JAERI-Data/Code
98-020

レベル1・2並列ベンチマーク仕様及びそれに基づくスカラ並列計算機SP2のベンチマークテスト

1998年6月

折居茂夫

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。
入手の問合わせは，日本原子力研究所研究情報部研究情報課（〒319-1195　茨城県那珂郡東海村）あて，お申し越しください。なお，このほかに財団法人原子力弘済会資料センター（〒319-1195　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 Research Information Division, Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, 319-1195, Japan.

編集兼発行　　日本原子力研究所

JAERI-Data/Code 98-020

レベル１・２並列ベンチマーク仕様及びそれに基づくスカラ並列計算機ＳＰ２のベンチマークテスト

日本原子力研究所計算科学技術推進センター
折居　茂夫

(1998年５月18日受理)

数値計算を対象にした並列計算機の性能評価のためのベンチマーク仕様を提案する。従来のコード全体の処理時間を用いて性能を評価するベンチマークをレベル１とし、その性能が如何にして実現されるかを評価するレベル２ベンチマークを新たに設けた。この仕様に基づき、スカラ並列計算機ＳＰ２の性能を並列化分子動力学コードを用いて評価した。その結果、レベル２ベンチマークにより、並列性能を阻害する主な原因が通信のバンド巾、立ち上がり時間の両方から生じていることがわかった。特にこの立ち上がり時間は、プロセッサ数のみならず粒子数にも比例して増加することが、明らかになった。

日本原子力研究所：〒153－0061　東京都目黒区中目黒２－２－54

JAERI-Data/Code 98-020

The Level 1&2 Specification for Parallel Benchmark and a Benchmark Test of Scalar-Parallel Computer SP2 Based on the Specifications

Shigeo ORII

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

(Received May 18, 1998)

A benchmark specification for performance evaluation of parallel computers for numerical analysis is proposed. Level 1 benchmark, which is a conventional type benchmark using processing time, measures performance of computers running a code. Level 2 benchmark proposed in this report is to give the reason of the performance. As an example, scalar-parallel computer SP2 is evaluated with this benchmark specification in case of a molecular dynamics code. As a result, the main causes to suppress the parellel performance are maximum band width and start-up time of communication between nodes. Especially the start-up time is proportional not only to the number of processors but also to the number of particles.

Keywords: Parallel, Parallel Processing, Performance, Performance Evaluation, Scalar Parallel, SP2, MD, Molecular Dynamics, Communication

目　　次

Contents

1.初めに

日本原子力研究所計算科学技術推進センターでは，各々のユーザコードに適した並列計算機を選択する方法を開発して科学技術計算における並列処理の普及を図り，次世代並列計算機の設計に必要な項目を指摘して今後の並列計算機システムの開発に寄与することを目的として，並列計算機用ベンチマークテスト・システムを研究開発している[1]．本報告書は，このシステムにおいて開発された性能評価方法を基にした，並列ベンチマークのルールと仕様を示す．更にこれらの仕様に基づいて実施した，並列化分子動力学コードによるスカラ並列計算機 SP2 のベンチマークテストの手順及び結果を示す．

1.1 背景

クロック周波数の限界による単一プロセッサの性能限界が予測され，性能向上の切り札として並列処理が見直されて幾年かが経過した．この間に多数の並列計算機が製品化され，一部の研究利用においては目覚ましい成果を上げたが，その利用人口は伸び悩んでいるのが現状である．この理由は次の３つに集約できる．

(1)性能の保証が得られない
(2)並列プログラミングを余儀なくされる
(3)ポータビリティがない

(1)はアムダールの法則，即ち全プログラムの処理時間に占める並列処理時間の割合，及び通信時間や同期時間等の並列オーバーヘッドにより生じる．特に並列オーバーヘッドは並列処理特有の現象で，プロセッサ数の増加と共に増加する傾向があるため，台数効果が得られないばかりか，プロセッサ数の増加と共に全体の性能が低下する場合がある．(2)は，性能の台数効果を引き出す自動並列コンパイラが存在しないため，並列性能を出すため MPI や PVM 等の並列ライブラリを使用するか，HPF 等の並列言語または言語拡張を使用することにより生じる．(3)は，自動並列コンパイラが使えないため，種々の並列ライブラリや並列言語等を使うことにより生じる．

アムダールの法則が当てはまるベクトル計算機とパイプライン処理を取り入れたスカラ計算機では，優れた最適化機能を持つ自動コンパイラにより，有る程度 (2)と(3)をクリアできる．このため，これらの計算機ではベンチマークテスト（以後，BMT）を実施することにより(1)の性能保証を比較的容易に調べる事ができる．

一方，並列計算機の性能を引き出せる自動並列コンパイラは，1998 年現在存在しない．このため新しいプログラムの性能を確認する毎に，(2)の理由で，大変な人的労力を必要とする．また，並列化したプログラムが(3)の理由で，他並列計算機で動かず，異なった並列計算機の性能比較が容易でない．その結果，あるユーザコードに有効な並列計算機を選ぶことが容易でない．

このように並列処理普及のボトルネックは，性能を出せる自動並列コンパイラが無いという一事に帰着すると言っても過言でない．また近日中に，このボトルネックの解消は望めない．そこでこのボトルネックの存在を認め，予め並列化したコードを用いて並列 BMT を実施し，(1)の性能保証を議論することが，並列計算機の普及にとっても次世代並列計算機の設計にとっても有意義な事と考える．具体的に(2)と(3)を解決する方法として，ユーザコードを基にし，事実上の業界標準になりつつある MPI を用い並列化した BMT コードを開発し[1]，これを用いて多種の並列計算機の BMT を実施することにした．

並列計算機の処理時間は，プロセッサ数増加と共に減少する並列処理される部分の処理時間及び並列処理できない部分の処理時間と，プロセッサ数増加と共に増加する並列オーバーヘッド時間の和で決まる．この様に相反するもののバランスにより性能が決定する場合，性能を議論するためには定量的な取り扱いが必要となる．定量的な取り扱いを行う場合，今までの多くのベンチマークの様に問題の大きさを固定したものでは不十分であり，処理するデータ量即ち問題の大きさの考慮が必要となる．

更に，実現した性能が妥当なものであるかを評価することが必要となる．計算機の性能を活かし切れない最適化オプションの指定や十分最適化されていない通信ライブラリの使用による評価は，回避されなければならない．

これらを考慮するため並列計算機用性能評価方法を研究開発し[2]，並列計算機用ベンチマークのルールや仕様を確立して本報告書に事例と共に示すことにした．

1.2 並列ベンチマークと性能評価

種々の設計及び性能仕様を持った並列計算機の性能評価を行うための方法として，レベル１ベンチマーク及びレベル２ベンチマーク仕様を作成した．レベル１ベンチマークでは，ユーザコード全体の性能を評価する．レベル２ベンチマークでは，新たに導入した性能評価指標により，ユーザコードのループ及び並列ライブラリ等の性能を評価することができる．

並列処理の普及と次世代並列計算機の設計に必要な項目を指摘するための性能評価は，次の３つの見地で行うことが必要となる[2]．レベル２ベンチマーク仕様により，従来のベンチマーク[3,4]ではできなかった「プログラム開発の見地」と「計算機設計の見地」からの性能評価が可能となる．

・計算機利用の見地
・プログラム開発の見地
・計算機設計の見地

計算機利用の見地からの性能評価は，数ある計算機中から各々のユーザコードに適した並列計算機を選択することが目的となる．そこではプロセッサ数の増加に比例して処理時間がいかに減少するかという台数効果が評価され，ある問題の規模の計算をある時間内に実現できるかという基準時間のクリア等が議論される．また，プロセッサの利用率（並列効率）により評価されることもある．

プログラム開発の見地からの性能評価は，与えられた計算機において最大の性能を出すことが目的となる．そこでは，与えられた計算機で実現しているユーザコード全体の処理時間の妥当性を評価する．これは，各ループ，各並列ライブラリの性能評価により行う．これらの評価を基にして計算時間に寄与している処理の性能要因を特定する．これがアルゴリズムの変更で回避できれば，　各々のユーザコードに適した並列計算機の選択範囲を広げることが可能となる．またアルゴリズムの変更で回避が不可能な場合，次世代並列計算機の設計に必要な項目として揚げることが可能となる．

