超並列計算機CP-PACSにおける大規模分子動力学法シミュレーション

松原正純[†] 板倉憲一[†,☆] 朴泰祐[†]

本論文では，超並列計算機CP-PACS上で分子分割法および空間分割法による分子動力学法（MD）シミュレーションを実行し，その性能評価ならびに両並列化手法の比較検討を行う．広域通信に強いというCP-PACSのネットワーク特性を活かすことにより，両並列化手法において負荷分散を考慮したデータと処理の分割が可能である．さらに，CP-PACSにおける分子分割法によるMDシミュレーションの問題点であったメモリ容量の制約に関しても，空間分割法を用いることによりこれを克服し，大規模なシミュレーションも可能であることを示す．実測の結果，対象とする分子数と必要メモリ量の関係から，最適な問題分割アルゴリズムがノード台数と分子数に依存して決まることも明らかになった．この結果，64 MB/ノードという比較的少量のメモリで，256ノードを用いて6700万分子の液体アルゴンのシミュレーションが約130秒/ステップで行えることが分かった．

Molecular Dynamics Simulations on CP-PACS

MASAZUMI MATSUBARA,[†] KEN'ICHI ITAKURA[†,☆] and TAISUKE BOKU[†]

In this paper, we introduce Molecular Dynamics (MD) simulations based on particle decomposition method and spatial decomposition method on CP-PACS massively parallel processor. By utilizing CP-PACS' powerful wide area communication feature, we can efficiently implement MD simulations with spatial decomposition method, realizing natural load balancing based on cyclic mapping. In addition, we show this method is also effective from the view point of required memory amount compared with the particle decomposition method. It is shown that the suitable method can be determined by the number of molecules and available memory capacity. Using the spatial decomposition algorithm, we can handle 67 million molecules problem with 256 nodes each of which has only 64 MB of main memory, in speed of about 130 sec/stcp.

1. はじめに

分子動力学法（Molecular Dynamics：以下MD）シミュレーションとは，物理法則に従って運動する粒子の集合体が位相空間中に描く軌跡を計算することにより，実験物理学などでは得られない微視的な物質の性質を調べる手法である．これまでも様々な並列計算機上でMDシミュレーションの研究がなされてきたが，それは$10^6 \sim 10^7$程度の分子数で実時間にしてたかだか数十ナノ秒のシミュレーションであった．しかし，実際には10^9程度の分子について数秒のシミュレーションが行えることが望まれている．そこで本研究では，MDシミュレーションの並列化と高速化について，超並列計算機CP-PACS（Computational Physics by Parallel Array Computer System）に適した方法を考察し，実測による実験を行う．その評価結果から，CP-PACSのように大域的通信において高い性能を発揮する並列計算機上で，MDシミュレーションを効率的に行う手法について検討する．

† 筑波大学電子・情報工学系
Institute of Information Sciences and Electronics, University of Tsukuba
☆ 現在，筑波大学計算物理学研究センター
Presently with Center for Computational Physics, University of Tsukuba

2. 分子動力学法

以下では本研究におけるMDのシミュレーション方法を述べる．

2分子間に働く力は，簡略化したポテンシャル関数によって与える．ここでは式(1)で表されるレナード・ジョーンズの2体間ポテンシャル[1]を用いる．

$$\phi(r) = 4\varepsilon\left\{\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right\} \quad (1)$$

rは分子間距離，ε，σは物質によって決まる定数である．N分子でのシミュレーションでは，$(N-1)$

個の分子対について相互作用の和をとることで各分子に加わる力が求められる．

MD シミュレーションでは，分子数 N が大きくなると分子対の数が急激に増大し，特に工夫を施さなければ，全実行時間の大半が相互作用の計算に使われてしまう．そこで，分子間相互作用の計算コストを縮小するために，精度を確かめたうえで特定の分子間距離以上の相互作用をカットする手法を用いる．

レナード・ジョーンズ・ポテンシャルのように，分子間距離が増大すると相互作用の大きさが急速に 0 に近付くポテンシャルでは，固定距離でのカットが有効となる．式 (1) によると，$r = 2.5\sigma$ のときに約 $-1.63 \times 10^{-2}\varepsilon$ となり十分 0 に近付くので，ここではカットオフ距離として 2.5σ を用いる．

式 (1) を用いてニュートンの運動方程式を解くわけだが，実際には式を差分化し，短い時間刻み幅 Δt ごとに運動方程式を解く．本研究では，次の式 (2)，(3) で表される Verlet の速度形式[1] と呼ばれるアルゴリズムを用いる．

$$r_i^{n+1} = r_i^n + v_i^n \Delta t + \frac{1}{2m} F_i^n (\Delta t)^2 \qquad (2)$$

$$v_i^{n+1} = v_i^n + \frac{1}{2m}(F_i^{n+1} + F_i^n)\Delta t \qquad (3)$$

r_i^n，v_i^n，F_i^n はそれぞれステップ n における分子 i の位置，速度，力を表す．初期値 r_i^0，v_i^0 を与えて，時間発展的に計算を繰り返すことにより，各ステップにおける位置と速度が次々に求められる．

シミュレーションは単純化のために立方体のドメインを想定してその中で行う．その際，ドメインの壁と分子との相互作用を無視できるように，周期境界条件を用いて擬似的に無限に大きい体積空間を作る．また，分子間の相互作用は，最近接鏡像法に従って計算する．

3. CP-PACS

本章では，本研究に用いる超並列計算機 CP-PACS について説明する．

CP-PACS[2] は MIMD 処理方式の分散メモリ型並列計算機で，2048 台の PU（計算専用プロセッサ）と 128 台の IOU（入出力プロセッサ），計 2176 台のノードから構成されている（**図 1**）．

