NAMD at Extreme Scale

Presented by: Eric Bohm
Team: Eric Bohm, Chao Mei, Osman Sarood, David Kunzman, Yanhua, Sun, Jim Phillips, John Stone, LV Kale

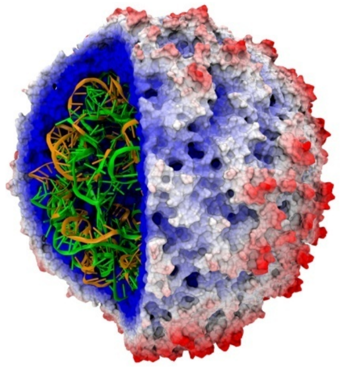
Overview

- NAMD description
- Power7 Tuning
- Support for Large Molecular Systems
- Petascale Tuning
- Torrent Network optimizations
- Exascale Feasibility
- Summary/Future work

NAMD Serving NIH Users and Goals

Practical Supercomputing for Biomedical Research

- 40,000 users can't all be computer experts.
  - 18% are NIH-funded; many in other countries.
  - 10,000 have downloaded more than one version.
  - 1700 citations of NAMD reference papers.
- One program for all platforms.
  - Desktops and laptops – setup and testing
  - Linux clusters – affordable local workhorses
  - Supercomputers – free allocations on TeraGrid
  - Blue Waters – sustained petaflop/s performance
  - GPUs - next-generation supercomputing
- User knowledge is preserved.
  - No change in input or output files.
  - Run any simulation on **any number of cores.**
- Available free of charge to all.



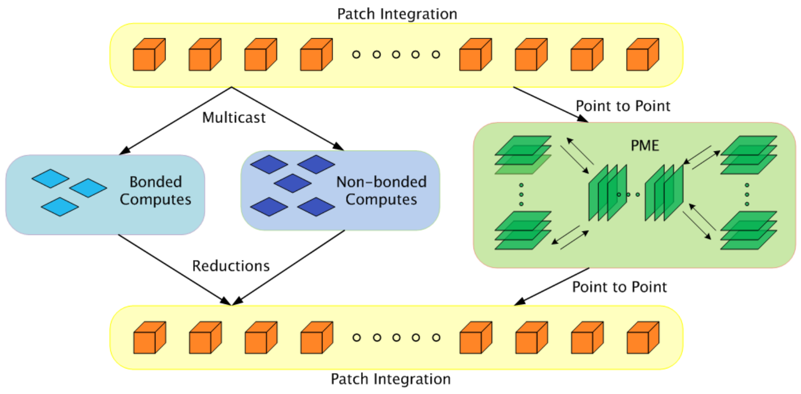
Phillips *et al*., *J. Comp. Chem.* 26:1781-1802, 2005.

NSF/NCSA Blue Waters Project

- Sustained Petaflops system funded by NSF to be ready in 2011.
  - System expected to exceed 300,000 processor cores.
- NSF Acceptance test: 100 million atom Bar Domain simulation using NAMD.
- NAMD PRAC The Computational Microscope
  - Systems from 10 to 100 million atoms
- A recently submitted PRAC from an independent group wishes to use NAMD
  - 1 Billion atoms!

NAMD Parallelization

- Molecular Dynamics simulation of biological systems
- Uses the Charm++ idea:
  - Decompose the computation into a large number of objects
  - Have an Intelligent Run-time system (of Charm++) assign objects to processors for dynamic load balancing

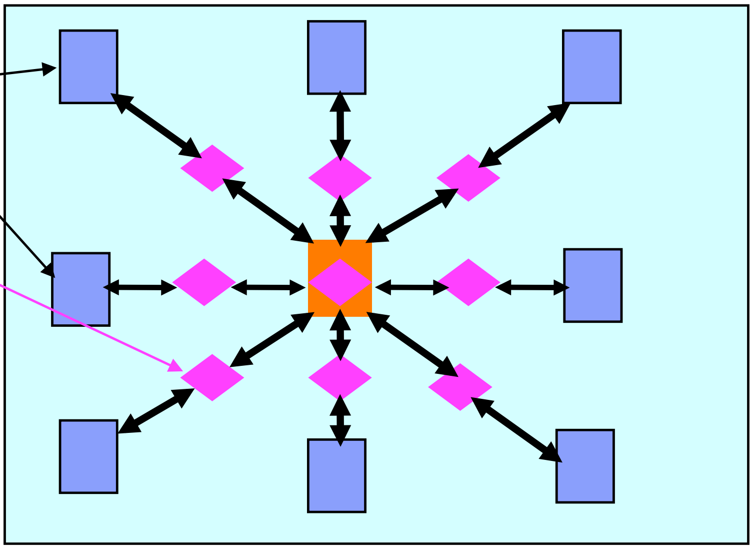


Hybrid of spatial and force decomposition:

•Spatial decomposition of atoms into cubes (called patches)

•For every pair of interacting patches, create one object for calculating electrostatic interactions

•Recent: Blue Matter, Desmond, etc. use this idea in some form



BW Challenges and Opportunities

- Support systems >= 100 Million atoms
- Performance requirements for 100 Million atom
- Scale to over 300,000 cores
- Power 7 Hardware
  - PPC architecture
  - Wide node at least 32 cores with 128 HT threads
- BlueWaters Torrent interconnect
- Doing research under NDA