計算機設計の見地からの性能評価は，性能を悪化する要因を改善する設計をした時にそれがどのくらいBMTコード全体の処理時間の短縮につながり性能が向上されるか等の，ユーザコードと計算機の適合性を評価する．これにより如何なるアーキテクチャ，機能，性能仕様を持つ計算機が必要なのかを検討し，並列計算機が持つ汎用性を明らかにしつつ，次世代並列計算機の設計に必要は項目を指摘することが可能となる．

計算機利用の見地では，種々の並列計算機を用いた性能評価となる．一方，プログラム開発の見地では，１つの並列計算機を対象とした性能評価となる．計算機設計の見地では，これらが混在した性能評価となる．
レベル１とレベル２ベンチマーク仕様により得られた性能評価指標を用い，これら３つの見地からの性能評価が可能となる．

1.3 並列計算機用ベンチマークテスト・システム

BMT コード開発及び，レベル１とレベル２ベンチマークを用いた BMT 実施を行い，さらにこれを公開して普及を図るための仕組みとして，日本原子力研究所では Fig. 1 の並列計算機用ベンチマークテスト・システムを構築した．普及を念頭に置いて，このシステムで開発する BMT コードは利用者コードに基づいている．また，数多い利用者コードの並列化を通して利用分野を分類して開発する．またベンチマークを容易にするため，性能解析ツールを開発している[3]．またこのシステムではインターネットにより次の３つの項目に対して相互通行の情報公開という開かれた仕組みを設け，並列計算機利用の普及を促進する．

(1)BMT 結果，BMT コード，BMT データ，性能評価ツールの公開
(2)公開した BMT コードにより第三者が実施したベンチマーク結果の受け入れと公開
(3)公開した BMT コードに対する第三者の BMT データの受け入れ

(1)により，多数の利用者に性能評価の環境を提供する．(2)により性能評価のクロスチェックが可能となる．また(3)により多数の利用者データに基づいた，性能のデータ依存性が明らかになる．

BMT コード，BMT データ，BMT 結果を蓄積し整理し公開することにより，並列計算機を持たない多数のユーザが，アーキテクチャとピーク性能が異なる並列計算機同士を比較し，各計算機の長所短所を把握することが可能となり，各々のユーザコードに適した並列計算機の選択が可能となる．また，種々の角度から次世代並列計算機の設計に必要な項目を指摘することが可能となる．

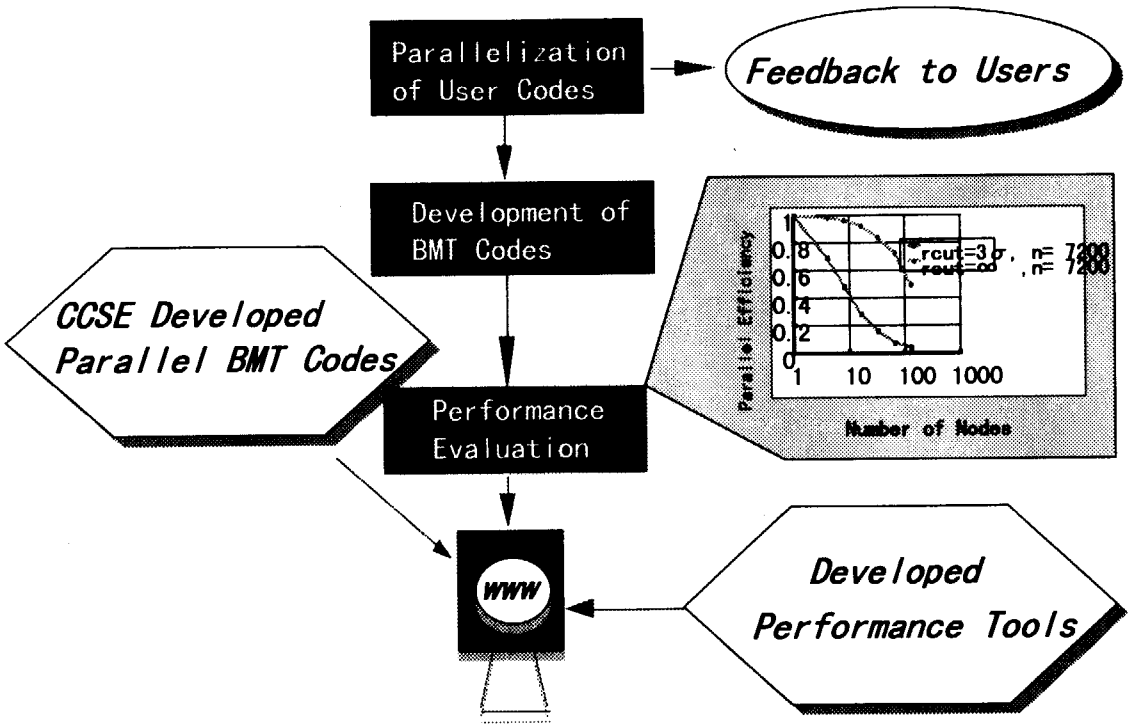

Fig.1 CCSE parallel benchmark test system.

2.レベル1ベンチマーク仕様

(1)与えられたBMT用コードとデータに対し，プロセッサ数pに対するコード全体の処理時間$(\tau)_{TOTAL}$を測定する．
(2)縦軸を$(\tau)_{TOTAL}$横軸をpとするグラフを作成する．
(3)並列効率Epを計算する．
ここに，p：プロセッサ数，$(\tau)_{TOTAL}$：ウォールクロック時間で表わしたコード全体の処理時間(sec)，n：問題の規模，Ep(p,n) 並列効率:=S(p,n)/p，S(p,n)= $\tau(1,n)/\tau(p,n)$.

レベル1ベンチマーク仕様では，BMTコードとデータを特定し，プロセッサ数を変えて処理時間を測定する． 性能評価指標は，この処理時間と並列効率である．

3.レベル2ベンチマーク仕様

(1)与えられたBMT用コードとデータに対し，プロセッサ数pに対する各ループと並列ライブラリの処理時間τを測定する．
(2)与えられたBMT用コードとデータに対し，問題の規模nに対する各ループと並列ライブラリの処理時間τを測定する．
(3)処理時間モデルの作成（既存のモデルが無い場合）
モデルは基本的に$\tau(p,n)=t_0+f(n)/$(効率×ピーク性能）として作成する．

ここに，$\tau(p,n)$：ウォールクロック時間で表わした各ループと並列ライブラリの処理時間(sec)，p：プロセッサ数，n：問題の規模，t_0：立ち上がり時間(sec)，f(n)：ループの場合は四則演算数，並列ライブラリの場合は通信量(MB)，効率：効率を表わすモデル係数(%)，ピーク性能：演算器のピーク(Mflop/s)或は通信のピーク(MB/sec)或は並列処理のピーク（倍）．

(4)測定値と処理時間モデルよりモデル係数決定を決定する．
測定時間と処理時間モデルを回帰分析によりフィッティングし，モデル係数即ち性能評価指標を，各ループと並列ライブラリに対して求める．尚，具体的なフィッティング方法は付録Aを参照されたい．
(5)処理時間モデルとモデル係数の有効性を確認する．
処理時間モデルにモデル係数を代入して$(\tau)_{TOTAL}$を求め，レベル1ベンチマーク仕様による測定値の比較．

レベル２ベンチマークでは，並列化 BMT コードの処理時間を構成する，ループ及び MPI ライブラリに対する性能評価指標を決定する．これらの性能評価指標は，各ループ及び呼び出し場所毎の MPI ライブラリ処理時間を測定し，回帰分析により処理時間モデルにフィッティングして決定する．各ループの処理時間モデルは，各 BMT コードと共に提供されるので原則としてこれを用いる．一方，MPI ライブラリの処理時間モデルは，一部を除いて BMT 実施者が作成する必要が生じる．作成方法は，本報告書の 6.7.1 節あるいは文献[2]を参照されたい．またモデル係数の精度のため，測定では，モデル係数の決定が有意になるようにプロセッサ数，問題の規模を変更して測定することが必要となる．

3.1 レベル 2 のためのツール

レベル２ベンチマークの時間測定のため，時計を自動挿入するツール kpx を開発した．kpx の k ツールではループの処理時間を， m ツールでは MPI ライブラリの処理時間を測定する．

kpx では，ループ，MPI ライブラリ等の測定対象を選択できる．選択は実行パラメータで行う．実行パラメータのデフォルト値では，コードに記述されているほとんどの do ループと MPI ライブラリが測定対象となる．これは，コードの実行時間を決定するループや MPI ライブラリが，一般にコードのどこに存在するか不明なためである．一方，ベンチマークの測定では，測定箇所を限定して測定のオーバーヘッドを減らし，測定精度の向上を図る．この限定は，kpx の実行パラメータを変更することにより，容易に行うことができる． kpx 利用については文献[5,6]を参照されたい．

