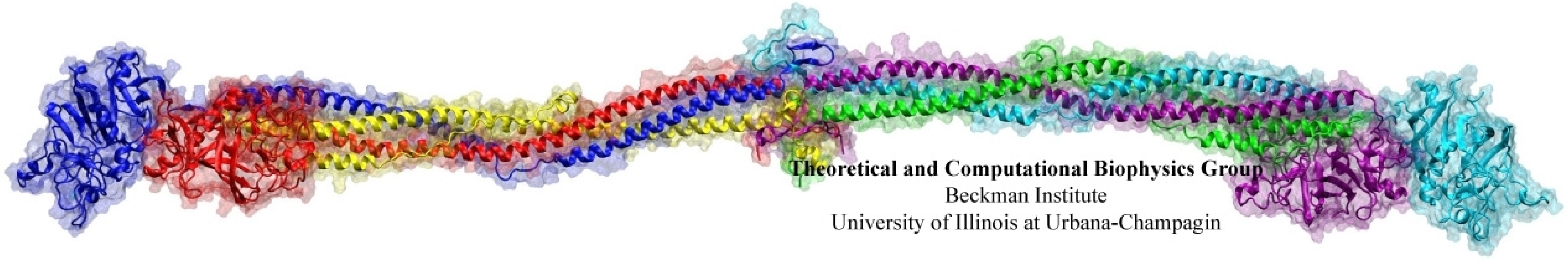

Theoretical and Computational Biophysics Group
Beckman Institute
University of Illinois at Urbana-Champagin

Overcoming Scaling Challenges in Bio-molecular Simulations

Abhinav Bhatelé, David Kunzman, Chee Wai Lee, Chao Mei, Osman Sarood, Yanhua Sun, Gengbin Zheng, Laxmikant V. Kalé (PPL)
Robert Brunner, James C. Phillips, Klaus Schulten (TCBG)
Sameer Kumar (IBM Research)

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

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

Outline

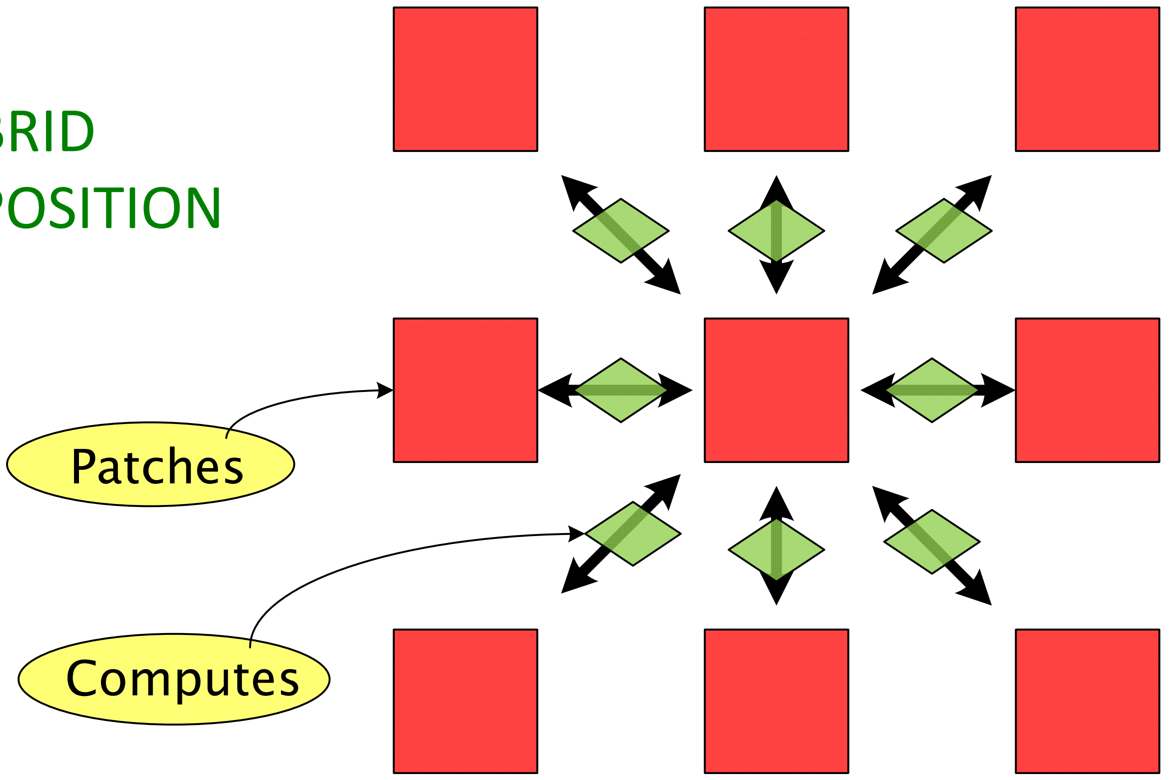
- NAMD: An Introduction
- Scaling Challenges
  - Conflicting Adaptive Runtime Techniques
  - PME Computation
  - Memory Requirements
- Performance Results
- Recent Work
- Summary