ノードプロセッサには，Hewlett Packard 社の PA-RISC1.1 アーキテクチャに基づく RISC プロセッサに PVP-SW（Pseudo Vector Processor based on Slide-Windowed registers）機構を付加したものを使用している．PVP-SW はキャッシュを用いずにメモリアクセス・レイテンシを隠蔽する方法を提供し，擬似的なベクトル処理を可能にする．このプロセッサはクロック周波数は 150 MHz で動作し，理論ピーク性能は 300 MFLOPS である．主記憶容量は 64 MB である．ライトスルー/ダイレクトマップ方式の 2 レベルキャッシュを用い，容量は 1 次が命令用・データ用ともに 16 KB，2 次が命令用・データ用ともに 512 KB である．

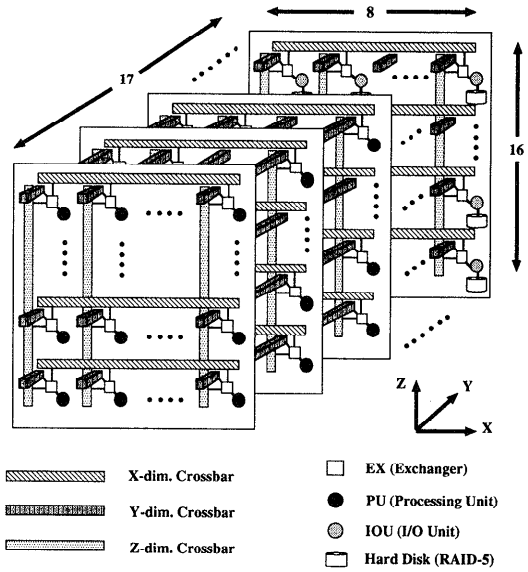


図 1 CP-PACS の構成図
Fig. 1 Overview of CP-PACS architecture.

ノード間の相互結合網には，3 次元ハイパクロスバ・ネットワーク（Hyper-Crossbar Network：以下 HXB）を採用している．3 次元 HXB は 3 次元直交座標上の各格子点にノードを配置し，各次元方向のノードをクロスバスイッチ（XB）で結合したネットワークである．ノード番号が 2 次元以上異なるノード間では，エクスチェンジャ（EX）と呼ばれる小規模なクロスバスイッチにより，複数の次元の XB を経由してデータを交換する．また，メッセージ転送は wormhole ルーティングを採用しており，ネットワーク中でのメッセージ間のデッドロックを回避するため，固定ルーティング方式を用いている．ネットワークの最大転送スループットは 1 リンクあたり 300 MB/sec である．HXB では，隣接転送だけではなく，多くのパターンの一斉転送についても無衝突で通信できる[3]．

CP-PACS のノード間通信プロトコルは，リモート DMA 転送と呼ばれる．これは，送信側・受信側ともにユーザの仮想アドレス空間の一部を物理メモリ上に固定的に割り当てておき，それらのメモリ間でデー

タ転送を行う高速通信方式である．異なったノード上のユーザ空間どうしで直接データ転送を行うため，カーネルとユーザ空間との間でのデータコピーが発生せず（いわゆるゼロコピー通信），ネットワークのスループットを最大限に活かすことができる[3]．

4. 並列化手法

本章では，MDシミュレーションの並列化手法について述べる．並列化の方法としては，処理の分割方法から分子分割法と空間分割法の2つに大別できる．以下にそれぞれの概念を説明する．

4.1 分子分割法

分子分割法では，分子の物理的な位置を考慮せず各ノードに均等な数の分子を配置する（図2）．各ノードは自分が持つ分子についての運動を計算する．ノード間での分子の移動はない．

分子分割法で必要となるノード間通信は，主に以下の1種類だけである．

- 相互作用計算時の自分以外のノードが持つ分子の位置情報の参照

分子分割法では，位置にかかわらず分子をノードに分配することができ，さらにシミュレーション中にも各ノードで処理を行う分子数は固定される．よって，計算負荷を自然に分散することができる．

ただし，この分割方法では自ノードが持つ分子の近隣分子がどのノードにあるか特定することができないため，分子間相互作用のカットオフを行うためには全ノード間でデータ交換を行う必要がある．

4.2 空間分割法

空間分割法は，ドメインをメッシュ状に分割して各ノードに割り当てる方法である．各ノードは割り当てられた部分空間（以下セル）の中に存在する分子についてのみ，運動の計算を行う．そして計算した結果，分子がセル外に飛び出した場合は，その2つのセル間，すなわち2ノード間で分子の受渡しが行われる．

空間分割法で必要となるノード間通信は，主に以下の2種類に分けられる．

- 相互作用計算時の自分のセルの外にある分子の位置情報の参照
- 分子がセルを出入りすることによる分子の受渡し

空間分割法は，セルのノードへのマッピング方法により，さらにブロック分割法およびサイクリック分割法の2通りに分けることができる．

4.2.1 ブロック分割

ブロック分割とは，各ノードが1つの領域の情報しか持たない空間分割方法である（図3）．

ブロック分割法の場合，セルの1辺の長さを分子間相互作用のカットオフ距離より長くすれば，相互作用計算時の通信を大幅に限定することができる．特に，メッシュ型のノード間ネットワークを持つ並列計算機や，HXBを持つCP-PACSでは，3次元空間をそのままマッピングすることができるので，どちらの通信も隣接ノード間通信となり，非常に有効である．

しかし，ブロック分割ではノードごとの計算負荷が変動する可能性が指摘されている[4]．特に低温，低密度の物理状態のシミュレーションを行った場合，分子が空間で局所的に集中する可能性がある．したがって，空間分割法ではいかにして動的負荷分散を行うかが問題となってくる[5]．

