

JAERI-Data/Code

96-011



短距離力分子動力学法のベクトル並列化手法の性能評価

1996年3月

田中靖久・蕪木英雄・横川三津夫

日本原子力研究所
Japan Atomic Energy Research Institute

本レポートは、日本原子力研究所が不定期に公刊している研究報告書です。

入手の間合わせは、日本原子力研究所技術情報部情報資料課（〒319-11 茨城県那珂郡東海村）あて、お申し越してください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11 茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

This report is issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division, Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1996

編集兼発行 日本原子力研究所
印刷 ㈱原子力資料サービス

短距離力分子動力学法のベクトル並列化手法の性能評価

日本原子力研究所計算科学技術推進センター

田中 靖久・蕪木 英雄・横川三津夫

(1996年2月9日受理)

短距離力分子動力学法 (MD) プログラムをベクトル並列化し, MDのベクトル並列化手法を検討した。並列化は分散メモリ型ベクトル並列計算機である富士通V P P500で行った。

有力なベクトル化手法として知られるLayered Link Cell 法をベースにしたペアリストの分割と粒子分割法を組み合わせた並列化手法を検討し, その並列化アルゴリズムの性能評価を行い, その特性を明らかにした。

Performance Estimation of a Vector Parallel Calculation
for Short-range Molecular Dynamics

Yasuhisa TANAKA, Hideo KABURAKI and Mitsuo YOKOKAWA

Center for Promotion of Computational Science and Engineering
Japan Atomic Energy Research Institute
Nakameguro, Meguro-ku, Tokyo

(Received February 9, 1996)

A vector parallel calculation for short-range molecular dynamics has been performed on Fujitsu VPP500.

A combination method of pair-lists division based on a vectorization technique of Layered Link Cell method and atomic decomposition to parallelization was adopted. Performance of the parallel calculation was estimated.

Keywords: Short-range Molecular Dynamics, Vector Parallel Calculation, Layered Link Cell Method, Atomic Decomposition, VPP500

目 次

1. はじめに	1
2. 分子動力学法計算	2
2.1 系	2
2.2 運動方程式	3
2.3 分子間ポテンシャル	3
2.4 初期条件	4
2.4.1 初期配置	4
2.4.2 初期速度	5
2.5 速度スケーリング	5
2.6 数値積分	5
3. プログラム概説	6
3.1 入力データ	6
3.2 プログラム構造	6
4. ベクトル化	8
4.1 ベクトル化手法	8
4.1.1 ペアリストの作成	8
4.1.2 L.C 法	8
4.1.3 Double list data structure	9
4.1.4 L.L.C.法	11
4.2 ベクトル化性能評価	14
4.2.1 計算条件	14
4.2.2 性能評価結果	14
5. 並列化手法	18
5.1 演算	18
5.1.1 リスト生成および力の計算	18
5.1.2 その他の計算	19
5.2 通信	19
5.2.1 位置情報の通信	20
5.2.2 力の部分和の通信	22
6. 並列化性能評価	25
6.1 計算条件	25
6.2 性能評価	25
6.2.1 演算の性能評価	26
6.2.2 全体の性能評価	29

7. 結 言	31
参考文献	32

Contents

1. Introduction	1
2. Molecular Dynamics	2
2.1 System	2
2.2 Equation of Motion	3
2.3 Intermolecular Potential	3
2.4. Initial Conditions	4
2.4.1 Initial Positions	4
2.4.2 Initial Velocities	5
2.5 Scaling of Velocities	5
2.6 Numerical Integral Method	5
3. Program	6
3.1 Input Data	6
3.2 Program Structure	6
4. Vectorization	8
4.1 Vectorization Methods	8
4.1.1 Pair List Generation	8
4.1.2 Link Cell Method	8
4.1.3 Double List Data Structure	9
4.1.4 Layered link Cell Method	11
4.2 Performance Evaluation for Vectorization	14
4.2.1 Calculation Condition	14
4.2.2 Performance Evaluation	14
5. Parallelization Methods	18
5.1 Computation	18
5.1.1 Pair List Generation and Force Computation	18
5.1.2 Other Computation	19
5.2 Communication	19
5.2.1 Communication of Positions	20
5.2.2 Communication of Force	22
6. Performance Estimation for Parallelization	25
6.1 Calculation Condition	25
6.2 Performance Estimation	25
6.2.1 Performance Estimation for Computation	26
6.2.2 Total Performance Estimation	29

7. Conclusions	31
References	32

1. はじめに

近年における計算機の著しい進歩に伴い、先進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。例えば、我々の身の回りを取り巻く物質・材料についても、材料を構成する原子や分子数が莫大であるため、最近に至るまで非常に簡単な現象以外は基本法則の具体的適用がかなり限られてきた。ところが、スーパーコンピュータおよび理論それ自体の進歩により、物質・材料の諸性質を非経験的に計算する可能性も急速に開けてきた。これにより、物性予測が一部でも可能になれば、新物質の探索など材料開発に大きく寄与できることが期待できる¹⁾。

ここではそうした数値シミュレーションの一手法である分子動力学法 (MD) のベクトル並列計算法についての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大概 $10^3 \sim 10^6$ 個のオーダーである。一方、本来の物質はアボガドロ数 6.02×10^{23} 個程度の分子の集まりであるため、現在計算可能な物性は量的・質的に限られており、並列コンピュータによる高速化が期待されている。

欧米ではとりわけスカラ並列化手法に関する研究が盛んに行われている。そこで、本研究においては欧米では比較的研究例の少ないベクトル並列化手法に絞って検討し、その手法を分散メモリ型ベクトル並列計算機である富士通VPP500上で実現した。

ベクトル並列化の目的は、ベクトル加速率と並列加速率の積で表されるトータルの加速率を最大にし、これにより、計算パワー不足のために今までにできなかった大規模シミュレーションを可能にし、新たな知見を得ることである。このように大規模な系を対象とする数値シミュレーションを行うには、既知の有力なベクトル化手法をベースにした並列化手法の研究は意義がある。

本研究においては、MIT開発の短距離力MDプログラム²⁾をベクトル並列化した。MD計算の演算時間の大部分を占めるリスト生成および力の計算を評価するために、動径分布関数 $g(r)$ の計算は除外した。

ここでは、有力なベクトル化手法として知られるL. L. C. (Layered Link Cell) 法³⁾をベースにした並列化手法を検討し、その並列化アルゴリズムの性能評価を行い特性を明らかにした。

2. 分子動力学法計算

本研究において取り扱う分子動力学法計算の概略を述べる。

2.1 系

たとえ対象が巨視的なサイズの試料であるとしても、もっと小さい空間のスケールで見られる粒子の集団の様相がわかれば、巨視的特性の本質は説明できるものとする。従って、巨視的試料の内部に一辺の長さ L 、体積 $V=L^3$ の仮想的な立方体の枠を設定する(図2.1)。この仮想的な立方体の枠の内部にある粒子数が N とする。実際の対象の原子数密度 ρ_N に対して、 N と L が(2-1)式の関係を満たすものとする。対象の密度 ρ を、原子の質量を m とすると(2-2)式が満たされるものとする。

$$\rho_N = \frac{N}{L^3} \quad (2-1)$$

$$\rho = \frac{mN}{L^3} \quad (2-2)$$

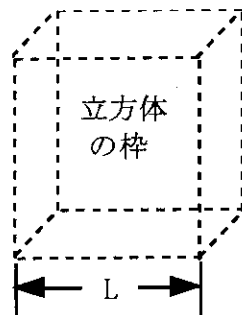


図2.1 基本立方体の枠

立方体の表面はあくまでも仮想的なものであることを保証するために、周期境界条件を仮定する。すなわち、図2.2のように立方体の各面に接して、全く同一の内部粒子の配置をもつ仮想的な枠が三次元的に並んでいるものと仮定する。

粒子 i が粒子 j から受ける力の計算については、最近接鏡像法 (minimum image criterion) ⁴⁾を用いる。すなわち、最初の仮想的な立方体(基本立方体)内の粒子 j に対して隣接する立方体内の粒子(鏡像、 j_1, \dots, j_8)を考え、粒子 j と鏡像 j_1, \dots, j_8 の中で粒子 i との距離が最小のもの(図では j_3)を採用する。

N 個の同一粒子が、体積 V の立方体の容器の中に入っている系で、系の全エネルギー E が一定に保たれるような系を考える。いわゆる、 (N, V, E) アンサンブルと呼ばれるものである。統計力学の言葉では、マイクロカノニカルアンサンブルと呼ばれる。

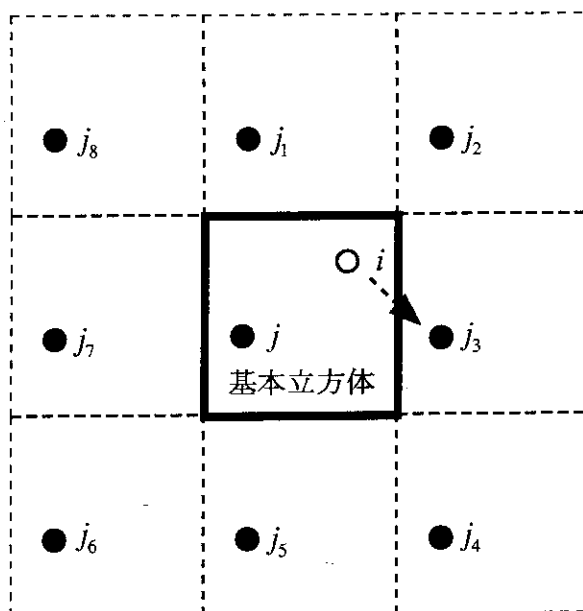


図2.2 最近接鏡像法

2.2 運動方程式

分子動力学法とは物質を構成する各々の粒子の運動を模擬することにより、物質の状態を調べる計算手法である。系が従うべき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 (2-3) 式を用いる。(2-3) 式を解いて、各時刻における全粒子の位置 $r_i(t)$ と速度 $v_i(t)$ を求める。

$$m \frac{d^2 r_i}{dt^2} = m \frac{dv_i}{dt} = F_i(t) = -\frac{\partial}{\partial r_i} U(r^N) \quad , \quad r^N \equiv (r_1, r_2, \dots, r_N) \quad (2-3)$$