NAMD on BW

- Leverage Software Stack (XL, etc)
- Use SMT=4 effectively
- Use Power7 effectively
  - Shared memory topology
  - Prefetch (dcbt)
  - Loop unrolling
  - SIMD VSX
- Use Torrent effectively
  - LAPI now, soon PAMI

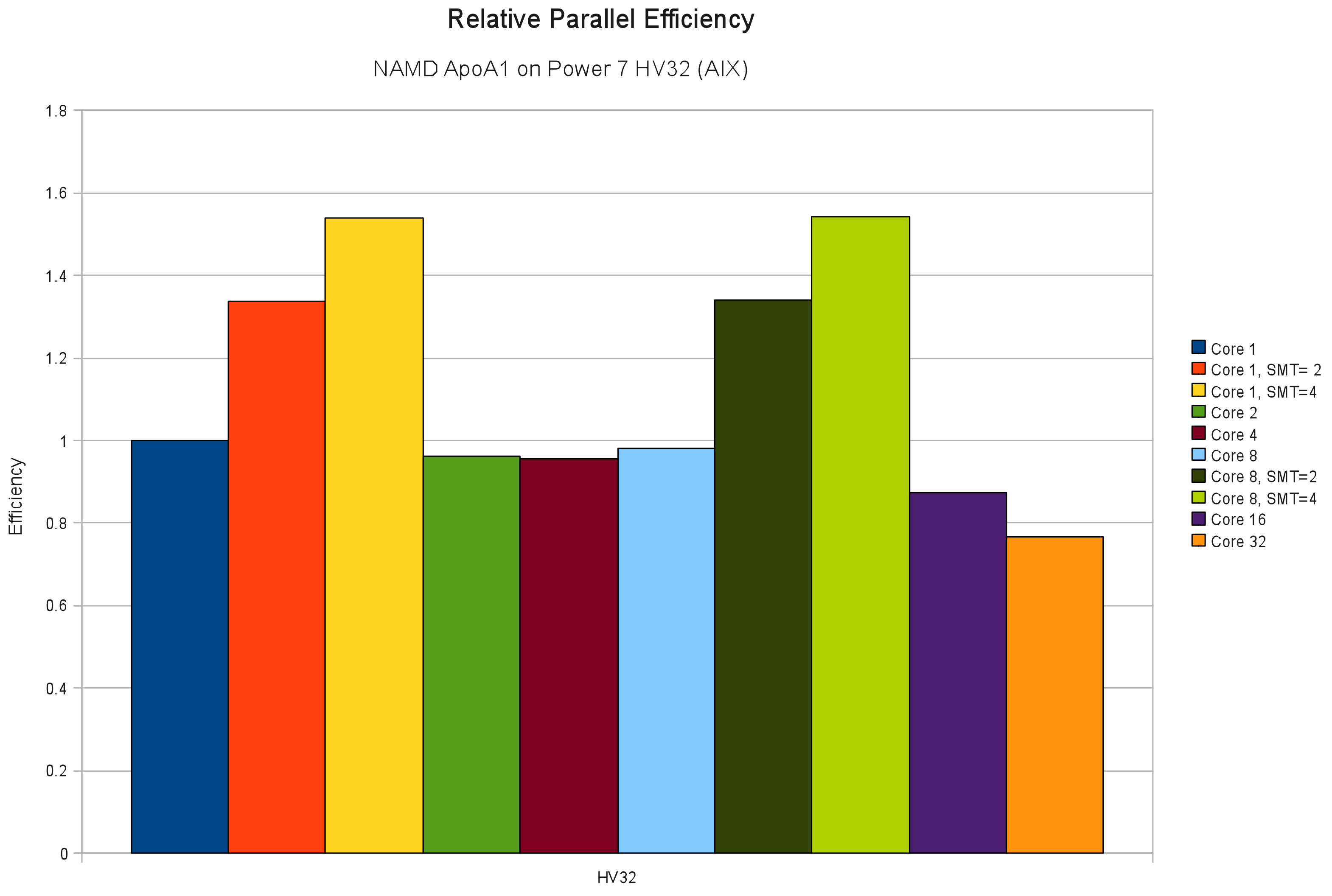
Petascale Scalability Concerns

- Centralized load balancer - solved
- IO
  - Unscalable file formats - solved
  - input read at startup - solved
  - Sequential output – solved
    - Performance tuning ongoing
- Fine grain overhead – in progress
- Non-bonded multicasts – being studied
- Particle Mesh Ewald
  - Largest grid target <= 1024
  - Communication overhead primary issue
  - Considering Multilevel Summation alternative

NAMD and SMT=4

- P7 hardware threads are prioritized
  - 0,1 highest
  - 2,3 lowest
- Charm runtime measure processor performance
  - Load balancer operates accordingly
- NAMD on SMT=4 35% faster than SMT=1
  - No new code required!
- At the limit it requires 4x more decomposition

NAMD on Power7 HV 32 AIX



Performance on P7

- Full node scaling to 32 cores 128 threads
  - Not on MR system
  - BlueDrop memory bandwidth inadequate
  - Good scaling on NDA hardware
    - Cannot report those numbers here

SMT=4 helps

- Need latency tolerance
- One thread works while others blocked on load/store

Finer decomposition

- More synchronization
- More overhead

SIMD -> VSX

- VSX adds double precision support to VMX
- SSE2 already in use in 2 NAMD functions
- Simple MD-SIMD test model performed well.

