IS TOPOLOGY IMPORTANT AGAIN?

Effects of Contention on Message Latencies in Large Supercomputers

Abhinav S Bhatele and Laxmikant V Kale

ACM Research Competition, SC '08

SC08 AUSTIN,TX

PARALLEL PROGRAMMING LAB
DEPT. OF COMPUTER SCIENCE, UNIVERSITY OF ILLINOIS
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

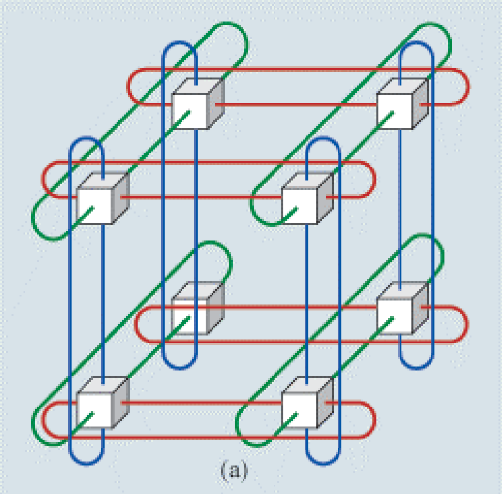
Obtaining topology information: TopoManager API
Case Study: OpenAtom

