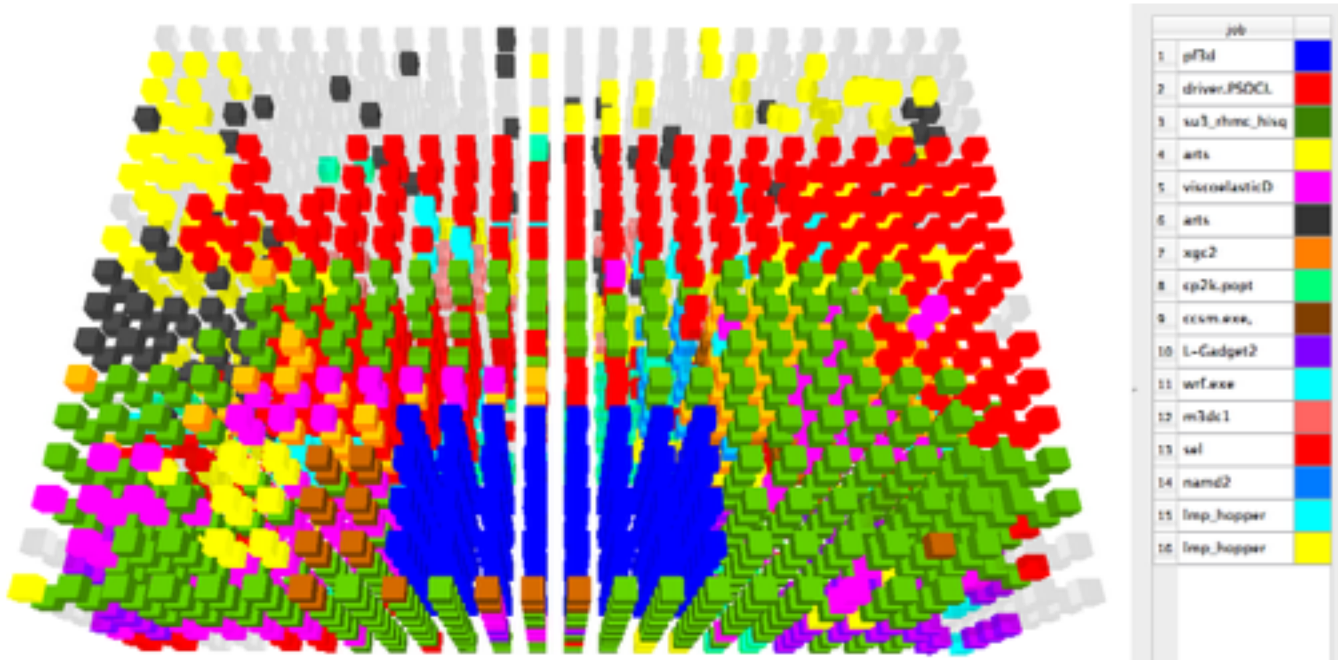


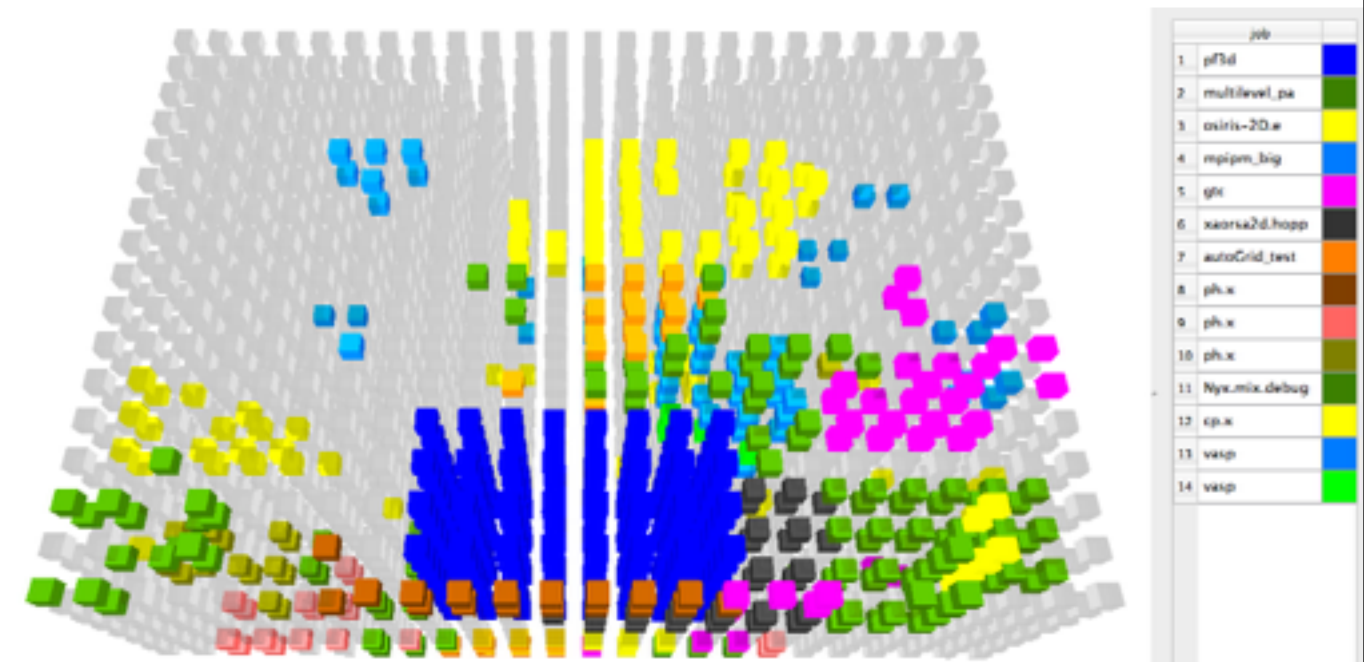
April 16

https://scalability.llnl.gov/performance-analysis-through-visualization/software.php