NSF benchmark requires double precision, reducing SIMD benefits

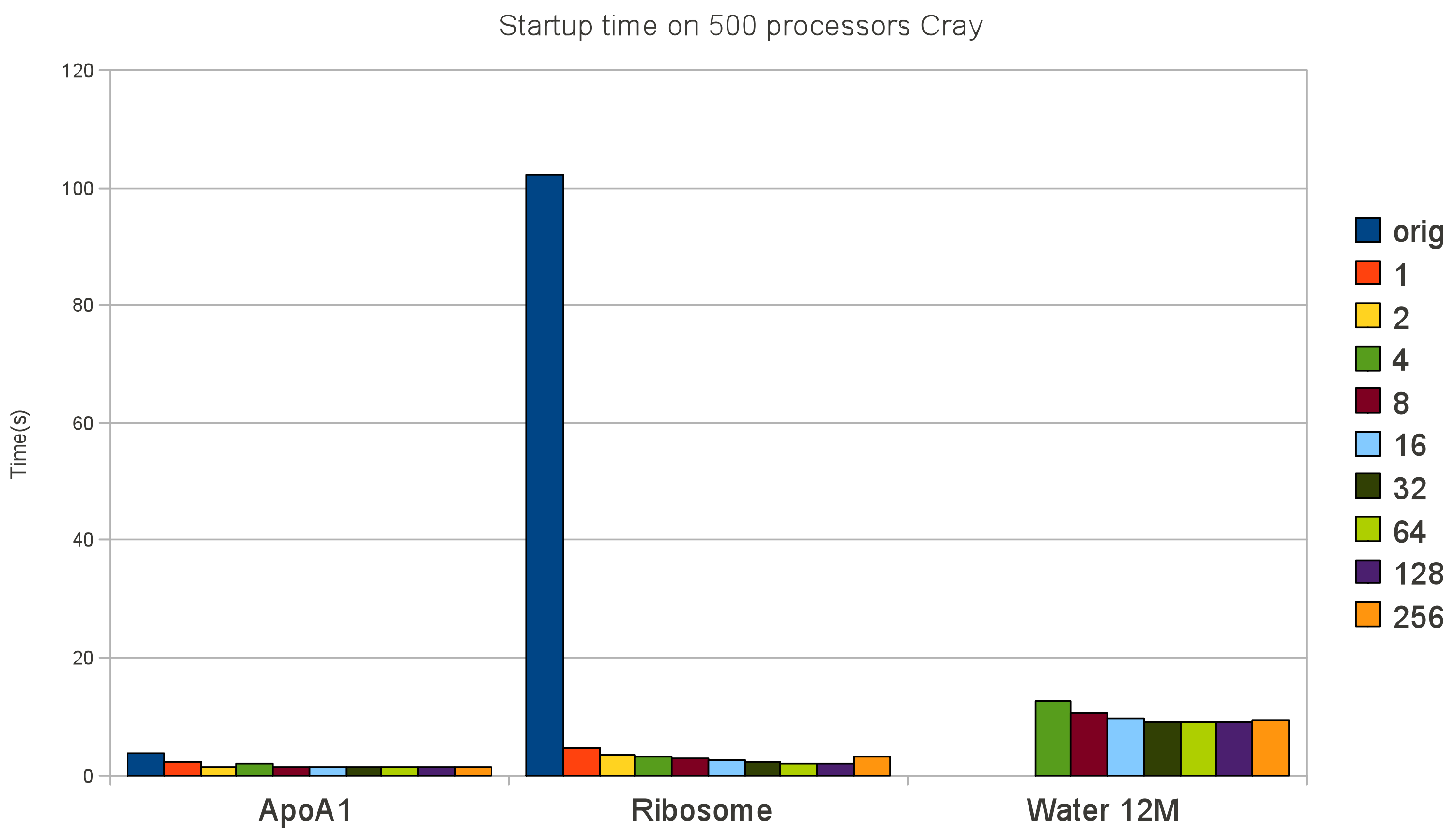
1-2k LOC to refactor

Implementing platform independent short vector SIMD kernel

Support for Large Molecular Systems

- New Compressed PSF file format
  - Supports >100 million atoms
  - Supports parallel startup
  - Support MEM_OPT molecule representation
- MEM_OPT molecule format reduces data replication through atom signatures
- Parallelize reading of input at startup
  - Cannot support legacy PDB format
  - Use binary coordinates format
- Changes in VMD courtesy John Stone

