Scalable Molecular Dynamics

Neil Stringfellow, Peter Coveney,
Fabrizio Giordanetto

Molecular Dynamics

- Computational Chemistry Application
- Time Dependent Behaviour of Molecular System
- Developed in 1950s (Alder and Wainwright)
- First Protein Simulation in 1970s
- Now – Biological Applications
  - Simulation of solvated proteins
  - Lipid Systems

How Does it Work ?

- Based on Newton's second law

  $F=ma$
- Evaluate forces at each time-step
- Move atoms based on force calculations
- Statistical mechanics to look for convergence properties

Current Codes

- Amber
- CHARMM
- DL_POLY

- High usage, but small jobs (16 or 32 processors maximum)

Current Methods

- Replicated Data Model
  - All data from all atoms on all processes
  - Uses a lot of memory
  - Not scalable
- One time-step for all forces

New Codes

- NAMD
- LAMMPS

- Designed as large scale parallel codes.

Design Strategies

- Distributed data model
  - Atoms distributed across processes
  - More memory efficient
  - Highly scalable for large problems
- Multiple time-step algorithms for different forces

Limitations of Strategies

- Replicated data
  - Large memory requirements
  - Need to communicate all atom data at each step
- Distributed data
  - Electrostatic forces need full atom information
    - This is not stored on each process
    - Difficult algorithms to parallelise

Advantages of Strategies

- Replicated Data
  - Electrostatic forces can be computed locally – save communication time
- Distributed data
  - Larger problems can be solved
  - Fewer communications for short-range forces
  - Can use multiple time stepping