ここで、 m は粒子の質量、 r^N は粒子の位置ベクトル、 $F_i(t)$ は時刻 t における粒子 i に及ぼされる力、 $U(r^N)$ は分子間ポテンシャルである。

2.3 分子間ポテンシャル

分子間ポテンシャルとしては、最もよく用いられる Lennard-Jones の 6-12 ポテンシャル (2-4) 式を仮定する。但し、 ϵ および 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 Ar や He などの電氣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経験的な球対称ポテンシャルである。ここでは固体 Ar の値を使用する。

(2-5) 式に示すように、長さについては σ で規格化し、エネルギーについては ϵ で無次元化すると、(2-4) 式は (2-6) 式となる。

粒子数 N なので、力の計算の組み合わせは $N(N-1)/2$ 通りであり、 $O(N^2)$ のオーダーで計算コストがかかる。そこで、分子間距離が増大すると分子間相互作用の大きさが急速に0に近づく場合にはある距離以上で相互作用を0として力の計算時間を短縮する。この距離を「力の打ち切り距離 (cut-off distance)」というが、ここでは経験的に良く用いられる(2-7)式を採用する。

$$U(r^N) = u_{LJ}(r) = 4\epsilon \left\{ \left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right\} \quad (2-4)$$

$$u^* = \frac{u}{\epsilon}, \quad r^* = \frac{r}{\sigma} \quad (2-5)$$

$$u^*(r^*) = 4 \left\{ \frac{1}{r^{*12}} - \frac{1}{r^{*6}} \right\} \quad (2-6)$$

$$r_c^* = 2.5 \quad (2-7)$$

このままでは、距離 r にある粒子間に及ぼす力 $F(r)$ が $r=r_c$ において不連続になる。そこで、力とポテンシャルを(2-8)式および(2-9)式に示すようにシフトし、 $r^*=r_c^*$ において滑らかに(微係数が連続になるように)接続する。

$$F_{shift}^*(r^*) = -\frac{d}{dr^*} u^*(r^*) + \left(\frac{du^*(r^*)}{dr^*} \right)_{r^*=r_c^*} \quad \text{for } r^* < r_c^* \\ = 0 \quad \text{for } r^* \geq r_c^* \quad (2-8)$$

$$u_{shift}^*(r^*) = u^*(r^*) - u^*(r_c^*) - (r^* - r_c^*) \left(\frac{du^*(r^*)}{dr^*} \right)_{r^*=r_c^*} \quad \text{for } r^* < r_c^* \\ = 0 \quad \text{for } r^* \geq r_c^* \quad (2-9)$$

2.4 初期条件

2.4.1 初期配置

固体Arを対象としているため、初期配置においては、FCC(面心立方格子)の結晶構造に従うものとする。FCCにおいては単位セルあたり4個の粒子が含まれるので、体積 V の中に含まれる単位格子の個数を N_c^3 とすると、粒子数 N は(2-10)式で表わされる。

$$N = 4N_c^3 \quad (2-10)$$

2.4.2 初期速度

Maxwell分布に従って、仮の速度 $\{v_i^*(0)\}$ を定める。しかし、このままでは系全体がある方向に動く（剛体運動）可能性があるため、全粒子平均の速度ベクトルが0になるように、(2-11) 式に従って $v_i^{*new}(0)$ にシフトして初期速度とする。

$$v_i^{*new}(0) = v_i^*(0) - \frac{1}{N} \sum_{j=1}^N v_j^*(0) \quad (2-11)$$

2.5 速度スケーリング

ある程度平衡状態を作り出して（平衡計算）から、実際にマクロ量のサンプリング計算（本計算）を始める。平衡計算終了時の速度を $\{v_i^*(t_{eq})\}$ とすると、その時刻における温度 T_{eq}^* は (2-12) 式で表わされる。

平衡計算終了後、本計算が始まる前に、入力データで指定の温度 T^* になるように (2-13) 式に従って速度スケーリングを行う。

$$T_{eq}^* = \frac{1}{3N} \sum_{i=1}^N \{v_i^*(t_{eq})\}^2 \quad (2-12)$$

$$v_i^{*new}(t_{eq}) = v_i^*(t_{eq}) \sqrt{\frac{T^*}{T_{eq}^*}} \quad (2-13)$$

2.6 数値積分

Gearによる5次の予測子-修正子法（predictor-corrector）⁵⁾により、時刻 t での位置 $r(t)$ および速度 $v(t)$ の時間発展として、運動方程式 (2-3) 式を解く。この方法は計算精度が良く、しかも力の計算が1ステップあたり1回で済むというメリットがある。

3. プログラム概説

3.1 入力データ

入力データを表3.1に示す。

表3.1 入力データ

変数名	意味
NP	粒子数
NEQ	平衡計算タイムステップ数
MAXKB	一本計算タイムステップ数
TR	温度T (無次元化単位系)
DR	密度 ρ (無次元化単位系)

3.2 プログラム構造

プログラム構造の概略を図3.1に示す。

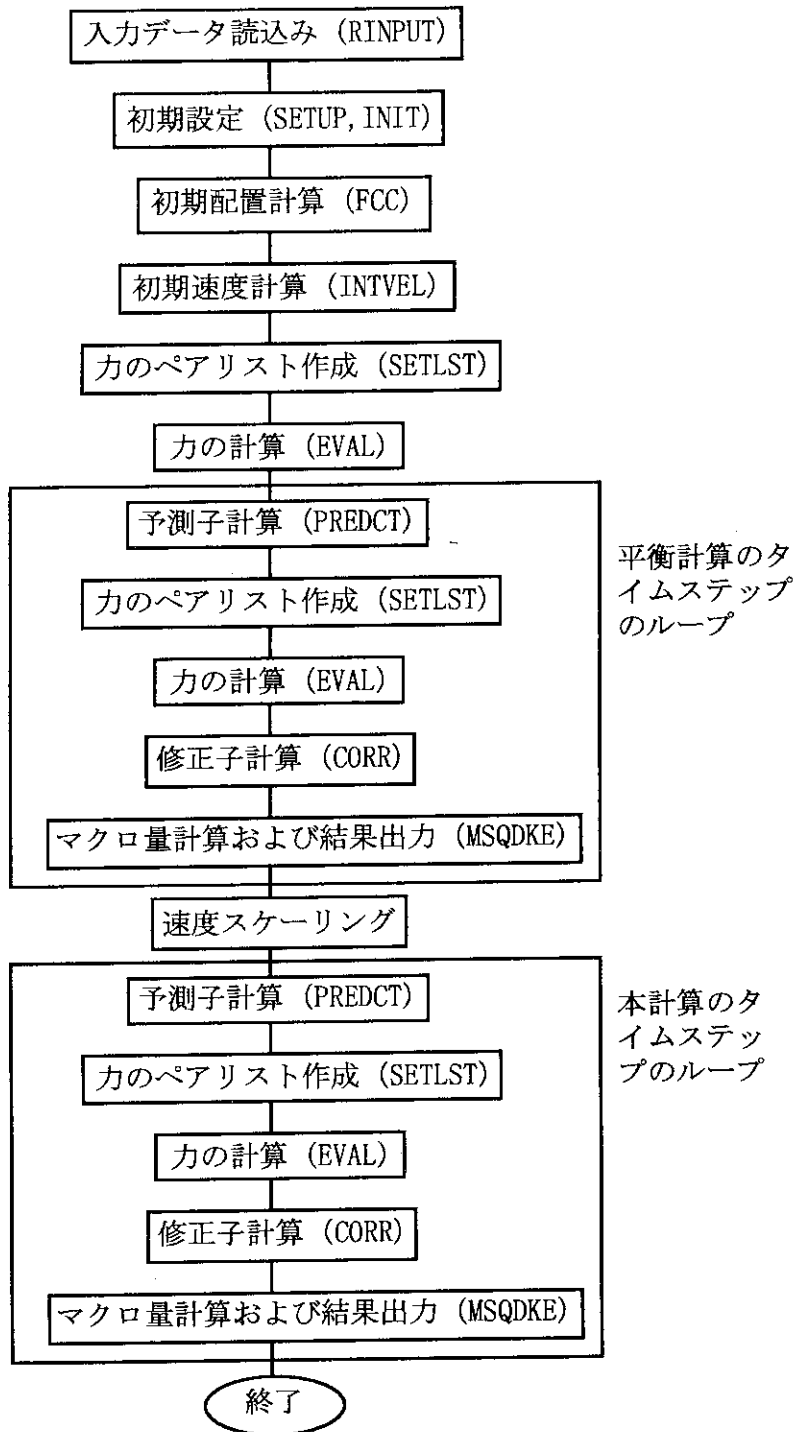


図3.1 プログラム構造の概略

4. ベクトル化

4.1 ベクトル化手法

4.1.1 ペアリストの作成

2体ポテンシャルを用いる短距離力分子動力学法では、粒子間の距離の打ち切り距離 r_c^* (=2.5) 以内にある全てのペア (i, j) に対して、力の計算をする必要がある。時間ステップ毎に、全ての粒子のペアの距離をチェックするのは力の打ち切り距離よりも離れた粒子の方が多く、実際に力の計算を必要としない距離の計算が多いため棄却率が大きく、無駄な計算が多い。

そこで、あるタイムステップごとに、打ち切り距離 r_c^* より若干大きいリスト生成距離 $L_c^* = r_c^* + r_s^*$ 以内の粒子のペアリストを作成しておく。そのペアリストの中から打ち切り距離 r_c^* 以内にあるかをチェックすれば棄却率が小さくなる。 $r_s^* = 0.3$ が経験的によく用いられる。

ペアリストを作成・更新する間隔は (4-1) 式を満足する間隔の中で最大のものが (リストの生成回数が最小となる意味で) 最適であり、 r_s^* のとり方、系の速度場および時間ステップ Δt に依存して決まる。

$$(\text{その間隔内の全粒子の最大変位}) \leq r_s^* / 2 \quad (4-1)$$

4.1.2 L.C. 法

L.C. 法 (Link Cell Method) ³⁾ では、系全体を一辺の長さ L_{cell}^* の立方体セルに等分割する (図4.1)。ここで、一辺の長さ L_{cell}^* は等分割可能で、かつ (4-2) 式を満たす中で最小値を選択する。等号が成立するのが最適であるのは言うまでもない。

$$L_{cell}^* \geq L_c^* \quad (4-2)$$

図4.1に示すように、あるセルに含まれる粒子のペアの相手は、同じセルに含まれるか (セル内相互作用、inner cell interaction)、もしくは隣り合う26個 (二次元では8個) のセルに含まれるか (セル間相互作用、inter cell interaction) のどちらかである。