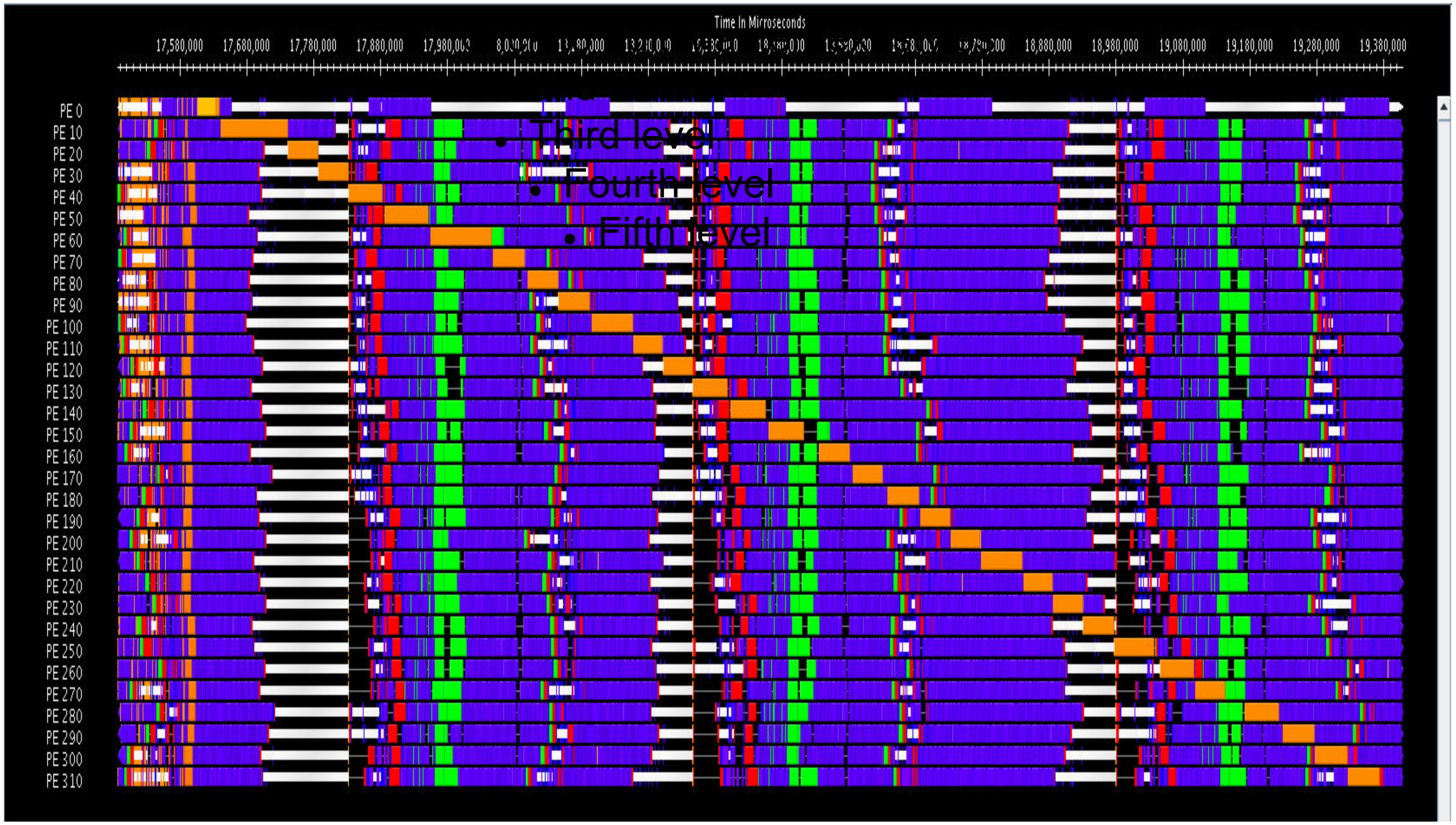
Parallel Startup



Parallel Output

- Coordinate and velocity restart files
- Coordinate and velocity trajectory files
- Memory footprint from sequential output impossible for large systems
- Total data not immense, but is proportional to number of atoms