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

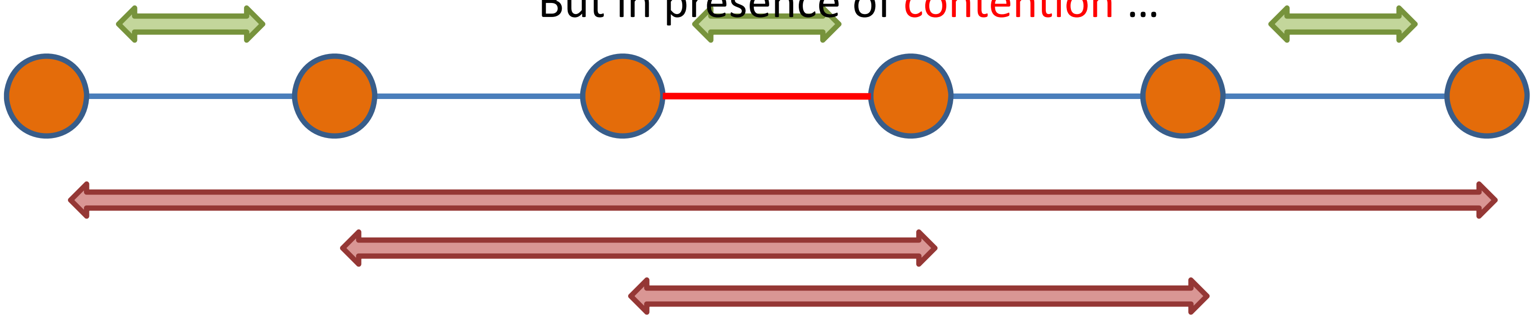
$$(L_f/B) \times D + L/B$$

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 …



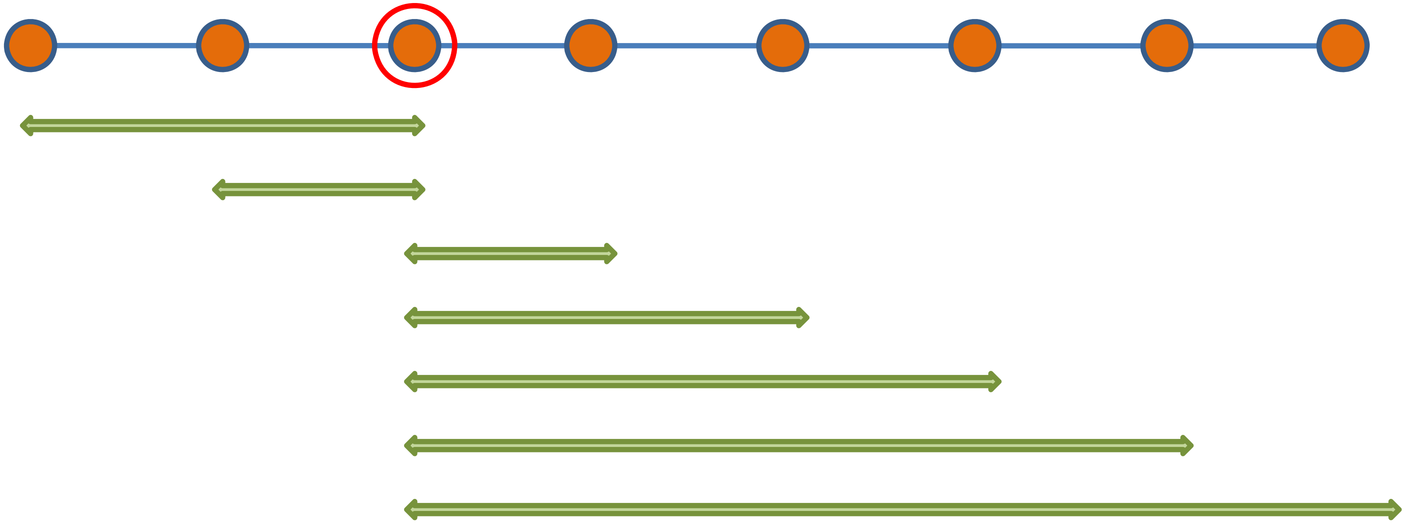
MPI Benchmarks†

- 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

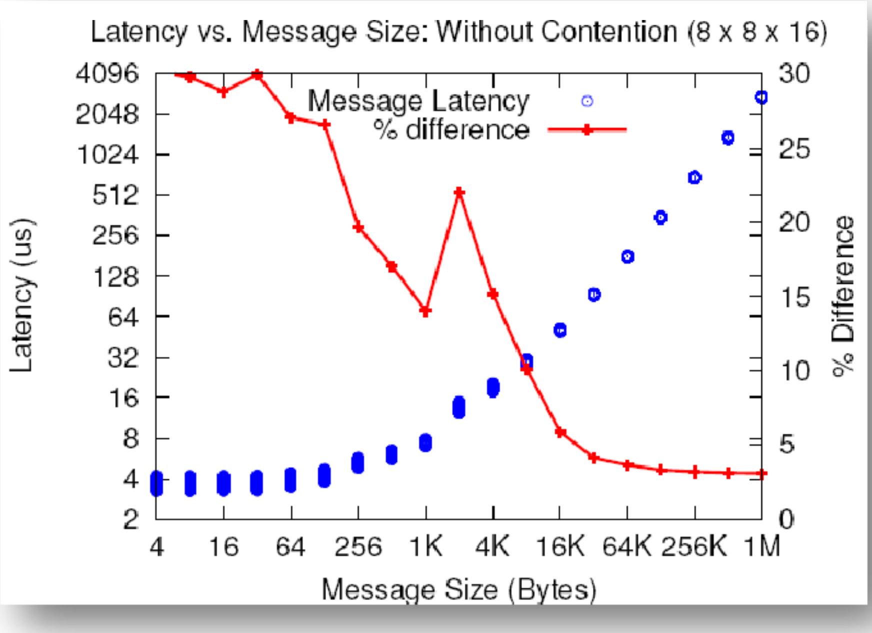
WOCON: No contention

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

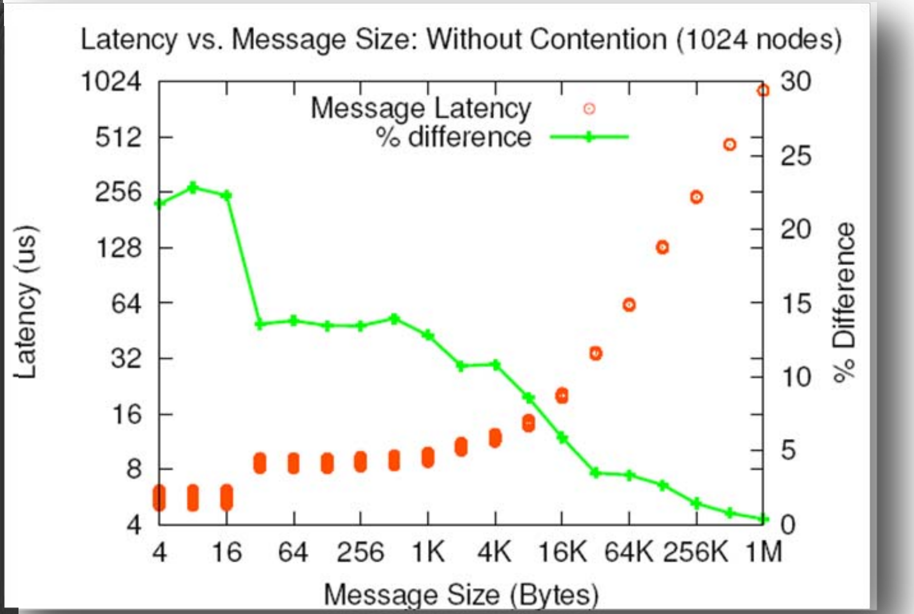


WOCON: Results

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



ANL Blue Gene/P

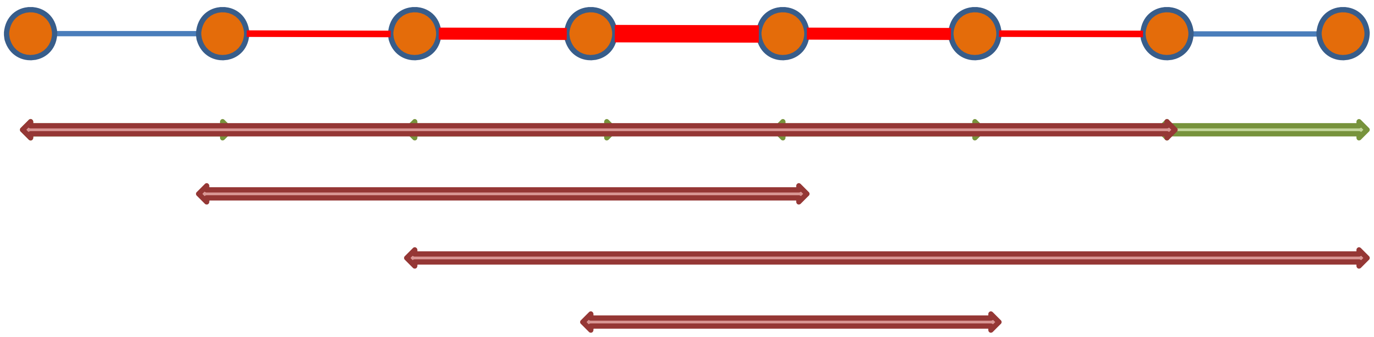


ORNL XT4

PSC XT3

WICON: With Contention

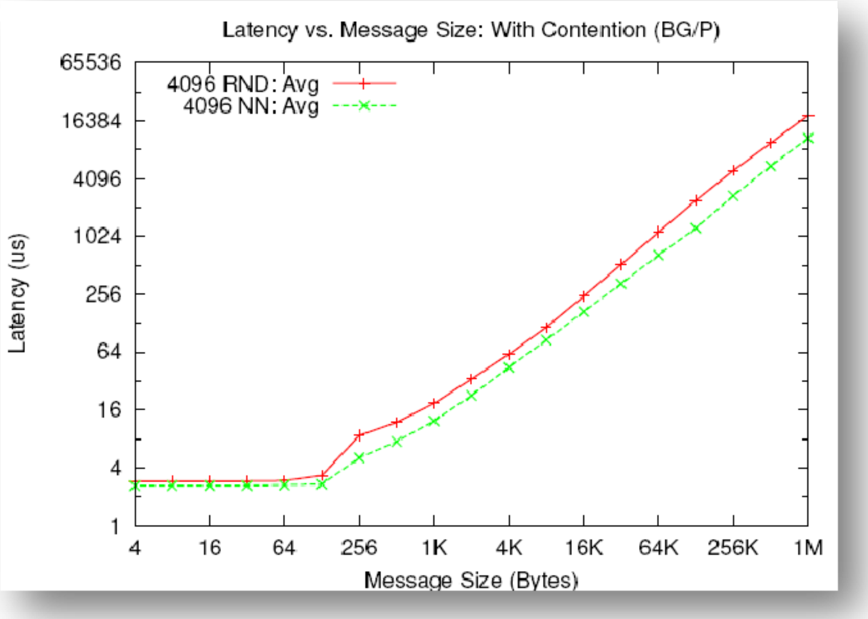
- Divide all ranks into pairs and everyone sends to their respective partner simultaneously



