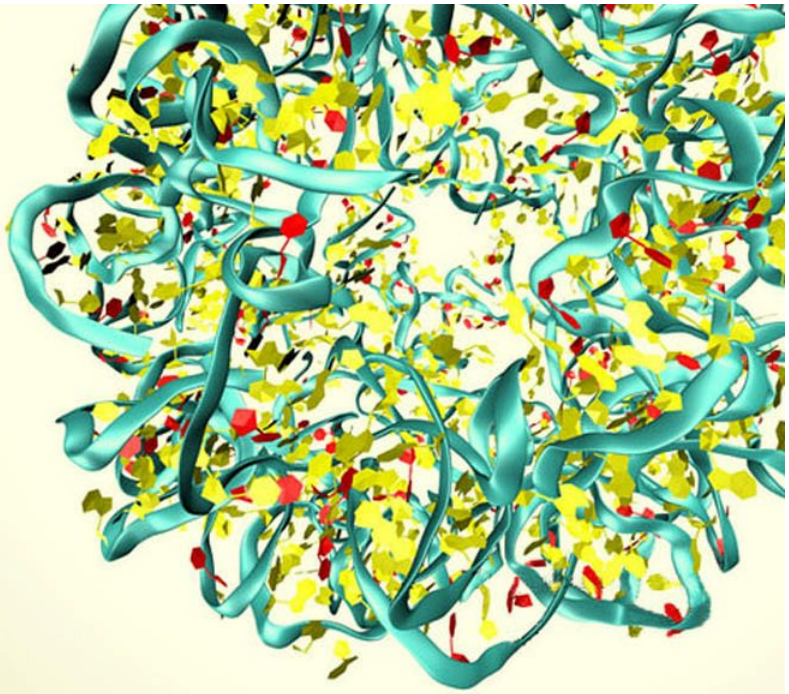
NAMD Team:
Abhinav Bhatele
Eric Bohm
Sameer Kumar
David Kunzman
Chao Mei
Chee Wai Lee
Kumaresh P.
James Phillips
Gengbin Zheng
Laxmikant Kale
Klaus Schulten

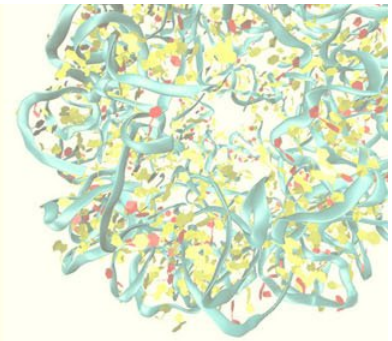


Scaling Challenges in NAMD: Past and Future

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

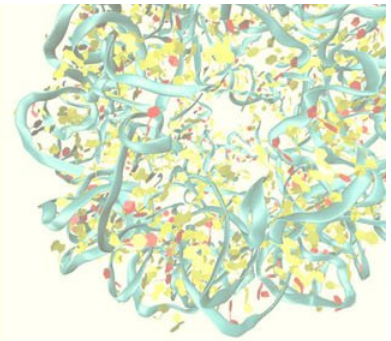
PARALLEL PROGRAMMING LAB
DEPT. OF COMPUTER SCIENCE, UNIVERSITY OF ILLINOIS
PPL UIUC

Outline



- NAMD: An Introduction
- Past Scaling Challenges
  - Conflicting Adaptive Runtime Techniques
  - PME Computation
  - Memory Requirements
- Performance Results
- Comparison with other MD codes
- Future Challenges:
  - Load Balancing
  - Parallel I/O
  - Fine-grained Parallelization

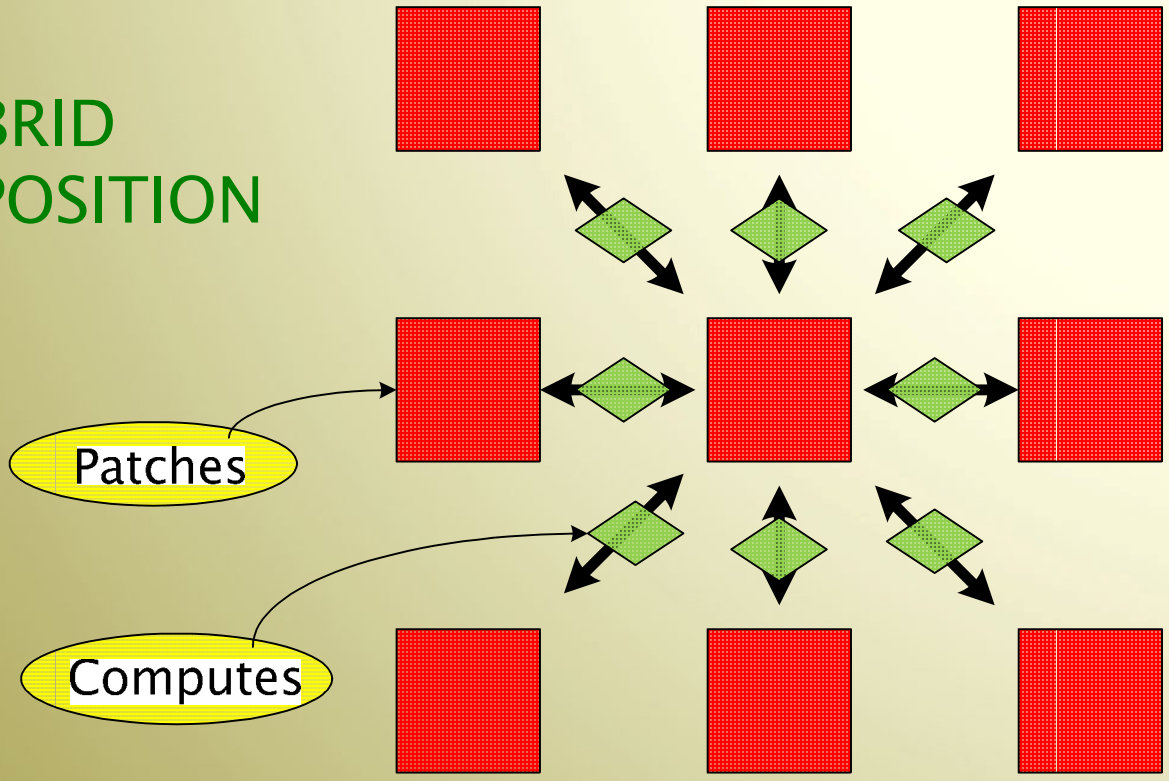
What is NAMD ?