3.2 ベンチマークルール

(1)ウォールクロック時間で時間測定

単位は秒を用い，測定結果は有効数字4桁以上で記述する．測定結果が統計的に有意になるまで繰り返し測定する．

(2)BMTコード使用のMPIライブラリを他のMPIライブラリで置き換え可

例えばMPI_BCASTをMPI_SENDとMPI_RECVを使って記述する等，MPIライブラリを他のMPIライブラリで置き換えることを可とし，置き換えたソースコードを公開するものとする．またMPIライブラリを置き換えた理由を明記する．

(3)MPIライブラリ以外のBMTコードの変更不可

ソースコードの変更は不可であるが，例外としてコンパイラ指示行挿入は可とし，挿入したソースコードを公開するものとする．挿入したコンパイラ指示行は，特定な計算機上でのみ有効であってはならない．

(4)誰でも入手できるプリプロセッサの使用は可

名称，バージョン，レベル，使用した機能を明記．使用したプリプロセッサは，同一機種ではサイトによらず，機能が実現できる事が保証されなければならい．

CCSEベンチマークテスト・システムが提供するBMTコードは，プログラム開発者が自然に記述するコーディングに近い状態で記述した．たとえばx，y，zのメッシュに関する3重ループを1重ループ化する等の最適化はしていない．これにより，コンパイラによる自動最適化機能を評価する．BMTコードの変更を不可とした理由は，このような単一プロセッサにおける自動最適化の評価が組み込まれているためである．

CCSEベンチマークテスト・システムが提供するBMTコード中のMPIライブラリは，主にMPIライブラリの集団通信を用いた．一般にネットワークの種類により最適通信経路は異なり，各計算機ごとに異なった最適化が必要であるが，そのような最適化は集団通信ライブラリ中に最も隠蔽し易いと考えるからである．残念ながら現在，集団通信のライブラリは必ずしもハードウェアの性能を反映しない．またその状況は各並列計算機によってまちまちである．これを考慮して，MPIライブラリを他のMPIライブラリで置き換え可能とした．

4. ベンチマークテストコード入手方法

BMT コード，レベル２ベンチマークのためのツール kpx は，原研のホームページから，anonymous FTP で提供予定である．

5. ベンチマークテスト結果の公開

原研では，並列処理の普及を図るため，BMT を実施する．更に他のサイトで実施された BMT 結果を受け入れあわせて公開する．公正を規した受け入れのため，受け入れフォーマットを定める．受け入れフォーマットの雛形は，原研のホームページから提供予定である．

6.分子動力学コードMDによるSP2のベンチマークテスト

6.1 ベンチマークコード

ベンチマークコードは，アルゴン原子の熱伝導状態，対流渦の巨視的挙動を解析する２次元の分子動力学コード[7]を粒子分割法で並列化したコード[8]を用いる．mをアルゴン原子の質量，$\mathbf{v}_i$をi番目のアルゴン原子の速度，$\mathbf{r}_i$を位置座標，$\mathbf{F}_i$をi番目の原子に働く力であるとすると，時刻tにおけるi番目のアルゴン原子の運動方程式は次のように書ける．

$$\frac{d\mathbf{v}_i(t)}{dt} = \mathbf{F}_i(t)/m + \mathbf{g}, \qquad \mathbf{v}_i(t) = \frac{d\mathbf{r}_i(t)}{dt} \tag{5.1}$$

ここに，$\mathbf{g}$は重力加速度である．力$\mathbf{F}_i$は２体間ポテンシャル$\phi(\mathbf{r})$より次のように書ける．

$$\mathbf{F}_i = -m\sum_{j\neq i}\frac{\partial\phi(r_{ij})}{\partial\mathbf{r}_i} \tag{5.2}$$

ここに，r_{ij}はi番目の粒子とj番目の粒子との距離$|\mathbf{r}_i - \mathbf{r}_j|$である．
モデルポテンシャル$\phi(\mathbf{r})$はLennard-Jonesポテンシャルである．式(5.3)にこれを示す．
ここに，εとσはそれぞれエネルギー及び長さの次元を持つ量である．

$$\phi(r) = 4\varepsilon\left\{\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right\} \tag{5.3}$$

系を現すマクロな物理量は，矩形領域をサンプリングセルに分割し，セル内の平均物理量として定義する．セルの番号をkとすると，各セルの物理量の総和A_kの平均物理量は式(5.4)で現すことができる．

$$\langle a_k\rangle_t \equiv \frac{1}{t - t_0}\int_{t_0}^{t}\frac{A_k(s)}{N_k(s)}ds \tag{5.4}$$

ここに，N_kは時刻sにおけるセルk内の粒子数である．物理量の総和A_kは，時刻sにおけるセルk内の運動量の総和又はエネルギの総和である．$\langle\cdots\rangle_t$は，t_0からt間の時間平均である．

取り扱う２次元の矩形領域のサンプリングセル数は 40×20，温度差がある拡散反射面で底面と上面を，鏡面反射面で左右を囲まれている． 粒子の初期状態は，空間的に等間隔配置で，初期速度は Maxwell 分布である．

6.2 粒子分割法による並列化コード

分子動力学では， 各々の粒子は自分以外の全ての粒子からの力の影響を受けるため，力の計算回数は粒子数の自乗にほぼ比例する．しかしこのプログラムでは，力の及ぶ範囲を限定する遮蔽距離を導入して計算量の軽減を図るブックキーピィング法を用いるため，力の計算回数は粒子数に比例する程度に減少する．この方法は，一般に粒子 i に対し距離 $\alpha_{cut} \cdot r_{cut}$ 内の粒子番号を記述した表を作成し，ブックキーピィング表に記載されている粒子との距離を計算し，更に遮蔽距離 r_{cut} 内の粒子か否かを判定し，r_{cut} 内の粒子のみ力の計算を行う．この表作成は粒子数の自乗に比例した計算量があるが，粒子の移動を考慮した一定の大きな時間間隔で再作成される．この分子動力学コードでは，r_{cut} は 3σ をとり α_{cut} は 4 としている．

並列化コードは，Fortran で記述され，MPI ライブラリを用いて並列化されている．フローチャートを Fig.2 に示す．ブックキーピィング表により力の計算 force 以外の部分の計算時間の割合が増す．このため，ブックキーピィング表作成の table，式(5.4)の物理量の計算をする propcel が並列化されている．並列化に伴う主な通信は force における力の総和計算と，propcel における式(5.4)の時間平均物理量の総和計算で生じる．

init では，粒子の初期位置を計算する．table は時間ループ中では間隔 n_{table} 毎に呼ばれる．maxwell では初期値として乱数から各粒子の速度を計算する．pcross と chkbnd で境界面を判定し，pmoves で粒子の移動と境界面の反射を計算する．式(5.1)の各粒子の速度の計算は main プログラムで行う． main プログラムでは，時間積分のくり返し回数 Is の制御，サンプリング回数 n_{sample} の制御を行う．