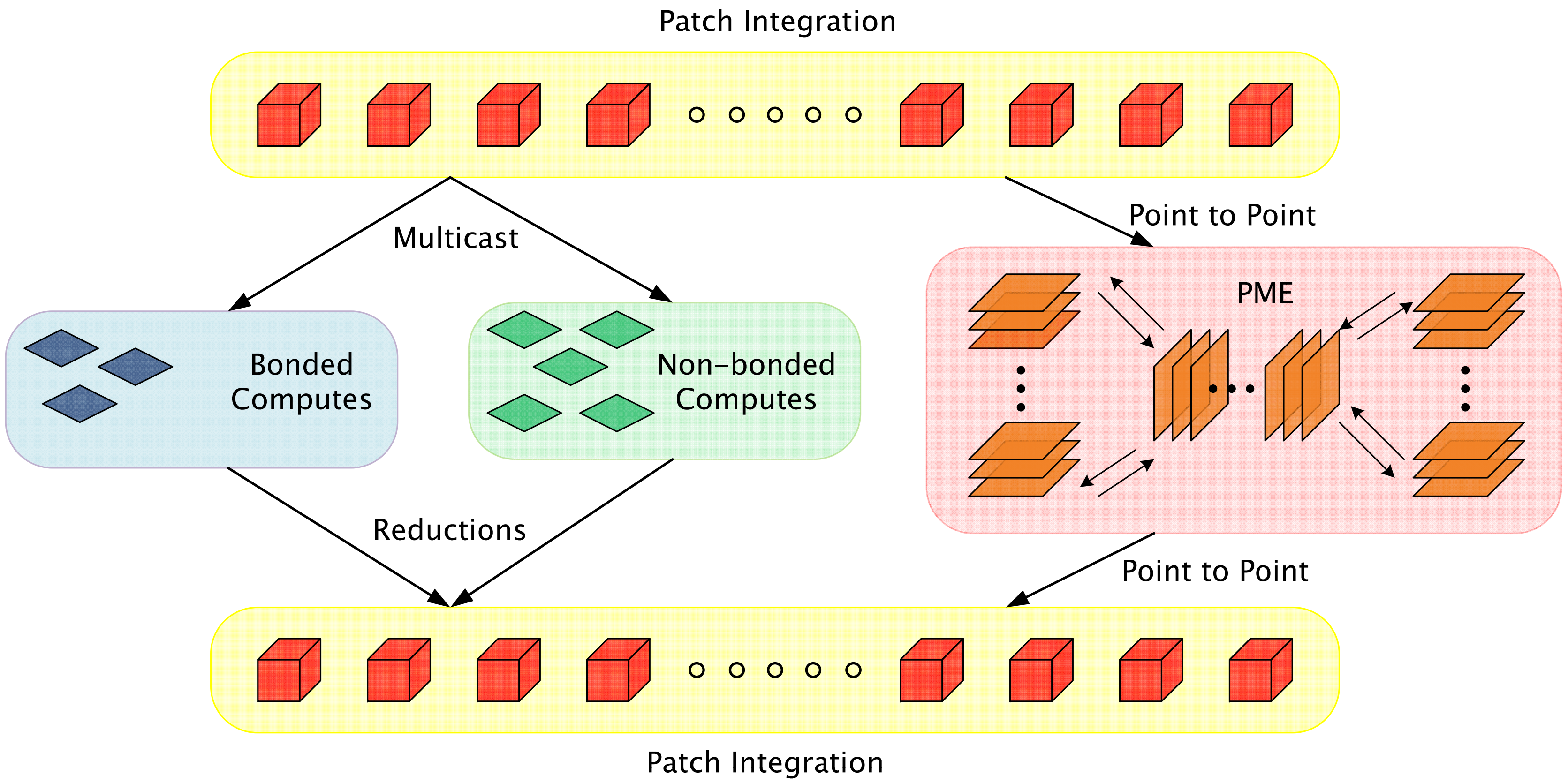
What is NAMD ?

- A parallel molecular dynamics application
- Simulate the life of a bio-molecule
- How is the simulation performed ?
  - Simulation window broken down into a large number of time steps (typically 1 fs each)
  - Forces on every atom calculated every time step
  - Velocities and positions updated and atoms migrated to their new positions

How is NAMD parallelized ?