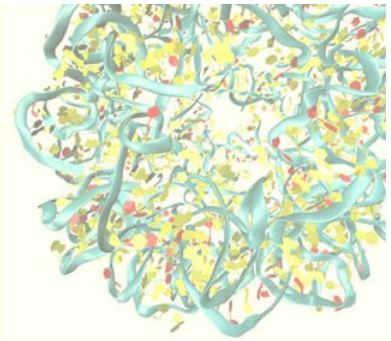


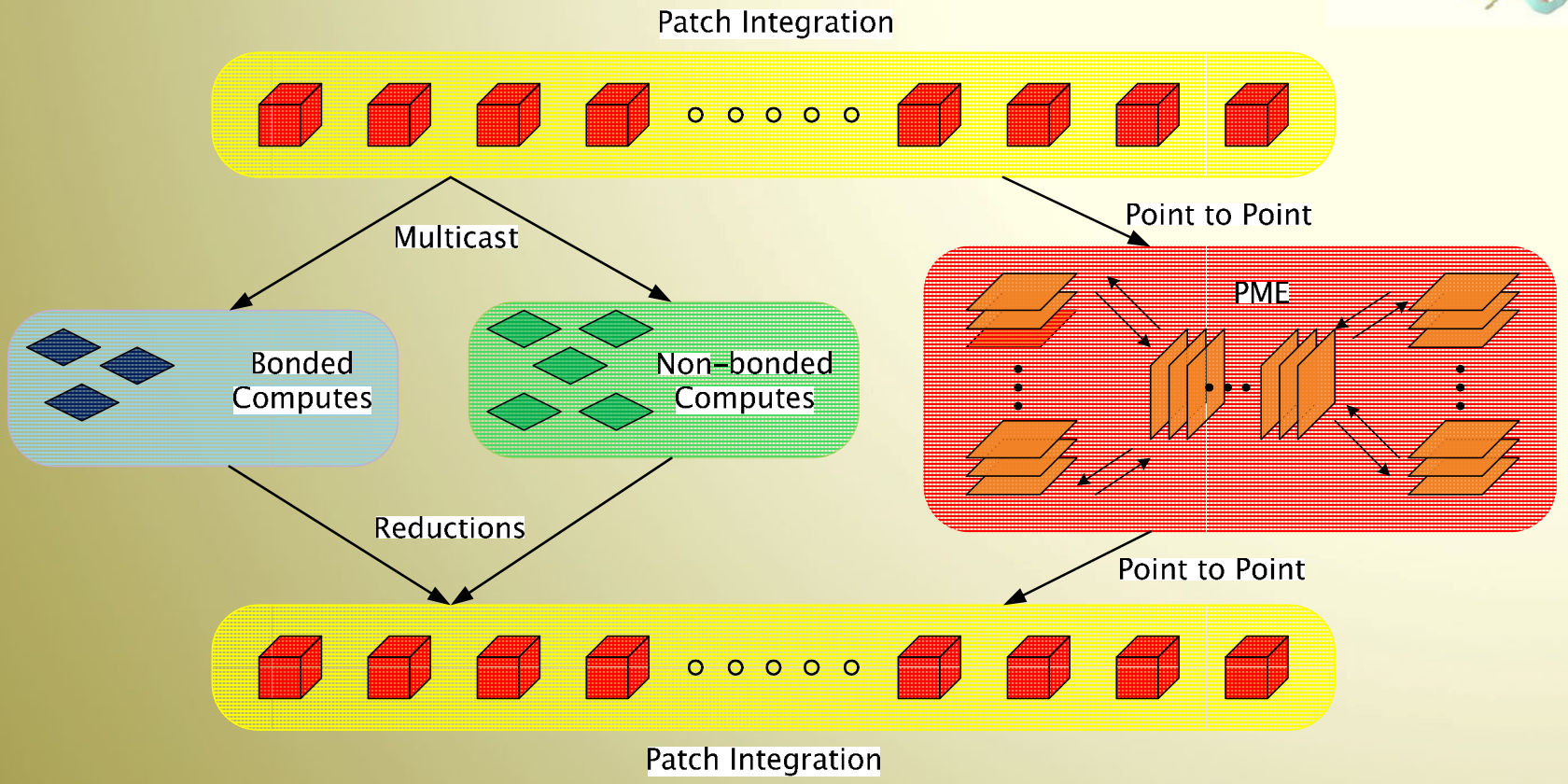
- A parallel molecular dynamics application
- Simulate the life of a bio-molecule
- How is the simulation performed ?
  - Simulation window broken down into a large number of time steps (typically 1 fs each)
  - Forces on every atom calculated every time step
  - Velocities and positions updated and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

How is NAMD parallelized ?

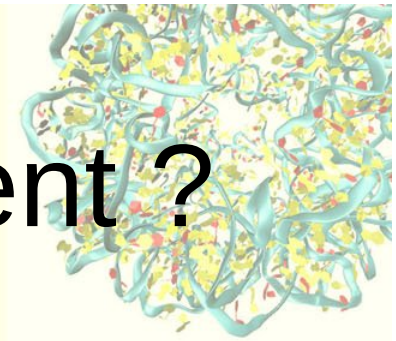
HYBRID DECOMPOSITION



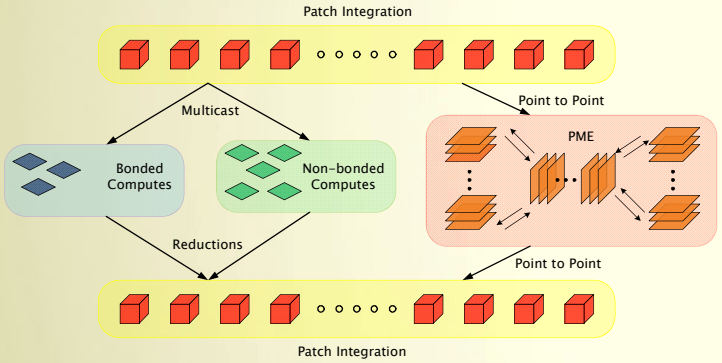


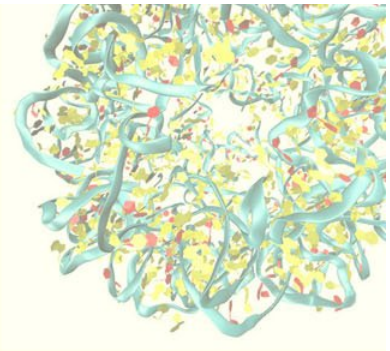

Patch Integration
Multicast
Point to Point
Bonded Computes
Non-bonded Computes
PME
Reductions
Point to Point
Patch Integration

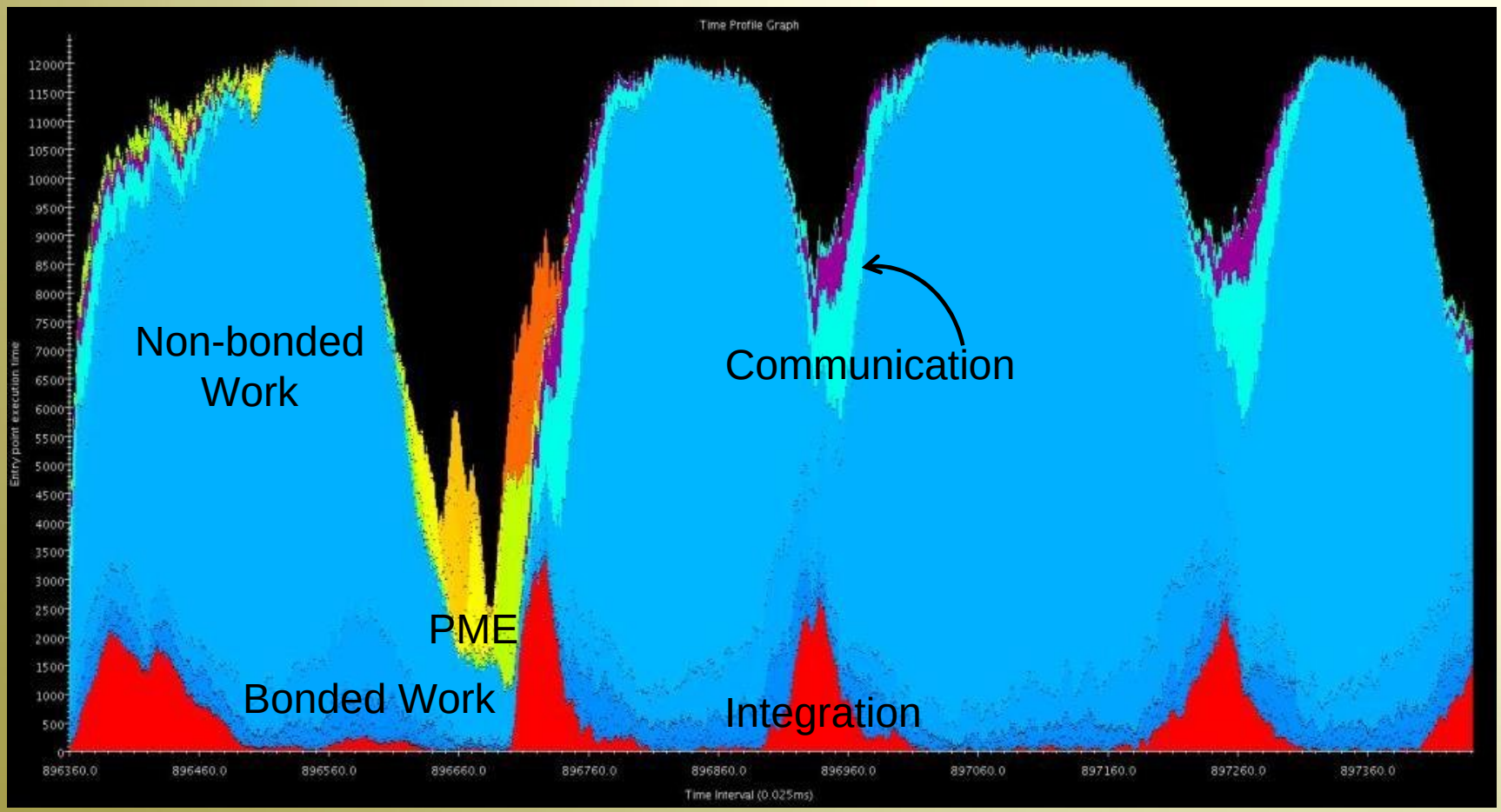
What makes NAMD efficient ?