4x8x8-shaped pF3D job







April 11

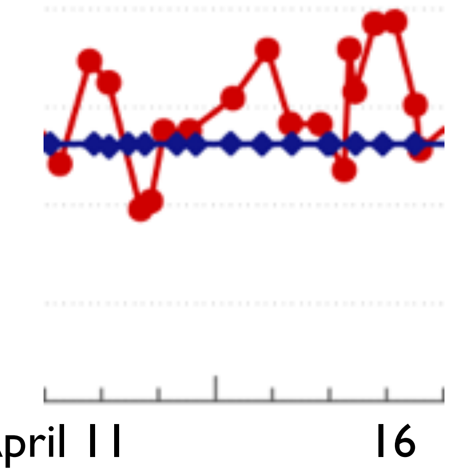
MILC job in green

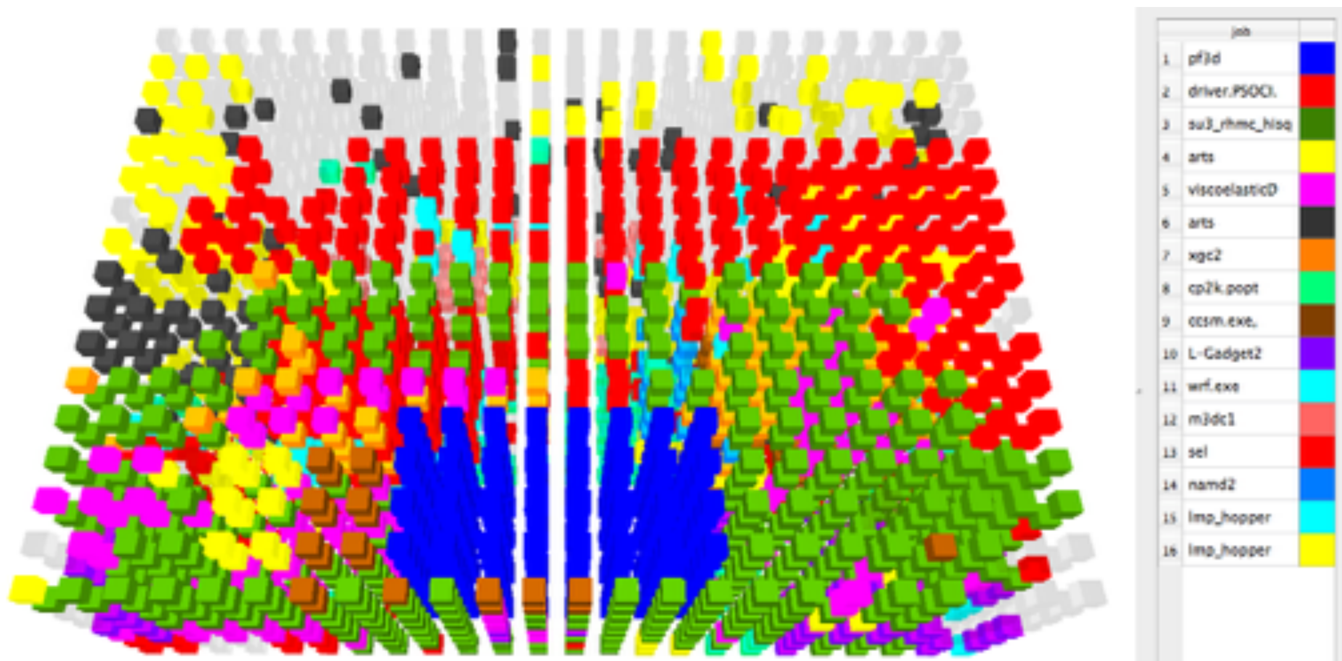
April 16

25% higher messaging rate

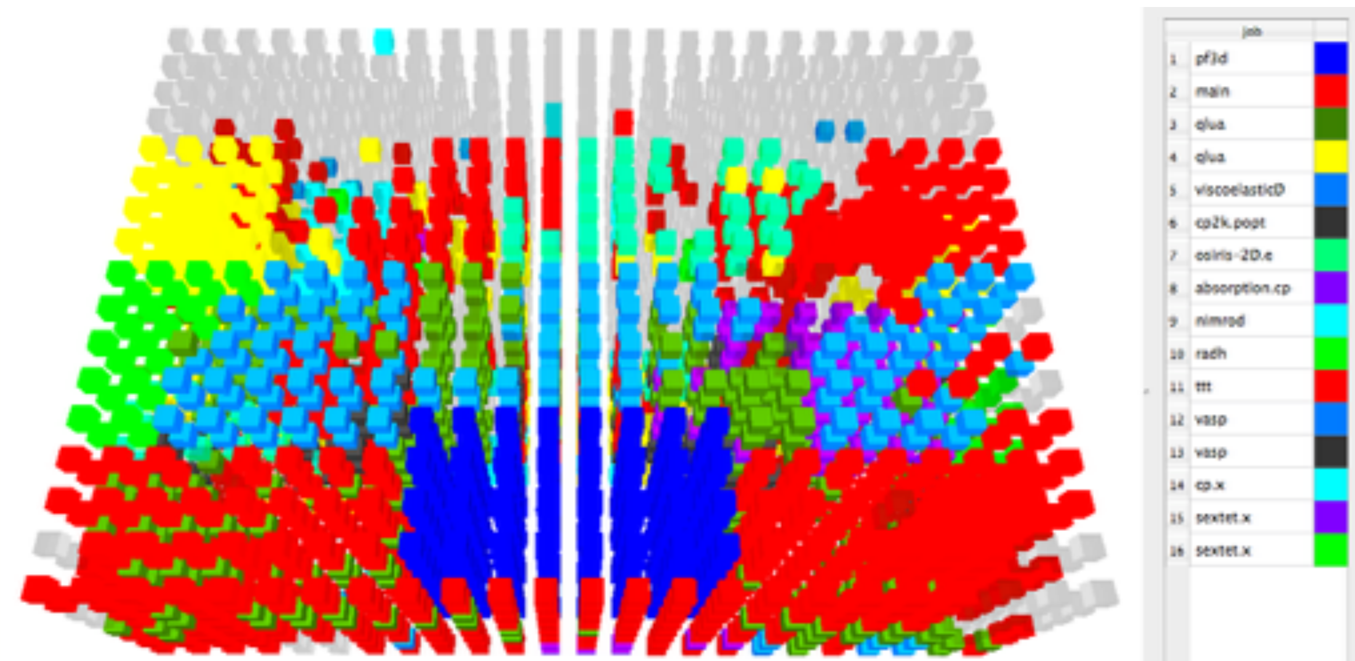
https://scalability.llnl.gov/performance-analysis-through-visualization/software.php