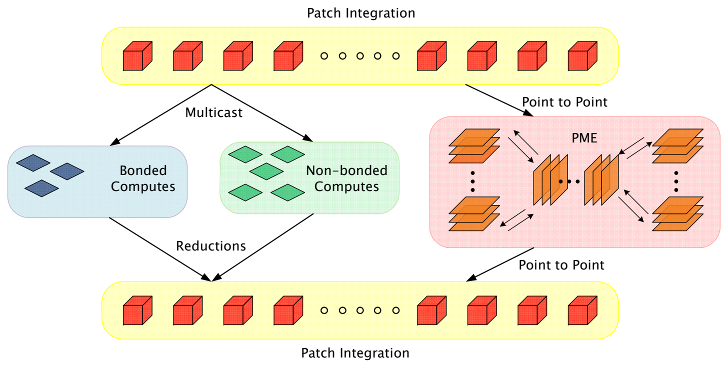
HYBRID DECOMPOSITION

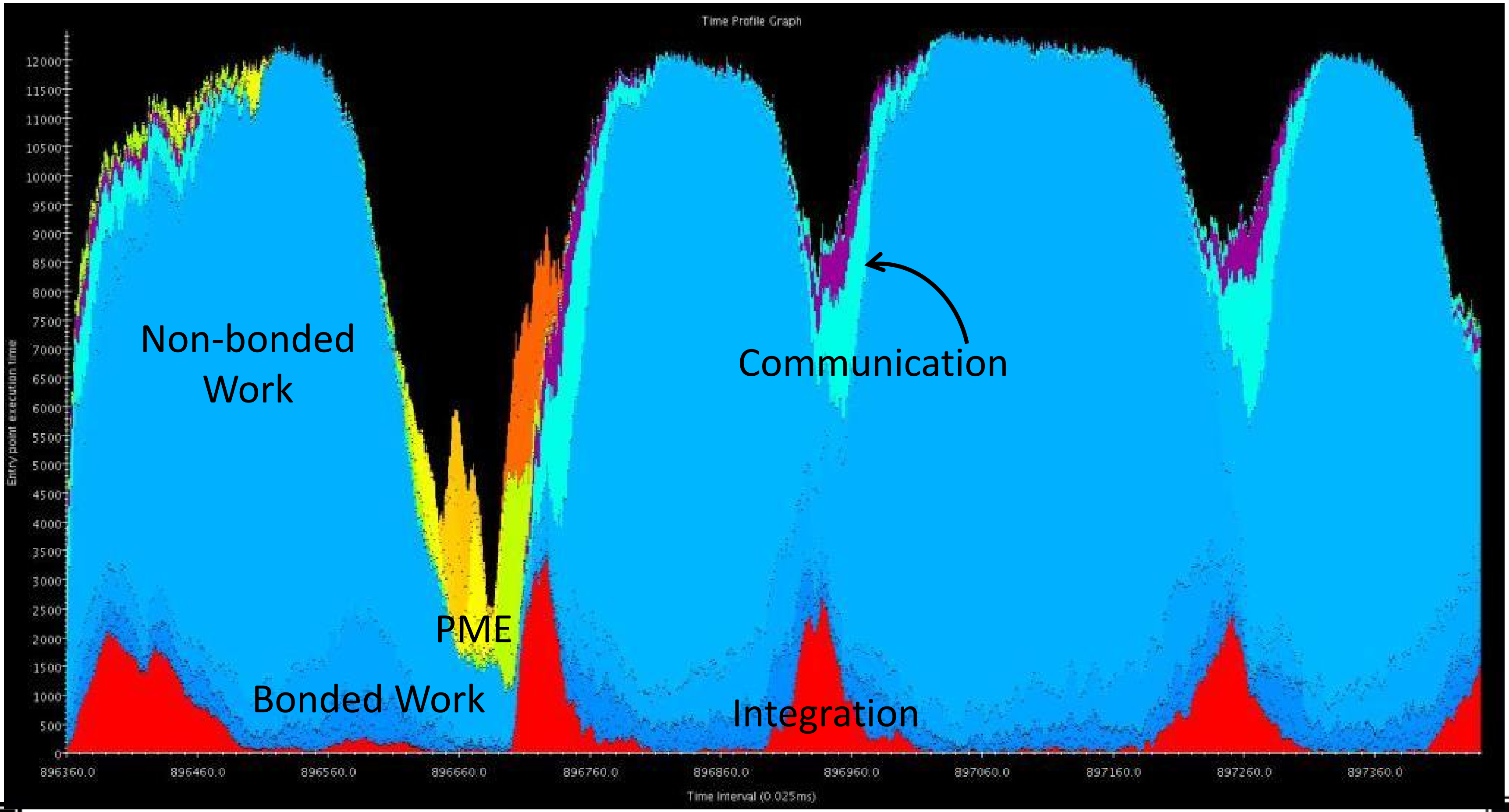



Patch Integration
Multicast
Point to Point
PME
Bonded Computes
Non–bonded Computes
Reductions
Point to Point
Patch Integration

What makes NAMD efficient ?

- Charm++ runtime support
  - Asynchronous message-driven model
  - Adaptive overlap of communication and computation


Patch Integration
Multicast
Point to Point
Bonded Computes
Non-bonded Computes
PME
Reductions
Point to Point
Patch Integration

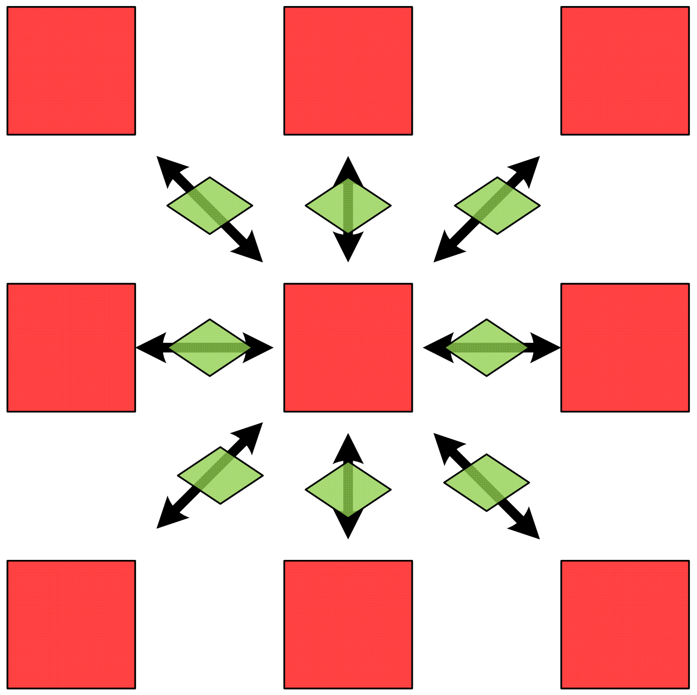

Time Profile Graph
Entry point execution time
Time Interval (0.025ms)
Non-bonded Work
Communication
PME
Bonded Work
Integration

