

分子動力学法シミュレーションの並列化

林 亮子 堀口 進

北陸先端科学技術大学院大学 情報科学研究科

物理、化学の分野で用いられる数値シミュレーションの一手法、分子動力学法の並列化について述べる。本論文では、3次元のレナード・ジョーンズ系分子動力学法の並列シミュレーションについて述べる。この分子動力学法については、これまで種々の並列計算機上への実装法が提唱されている。ここでは単純分割法、2ステージ法や近傍空間参照法 (Coarse-grained cell) による並列シミュレーションアルゴリズムを並列計算機 CM-5 上に実装し、計算時間および高速化率の比較検討を行った。短距離相互作用の並列計算のうち、現在最も有効な方法として近傍空間参照法が挙げられる。本研究結果からも、並列化手法として近傍空間参照法が有効であることが明らかになった。

Parallelization of molecular dynamics simulation

Ryoko Hayashi Susumu Horiguchi

School of Information Science
Japan Advanced Institute of Science and Technology

This paper addresses parallel computing of molecular dynamics for simulation model of 3D short-range Lennard-Jones forces. Three parallel computation strategies are implemented on massively parallel computer, CM-5. The performance of three strategies; simple partition, 2-stage calculation and Coarse-grained cells, are discussed in detail. It is shown by parallel simulation that Coarse-grained cells method is one of the powerful strategy to simulate a number of particles.

1 はじめに

近年、計算機の高速化にともなって先端科学技術分野でのコンピュータ・シミュレーションの果たす役割が大きくなっている。ここでは、そのような数値計算の手法の一つ、分子動力学法の並列シミュレーションについて研究結果を報告する。

分子動力学法で扱われている分子の数は逐次処理で数千個から、非常に簡単な相互作用の場合で百万個程度である。本来物質はアボガドロ数 6.02×10^{23} 程度の分子の集まりなので、現在計算可能な物性は量的、質的に限られたものであり、並列コンピュータによる高速化が期待されている。そのため、ノード数、CPU、結合方式の面で様々な並列計算機を用いた研究が行なわれている。Beazley, Lomdahl [1] はメッセージ交換を用いた分子動力学法の並列シミュレーションに関する報告を行なった。この中で空間を分割して PE に割り当てる方法 Coarse-grained cells を用いて、 6.7×10^7 分子を計算している。また、Hedman [2]、Tamayo [3] 等は、Coarse-grained cells をベースに、さらに計算量を減らす手法を検討している。しかし、これらの研究では、異なった並列化手法に対する処理効率については十分に研究されていない。本論文では、短距離相互作用の分子動力学法シミュレーションを 3 種類の並列化手法を用いて計算時間や高速化率について検討した結果について報告する。

本論文の以下の構成を述べる。2 章では本研究で扱った分子動力学法に触れる。3 章では使用した計算機環境、並列処理の手法について述べる。4 章では実装によって得られた結果を示し、5 章で今後の課題について検討する。

2 分子動力学法について

分子動力学法とは、物質を構成する分子一つ一つの運動を模擬することによって、物質の状態を調べる計算手法である。運動方程式としては多くの場合、ニュートンの運動方程式を用い、分子に働く力は分子の 2 体力の和によって与える。そのとき扱う分子の数を N とすると、力の組みあわせは

$${}_N C_2 = \frac{N(N-1)}{2}$$

通りである。そのため、一般に計算時間は $O(N^2)$ で増加する。分子間にはたらく相互作用は、空間的に有効な範囲という視点から短距離相互作用、長距離相互作用の 2 種類に大別できる。分子動力学法の超並列化に際しては、短距離相互作用がその局所性を利用できるので、現在さかんに研究されている。

本研究では、短距離相互作用で比較的簡単な相互作用を選択する。具体的にはレナード・ジョーンズ型ポテンシャル

$$V(r) = 4\epsilon \left\{ \left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right\}$$