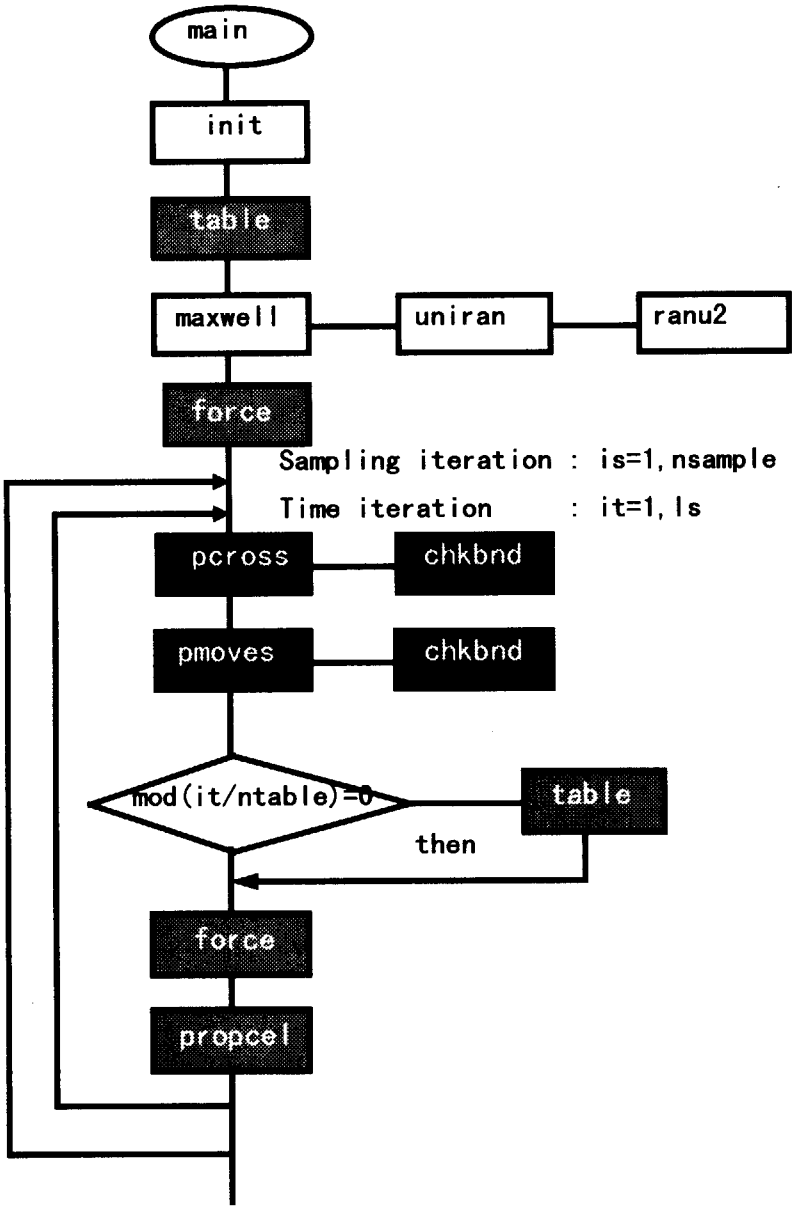

Fig.2 Flowchart of molecular dynamics(MD) code.

6.3 ベンチマークデータ

粒子数に次の 11 のデータを用い，問題の規模を考慮する．

800, 3200, 7200, 12800, 20000, 28800, 39200, 51200, 64800, 80000, 96800

6.4 ベンチマークマシン SP2

6.4.1 システム構成

BMT は，原研の中目黒地区の多段結合型スカラ並列計算機システム SP2 において実施された．このシステムは，並列計算用に 1 フレーム当たり 16 ノードのフレームが 3 つの，計 48 ノードで構成されている．また 1 フレーム当たり 2 ノードのフレームが 1 つあり，この 2 ノードはユーザがログインできるインタラクティブの環境を持つ．SP2 システム構成を Fig.3 に示す．ノードは 9076-SP2．1 ノードの最大論理性能は 1 サイクルに乗加算同時実行 × 2，マシンサイクル 66MHz で，266Mflop/s である．データ・キャッシュは 256KB で 4 重のセット連想写像方式，アクセス時間は 1 クロックである．並列計算用のノードには 128MB のメモリ，インタラクティブ用ノードには 512MB のメモリを持つ．アクセス時間は 10～20 クロックである．各ノード間の通信は 4 段の多段結合方式のスイッチ（ハイパフォーマンス・スイッチ）で接続されている．データ転送速度は 1 ノード当たり双方向で 40MB/秒である．

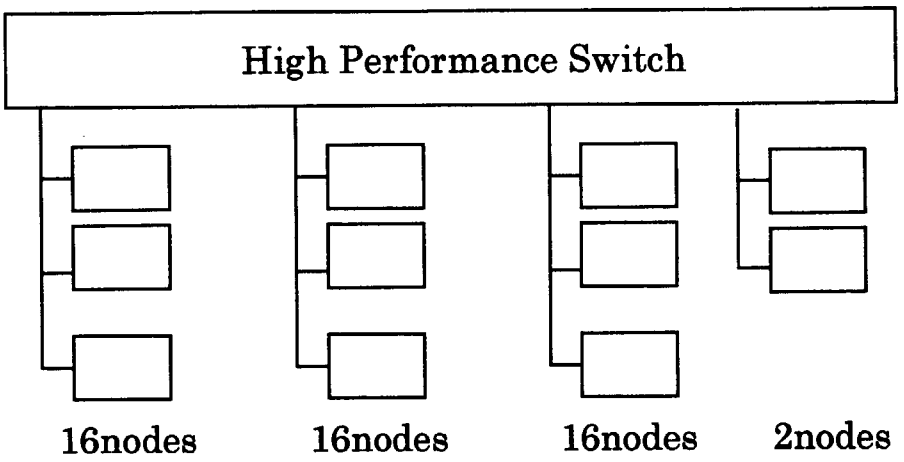

Fig.3 SP2 system configuration at Nakameguro JAERI.

6.4.2 コンパイル

コンパイラは，AIX XL Fortran Version03.02.003.000 を用い，次のコマンドとオプションを用いコンパイルした[9].

```
mpxlf -O3 -qtune=pwr2 -qarch=pwr2 -qstrict
```

6.4.3 実行環境

測定を完了した 1998 年 1 月現在，日本原子力研究所中目黒地区の SP2 システムは，オペレーティングシステム AIX Version4 の下で LoadLeveler の基本機能を用いて並列のジョブスケジュールを行っていた．この機能は，ノードの空き状態を調べ，自動的にジョブの割り付けをする．この機能により，ノードを意識せず，ジョブクラスを指定するだけで SP2 システムを利用することができる．LoadLeveler を用いることにより不特定多数の人がシステムを利用する場合有効に働くが一方，性能評価を実施する場合，次の問題が生じる．

・実行するノードを指定できないため，通信のノード間の定額 40MB/秒と，同一スイッチチップを使用しているノード間の性能 320MB/秒[10]の差を検出するような測定ができない．
・同じジョブクラスで実行してもノードの割り付けに再現性がない

6.4.4 時間測定方法

経過時間を測定する時計は関数 rtc を用いた．

ベンチマークコードで用いている MPI ライブラリの MPI_ALLTOALL は，実行の前後でバリア同期をとらない．そのため並列化された do ループとその直後にある MPI_ALLTOALL を，各々時計を挟んで同時に時間測定すると，MPI_ALLTOALL の測定が do ループのロードインバランスの影響を受けてしまう．すなわち特定のプロセッサにおける do ループの計算時間が長いと，MPI_ALLTOALL において各プロセッサでその特定のプロセッサのデータを待つため，通信時間が長く見えてしまう現象が生じる．これを防ぐため，do ループと MPI ライブラリの時間を別々に測定した．さらに，MPI ライブラリの時間測定の場合は，直前に MPI_BARRIER を call してロードインバランスの影響を排除した．

LoadLeveler による測定のばらつきの対策として，測定は，同一のプロセッサ，粒子数，ロードバランス対策に対して 3 回実施して有意な値を用いることとした．

全測定回数は，15 通りのプロセッサ数×11 通りの粒子数×3 回の測定環境対策×2 回のロードインバランス対策で，990 回となった．

6.5 MPI ライブラリのチューニング

コード中のMPIライブラリをより高速な他のMPIライブラリに置き換えた．並列化MDコードでは，MPI_ALLREDUCE を用いてプロセッサ間の総和計算を行う．この総和計算を MPI_ALLTOALL で実現すると，現在の SP2 の POE(Parallel Operating Environment)2.2 で提供されているMPIでは，より高速な処理が実現できる．そのフローチャートを Fig.4 に示す．プロセッサ数 NP で要素 n を持つ配列 F(n)の総和計算を２回のMPI_ALLTOALLを用いて実現する．ここに，FW は作業配列である．

```
      subroutine all2(n,F,FW,NP)
        real*8 F(n),FW(n)
        include  'mpif.h'

c       call MPI_ALLREDUCE(F,FW,n,MPI_DOUBLE_PRECISION,MPI_SUM,
c      &                   MPI_COMM_WORLD,IERR)

        nLOCAL=(n+NP-1)/NP

        call MPI_ALLTOALL(F ,nLOCAL,MPI_DOUBLE_PRECISION,
       &                  FW,nLOCAL,MPI_DOUBLE_PRECISION,
       &                  MPI_COMM_WORLD,IERR)

        do j=2,NP
          k=(j-1)*nLOCAL
          do i=1,nLOCAL
            FW(i)=FW(i)+FW(i+k)
          end do
        end do

        do j=2,NP
          k=(j-1)*nLOCAL
          do i=1,nLOCAL
            FW(i+k)=FW(i)
          end do
        end do

        call MPI_ALLTOALL(FW, nLOCAL,MPI_DOUBLE_PRECISION,
       &                  F ,nLOCAL,MPI_DOUBLE_PRECISION,
       &                  MPI_COMM_WORLD,IERR)
        return
        end
```

Fig.4 Global summation operation using MPI_ALLTOALL instead of MPI_ALLREDUCE.