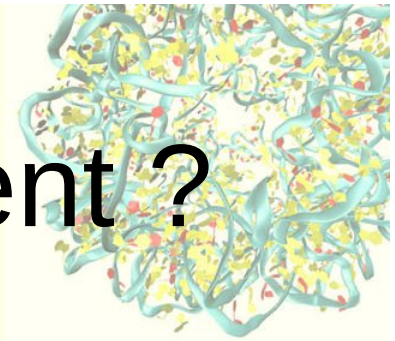
- Charm++ runtime support
  - Asynchronous message-driven model
  - Adaptive overlap of communication and computation


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



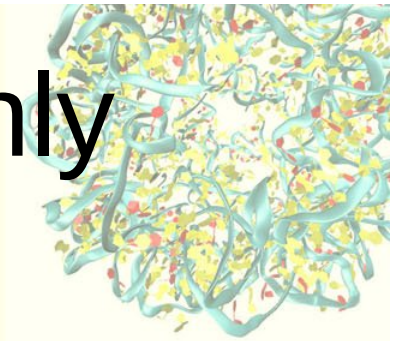

Time Profile Graph
Entry point execution time
Time Interval (0.025ms)
Non-bonded Work
Communication
PME
Bonded Work
Integration

What makes NAMD efficient ?

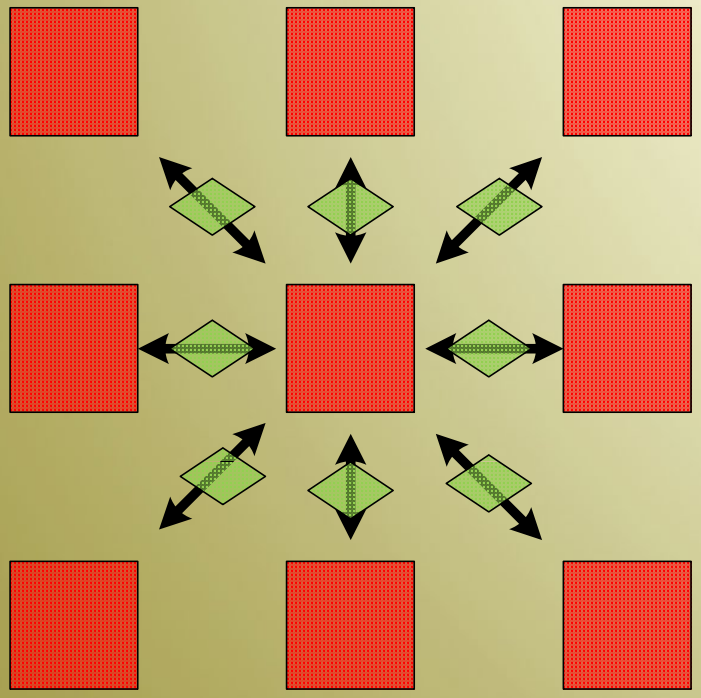


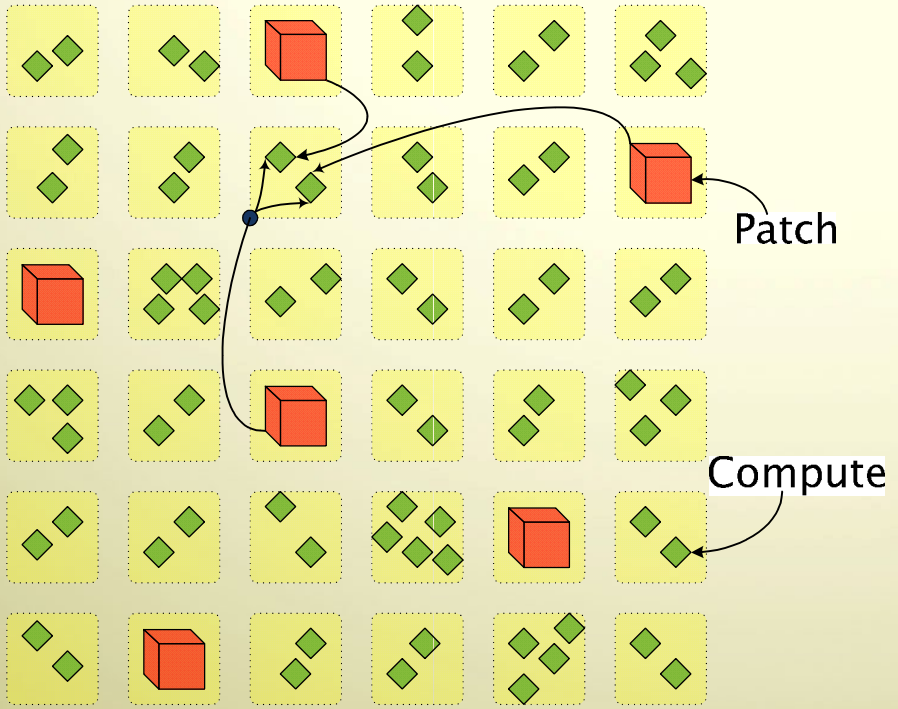
- Charm++ runtime support
  - Asynchronous message-driven model
  - Adaptive overlap of communication and computation
- Load balancing support
  - Difficult problem: balancing heterogeneous computation
  - Measurement-based load balancing

What makes NAMD highly scalable ?

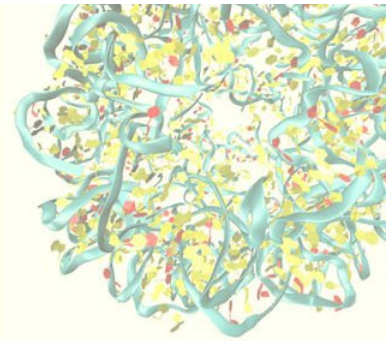


- Hybrid decomposition scheme
- Variants of this hybrid scheme used by Blue Matter and Desmond



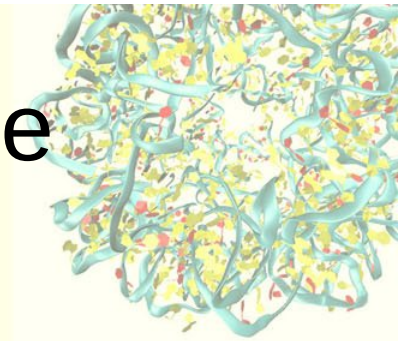


Scaling Challenges

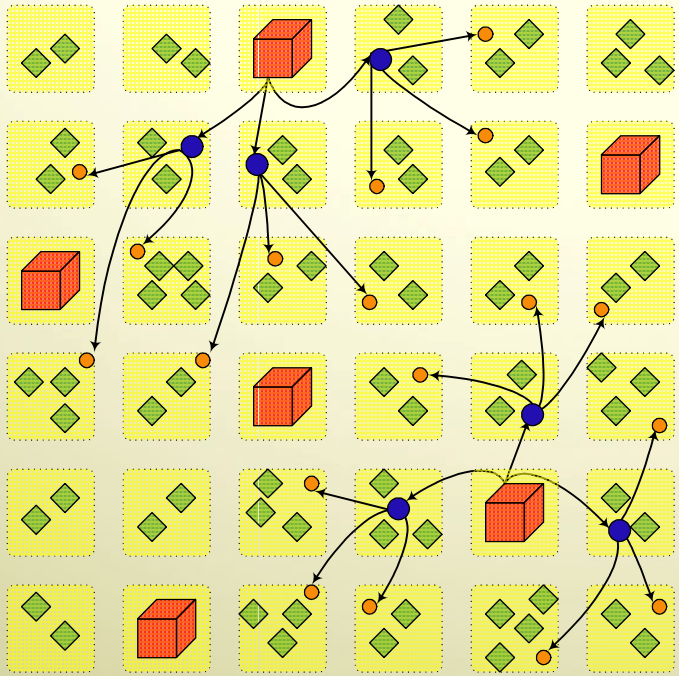


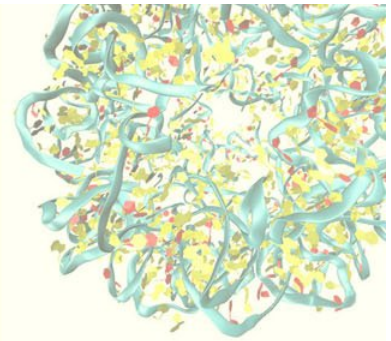
- Scaling a few thousand atom simulations to tens of thousands of processors
  - Interaction of adaptive runtime techniques
  - Optimizing the PME implementation
- Running multi-million atom simulations on machines with limited memory
  - Memory Optimizations

