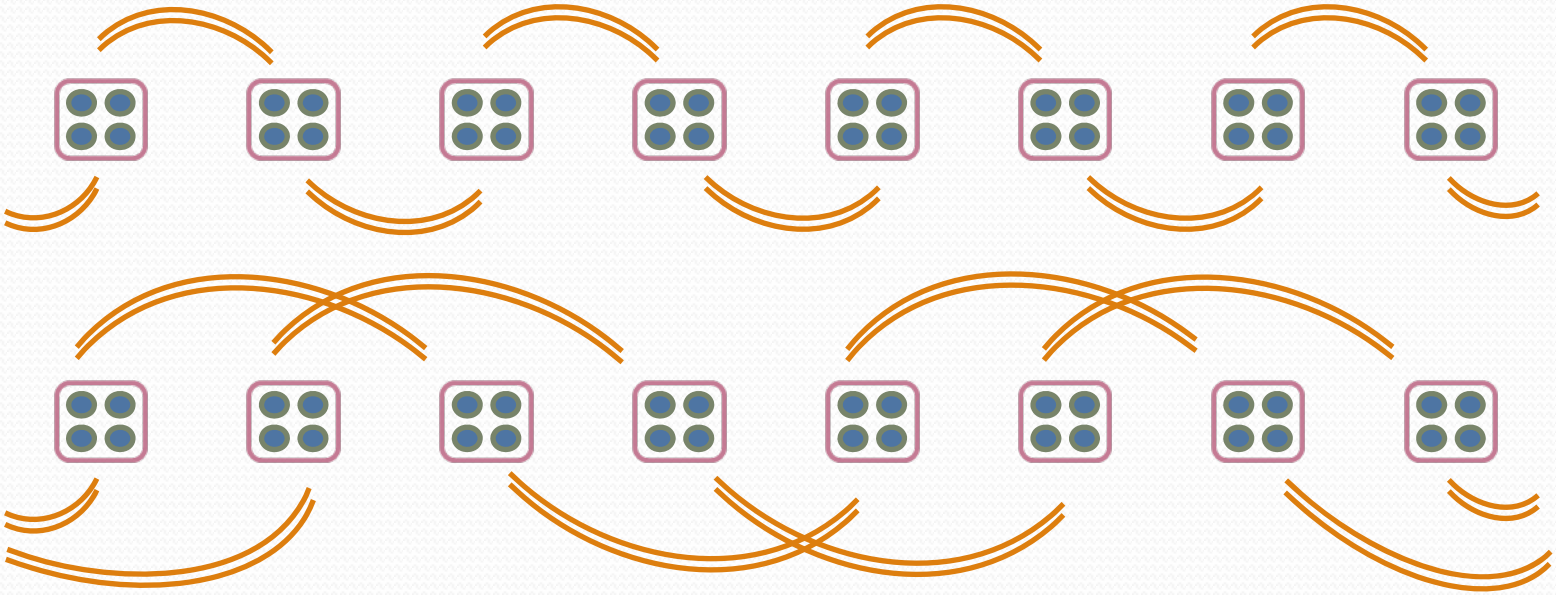
Dynamic Topology Aware Load Balancing Algorithms for MD Applications

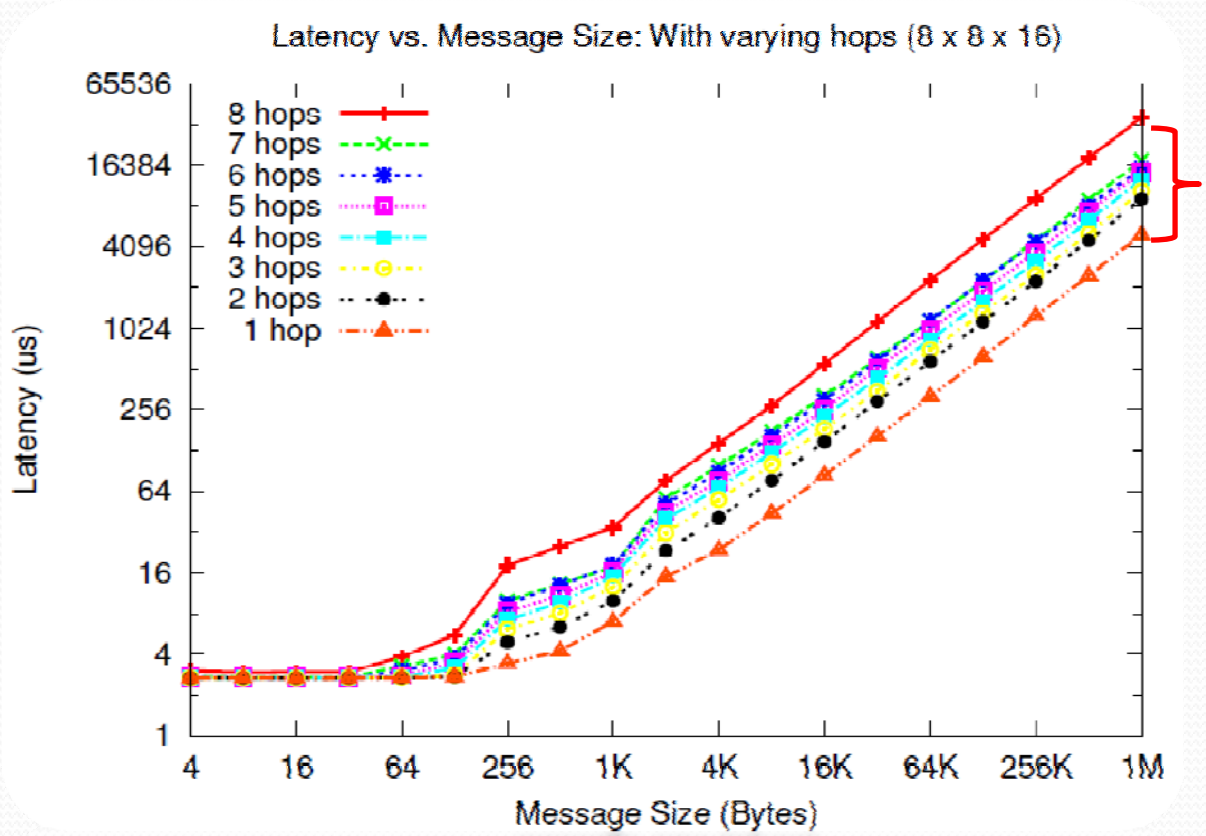
Abhinav Bhatele, Laxmikant V. Kale
University of Illinois at Urbana-Champaign
Sameer Kumar
IBM T. J. Watson Research Center