Only One Writes at a Time

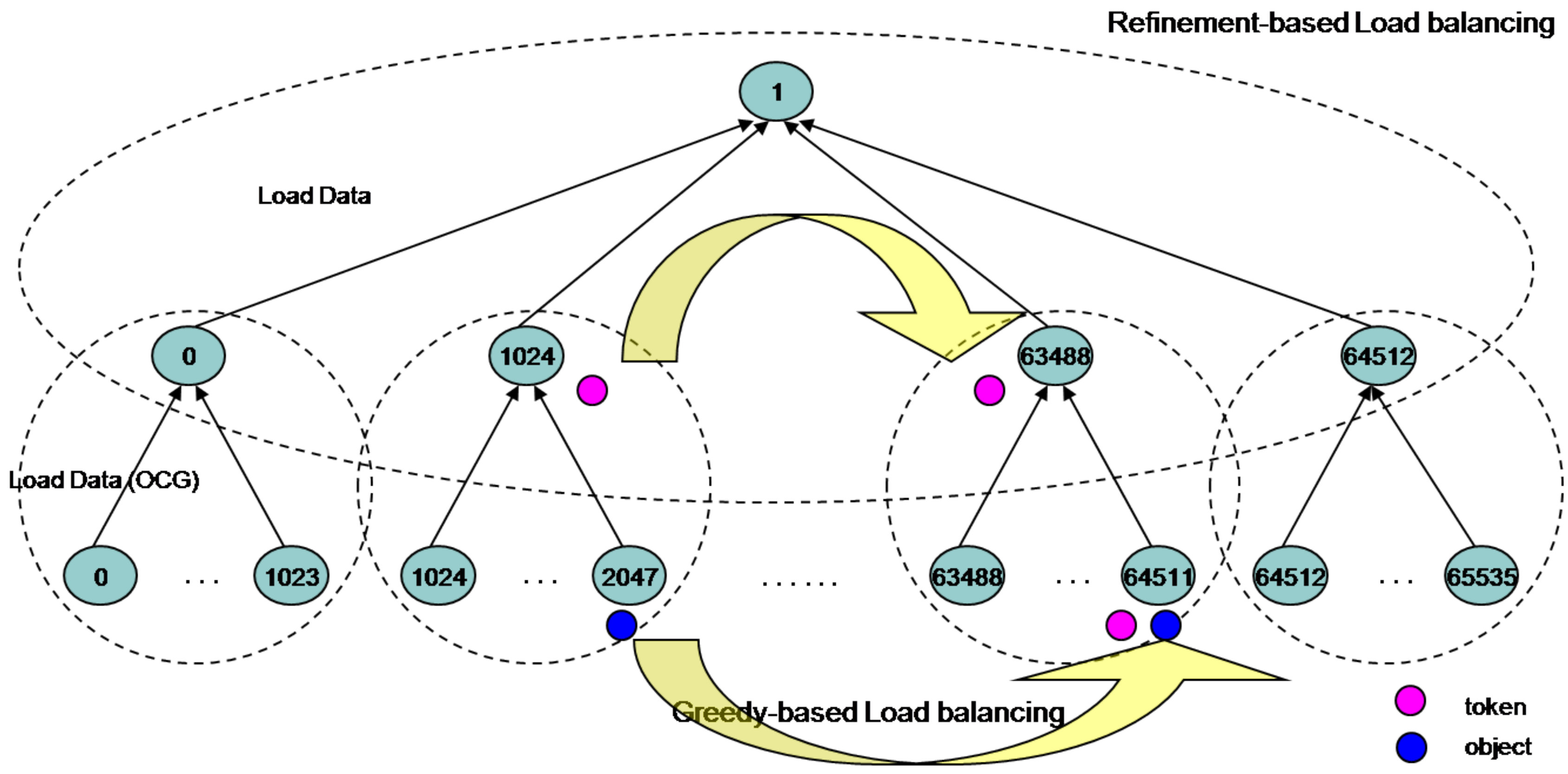


1. **Overlapped with computation**
2. **Crossed multiple timesteps**
3. **Still too long Λ**