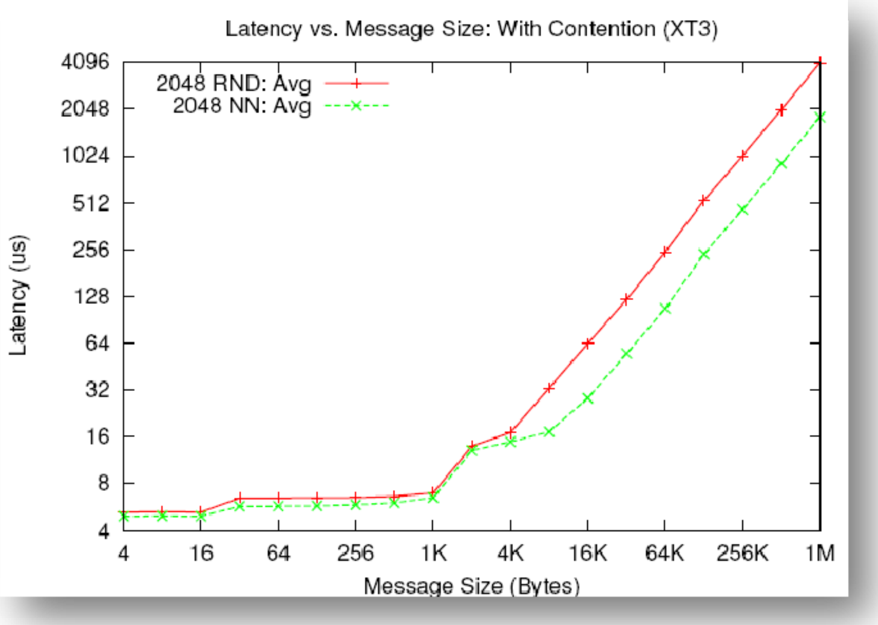
Near Neighbor: NN

Random: RND

WICON: Results



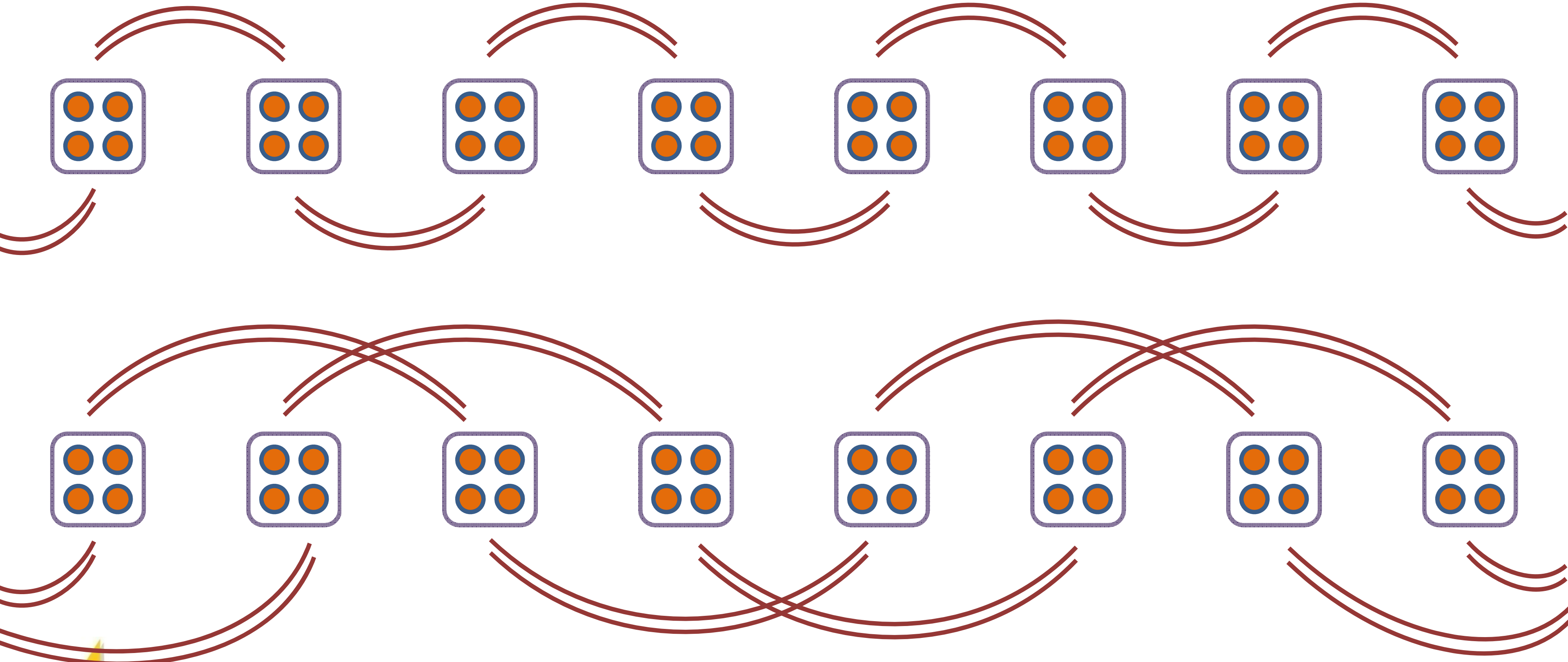
ANL Blue Gene/P

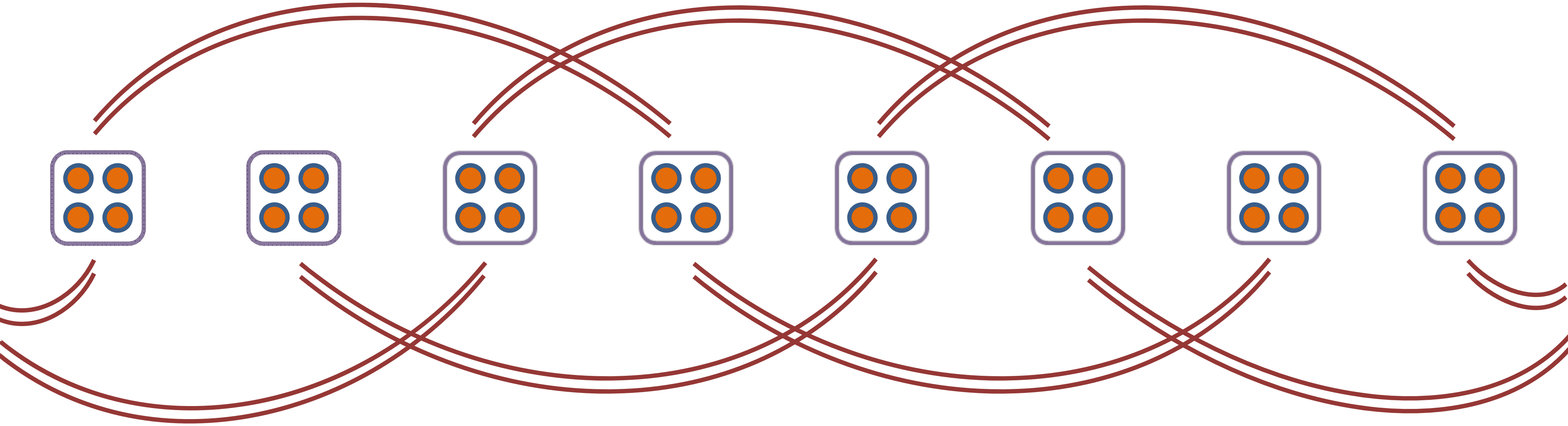


PSC XT3

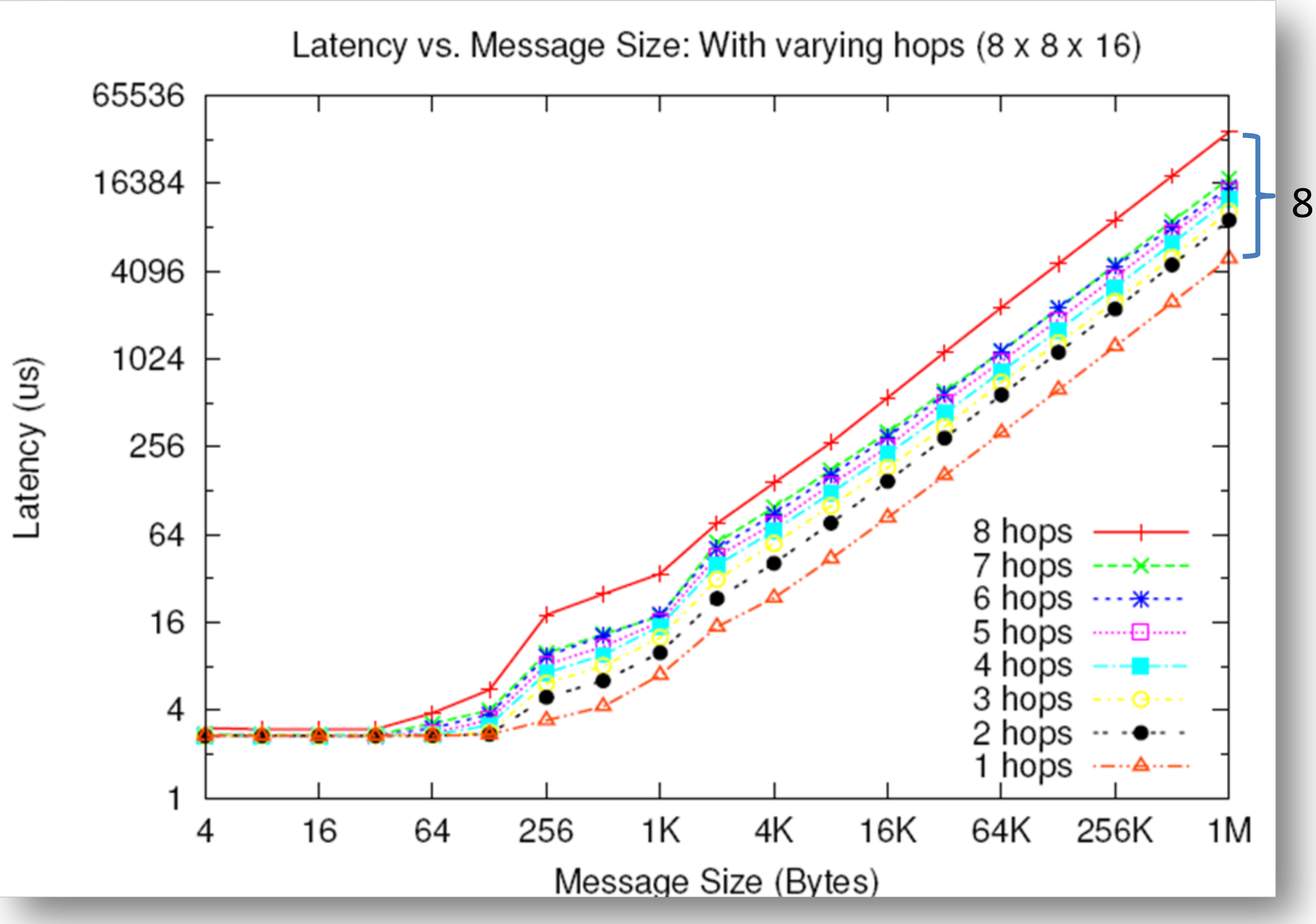
Message Latencies and Hops

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

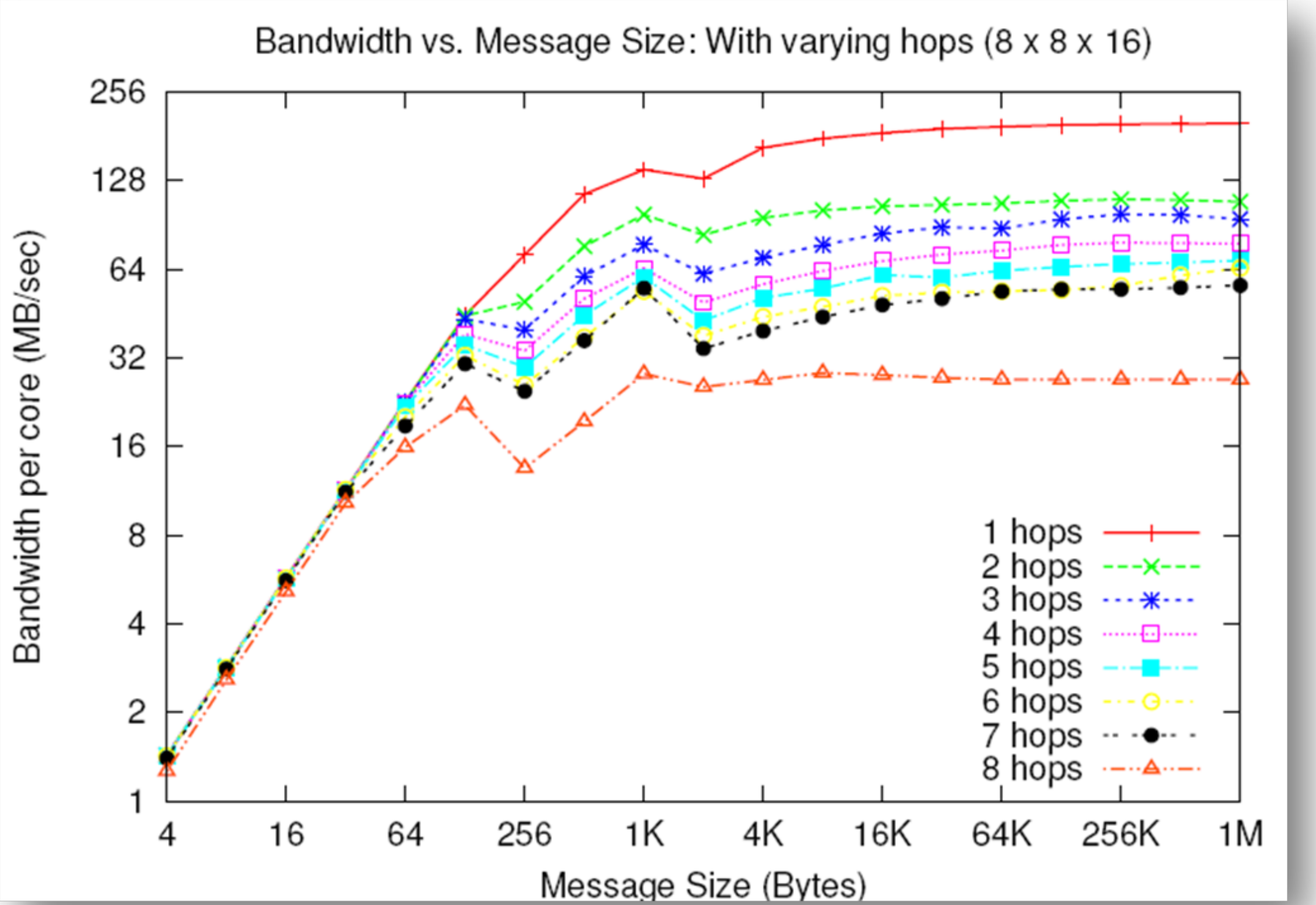




Results



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
Message Size (Bytes)
4
16
64
256
1K
4K
16K
64K
256K
1M
1 hops
2 hops
3 hops
4 hops
5 hops
6 hops
7 hops
8 hops

Difference from previous work

Then	Now