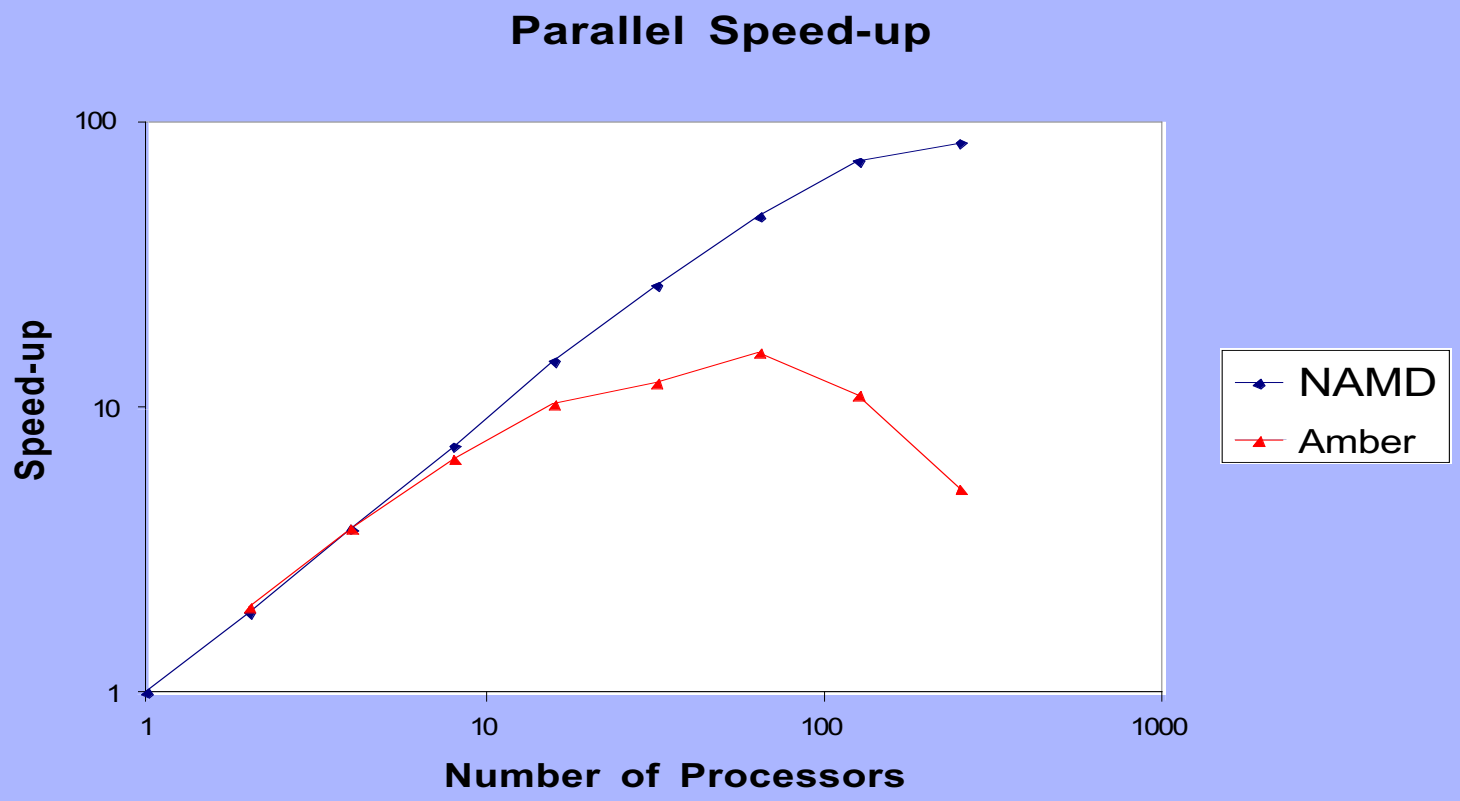
Multiple Time Stepping

- Non-bonded forces can be calculated less frequently
- Electrostatics can be calculated less frequently still
- Typically
  - 1 femtosecond bonded forces
  - 2 femtosecond non-bonded forces
  - 4 femtoseconds long-range electrostatic forces

