

分子動力学プログラムのWindowsMPI並列処理

5H-4

橋本 孝一
茨城大学大学院理工学研究科

石黒 美佐子
茨城大学工学部

1. はじめに

分子動力学法(MD法[1])は、本来、多量の粒子による多量のタイムステップからなる膨大な計算で構成される。この計算をPC上で並列に実行し、高速化する方法を確立する。

本研究では、MD法プログラムを非均一な速度のPCから成るクラスタを用いて並列化する。さらに、2プロセッサ搭載のPCではスレッド[2]を用いて並列化する。OSは、通常PC上の並列処理でよく使われているLinuxではなく、Windows NT/2000とし、通信はWMPI通信ライブラリ(Windows Message Passing Interface [3])を使用する。並列の方法には、空間分割法を用い、自動負荷分散を考慮する。

2. 分子動力学プログラムの並列処理

MD法とは、ニュートンの運動方程式に従って粒子を動かしていくことにより、多粒子系の動きを調べる計算機シミュレーション法である[1]。

MD法において、粒子間の相互作用はポテンシャルエネルギーで与えられる。一般に、粒子 ij の粒子間距離が r_{ij} のとき、ポテンシャルエネルギーは $\phi(r_{ij})$ で与えられる。粒子間の相互作用力 F_{ij} および粒子 i の受ける力 F_i は次のようになる。

$$F_{ij} = -\text{grad } \phi(r_{ij})$$

$$F_i = \sum_j F_{ij} \tag{1}$$

式(1)より粒子 i について、次の運動方程式が得られる。ここで、 $r_i(t)$ は時刻 t における粒子 i の位置ベクトルであり、 m_i は粒子の質量である。

$$m_i d^2 r_i(t) / dt^2 = F_i(t) \tag{2}$$

式(2)をヴェルレ法で、粒子の位置・速度を更新して行き、シミュレーションを行う。

なお、本研究では粒子間の相互作用を示すポテンシャルエネルギーに、レナード・ジョーンズポテンシャルを用いる。このポテンシャルは、

$$\phi(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] \tag{3}$$

と表される。ここで、 σ は粒子の直径、 ϵ はエネルギーの次元を持つ定数である。

本研究で扱う粒子は、面心立方格子状で1つの格子当たり平均4つずつ配置される。この面心立方格子を x, y, z 方向に積み重ね、計算対象とする空間を得る。

これを並列化する際のデータの分割法には、粒子の数で分割する「粒子分割法」と、計算対象空間を分割する「空間分割法」とがある。粒子分割法では、各PCが担当する粒子の数が一定になるように、粒子をPCに割り当てる。これにより各PCの負荷は等しくなるが、各PCは粒子間の相互作用の計算を行うために全ての粒子データを参照するので、全てのPCと通信を行う必要がある。

一方、空間分割法では相互作用の非常に小さい遠距離粒子をカットすれば、隣接PCとの通信のみで処理できる(図1参照)。また、台数が多いほど、カットオフの判断に要する計算が減少する隠れた利点もある。ただし、この方法ではタイムステップ毎で各PCが受け持つ粒子数が変化するので、粒子を管理するテーブルが必要となる。今回は z 軸方向で分割し、カットオフを $2a$ (a は格子サイズ)とする。

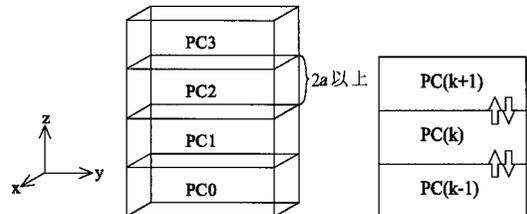


図1 計算領域の割当てとPC間の通信

非均一なPCクラスタでの並列計算に生じる問題は、負荷分散の方法にある。ユーザは必ずしも最適な負荷分散が分かっているとは限らないので、自動的に行われる方がよい。そこで、プログラムの開始時にPCの速さを測定し、それに応じて負荷分散することにする。

各PCが担当する領域の大きさは、下記の式より決定する。ただし、カットオフを考慮し、最低 $2a$

$$(\text{領域}) = \frac{(\text{当該PCの速さ})}{(\text{使用PCの速さの総和})} \times (z \text{ 軸方向の格子数})$$

