IS TOPOLOGY IMPORTANT AGAIN?

Effects of contention on message latencies in large supercomputers

Abhinav S Bhatele and Laxmikant V Kale

Parallel Programming Laboratory, UIUC

PPL
UIUC

Outline

Why should we consider topology aware mapping for optimizing performance?

Demonstrate the effects of contention on message latencies through simple MPI benchmarks

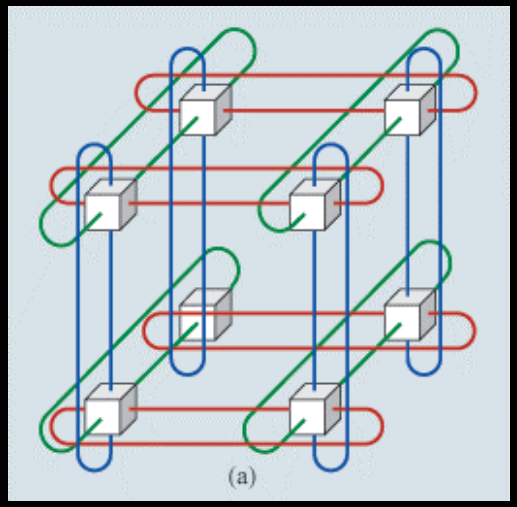
Obtaining topology information: TopoManager API
Case Study: OpenAtom

The Mapping Problem

- Given a set of communicating parallel "entities", map them onto physical processors
- Entities
  - COMM_WORLD ranks in case of an MPI program
  - Objects in case of a Charm++ program
- Aim
  - Balance load
  - Minimize communication traffic

Target Machines

- 3D torus/mesh interconnects
- Blue Gene/P at ANL:
  - 40,960 nodes, torus - 32 x 32 x 40
- XT4 (Jaguar) at ORNL:
  - 8,064 nodes, torus - 21 x 16 x 24
- Other interconnects
  - Fat-tree
  - Kautz graph: SiCortex



Motivation

- Consider a 3D mesh/torus interconnect
- Message latencies can be modeled by

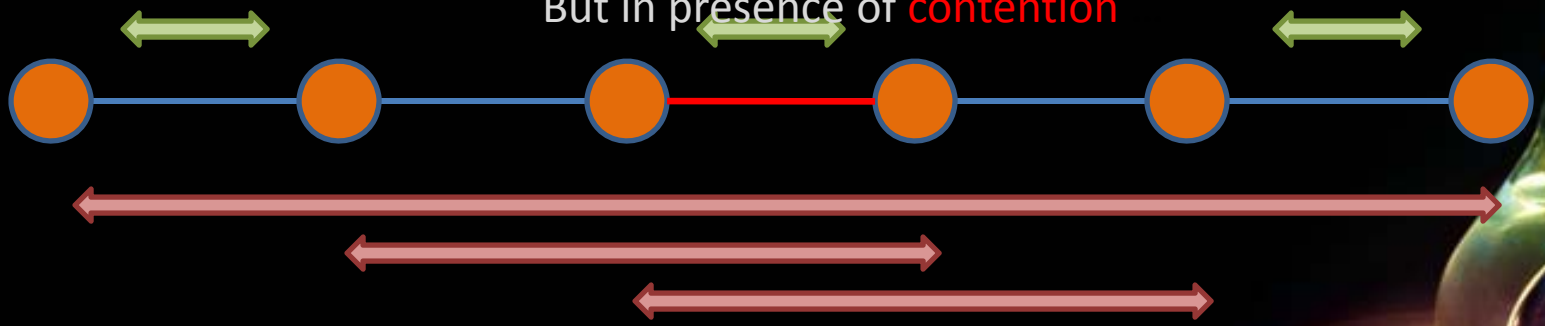
$$(L_f/B) \times D + L/B$$

L_f = length of flit, B = bandwidth,

D = hops, L = message size

When $(L_f * D) << L$, first term is negligible

But in presence of contention



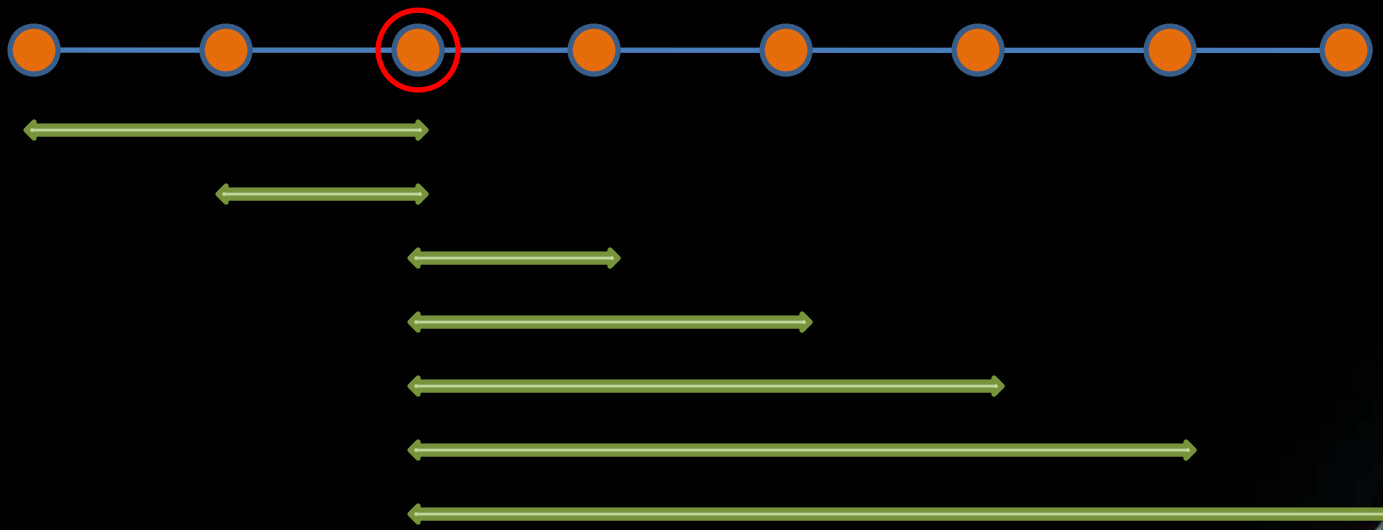
MPI Benchmarks†

- Quantification of message latencies and dependence on hops
  - No sharing of links (no contention)
  - Sharing of links (with contention)