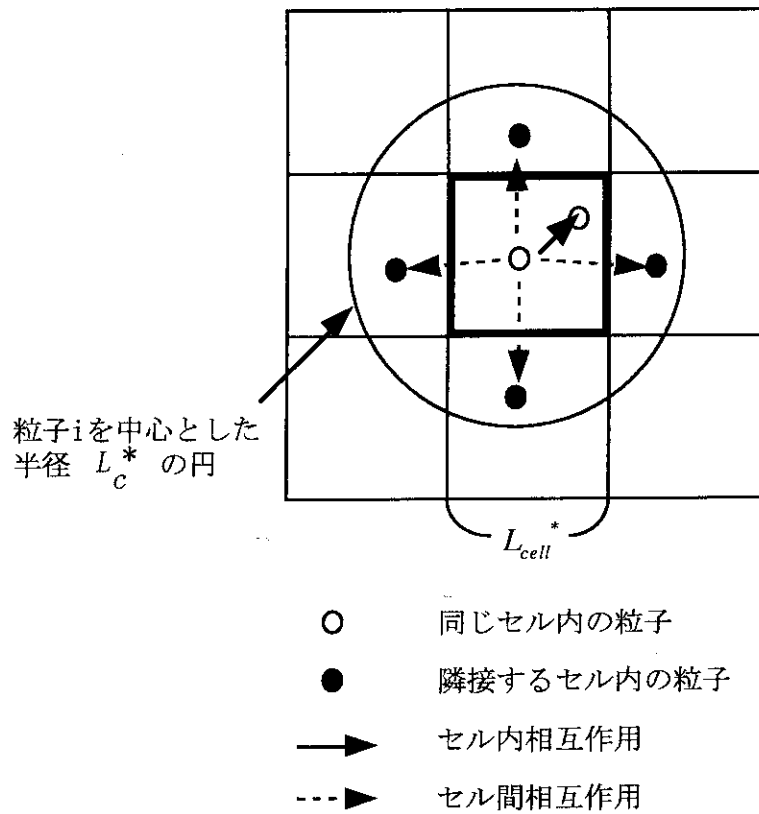


図4.1 L.C.法による相互作用のチェック

4.1.3 Double list data structure

ベクトル化する際にペアリストのデータ構造として、double list data structure³⁾がよく用いられる。

このデータ構造では全てのペア (*i*, *j*) に対して、*i*および*j*をそれぞれ一次元配列LIST1 (*K*) (*K*=1, NLIST) およびLIST2 (*K*) (*K*=1, NLIST) に格納する。ここで、NLISTはペアリストの総数である。このデータ構造をベースに力の計算を行うFORTRANプログラムとしては、図4.2に示すようなプログラム構造が考えられる。このままでは、力の和を求める処理が再帰処理のため (リストベクトルLIST1, LIST2による間接アドレス参照のため、異なる*K*に対して同じ*i*および*j*が現われる可能性があるため)、ベクトル化できない。

```

DO 10 K=1, NLIST
  I=LIST1(K)
  J=LIST2(K)
  IとJとのinteraction force RFX, RFY, RFZの計算
  FX(I)=FX(I)+RFX
    
```

```

FX(J)=FX(I)-RFX
FY(I)=FY(I)+RFY
FY(J)=FY(I)-RFY
FZ(I)=FZ(I)+RFZ
FZ(J)=FZ(I)-RFZ
10 CONTINUE

```

図4.2 力の計算を行うFORTRANプログラム構造 (ベクトル化不可)

ベクトル化するためには、総数NLISTのペアリストを「そのサブリスト中のiおよびjの中に同じ粒子番号が2回以上現われない」ような複数のサブリストに分割することが必要となる。このサブリストの分割が本研究で採用するL. L. C. (Layered Link Cell) 法のベースになる考え方である。

サブリストの個数をIPACK、各サブリストの先頭ポイントをLPOINT(L) (L=1, IPACK) とする。便宜的に、LPOINT(IPACK+1)=NLIST+1とすると、サブリストのデータ構造をベースに力の計算を行うFORTRANプログラム構造を図4.3に示す。

ここで、「*VOCL LOOP, NOVREC」は、無指定ではコンパイラが再帰処理であると認識するため、以下のループが再帰処理でないことをコンパイラに示す指示行である。VPP500ではこのVOCL文で指定する。

```

DO 20 L=1, IPACK
  KSTA=LPOINT(L)
  KEND=LPOINT(L+1)-1
  *VOCL LOOP, NOVREC
  DO 10 K=KSTA, KEND
    I=LIST1(K)
    J=LIST2(K)
    IとJとのinteraction force RFX, RFY, RFZの計算
    FX(I)=FX(I)+RFX
    FX(J)=FX(I)-RFX
    FY(I)=FY(I)+RFY
    FY(J)=FY(I)-RFY
    FZ(I)=FZ(I)+RFZ
    FZ(J)=FZ(I)-RFZ
  10 CONTINUE
20 CONTINUE

```

図4.3 力の計算を行うFORTRANプログラム構造 (ベクトル化可能)

4.1.4 L.L.C. 法

ここでは、L.L.C. (Layered Link Cell) 法³⁾ におけるサブリスト作成の考え方を記す。

セル番号 NC に含まれる粒子数を $N_{BOX}(NC)$ ($NC=1 \sim N_{cell}$ 、ここで N_{cell} はセル総数)、セル番号 NC の KC 番目 (「セル内番号 KC 」) の粒子番号を $I_{BOX}(NC, KC)$ ($NC=1 \sim N_{cell}$, $KC=1 \sim N_{BOX}(NC)$) とする。各セルに含まれる粒子数 $N_{BOX}(NC)$ の最大値を N_{max} とする。便宜上、 $N_{BOX}(NC) < N_{max}$ なるセル番号 NC の、 $KC = N_{BOX}(NC) + 1 \sim N_{max}$ なる「セル内番号 KC 」 $I_{BOX}(NC, KC)$ については、ダミー粒子として粒子番号 0 をセットする。ダミー粒子との相互作用は当然考えない。

L.L.C. 法においては、以下の2つの相互作用についてサブリストに分割し、ペアリストを生成する。

(1) 同じ「セル内番号 KC 」をもつ粒子間の相互作用

隣接するセル内の粒子間の相互作用のみチェックすればよい。作用・反作用の法則により、半分の隣接するセルに対してチェックすればよい。サブリストの個数が高々 13 (二次元では 4) となる。

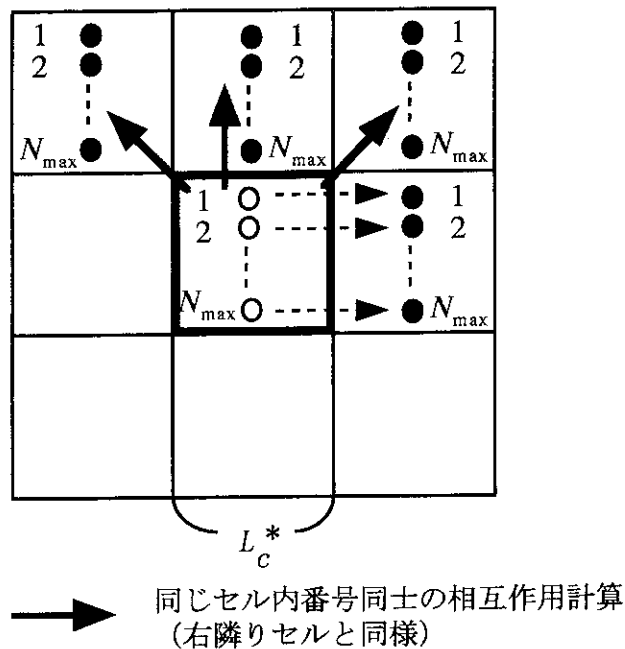


図4.4 同じ「セル内番号 KC 」をもつ粒子間の相互作用

(2) 異なる「セル内番号 KC 」をもつ粒子間の相互作用

同じセル内の粒子間の相互作用（セル内相互作用）、および隣接するセル内の粒子間の相互作用（セル間相互作用）に対してチェックする必要がある。27個（二次元では8個）のセルに対してチェックすることになる。

いずれの場合も、チェックすべき「セル内番号 KC 」のオフセットは、1から N_{PACK} ($= N_{max}/2$) までである。オフセット1の例を、図4.5および4.6に示す。この時、サブリストの個数は高々 $27 * N_{PACK}$ （二次元では $9 * N_{PACK}$ ）となる。

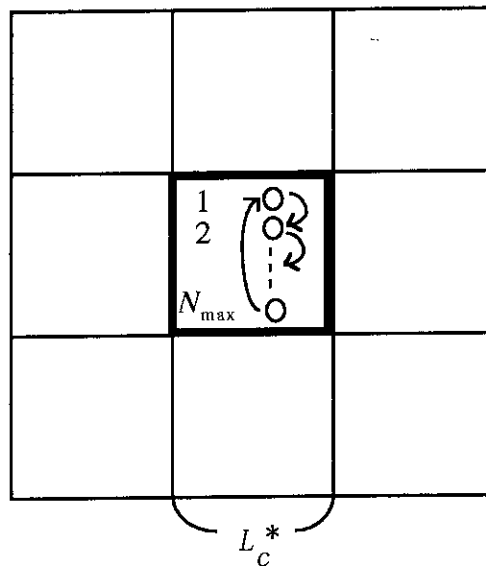


図4.5 異なる「セル内番号 KC 」をもつ粒子間のセル内相互作用

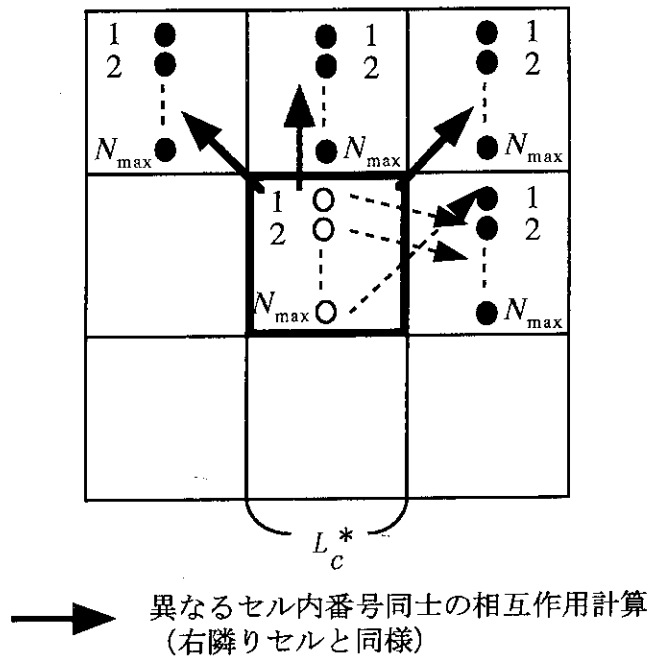


図4.6 異なる「セル内番号KC」をもつ粒子間のセル間相互作用

4.2 ベクトル化性能評価

本研究で採用するL.L.C.法のベクトル化手法としての基本特性（ベクトル加速率、演算コストの性能評価）を確認する。

4.2.1 計算条件

計算条件および計算粒子数Nを表4.1および表4.2に示す。

表4.1 計算条件

意味	入力値
平衡計算タイムステップ数	140
本計算タイムステップ数	1000
温度T（無次元化単位系）	1.1
密度 ρ （無次元化単位系）	0.7

表4.2 計算粒子数N

評価目的	粒子数N
ベクトル加速率	864, 2048, 4000, 8788の4ケース
演算コストの性能評価	864, 2048, 4000, 8788, 16384, 32000の6ケース