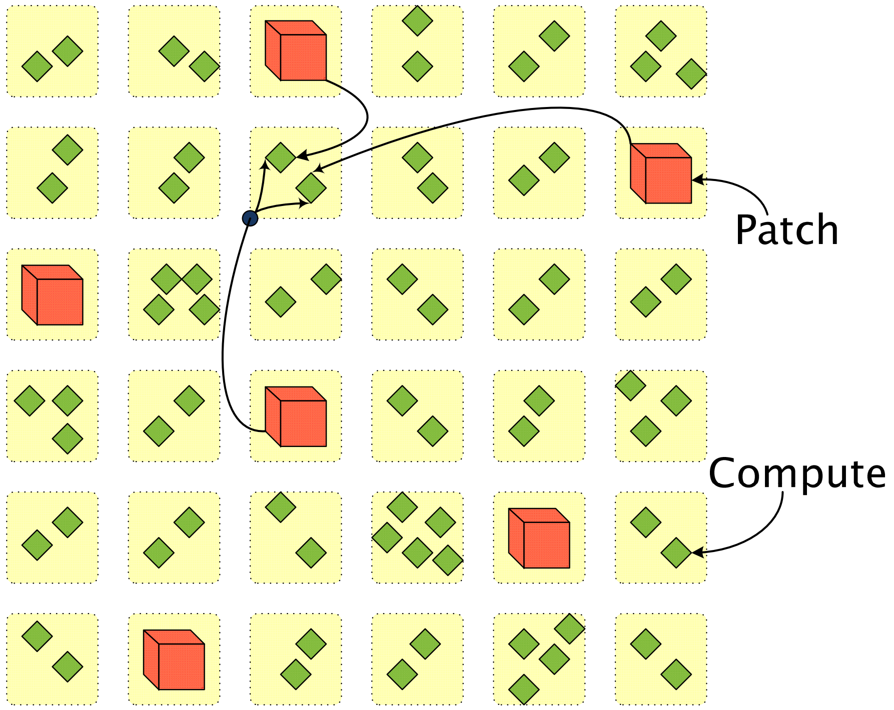
What makes NAMD efficient ?

- Charm++ runtime support
  - Asynchronous message-driven model
  - Adaptive overlap of communication and computation
- Load balancing support
  - Difficult problem: balancing heterogeneous computation
  - Measurement-based load balancing

What makes NAMD highly scalable ?

- Hybrid decomposition scheme
- Variants of this hybrid scheme used by Blue Matter and Desmond