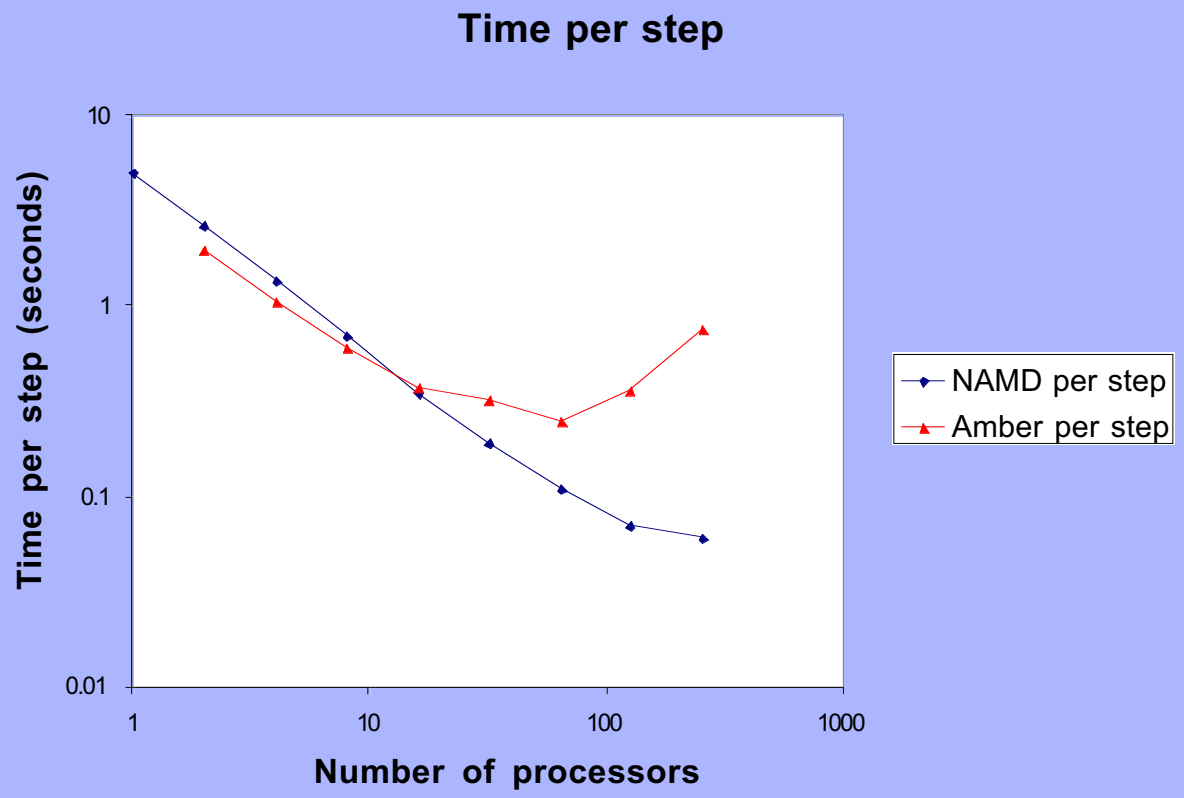
Multi-time Consequences

- Long-range electrostatics isn’t as costly
- Faster run-times
- Better speed-up for Distributed Data models
- Fewer communications