Motivation: Contention Experiments



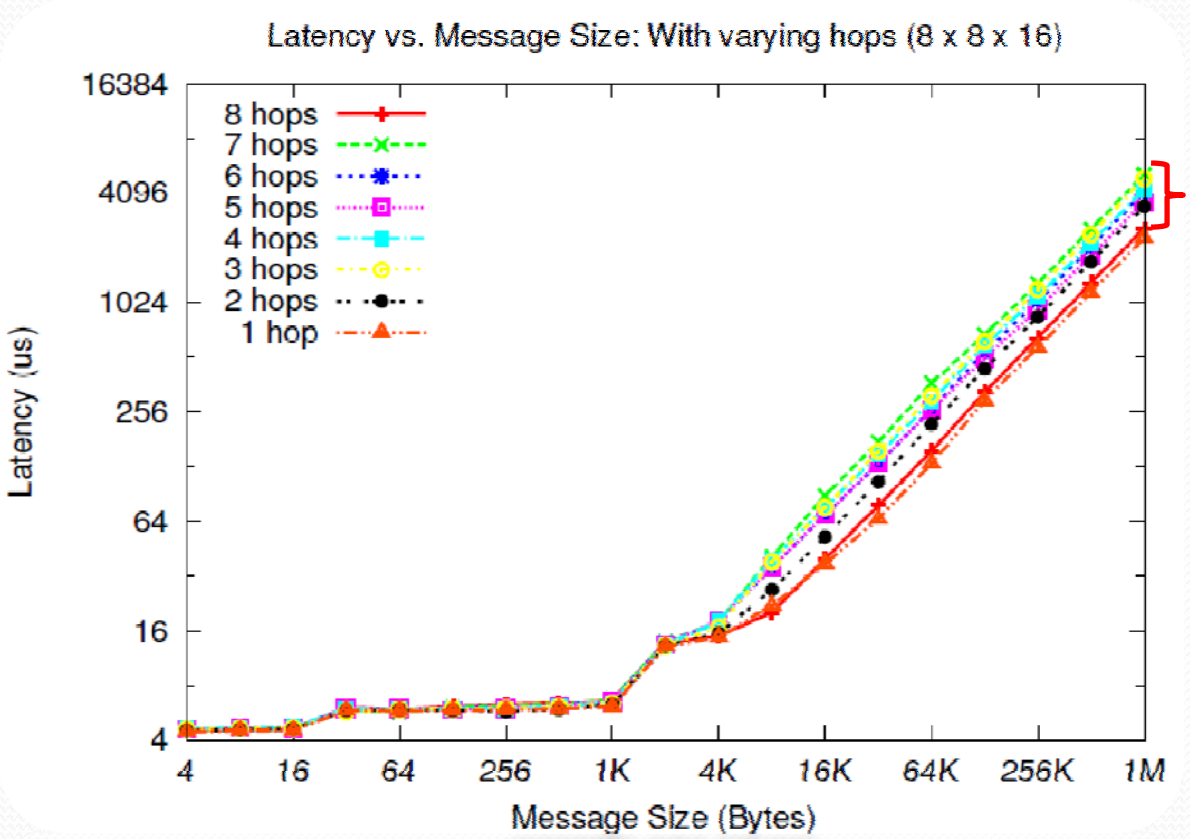
Bhatele, A., Kale, L. V. 2008 **An Evaluation of the Effect of Interconnect Topologies on Message Latencies in Large Supercomputers**. In *Proceedings of Workshop on Large-Scale Parallel Processing (IPDPS)*, Rome, Italy, May 2009.