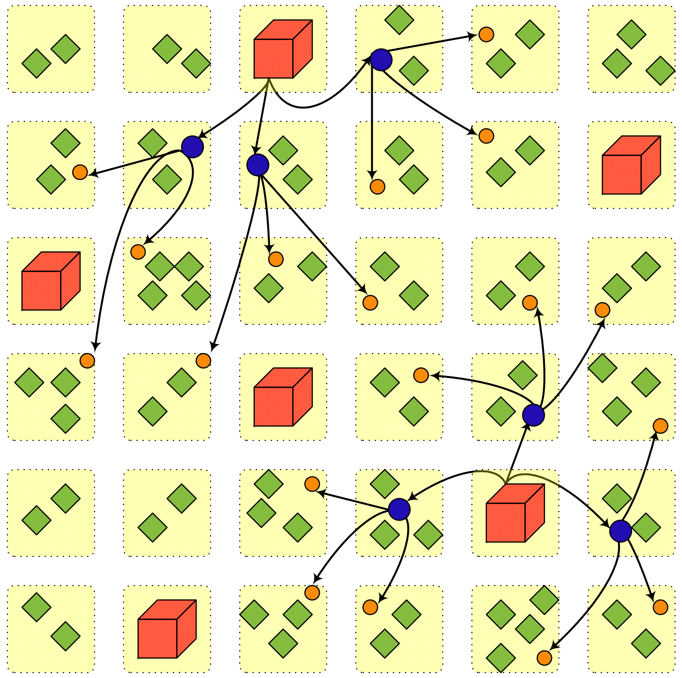
Scaling Challenges

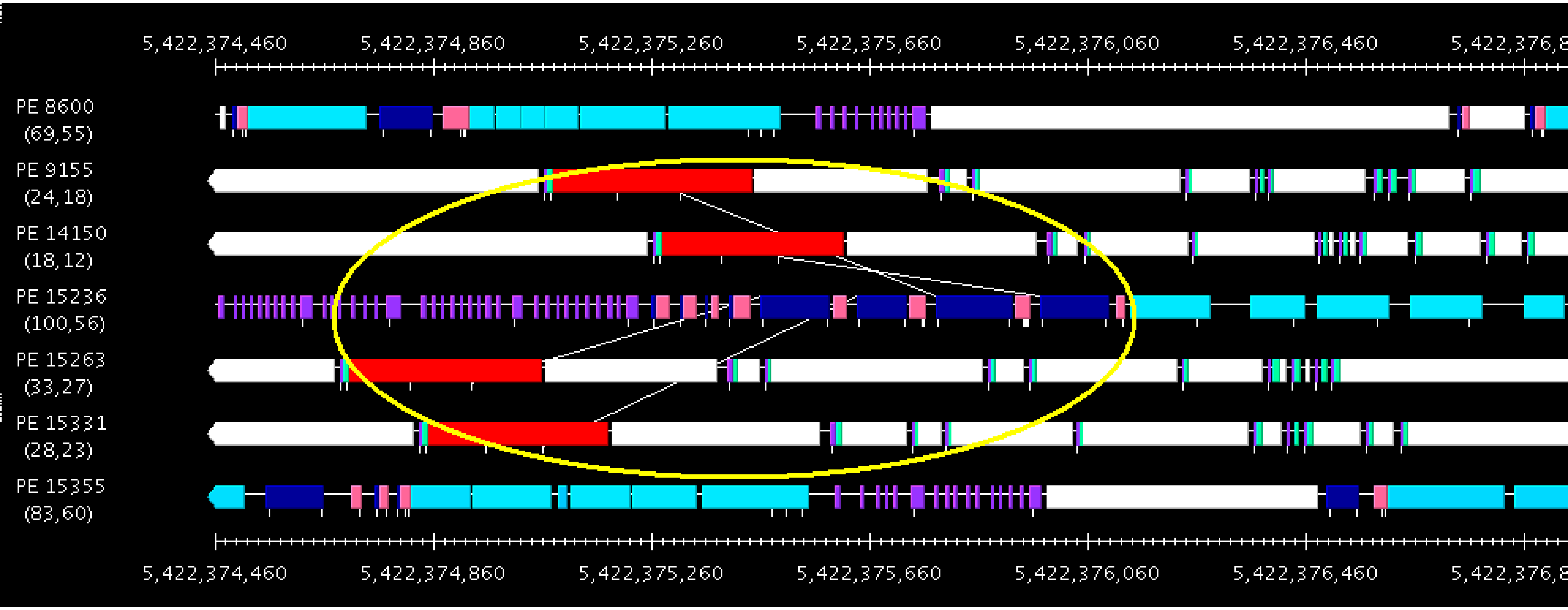
- Scaling a few thousand atom simulations to tens of thousands of processors
  - Interaction of adaptive runtime techniques
  - Optimizing the PME implementation
- Running multi-million atom simulations on machines with limited memory
  - Memory Optimizations

Conflicting Adaptive Runtime Techniques

- Patches multicast data to computes
- At load balancing step, computes re-assigned to processors

- Tree re-built after computes have migrated




5,422,374,460
5,422,374,860
5,422,375,260
5,422,375,660
5,422,376,060
5,422,376,460
5,422,376,8
PE 8600
(69,55)
PE 9155
(24,18)
PE 14150
(18,12)
PE 15236
(100,56)
PE 15263
(33,27)
PE 15331
(28,23)
PE 15355
(83,60)

- Solution
  - Persistent spanning trees
  - Centralized spanning tree creation
- Unifying the two techniques

PME Calculation

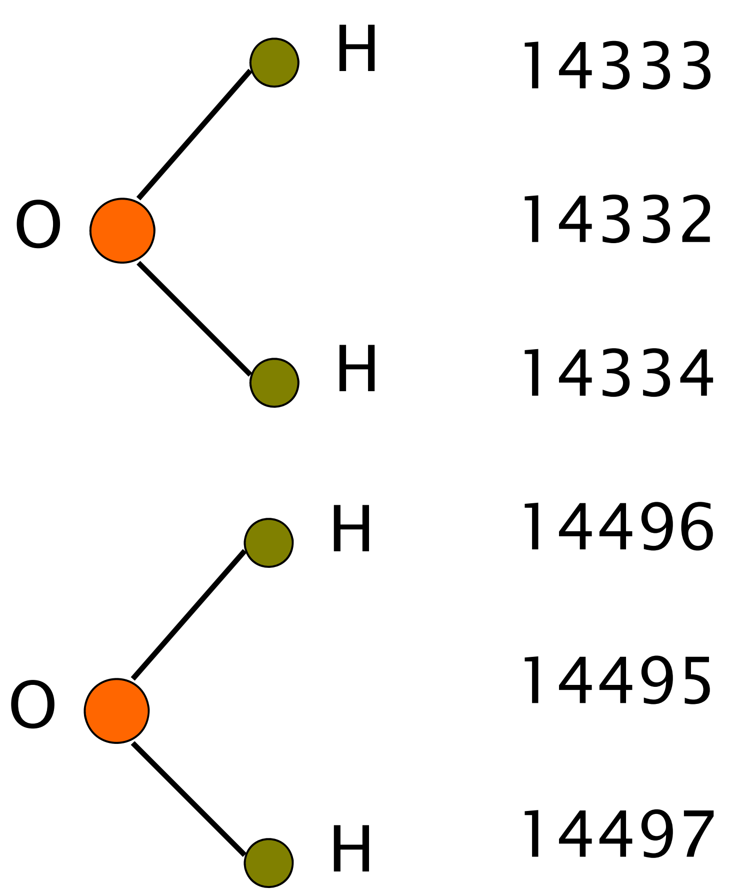
- Particle Mesh Ewald (PME) method used for long range interactions
  - 1D decomposition of the FFT grid
- PME is a small portion of the total computation
  - Better than the 2D decomposition for small number of processors
- On larger partitions
  - Use a 2D decomposition
  - More parallelism and better overlap