6.6 レベル1 ベンチマーク

並列化 MD コードを用いた SP2 のレベル1 ベンチマークテスト結果を Fig.5 及び Fig.6 に示す．経過時間の測定は，15 通りのプロセッサ数(p=1, 2, 4, 6, 8, 10, 12, 14, 16, 20, 24, 30, 36, 42, 48)に対して実施した．粒子数 n=8000 及び 96800 では，メモリ不足でプロセッサ数 p=1 の経過時間を測定していないため，Fig.6 のこれらの並列効率は，p=1 の経過時間を線形外挿で推定して求めた．

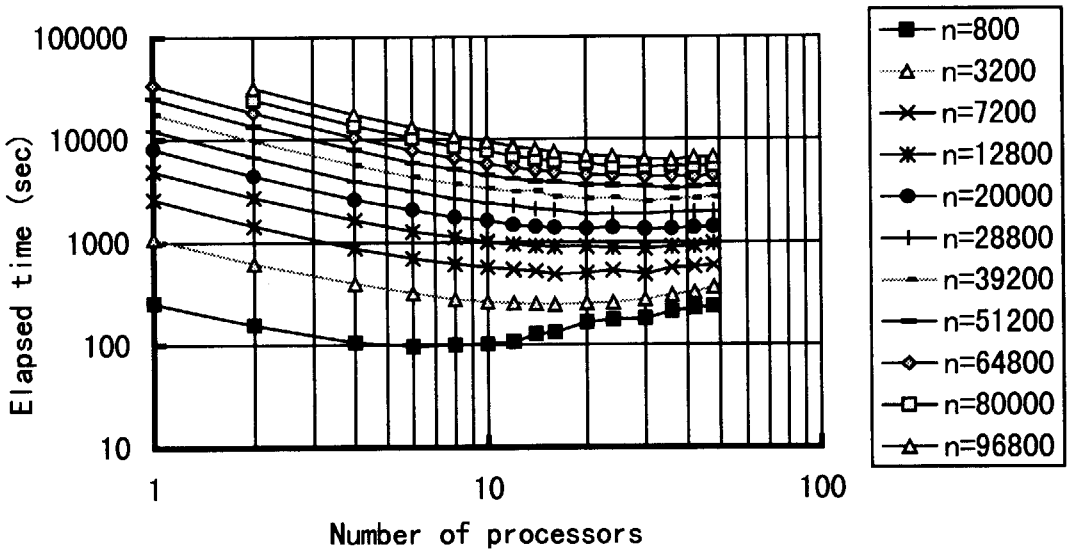

Fig.5 SP2 parallel performance with elapsed time.

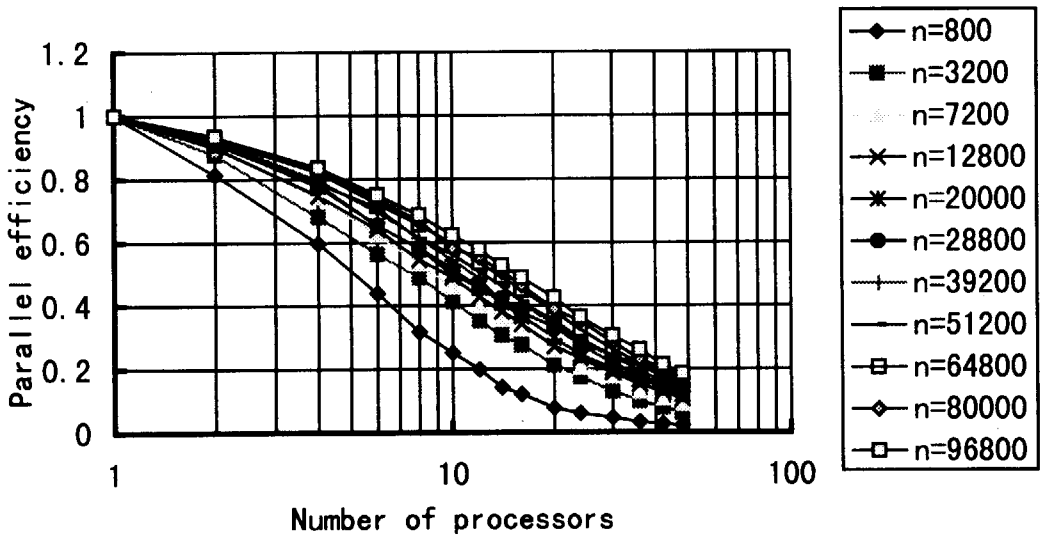

Fig.6 SP2 parallel performance with parallel efficiency.

6.7 レベル2ベンチマーク

6.7.1 処理時間モデル

Table 1に4.1節で述べたレベル2ベンチマーク仕様に従って作成した並列MDコードの処理時間モデルを示す．粒子数をn，プロセッサ数をpで表わす．表の左端の列にサブルーチン名，次の列は，f：四則演算数（種類は数字の添字に表示），t_u：p=1の処理時間，t_p：p>1の処理時間，σ：並列オーバーヘッドである．コードがスタートしてから終了するまでの各サブルーチンの時間が(τ_p)である．コード全体の時間は$(\tau)_{TOTAL}$である．記号に付いている添字は，サブルーチン名，doループ番号を表わす．数字に付いている添字は，カウントした四則演算の種類を表わす．内部関数が出現した時は，四則演算と同等にカウントしてその関数名を添字として記述した．さらに，if文中の演算を区別するため，括弧で括って‘if’を添字とした．n，pを変数とし，処理時間の実測及びその値とモデルを回帰分析でフィッティングし，モデル係数即ち性能評価指標t_{0u}，a_u，t_{0p}，e_p，t_{0c}，e_c，t_{0sum}を求めることが，レベル2ベンチマークである．これらの指標は順に，doループ立ち上がり時間，doループの演算器使用効率(=実測時間/四則演算数 Mflop/公表性能値 Mflop/s)，doループの並列立ち上がり時間，doループの並列効率，通信立ち上がり時間，通信効率（=実測時間/通信量MB/通信バンド巾MB/s)，通信中に行われた総和計算のための四則演算時間である．詳しくは文献[2]を参照されたい．また，σを除いた処理時間モデルの構築方法もその文献を参照されたい．

ここでは，処理時間モデル$\sigma = 2\cdot\{2\cdot[t_{0c}\cdot p + n\cdot(p-1)/p\cdot X_8/(r_c\cdot e_c)] + t_{0sum}\}$の構築方法について述べる．このモデルはFig.4のコーディングに基づいて構築する．まずMPI_ALLTOALLに挟まれた総和計算のための四則演算時間はnに比例する．そこでこれを$n\cdot t_{0sum}$で表わす．次にMPI_ALLTOALLはデータを均等にn/pに分割して自分以外の全てのプロセッサにデータを送り，2回実行されるため$n\cdot\{2\cdot(p-1)/p\cdot X_8/(r_c\cdot e_c)\}$で表わす．ここに$X_8$はデータ1要素が8バイトであるり，$r_c$はノード間通信のバンド巾で40MB/秒である．$t_{0c}$は通信の立ち上がり時間で，主にパケット通信のためのヘッダを作る時間である．2が左乗されているのは，x方向とy方向に2回の計算が行われるためである．

Table 1 Timing model of a parallelized molecular dynamics program.

