Scalability Metrics for Parallel Molecular Dynamics

Parallel Efficiency

We define the efficiency of a parallel program running on P processors to solve a problem of size W. Let $T(W, P)$ be the execution time of this parallel program. ***Speed*** of the program is then $S(W, P) = W/T(W, P)$. ***Speedup***, S_P, on P processors is the speed of P processors divided by that of 1 processor, i.e., $S_P = S(W_P, P)/S(W_1, 1)$. To unambiguously define the speedup, we need to specify how the problem size, W_P, scales as a function of the number of processors, P (which will be discussed in the next few paragraphs). The ideal speedup on P processors is expected to be P, and therefore we define the ***parallel efficiency***, $E_P = S_P/P$.

CONSTANT PROBLEM-SIZE SCALING

In the constant problem-size scaling, the problem size $W_P = W$ is fixed constant. Therefore, the constant problem-size speedup is

$$S_P = \frac{S(W,P)}{S(W,1)} = \frac{W/T(W,P)}{W/T(W,1)} = \frac{T(W,1)}{T(W,P)},$$

and the parallel efficiency is

$$E_P = \frac{S_P}{P} = \frac{T(W,1)}{P \bullet T(W,P)}.$$

Amdahl's law: Consider a case, in which a fraction, f ($\in [0, 1]$), of the work W is inherently sequential and cannot be parallelized, but the rest, $1 - f$, can be divided and executed in parallel on P processors. Then, $T(W, P) = f \bullet T(W, 1) + (1 - f)T(W, 1)/P$. Therefore, the speedup is

$$S_P = \frac{T(W,1)}{T(W,P)} = \frac{1}{f + (1-f)/P}.$$

In the limit of large number of processors, $P \to \infty$, the asymptotic speedup is $S_P \to 1/f$. The constant problem-size speedup is thus limited by the fraction of the sequential bottleneck of the parallel program. For example, if 1% of the work is sequential, the maximum achievable speedup is 0.01/1 = 100, and it does not make sense to use more than ~100 processors to run this program.

ISOGRANULAR SCALING

In the isogranular scaling, the problem size W_P scales linearly with the number of processors: $W_P = P \bullet w$, where the ***granularity*** (or the work per processor), w, is constant. Therefore, the isogranular speedup is

$$S_P = \frac{S(P \bullet w,P)}{S(w,1)} = \frac{P \bullet w/T(P \bullet w,P)}{w/T(w,1)} = \frac{P \bullet T(w,1)}{T(P \bullet w,P)},$$

and the corresponding isogranular parallel efficiency is

$$E_P = \frac{S_P}{P} = \frac{T(w,1)}{T(P \bullet w,P)}.$$

Analysis of Parallel Molecular Dynamics Algorithm

Using the spatial decomposition and the $O(N)$ linked-list cell method, the parallel MD simulation of N atoms executes independently on P processors, and the computation time $T_{comp}(N, P) = aN/P$, where a is a constant. Here, we have assumed that atoms are on average distributed uniformly, so that the average number of atoms per processor is N/P. The dominant overhead of the parallel MD is atom caching, in which atoms near the subsystem boundary within a cutoff distance, r_c, are copied from the nearest neighbor processors and are processed. Since this nearest-neighbor communication scales as the surface area of each spatial subsystem, its time is $T_{comm}(N, P) = b(N/P)^{2/3}$, where b is a constant. Another major communication cost is for global summations, `MPI_Allreduce()`, which incurs $T_{global}(P) = c \log P$, where c is another constant.

The total execution time of the parallel MD program can thus be modeled as

$$\begin{aligned} T(N,P) &= T_{comp}(N,P) + T_{comm}(N,P) + T_{global}(P) \\ &= aN/P + b(N/P)^{2/3} + c\log P \end{aligned} .$$

CONSTANT PROBLEM-SIZE SCALING

For constant problem-size scaling, the global number of atoms, N, is fixed, and the speedup is given by

$$\begin{aligned} S_P &= \frac{T(N,1)}{T(N,P)} = \frac{aN}{aN/P + b(N/P)^{2/3} + c\log P} \\ &= \frac{P}{1 + \frac{b}{a}\left(\frac{P}{N}\right)^{1/3} + \frac{c}{a}\frac{P\log P}{N}} \end{aligned} ,$$

and the parallel efficiency is

$$E_P = \frac{S_P}{P} = \frac{1}{1 + \frac{b}{a}\left(\frac{P}{N}\right)^{1/3} + \frac{c}{a}\frac{P\log P}{N}} .$$

From this model, we can see that the efficiency is a decreasing function of P through both the $P^{1/3}$ and $P\log P$ dependences.