Amber Input in NAMD



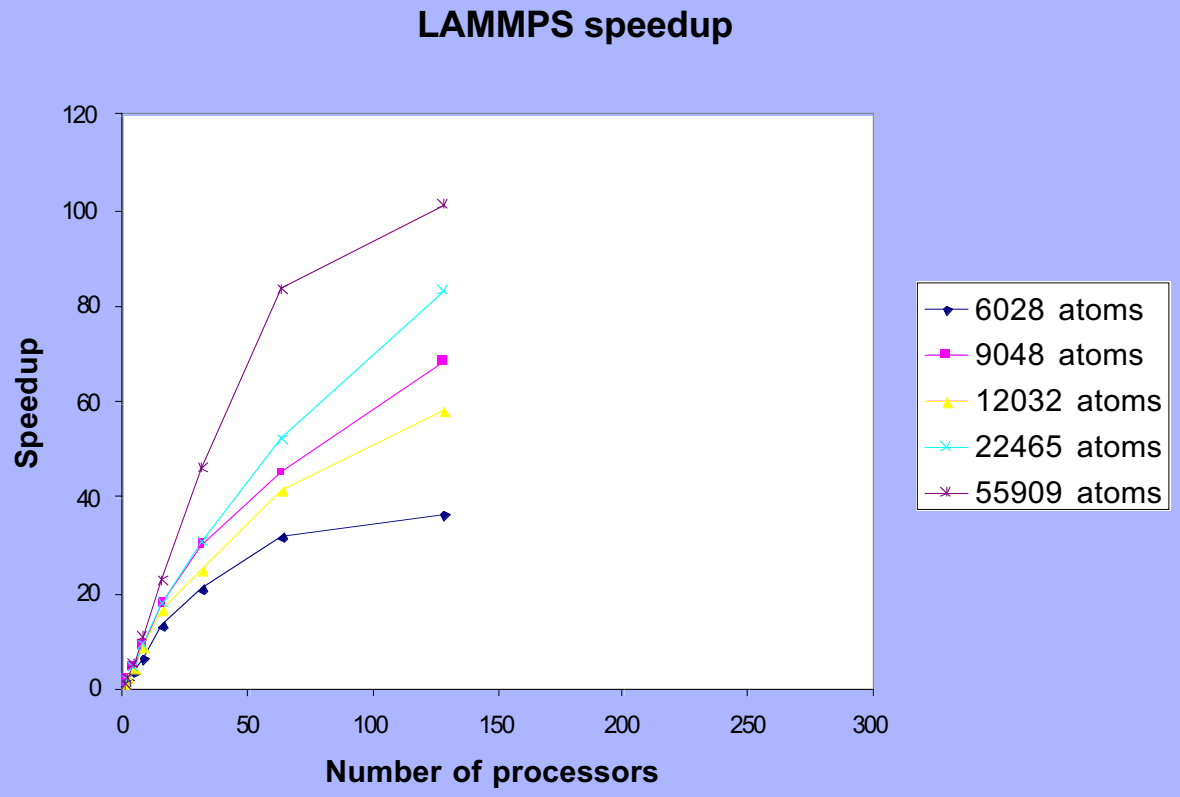
Time per Step NAMD v Amber



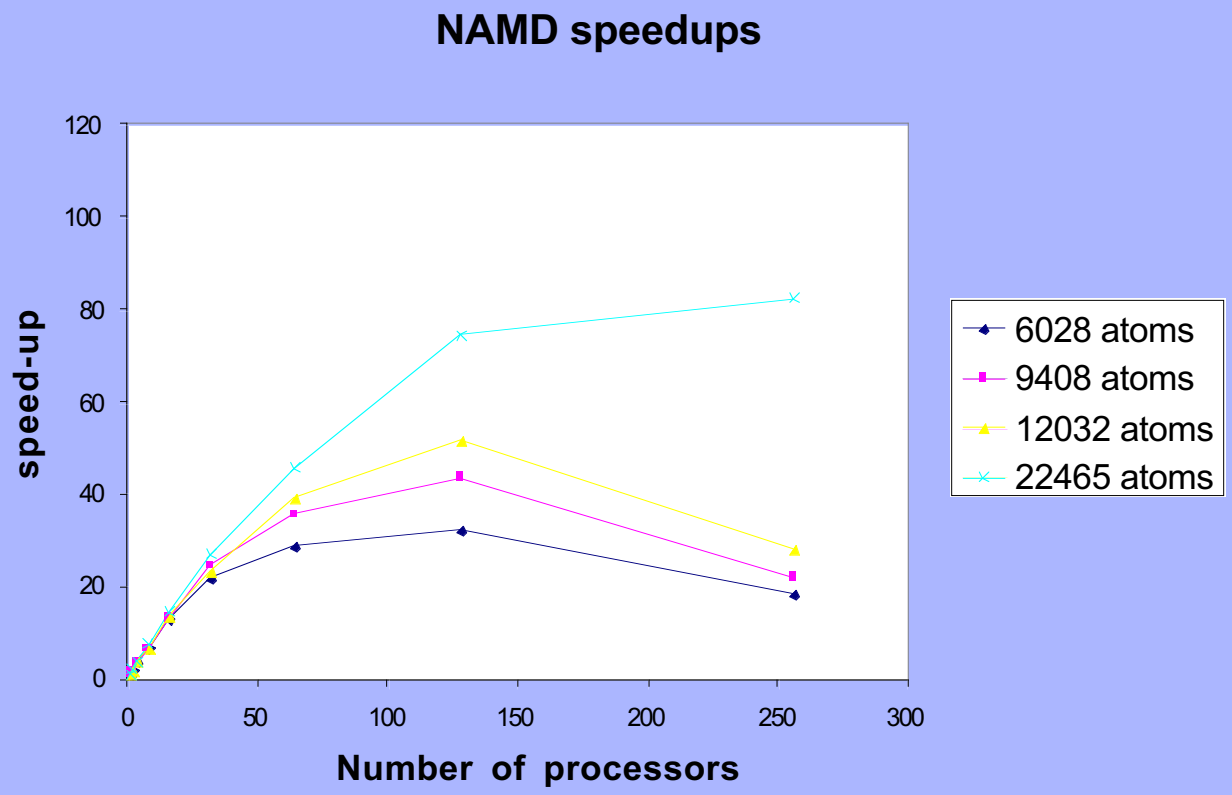
Common Perceptions

- Molecular dynamics codes don’t scale beyond 32 processors
- Only small problems can be simulated
- It takes too long to get results for many simulations