4.2.2 サイクリック分割

サイクリック分割では，ドメインをより細かく分割して図4のようにサイクリックにセルをノードに割り付ける．すなわち，各ノードは複数セルの情報を受け持つことになる．空間を十分小さいセルに分割すれ

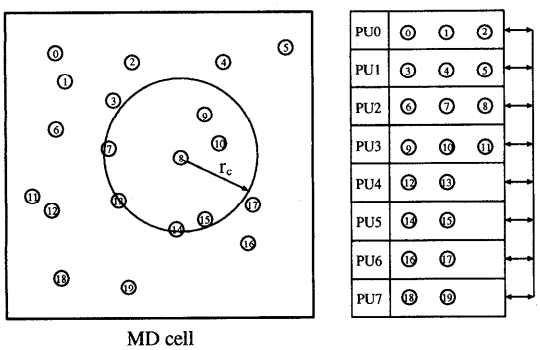


図2 分子分割法
Fig. 2 Atom-decomposition method.

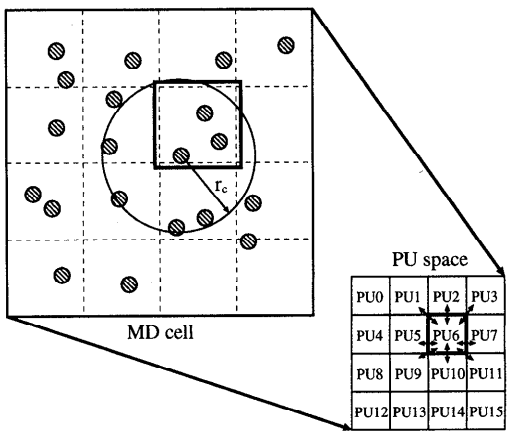


図3 ブロック分割による空間分割法
Fig. 3 Space-decomposition method by block mapping.

PU0	PU1	PU0	PU1
PU2	PU3	PU2	PU3
PU0	PU1	PU0	PU1
PU2	PU3	PU2	PU3

図 4 サイクリック分割による空間分割法
Fig. 4 Space-decomposition method by cyclic mapping.

表 1 各並列化手法の特徴
Table 1 Characteristics of each parallel method.

並列化手法	長所	短所
分子分割法	負荷分散	通信パターン（広域） 通信領域サイズ（大）
空間分割法 （ブロック）	通信パターン（隣接） 通信領域サイズ（小）	負荷分散
空間分割法 （サイクリック）	負荷分散	通信パターン（広域） 通信領域サイズ（大）

ば，たとえ分子が局所的に集中していたとしても計算負荷の分散が可能であると考えられる．したがって，サイクリック分割はブロック分割の欠点を補うことができる．

ブロック分割の場合は，セルがある程度大きければ，分子間相互作用計算時に必要となる通信を隣接転送のみに限定することができる．しかし，サイクリック分割ではセルが細かくなるため，相互作用の計算には，隣接するセルだけではなく，より遠方にあるセルの情報も必要になる．つまり，大域的な転送が必要となる．

4.3 各並列化手法の比較

各並列化手法の特徴をまとめると，**表 1** のようになる．この中で，空間分割法（ブロック分割）のみ計算負荷の不均衡が問題となっている．この問題については，これまで動的に負荷分散を行うための手法がいくつか提案されているが[5]，それらを用いても十分な負荷分散を行うことは難しいのが現状である．

一方，残りの 2 手法は，大域的な通信パターンが必要であることが問題となっている．しかし，本研究で用いる超並列計算機 CP-PACS は広域転送に強いため，その性能を十分に活かすことによりこの問題を解消することができる．

そこで本研究では，分子分割法および空間分割法（サイクリック分割）を用いた並列化を行う．以降では，両手法の CP-PACS 上での実現方法について述べる．

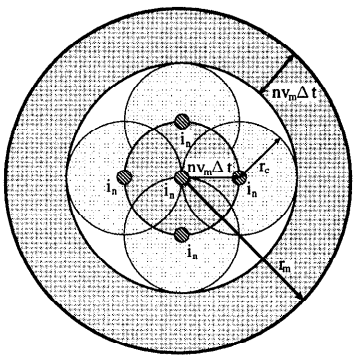


図 5 bookkeeping 法の概念図
Fig. 5 Concept of bookkeeping method.

5. 分子分割法の実現方法

分子分割法による MD シミュレーションでは，各ステップで以下の処理を行う[6]．

(1) 分子間相互作用の計算
(2) 差分方程式の求解

分子間相互作用の計算には，他ノードが持つ分子の位置情報が必要となるため，全ノード間でデータを交換する．HXB は広域通信に強く，特に Butterfly Collection アルゴリズムによる全対全転送は無衝突でメッセージを転送することができるため，この分子の位置情報の交換を非常に効率良く行える[3]．また，分子間相互作用の計算では，カットオフ距離以遠の分子間相互作用を無視するので，ある分子とそれ以外の全分子との位置関係を毎ステップ調べる必要がある．そこで，bookkeeping 法と呼ばれる高速化方法を適用する[7]．

図 5 に bookkeeping 法の概念図を示す．まず，n ステップ内に注目する分子 i がどの範囲に動く可能性があるかを考える．一般化するために，シミュレーション中の全分子における最大速度を v_m とすると，n ステップ間に分子 i と相互作用を及ぼす可能性のある分子が最初のステップにおいて存在する範囲は i_0 から，

$$nv_m\Delta t + r_c + nv_m\Delta t = 2nv_m\Delta t + r_c \qquad (4)$$

の半径の球内にある，と限定することができる．Δt は時間刻み幅を，r_c はカットオフ距離を表す．これを r_m とする．そこで，各分子について半径 r_m の球内にある分子番号をテーブルに記録しておけば，その時点から n ステップの間はテーブル中にある分子だけを対象にカットオフ判定を行えばよく，判定のための距離計算をある程度絞ることができる．そして，そのテーブルは n ステップごとに書き換えられることになる．