Results: Blue Gene/P



7.39 times

Results: Cray XT3



2.23 times

Molecular Dynamics

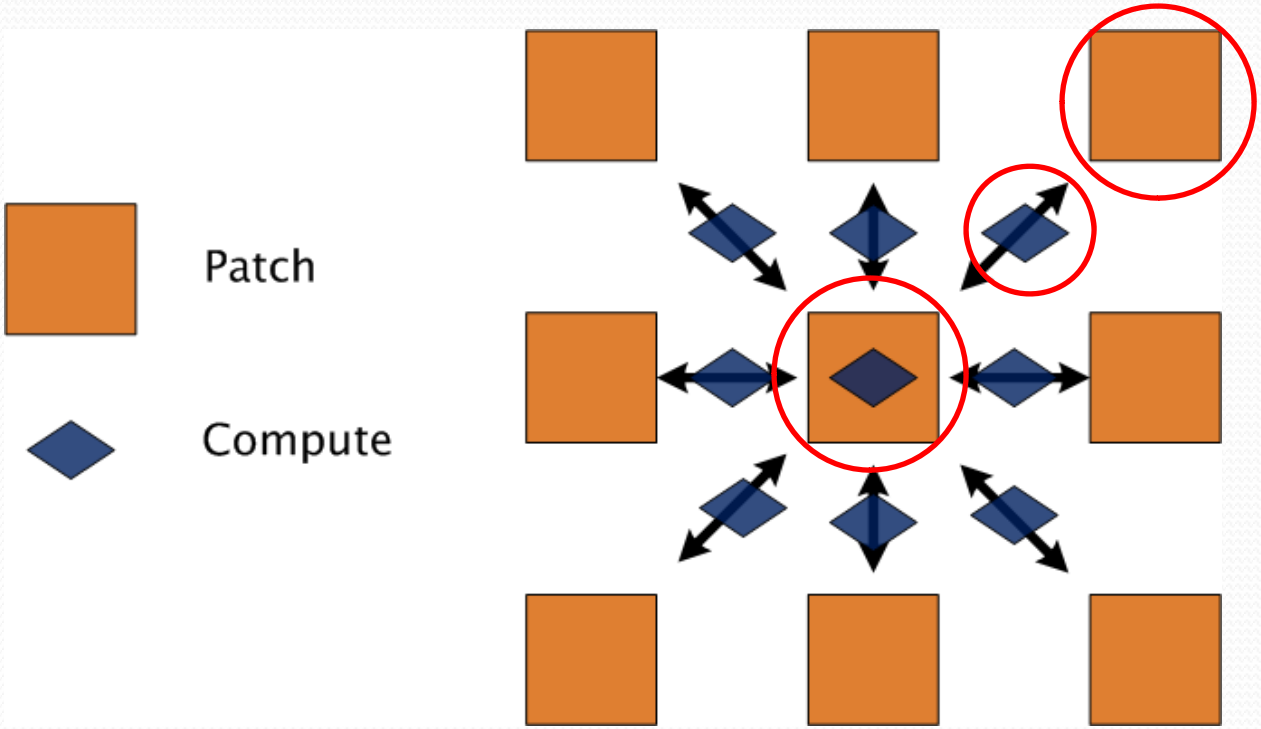
- A system of [charged] atoms with bonds
- Use Newtonian Mechanics to find the positions and velocities of atoms
- Each time-step is typically in femto-seconds
- At each time step
  - calculate the forces on all atoms
  - calculate the velocities and move atoms around

NAMD: NAnoscale Molecular Dynamics

- Naïve force calculation is $O(N^2)$
- Reduced to O(N logN) by calculating
  - Bonded forces
  - Non-bonded: using a cutoff radius
    - Short-range: calculated every time step
    - Long-range: calculated every fourth time-step (PME)

NAMD's Parallel Design

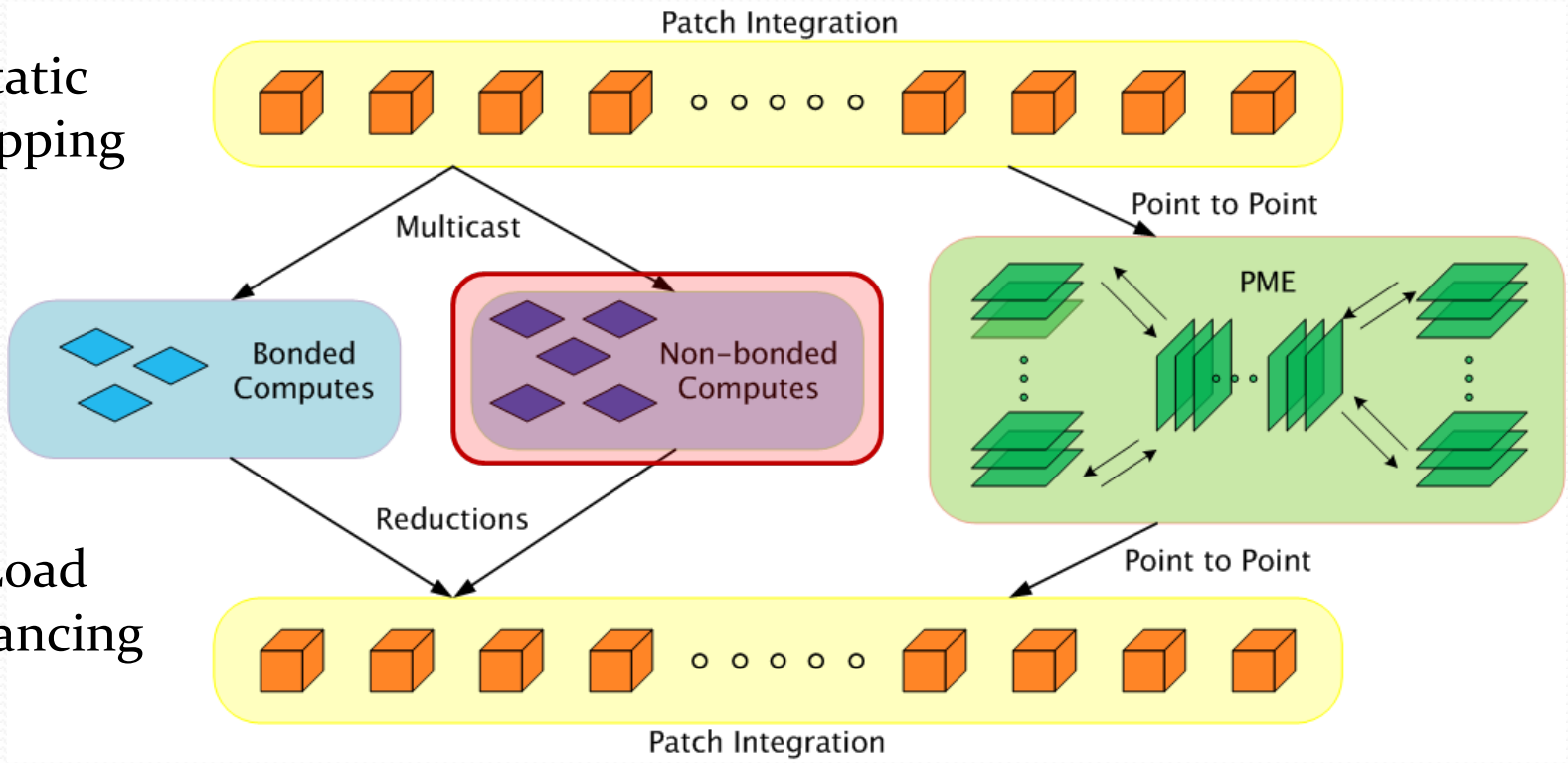
- Hybrid of spatial and force decomposition