4x8x8-shaped pF3D job



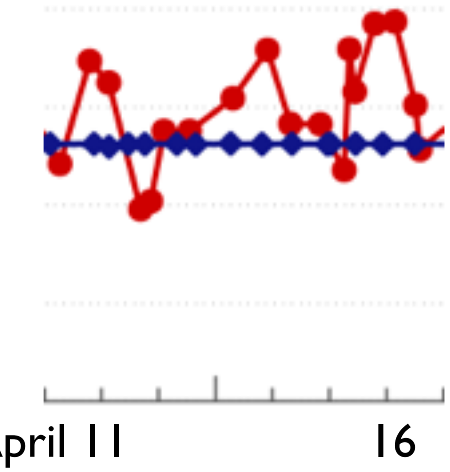


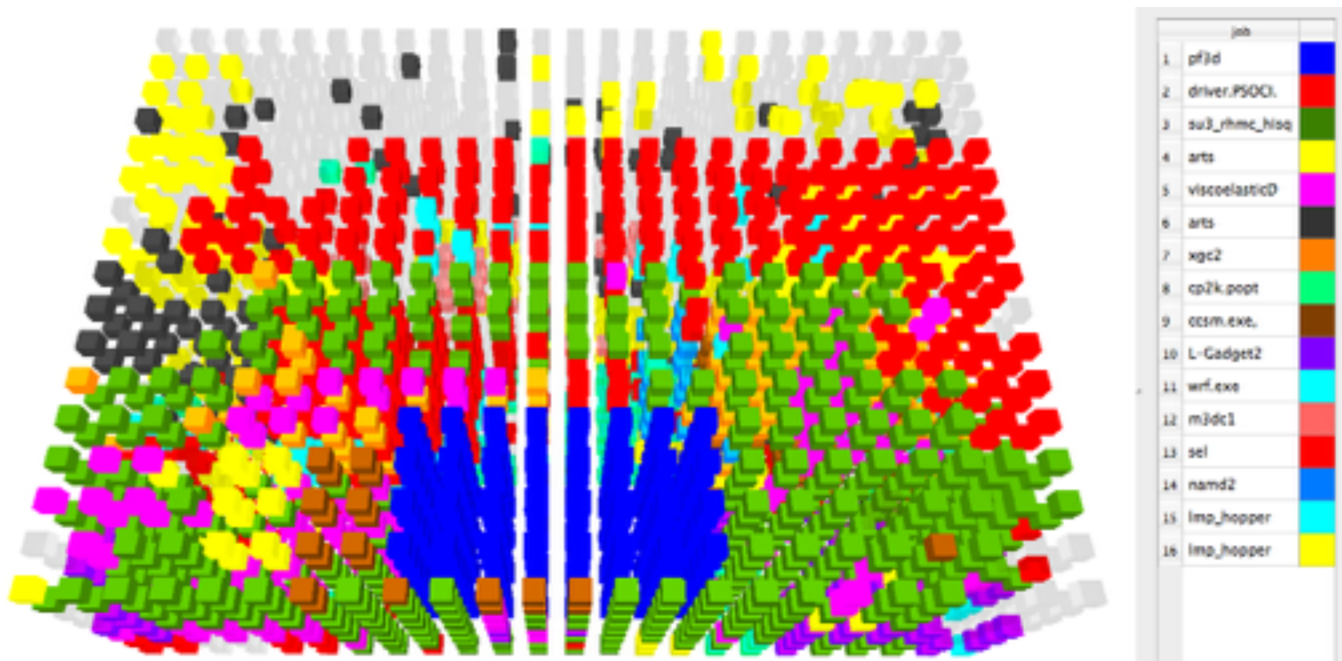
April 11



April 16b

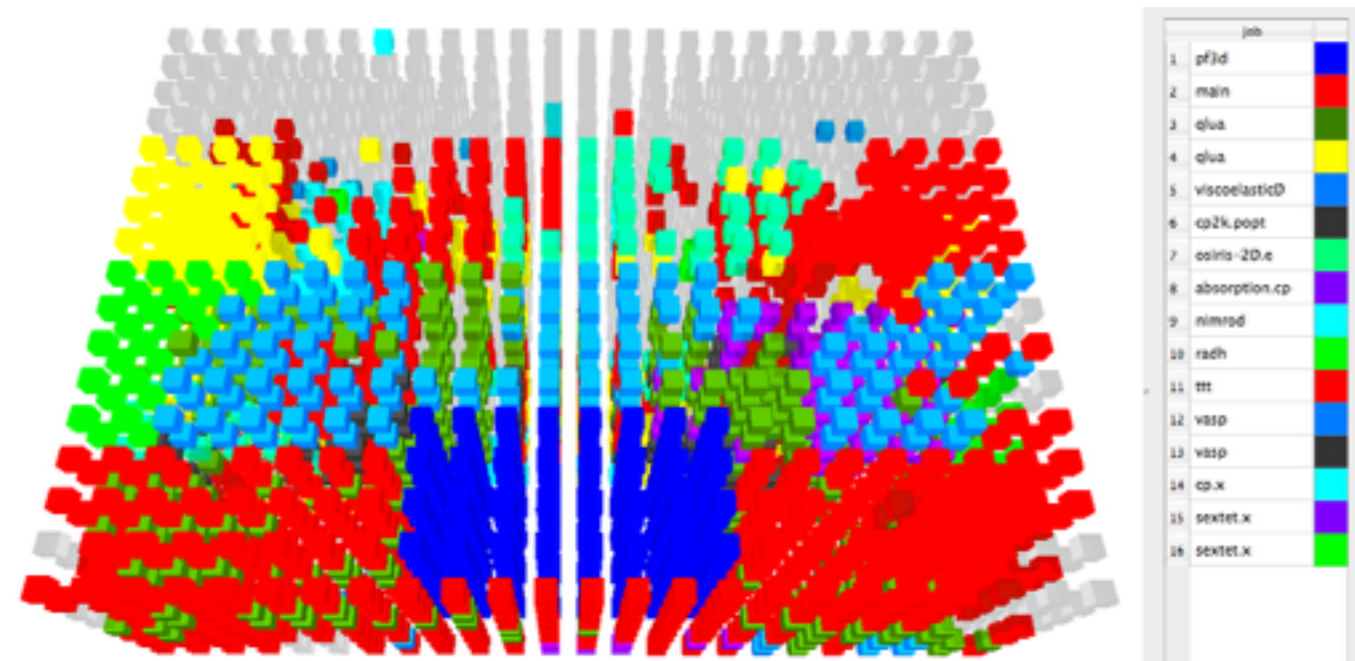
4x8x8-shaped pF3D job





April 11

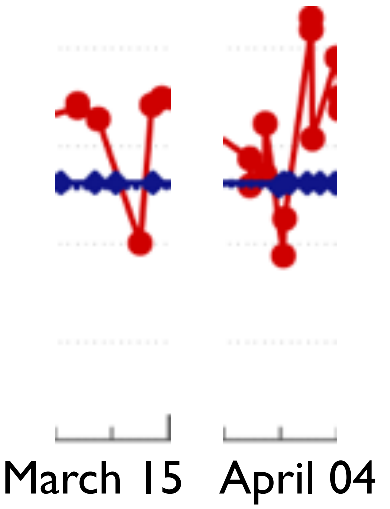
MILC job in green

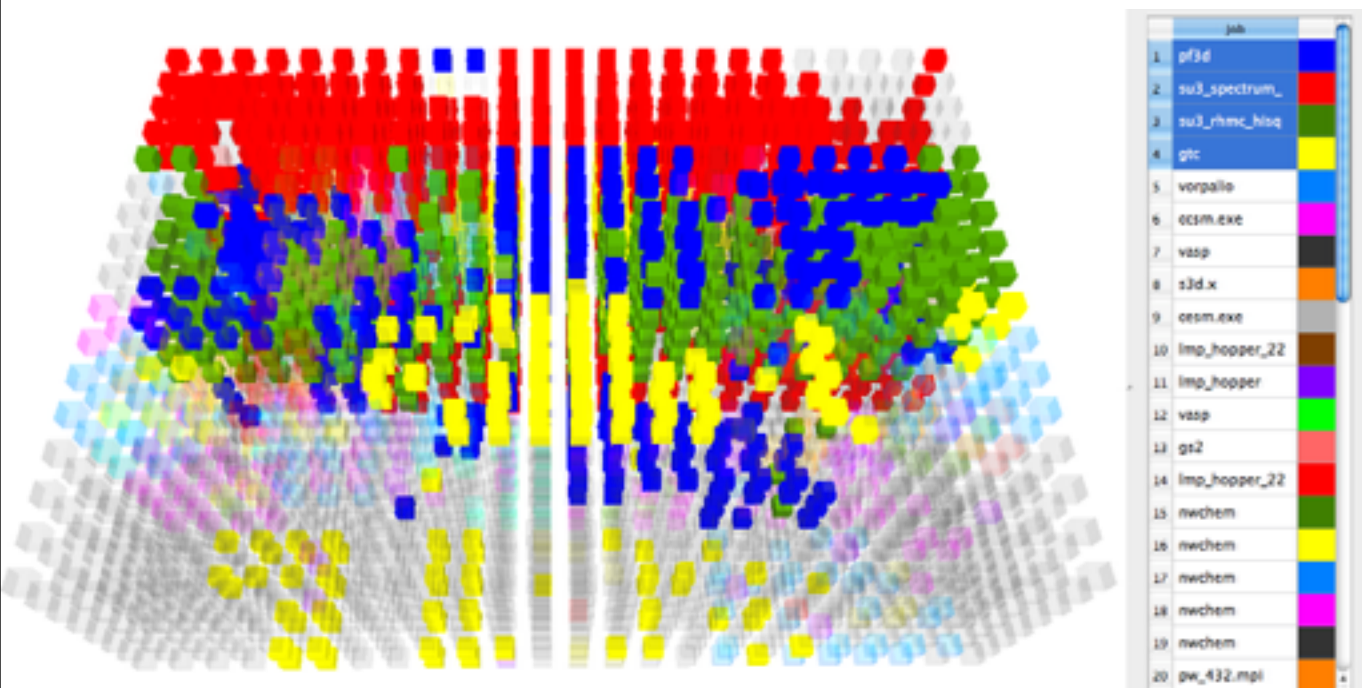


April 16b

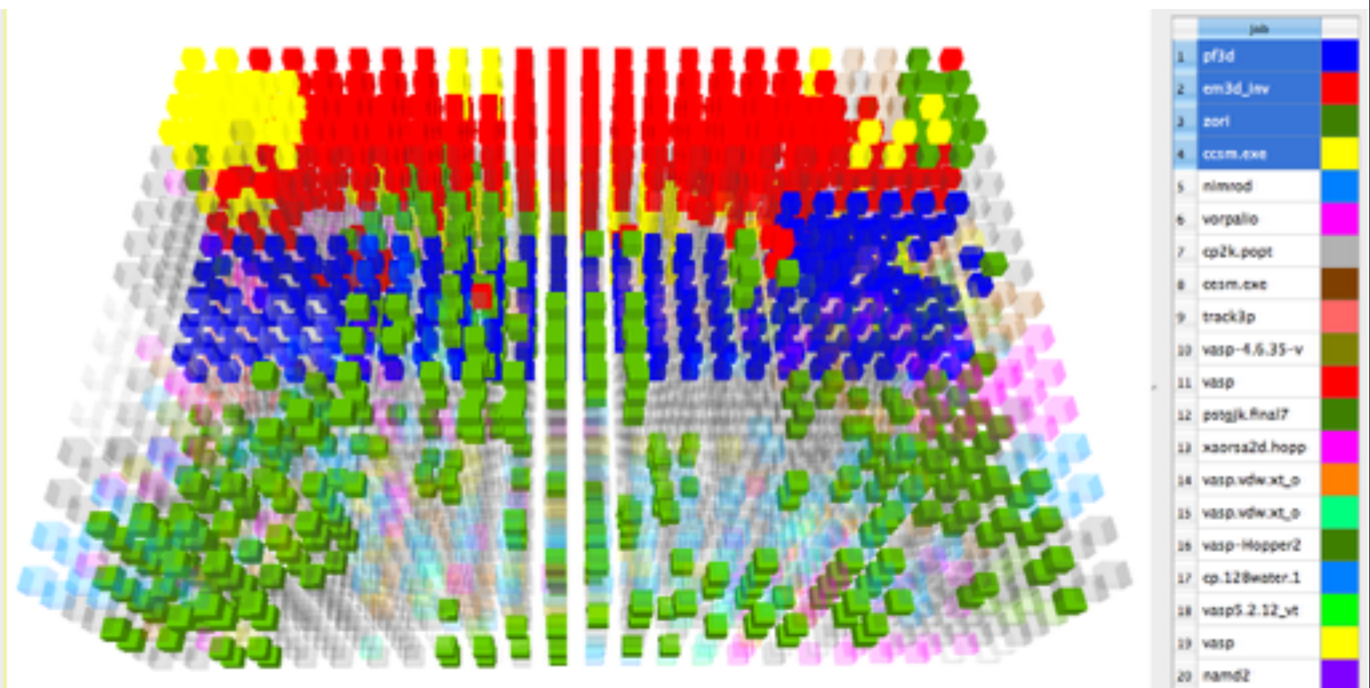
27.8% higher messaging rate,
LSMS is not communication-heavy