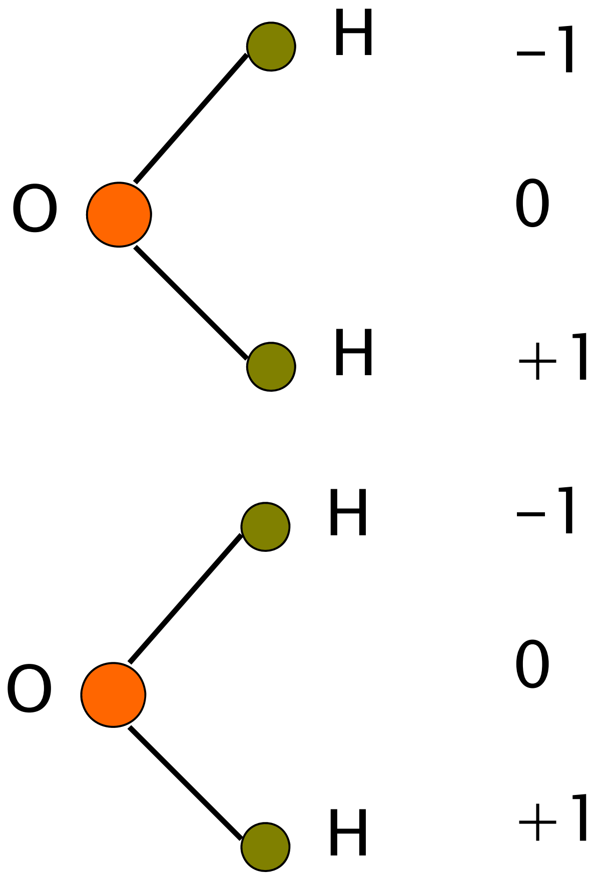
Automatic Runtime Decisions

- Use of 1D or 2D algorithm for PME
- Use of spanning trees for multicast
- Splitting of patches/computes for fine-grained parallelism
- Depend on:
  - Characteristics of the machine
  - No. of processors
  - No. of atoms in the simulation

Reducing the memory footprint

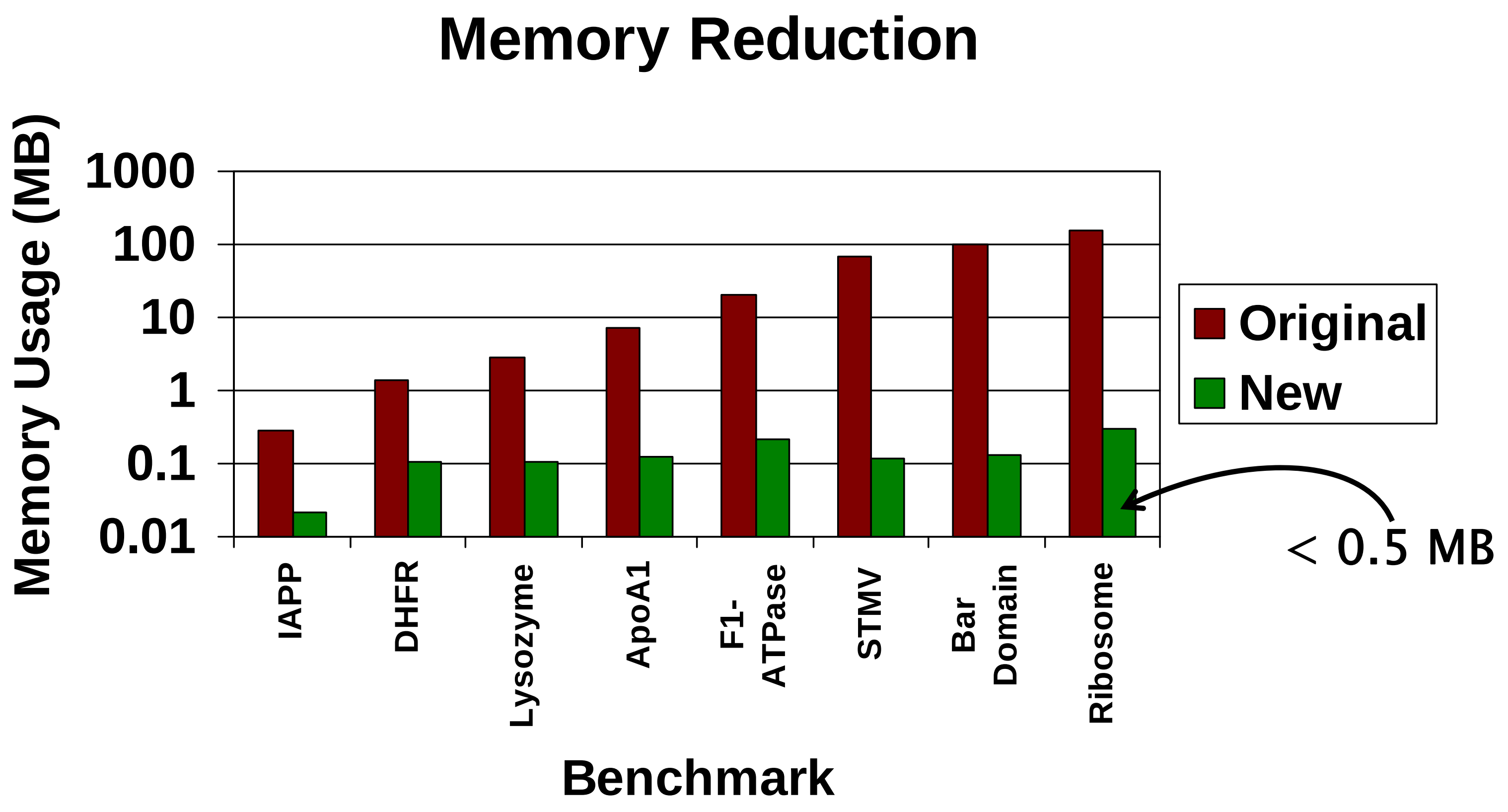
- Exploit the fact that building blocks for a bio-molecule have common structures
- Store information about a particular kind of atom only once