Parallelization using Charm++

Static Mapping

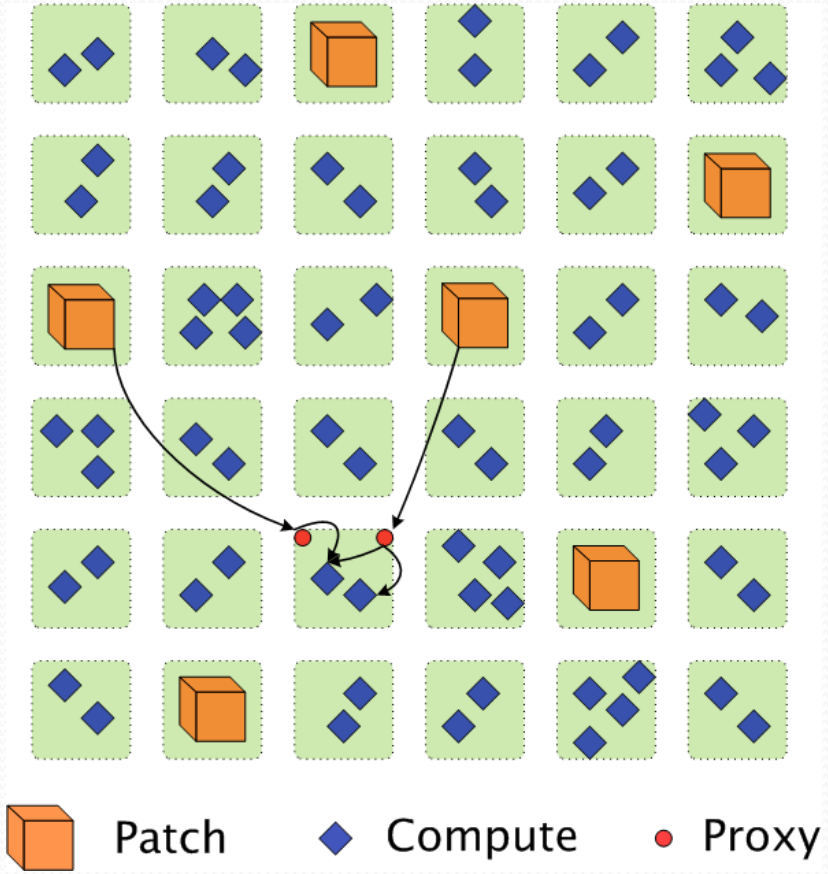
Load Balancing



Bhatele, A., Kumar, S., Mei, C., Phillips, J. C., Zheng, G. & Kale, L. V. 2008 **Overcoming Scaling Challenges in Biomolecular Simulations across Multiple Platforms**. In *Proceedings of IEEE International Parallel and Distributed Processing Symposium*, Miami, FL, USA, April 2008.