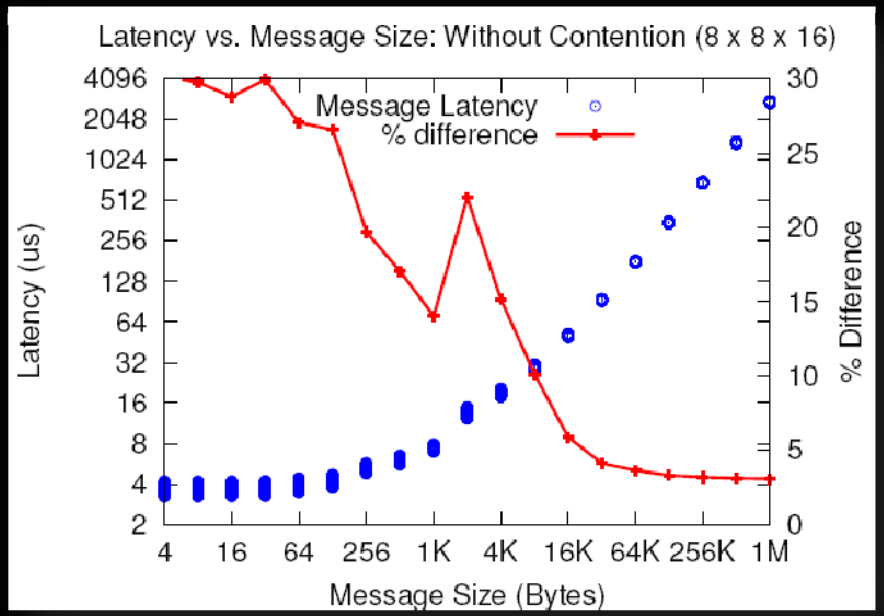
† http://charm.cs.uiuc.edu/~bhatele/phd/contention.htm

WOCON: No contention

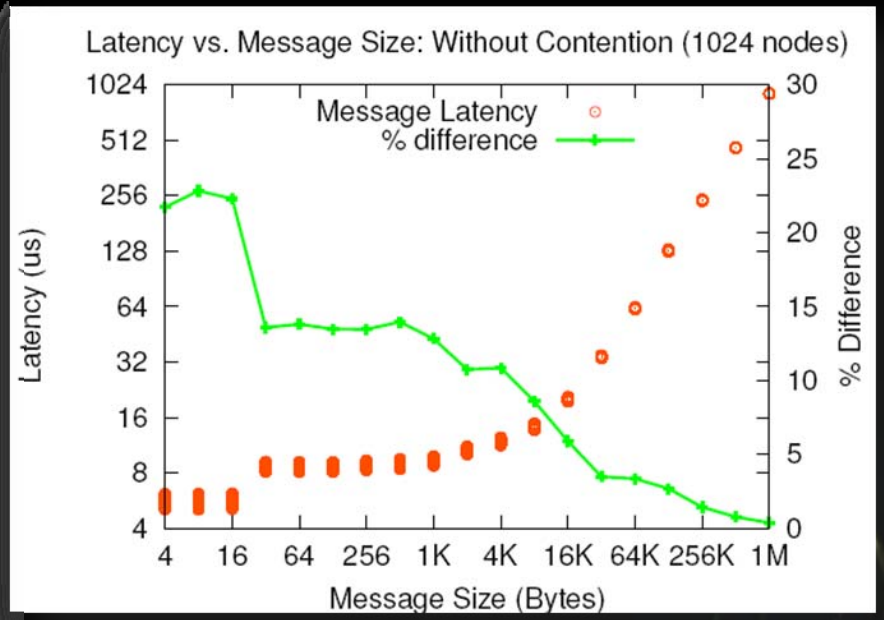
- A master rank sends messages to all other ranks, one at a time (with replies)



