計算機アーキテクチャ 147－3
ハイパフォーマンス 89－3
コンピューティング
（２００２．３．７）

WindowsMPIによる非均一PCクラスタ並列分子動力学法

橋本 孝一† 石黒 美佐子* 絵幡 大輔†
† 茨城大学大学院理工学研究科 * 茨城大学工学部
茨城県日立市中成沢町・0294-38-5194／ishiguro@ipc.ibaraki.ac.jp

分子動力学法は，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．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ではマルチスレッドを用いて並列化する．OSは，PC上の並列処理でよく使われているLinuxではなく，通常のWindows NT/2000とし，通信はWMPI通信ライブラリ(Windows Message Passing Interface)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．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．その上，PC台数が多いほど，カットオフの判断に要する計算が減少し，並列化効果が台数効果を上回るという隠れた利点があることがわかった．

キーワード：分子動力学法，並列プログラム，WMPI，Windows，空間分割法，自動負荷分散，PCクラスター，非均一PC，カットオフ，コンピュータシミュレーション

Parallel Molecular Dynamics Method on Heterogeneous PC Cluster with WindowsMPI

Koichi Hashimoto†, Misako Ishiguro*, and Daisuke Ebata†
† Graduate School of Science and Engineering, Ibaraki University, *Faculty of Engineering, Ibaraki University
Nakanarusawa-Cho, Hitachi city, Ibaraki , 0294-38-5194 / ishiguro@ipc.ibaraki.ac.jp

Molecular dynamics (MD) method consists of the calculation of enormous particles and enormous time steps. In this study, we try to parallelize the MD program using a PC cluster composed of heterogeneous PCs. The program is also executed in parallel on dual processor PC by multi-threads of Visual C++. Here, usual Windows NT/2000 operating system is used rather than Linux, and accordingly, Windows MPI (WMPI) communication library is applied as the message passing interface. Cut-off method is introduced to calculate particle-particle force. Domain decomposition method is applied on the basis of an automatic load balancing policy. As the result, it is found that the cut-off brings into an additional parallelization effect under the domain decomposition, where the speed-up ratio exceeds the CPUs number effect because of saving of computing time required for the cut-off judgment.

Keywords: Molecular dynamics, Parallel program, WMPI, Windows, Domain decomposition, Automatic load balancing, PC clusters, Heterogeneous PC, Cut-off, Computer simulations

1．はじめに

分子動力学法（Molecular Dynamics; MD）法[1]は，多数の粒子からなる仮想的な系を考え，ポテンシャル関数によって規定された粒子間相互作用を用いて各粒子に働く力を求め，ニュートンの運動方程式により全粒子を動かし，粒子の位置と速度などの情報から種々のマクロ量を求めるものである．計算化学や材料物性のシミュレーションなど様々な研究，業務に使用されてきた．

分子動力学法は，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．分子動力学法については既に種々の並列計算機で種々の並列化方法に関する工夫が成され成果が報告されている[2],[3].

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 Visual C++言語のマルチスレッド[4]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 [5]ではなく，通常業務で最もよく利用されている Windows NT/2000 環境とするところに特徴がある．通信には WMPI（Windows Message Passing Interface[6]）通信ライブラリを用いる．並列の方法には，空間分割法を用い，実測に基づく自動負荷分散を採用する．以上の方法を用いて非均一な PC クラスタによる並列化効果を検証する．

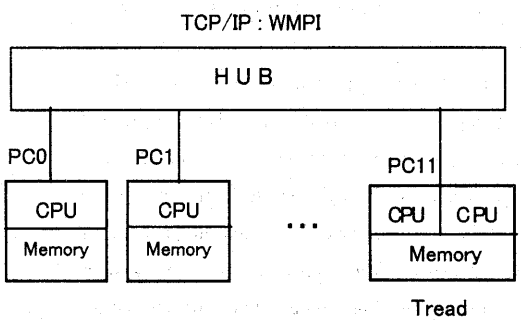

図 1　非均一 PC クラスタの構成

2．MD 法の並列計算処理

2.1 数学モデル

MD 法は，粒子シミュレーションの 1 つの手法で，ポテンシャル関数，粒子数とその初期位置，初速度などを入力パラメータで与え，ニュートンの運動方程式に従い，粒子の位置と速度を更新していくものである．このとき粒子 i,j の粒子間力 $\mathbf{F}_{ij}$ は距離 r_{ij} の関数として