を用いる。ただし、 ϵ 、 σ は物質によって異なる定数である。このポテンシャルはアルゴンやヘリウムなどの、電気的に中性な稀ガス元素の単原子分子の気体および液体について、適用されてきた球対称ポテンシャルである。ここではアルゴンの値を使用し、物理条件としては、分子の数 N 、体積 V 、温度 T を一定とした。ポテンシャルの関数型から、分子間距離の増大にともなって相互作用の大きさが急速に 0 に近づくことがわかる。すなわち相互作用の有効な範囲は限定されている。このことを用いてある距離 (分子の直径の数倍程度) 以上では相互作用を 0 とすることによって、力の計算時間を短縮することが通常行なわれている (力の切断)。単原子分子の直径を単位として、力

の切断距離は通常 2.5 ~ 3.5 が用いられるが、ここでは 2.5 を使用した。時刻 t での位置 $r(t)$ 、速度 $v(t)$ の時間発展に用いる数値積分は、ベルレの方法による式

$$r_i(t+\Delta t) = r_i(t) + \Delta t v_i(t) + \frac{\Delta t^2}{2} \sum_{j \neq i} f(r_{ij}(t))$$

$$v_i(t + \Delta t) = v_i(t)$$

$$+ \frac{\Delta t}{2} \left(\sum_{j \neq i} f(r_{ij}(t)) + \sum_{j \neq i} f(r_{ij}(t + \Delta t)) \right)$$

を用いた。

3 並列計算への実装

3.1 計算機環境

本研究の、分子動力学法によるシミュレーションの計算機環境を示す。

逐次処理 Sparc Station 2, FORTRAN77

並列処理 CM-5, CM Fortran, CMMD

である。CM-5 はプロセッシングエレメント (PE) に SPARC microprocessor を用いているため、同種のチップを用いている Sparc Station 2 を比較の対象とした。また、CM-5 は MIMD 型計算機であるが、SIMD 型に使用することもできる。CM Fortran のみを使用すると SIMD 型の計算を行ない、CM Fortran に CMMD ライブラリを併用することで MIMD 型の計算を行なうことができる。逐次処理、並列処理 (SIMD 型、MIMD 型) による実装結果を、処理時間や高速化率について比較検討した。

3.2 逐次処理

逐次処理については各時間ステップごとに全ての組み合わせについて分子間距離を計算し、力の切断距離を以内にある分子との相互作用のみを計算する方法を用いた。このとき計算量は分子の数 N に対して

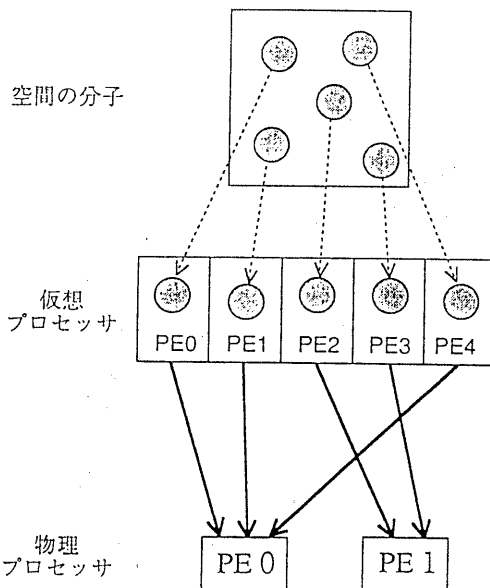


図 1: 単純分割法 の概念図

- 分子間距離の計算 $O(N^2)$
- 力の計算 $O(N)$

と考えられる。本研究で用いたプログラムを実装した結果、分子の数を増加させると $O(N^2)$ 的に計算時間が増加したため、短距離相互作用の計算においては力の切断を行なった場合でも、分子間距離の計算量の効果により、計算複雑度は $O(N^2)$ となることがわかる。

3.3 並列処理

並列処理では以下に挙げる 3 つの方法を実装した。

(1) 単純分割法

単純分割法の概要を図 1 に示す。CMFortran では仮想プロセッサを用い、コンパイラが仮想プロセッサを物理プロセッサに割りあてて。単純分割法では、仮想プロセッサ 1 つに対して分子を一つ割りあてるので、分子の数だけ仮想プロセッサを使用する。そのため、最も単純な形の並列処理の一つであ