WOCON: Results



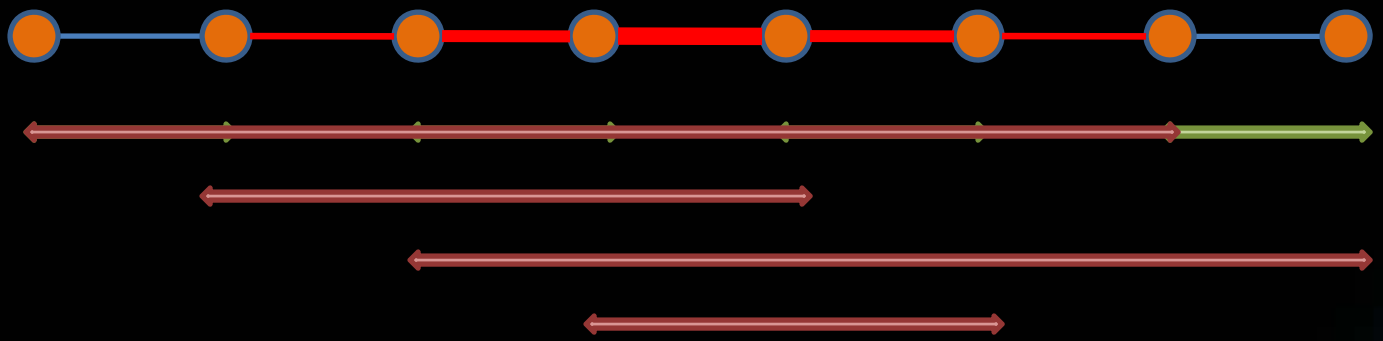
ANL Blue Gene/P



PSC XT3

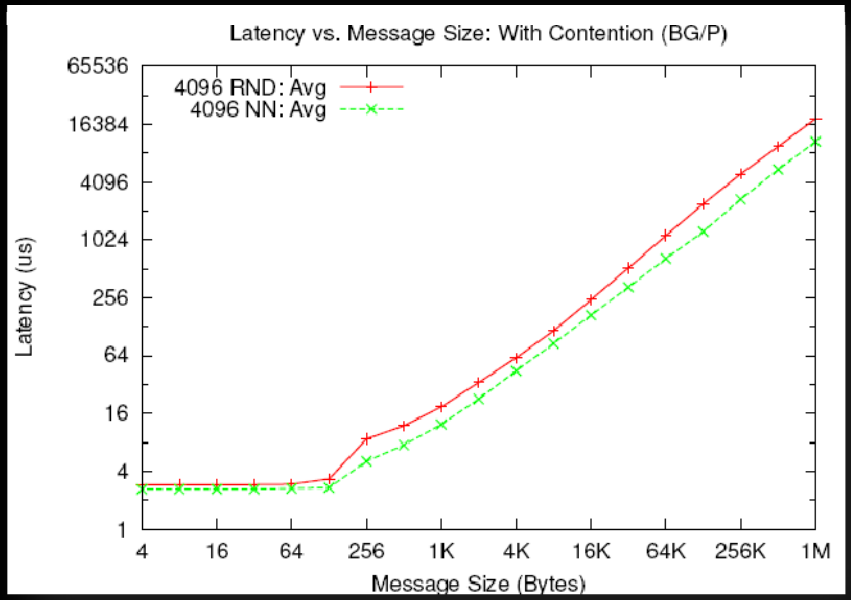
WICON: With Contention

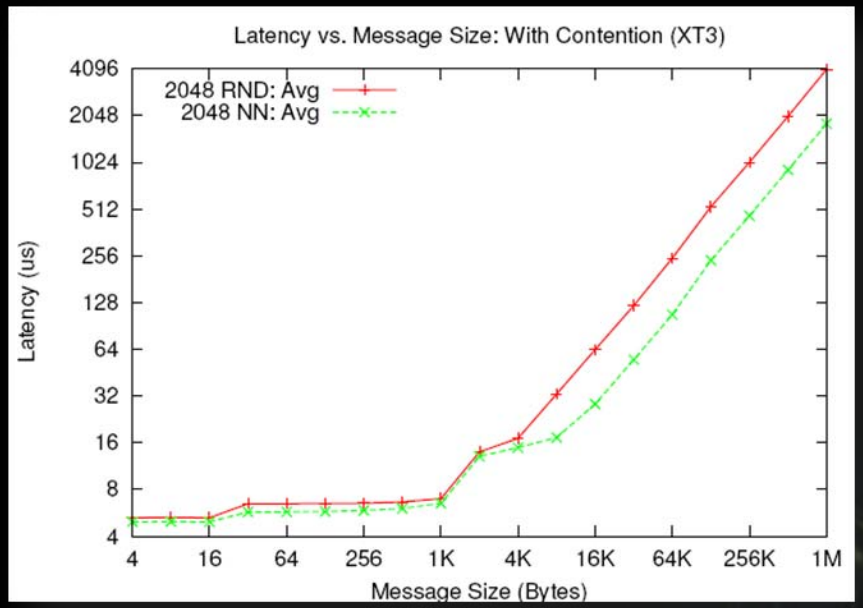
- Divide all ranks into pairs and everyone sends to their respective partner simultaneously