H
14333
O
14332
H
14334
H
14496
O
14495
H
14497

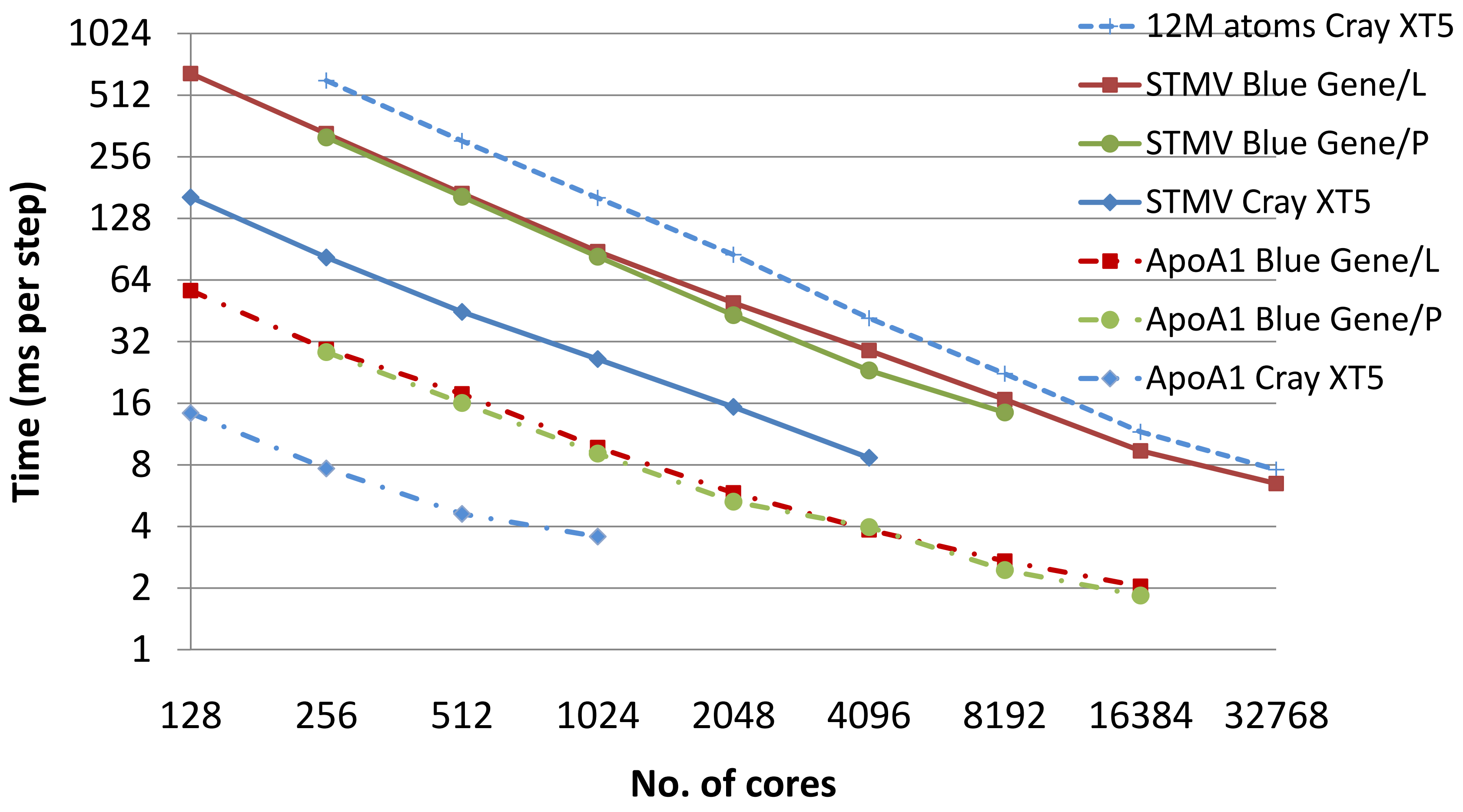

H
−1
O
0
H
+1
H
−1
O
0
H
+1

Reducing the memory footprint

- Exploit the fact that building blocks for a bio-molecule have common structures
- Store information about a particular kind of atom only once
- Static atom information increases only with the addition of unique proteins in the simulation
- Allows simulation of 2.8 M Ribosome on Blue Gene/L


Memory Reduction
Memory Usage (MB)
1000
100
10
1
0.1
0.01
IAPP
DHFR
Lysozyme
ApoA1
F1-ATPase
STMV
Bar Domain
Ribosome
Benchmark
Original
New
< 0.5 MB