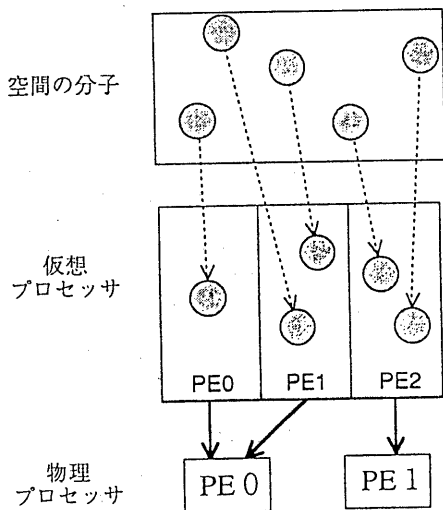


図 2: 2 ステージ法の概念図

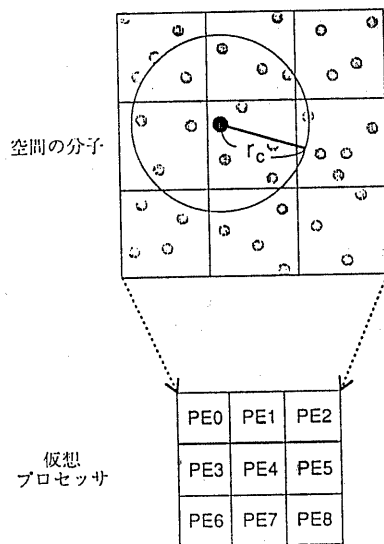


図 3: 近傍空間参照法の概念図

る。各 PE は、相互作用を計算するときに、その都度相手の分子データを参照する。そのため、全体としてプロセッサ間通信は無秩序に行なわれ、通信効率が悪いと考えられる。また、数密度などの物理条件によって計算時間が変化する要因はない。

(2) 2 ステージ法

2 ステージ法の概要を図 2 に示す。ここでは分子をある数のグループに分け、グループを仮想プロセッサに割りあてる。その際、相互作用の計算は

1. グループ内の分子間相互作用
(プロセッサ間通信は行なわない)
2. グループ相互の分子間相互作用
(一般にプロセッサ間通信が必要)

の 2 つのステージによって計算する。この方法では全ての組み合わせの分子間距離を計算する。そのため、すべての仮想プロセッサとデータをやりとりする。仮想プロセッサの持つデータをまとめて転送することによって、単純分割法に比較して通信回数を

減らすことが期待される。同じ数の分子の計算でも、グループ数が大きいときは通信回数が増加し、グループ数が小さいときは通信回数が減少すると予想される。

2 ステージ法において、数密度などの物理条件によって計算時間が変化する要因はない。また、ベクトルユニットの効率とプロセッサ間通信の 2 つの効果による影響は予測が困難である。

(3) 近傍空間参照法

近傍空間参照法の概要を図 3 に示す。近傍空間参照法では相互作用の局所性を利用し、2 ステージ法と同様に、相互作用を 2 ステージで計算する。ここでは、計算する空間を同じ大きさの立方体に分割し (セル)、セルの中にある分子を 2 ステージ法のグループと同様に、まとめて扱う。分子が移動するとセルの出入りが起こるので、位置を時間発展させるごとに分子が所属するセルを確認し、移動させる必要がある。セルの一辺を力の切断距離よりも大きくとれば、 $3 \times 3 \times 3$ セルの空間を考慮して、セル内部と 26 近傍のセルとの相互作用のみを計算すれば良い。

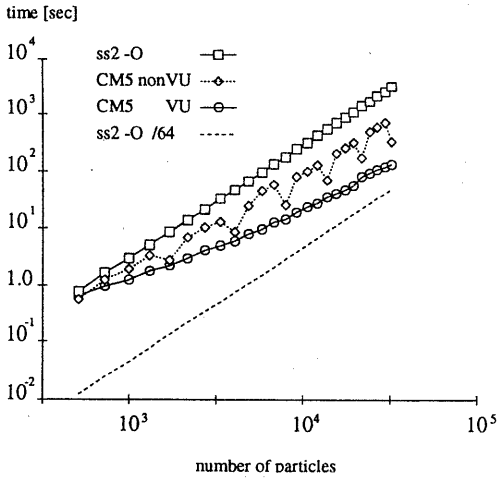


図 4: 単純分割法による計算時間

すなわち相互作用の計算では、

1. セル内の分子間相互作用
(プロセッサ間通信はない)
2. セル相互の分子間相互作用
(プロセッサ間通信をとまなう)