$$\mathbf{F}_{ij} = -grad\ \Phi(r_{ij}). \qquad (1)$$

$\phi(r)$ はポテンシャル関数で分子や原子を扱う計算化学分野では次の Lennard-Jones (L-J) ポテンシャルがよく使われる．

$$\Phi(r) = 4\varepsilon\left[(\sigma/r)^{n} - (\sigma/r)^{m}\right]. \qquad (2)$$

ここでσ は粒子の直径，εはエネルギーの次元を持つ量を表す．L-J ポテンシャル関数について，計算化学等でよく使われる $m=6$，$n=12$ を用いる．

粒子 i に対するニュートンの運動方程式

$$m_i d^2\mathbf{r}_i(t)/dt^2 \ (=m_i\mathbf{a}_i) = \sum_j \mathbf{F}_{ij}\ (t) \qquad (3)$$

m_i：粒子 i の質量，$\mathbf{a}_i$：加速度

に基づき $\mathbf{a}_i$ を求め，Verlet 法を用いて粒子の位置や速度を更新する：

$$\mathbf{r}_i(t+\Delta t) = 2\,\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \mathbf{a}_i(t)\,\Delta t^2$$

$$\mathbf{v}_i(t) = \{\,\mathbf{r}_i(t+\Delta t) - \mathbf{r}_i(t-\Delta t)\,\}/2\Delta t. \qquad (4)$$

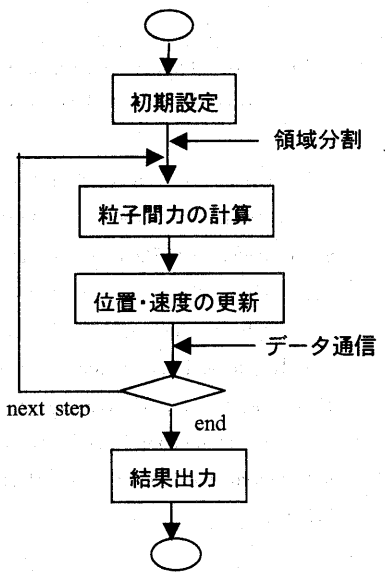

図 2　MD 法計算の流れ

これらの計算の精度を保つためには，全系のエネルギーが保存される必要がある．

$$H = 1/2 \sum_i m_i \mathbf{v}_i{}^{\mathrm{T}} \mathbf{v}_i + \sum_{i,j\,(i<j)} \Phi(r_{ij}). \qquad (5)$$

ここで H はハミルトニアンと呼ばれる量である．

2.2 並列処理概要

ここで扱う粒子は，面心立方格子状で1つの格子当り平均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3参照)．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．今回は主に z 軸で分割し，カットオフ距離は計算条件によって $2a$〜$4a$ (a は格子サイズ)にとる．空間分割法には，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という隠れた利点もある．

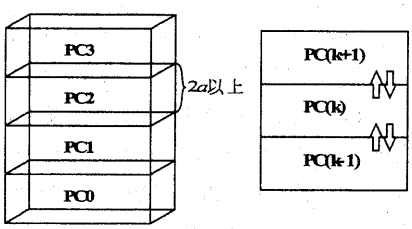

図3　計算領域の割当てとPC間の通信

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，例えばz方向分割の場合下記の式より決定する．ただし，必ずカットオフ距離以上の領域を担当させる．領域数は必ずしも整数でなくても良い．

領域数 =(当該 PC の速度) / (全 PC 速度の和)
×(z 方向格子数)