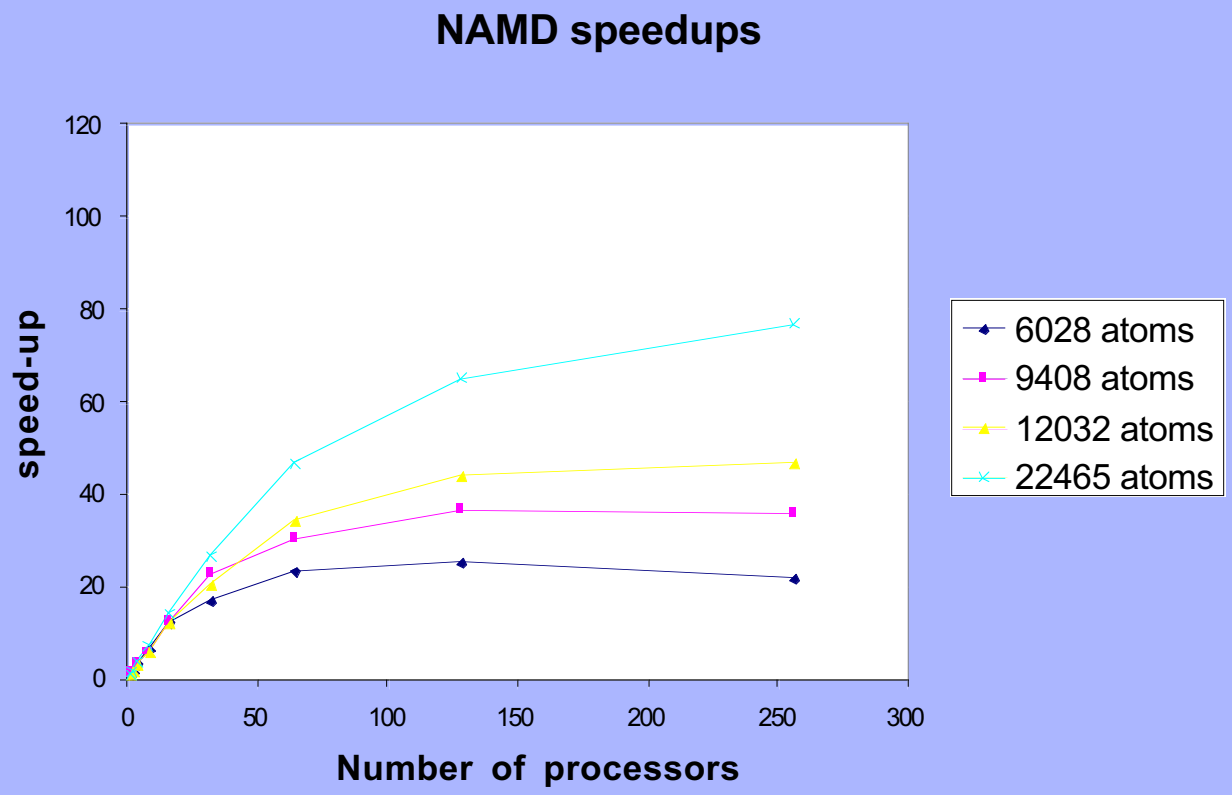
Examples of LAMMPS Scaling



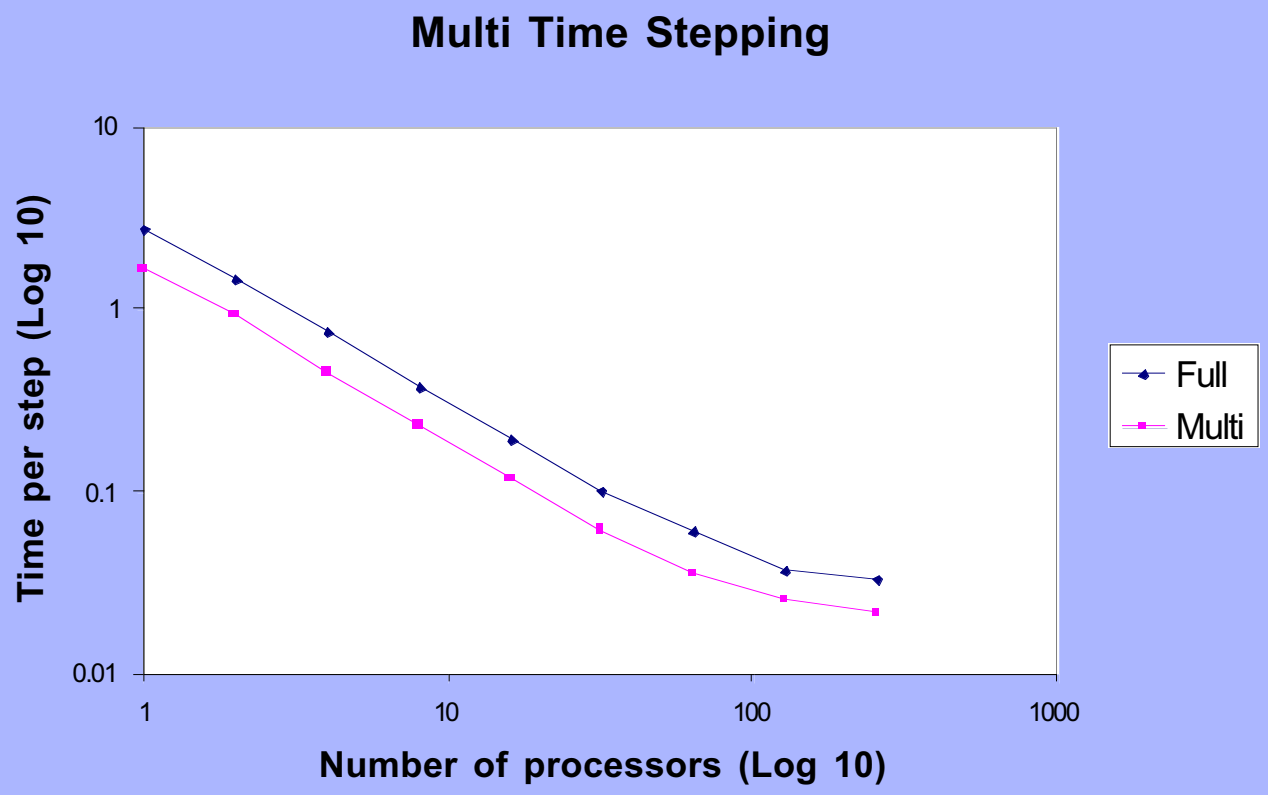
Examples of NAMD Scaling



Multiple Time-stepping



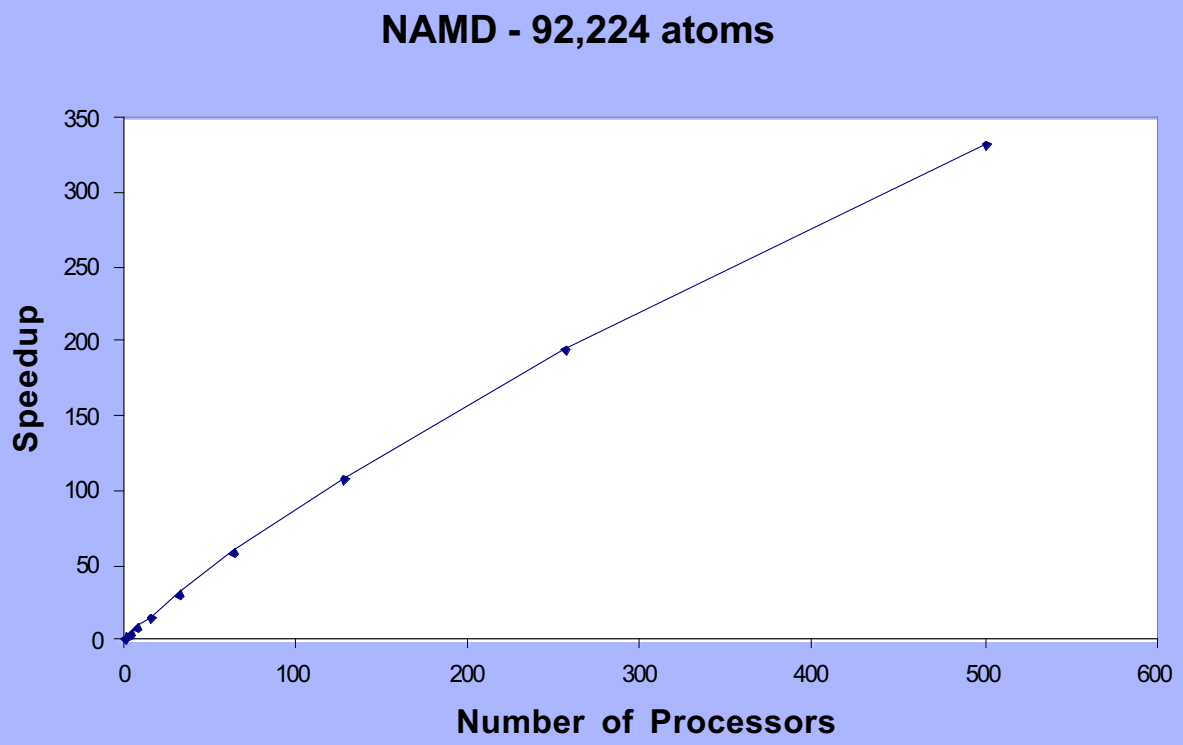
Time Per Step



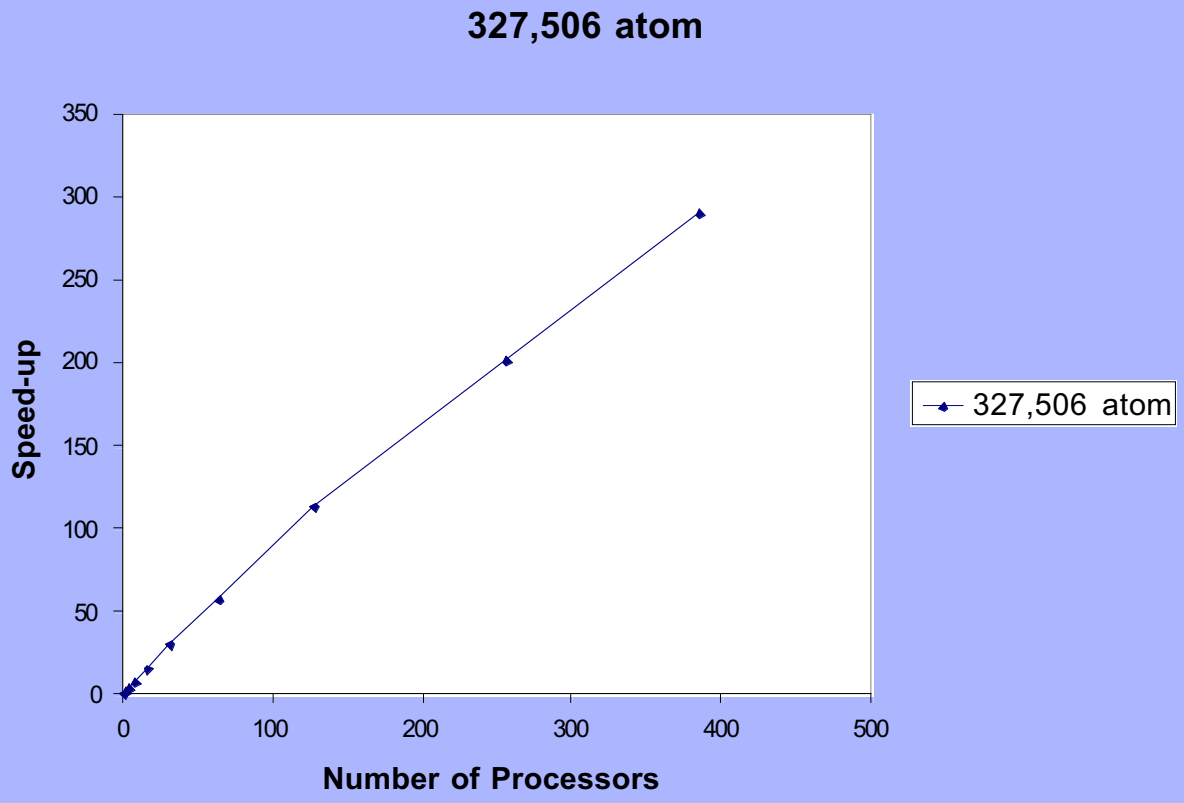
Small Problems

- Scaling is better than Amber
- These problems are too small to scale well
- E.g. 6028 atom problem
  - On 256 processors
  - Each process gets around 24 atoms each
    - Poor load balancing
    - Bad communication to computation ratio

A Bigger Example



Bigger Still …



Real World Applications

- The 92,224 atom simulation
- 256 processors for 24 hours
- Generated 710,000 steps
- Good convergence by this stage
- 1 Nanosecond would take about 1.5 days

Conclusions

- New codes such as NAMD and LAMMPS allow large simulations
- These codes use established force-fields
- Convergence can be achieved in small time scales
- For large problems, scalability is very good
- Scalable molecular dynamics is possible