ISOGRANULAR SCALING

For isogranular scaling, the number of atoms per processor, $N/P = n$, is constant, and the isogranular parallel efficiency is

$$E_P = \frac{T(n,1)}{T(nP,P)} = \frac{an}{an + bn^{2/3} + c\log P} = \frac{1}{1 + \frac{b}{a}n^{-1/3} + \frac{c}{an}\log P} .$$

For a given number of processors, the efficiency E_P is larger for larger granularity n. For a given granularity, E_P is a weakly decreasing function of P, due to only the $\log P$ dependence.

Analysis of a Parallel Fast Multipole Method Algorithm

Step 1: Compute multipoles $\Phi_{L,c}$ for all the cells c at the leaf level L of the octree. This computation is performed locally in each processor, and its time complexity is $\tau_1^{comp} = c_1 N/P$, where c_1 is a constant, N is the number of charged particles, and P is the number of processors.

Step 2 (upward pass): For octree levels $l = L-1, L-2, ..., 0$, compute $\Phi_{l,c}$ for all the cells c by summing the multipoles of their 8 children cells (see Fig. 1),

$$\Phi_{l,c} = \sum_{c' \in children(c)} T_{M \leftarrow M}\left(\Phi_{l+1,c'}\right) \tag{1}$$

where the multipole-to-multipole (M2M) translation operator $T_{\mathrm{M}\leftarrow\mathrm{M}}$ shifts the origin of the multipole representation, and *children*(c) is the set of 8 children cells of parent c.

For $l \geq \log_8 P$, each processor computes the multipoles for $8^l/P$ cells. For $l < \log_8 P$, each processor is assigned only one cell to compute. Therefore,

$$\tau_2^{comp} = 8c_2\left(\sum_{l=\log_8 P}^{L-1} \frac{8^l}{P} + \sum_{l=0}^{\log_8 P - 1} 1\right) \sim 8c_2\left(\frac{1}{7\xi^3}\frac{N}{P} + \log_8 P\right) \tag{2}$$

where $c_2 = c_{MM}$ is the computation time associated with each M2M shift operation. The approximation in Eq. (2) holds for the number of leaf cells $N_c = 8^L >> P >> 1$, assuming uniform atomic density. The leaf-cell length in unit of $(\Omega/N)^{1/3}$ is denoted by ξ (Ω is the total volume of the simulated system).

For $l \geq \log_8 P$, the M2M operations are performed locally in each processor. For $l < \log_8 P$, the multipoles of only one child cell are locally available out of 8. The rest must be copied from other processors with the communication cost, $\tau_2^{comm} = 7 d_m \log_8 P$ (d_m is the time required for copying the multipoles of one cell).

Step 3: If periodic boundary conditions are used, summation over infinitely repeated image boxes is carried after step 2. The multipoles $\Phi_{0,0}$ of the total simulation box is used to compute the local Taylor expansion $\Psi_{0,0}$ of the potential due to the $\infty-27$ well-separated images. This step involves a constant cost, $\tau_3^{comp} = c_3$ (the computation is duplicated in all the processors).

Step 4 (downward pass): For $l = 1, 2, ..., L$, compute the local Taylor expansion $\Psi_{l,c}$ for all the cells c (see Fig. 1),

$$\Psi_{l,c} = T_{L\leftarrow L}\left(\Psi_{l-1,parent(c)}\right) + \sum_{c' \in \mathrm{int}\, eractive(c)} T_{L\leftarrow M}\left(\Phi_{l,c'}\right) \tag{3}$$

The first term in Eq. (3) is the contributions from the parent's well-separated cells. This is inherited from the parent, *parent*(c), of the c-th cell by shifting the origin of the Taylor expansion using the local-to-local (L2L) translation operator $T_{\mathrm{L}\leftarrow\mathrm{L}}$. The second term is due to the atoms in the interactive cells, which are the children of the parent's nearest-neighbors but are well-separated from the c-th cell. The set of all the interactive cells of the c-th cell is denoted by *interactive*(c). The multipole-to-local (M2L) translation operator $T_{\mathrm{L}\leftarrow\mathrm{M}}$ converts the multipoles of an interactive cell to local Taylor expansion coefficients centered at the c-th cell.

Since each cell has $6^3-3^3 = 189$ interactive cells, the computation time per cell is $c_4 = c_{LL} + 189c_{ML}$, where c_{LL} and c_{ML} denote the computation costs associated with the L2L and M2L operators, respectively. For $l \le \log_8 P$, each processor computes the local expansions for one cell. For $l > \log_8 P$, each processor is assigned the $8^l/P$ local cells.

$$\tau_4^{comp} = c_4\left(\sum_{l=\log_8 P+1}^{L}\frac{8^l}{P} + \sum_{l=1}^{\log_8 P}1\right) \sim c_4\left(\frac{1}{7\xi^3}\frac{N}{P} + \log_8 P\right), \tag{4}$$