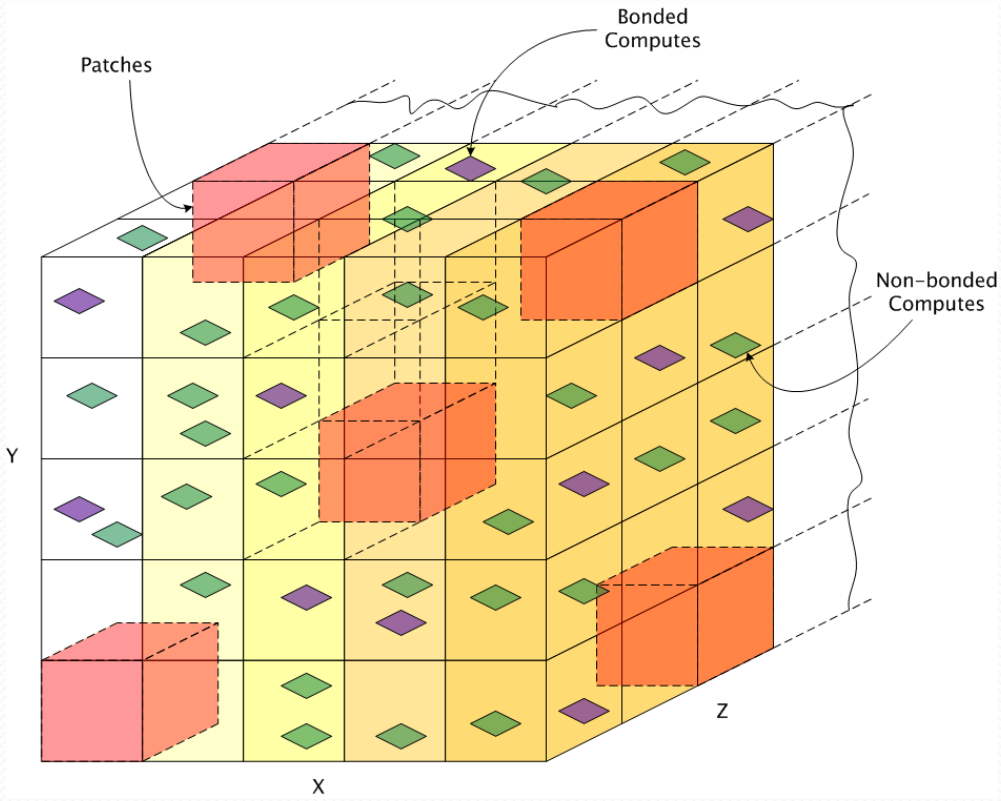
Communication in NAMD

- Each patch multicasts its information to many computes
- Each compute is a target of two multicasts only
- Use 'Proxies' to send data to different computes on the same processor



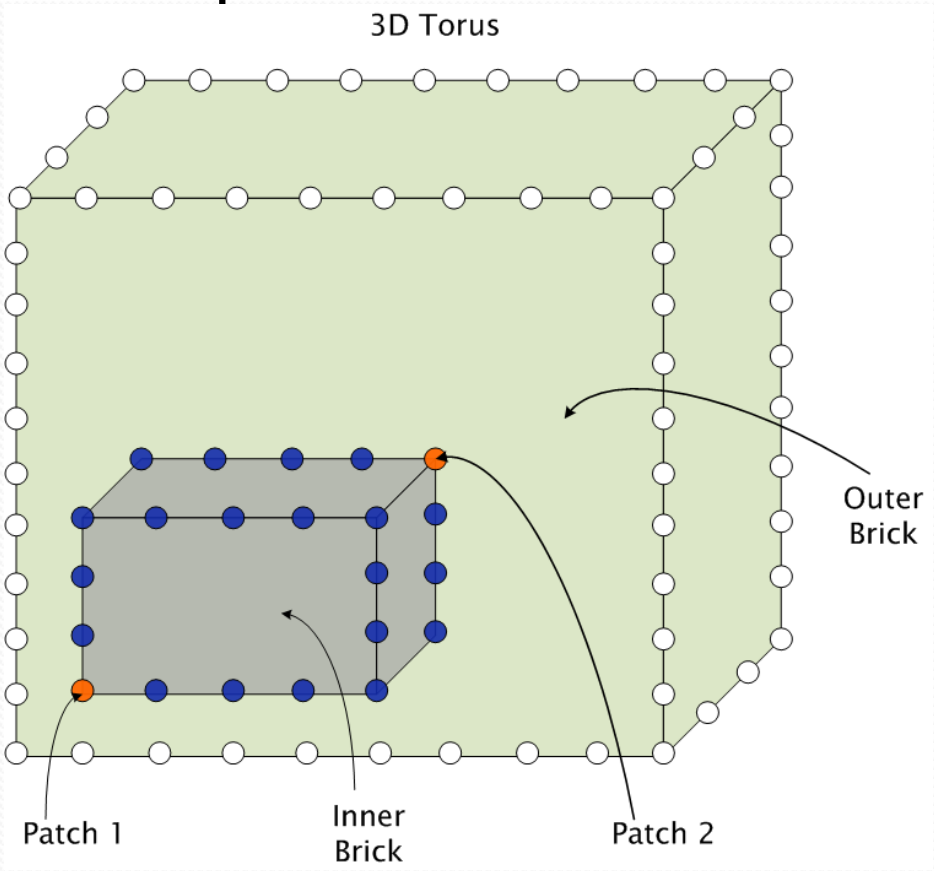
Topology Aware Techniques

- Static Placement of Patches



Topology Aware Techniques (contd.)

- Placement of computes



Load Balancing in Charm++

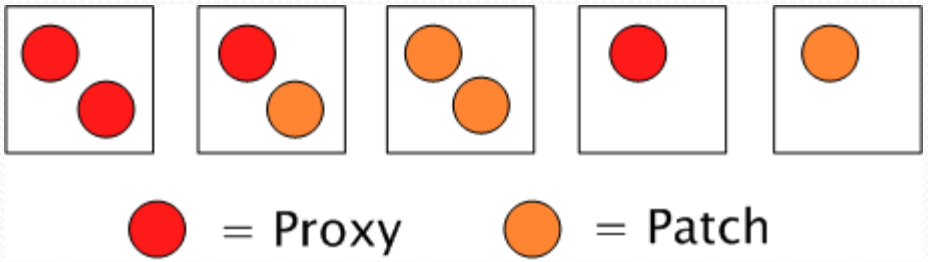
- Principle of Persistence
  - Object communication patterns and computational loads tend to persist over time
- Measurement-based Load Balancing
  - Instrument computation time and communication volume at runtime
  - Use the database to make new load balancing decisions

NAMD's Load Balancing Strategy

- NAMD uses a dynamic centralized greedy strategy
- There are two schemes in play:
  - A comprehensive strategy (called once)
  - A refinement scheme (called several times during a run)
- Algorithm:
  - Pick a compute and find a "suitable" processor to place it on

Choice of a suitable processor

- Among underloaded processors, try to:
  - Find a processor with the two patches or their proxies
  - Find a processor with one patch or a proxy
  - Pick any underloaded processor