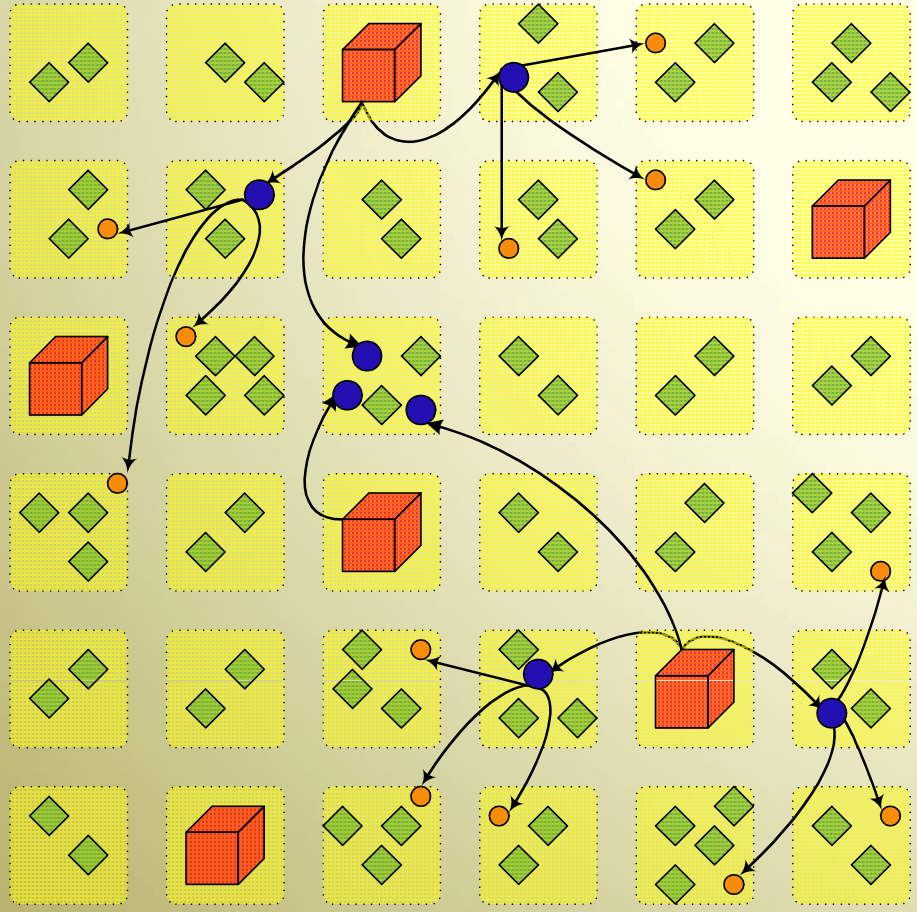
Conflicting Adaptive Runtime Techniques

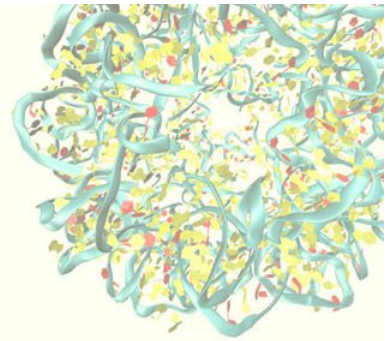


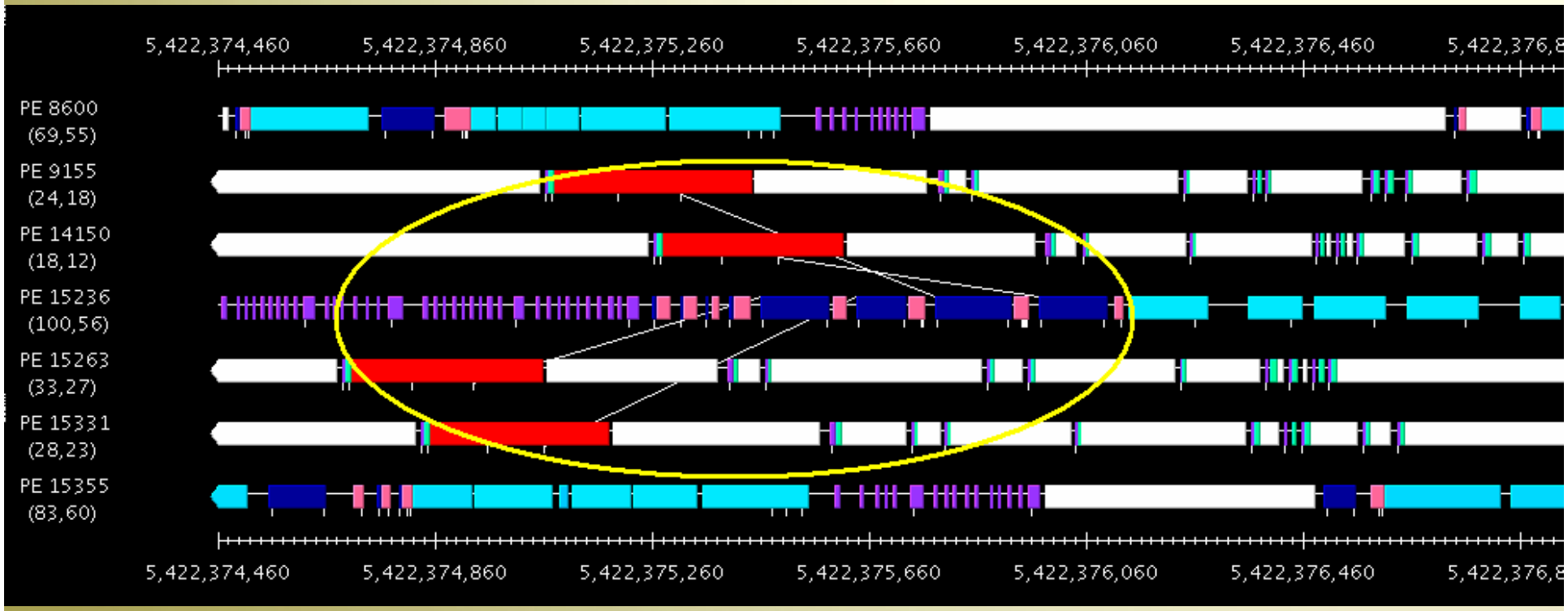
- Patches multicast data to computes
- At load balancing step, computes re-assigned to processors
- Tree re-built after computes have migrated

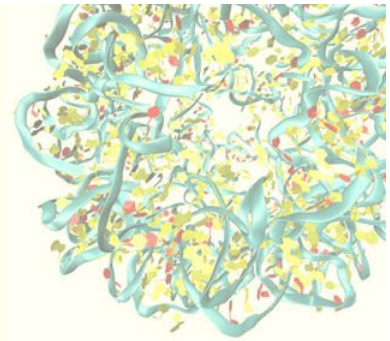






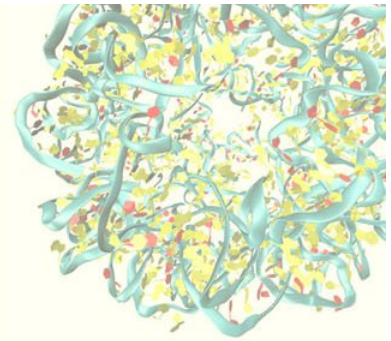



5,422,374,460
5,422,374,860
5,422,375,260
5,422,375,660
5,422,376,060
5,422,376,460
PE 8600 (69,55)
PE 9155 (24,18)
PE 14150 (18,12)
PE 15236 (100,56)
PE 15263 (33,27)
PE 15331 (28,23)
PE 15355 (83,60)



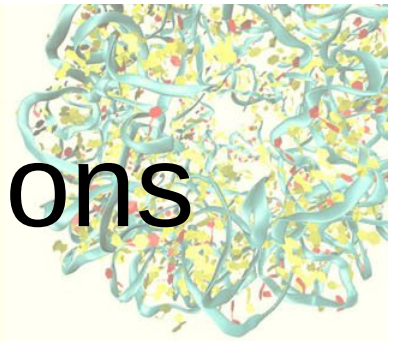
- Solution
  - Persistent spanning trees
  - Centralized spanning tree creation
- Unifying the two techniques