の2つのステージによって計算する。この方法では計算量と通信回数を削減できると期待される。近傍空間参照法では、セルの数が数密度に依存するので、物理条件によって計算時間が変化する可能性がある。

4 大規模シミュレーション

4.1 単純分割法

単純分割法をSIMD型処理で実装し、得られた計算時間を図4に示す。

ss2 -O オプティマイズ -O を使用したSS2による計算時間

CM5,nonVU ベクトルユニットを使用しないCM-5による計算時間

CM5,VU ベクトルユニット使用のCM-5による計算時間

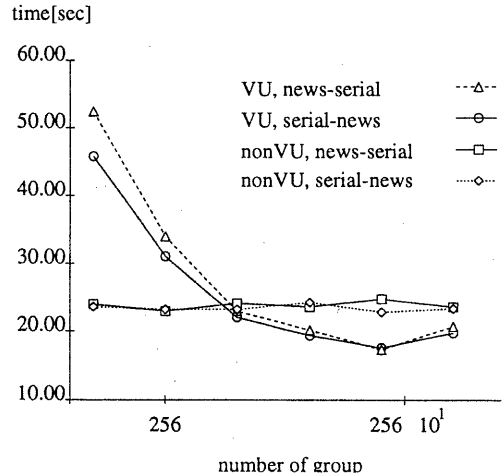


図 5: 実装方法に対する2ステージ法の計算時間の比較(4096分子)

ss2 -O /64 SS2による結果を64で除した時間

この結果によると、分子の数が增加するにつれて、並列処理による計算時間が逐次処理よりも大幅に減少することがわかる。また、ベクトルユニットを使用した場合(VU)の方が、使用しない場合(sparc)よりも高速であった。sparcでは、最適な問題の大きさが強く現れて、最適ではないサイズを計算した場合に計算時間は大きくなる。また、分子の増加にともなって計算時間は $O(N^2)$ 的に増加している。このプログラムで現れるCM-5の性能は、実測した範囲では分子の数の増加とともに増加する。125,000分子を計算した場合には284Mflop/sであった。ベクトルユニットを使用した64PEのCM-5では、理論ピーク性能は $R_{peak} = 8GFlop/s$ なので、CPUの使用効率は3.5%程度である。

4.2 2ステージ法

2ステージ法をSIMD型処理を用いて実装した。ベクトルユニットを使用した場合、使用しない場合について4096分子を種々の仮想プロセッサ数に分割した結果を図5に

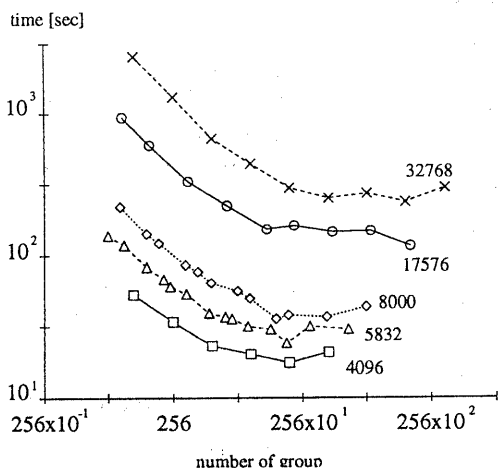


図 6: 2 ステージ法による計算時間 (ベクトルユニット使用)

示す。この方法は 2 次元配列を用いており、2 つの並列手法が考えられる。それぞれについても実装し比較した。この結果によると、並列化の指定方法による違いは小さい。またベクトルユニットを使用しない場合は仮想プロセッサ数による計算時間の違いは小さい。ベクトルユニットを使用する場合は、仮想プロセッサ数によって、計算時間が大幅に変化する。計算効率の悪い場合はベクトルユニットを使用しない場合よりも計算時間が大きくなり、効率の良い場合は計算時間に大幅な改善がみられる。効率の良くなるのは仮想プロセッサ数を大きくとった場合である。

さらに、5 種類の分子の数について、ベクトルユニットを使用し、グループの数を種々に変化させた計算時間を図 6 に示す。図中の数字は全体の分子の数である。グループの数を多数 (ベクトルユニット数×10 程度以上) にした場合に計算時間が短くなるという結果が得られた。すなわち、CM-5 においてベクトルユニットを使用する場合には、グループ数を大きくすることが効果的であると考えられる。