Slowest vs. fastest job



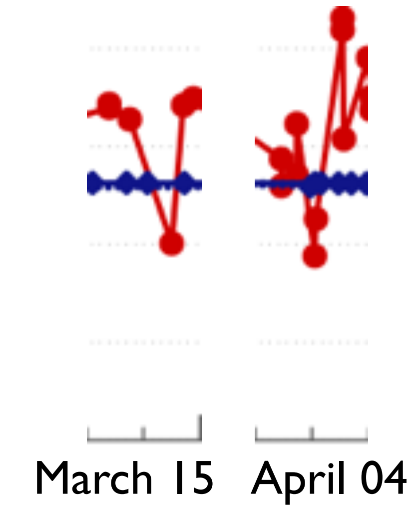


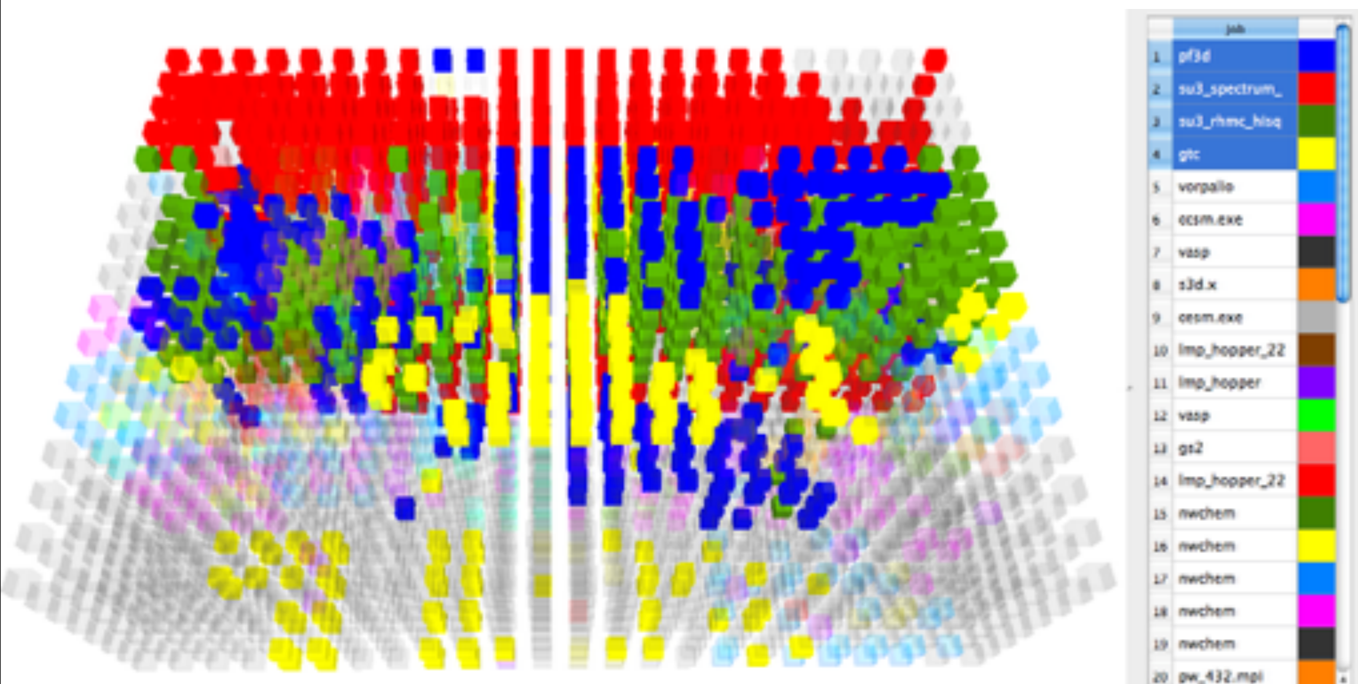
March 15



April 04

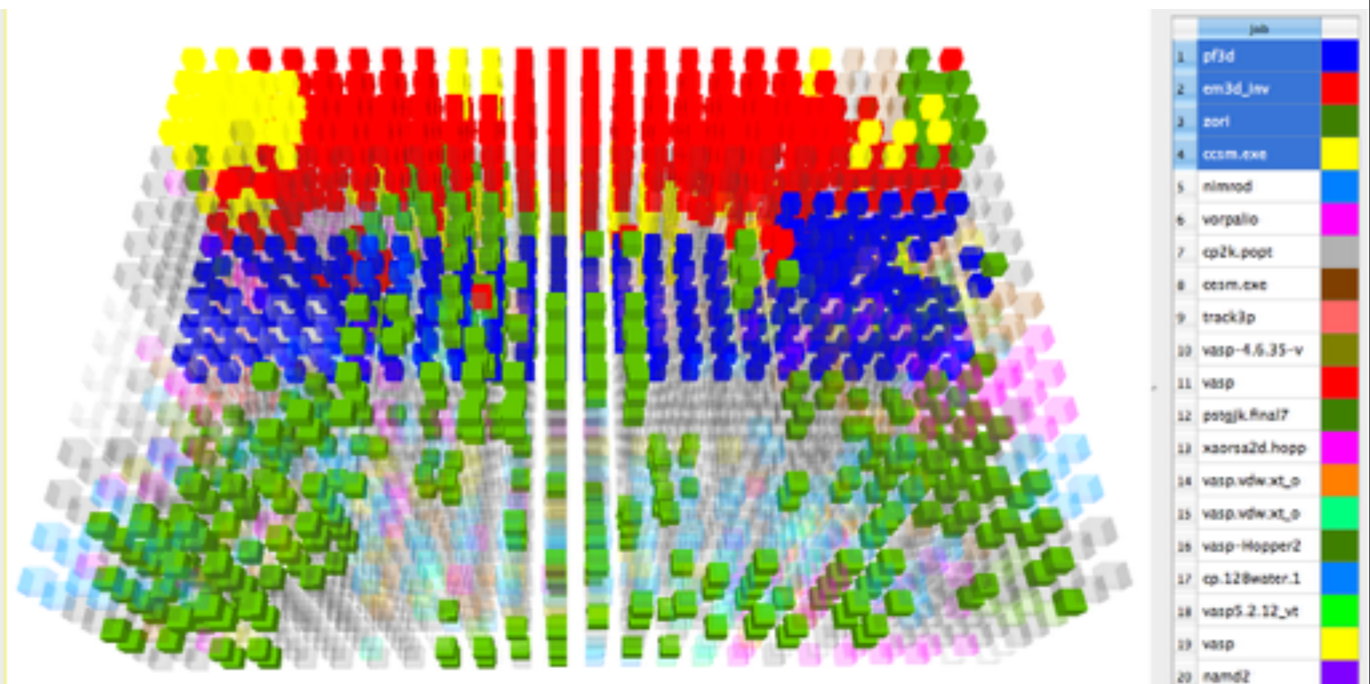
Slowest vs. fastest job





March 15

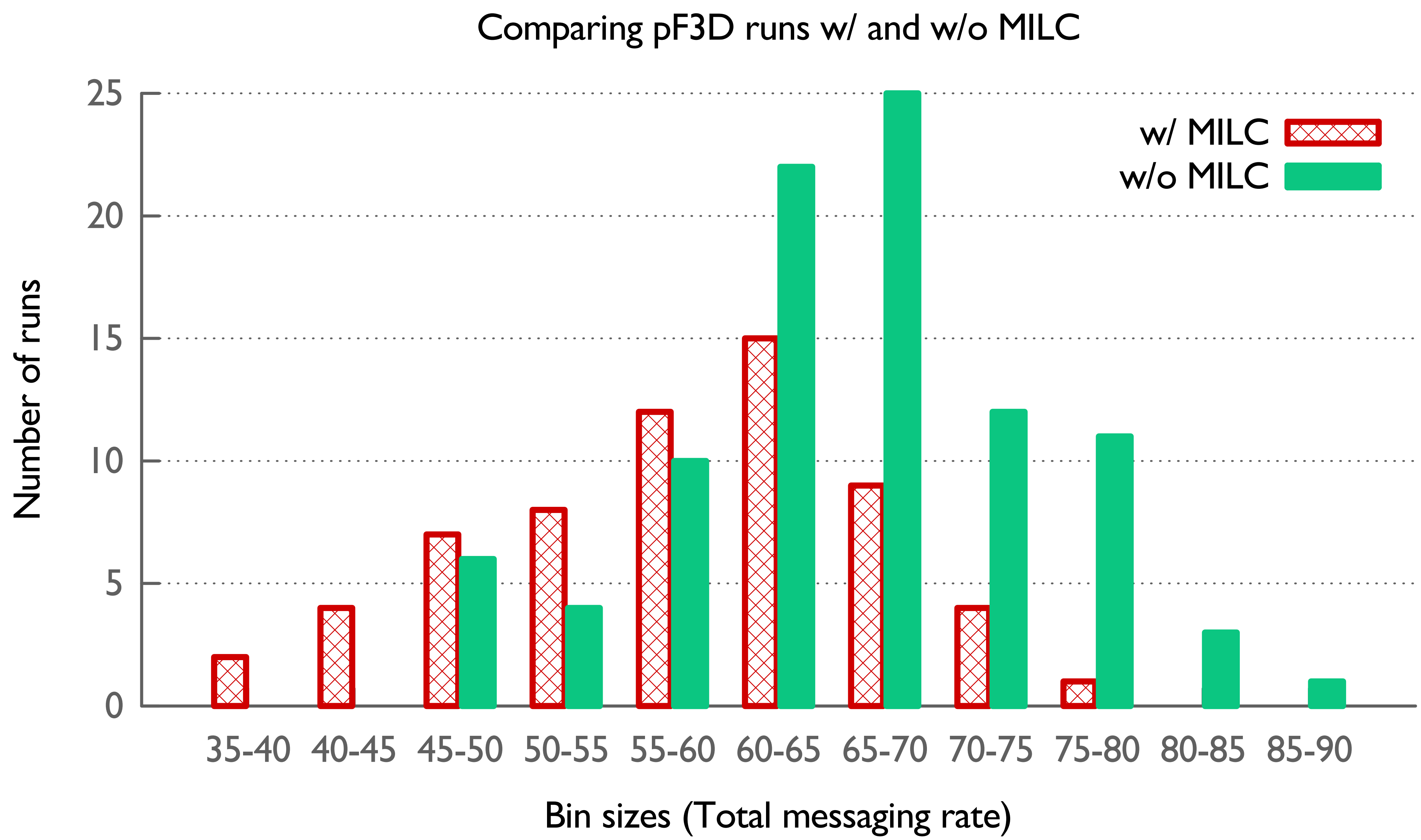
Three conflicting jobs, two MILC