このbookkeeping法は，テーブルのサイズが大きくなるとキャッシュミスを引き起こし，RISCプロセッサ上では性能が低下する場合がある．しかし，CP-PACSではPVP-SW機構を有効に利用することにより，この欠点を克服でき，効果的に性能向上を図ることができると考えられる[8]．

6. 空間分割法（サイクリック分割）の実現方法

サイクリック分割を用いたMDシミュレーションの各ステップでは，以下の処理を行う．

(1) 分子間相互作用の計算
(2) 差分方程式の求解
(3) セル間での分子移動

この中でも特に分子間相互作用の計算，そしてセル間での分子移動が実行時間の大部分を占め，それぞれノード間通信を必要とする．本章では，この2つの処理で行う通信について述べる．

ところで，ブロック分割では各ノードは1ブロック分の分子を担当し，近隣のブロックに存在する分子の情報さえあれば相互作用の計算を行える．一方，サイクリック分割では各ノードは複数のセルを持つことになり，また分子間相互作用の計算には遠方のセルを必要とするため，場合によっては全分子の位置情報を取得する必要がある．したがって，サイクリック分割では，ブロック分割に比べて通信量および通信領域が増大することが予想される．そこで，本研究では，図4の太枠で囲った部分（以下サブドメイン）を単位とし，その単位ごとに処理することにより通信領域を減らす．

6.1 分子間相互作用の計算

各ノードが持つ分子の位置情報の交換は，1対1転送を複数回繰り返すことによって行う．通信は広域に及ぶが，CP-PACSの優れたネットワーク特性により，そのコストは比較的小さく抑えられると考えられる．

また，サブドメイン単位で処理を行うので，各処理ごとで必要となる通信領域を再利用することで通信領域サイズも小さく抑えることができる．

6.2 セル間での分子移動

異なるセルへの分子移動を行うために，1対1転送を用いて分子を移動することにする．しかし，通常MDシミュレーションの時間刻み幅は十分に小さいため，ほとんどの分子移動は隣接転送だけで済む．

つまり，3次元空間の場合は隣接する26セルに対して1対1転送を繰り返すことでそのセル外に飛び出した分子を移動することができる．この通信をサブドメインの数だけ繰り返せば，分子移動がすべて完了する．

7. コストの見積り

本章では，上記の実装方法による両並列化手法のコストについて考察する．

両手法の内部処理時間・通信時間の規模および通信領域サイズは**表2**のように見積もることができる．なお，内部処理時間の大部分はポテンシャルのカットオフ計算に費やされるので，ここでの内部処理時間の見積りはカットオフ計算にのみ注目する．Nは総分子数，Pはノード台数，そしてCは総セル数である．

分子分割法では，ステップごとに各分子は他の$(N-1)$分子とのカットオフ計算を行う必要がある．bookkeeping法を用いているので数ステップに1回全分子とのカットオフ計算を行えばよいが，計算量のオーダーとしては$O(N^2/P)$で変わらない．一方，空間分割法はカットオフ距離に従ってカットオフ計算を行うべきセルおよびセル数が一意に決まる．分子密度が同じならば，1セル内の平均分子数はセル・サイズによって固定なので，1分子あたりのカットオフ計算量は総分子数によらずつねに一定である．したがって，カットオフ計算量は$O(N/P)$である．

通信時間については，分子分割法はButterfly Collectionによる全対全転送を行っているため，全転送量は分子数Nに比例し，転送回数は$log_2 P$となる．一方，空間分割法は，分子情報の交換およびセル間での分子の受渡しで両方とも1対1転送を用いており，通信時間のオーダーは$O(N/P)$となる．

通信領域サイズ，すなわち受信バッファのサイズについては，分子分割法では全分子の位置情報を各ノードが保持する必要があるため$3N(DoubleWord)$と非常に大きくなる．一方，空間分割法（サイクリック分割）では，1セル内の分子数は刻々と変わるため，どのセルに対してもそのセル内の分子の情報がすべて入りきるだけの領域を確保する必要がある．そこで，必要十分な量の通信領域を確保するためのパラメータa, bを導入して，領域サイズを調整する．1セルあた

表2 計算量・通信時間・通信領域サイズの見積り
Table 2 Estimation of the computation time, the communication time and the size of communication buffer.

	分子分割法	空間分割法
計算量	$O(N^2/P)$	$O(N/P)$
通信時間	$O(log_2 P + N)$	$O(N/P)$
通信領域	$O(N)$	$O(NP/C)$

りの平均分子数と同数の分子が入っているセルが必要とする通信領域サイズを1としたとき，ポテンシャル・カットオフ計算時にともなう通信で必要となる領域サイズにかかる係数を a とする．同様に，分子移動時にともなう通信で必要となる領域サイズにかかる係数を b とする．a, b を用いると，通信領域サイズは $(3aP+162b)N/C(DoubleWord)$ と表される．シミュレーションの条件（分子配置の分散状況，セル・サイズなど）により，a は約1.3～10，b は約0.2～2.0の範囲で設定する．a, b が最悪の場合を想定しても，空間分割法（サイクリック分割）はサブドメイン単位で処理を進めるため，全体としての通信領域は分子分割法よりもかなり小さい．