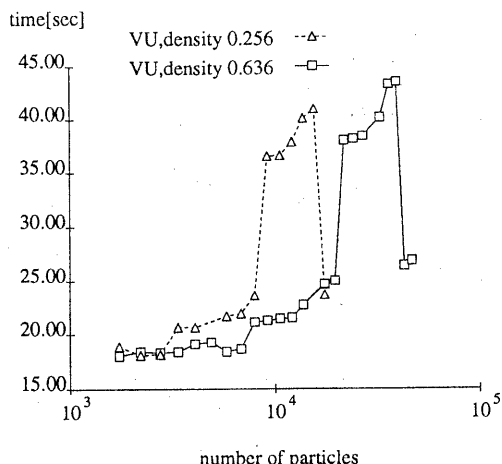


図 7: 近傍空間参照法による計算時間 (ベクトルユニット使用時のみ)

4.3 近傍空間参照法

4.3.1 SIMD 型処理

近傍空間参照法、SIMD 型処理による計算時間を図 7 に示す。分子の数および数密度から、空間の体積を定めて計算を行なうので、この 2 つのパラメータによってセルの数が増える。計算時間は分子の数よりも、むしろセルの数に依存すると思われる。2 種類の数密度を用いて計算した結果を図 8 に示す。非常に計算時間の増大する部分が見られるが、これはセルの数とベクトルユニットの関係によるものと考えられる。2 つの数密度では、計算時間のピークに対応する分子の数は異なるが、セルの数が等しい。すなわち、セルの数によって計算時間が規定され、密度の違いは、その計算時間が増大する部分に対応する分子の数をスライドさせることになる。

ここで得られる計算時間は逐次処理に比較して著しい向上を示す。しかし扱うデータ量からみると、本来得られるよりも高速化率は小さくなる。ここでは、配列の大きさは、セルに存在しうる上限の分子の数を固定的に定義している。ここでは 28 個まで対

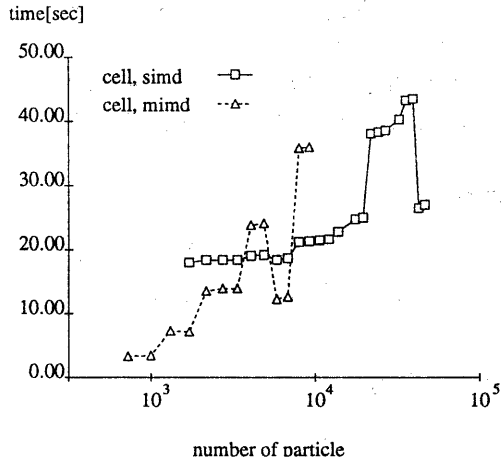


図 8: 近傍空間参照法を用いた SIMD 型処理と MIMD 型処理の比較

応できる。しかし、例えば数密度が 0.256 のときには、この大きさのセルに入る分子の数は平均 4 個であり、計算すべきデータの 7 倍の大きさの配列を定義していることになる。SIMD 型の計算では存在する分子に合わせて必要最小限の計算を行なうことが困難なので、計算効率は必ずしも高くない。すなわち配列中で有効なデータ量の割合が大きい場合でなければ、実行する計算量に対して有効な計算量が小さくなってしまう。

効率の比較的良好な数のセルを計算した場合は、分子の増加に対して計算時間の増加は小さく、問題サイズの拡大に対して近傍空間参照法は有効である。

4.3.2 MIMD 型処理

MIMD 型処理では hostless モデルを使用した。これは、全ての PE が一つの同じプログラムを実行し、host ↔ node のような上下関係はなく、並列な関係で同様の処理を行なうものである。