Highest Priority → Lowest Priority

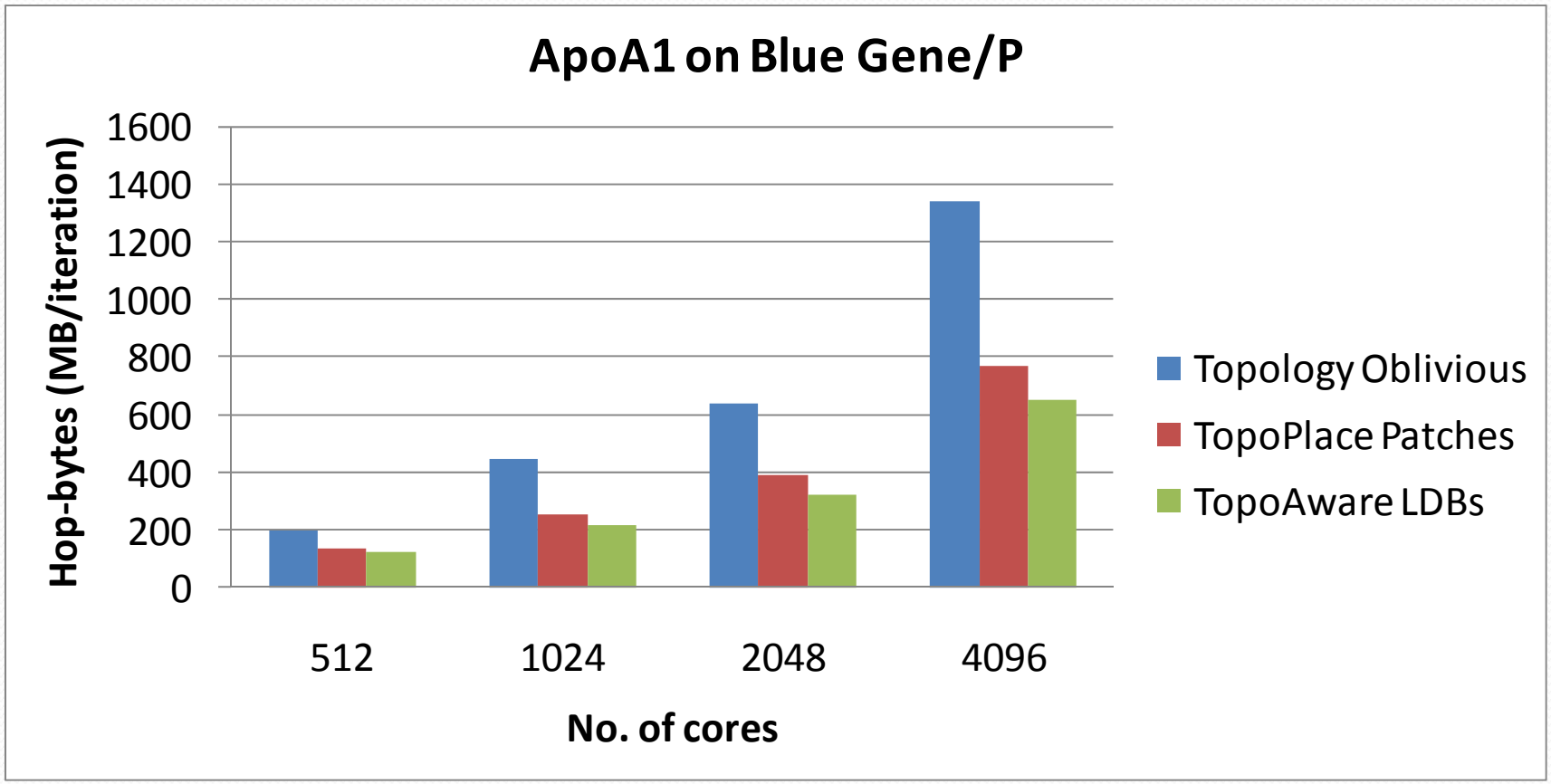
Load Balancing Metrics

- Load Balance: Bring Max-to-Avg Ratio close to 1
- Communication Volume: Minimize the number of proxies
- Communication Traffic: Minimize hop bytes