8. 性能評価および考察

本研究では，分子分割法および空間分割法（サイクリック分割）によるMDシミュレーションプログラムを作成し，CP-PACSに実装した．プログラムはFortran90で記述し，通信はすべてリモートDMA転送を使用した．本章では，CP-PACS上に実装したMDシミュレーションの性能評価および考察を行う．

本測定では，単原子液体の液体アルゴンをエネルギー一定の分子動力学法でシミュレートする．特に断らない限り各測定結果は，無次元単位で時間刻み0.064，系の平均温度2.53，分子密度0.636，そしてカットオフ距離 2.5σ という条件下での結果である．時間計測にはプロセッサ内のクロックカウンタを用いているため，μsec 以下の精度で測定が行える．また，今回の測定では，最大256台のノードを用いる．

以降では，まず始めに分子分割法・空間分割法（サイクリック分割）の各々について評価する．そして，最後に両手法の性能比較を行う．

8.1 分子分割法の評価

8.1.1 テーブル更新の間隔

本項では，bookkeeping法におけるテーブル更新の頻度をどれだけに設定するのが最も効率的かを調べる．

256分子のシミュレーションを1～256ノードを用いて，近隣分子をリストアップする間隔（式(4)における n）を5・10・15ステップと変えた場合と，bookkeeping法を行わない場合との実行時間を比較する．

理論的な r_m の値，すなわち $2nv_m\Delta t+r_c$ の値と，実際の r_m の最小値とは異なった値となる．理論的な r_m の値は，分子が n ステップの間つねに最高速度で直線的に運動した場合のもので，現実的には最高速度以下の速度で非直線的な運動をするため，実際には r_m の値は理論値より小さくてもよい．そこで，間引きなしの結果と変わらない範囲で r_m を0.1単位でどこまで小さくできるか実験し，それによって求められた r_m の最小値を使用する．

表3 分子分割法における分子数256のときの実行時間
Table 3 Evaluation of the interval of table update on atom-decomposition method ($N=256$).

(msec)

ノード数	間引きステップ数			
	なし	5	10	15
1	15.737	11.255	11.434	12.708
2	11.312	7.350	7.473	8.433
4	6.481	4.108	4.208	4.832
8	3.447	2.167	2.210	2.498
16	1.797	1.212	1.211	1.359
32	1.016	0.722	0.713	0.785
64	0.637	0.489	0.480	0.512
128	0.440	0.373	0.368	0.389
256	0.359	0.319	0.322	0.328

各シミュレーションの1ステップあたりの実行時間を**表3**に示す．なお，表中で下線を引いたものがそのノード数における最も良い値である．

全体的なbookkeeping法の効果としては，ここでは1.1～1.4倍程度の性能向上として現れている．間引きステップ数が小さすぎると，r_m を小さくできるのでカットオフのオーバーヘッドが小さくなるが，テーブルの更新が頻繁に起こって効率が悪くなる．一方，ステップ数が大きくなりすぎると，テーブル更新の頻度は下がるが，r_m が大きくなりカットオフの効率が悪くなる．このように，間引きのステップ数と r_m はトレードオフの関係にあるといえる．分子数256の結果では，1ノードの実行時間を見ると $n=5$ の場合が最適なポイントになっているようである．ただし，256ノードでは粒度が小さくなりすぎて並列化効率が頭打ちになっており，間引きステップによる差はそれほど見られない．

今後の分子分割法の測定では，最適な n を用いた結果を示す．

8.1.2 並列化効率

次に，並列化効率について評価を行う．分子数256・864・2048で，ノード数を1～256としたときの1ステップあたりの実行時間を**表4**に，そのときの速度向上率のグラフを**図6**に示す．

速度向上率を見ると，$N=256$ の256ノードで頭打ちになっているもの以外は，2ノードを基本に見るとかなり良い速度向上率が得られている．しかし，1ノードから2ノードへの速度向上が低く，全体として

表 4 分子分割法における分子数とノード数の違いによる実行時間の変化
Table 4 Execution time on atom-decomposition method.

(msec)

ノード数	分子数 256	864	2048
1	11.225	50.208	172.778
2	7.350	33.402	98.918
4	4.108	18.926	62.293
8	2.167	9.771	30.268
16	1.211	5.048	15.463
32	0.713	2.688	8.005
64	0.480	1.560	4.227
128	0.368	0.979	2.389
256	0.319	0.741	1.538

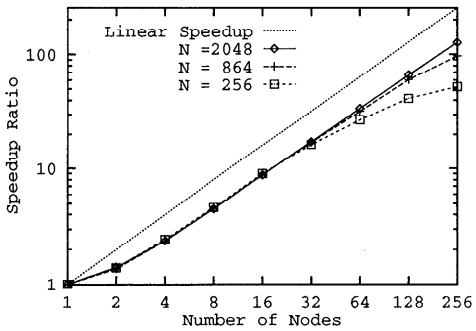


図 6 分子分割法における分子数の違いによる速度向上率の変化
Fig. 6 Speedup ratios on atom-decomposition method.

あまり良いように見えない．これはあるノード内にある分子とその他のノードにある分子との相互作用を計算するとき，逐次実行ならば 1 回で済む計算を，並列実行では各ノードで別々に計算するため，二度手間になっているためである．

8.1.3 実行時間の内訳

実行時間を内部処理時間と通信時間に分けて見てみる．256 ノード，256～442368 分子での測定結果を**表 5** に示す．

256 分子のシミュレーションでは通信時間が全実行時間の約 18%を占めているものの，分子数が増大するにつれて全実行時間に占める通信時間の割合は減少しているのが分かる．44 万分子のシミュレーションでは，通信時間は 1%にも満たない．これは，HXB ではこの Butterfly Collection を無衝突で行うことができるため，非常に効率の良い通信となっているからである[8]．

8.2 空間分割法（サイクリック分割）の評価

8.2.1 セル・サイズ

まず始めに，最も効率的なセル・サイズについて考

表 5 分子分割法における実行時間の内訳（256 ノード）
Table 5 Breakdown of the execution time on atom-decomposition method (256 nodes).