Mainly for theoretical object graphs on hypercubes, shuffle exchange and other theoretical networks	Object graphs from real applications on 3D torus/mesh topologies
Most techniques were used offline – slow	Fast, runtime solutions
Demonstrated on graphs with 10-100 nodes	Scalable techniques for large machines
No cardinality variation	Multiple objects per processor
Not tested with real applications on actual machines – theoretical work	Targeted at production codes – tested with real applications

Difference from recent work

[5] G. Bhanot, A. Gara, P. Heidelberger, E. Lawless, J. C. Sexton, and R. Walkup. Optimizing task layout on the Blue Gene/L supercomputer. IBM Journal of Research and Development, 49(2/3), 2005.

- Use of simulated annealing – slow and solution is developed offline

[6] Hao Yu, I-Hsin Chung, and Jose Moreira. Topology mapping for Blue Gene/L supercomputer. In SC '06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing, page 116, New York, NY, USA, 2006. ACM

- Node mappings for simple scenarios (1D rings, 2D meshes, 3D)
- Only useful in case of simple near-neighbor communication

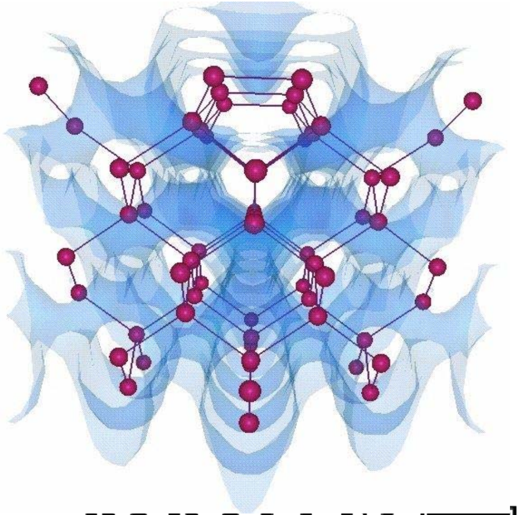
Topology Manager API†

- 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

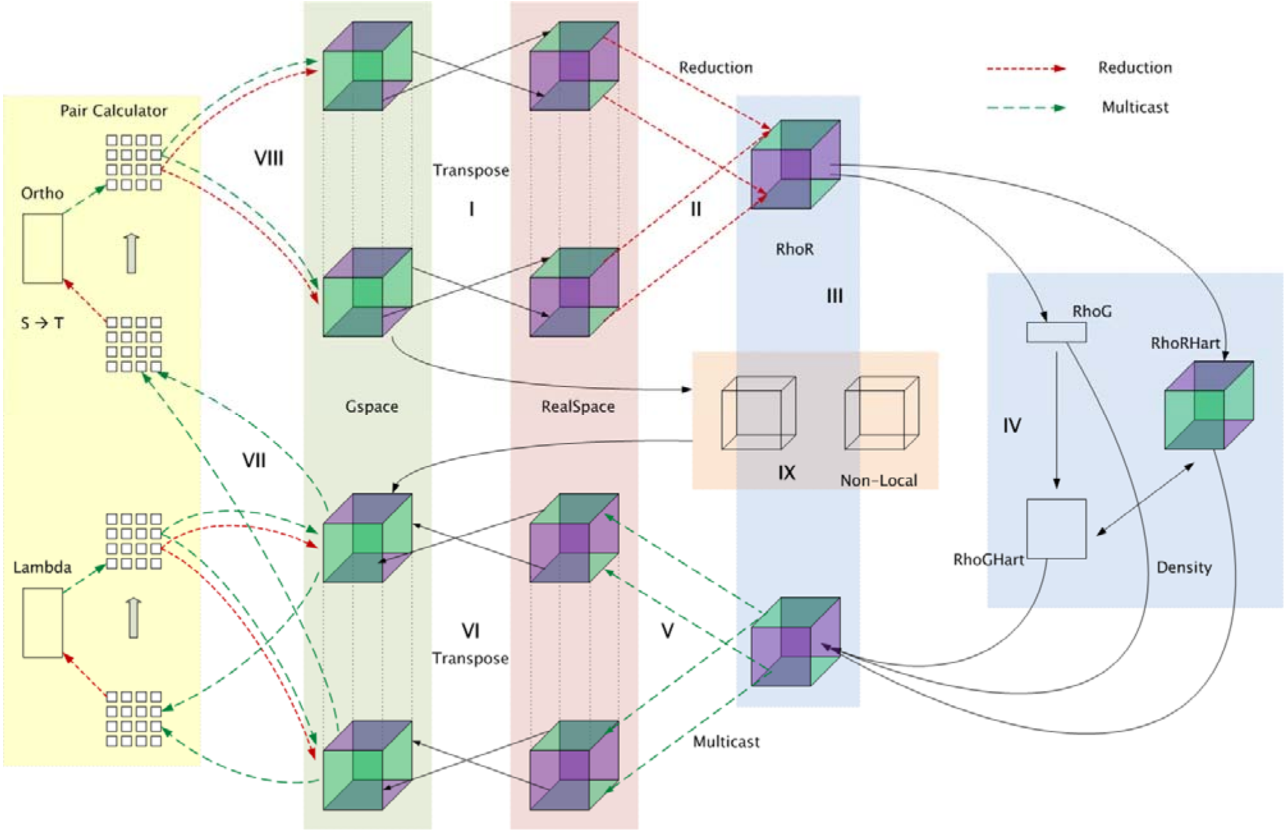
† http://charm.cs.uiuc.edu/~bhatele/phd/topomgr.htm