4.2.2 性能評価結果

(1) ベクトル加速率

粒子数N=864, 2048, 4000, 8788に対する、VPP500のプロセッサ1台によるスカラ計算時間とベクトル計算時間、およびベクトル加速率を表4.3に示す。粒子数Nの増加に伴い、ベクトル加速率も単調に増加している。

表4.3 ベクトル加速率 (N=864, 2048, 4000, 8788)

粒子数N	スカラ計算時間 (sec)	ベクトル計算時間 (sec)	ベクトル加速率
864	142.868	24.799	5.761
2048	275.493	26.334	10.461
4000	598.344	46.061	12.990
8788	1689.773	91.425	18.483

(2) ベクトル計算時間

粒子数N=864, 2048, 4000, 8788, 16384, 32000に対する、性能評価に用いるLink Cellの一辺の長さ $L_{cell}^*(N)$ とVPP500のプロセッサ1台によるベクトル計算時間 (単位: sec) を表4.4に示す。ここで、 C_{nbd} は (4-3) で表わされる (性能評価モデルに用いる) 定数、 C_{dis} は (4-4) で表わされる定数、リスト生成距離 $L_c^* = r_c^* + r_s^* = 2.8$ とする。

$$C_{nbd} = (6 + 12\sqrt{2} + 8\sqrt{3})/26 = (3 + 6\sqrt{2} + 4\sqrt{3})/13 \approx 1.4164 \quad (4-3)$$

$$C_{dis} = C_{nbd} \times L_{cell}^*(N) - L_c^* \quad (4-4)$$

表4.4 ベクトル計算時間 (N=864, 2048, 4000, 8788, 16384, 32000)

N	Lcell*	Cdis	ベクトル計算時間
864	3.5756	2.2646	24.799
2048	2.8605	1.2517	26.334
4000	2.9797	1.4205	46.061
8788	2.9052	1.3150	91.425
16384	2.8605	1.2517	161.368
32000	2.9797	1.4205	322.850

(3) ベクトル計算時間の性能評価

ここでは、ベクトル計算時間の性能評価モデルを仮定し、最小自乗法によりモデル中の各定数を求め、計測データとの比較を行う。以下に示す性能評価モデルは報告者が提案するものである。

短距離力分子動力学法計算では、リスト生成と力の計算の2つが全体の計算時間の大部分

を占めるが、なかでも粒子間距離の計算コストが支配的となる。そのコストはペアリストの総数に密接に関連する。従って、ベクトル計算時間を決める要因として、まずペアリストの総数に関連する粒子数 N が挙げられる。ここでは、ベクトル計算時間はそのべき乗 N^α に比例するものと仮定する。

次に粒子数 N を固定したとき、ベクトル計算時間に影響を与える要因を考える。L. L. C. 法 (L. C. 法も同様) では、隣接するセル内の粒子との相互作用をチェックするが、リスト生成距離 L_c^* 以上の距離の計算は無駄になる(棄却される)。Link Cellの長さ $L_{cell}^*(N)$ とリスト生成距離 L_c^* の差が大きいく程、この棄却の割合は大きくなる(4.1.2参照)。従って、何らかの意味でLink Cellの長さ $L_{cell}^*(N)$ を考慮した性能評価モデルが必要となる。

そこで、リスト生成時にチェックする粒子間の平均距離を近似的に計算する。簡単のために、セル中心と、26個の隣接セルの中心との距離を考える。3次元セルの場合は、以下のようになる。

- ・ 中心間距離 $L_{cell}^*(N)$: 6個の隣接セル
- ・ 中心間距離 $\sqrt{2}L_{cell}^*(N)$: 12個の隣接セル
- ・ 中心間距離 $\sqrt{3}L_{cell}^*(N)$: 8個の隣接セル

従って、リスト生成時にチェックする粒子間の平均距離は $C_{nbd} \times L_{cell}^*(N)$ と近似できる。ここで、 C_{nbd} は(4-3)で表わされる定数である。

これにより、棄却の割合は、「リスト生成時にチェックする粒子間の平均距離とリスト生成距離 L_c^* の差」 $C_{nbd} \times L_{cell}^*(N) - L_c^*$ に密接に関連するとし、そのべき乗 $(C_{nbd} \times L_{cell}^*(N) - L_c^*)^\beta$ にベクトル計算時間は比例するものと仮定する。

厳密には物理的計算条件に依存する動径分布関数 $g(r)$ を用いて性能評価をすべきであるが、ここでは単に、粒子数 N とLink Cellの一辺の長さ $L_{cell}^*(N)$ を用いた性能評価モデル(4-5)式を仮定する。

ベクトル計算時間の計測データ6点(表4.4)より、最小自乗法によりモデル中の各定数を求めると(4-6)のようになった。この性能評価モデルと計測データとの比較を図4.7に示す。十分大きい粒子数 N に対するベクトル計算時間を予測するには十分な精度と考えられる。

本性能評価モデルにおいては、粒子数 N が小さい場合には $(C_{nbd} \times L_{cell}^*(N) - L_c^*)^\beta$ の項を考慮する必要がある。しかし、粒子数 N が十分大きい場合には、Link Cellの長さ $L_{cell}^*(N)$ とリスト生成距離 L_c^* との差は十分小さくなるため、 $(C_{nbd} \times L_{cell}^*(N) - L_c^*)^\beta$ は定数 $\{(C_{nbd} - 1) \times L_c^*\}^\beta$ と近似できる。結局、ベクトル計算時間がオーダー $O(N^\alpha)$ で決まることがわかる。

$$t_{comp}(N) = C_{comp} N^\alpha (C_{nbd} \times L_{cell}^*(N) - L_c^*)^\beta \quad (4-5)$$

$$C_{comp} = 0.0244 \quad , \quad \alpha = 0.8727 \quad , \quad \beta = 1.2217 \quad (4-6)$$

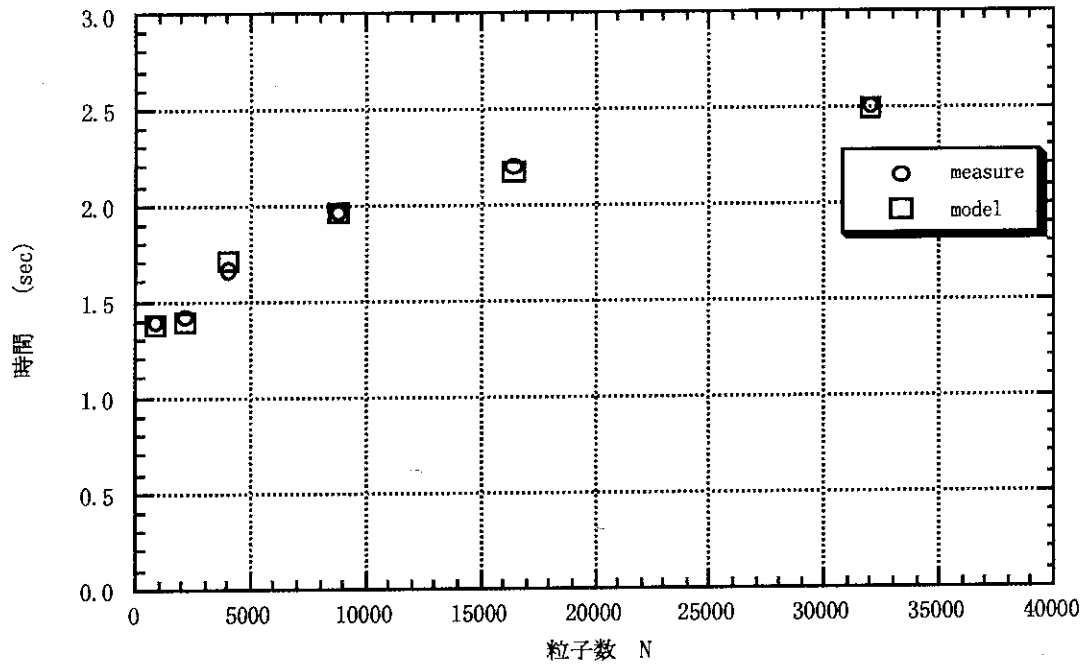


図4.7 性能評価モデルとベクトル計算時間との比較

5. 並列化手法

本研究で採用した並列化手法は、粒子分割法 (Atomic Decomposition) ⁶⁾ と L. L. C. 法をベースにしたサブリスト分割法を組み合わせたものである。

5.1 演算

本研究で用いた並列化手法の概略を表5.1に示す。詳細は以下に記す。

表5.1 並列化手法の概略

	リスト生成 および力の計算	その他の計算 (予測子・修正子計算等)
手続き分割	サブリストの分割	均等分割
データ分割	各プロセッサごとに担当する サブリスト情報を重複なく格納	均等分割
位置情報	各PEが全粒子の位置 情報を重複して格納	各PEが担当するN/P個の 粒子の位置情報のみ更新
力の情報	各PEが担当した力の計算結果を 全粒子の力の部分和として格納	各PEが担当するN/P個の粒子に かかる力の情報のみ使用