以上の領域を担当させる。

3. PCクラスタによる数値実験

計算モデルにはテスト時間を考慮し、面心立方格子を x, y, z 方向にそれぞれ 8, 8, 20 ずつ組み合わせた、粒子数が 5120 の小規模のものを用いる。計算は 300 ステップとする。PC は表 1 の性能のものを最大 6 台使用する。ここでは、二通りの構成でテストする。1 つは、クロック周波数の高い PC による構成（ケース 1）である。もう 1 つは、非均一な PC による構成（ケース 2）である。各台数での構成と、その時の z 方向領域の割合を表 2、表 3 に示す。各ケース、各台数での計算時間を測定し、速度向上を算出する。この結果を図 2 に示す。

表 1 各 PC の性能

	総 数	CPU		Cache (kB)		Bus speed
		個数	クロック (Mhz)	1 次	2 次	
A	4	1	1100	128	256	full
B	2	1	850	128	512	2/5
C	1	1	450	32	512	half
D	1	2	450	32	512	full

表 2 PC の構成と領域の割合（ケース 1）

台数	構成	z 方向領域の割合
1	A	20
2	A×2	10:10
3	A×3	6.7:6.6:6.6
4	A×4	5.5:5:5
5	A×4+B	4.2:4.2:4.2:4.2:3.2
6	A×4+B×2	3.7:3.7:3.6:3.6:2.7:2.7

表 3 PC の構成と領域の割合（ケース 2）

台数	構成	z 方向領域の割合
1	A	20
2	A+B	11.3:8.7
3	A+B+C	9.3:7.0:3.7
4	A+B+C+D	7.9:5.9:3.1:3.1
5	A×2+B+C+Ds*	5.7:5.6:4.3:2.2:2.2
6	A×2+B×2+C+Ds	4.5:4.5:3.5:3.5:2.0:2.0

Ds:1 プロセッサ使用, Dd:2 プロセッサ使用

・ケース 1 では 4 台以上のとき、台数効果以上の結果が得られた。これは、カットオフの判断に要する計算が減少するためである。

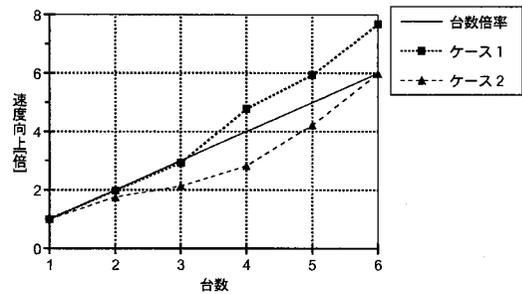


図 2 台数による速度向上

・ケース 2 では一部低速な PC を用いているが、クロック周波数に応じて負荷分散を行ったため、よい結果が得られた。

4. スレッド処理による数値実験

2 プロセッサ搭載した PC での並列化の実験を行う。計算モデルは PC クラスタと同等のものを用いる。共有メモリのために通信の必要がないものの、スレッド制御のオーバーヘッドによって、逐次計算より若干遅くなった。

表 4 スレッド処理による結果

	計算時間 [秒]	速度向上
Ds	3303.6	1.0
Dd	1706.3	1.9

5. まとめ

PC クラスタにおける WMPI を用いた並列処理が実現できた。このとき、カットオフの導入による隠れた利点により、台数効果以上の速度向上が得られた。また、性能が異なる PC を用いた場合、クロック周波数に合わせた負荷分散に近づけるとよい速度向上が得られた。

2 プロセッサ搭載の PC 上でのマルチスレッドによる並列処理ができた。このとき、速度向上は約 1.9 倍であった。

今後は、マルチスレッド処理を含め種々の PC による、より効果的な負荷分散を行いたい。

参考文献

- [1] 大澤映二, 片岡洋右: 分子動力学法とモンテカルロ法, 講談社サイエンティフィック, 1995.
- [2] James E. Beveridge, Robert Wiener: Win32 マルチスレッドプログラミング, アスキー出版局, 1999.
- [3] Critical Software: WMPI

<http://www.criticalsoftware.com/wmpi/home/index.html>