OpenAtom

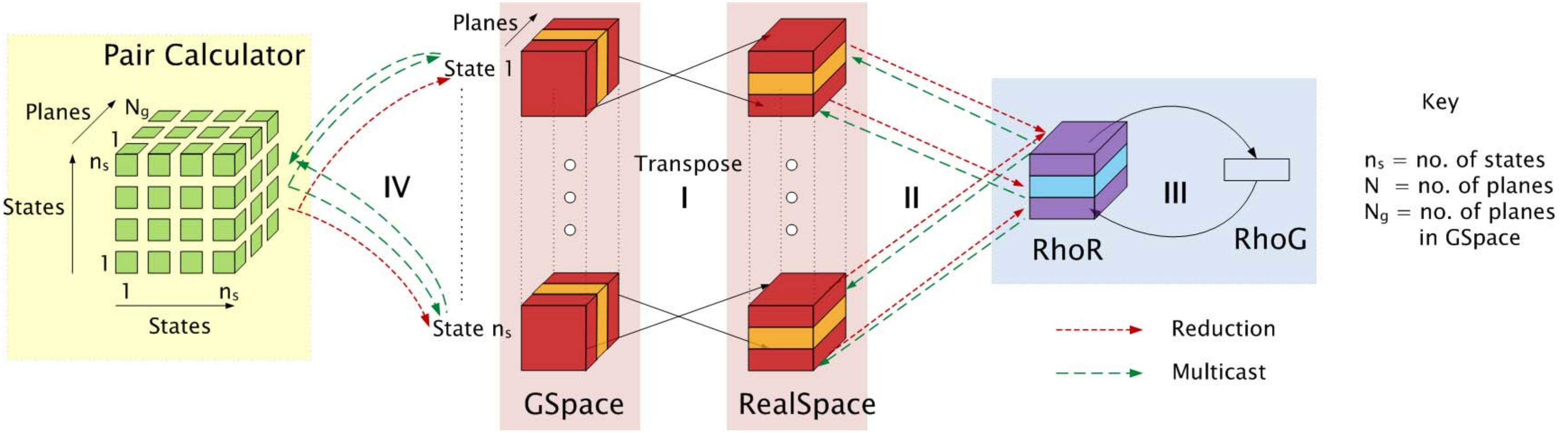
- Ab-Initio Molecular Dynamics code
- Communication is static and structured
- Challenge: Multiple groups of objects with conflicting communication patterns



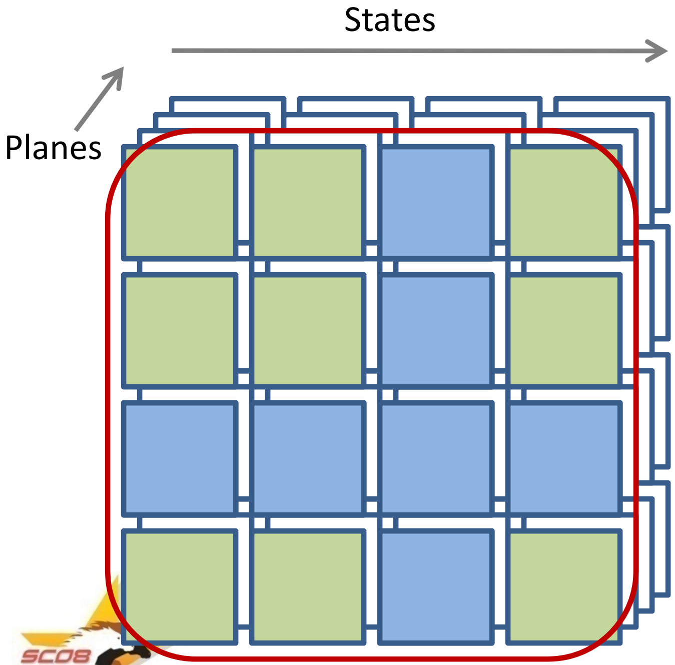
Parallelization using Charm++



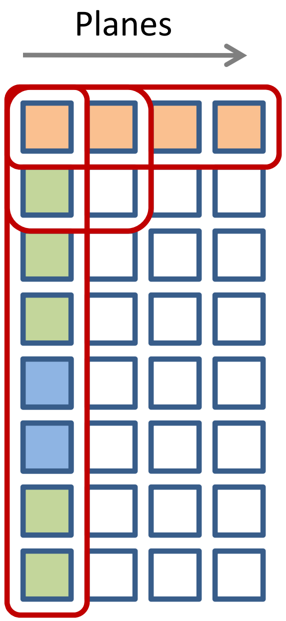
[10] Eric Bohm, Glenn J. Martyna, Abhinav Bhatele, Sameer Kumar, Laxmikant V. Kale, John A. Gunnels, and Mark E. Tuckerman. **Fine Grained Parallelization of the Car-Parrinello ab initio MD Method on Blue Gene/L**. *IBM J. of R. and D.: Applications of Massively Parallel Systems, 52(1/2):159-174*, 2008.



