Parallel Computing in Molecular Dynamics

Xingjie Pan
2016-01-28

1 Bio

I am a first year PhD student from the bioengineering department. My research interest focuses on simulation of bio-molecules, including molecular dynamics, protein folding, protein design and quantum chemistry simulation. The computational motifs of these problems are particle method, nonlinear optimization and linear algebra.

From this class, I wish to learn the programming model and algorithms of parallel computing which could help me write good parallel programs to solve interesting problems.

2 An parallel application: OpenMM

2.1 The problem to solve

OpenMM is a molecular dynamics simulation software. Molecular dynamics is a classical mechanics based method to simulate molecules, in which atoms are represented by points of mass and their trajectories are calculated by Newton’s law.

For biomolecular simulation, a system usually contains the order of $10^5$ atoms and therefore $10^{10}$ pairwise interactions. For the sake of stability, the time step is usually chosen to be 1 femtosecond, while interesting biological phenomena often happen in the time scale of nanosecond to second, which means $10^6$ to $10^{15}$ steps should be calculated. So a major problem is how to make molecular dynamics simulation faster and faster.

2.2 How well did the application achieve its objective?

OpenMM could run sequentially on CPU or parallelly on GPU. The Table 1 shows the benchmark of OpenMM on the dihydrofolate reductase model on different platforms. The numbers in the table are simulation speeds under the unit ns/day. We can see that parallel computing substantially increases the simulation speed.
3 Does the application ”scale” to large problems on many processors?

The most time consuming step in molecular dynamics simulations is calculating long-range forces between atoms, which scales as $O(N^2)$, where $N$ is the number of atoms. Fortunately, the forces are pairwise decomposable and independent to each other, which makes this a strong scaling problem.

However, in the analysis above, we assume that the force between each pair of atoms are saved independently, which requires $O(N^2)$ memory. In practice, after a force is calculated, it will be added to the total force of the atom it acts on. In this case, the race condition emerges.

---

1The data comes from http://wiki.simtk.org/openmm/BenchmarkOpenMMDHFR