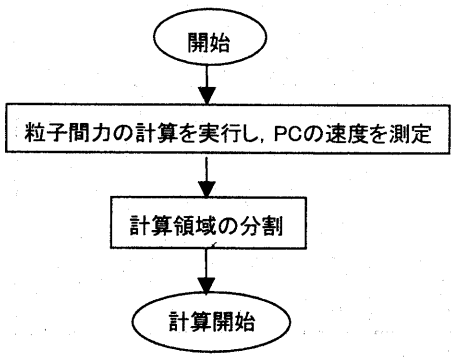

図4　実測による自動負荷分散

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 または 8×8×40 組み合わせ，粒子数がそれぞれ 5120 と 10,240 の比較的小規模のものを用いる．性能測定には 300 ステップ計算する．実験に用いる PC の性能は A〜E（表1）の5種からなる．D はデュアル PC で，D_s は 1 プロセッサ使用のとき，D_d は 2 プロセッサ使用マルチスレッド処理によるものを示す．E はノートパソコンである．

ここでは，z 方向による分割を採用し，2通りの構成でテストする．1つはほぼ均一な性能の PC からなる構成で，クロック周波数の高いものを最大 6 台使用する．8×8×20 の系で計算し，クロック周波数による空間分割を用いる（ケース1）．もう1つは，A〜E までの5種の PC を用いた非均一構成で，デュアル PC を含む最大 12 台（13 CPU）を使用する．8×8×40 の系を用い，実測による自動負荷分散とする（ケース2）．PC の構成と z 方向の領域割当量を表2，表3に示す．

表1　各PCの性能

P	総	CPU		Cache (kB)		Bus
C	数	個数	ｸﾛｯｸ (Mhz)	1次	2次	Speed
A	4	1	1100	128	256	Full
B	2	1	850	128	512	2/5
C	4	1	450	32	512	Half
D	1	2	450	32	512	Full
E	1	1	500	32	128	Full

表2　PCの構成と領域の割合（ケース1）

台数	PC構成	z 方向領域の割合
1	A	20
2	2A	10.0×2
3	3A	6.7: 6.6×2
4	4A	5.0×4
5	4A+B	4.2×4:3.2
6	4A+2B	3.7×2:3.6×2:2.7×2

表3　非均一PCの構成と領域の割合（ケース2）

CPU	PC構成	z 方向領域の割合
1	A	40
2	2A	19.9:20.1
4	4A	10.0×2:9.9:10.1
6	4A+2B	7:7.2×3:5.7×2
8	4A+2B+2C	6.2×2:6.3:6.4:4.9:5.2:2.5×2
10	4A+2B+4C	5.5:5.6×2:5.7:4.4×2:2.2×4
11	4A+2B+4C +E	5.1:5.3×3:4.1:4.2:2.1×4:2.3
12	4A+2B+4C +D_d	5.0×2:5.2×2:4.0:3.9:2.0×4:3.7
13	4A+2B+4C +D_d+E	4.7×3:4.8:3.8:3.7: 2.1:2.0×3:3.5:2.0

D_s：シングルCPU，　D_d：デュアルCPU

表3 非均一PCの構成と領域の割合

台数	PC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_s	7.9:5.9:3.1:3.1
5	2A+B+C+D_s	5.7:5.6:4.3:2.2:2.2
6	2A+2B+C+D_s	4.5×2:3.5×2:2.0:2.0

各ケース，各 CPU 台数での計算時間を測定し速度向上率を算出し，その結果を図5～6に示す．

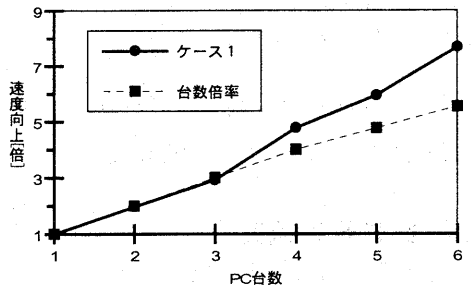

図5　均一PCクラスタによる速度向上（ケース1）（クロック周波数による分散）

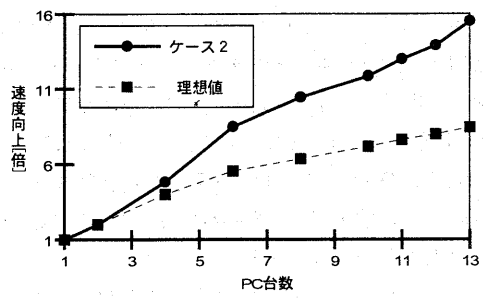

図6 非均一PCクラスタによる速度向上（ケース2）（実測による自動負荷分散）

均一な構成（ケース1）では4台以上のとき，理想値（クロック周波数で換算した台数効果）以上の速度向上倍率が得られた．理想値5.5倍に対し7.8倍と40%の向上となった（図5）．これは，カットオフの判断に要する演算が隣接PCの範囲に収まり，隣接PCへの領域割当量が台数と共に小さくなるためである．