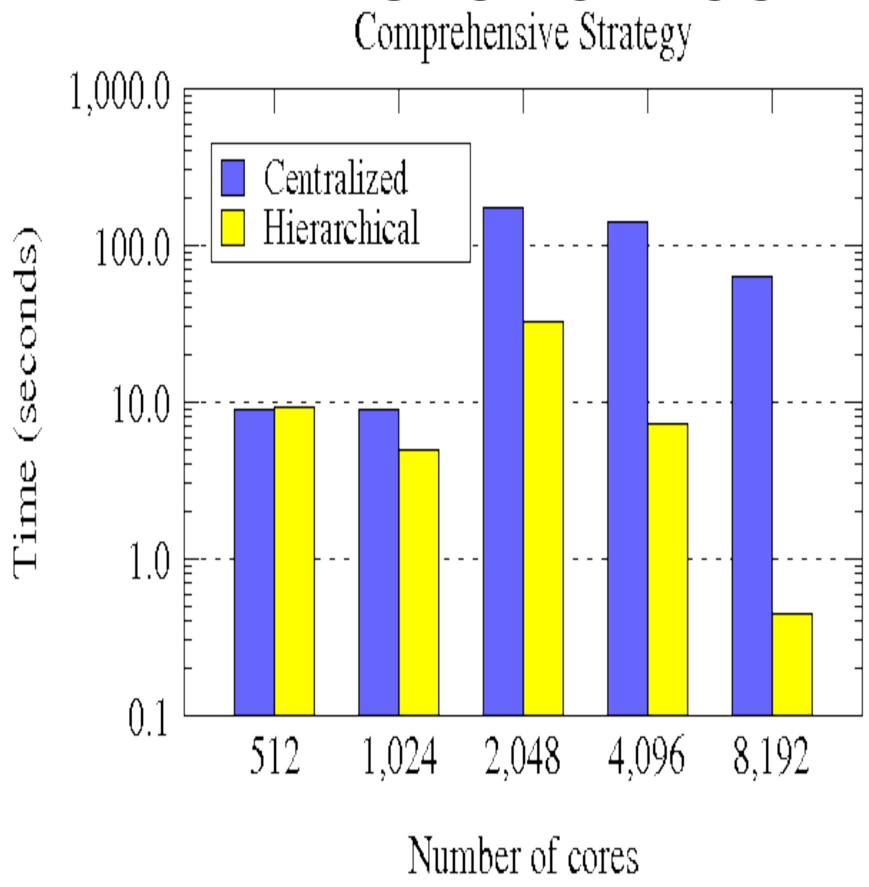
Output ongoing work

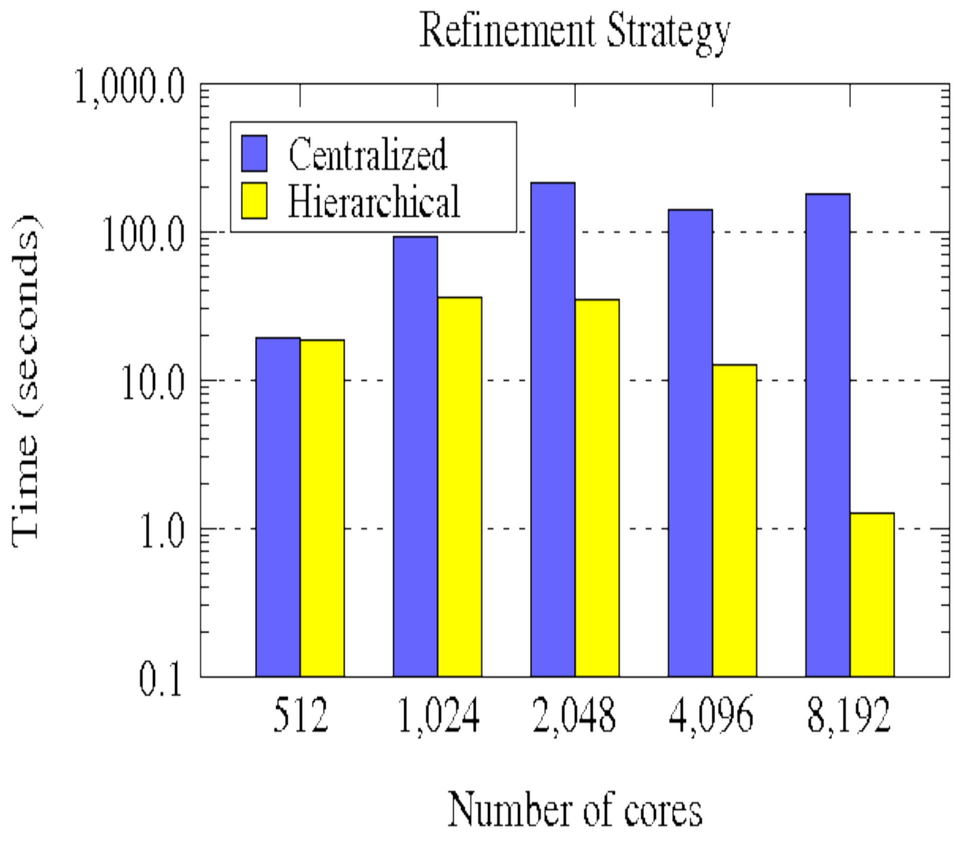
- Time to explore multiple output files
  - Lazily concatenate
  - Or post process
  - Or leave separate when tool chain catches up
- Parallel file systems can usually these well as long as number of files is less than number of cores at the limit
  - Requires some sweet spot discovery for number of writers and files

Hierarchical Load Balancing



Hierarchical LB decision time





Fine grain overhead

- End user targets are all fixed size problems
- Strong scaling performance dominates
  - Maximize number of nanoseconds/day of simulation
- Non-bonded cutoff distance determines patch size
  - Patch can be subdivided along x, y, z dimensions
    - 2 away X, 2-away XY, 2 away XYZ
      - Theoretically K-away...
      - 3 away or even 5 away may provide better initial balance of work
      - Currently researching adaptive decomposition