Radix: NN

WICON: Results



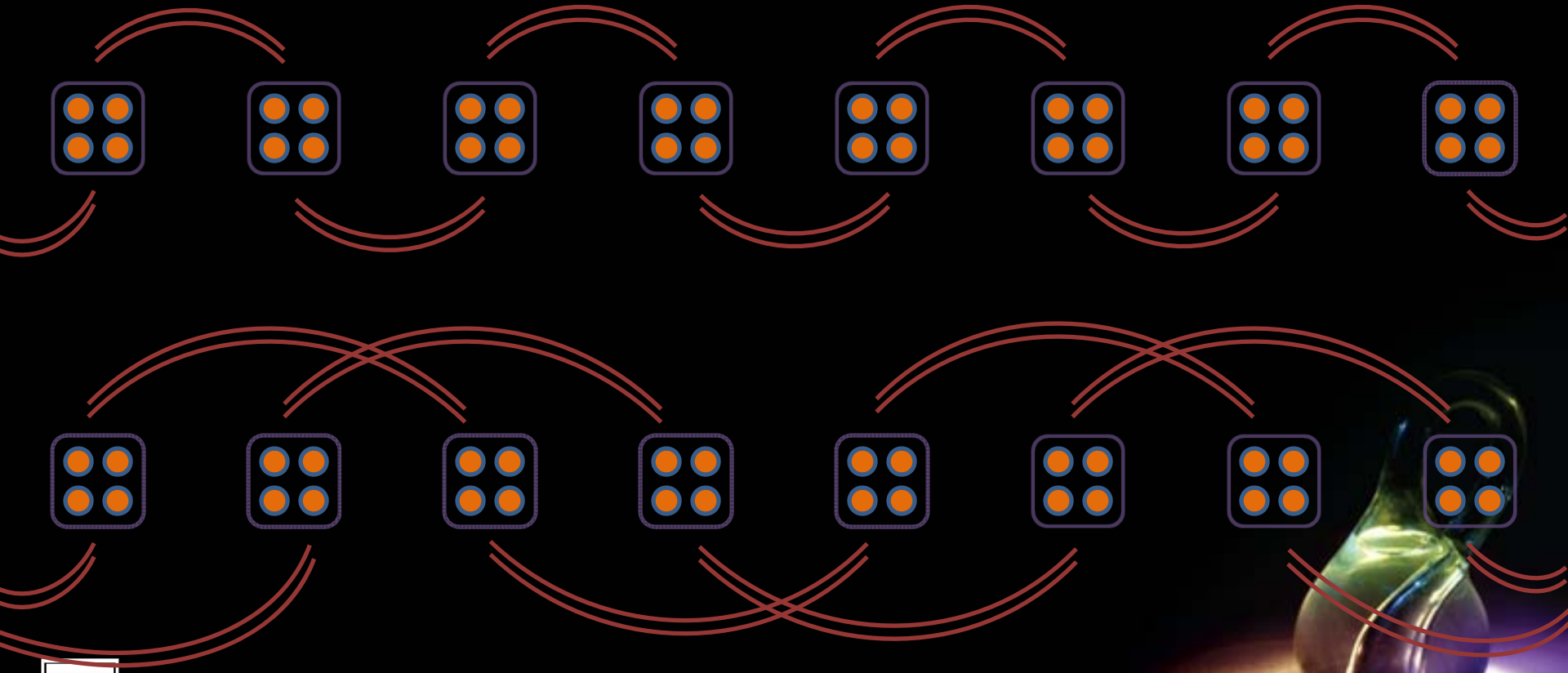


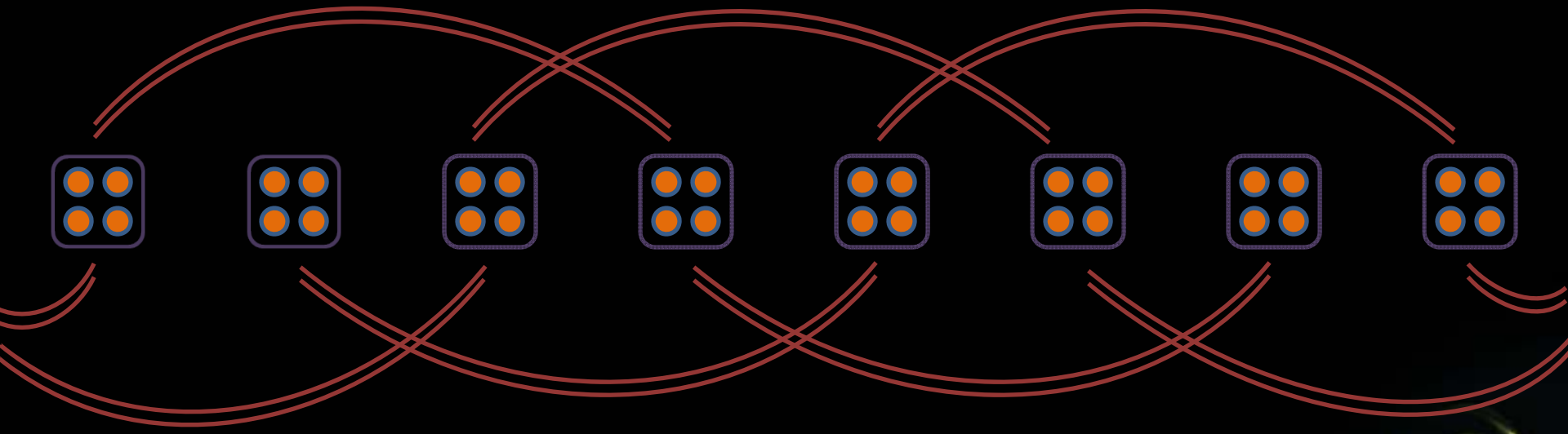
ANL Blue Gene/P

PSC XT3