セルの PE への割り当て方法は、セルに端から順番に番号付けし、ノードの番号を *node*、セルの番号を *cell number* とすると、セルを担当するノードを

$$node = cell\ number \ mod\ 64$$

のように定めている。そのためセルの空間的な配置と PE への割り当て方は、一般に、特別な対応づけはない。通信は同期通信を用い、1 タイムステップの計算ではプログラムのうち力の計算をはさんで、3 か所で一齐に通信を行なっている。MIMD 型処理、SIMD 型処理による計算時間を図 8 に示す。ここでは、扱う配列の大きさの問題から、他の方法に比較して小規模な数の分子にとどめている。現在得ている結果から、プロセッサ間通信がオーバーヘッドとなっており、SIMD 型に比較すると速度の改善が行なわれていないことが分る。効率良く通信を行なうことは MIMD 型処理の今後の課題である。

4.4 高速化率

実装方法による比較のため、これまでに CM-5 で得られた並列処理による高速化率を図 9 に示す。ここで扱う高速化率は、1 つの問題の計算が並列化によってどの程度高速化するか、という観点から、逐次処理の SS2 を用いた実行時間に対するものとした。また、ベクトルユニットを使用した場合のみを示す。

simple 単純分割法

grouping 2 ステージ法

cell(density 0.636) 近傍空間参照法
(SIMD 型) 密度 0.636 の場合

全体に分子の数が増加するにつれて高速化率も向上している。中でも近傍空間参照法は、逐次処理と比較できる範囲では大幅な向上を示す。これは、近傍空間参照法の結果が、分子の増加に対して計算時間の増加が非常に小さいためである。

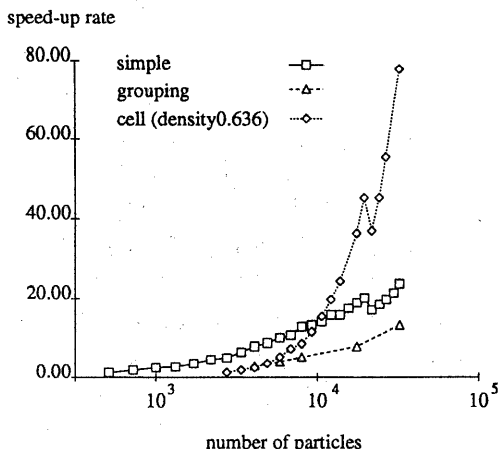


図 9: 高速化率

5 今後の課題

近傍空間参照法は大規模なシミュレーションを行なう際に有効であることが示された。しかし、物理状態によっては有効性が損なわれる場合も考えられる。分子動力学法で用いられる相互作用では、多くの場合に引力のはたらく部分があり、温度が低い場合には、その引力によって分子が結合する確率が高くなる。ある分子の周囲に他の分子が集まり、部分的に密度の大きい空間ができると、セル中の分子のデータ量が不均等になるので、PEごとに計算負荷に偏りが生じる。このことを考慮して、PEごとに負荷が均等になるように、何らかの方法で、データを移動させることが必要になる。(動的負荷分散) 実際に、分子の密度変化によって計算時間にゆらぎが生じることが指摘されている [1]。動的負荷分散の手法としては、2次元のシミュレーションにおいて、空間を短冊状に分割し、境界を分子の数の変化に従って移動させる方法が考案、実装され、効果が報告されている [4]。しかし3次元の立方体型のセルに関しては、境界を変型させる方法は、近傍空間のみを参照すれば良い点と適合しない可能性があるため、妥当ではない。大規模シミュレーションにおいて

は、セルが多数となるので、セルを単位としてタスクスケジューリングを行なうことが有効であると考えている。今後は負荷分散を考慮した実装を行なう予定である。

謝辞

本研究の一部は文部省科学研究費を用いて行なわれた。関係各位に感謝する。

参考文献

- [1] D. M. Beazley and P. S. Lomdahl, *Message-passing multi-cell molecular dynamics on the Connection Machine 5*, *Parallel Computing* 20(1994)173-195.
- [2] F. Hedman and A. Laaksonen, *An approach to data parallel molecular dynamics for liquids*, *International Journal of Modern Physics C* Vol.4, No.1 (1993) 41-48
- [3] P. Tamayo, J. P. Mesirov and B. M. Boghosian, *Parallel Approaches to Short Range Molecular Dynamics Simulations*, *Proceedings of Supercomputing 91*(1991).
- [4] F. Bruge, S. L. Fornili, *Concurrent molecular dynamics simulation of spinodal phase transition on transputer arrays*, *Computer Physics Communications* 60(1990)31-38