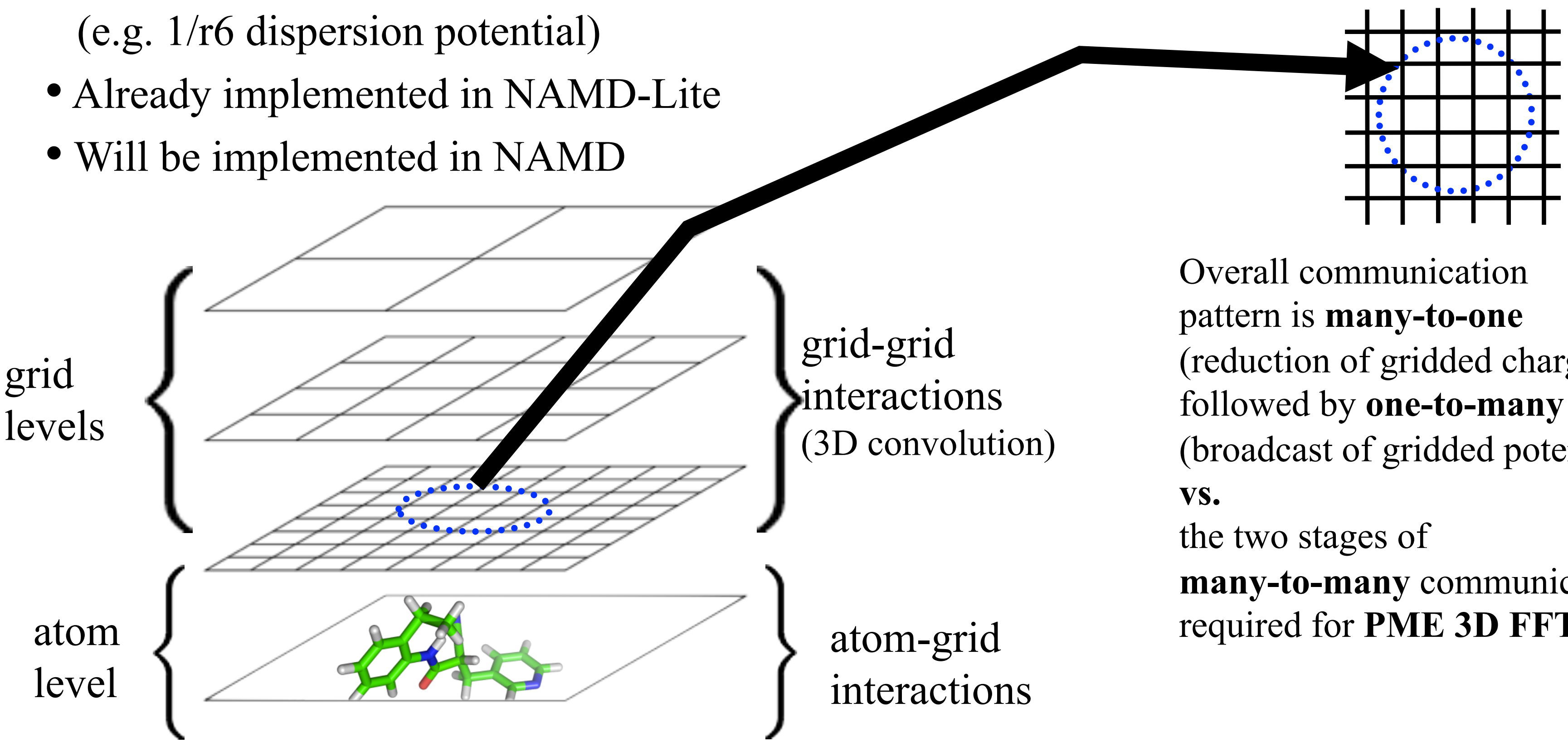
Fine-grain overhead reduction

- Distant computes have little or no interaction
  - Long diagonal opposites of 2-awayXYZ mostly outside of cutoff
- Optimizations
  - Don't migrate tiny computes
  - Sort pairlists to truncate computation
  - Increase margin and do not create redundant compute objects
- Slight (<5%) reduction in step time
- Avoid carrying redundant data in pairlists
  - 10% sequential performance improvement on power 7

Multilevel Summation Method

- N-body solver with better parallel scalability than PME (no 3D FFTs required)
- Supports **periodic** and **non-periodic** boundary conditions
- Algorithmic complexity is **linear** in the number of atoms
- Approach can be applied to other types of potentials (e.g. 1/r6 dispersion potential)
- Already implemented in NAMD-Lite
- Will be implemented in NAMD

Localized communication at each grid level



Overall communication pattern is **many-to-one** (reduction of gridded charge) followed by **one-to-many** (broadcast of gridded potential) **vs.** the two stages of **many-to-many** communication required for **PME 3D FFTs**

Interpolate "smoothings" of the 1/r electrostatic potential from multiple grid levels

PAMI optimizations

- Parallel Active Message Interface
- PAMI is currently NDA
  - Open Source by the time BG/Q is accepted
- Active messages express Charm++ event driven paradigm well
  - Cautiously optimistic about PAMI performance
- Asynchronous Collectives
  - Express communication directly in PAMI primitives
    - More efficient and scalable than building on PtP

Exascale Computation Model

- N = Amount of computation
- Pc = number of processor cores
- n= floating point operations
- tc= time for computing a flop
- 1/η= efficiency factor

$$T_{comp} = 1/\eta \times f(N, P_c) \times n \times t_c$$

Exascale Communication Model

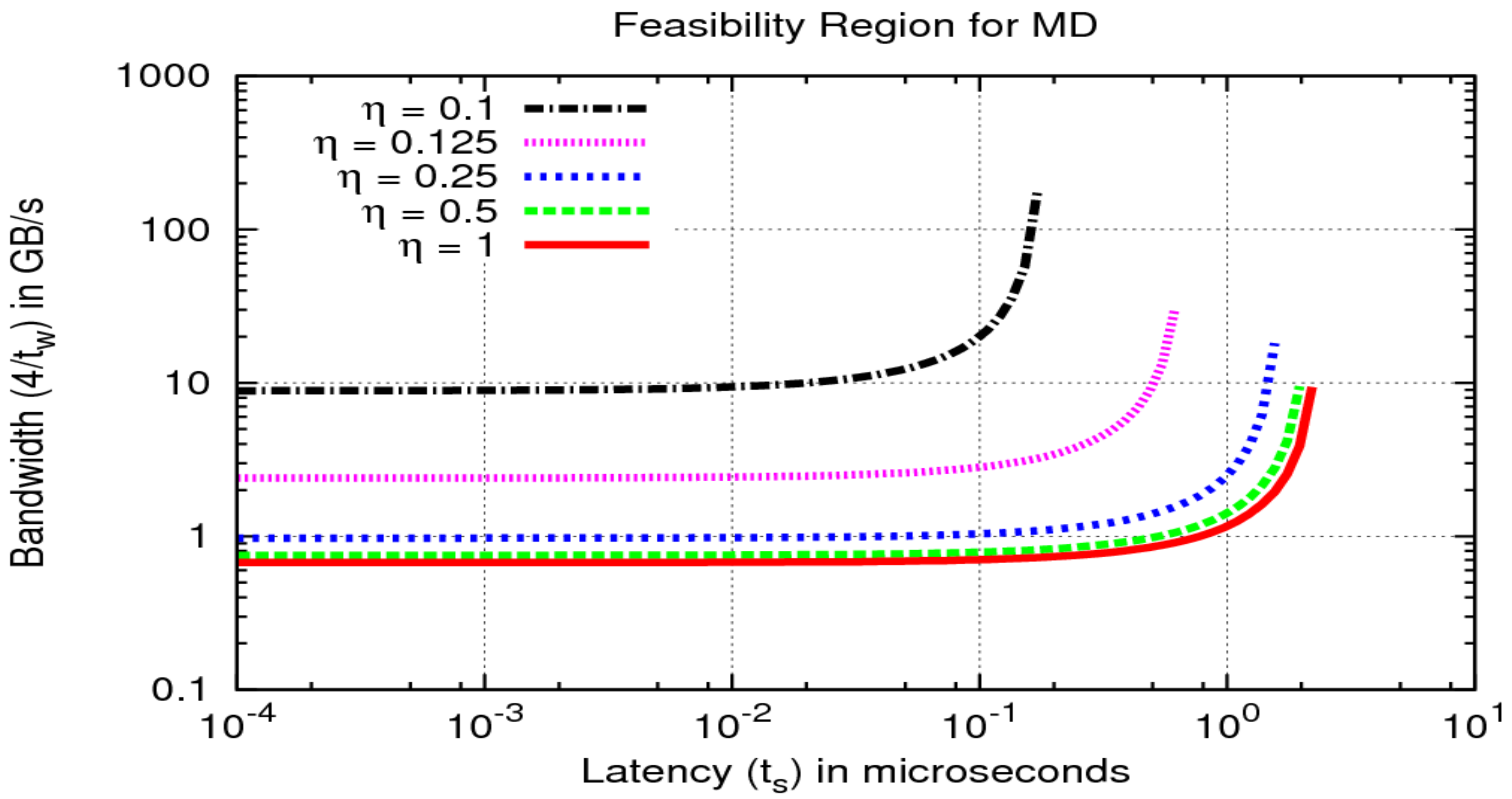
- l= number of links traversed
- Bw = Bandwidth
- ts = time for message handling sender+ receiver
- th = time spent at each link (switch/router/etc)
- tw = per word time (inverse of bandwidth)
- M = size of message in bytes