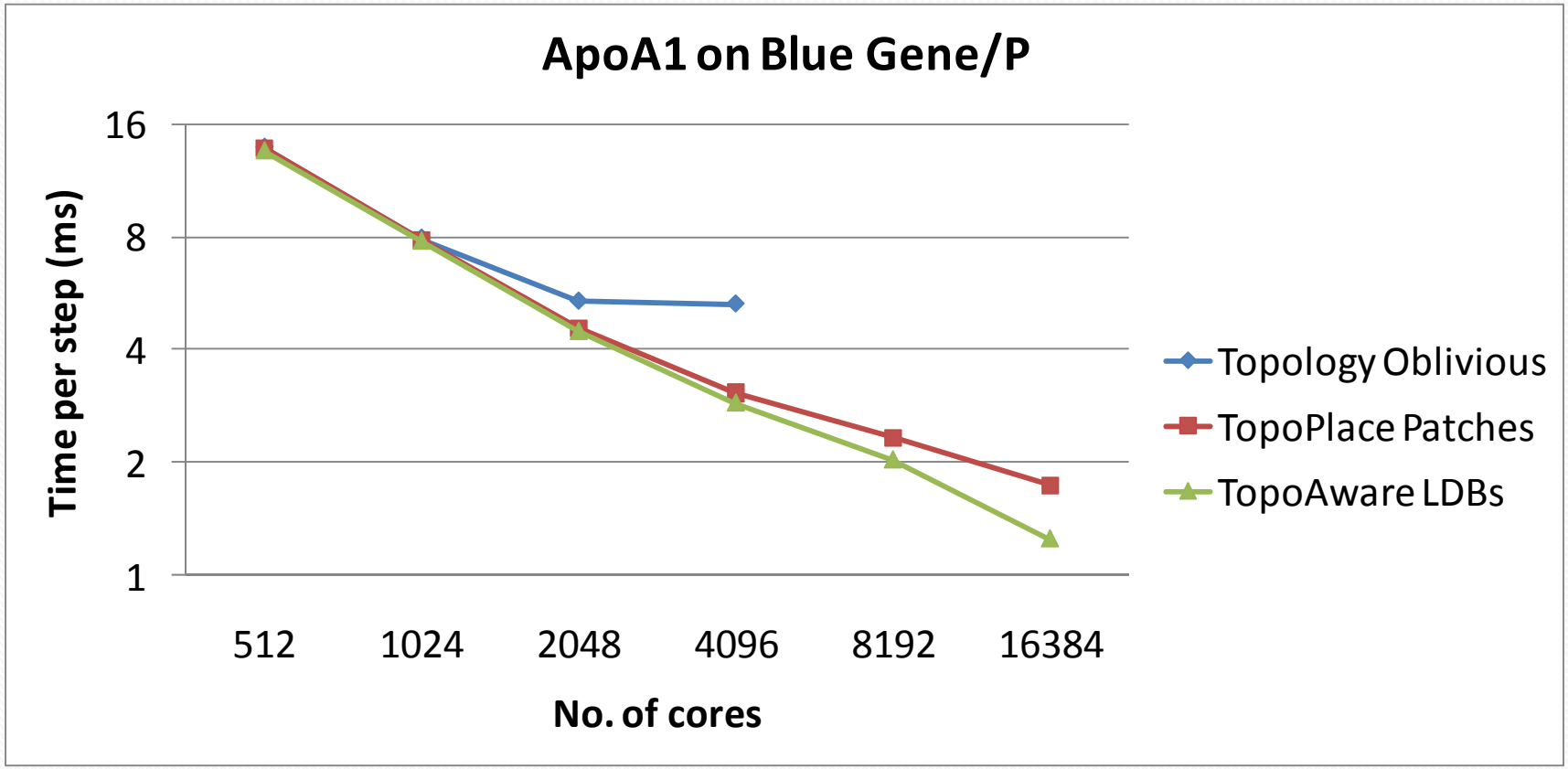
Hop-bytes = $\sum$ Message size * Hops traveled by message

Agarwal, T., Sharma, A., Kale, L.V. 2008 **Topology-aware task mapping for reducing communication contention on large parallel machines,** In *Proceedings of IEEE International Parallel and Distributed Processing Symposium*, Rhodes Island, Greece, April 2006.