(msec)

分子数	内部処理	通信
256	0.261	0.058
864	0.627	0.114
2048	1.336	0.202
16384	38.110	2.642
131072	2149.178	17.521
186624	4326.037	24.518
256000	8123.920	32.589
340736	14399.651	43.557
442368	24152.269	73.214

表 6 空間分割法におけるセル・サイズの評価（$N = 10^6$）
Table 6 Evaluation of the size of cell on space-decomposition method by cyclic mapping ($N = 10^6$).

(sec)

分子数	Move all	Move comm	Force all	Force comm
256	0.031	0.024	11.085	0.288
32	0.053	0.034	1.833	0.254
4	0.305	0.186	3.631	2.053

察する．256 ノードを用いた 100 万分子のシミュレーションを行った結果，シミュレーション 1 ステップあたりの実行時間の内訳は**表 6** のようになった．表中の Move はセル間の分子移動，Force は分子間相互作用の計算に要した時間である．all とはその処理全体の実行時間（sec），comm はその処理中の通信時間（sec）である．この Force と Move を足したものが，シミュレーション 1 ステップに要する時間とほぼ等しい．また，各ノードはセル内の平均分子数が 256 のときは 16 セル，32 分子のときは 128 セル，4 分子のときは 1024 セルをそれぞれ担当する．

相互作用のカットオフ計算は，最終的には 1 分子に対しその分子のカットオフ範囲に入ったセル内の全分子とのチェックになるため，セルを大きくすると 1 セルあたりの分子数が増加してカットオフ計算量が増大する．よって，セル内の平均分子数が 256 の場合は，Force に時間がかかる．また，セルを大きくするとセル内の分子数に不均衡が生じる．したがって，負荷の軽いノードは内部処理が終わると他のノードよりも早く通信処理に入って，他のノードからのデータ転送を待つため，Force の通信時間が増大する．

反対に，セルを小さくすると 1 セル内の分子数は減り，負荷分散がしやすくなる．しかし，細かい通信が多いため，通信時間が増える．さらにセル間での分子移動数も増えるため，Move 全体の時間も増加する．

表 7 空間分割法における実行時間の内訳（256 ノード）
Table 7 Breakdown of the execution time on space-decomposition method by cyclic mapping (256 nodes).

(sec)

分子数	内部処理	通信
2048	0.003	0.032
16384	0.024	0.019
131072	0.185	0.048
1048576	1.603	0.290
8388608	12.947	2.271
67108864	107.609	19.025

これが 1 セルあたりの平均分子数が 4 の場合である．

今回の測定では，1 セルあたりの平均分子数が 32 の場合が最もバランスがとれており，全実行時間が最短となった．そこで以降の測定では，セル内の平均分子数が 32 になるようにセル・サイズを定める．

8.2.2 実行時間の内訳

シミュレーション規模により，内部処理，通信の割合がどのように変わるかを見るために，1 ステップの実行時間の内訳を調べる．**表 7** に 256 ノード，$2048 \sim 6.7 \times 10^7$ 分子での測定結果（sec）を示す．

表 7 より，トータル時間は線形に増えているのが分かる．これは，ノード台数が等しい場合は，内部処理時間，通信時間共 $O(N)$ だからである（表 2）．したがって，より大規模なシミュレーションについても容易に実行時間の見積りを立てることができる．

また，通信時間が全実行時間の 15%以上を占めているのが分かる．これはサイクリック分割にしたため，通信が細粒度になっていることが原因としてあげられる．

8.2.3 並列化効率

図 7 に，100 万分子のシミュレーションを 16～256 ノードで行ったときの速度向上率を示す．ここでは，16 ノードの場合の性能を 1 とする．図中 “Linear Speedup” で示される線は線形な速度向上を表し，“MD（SD）” は実測値を表す．

全体としては良い並列化効率が得られている．しかし，ノード台数が増えると若干並列化効率が落ちている．分子間相互作用およびセル間での分子移動時に 1 対 1 転送を行うが，ノード台数が少ないほどこのときの通信相手が重なる．このとき，同一ノードに対する通信は 1 回の通信で済むため，ノード台数が少ないほど通信に関して有利となる．同様に，通信相手が重なるということは，受信側のノードでは受信したデータを複数回参照するため，キャッシュが有効に働き，内部処理も効率良く行える．以上のことが，並列化効率が落ちる原因であると考えられる．ただし，通信相手が重ならないほど多数のノード台数になると，それ以上のノード台数では一定の並列化効率が得られるものと推測される．

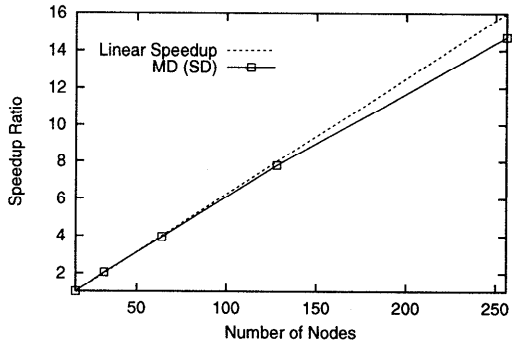


図 7 空間分割法における並列化効率
Fig. 7 Speedup ratios on space-decomposition method by cyclic mapping.