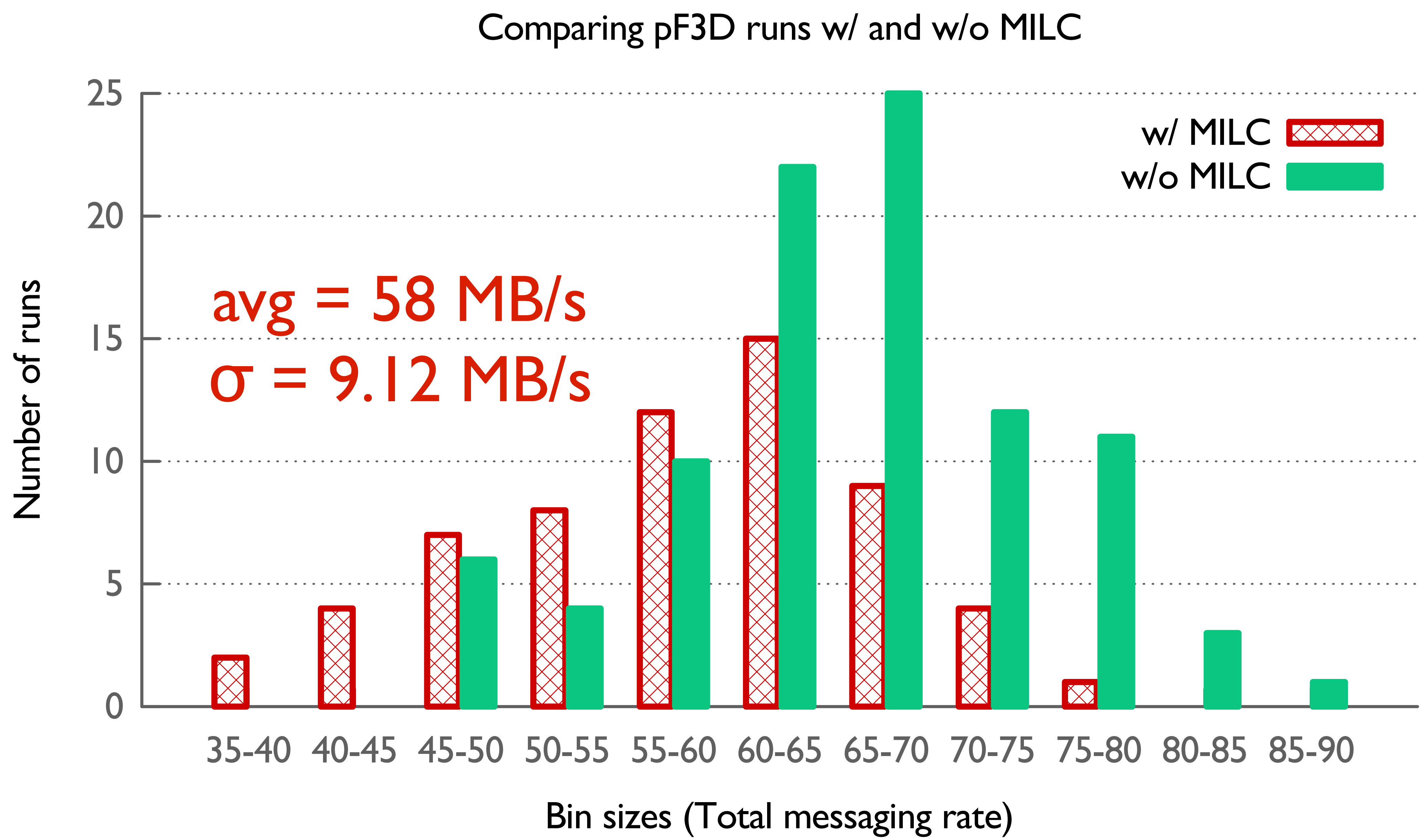
April 04

2.29X higher messaging rate

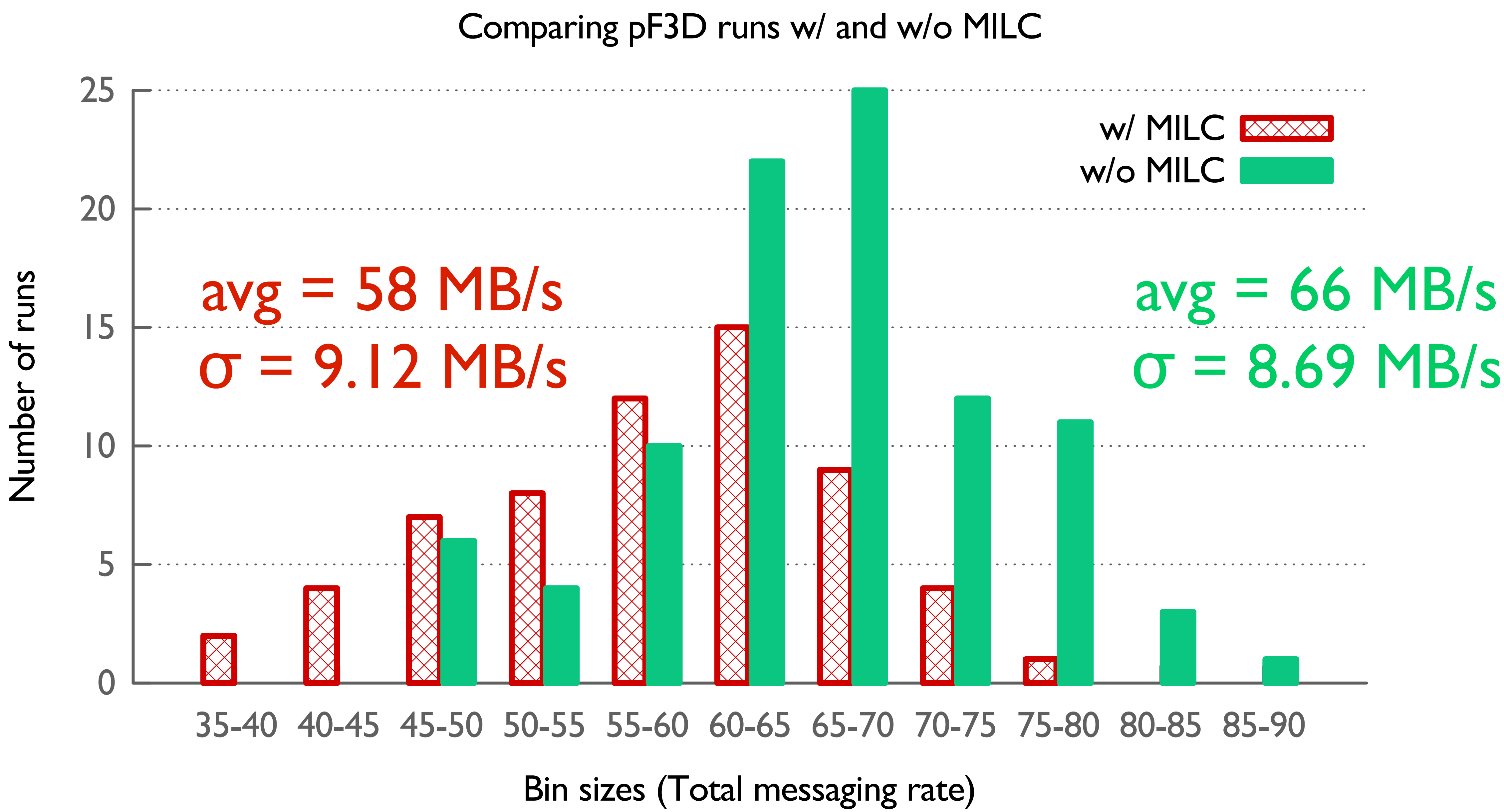
Effect of MILC on pF3D



Effect of MILC on pF3D

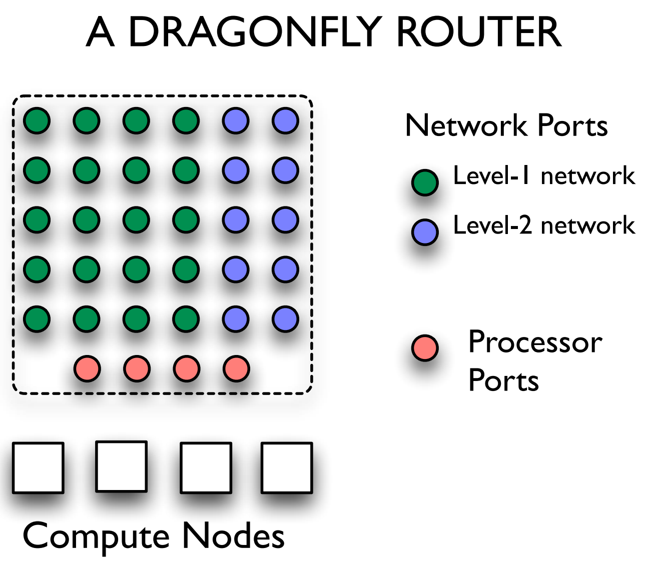


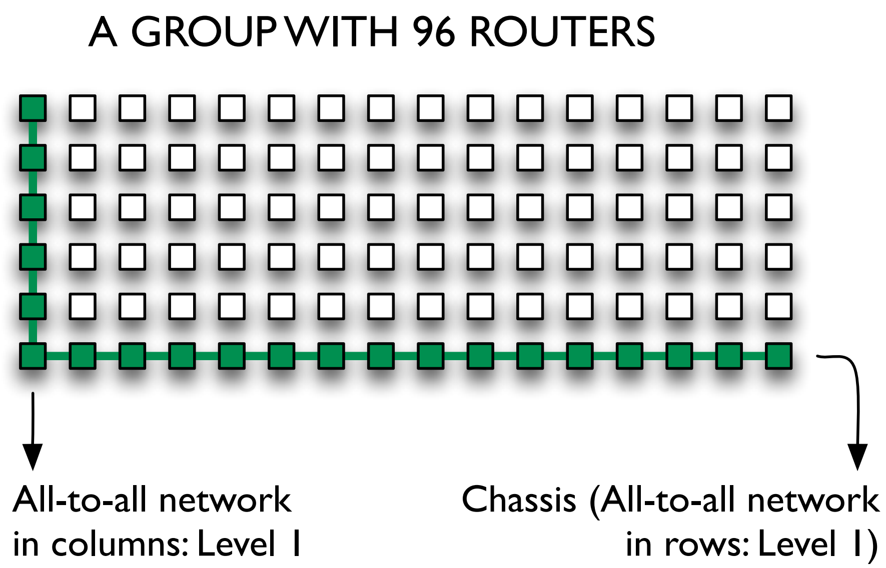
Effect of MILC on pF3D