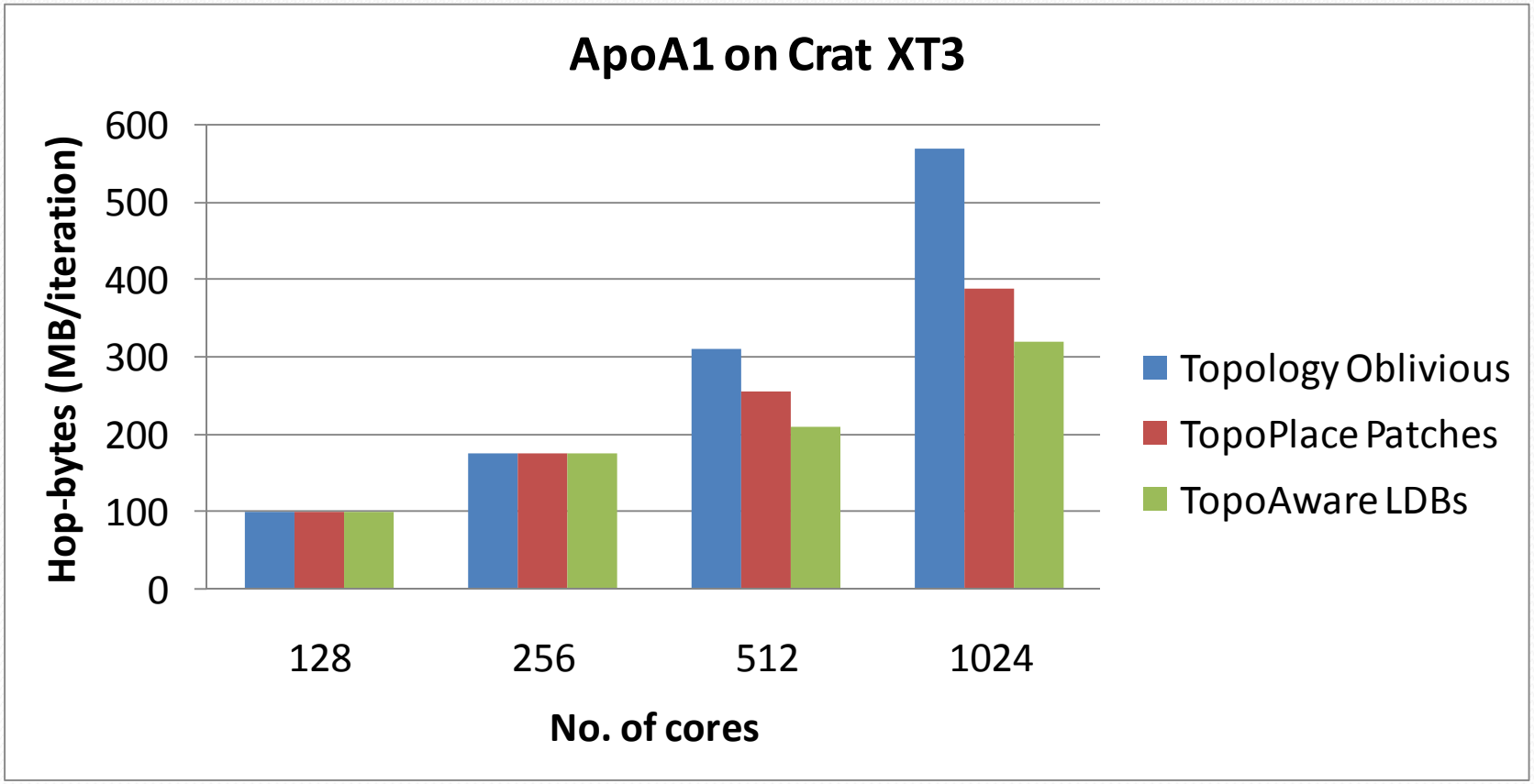
Results: Hop-bytes



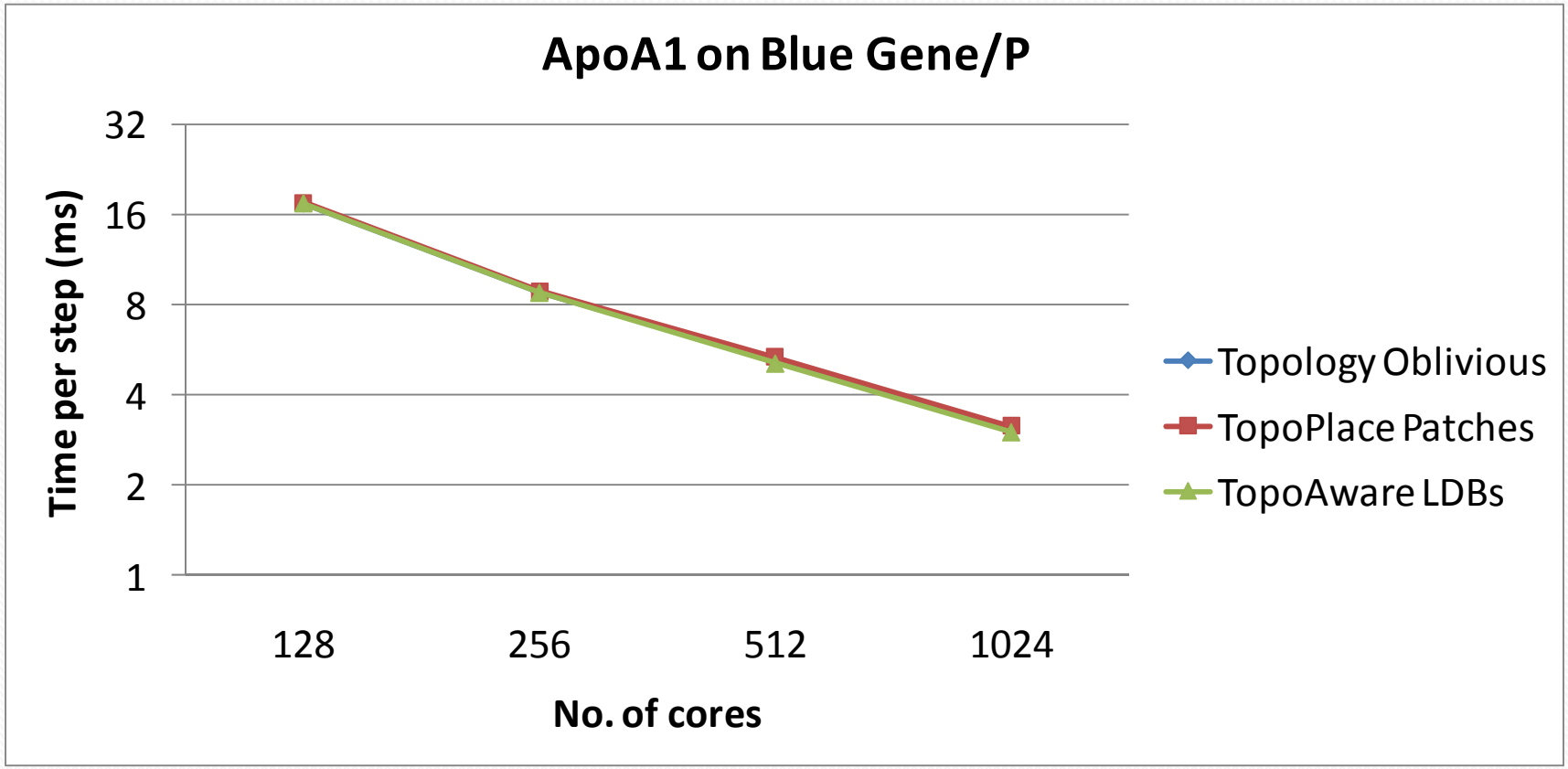
Results: Performance



Results: Hop-bytes



Results: Performance



Ongoing Work

- Observed that a simplified model was used for recording communication load
- Addition of new proxies disturbs the actual load
- Correction factor on addition/removal of proxies
- Leads to improvements of ~10%

Future Work

- SMP-aware techniques
  - Favor intra-node communication
- A scalable distributed load balancing strategy
- Generalized Scenario:
  - multicasts: each object is the target of multiple multicasts
  - use topological information to minimize communication
- Understanding the effect of various factors on load balancing in detail

Thanks!

NAMD Development Team:

Parallel Programming Lab (PPL), UIUC – Abhinav Bhatele, David Kunzman, Chee Wai Lee, Chao Mei, Gengbin Zheng, Laxmikant V. Kale

Theoretical and Computational Biophysics Group (TCBG), UIUC – James C. Phillips, Klaus Schulten

IBM Research - Sameer Kumar

Acknowledgments:

Argonne National Laboratory, Pittsburgh Supercomputing Center (Shawn Brown, Chad Vizino, Brian Johanson), TeraGrid