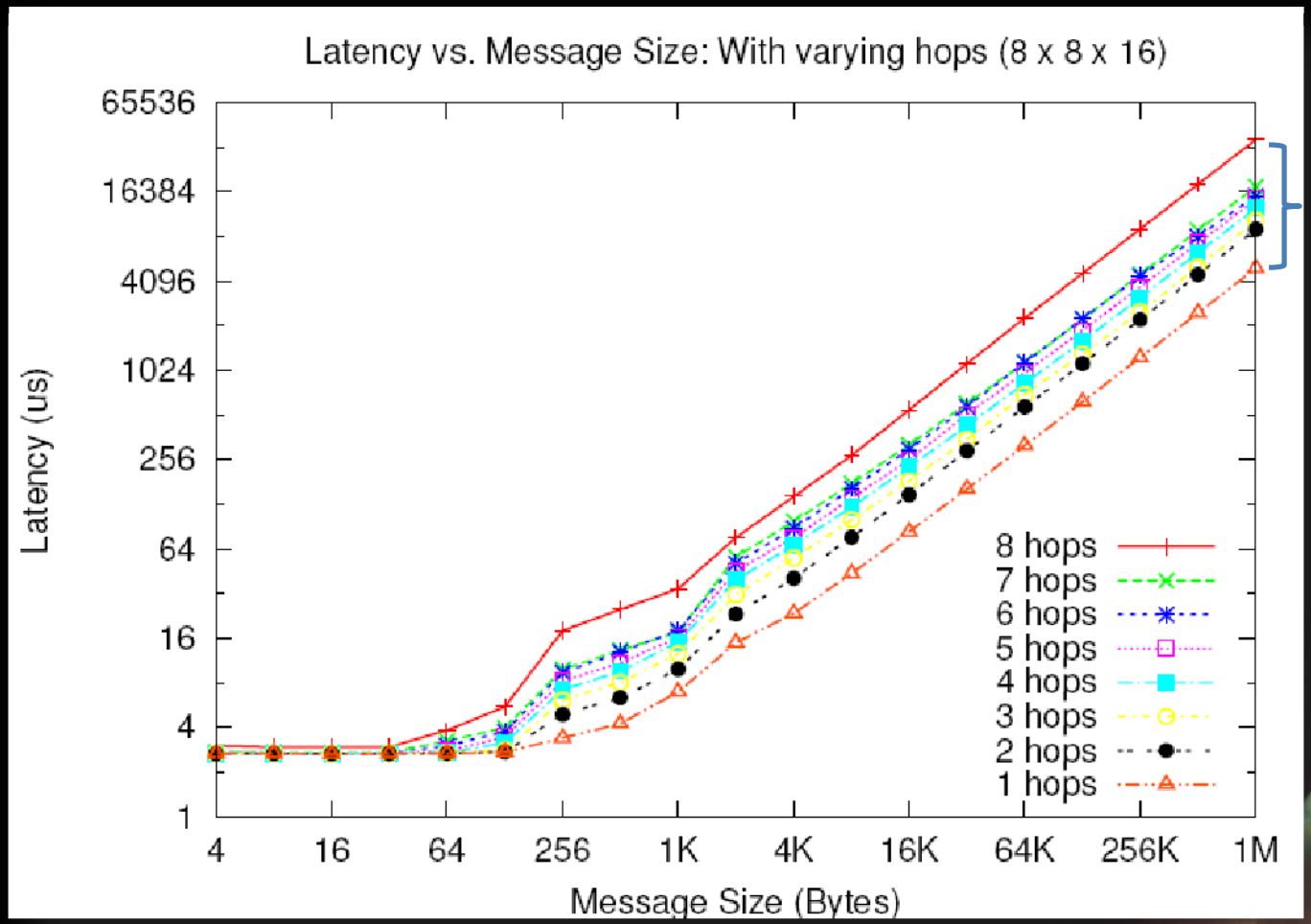
Message Latencies and Hops

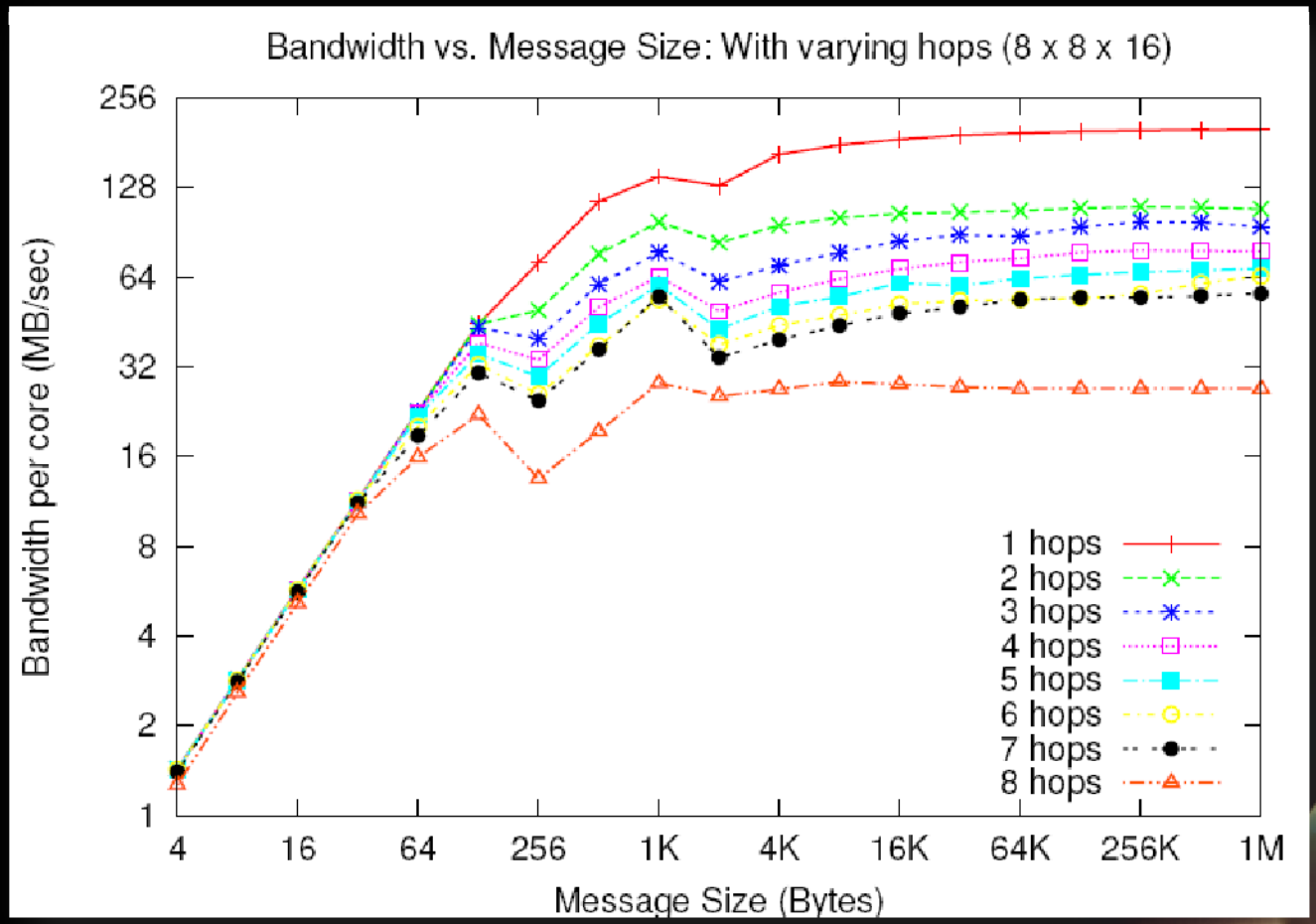
- Pair each rank with a partner which is ‘n’ hops away