PME Calculation



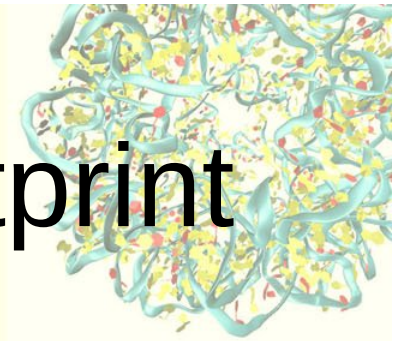
- Particle Mesh Ewald (PME) method used for long range interactions
  - 1D decomposition of the FFT grid
- PME is a small portion of the total computation
  - Better than the 2D decomposition for small number of processors
- On larger partitions
  - Use a 2D decomposition
  - More parallelism and better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

Automatic Runtime Decisions

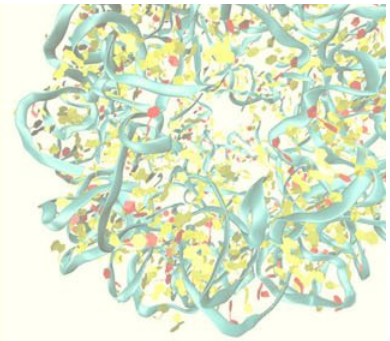


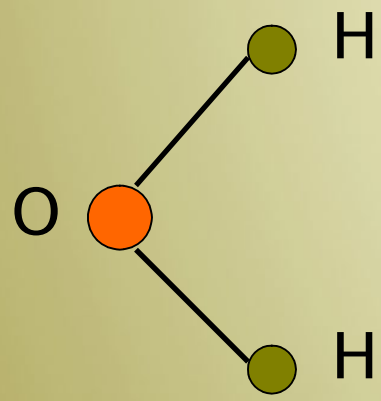
- Use of 1D or 2D algorithm for PME
- Use of spanning trees for multicast
- Splitting of patches for fine-grained parallelism
- Depend on:
  - Characteristics of the machine
  - No. of processors
  - No. of atoms in the simulation

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

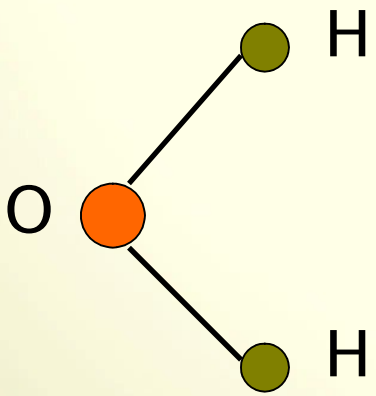




14333

14332

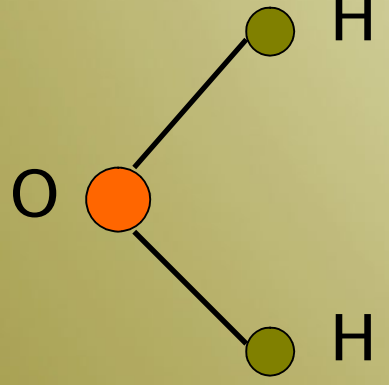
14334



-1

0

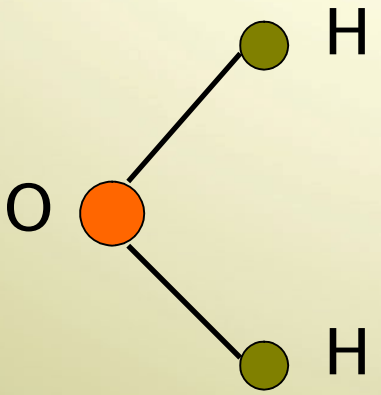
+1



14496

14495

14497

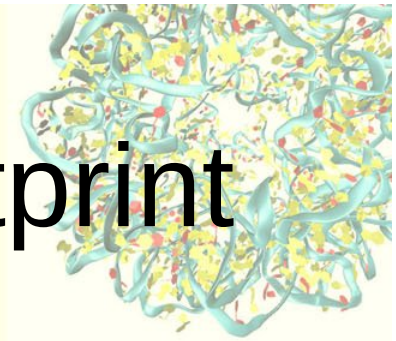


-1

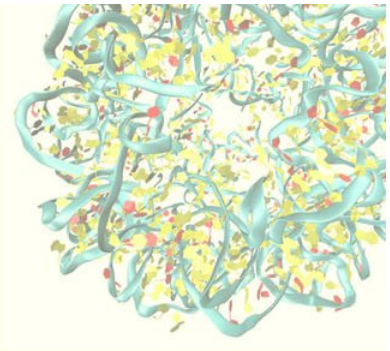
0

+1

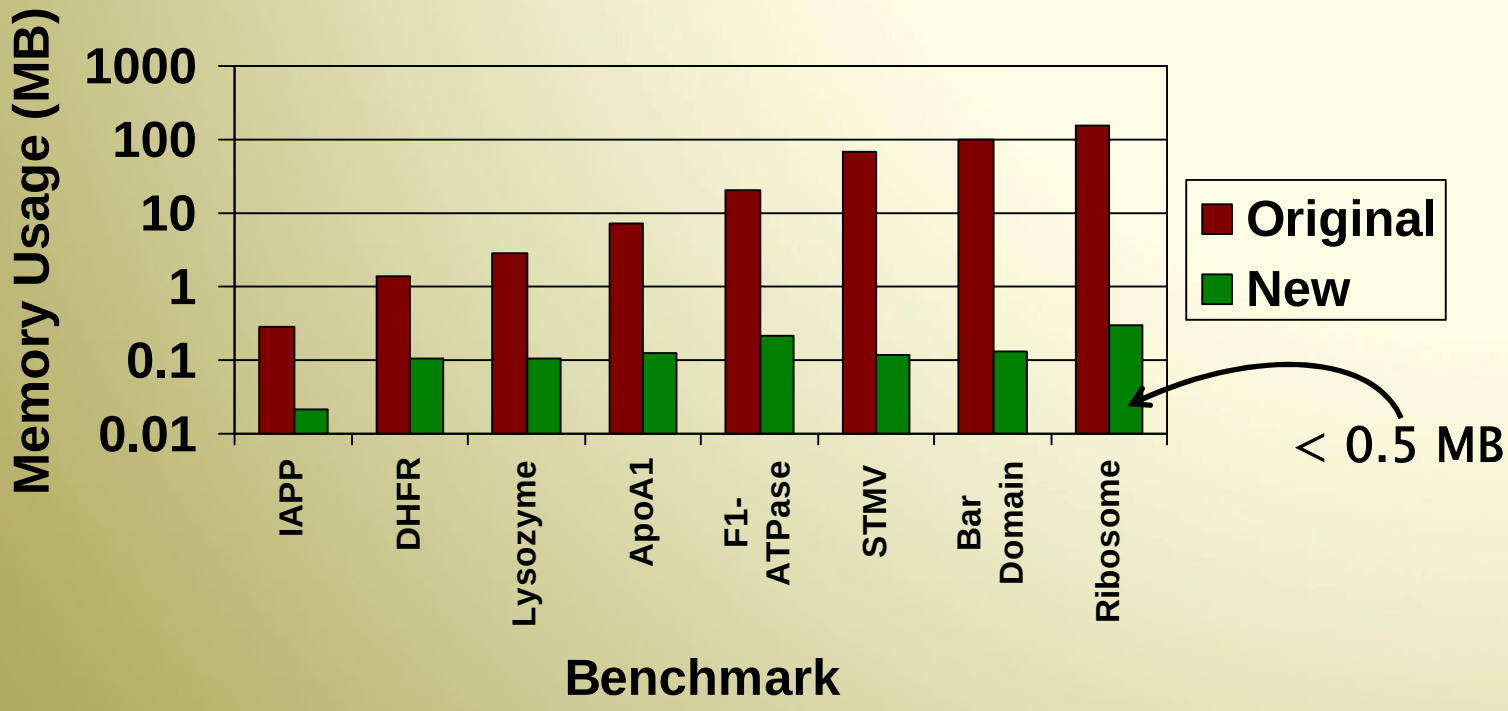
Reducing the memory footprint



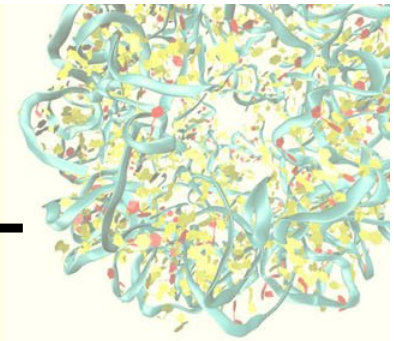
- Exploit the fact that building blocks for a bio-molecule have common structures
- Store information about a particular kind of atom only once
- Static atom information increases only with the addition of unique proteins in the simulation
- Allows simulation of 2.8 M Ribosome on Blue Gene/L