(force)	f_{force}	$= (n\cdot(n-1)/2\cdot\{C_1\cdot[3_{\pm}+2_{*}+C_2\cdot(1_{/}+5_{\pm}+7_{*})_{if}]_{271}\}_{270}$
		$= (n-1)/2\cdot[C_1\cdot n\cdot(5+13\cdot C_2)]$
	t_u	$= t_{0u270}+(n-1)\cdot\{t_{0u271}+C_1\cdot n/2\cdot[(5+13\cdot C_2)/(r_a\cdot a_{u271})]\}$
	t_p	$= t_{0p270}+t_u/(p\cdot e_{p270})+\sigma_{270}$
	σ_{270}	$= 2\cdot\{2\cdot[t_{0c270}\cdot p+n\cdot(p-1)/p\cdot X_8/(r_c\cdot e_{c270})]+t_{0sum}\}$
	$(\tau_p)_{force}$	$= (n_{sample}\cdot I_s+1)\cdot t_p$
(table)	f_{table}	$= (n\cdot(n-1)/2)\cdot[(3_{\pm}+2_{*}+C_1\cdot(1_{+})_{if}]_{101}$
	t_u	$= t_{0u101}+(n-1)\cdot[t_{0u100}+n/2\cdot[(5+C_1)/(r_a\cdot a_{u100})]]$
	t_p	$= t_{0p100}+t_u/(p\cdot e_{p101})$
	$(\tau_p)_{table}$	$= (n_{sample}\cdot I_s/n_{table}+1)\cdot t_p$
(propcel)	$f_{propcel}$	$= n\cdot[2_{\max}+2_{\min}+2_{\text{int}}+2_{*}+2_{+})_{302}+(5_{+}+2_{*})_{303}]$
	t_{u302}	$= t_{0u302}+10\cdot n/(r_a\cdot a_{u302})]$
	t_{p302}	$= t_{0p302}+t_{u302}/(p\cdot e_{p302})$
	t_{u303}	$= t_{0u303}+7\cdot n/(r_a\cdot a_{u303})]$
	t_{p303}	$= t_{0p303}+t_{u303}/(p\cdot e_{p303})$
	t_u	$= t_{u302}+t_{u303}$
	t_p	$= t_{p302}+t_{p303}+\sigma_{303}+C_3\cdot\sigma_{306}$
	σ_{303}	$= \{2\cdot[t_{0c306}\cdot p+(n_x\cdot n_y)\cdot(p-1)/p\cdot X_8/(r_c\cdot e_{c306})]+t_{0sum}\}$
	σ_{306}	$= 3\cdot\{2\cdot[t_{0c306}\cdot p+(n_x\cdot n_y)\cdot(p-1)/p\cdot X_8/(r_c\cdot e_{c306})]+t_{0sum}\}$
	$(\tau_p)_{propcel}$	$= n_{sample}\cdot I_s\cdot t_p$
(pcross)	f_{pcross}	$= n\cdot[(3_{+}+2_{*}+\{[1_{/}+C_4\cdot 1_{*}+(1-C_4)\cdot(1_{-}+1_{*})]_{if}$ $+[1_{/}+C_5\cdot 1_{+}+(1-C_5)\cdot(1_{-}+1_{*})]_{if}\}_{chkbnd}]_{10}$
	t_u	$= t_{0u10}+10\cdot n/(r_a\cdot a_{u10})]$
	$(\tau_u)_{pcross}$	$= n_{sample}\cdot I_s\cdot t_u$
(pmoves)	f_{pmoves}	$= n\cdot(C_6\cdot(5_{+}+4_{*})_{if}+[(1-C_6)\cdot(1_{+})_{if}]_{300}$
	t_u	$= t_{0u300}+9\cdot n/(r_a\cdot a_{u300})]$
	$(\tau_u)_{pmoves}$	$= n_{sample}\cdot I_s\cdot t_u$
[TOTAL]	F_{total}	$= n_{sample}\cdot I_s\cdot(f_{force}+f_{table}/n_{table}+f_{propcel}+f_{pcross}+f_{pmoves})+f_{force}+f_{table}$
	$(\tau)_{TOTAL}$	$= (\tau_p)_{force}+(\tau_p)_{table}+(\tau_p)_{propcel}+(\tau_u)_{pcross}+(\tau_u)_{pmoves}$

α_{cut}：4，r_{cut}：3，n_{sample}：6，I_s：2000，n_{table}：$14.16\cdot\alpha_{cut}\cdot r_{cut}$，$n_x$：40，$n_y$：20，$X_8$：8バイト，$X_4$：4バイト，$C_1$：$\pi\cdot(\alpha_{cut}\cdot r_{cut})^2$/系の面積$=\pi\cdot(\alpha_{cut}\cdot r_{cut})^2/(2.5\cdot n)$，$C_2$：$1/\alpha^2_{cut}$，$C_3$：$1/I_s$，$C_4$：0.5（x方向の粒子速度が正である確率），$C_5$：0.5（y方向の粒子速度が正である確率），$C_6$：~1（境界と接触しない粒子の割合）

6.7.2 ベンチマークテストの結果

測定時間とTable 1に示した処理時間モデルを回帰分析によりフッティングしてモデル係数を求めた結果をTable 2に示す．モデル係数決定方法は付録Aに示す．

Table 2 Value of model parameters of SP2.

(name) loop	t_{0u} (sec)	a_u	t_{0p} (μ sec)	e_p	t_{0c} (n sec)	e_c	t_{0sum} (n sec)
(force)							
do270	-0.0143	—	$1.19\cdot10^{3}+0.2671\cdot10^{-7}\cdot n$	1.0	$8.6\cdot n$	0.61	0.98
do271	0	0.071	—	—	—	—	—
(table)							
do101	0.45	—	$0.120\cdot10^{-9}\cdot n^{2}$	0.89	—	—	—
do100	-97 μ	0.260	—	—	—	—	—
(propcel)							
do302	56 μ	0.161	0	0.976	—	—	—
do303	-32 μ	0.122	0	1.0	$61\cdot n_x\cdot n_y$	0.34	0
(pcross)							
do10	-0.00025	0.0450	—	—	—	—	—
(pmoves)							
do300	-0.00065	0.117	—	—	—	—	—

これらのモデル係数がMDコード全体の性能を反映することは，Fig.7の処理時間の粒子数依存性と，Fig.8の処理時間のプロセッサ数依存性から確認できる．Fig.7aはプロセッサ数p=1，Fig.7bはp=48のコード全体の処理時間とそれを構成するサブルーチンの処理時間を示す．p=1ではforceとtableが，p=48ではforce，propcel，table，pmovesがコード全体の処理時間を決定する．

Fig.8aは粒子数n=7200，Fig.8bは粒子数n=96800のコード全体の処理時間とそれを構成するサブルーチンの処理時間を示す．n=7200ではforce，table，pcrossとpropcelが，n=96800ではforce，table，pcrossがコード全体の処理時間を決定する．
サブルーチンの処理時間を構成する要素は，forceのdo270とプロセッサ間の総和である．これをFib.9aに示す．propcelはdo302，do303とdo303のプロセッサ間の総和である．これをFib.9bに示す．

時間は，コードがスタートしてから終了するまでの間，これらのdoループが実行する毎の処理時間を積算し，各プロセッサ毎で測定する．

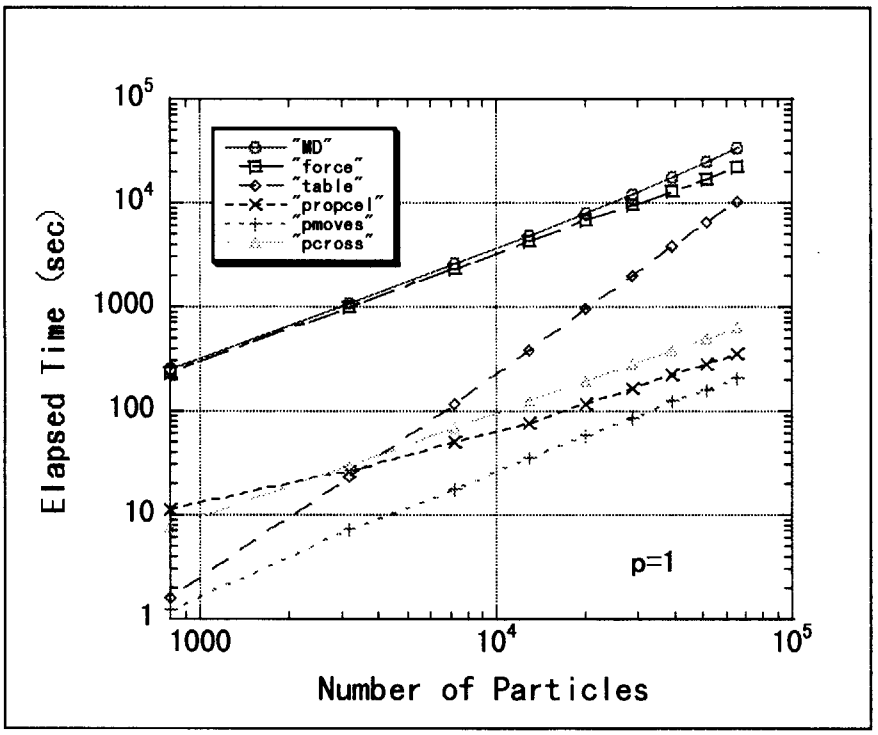

Fig. 7a Processing time(p=1) of MD code and subroutines.