5.1.1 リスト生成および力の計算

(1) リスト生成

高々 $13+27*N_{PACK}$ 個のサブリスト (4.1.4参照) を、図5.1に示すように分割して、ペアリストを生成する。まず、「同じセル内番号をもつ粒子同士の相互作用」に対して、プロセッサ番号の小さい方から順に、13個のサブリストを割り振ってペアリスト生成処理を行う。次に、「異なるセル内番号をもつ粒子同士の相互作用」に対しては、逆にプロセッサ番号の大きい方から順に、 $27*N_{PACK}$ 個のサブリストを割り振ってペアリストを生成する。

非常に簡単な方法であるが、各プロセッサのペアリストの個数が均等に割り振られるならば、リスト生成処理における各プロセッサのロードバランスは良くなる。

(2) 力の計算

リスト生成処理において各プロセッサに割り振られたペアリストに従って、各プロセッサは打ち切り距離内にある粒子間の力の計算を行う。ここでも、リスト生成処理において、

各プロセッサのペアリストの個数が均等に割り振られるならば、各プロセッサのロードバランスは良くなる。

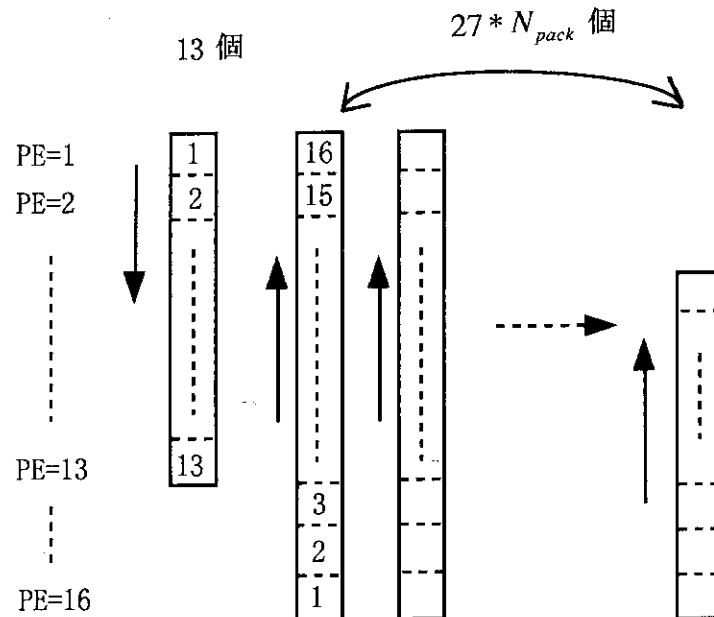


図5.1 サブリストのランダム分割

5.1.2 その他の計算

予測子・修正子計算などの計算においては、 N 個の粒子を各プロセッサに均等に N/P 個ずつ手続き分割する。

位置のデータは各プロセッサが全粒子のデータを所有しているが、各プロセッサが担当する N/P 個のデータのみ更新する。力のデータについて各プロセッサが担当する N/P 個のデータのみ使用する。

位置と力以外の各データ（速度、加速度などのデータ）は、手続き分割に合わせて、各プロセッサに均等に N/P 個ずつデータ分割する。

5.2 通信

本並列アルゴリズムにおいては、位置情報および力の部分和に対して、図5.2および図5.6に示す2種類のグローバル通信を必要とする。

粒子数 N 、プロセッサ数 P のグローバル通信としては、通信回数が $\log_2 P$ で、通信量が

$\frac{P-1}{P}N$ で最適であると思われる expand-fold方式⁶⁾を採用する。この通信アルゴリズムは2のべき乗個のプロセッサ数をもつ場合に最も自然に実現される。

5.2.1 位置情報の通信

ペアリストを作成するサブルーチンSETLSTの前に、サブルーチンSETLSTがCALLされない場合には力の計算を行うサブルーチンEVALの前に、各プロセッサが担当するN/P個の粒子の位置情報を他の全プロセッサに送信して、全プロセッサが全ての粒子の位置情報を所有することが必要となる(図5.2)。図5.3~図5.5に示すexpand方式によりグローバル通信を行う。この方式はプロセッサ数 $P=2^k$ である場合には、k回の通信により実現される。具体的な通信手順を図5.3~図5.5に示す。この例では、プロセッサ数 $P=4$ であるので、2回の通信により行われる。まず、各プロセッサが所有する粒子の位置情報を隣接するプロセッサに送信する(図5.3)。次に、もともと所有していた粒子と前ステップで受信した粒子を合わせて、2つ隣のプロセッサに送信すると(図5.4)、最終的には全プロセッサが全ての粒子の位置情報を所有する(図5.5)。

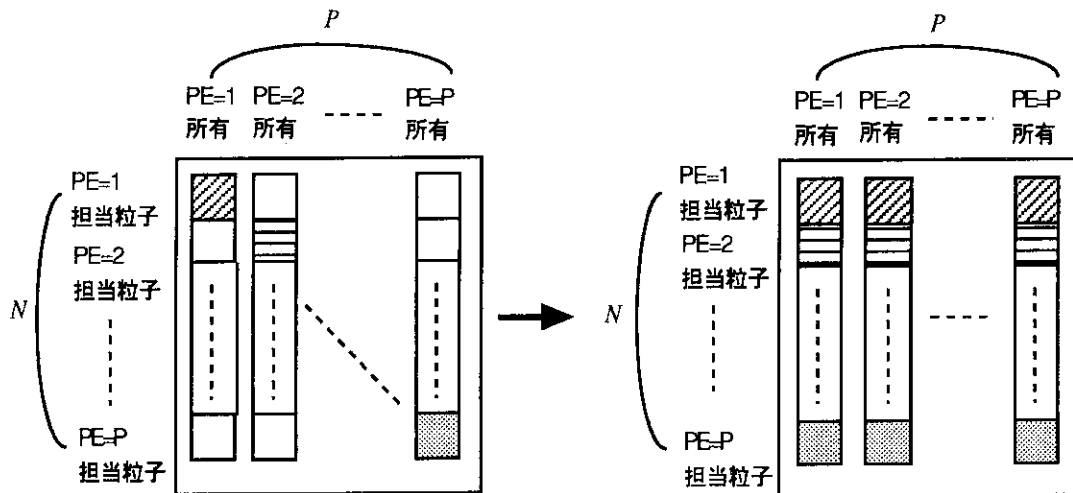


図5.2 位置情報の通信 (expand方式)