非均一な構成（ケース2）においても4CPU以上ではカットオフ効果が見られた．PCの数を12台まで増やして13 CPU用いた計算では，理想値8.4倍に対し実際の速度向上倍率は14.7倍で70%も上回る（図6）．PC構成4A＋2B（理想値は5.5倍）のもとでz =20のときとz =40のときの速度向上倍率を比べると，7.8から8.5倍に上昇した．計算モデルが大きい方がカットオフによる並列処理効果は大きい．

遠く離れた粒子の粒子間力への影響は非常に小さいのでカットすることはMD法では常識である．通常は，2粒子間の距離（r）を周期周期境界条件のもとでまず求め，if文でカットオフするかどうかの

判断をしている．カットオフされる粒子数が多いと粒子間力の計算負荷は小さくなり，カットオフの判断に要する計算負荷が相対的に大きくなる（図7）．空間分割法では時間ステップごとにカットオフされる粒子群を一括して前もって除去できるというプラスの副作用があり，並列計算に係わらず演算量の減少が期待できる．カットオフ距離は計算モデルによって異なり，大きくし過ぎると (5)式のハミルトニアンが保存されなくなる．今回のモデルでは，粒子の速度が小さいので（ε=1），カットオフ距離は $2a$ でも良い．

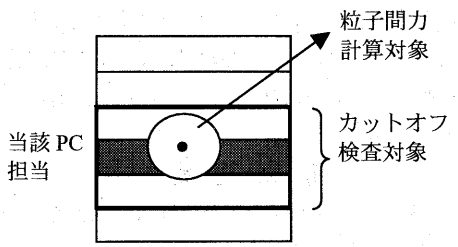

図7　カットオフによる演算減少

4．各種数値実験の考察

4.1　自動負荷分散による数値実験

図8では，実測による自動負荷分散を採用した場合の効果をクロック周波数で分散した場合と比較して示している．計算は表4の非均一な PC による構成で1～6台使用し，z =20 の系を計算対象としている．自動負荷分散においては速度向上倍率が約7％増加した．これは単なる倍率の差に留まらず，非均一 PC クラスタではそれぞれの PC のクロック性能等をユーザが前もって把握できないことが多い上，演算性能はキャッシュ効果によっても大きく変わるので，目的とするシミュレーションの演算の核となる部分の演算速度を実測する方法の方がより現実的，かつ有効であると言える．

4.2 マルチスレッド処理による数値実験

2プロセッサ搭載した PC（D_d）での並列化の実験を行う．計算モデルは 8×8×20 の系を用いる．共有メモリのために通信の必要がないものの，並列処理のオーバヘッドによって，シングル CPU（D_s）計

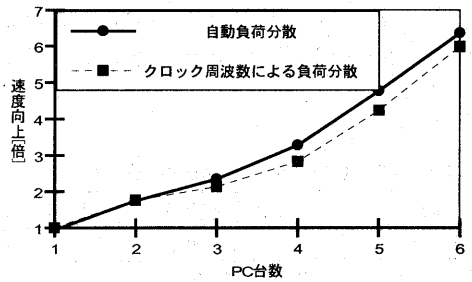

図8　非均一 PC クラスタ自動負荷分散による効果

表5　マルチスレッド処理による速度

	計算時間 [秒]	速度向上
D_s	3303.6	1.0
D_d	1706.3	1.9

算の 1.9 倍と若干遅くなった（表5）．この場合のオーバヘッドとしては，PC 上のキャシュメモリ効果の減少，スレッド制御，特にスレッド立ち上げに要する時間等が考えられる．MD 法ではデータ読込みと書込みのメモリ競合は生じない．