Results


Latency vs. Message Size: With varying hops (8 x 8 x 16)
Latency (us)
65536
16384
4096
1024
256
64
16
4
1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)
8 hops
7 hops
6 hops
5 hops
4 hops
3 hops
2 hops
1 hops
8 times


Bandwidth vs. Message Size: With varying hops (8 x 8 x 16)
Bandwidth per core (MB/sec)
256
128
64
32
16
8
4
2
1
4
16
64
256
1K
4K
16K
64K
256K
1M
Message Size (Bytes)
1 hops
2 hops
3 hops
4 hops
5 hops
6 hops
7 hops
8 hops

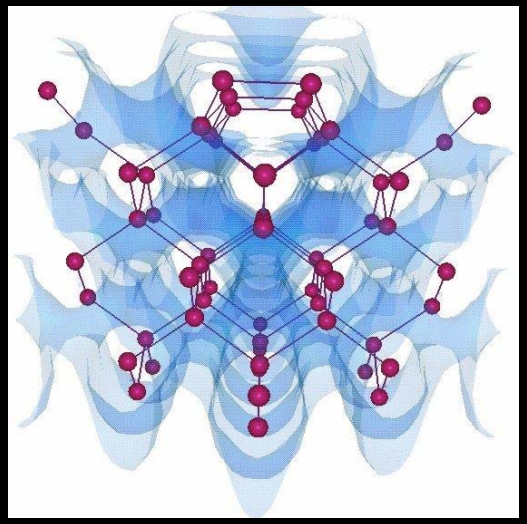
Topology Manager API†

- The application needs information such as
  - Dimensions of the partition
  - Rank to physical co-ordinates and vice-versa
- TopoManager: a uniform API
  - On BG/L and BG/P: provides a wrapper for system calls
  - On XT3 and XT4, there are no such system calls
    - Help from PSC and ORNL staff to discovery topology at runtime
  - Provides a clean and uniform interface to the application

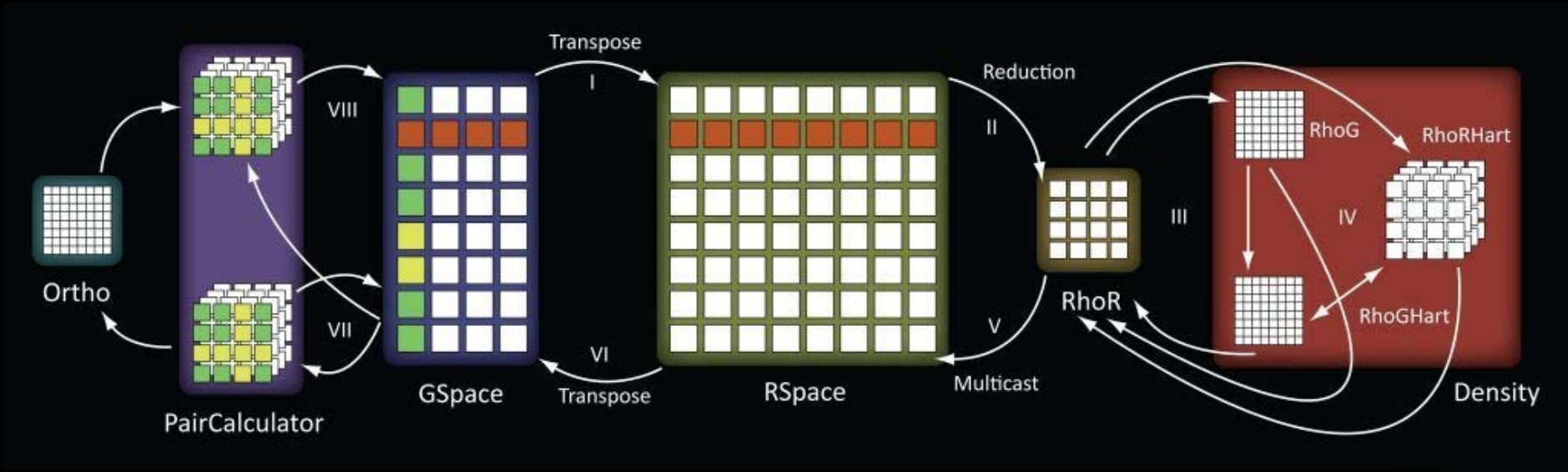
† http://charm.cs.uiuc.edu/~bhatele/phd/topomgr.htm

OpenAtom

- Ab-Initio Molecular Dynamics code
- Communication is static and structured
- Challenge: Multiple groups of objects with conflicting communication patterns