NAMD's Performance



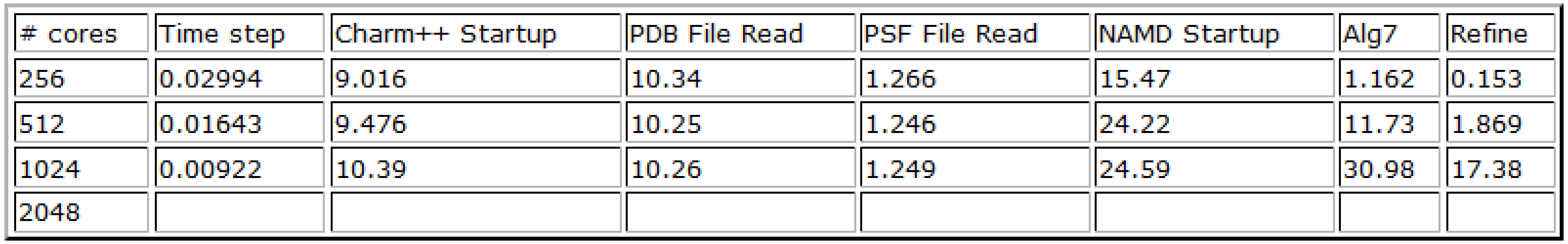
STMV: ~1 million atoms

ApoA1: ~92K atoms

More Challenges

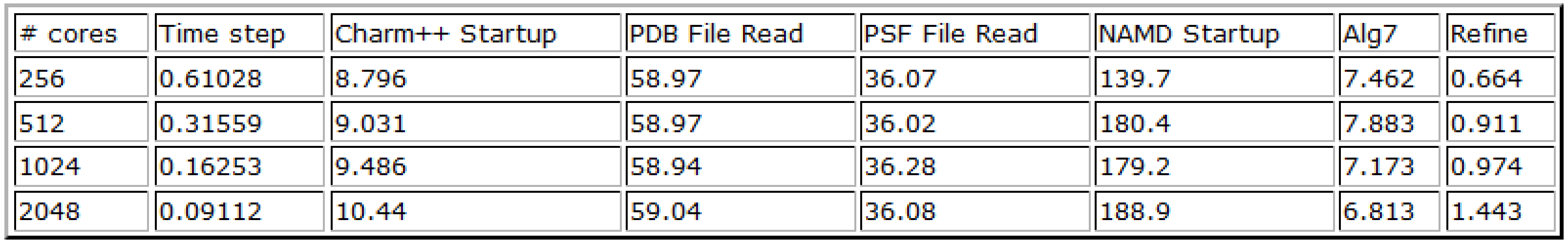
- Reducing start-up and load balancing time

ApoA1 (92,222 atoms)



# cores	Time step	Charm++ Startup	PDB File Read	PSF File Read	NAMD Startup	Alg7	Refine
256	0.02994	9.016	10.34	1.266	15.47	1.162	0.153
512	0.01643	9.476	10.25	1.246	24.22	11.73	1.869
1024	0.00922	10.39	10.26	1.249	24.59	30.98	17.38
2048							

Ribosome (2.8M atoms)



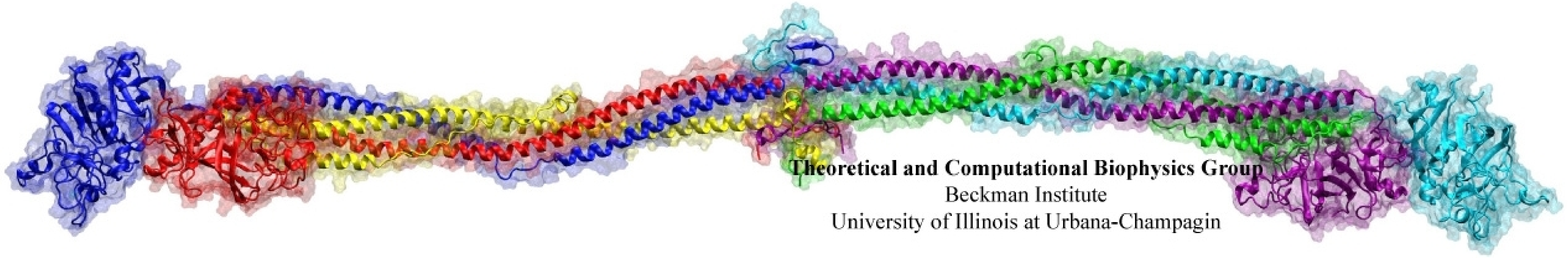
# cores	Time step	Charm++ Startup	PDB File Read	PSF File Read	NAMD Startup	Alg7	Refine
256	0.61028	8.796	58.97	36.07	139.7	7.462	0.664
512	0.31559	9.031	58.97	36.02	180.4	7.883	0.911
1024	0.16253	9.486	58.94	36.28	179.2	7.173	0.974
2048	0.09112	10.44	59.04	36.08	188.9	6.813	1.443

Recent Work

- Running NAMD on Petascale machines
  - Improved hierarchical load balancers
  - Parallel Input/Output
- Reducing communication overhead with increasing fine-grained parallelism

Summary

- NAMD is a highly scalable and portable MD program
  - Runs on a variety of architectures
  - Available free of cost on machines at most supercomputing centers
  - Supports a range of sizes of molecular systems
- Uses adaptive runtime techniques for high scalability
- Automatic selection of algorithms at runtime best suited for the scenario
- With new optimizations, NAMD is ready for the next generation of parallel machines


Theoretical and Computational Biophysics Group
Beckman Institute
University of Illinois at Urbana-Champagin

Questions ?

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

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