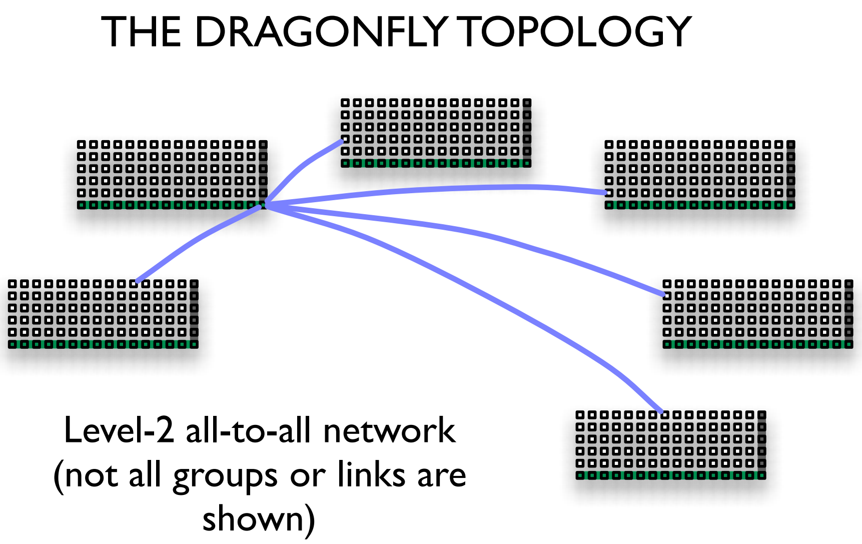


Modeling job placements and message routing

- Dragonfly topology: a two-level hierarchical topology
- Routing choices: static (deterministic) vs. dynamic (adaptive), direct vs. indirect (random jumps)
- Placement options: random, round-robin, blocked