For $l \le \log_8 P$, the multipoles of up to 189 interactive cells must be copied from other processors. For $l > \log_8 P$, each subsystem must be augmented by copying two boundary layers of cells from other processors. The associated communication cost is

$$\tau_4^{comm} = d_m\left[\sum_{l=\log_8 P+1}^{L}\left\{\left(\frac{2^l}{P^{1/3}}+4\right)^3 - \frac{8^l}{P}\right\} + \sum_{l=1}^{\log_8 P}189\right] \sim d_m\left\{\frac{16}{\xi^2}\left(\frac{N}{P}\right)^{2/3} + 189\log_8 P\right\}. \tag{5}$$

Step 5: The far-field contribution is evaluated at each atom's position using the local Taylor expansion $\Psi_{L,c(i)}$ at the leaf level, where $c(i)$ denotes the leaf cell to which the i-th atom belongs. This computation is performed locally with the cost $\tau_5^{comp} = c_5 N/P$.

Step 6 (near field): Contributions to the electrostatic potential from all the atomic pairs within the 27 nearest-neighbor cells are evaluated directly without using multipoles. Using the Newton's third law, $\tau_6^{comp} = (27N_{inner}c_6\xi^3/2)(N/P)$.

At step 6, each subsystem is augmented with the atoms in one boundary layer of cells with the cost,

$$\tau_6^{comm} = \frac{d_a N}{N_c}\left\{\left(\frac{2^L}{P^{1/3}}+2\right)^3 - \frac{8^L}{P}\right\} \sim 6d_a\xi\left(\frac{N}{P}\right)^{2/3}, \tag{6}$$

where d_a is the communication cost per copied atom.

The total execution time is the sum of the above contributions, $\tau(N, P) = \tau^{comp}(N, P) + \tau^{comm}(N, P)$, where $\tau^{comp}(N, P) = \tau_1^{comp} + \tau_2^{comp} + \ldots + \tau_6^{comp}$ and $\tau^{comm}(N, P) = \tau_2^{comm} + \tau_4^{comm}$. Using the isogranular scaling, the parallel efficiency of the program is defined as $E = \tau^{comp}(N/P, 1)/\tau(N, P)$. According to the above analysis, the parallel efficiency of the fast multipole method program is estimated to be

$$E^{-1} = 1 + \frac{(8c_2 + c_4 + 196d_m)P\log_8 P/N + \left\{16d_m/\xi^2 + 6d_a\xi\right\}(P/N)^{1/3}}{c_1 + c_5 + 8(c_2 + c_4)/7\xi^3 + 27c_6\xi^3/2}. \tag{7}$$

In deriving Eq. (7), we have omitted the term proportional to c_3, which was found small for the systems we have tested. For a large number of processors P, the term proportional to $P\log_8 P/N$ in Eq. (7) degrades the parallel efficiency significantly. The efficiency can be increased by increasing the granularity N/P.

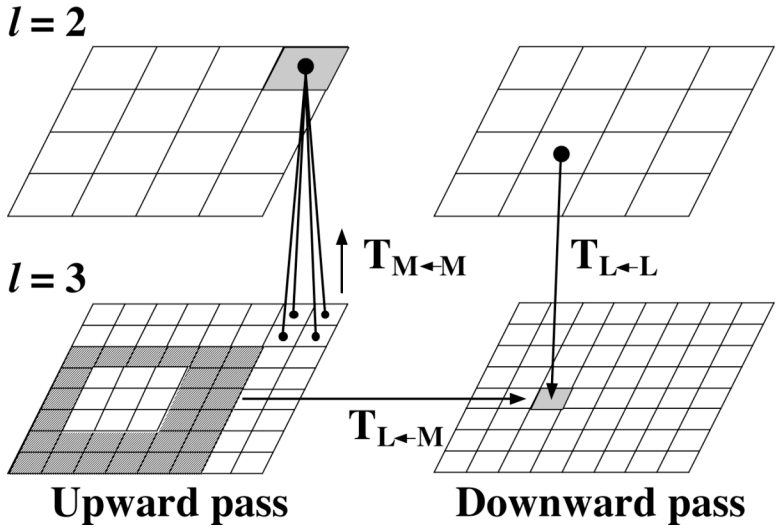

Fig. 1: Schematic of the far-field computation in a two dimensional system. In the left column, the multipoles of a parent cell (shown in gray) at level 2 is obtained by shifting the multipoles of its 8 children cells at level 3 by the M2M translation operator $T_{M\leftarrow M}$ and summing them. In the right column, the local Taylor expansion coefficients of a child cell (gray) at level 3 are computed from two contributions. First the local expansions of its parent at level 2 are inherited; the L2L translation operator $T_{L\leftarrow L}$ is used to shift the origin of the local expansion. The M2L translation operator $T_{L\leftarrow M}$ is then used to compute the contributions due to the interactive cells (shaded) at the same level.

Reference

1. "Parallel multilevel preconditioned conjugate-gradient approach to variable-charge molecular dynamics," A. Nakano, *Computer Physics Communications* **104**, 59-69 (1997).