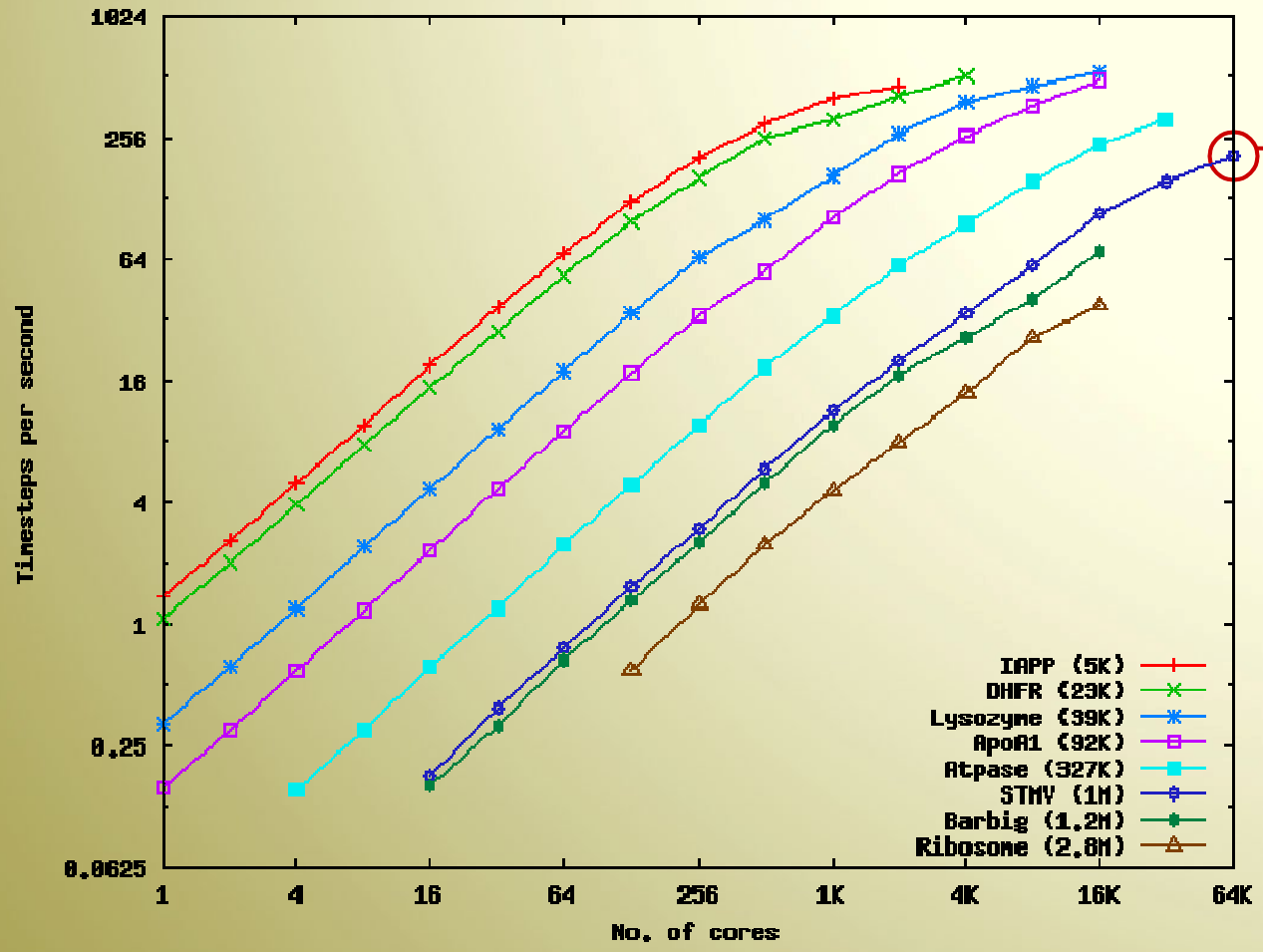


Memory Reduction



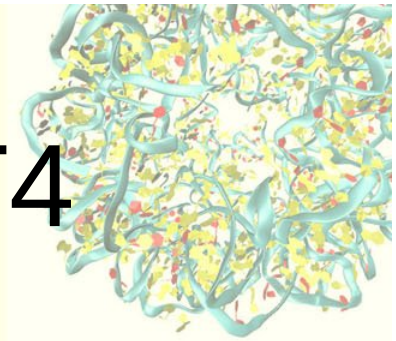
NAMD on Blue Gene/L

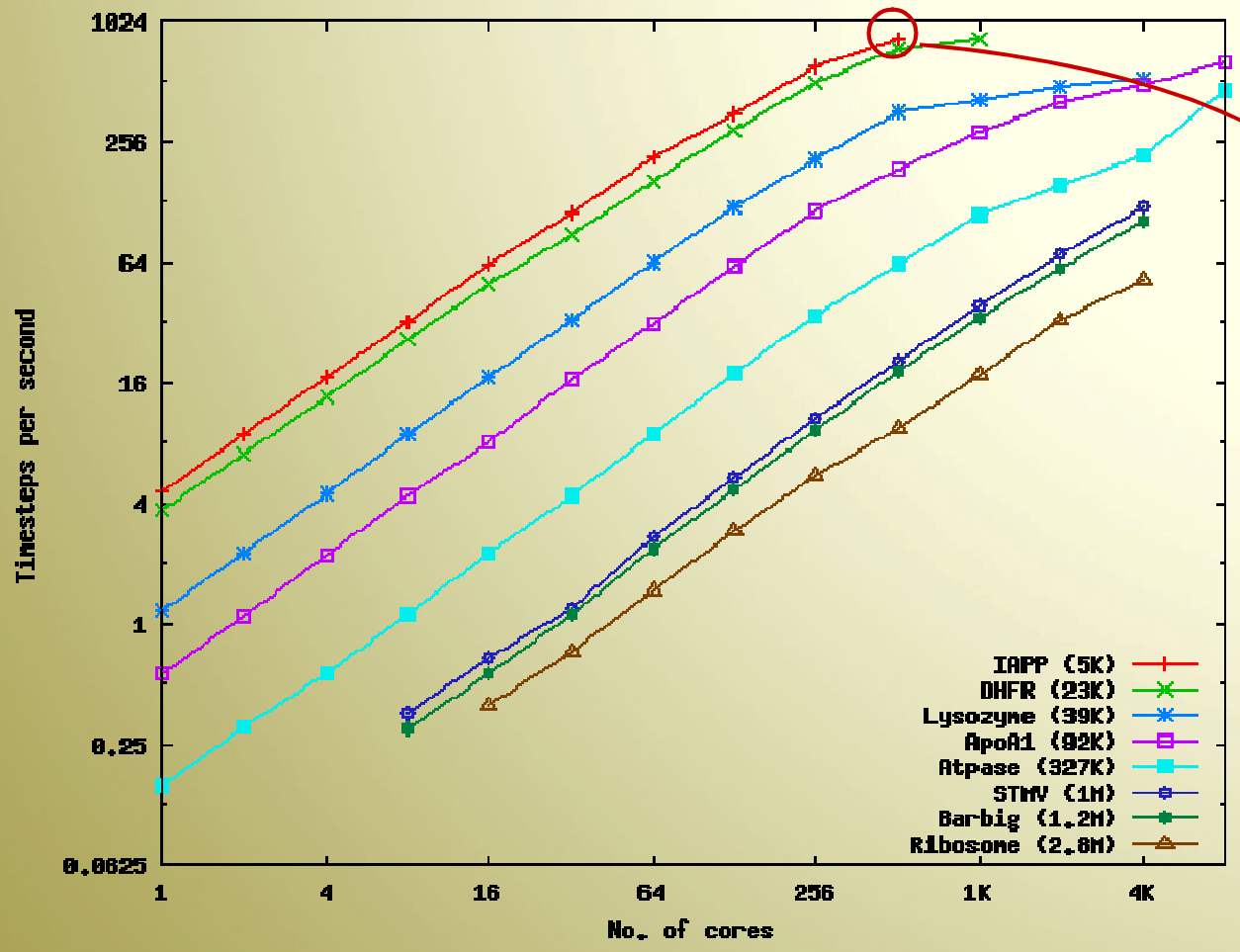




1 million atom simulation on 64K processors (LLNL BG/L)