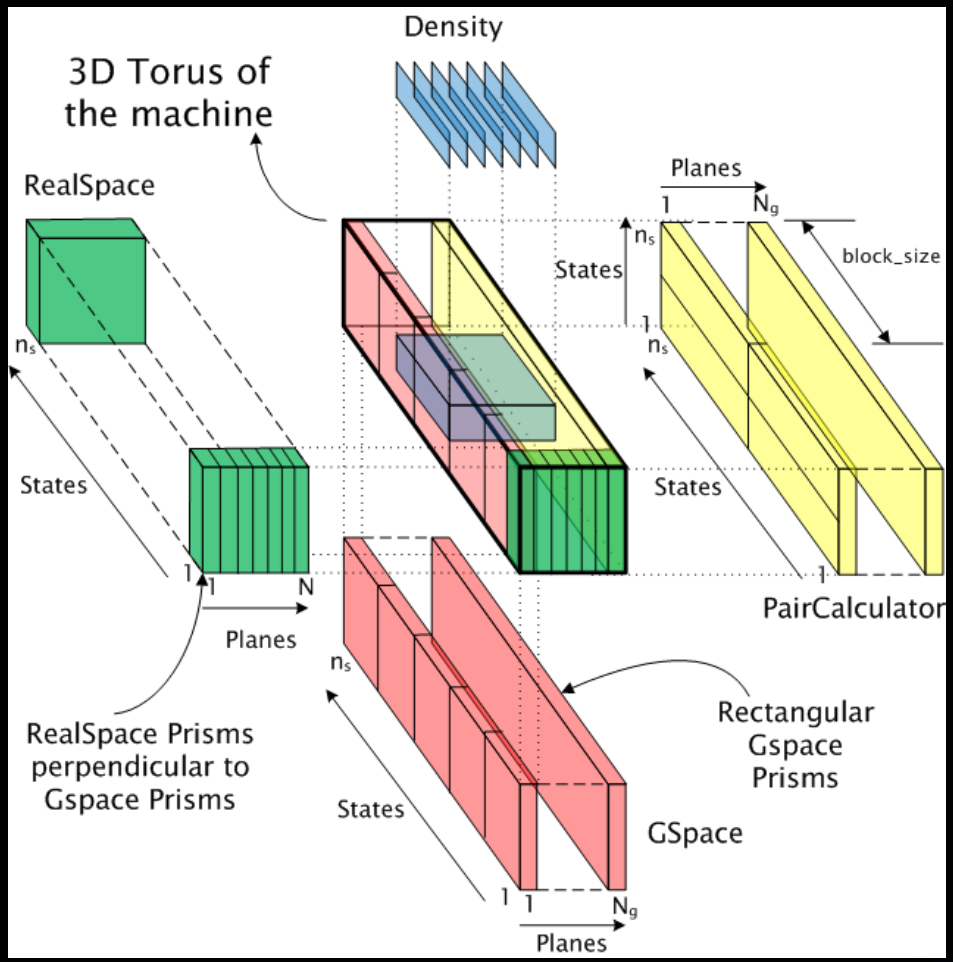


Parallelization using Charm++



[10] Eric Bohm, Glenn J. Martyna, Abhinav Bhatele, Sameer Kumar, Laxmikant V. Kale, John A. Gunnels, and Mark E. Tuckerman. **Fine Grained Parallelization of the Car-Parrinello ab initio MD Method on Blue Gene/L**. *IBM J. of R. and D.: Applications of Massively Parallel Systems, 52(1/2):159-174*, 2008.