4.3　y,z 軸2方向分割

ここでは，8×8×40 の計算モデルを対象とし，y,z 軸2方向を2×2分割して4PCを用いて計算した．その結果，表6に示すとおり速度向上は台数効果（4.0 倍）を下回っている．当然のことながら1つの方向に対して3分割以下ではカットオフによる演算量減少の恩恵はない（図7）．2方向分割では4×3で 12 分割して初めてカットオフ効果が得られるはずである．

空間分割による並列計算法では，タイムステップごとに各計算領域の粒子リストの更新が必要となる．2方向分割においては，周期境界条件の元ではダミー領域を入れて近隣8領域への粒子移動を考慮する必要があり（図9），粒子リスト更新のための情報交換や粒子の位置や速度についてのデータ交換が多く発生する．この負荷は，z 方向の空間分割が上下2領域とのデータ交換であるのに対し，2方向分割では4倍に増加する．このように，粒子リスト管理のオーバヘッドが大きいことも速度向上

を妨げる理由となっている．

一方，z 軸方向のみの分割では4PCによる並列計算ではカットオフ効果が顕著で，4倍を上回る速度向上となっている．

表6　4PC並列による計算速度（Aタイプ）

構成	計算時間[秒]	速度向上
A:　シングル	12309.1	1.0
4A: y,z分割 2×2	3855.3	3.2
4A: z軸分割　4	2553.9	4.8

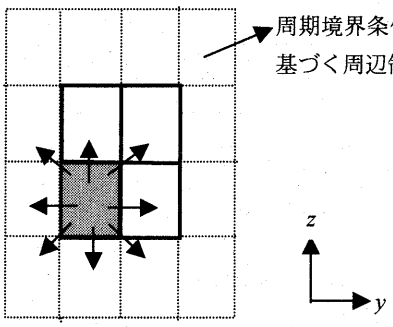

図9　2方向分割における近隣8領域

5．まとめ

(1) 通常業務で最もよく利用されているWindows環境において非均一PCからなるクラスタを構成し，WMPI通信ライブラリを用いた並列処理が実現できた．

(2) このとき，カットオフの導入による隠れた利点により，理想値（クロック周波数換算の台数効果）以上の速度向上が得られた．12 PC（13CPU）では理想値8.4倍に対し70%増の14.7倍の速度向上を得た．

(3) 空間分割法では，カットオフの判断に要する演算が隣接PCの範囲に収まり，隣接PCへの領域割当量が台数と共に小さくなるためである．カットオフされる粒子数が多い計算モデルにおいては粒子間力の計算負荷は非常に小さくなり，カットオフの判断に要する計算負荷が相対的に大きくなることが理由である．空間分割法では，時間ステップごとにカットオフされる粒子群を一括して前もって除去できるというプラスの副作用があり，並列計算に係わらず演算量の減少が期待できる．

(4) 非均一PCクラスタでは，実測による自動負荷分散で行うと良い結果が得られた．

(5) 2 CPU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1 CPUの約1.9倍であった．

(6) 1方向空間分割の方がカットオフによる演算量の減少のチャンスが大きい．これに加えて，1方向分割の方が空間分割法に特有の粒子リスト管理のオーバヘッドが小さいことも利点となる．結局，同じ空間分割数であれば，1方向分割の方が並列化による速度向上効果が大きいと言える．

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] 西村，中村，池口，清水：分子動力学法シミュレーションの並列化における分割法の一考察，情報処理学会研究報告，99-HPC-75, pp.61-66 (1999).

[3] 林亮子，堀口進：固定分配セルを用いた動的負荷分散法による並列分子動力学法シミュレーション，情報処理学会論文誌，Vol. 40, No.5, pp.2152-2162 (1999).

[4] 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.

[5] 蔡，安室，李，肖：Linux PCクラスタを用いた並列粒子シミュレーション，情報処理学会論文誌，Vol. 42, No. SIG 12 (HPS 4), pp.124-131 (2001).

[6] Critical Software: WMPI
http://www.criticalsoftware.com/wmpi/home/index.html