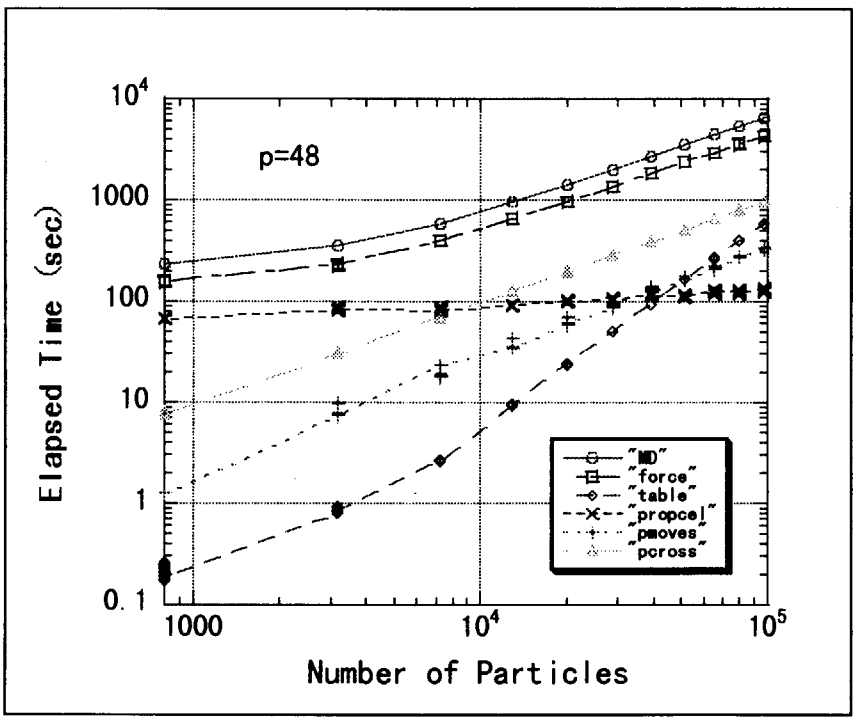

Fig. 7b Processing time(p=48) of MD code and subroutines.

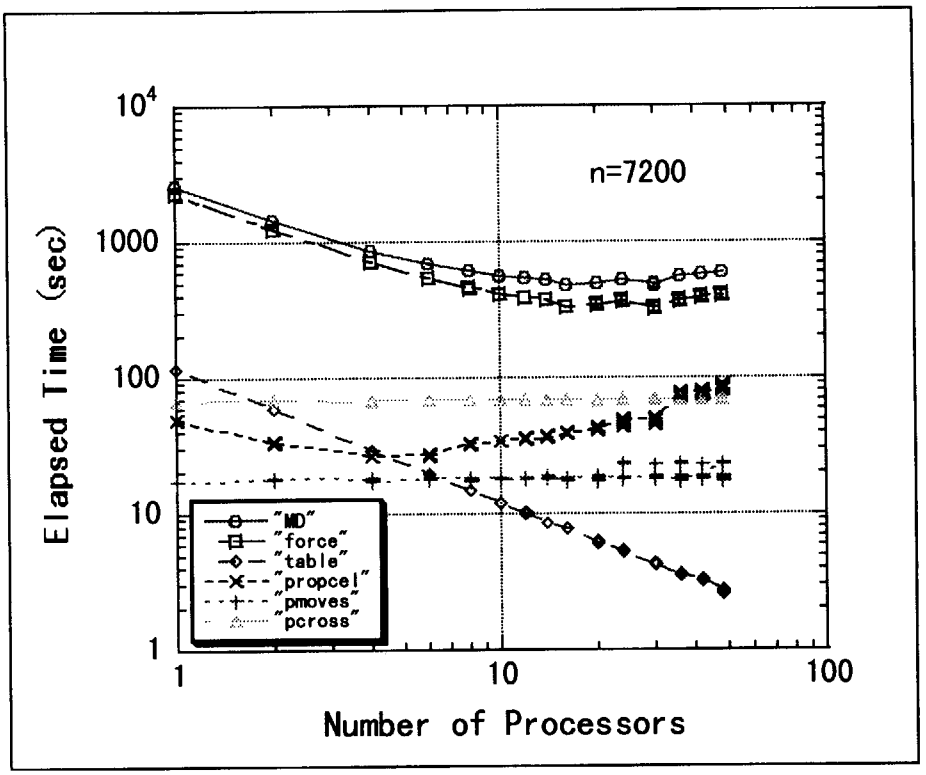

Fig. 8a Processing time(n=7200) of MD code and subroutines.

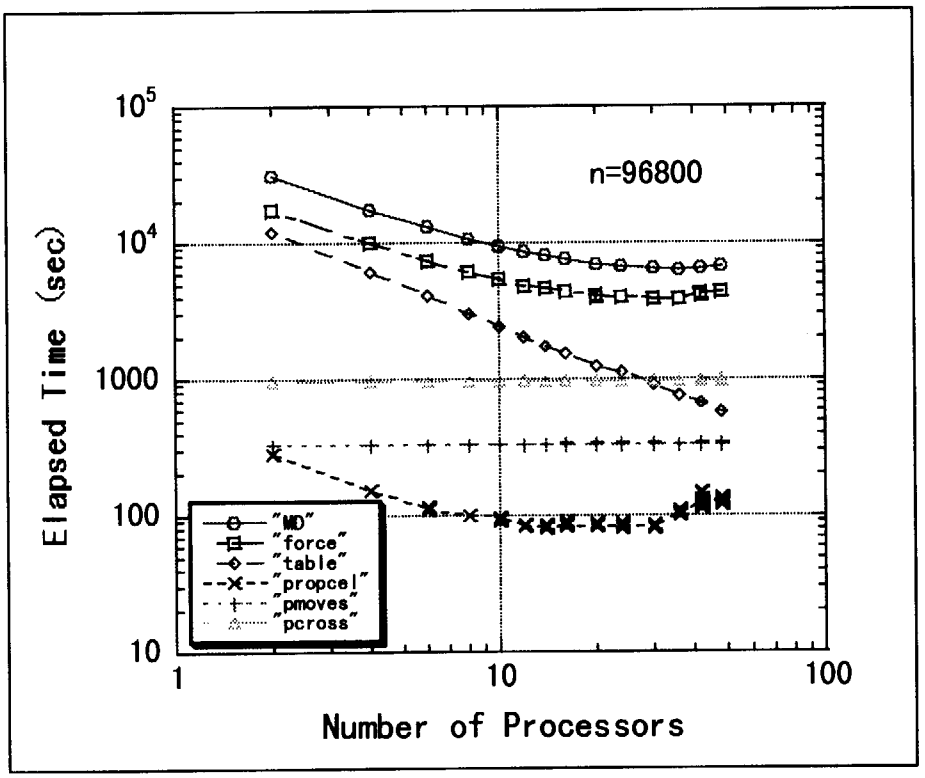

Fig. 8b Processing time(n=96800) of MD code and subroutines.

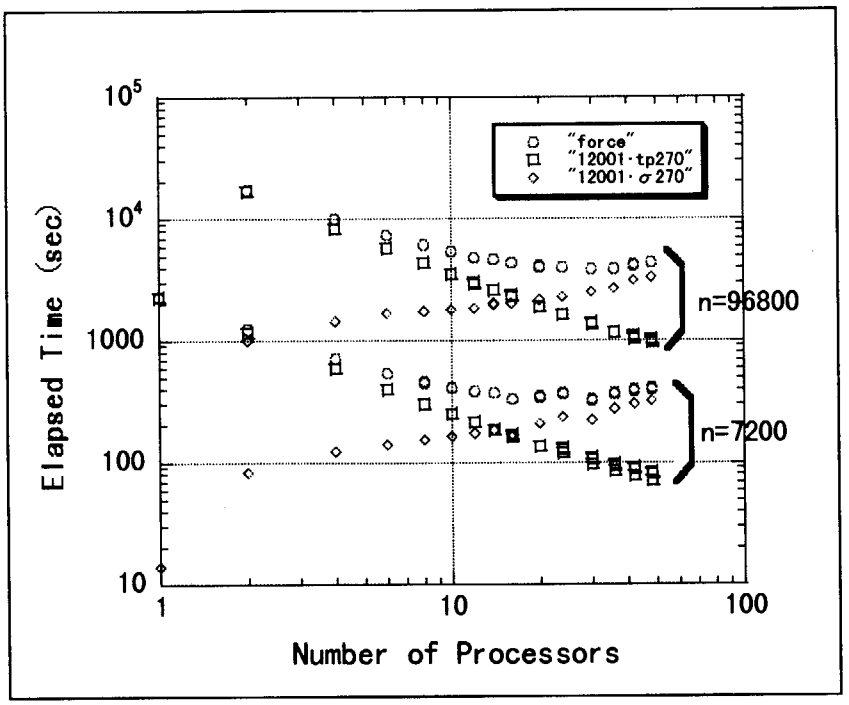

Fig. 9a Processing time of force, do270 and σ_{270}.