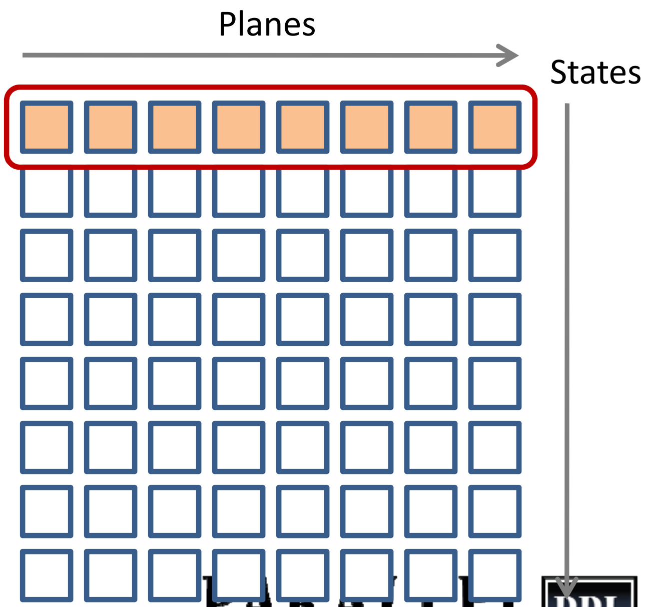
PairCalculator



GSpace

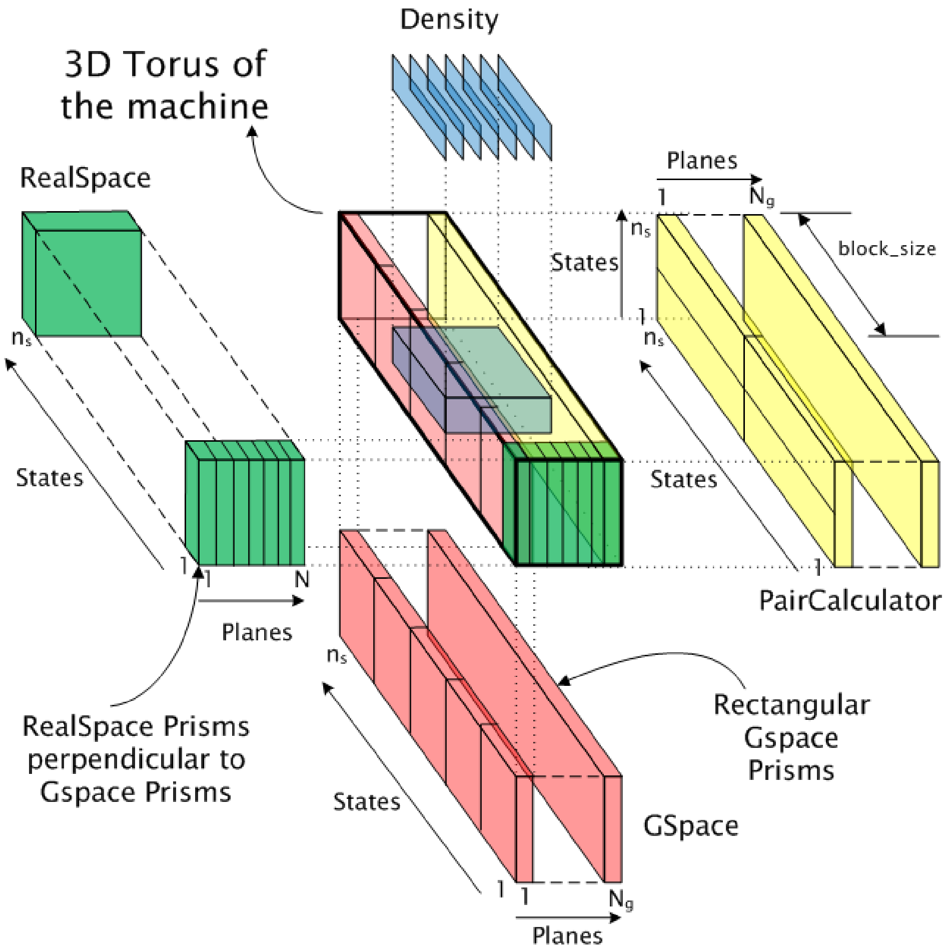


RealSpace



Topology Mapping of Chare Arrays

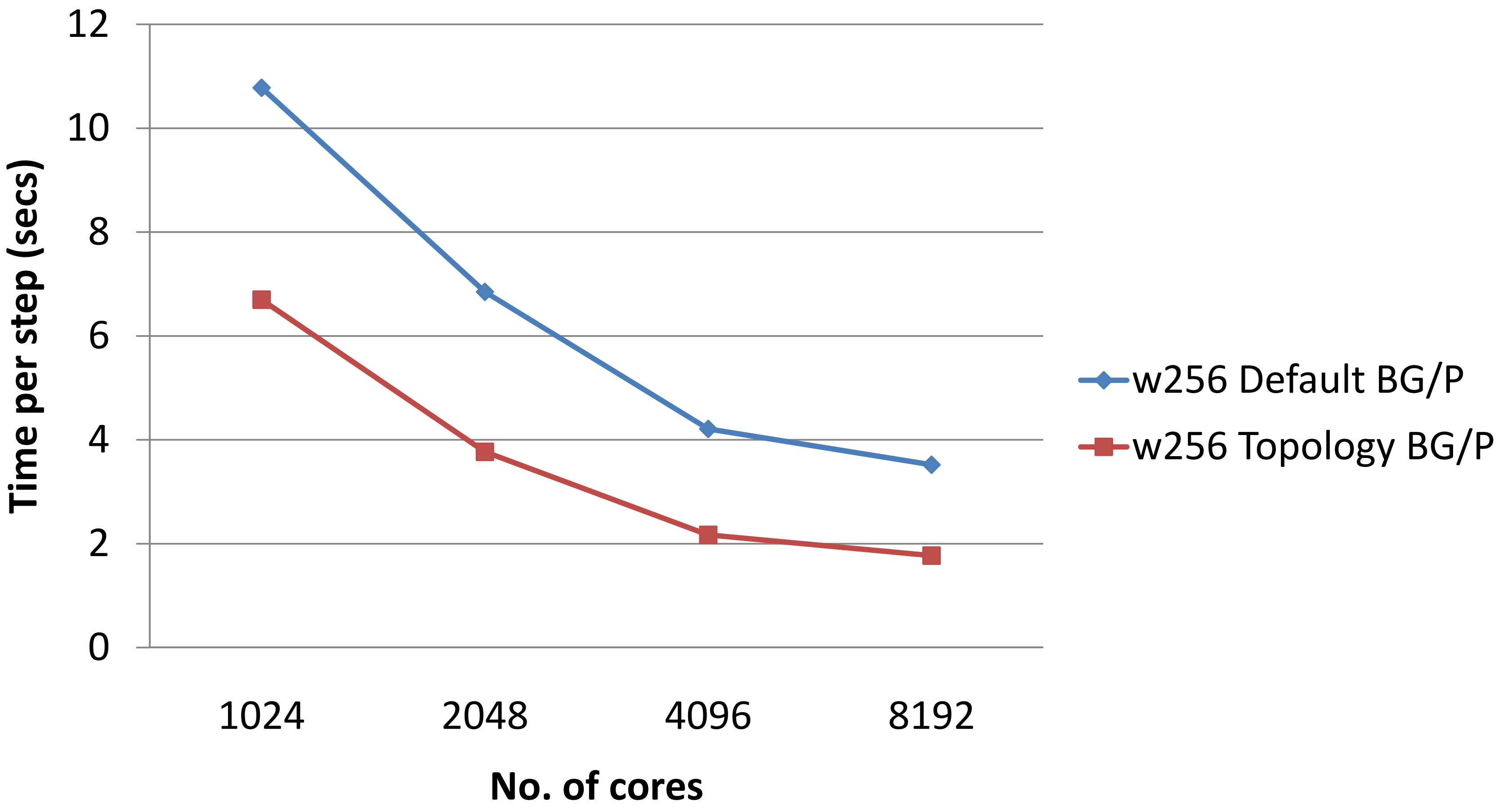
State-wise communication



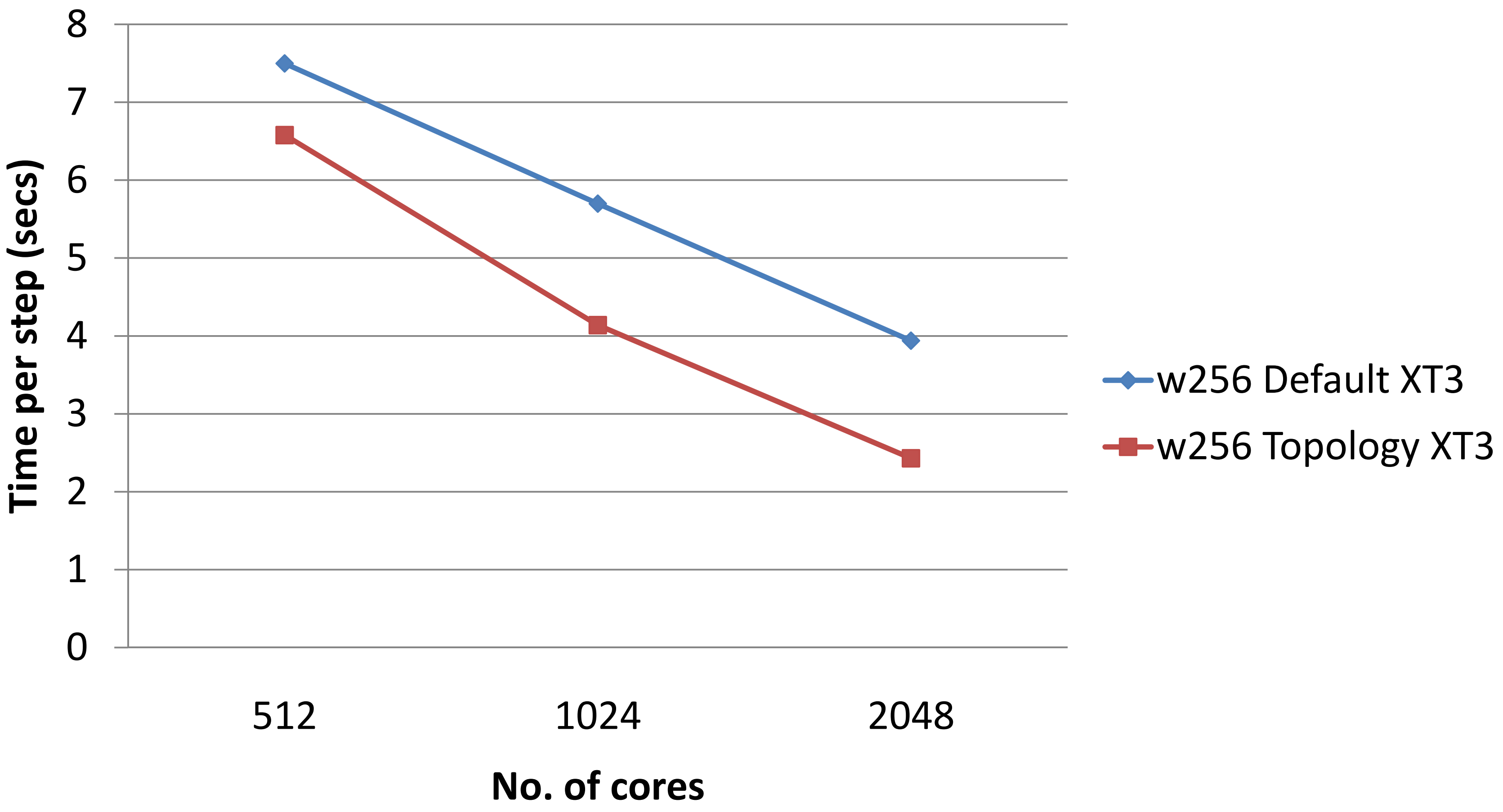
Plane-wise communication

Joint work with Eric J. Bohm

Results on Blue Gene/P (ANL)



Results on XT3 (BigBen)



Summary

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

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

1. Topology Manager API: A uniform API for IBM and Cray machines
2. Case Studies: OpenAtom, NAMD, Stencil
3. Eventually, an automatic mapping framework

Acknowledgements:

1. Argonne National Lab: 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**, *submitted to Philosophical Transactions of the Royal Society A*, 2008

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

3. Abhinav Bhatele, Laxmikant V. Kale, **Application-specific Topology-aware Mapping for Three Dimensional Topologies**, *Proceedings of Workshop on Large-Scale Parallel Processing (held as part of IPDPS '08)*, 2008