Topology Mapping of Chare Arrays

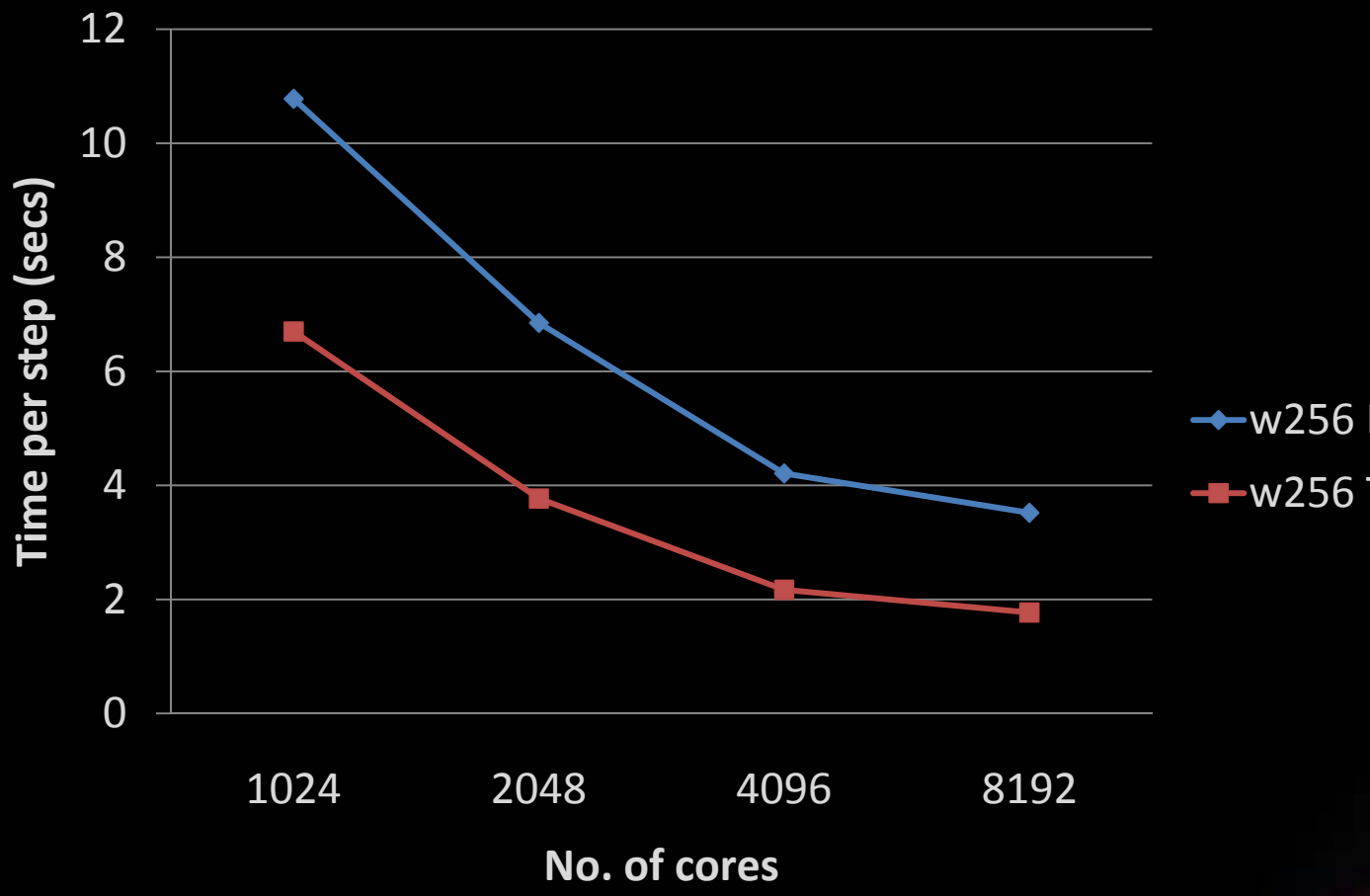


State-wise communication

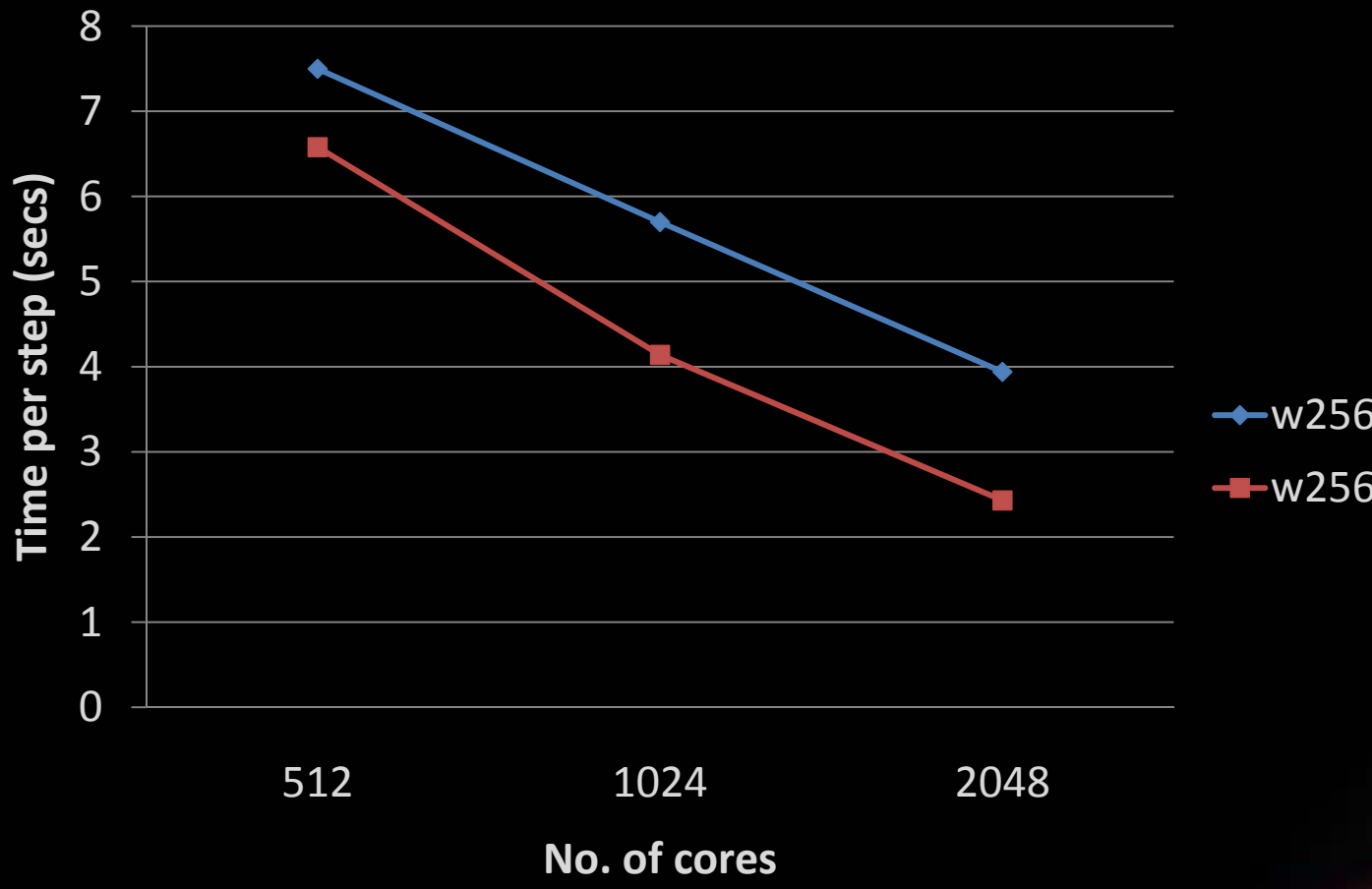
Plane-wise communication

Joint work with Eric J. Bohm

Results on Blue Gene/P (ANL)



Results on XT3 (BigBen@PSC)



Summary

1. Topology is important again
2. Even on fast interconnects such as Cray

3. In presence of contention, bandwidth occupancy effects message latencies significantly
4. Increases with the number of hops each message travels

5. Topology Manager API: A uniform API for IBM and Cray machines
6. Case Studies: OpenAtom, NAMD, Stencil
7. Eventually, an automatic mapping framework

Acknowledgements:

1. Argonne National Laboratory: Pete Beckman, Tisha Stacey
2. Pittsburgh Supercomputing Center: Chad Vizino, Shawn Brown
3. Oak Ridge National Laboratory: Patrick Worley, Donald Frederick
4. IBM: Robert Walkup, Sameer Kumar
5. Cray: Larry Kaplan
6. SiCortex: Matt Reilly

References:

1. Abhinav Bhatele, Laxmikant V. Kale, **Dynamic Topology Aware Load Balancing Algorithms for MD Applications**, *To appear in Proceedings of International Conference on Supercomputing*, 2009

2. Abhinav Bhatele, Laxmikant V. Kale, **An Evaluative Study on the Effect of Contention on Message Latencies in Large Supercomputers**, *To appear in Proceedings of Workshop on Large-Scale Parallel Processing (IPDPS),* 2009

3. Abhinav Bhatele, Laxmikant V. Kale, **Benefits of Topology-aware Mapping for Mesh Topologies**, LSPP special issue of Parallel Processing Letters, 2008