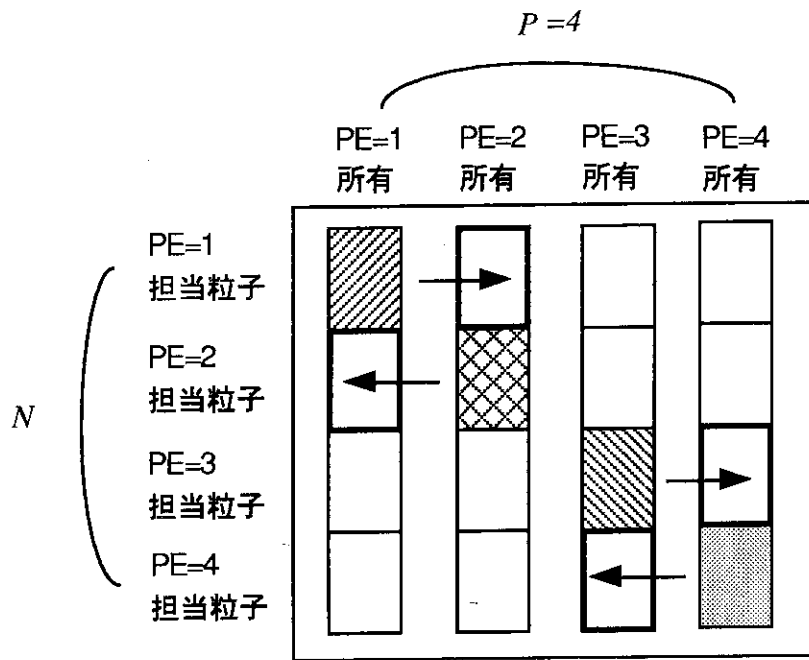


図5.3 expand方式の通信手順その1

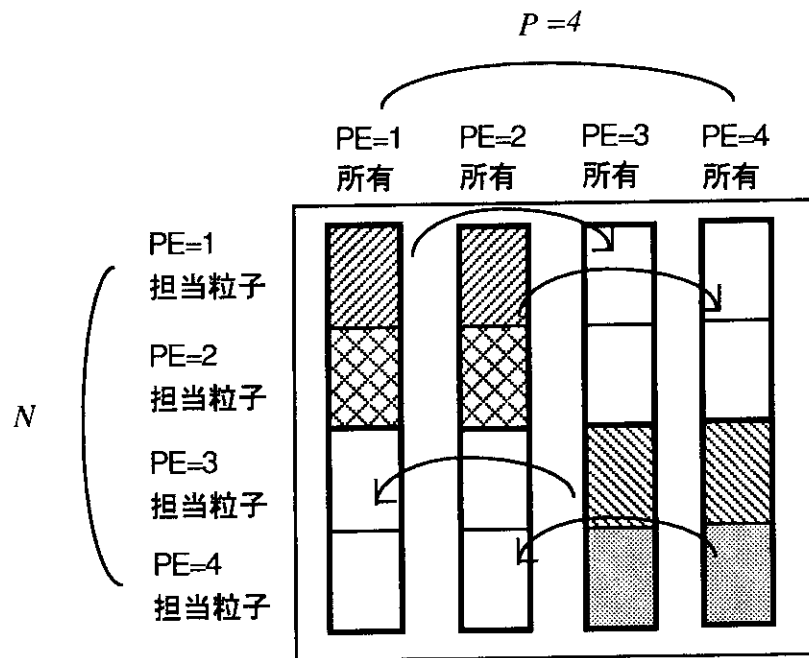


図5.4 expand方式の通信手順その2

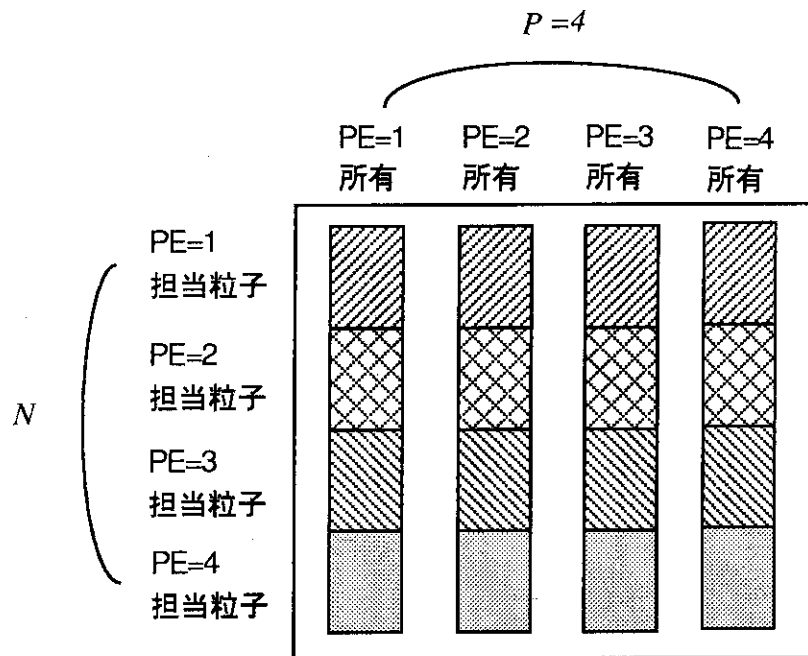


図5.5 expand方式の通信手順その3

5.2.2 力の部分和の通信

力の計算を行うサブルーチンEVALの中で、全粒子に対して、各プロセッサが計算した力の部分和の総和をとる必要がある。そのためには図5.6に示すように、各プロセッサが、担当粒子以外の $\frac{P-1}{P}N$ 個の粒子の力の部分和を、その粒子を担当するプロセッサに通信して、全プロセッサがそのN/P個の担当粒子にかかる力の総和計算を行うことが必要となる。図5.6において、 f_i^j はプロセッサiの担当粒子をプロセッサjが計算した力の部分和で、 \sum_i は(5-1)式で表わされるプロセッサiの担当粒子にかかる力の総和である。

図5.7~図5.9に示すfold方式によりグローバル通信を行う。この方式もプロセッサ数 $P=2^k$ である場合には、k回の通信により実現される。プロセッサ数 $P=4$ の例であるので、2回の通信により行われる。まず、図5.7に示すように力の部分和の情報を2つ隣のプロセッサに送信する。次に、前ステップで受信分の加算を行った後、隣接するプロセッサに相手の担当粒子の力の和の情報を送信する(図5.8)。最後に、各プロセッサが担当粒子の力の総和を求める(図5.9)。

$$\sum_i = \sum_{j=1}^P f_i^j \tag{5-1}$$

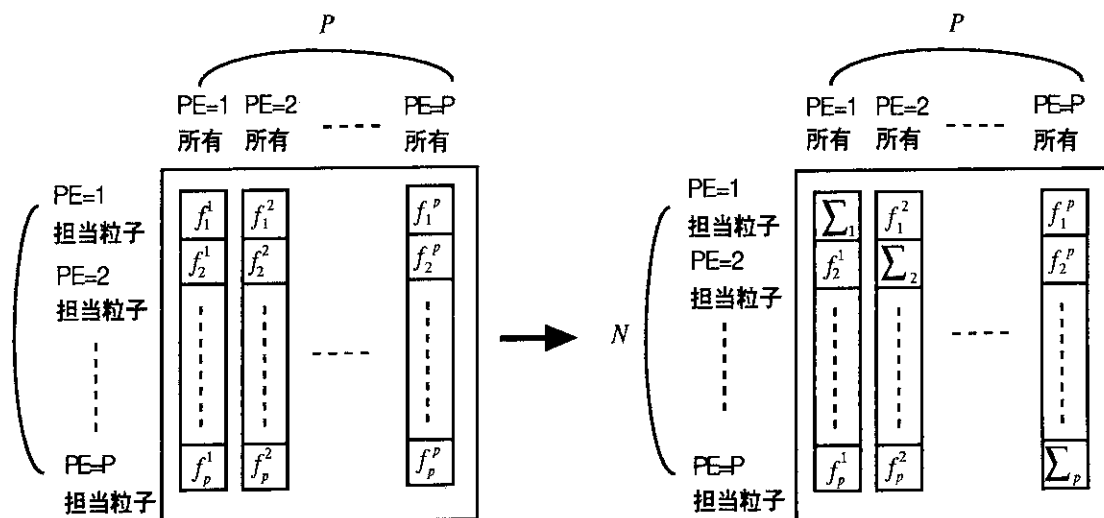


図5.6 力の部分和の通信 (fold方式)

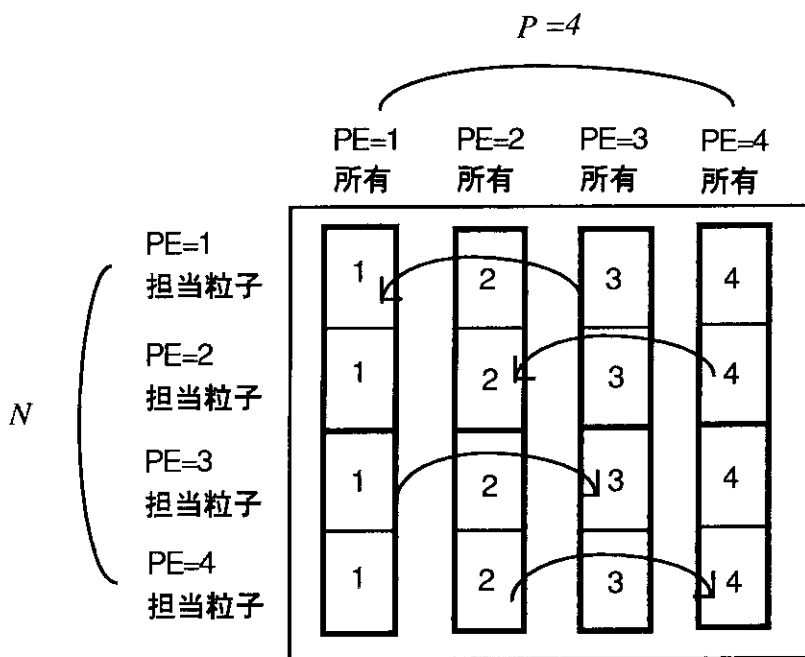


図5.7 fold方式の通信手順その1

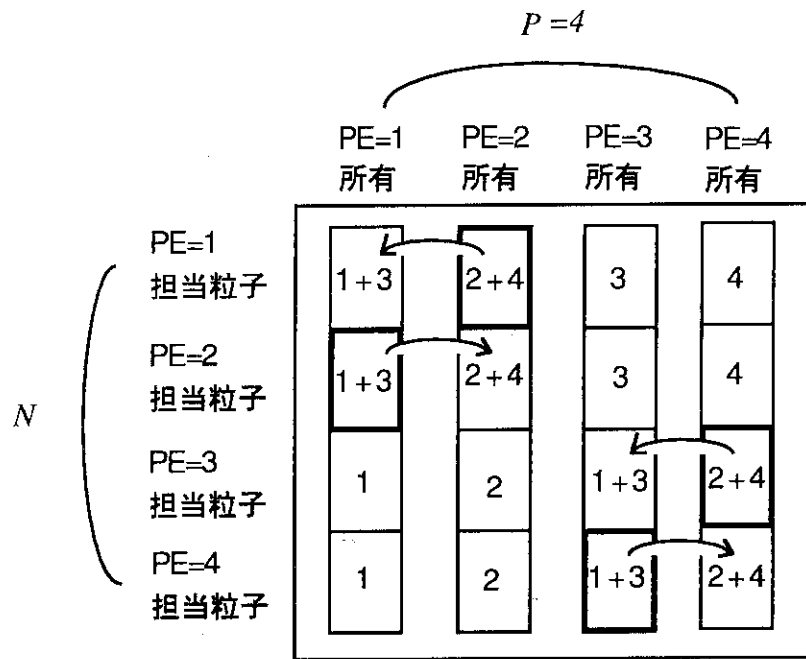


図5.8 fold方式の通信手順その2

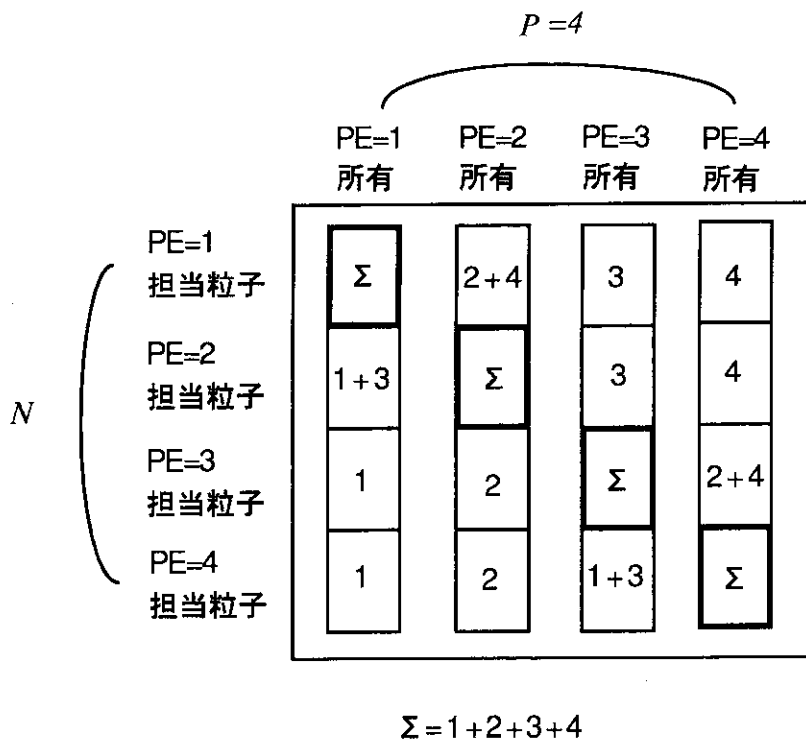


図5.9 fold方式の通信手順その3

6. 並列化性能評価

6.1 計算条件

計算条件を表6.1に示す。プロセッサ数Pとしては、P=1, 2, 4, 8, 16の5ケースに対して計算する。

表6.1 計算条件

意味	入力値
粒子数	4000, 6912, 16384, 32000 の4ケース
平衡計算タイムステップ数	140
本計算タイムステップ数	1000
温度T (無次元化単位系)	1.1
密度 ρ (無次元化単位系)	0.7