NAMD on Cray XT3/XT4

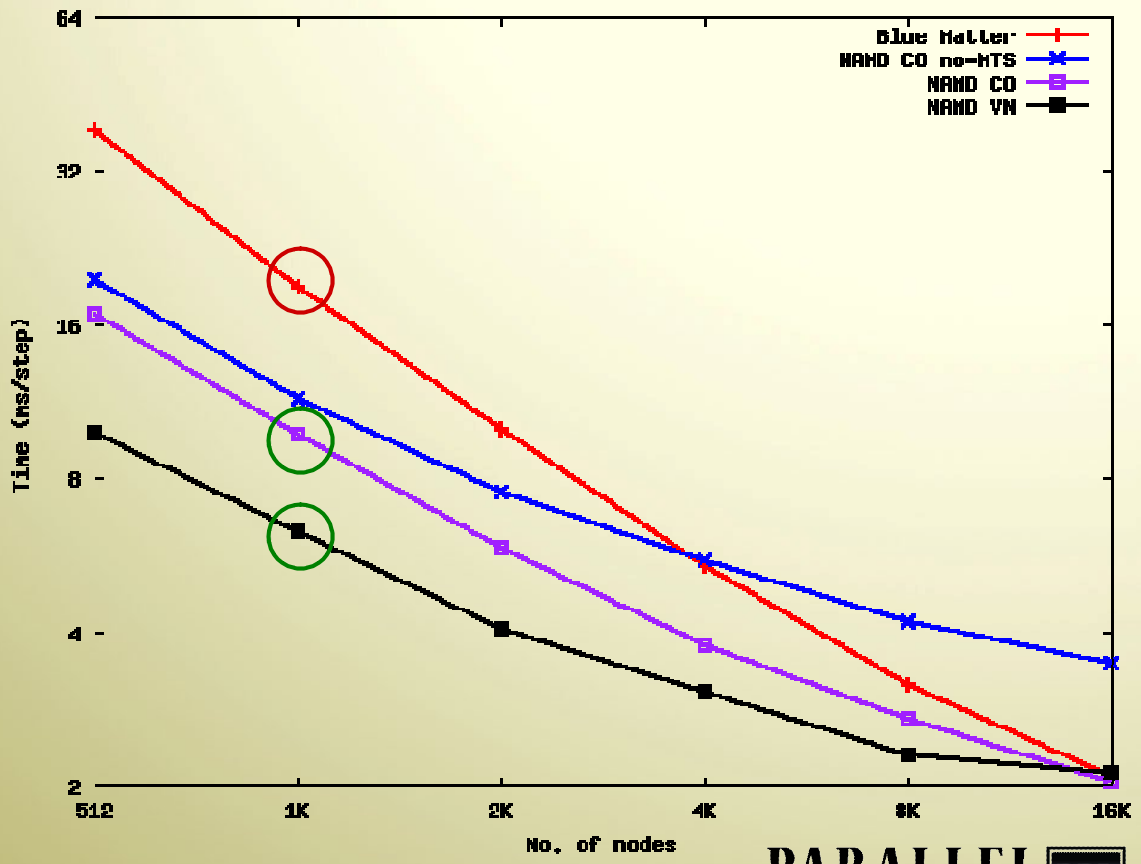




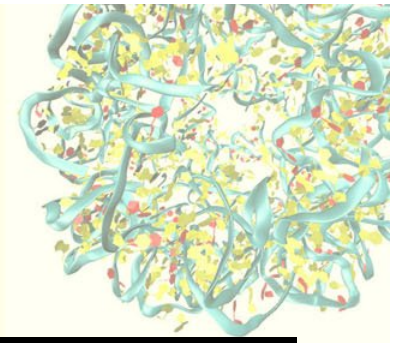
5570 atom simulation on 512 processors at 1.2 ms/step

Comparison with Blue Matter

- Blue Matter developed specifically for Blue Gene/L
- NAMD running on 4K cores of XT3 is comparable to BM running on 32K cores of BG/L

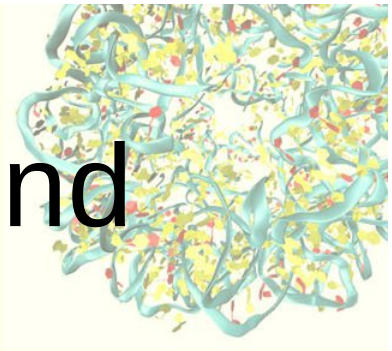


Time for ApoA1 (ms/step)

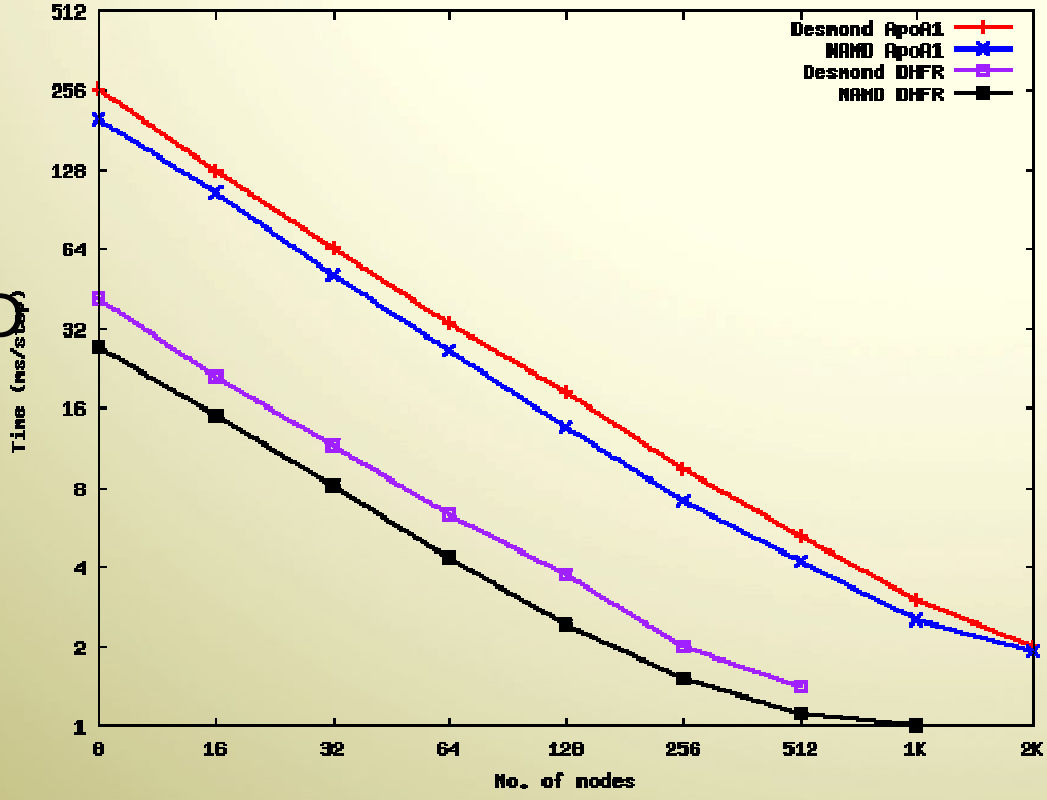


Number of Nodes	512	1024	2048	4096	8192	16384
Blue Matter (2 pes/node)	38.42	18.95	9.97	5.39	3.14	2.09
NAMD CO mode (1 pe/node)	16.83	9.73	5.8	3.78	2.71	2.04
NAMD VN mode (2 pes/node)	9.82	6.26	4.06	3.06	2.29	2.11
NAMD CO mode (No MTS)	19.59	11.42	7.48	5.52	4.2	3.46
NAMD VN mode (No MTS)	11.99	9.99	5.62	5.3	3.7	-

