

CS 267 HW0

WEI XIAO

Department of Materials Science and Engineering,
University of California at Berkeley
(Dated: January 28, 2004)

Research Interests

Computational materials, molecular dynamics, quantum computing.

Platform and Coding Experiences

No cluster experience. Unix/Linux. C, Java, Matlab.

Wei Xiao

CS 267 HW0 Spring 2004

EAM molecular dynamics

- What is the scientific or engineering problem being solved?

Molecular dynamics (MD) simulation is a commonly used tool in the solid-state physics and materials community for modeling solids and liquids at the atomic level. Each atom in the simulation is treated as a point mass and Newton's equations of motion are integrated to track the motion of each atom. All the physics of the model is contained in an energy functional for the system from which the forces on each atom can be computed. Thermodynamic and transport properties as well as general structural and high-temperature properties of materials can be computed from time averaging various quantities over the ensemble of atoms. Parallel computing offers new capabilities for using molecular dynamics (MD) to simulate larger numbers of atoms and longer time scales. S. J. Plimpton used the embedded atom method (EAM) formalism for molecular dynamics on multiple-instruction/multiple-data (MIMD) parallel computers.

- How well did the application achieve its scientific / engineering objective? Are simulation results compared to physical results?

Their new parallel MD algorithm, particularly well-suited for the EAM computation. It can calculate large system as 100,000 atoms.

- What parallel platform has the application targeted? (distributed v. shared memory, vector, etc.) What tools were used to build the application? (languages, libraries)

A 1024-processor nCUBE 2 parallel computer. MPI-based parallel machines can run the code. It is written by C.

- How well did the application perform? How does this compare to the platform's best possible performance?

The nCUBE 2 is a distributed parallel computer in the same price range as a single Y-MP processor. The timing results in the figure show the parallel machine to be 5-15x faster than a single Y-MP processor across a wide range of N (N is number of atoms).

- Does the application "scale" to large problems on many processors? If you believe it has not, what bottlenecks may have limited its performance?

These algorithms can be easily coded for (and ported between) any distributed memory parallel machines which perform message passing to exchange information between processors (MIMD parallel machines). Examples include the nCUBE 2, Intel (Gamma, Delta, and Paragon) machines, and the CM-5 (in message-passing mode).