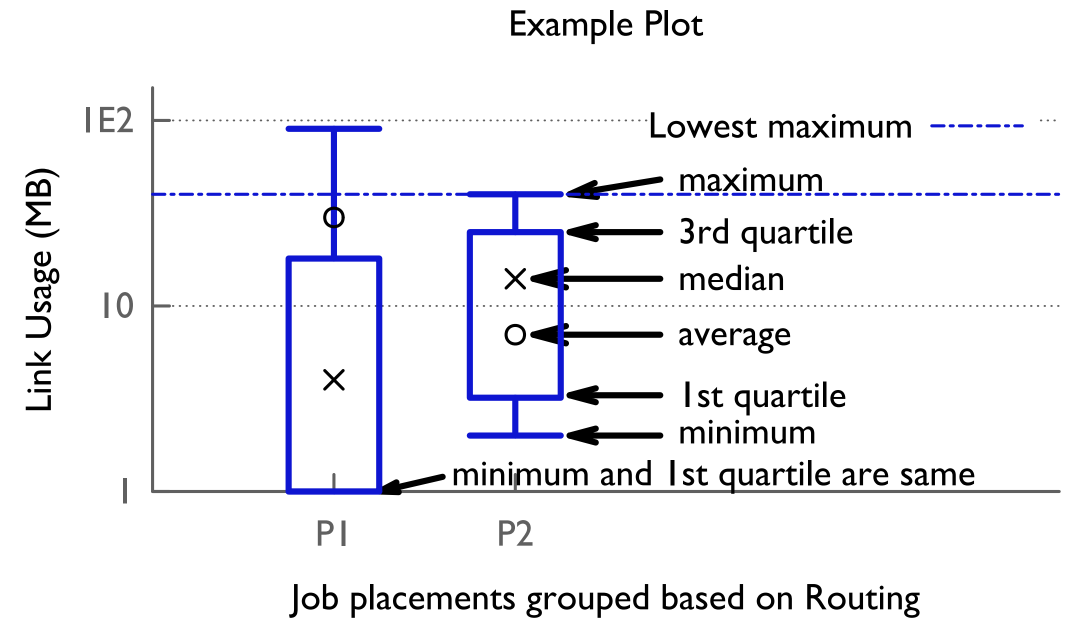


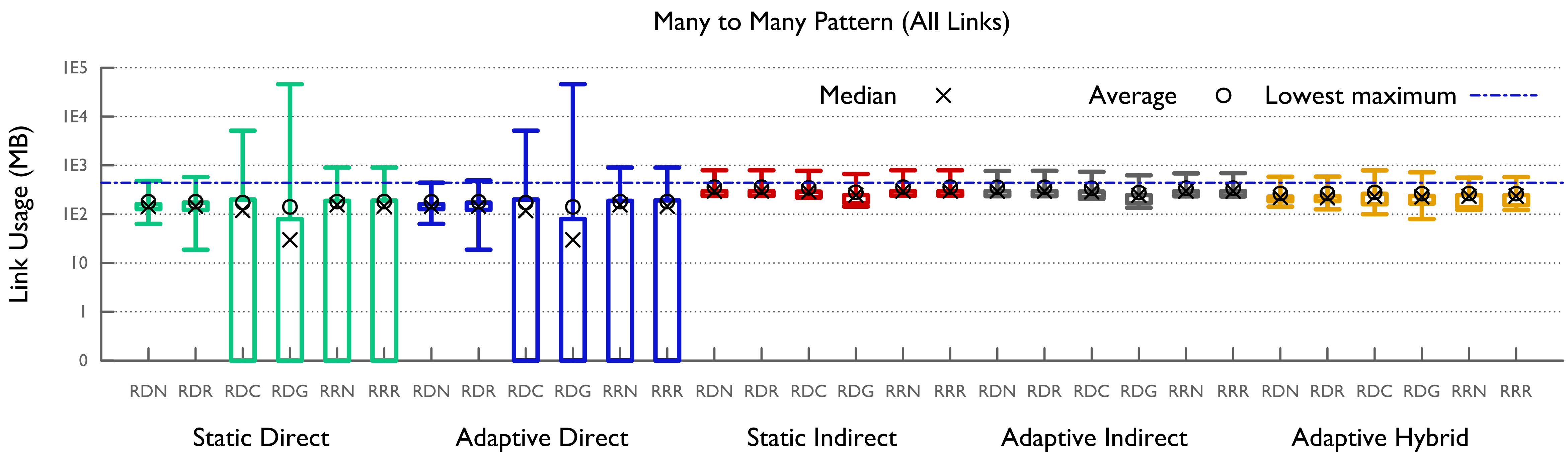




Single jobs

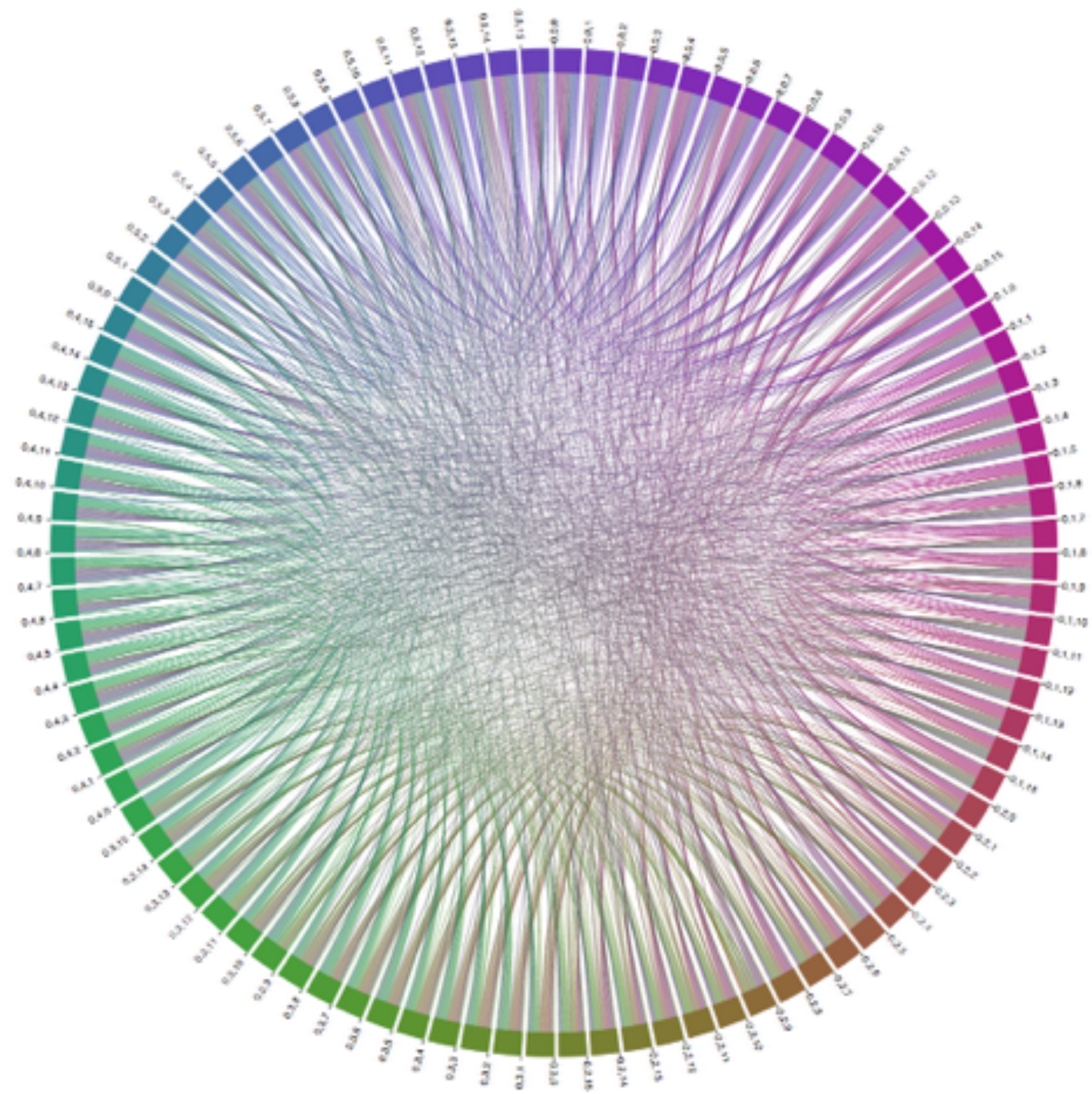
- All-to-all over sub-communicators
- Various traffic metrics



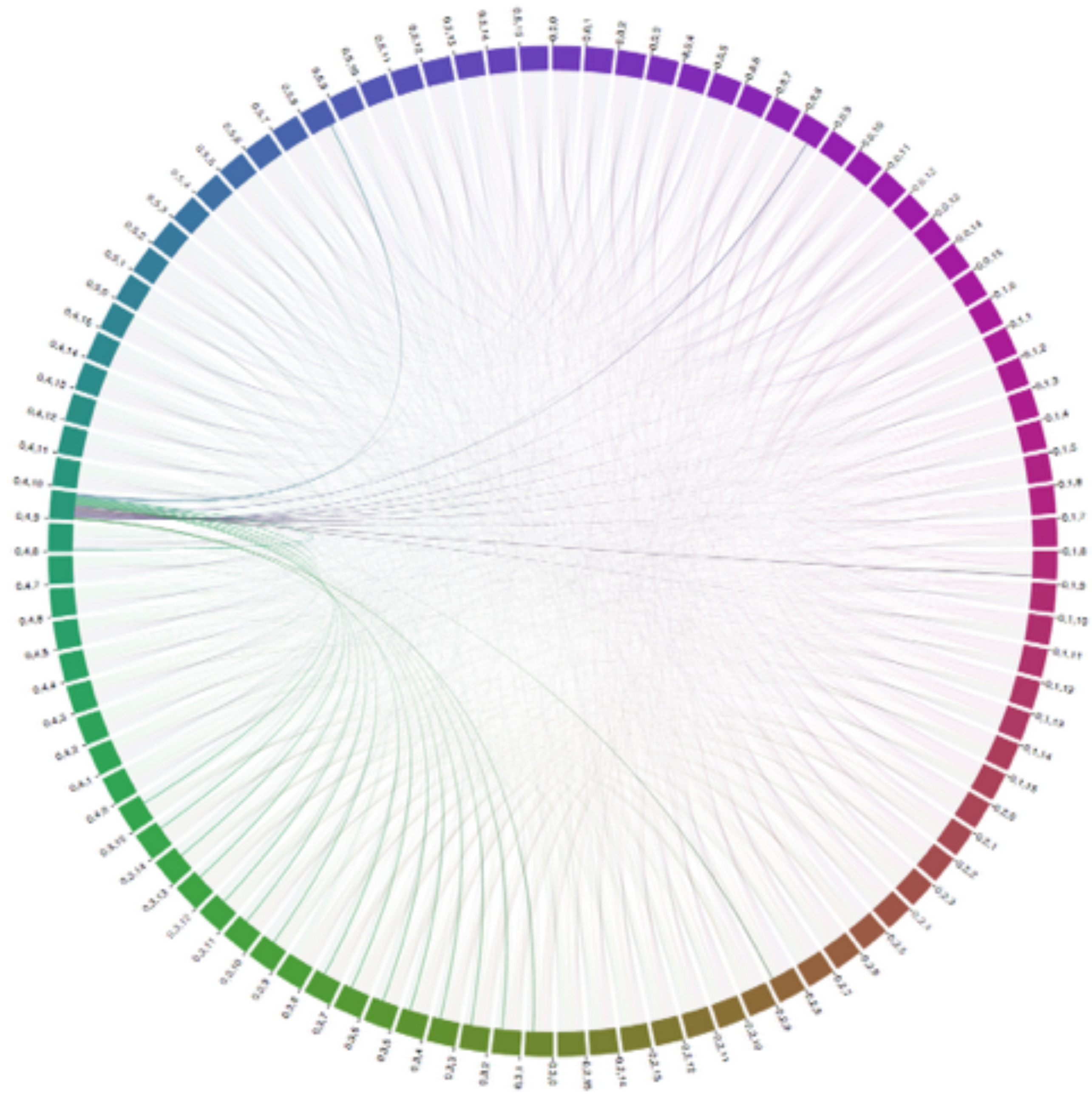


Edison @ NERSC

Edison @ NERSC



Edison @ NERSC



Summary

- Projecting information to non-traditional domains can help
- Rubik: Python-based tool for task mappings
- Boxfish:
  - Visualize network traffic over links
  - Visualize placement of jobs on the nodes

http://computation-rnd.llnl.gov/extreme-computing/interconnection-networks.php

This work was funded by the Laboratory Directed Research and Development Program at LLNL under project tracking code 13-ERD-055: STATE - **S**calable **T**opology **A**ware **T**ask **E**mbedding.

Petascale Tools Workshop ◆ August 04, 2014

LLNL: Abhinav Bhatele, Peer-Timo Bremer, Todd Gamblin, Katherine E. Isaacs, Steven H. Langer, Kathryn Mohror, Martin Schulz

Illinois: Ronak Buch, Nikhil Jain, Harshitha Menon, Laxmikant V. Kale, Michael Robson

Utah: Amey Desai, Aaditya G. Landge, Valerio Pascucci

Purdue: Ahmed Abdel-Gawad, Mithuna Thottethodi

LBL: Brian Austin, Nicholas J. Wright

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 Laboratory, P. O. Box 808, Livermore, CA 94551