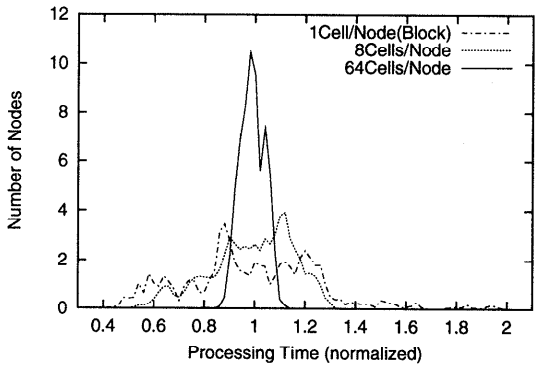


図 8 空間分割法における負荷分散状況
Fig. 8 Load-balancing on space-decomposition method by cyclic mapping.

8.2.4 計算負荷の分散

分子分割法ではつねに自然と計算負荷の分散が行えるのに対し，空間分割法（サイクリック分割）ではどの程度細かくセルに分けるかにより負荷分散の度合が違ってくる．そこで次に，サイクリック分割によりどの程度計算負荷を分散できているかを調べる．MD シミュレーションでは，低温低圧時に分子が局所的に集中することが知られているので[5]，本測定では温度 0.722，分子密度 0.256 とし，64 ノードで 16384 分子のシミュレーションを行う．また，シミュレーションは 2000 ステップ行い，十分に分子が偏っている状態になった 1600～2000 ステップについてステップごとに記録をとった．測定結果を**図 8** に示す．グラフの横軸は，1 ステップあたりの内部処理時間を平均内部処理時間によって正規化してある．

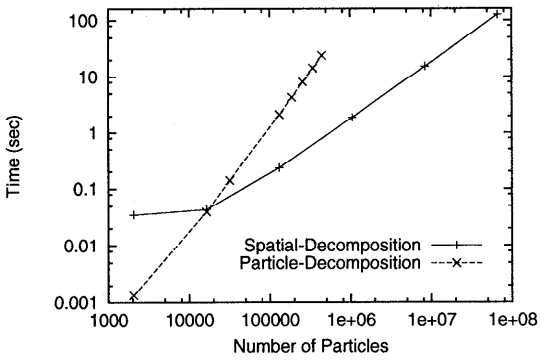


図 9 空間分割法と分子分割法の比較（256 ノード）
Fig. 9 Comparison of atom-decomposition method and space-decomposition method by cyclic mapping (256 nodes).

図 8 から，“1 Cell/Node” では平均内部処理時間の約 2 倍の時間を要するノードが存在するのに対し，“64 Cells/Node” は非常に良い負荷分散が行えているのが分かる．したがって，十分に細かいセルに分割したならば，サイクリック分割は良い負荷分散を行えることが実測によって示せた．

1 ステップあたりの実行時間は，“1 Cell/Node” は 1.2768 sec，“8 Cells/Node” は 0.2009 sec，“64 Cells/Node” は 0.2111 sec となった．カットオフ判定の計算量の観点から見ると，本測定で作成した MD シミュレーション・プログラムでは，セルあたりの平均分子数が 32 の場合が特に良い結果となる（表 6）．しかし分子が偏った状態のシミュレーションでは，さらに計算の負荷についても考慮する必要があることが本測定から明らかとなった．つまり，カットオフ判定の計算量と計算負荷の分散という 2 つの間でのトレードオフにより，最適なセル・サイズが決まる．

8.3 分子分割法と空間分割法の比較

最後に，分子分割法と空間分割法（サイクリック分割）との性能の比較を行う．

各分子数におけるシミュレーション 1 ステップの実行時間を**図 9** に示す．サイクリック分割による空間分割法では，サブドメインごとに通信を行うため分子分割法に比べて通信回数が多い．さらにシミュレーションする分子数が少ない場合，通信 1 回あたりの転送量が少なく，また内部処理時間も短いため，通信のオーバヘッドを隠蔽することができない．そのため，分子数が少ない場合は，空間分割法より処理が粗粒度である分子分割法の方が 1 ステップあたりの実行時間が短い．しかし，分子数が増えると分子分割法は急激に計算量が増大するため（表 2），16,000 分子のシミュレーション結果あたりから空間分割法の方が良い結果を示している．

さらに，分子分割法では $O(N)$ の通信領域サイズが必要なのに対して，空間分割法ではサブドメイン単位で処理を進めていくため分子分割法よりも少ないメモリ容量で済む．図中の 6.7×10^7 分子の場合は，第 7 の計算式に当てはめる（$a = 2$, $b = 0.5$, $C = 2097152$, $N = 67108864$）と，通信領域サイズは約 1 MB でしかない．よって，大規模なシミュレーションを行う場合には，メモリ消費量が少ない空間分割法の方が有効である．

MD シミュレーションは，用いられる分野や対象としている物質などによりそのシミュレーション規模は大小様々である．本測定の結果は，CP-PACS のような大域的通信に強い超並列計算機上で MD シミュレーションを行う場合に，そのシミュレーション規模によりどちらの並列化手法を用いればよいのかを示す指標になるものといえる．

8.4 他の計算機との比較

本研究で実装した MD シミュレーション・プログラムの CP-PACS 上における絶対性能評価の一例として，CM-5 でのシミュレーション結果[9] と比較する．ただし，完全に同じ条件で比較しうるデータはないので，本研究における実験結果からの類推により比較する．

64 ノードの CM-5 上では，17576 分子の液体アルゴンの MD シミュレーションを約 7 sec/step で行えるという研究結果が Hayashi らにより報告されている[9]．一方，同様のシミュレーションを 256 ノードの CP-PACS で行った場合，サイクリック分割による空間分割法および分子分割法の両者とも，約 0.05 sec/step で実行できることが図 9 より分かる．この問題を 64 ノードの CP-PACS で実行したとすると，図 6 および図 7 に示す速度向上率からの推察により，約 0.2 sec/step 前後で処理できるものと考えられる．なぜならば，問題サイズを変えずにノード数を 1/4 に落としているため，ノードあたりの処理粒度は大きくなっており，実行時間は 256 ノードの場合の 4 倍程度であって，それ以上かかることはないと考えられるからである．