6.2 性能評価

本アルゴリズムの並列化性能を、(6-1)式と(6-2)式で定義される速度向上比 S_p (speedup) と並列化効率 η_p (efficiency) を用いて評価する。ここで、 t_1 は1個のプロセッサで処理するのに必要な時間、 t_p はP個のプロセッサで処理するのに必要な時間である。

$$S_p = \frac{t_1}{t_p} \quad (6-1)$$

$$\eta_p = \frac{t_1}{P \times t_p} \quad (6-2)$$

6.2.1 演算の性能評価

演算の並列化性能は、idle時間を含めたロードバランス（負荷分散）の善悪により定まる。ここでは、計算コストの大部分を占めるリスト生成および力の計算に対して、並列化性能評価を行う。

計測はclock関数を用いて、各プロセッサに対して毎タイムステップの計算時間 t_k^i を計測した。プロセッサ数Pにおけるトータル計算時間として、(6-3)式で定義される t_{comp}^p および(6-4)式で定義される t_{comp}^{p+} を算出する。 t_{comp}^p は「各タイムステップにおける全プロセッサに対する最大時間」の全タイムステップを通した積算値、 t_{comp}^{p+} は「各プロセッサの全タイムステップにおける計算時間」の最大値を意味する。ここで、 N_t は計算タイムステップ数である。両者は(6-5)式の大小関係がある。

t_{comp}^p を用いて(6-6)式に従って計算される η_{comp}^p を、ここでは（各プロセッサのロードバランスによって決まる）各演算の並列化効率と呼ぶ。この演算の並列化効率 η_{comp}^p により、演算の性能評価を行う。 t_{comp}^{p+} は各タイムステップにおける演算負荷最大のプロセッサ番号のばらつきを見るために、比較データとして計算する。

$$t_{comp}^p = \sum_{k=1}^{N_t} \max_{1 \leq i \leq P} t_k^i \quad (6-3)$$

$$t_{comp}^{p+} = \max_{1 \leq i \leq P} \sum_{k=1}^{N_t} t_k^i \quad (6-4)$$

$$t_{comp}^{p+} \leq t_{comp}^p \quad (6-5)$$

$$\eta_{comp}^p = \frac{t_{comp}^1}{P \times t_{comp}^p} \quad (6-6)$$

(1) リスト生成

リスト生成の計算時間を表6.2に示す。 t_{comp}^p を上段に、 t_{comp}^{p+} を下段に示す。両者の差は高々1%前後である。

また、各粒子数Nに対する、リスト生成の並列化効率 η_{comp}^p を図6.1に示す。プロセッサ数Pの増加に伴い、リスト生成の並列化効率 η_{comp}^p は単調に減少している。プロセッサ数を固定した場合、各粒子数Nと η_{comp}^p との相関はあまり見られない。プロセッサ数16における η_{comp}^p は0.85~0.88の範囲にある。これは各プロセッサに割り当てられたペアリストの個数がどの位均等に分布しているかを反映している。ペアリストの分割方法には工夫が必要である。

表6.2 リスト生成の計算時間 (単位: sec)

粒子数N	プロセッサ数P				
	1	2	4	8	16
4000	14.273	7.304	3.757	1.953	1.054
		7.302	3.754	1.947	1.048
6912	24.640	12.481	6.337	3.305	1.758
		12.433	6.325	3.278	1.743
16384	46.015	23.316	12.001	6.193	3.364
		23.295	11.983	6.168	3.320
32000	100.366	50.824	26.026	13.425	7.095
		50.786	25.992	13.325	7.067

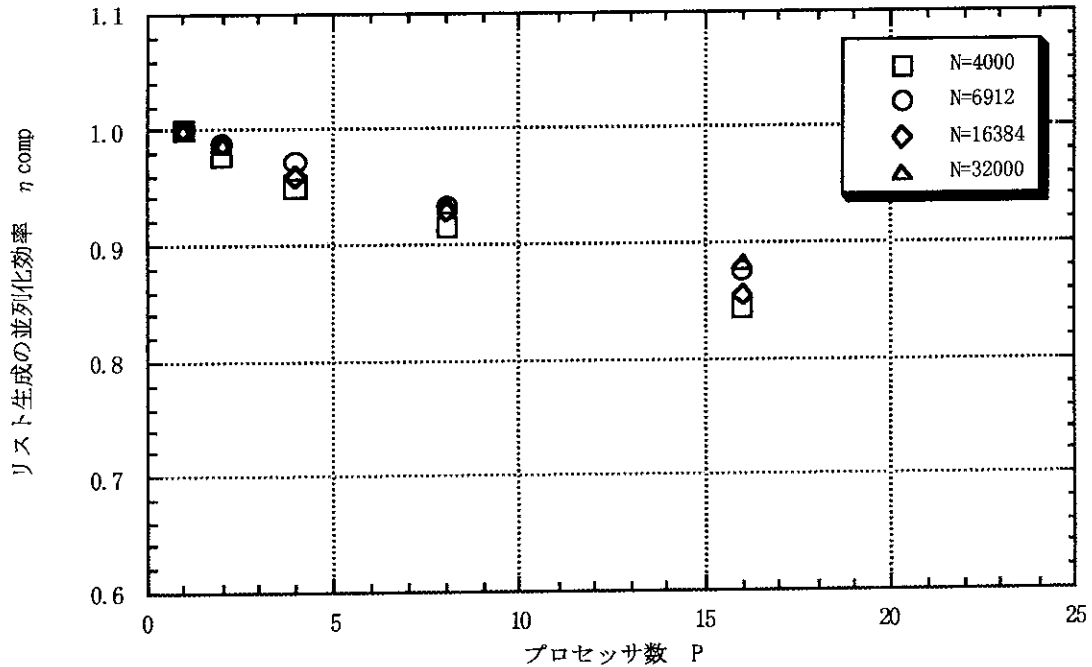


図6.1 リスト生成の並列化効率

(2) 力の計算

力の計算に要する計算時間を表6.3に示す。 t_{comp}^p を上段に、 t_{comp}^{p+} を下段に示す。プロセッサ数8以下では、両者が一致している場合が数多く見られるが、全タイムステップを通じて演算負荷最大のプロセッサ番号が一定になっていることを示す。一方、プロセッサ数16では、両者の差は大きく、粒子数32000個の場合では約6%になっている。これはタイムステップご

とに、負荷最大のプロセッサが大きく変わっているためと思われる。

各粒子数Nに対する、力の計算の並列化効率 η_{comp}^p を図6.2に示す。プロセッサ数Pの増加に伴い、力の計算の並列化効率 η_{comp}^p は単調に減少している。これもリスト生成と同様に、各プロセッサに割り当てられたペアリストの個数がどの位均等に分布しているかに関連しているかと思われる。しかし、リスト生成のロードバランスに比べて悪くなっているのは、打ち切り距離以下のペアの個数のばらつきがペアリストの個数のばらつきに比べてさらに大きくなっているためと思われる。

表6.3 力の計算に要する計算時間 (単位: sec)

粒子数N	プロセッサ数P				
	1	2	4	8	16
4000	30.128	15.822	8.218	4.742	2.640
		15.576	8.218	4.438	2.612
6912	49.391	25.414	13.217	7.051	3.951
		25.365	13.217	7.051	3.700
16384	111.533	57.274	30.912	16.033	9.219
		57.274	30.912	16.033	9.184
32000	215.998	110.069	57.273	30.667	18.253
		110.065	57.273	30.667	17.202

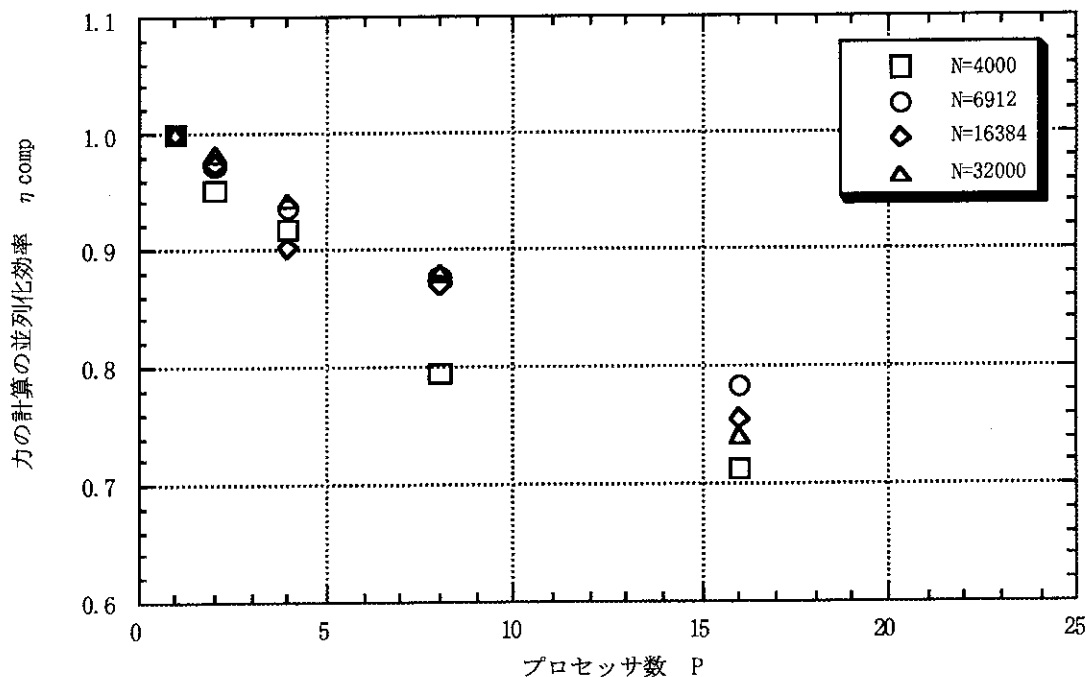


図6.2 力の計算の並列化効率

6.2.2 全体の性能評価

入出力と初期設定を除くトータルの計算時間（表6.4）により、本アルゴリズム全体の性能評価を行う。

各粒子数Nおよび各プロセッサ数Pに対する速度向上比 S_p および並列化効率 η_p を図6.3および図6.4に示す。粒子数Nが増加するに従って、全体の並列化効率は単調に増加している。特に粒子数N=32000、プロセッサ数P=16の場合の全体の並列化効率は約62.8%で、速度向上比は約10.0倍になっている。大規模問題に対しては、全体の並列化効率がより高くなることが期待できる。