$$Tcomm = M \times (ts + f\ (N, Pc\ ) \times tw\ )$$

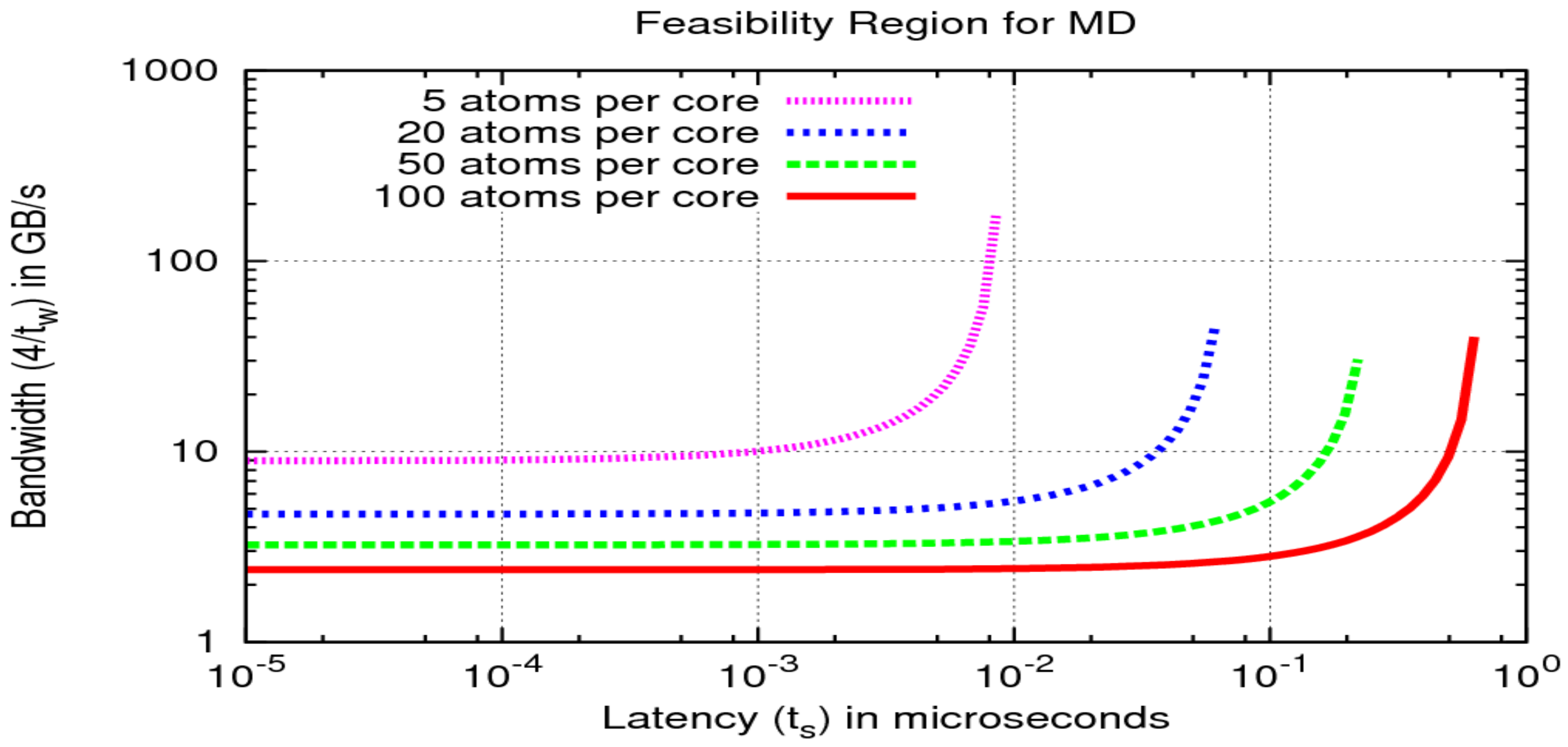
Exascale Feasibility

- Hypothetical exascale machine: 2^30 1 GHz cores, 10flops per cycle, 1000 cores per node
- Time per iteration
  - T = 1/η * flops * tc + M * (ts + b * tw)
- Target: flop/s > 1 Exaflop/s
  - flops/T > 10^18
- Assume 100 atoms/core
  - 107 billion atom system

Exascale MD Weak Scaling



Exascale MD Strong Scaling



Future work

- Improve granularity
- Leverage native communication API
  - PAMI not ready yet
- Particle Mesh Ewald improve/replace
  - Currently constructing analytical model to predict performance
- Parallel I/O optimization
- Exascale feasibility model improvements