よって，64 ノードでの比較では，今回の CP-PACS における性能は，CM-5 の約 35 倍であると考えられる．CP-PACS のノード・プロセッサのピーク性能（300 MFLOPS）と，CM-5 のノード・プロセッサのピーク性能（4.2 MFLOPS の SPARC マイクロプロセッサ＋ 32 MFLOPS のベクトルユニット 4 基）の差を考慮しても，本実装手法が非常に高い性能を実

現していることが分かる．これにより，本実装手法が CP-PACS の高いノード間通信性能を引き出し，効率的な MD シミュレーションが実現されているものと考えることができる．

9. おわりに

本研究では，超並列計算機 CP-PACS 上で分子分割法および空間分割法による分子動力学法シミュレーションを行うプログラムを作成し，その性能評価を行った．

その結果，CP-PACS の強力な通信性能を活かすことにより，両手法とも効率良く実行できることが分かった．分子分割法は，bookkeeping 法のテーブル更新間隔を適切に設定することにより，分子数が比較的少ない場合に特に有効な手法である．空間分割法では，通信領域を最小に抑えることができるため，分子分割法では実行できない大規模なシミュレーションも行えた．さらに，メッシュ/トーラス・ネットワークでは衝突が起きてしまうような大域的な一斉転送も CP-PACS の HXB 上では無衝突で行えるため，サイクリック分割を採用することができ，この結果計算負荷の分散も図れた．

CP-PACS は限られた計算物理学を対象とした計算機であるにもかかわらず，MD という対象外の科学技術計算に対しても有効であり，さらに複数の並列化手法を使い分けることができた．これは CP-PACS が強力な通信性能を持っていることが主因であり，CP-PACS のように大域通信に強い超並列計算機は様々な並列処理に有効であるといえる．

今後の課題として，

- シミュレーションから得られた物理量に対する検討
- より現実的な物質を対象にしたシミュレーション

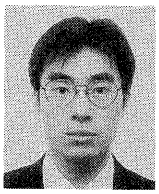
などがあげられる．

謝辞 CP-PACS を利用する機会を与えて下さった筑波大学計算物理学研究センターの関係者各位に感謝します．なお，本研究の一部は科学研究費補助（奨励（A）09780234）によるものである．

参考文献

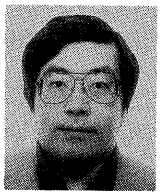
1) Heermann, D.W., 小澤 哲, 篠嶋 妥（訳）：シミュレーション物理学，シュプリンガー・フェアラーク東京 (1990).
2) 岩崎洋一，中澤喜三郎ほか：計算物理学と超並列計算機—CP-PACS 計画，情報処理, Vol.37, No.1, pp.10–42 (1996).
3) 松原正純，板倉憲一，朴 泰祐，中村 宏，中澤喜三郎：超並列計算機 CP-PACS のネットワーク性能評価，情報処理学会研究報告，HPC67-10, pp.55–60 (1997).
4) Beazley, D.M. and Lomdahl, P.S.: Message-passing multi-cell molecular dynamics on the Connection Machine 5, *Parallel Computing 20*, pp.173–195 (1994).
5) 林 亮了，堀口 進：並列分子動力学法シミュレーションにおける動的負荷分散法，並列処理シンポジウム JSPP'96 論文集，pp.81–88 (1996).
6) 服部正樹，松原正純，板倉憲一，朴 泰祐：超並列計算機 CP-PACS における分子動力学法シミュレーション，情報処理学会研究報告，HPC66-2, pp.7–12 (1997).
7) 田中 実，山本良一：計算物理学と計算化学，講談社 (1994).
8) 板倉憲一，安部井嘉人，松原正純，朴 泰祐，中村 宏，中澤喜三郎：超並列計算機 CP-PACS の基本性能評価，情報処理学会研究報告，ARC123-4, pp.19–24 (1997).
9) Hayashi, R. and Horiguchi, S.: Parallelized Simulation of Molecular Dynamics by Domain Decomposition Strategy, *Proc. World Congress on Systems Simulation*, pp.353–358 (1997).

(平成 10 年 8 月 31 日受付)
(平成 11 年 3 月 5 日採録)



松原 正純（学生会員）
昭和 48 年生．平成 8 年筑波大学第三学群情報学類卒業．平成 10 年同大学大学院工学研究科前期博士課程修了．現在，同研究科後期博士課程に在籍中．並列処理に関する研究に従事．

板倉 憲一（正会員）
昭和 44 年生．平成 5 年筑波大学第三学群情報学類卒業．平成 11 年同大学大学院工学研究科博士課程修了．博士（工学）．平成 11 年筑波大学計算物理学研究センターリサーチ・アソシエイト．並列計算機アーキテクチャ，並列入出力・可視化機構の研究に従事．



朴　泰祐（正会員）

昭和59年慶應義塾大学工学部電気工学科卒業．平成2年同大学大学院理工学研究科電気工学専攻後期博士課程修了．工学博士．昭和63年慶應義塾大学理工学部物理学科助手．平成4年筑波大学電子・情報工学系講師，平成7年同助教授，現在に至る．超並列処理ネットワーク，超並列計算機アーキテクチャ，ハイパフォーマンスコンピューティング，並列処理システム性能評価の研究に従事．電子情報通信学会，IEEE各会員．