プロセッサ数Pを固定した場合の、粒子数Nに対する全体の並列化効率の単調増加性について検討する。その前に、全体の並列化効率 η_p と演算の並列化効率 η_{comp}^p の関係を確認する。全体の並列化効率は (6-7) 式のように表わされる。ここで、 t_{comp}^p はプロセッサ数Pにおける演算コスト、 t_{comm}^p はプロセッサ数Pにおける通信コストである。また、 $f(x)$ は (6-8) 式で表わされる、 $x > 0$ において単調増加な関数である。結局、全体の並列化効率 η_p は、演算の並列化効率 η_{comp}^p と、演算コスト/通信コストの比率 t_{comp}^p/t_{comm}^p に単調増加な関数 $f(t_{comp}^p/t_{comm}^p)$ の積で表わされる。

$$\eta_p = \frac{t_1}{P \times t_p} = \frac{t_{comp}^1}{P \times (t_{comp}^p + t_{comm}^p)} = \eta_{comp}^p \times f(t_{comp}^p/t_{comm}^p) \quad (6-7)$$

$$f(x) = \frac{1}{1 + 1/x} \quad (6-8)$$

プロセッサ数Pを固定した場合の、粒子数Nに対する全体の並列化効率の単調増加性は以下の理由によるものと考えられる。

- ・プロセッサ数Pを固定した場合、粒子数Nが増加するに従って、演算コスト/通信コストの比率 t_{comp}^p/t_{comm}^p が増加する。
- ・演算の並列化効率 η_{comp}^p は6.2.1に示したように、粒子数の増加に伴い変動するが、その変動量は演算コスト/通信コストの比率 t_{comp}^p/t_{comm}^p の増加率に比べて小さい。

表6.4 入出力と初期設定を除くトータルの計算時間（単位：sec）

粒子数N	プロセッサ数P				
	1	2	4	8	16
4000	46.061	24.058	13.387	8.123	5.947
6912	76.230	39.697	21.494	12.550	8.410
16384	161.368	83.939	46.537	26.104	16.771
32000	322.850	167.030	89.822	50.660	32.129

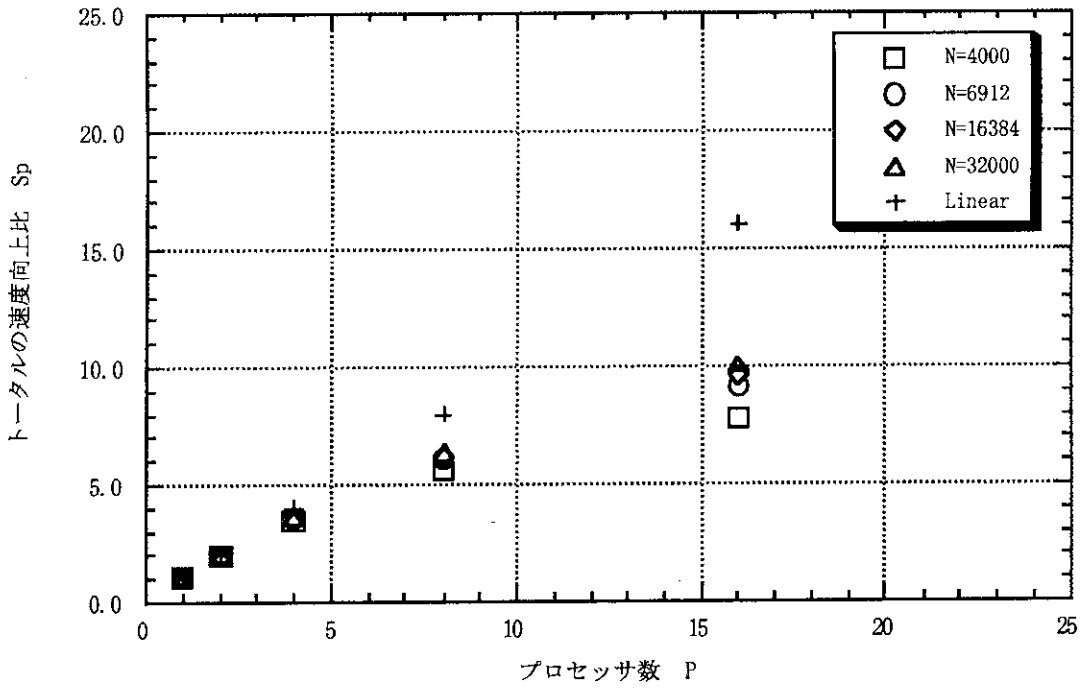


図6.3 入出力と初期設定を除くトータルの速度向上比

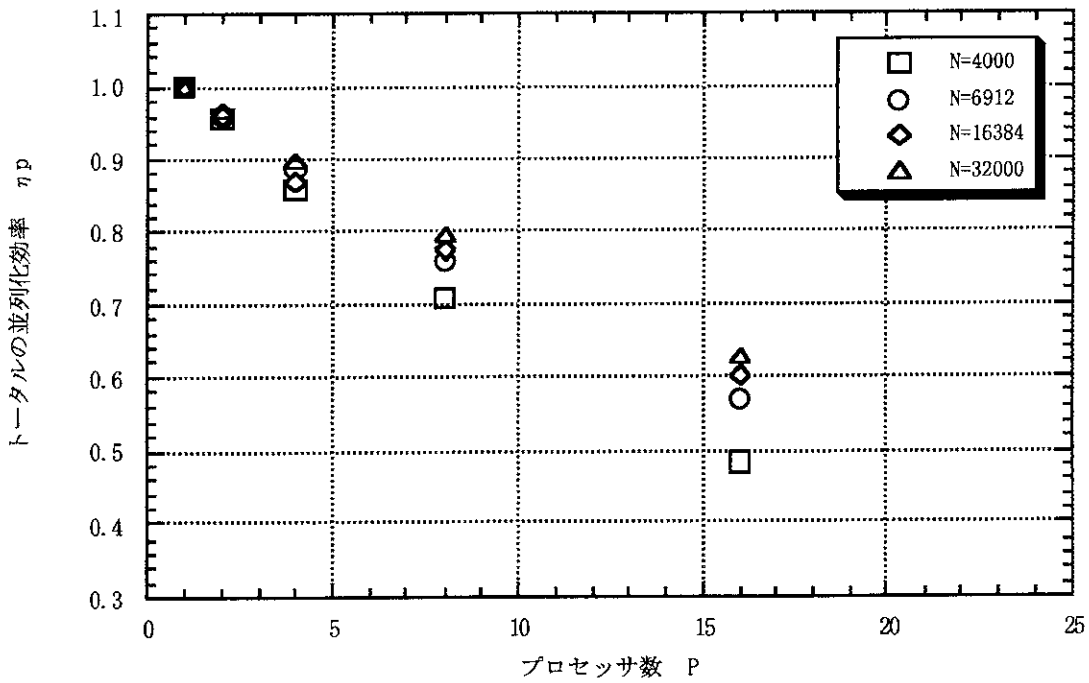


図6.4 入出力と初期設定を除くトータルの並列化効率

7. 結言

本研究では、粒子分割法 (Atomic Decomposition) と、ベクトル化手法であるL.L.C.法をベースにしたサブリスト分割法を組み合わせた並列化手法を採用した。

まず、ベクトル化手法であるL.L.C.法の基本特性を確認した。ベクトル計算時間は、粒子数 N とLink Cellの一辺の長さ $L_{cell}^*(N)$ を用いた性能評価モデルにより、近似的に評価できることがわかった。

次に、並列化アルゴリズムの性能評価を行った。演算については、計算コストの大部分を占めるリスト生成および力の計算の並列化効率 η_{comp}^p を評価した。プロセッサ数 $P=16$ で、リスト生成の並列化効率は84.6~88.4%の範囲に、力の計算の並列化効率は71.3~78.1%の範囲にある。これは各プロセッサのロードバランスがあまり良くないため、ペアリストの分割方法には工夫が必要である。

アルゴリズム全体の並列化性能評価については、粒子数 $N=32000$ でプロセッサ数 $P=16$ の場合の並列化効率は約62.8%で、速度向上比は約10.0倍であった。また、プロセッサ数固定の場合、粒子数 N が増加するに従って、並列化効率も単調に増加することも確認できた。今後、演算コスト t_{comp} および通信コスト t_{comm} を (7-1) 式および (7-2) 式で表わし、定量的に評価する予定である。

$$t_{comp}(N, P) = C_{comp} \frac{N^\alpha}{P^\gamma} (C_{nbd} \times L_{cell}^*(N) - L_c^*)^\beta \quad (7-1)$$

$$t_{comm}(N, P) = C_{comm} N^\delta (\log_2 P)^\epsilon \quad (7-2)$$

本研究では、手始めに粒子分割法によるベクトル並列化手法の性能評価を行ったが、今後、領域分割法 (一次元分割、二次元分割) によるベクトル並列化手法の性能評価を実施していく予定である。ローカル通信により通信コストを削除できることは期待できるが、動的負荷分散法を検討することが課題となる。さらには本研究では実現しなかった演算と通信の同時処理 (overlap) も検討課題と考えられる。

参考文献

- 1) 山本良一：「マテリアルデザイン」，丸善（1988）。
- 2) S. Yip, 私信, (1995) .
- 3) GREST, G. S., Dunweg, B. and KREMER, K. : " VECTORIZED LINK CELL FORTRAN CODE FOR MOLECULAR DYNAMICS SIMULATIONS FOR A LARGE NUMBER OF PARTICLES" , Comp. Phys. Comm., **55**, 269 (1989) .
- 4) 堂山昌男・山本良一：「計算材料科学」，海文堂（1987）。
- 5) Gear, C. W. : " Numerical Initial Value Problem in Ordinary Differential Equations" , Prentice-Hall, Englewood Cliffs (1971) .
- 6) Plimpton, S. : " Fast Parallel Algorithms for Short-Range Molecular Dynamics" , J. Com. Phys., **117**, 1 (1995) .