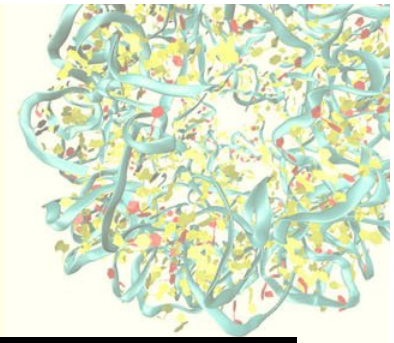
Comparison with Desmond



- Desmond is a proprietary MD program
- Uses single precision and exploits SSE instructions
- Low-level infiniband primitives tuned for MD

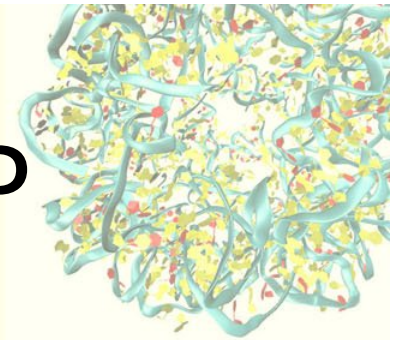


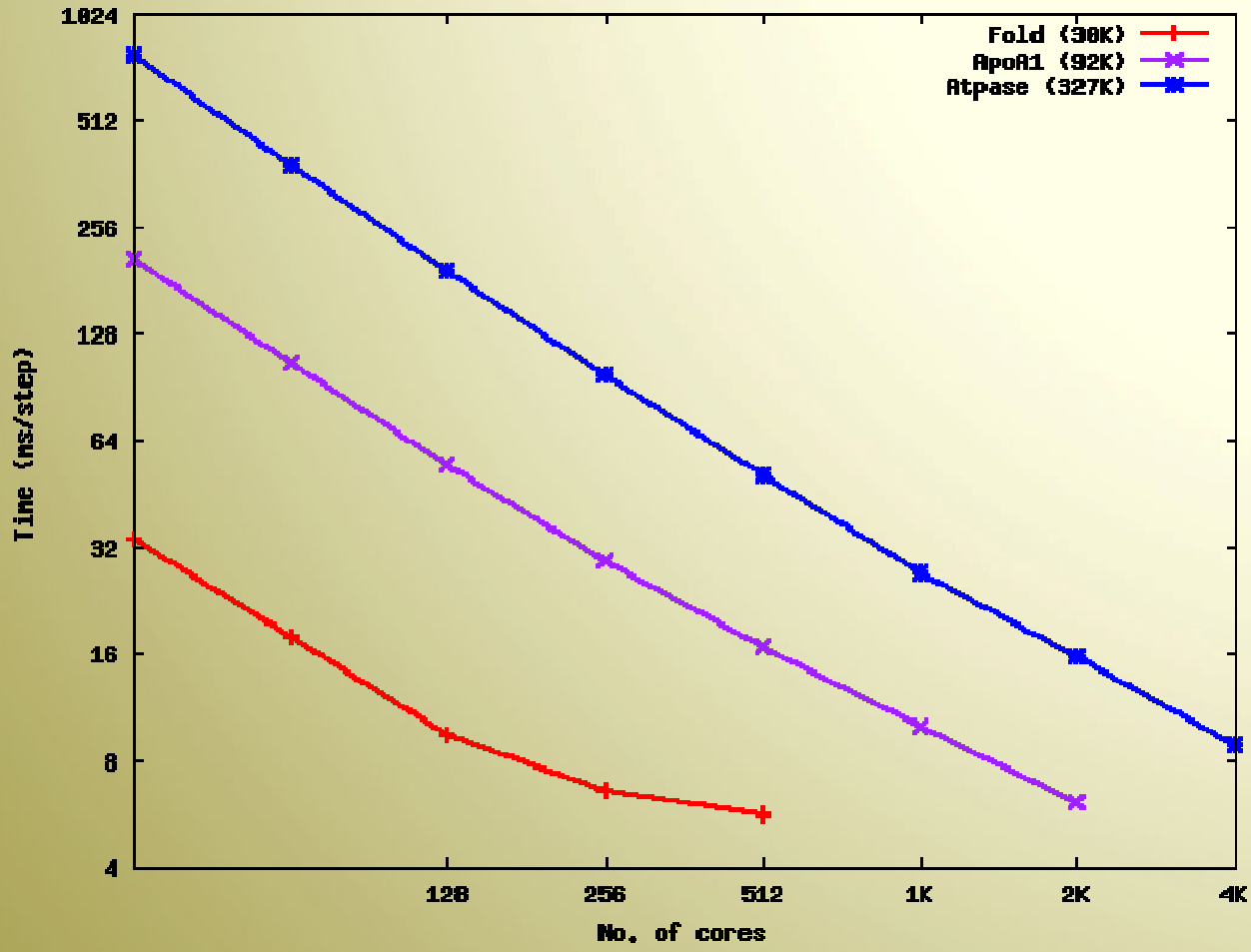
Time (ms/step) for Desmond on 2.4 GHz Opterons and NAMD on 2.6 GHz Xeons



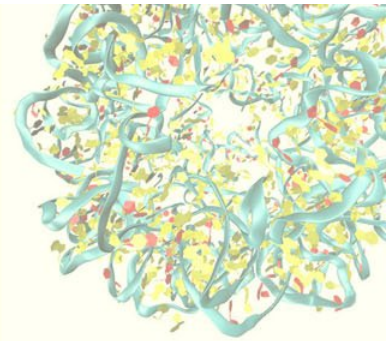
Number of Cores	8	16	32	64	128	256	512	1024	2048
Desmond ApoA1	256.8	126.8	64.3	33.5	18.2	9.4	5.2	3.0	2.0
NAMD ApoA1	199.3	104.9	50.7	26.5	13.4	7.1	4.2	2.5	1.9
Desmond DHFR	41.4	21.0	11.5	6.3	3.7	2.0	1.4	-	-
NAMD DHFR	27.3	14.9	8.09	4.3	2.4	1.5	1.1	1.0	

NAMD on Blue Gene/P



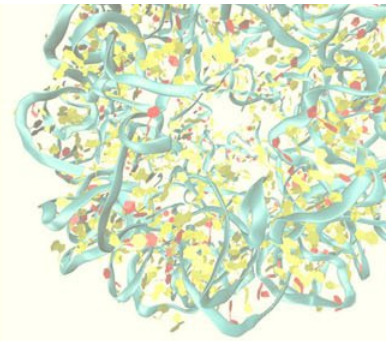


Future Work

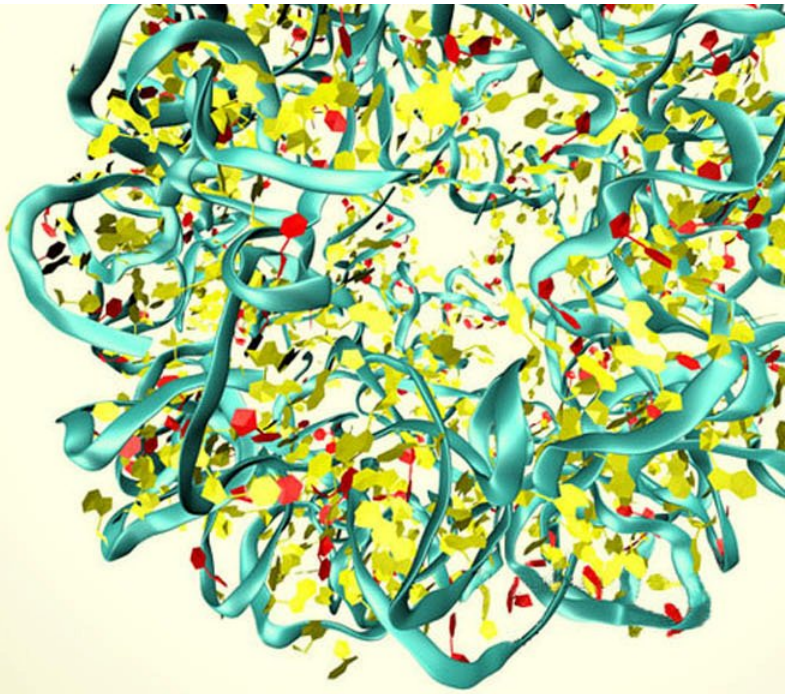


- Optimizing PME computation
  - Use of one-sided puts between FFTs
- Reducing communication and other overheads with increasing fine-grained parallelism
- Running NAMD on Blue Waters
  - Improved distributed load balancers
  - Parallel Input/Output

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



Questions ?