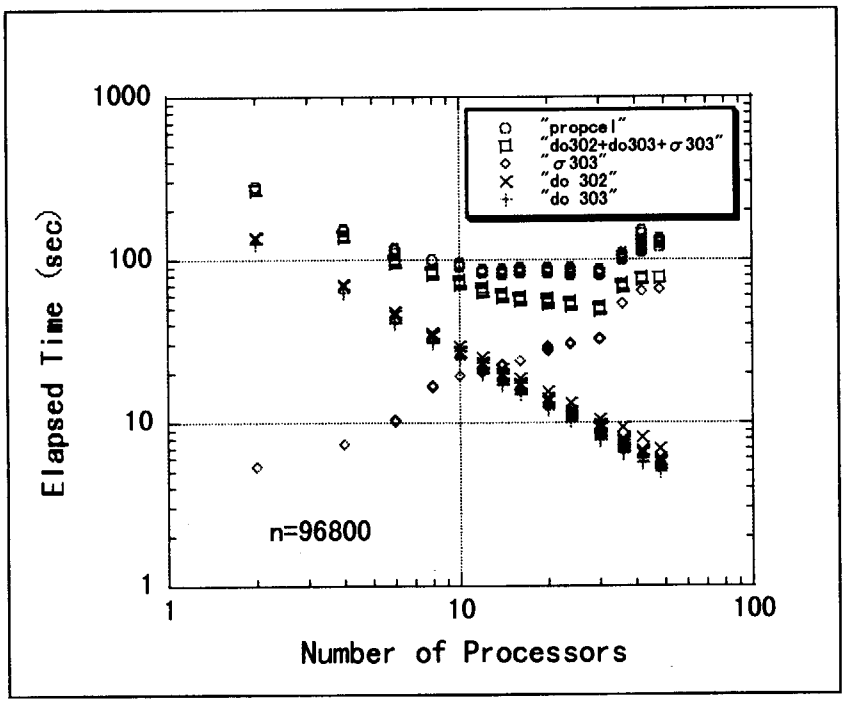

Fig. 9b Processing time of propcel, do302, do303 and σ_{303}.

モデルがどのぐらいコードの処理時間を記述しているかを評価するため，相対誤差を Fig.10 に示した．Fig.10a は粒子数に対する誤差である．n=800 の相対誤差は大きいが，それ以外は約 20%以下である．幾分プラス側にあることは全ての処理をモデル化していないためであるが，測定の揺らぎに対しては十分な精度である．同じ粒子数に対して複数のプロットデータが存在するのは，各プロセッサ毎の測定値をすべて使用してプロットしたためである．例えば 48 プロセッサで測定した場合，48 データをプロットした．1 プロセッサで測定した場合，1 データをプロットした．

Fig.10b はプロセッサ数の対する相対誤差である．相対誤差はプロセッサ数の増加に対してわずかに右肩上がりであるが，Fig.10a を参照して n=800 の寄与を差し引くと，相対誤差は約 20%以下である．

これらの図は，コード全体の処理時間が，n=800 より大きい問題の規模で，相対誤差約 20%以下でモデル化されていることを示す．

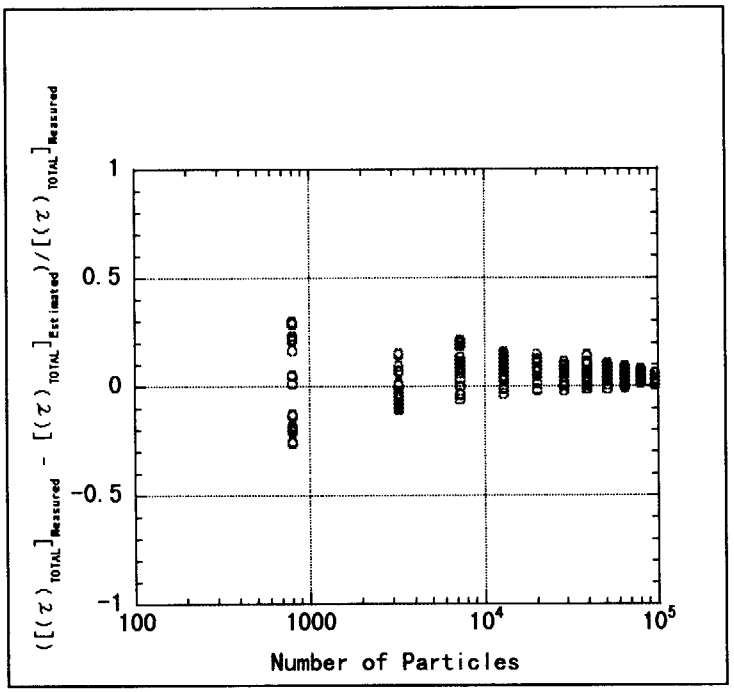

Fig.10a Accuracy of timing model of MD code with n.

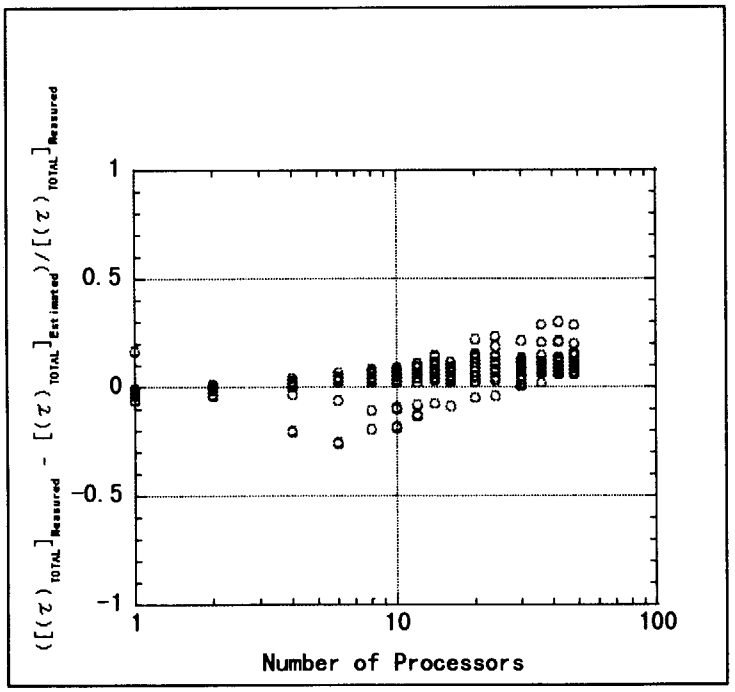

Fig.10b Accuracy of timing model of MD code with p.

7.議論

レベル1ベンチマークのFig.6によれば，粒子数800はプロセッサ数5で並列効率50%となり，粒子数96800ではプロセッサ数17で並列効率50%となる．

レベル2ベンチマークにおいて，Fig.7a，Fig.7b，Fig.8aとFig.8bによれば，サブルーチンforce, propcel, table, pmovesがコード全体の処理時間を決定することがわかった．その中でもforceの占める割合が大きい．forceの処理時間はプロセッサ数が増えるに従い計算部分の処理時間が減り，通信を含む総和の処理時間σ_{270}が増加する．Fig.9aを見ると，プロセッサ数48ではforceの処理時間のほとんどがσ_{270}である．この現象は各粒子数に共通した現象である．

Table 2を見るとt_{0c}はnに比例しているため，$\sigma=2\cdot n\cdot\{2\cdot[t_{0c}'\cdot p+(p-1)/p\cdot X_8/(r_c\cdot e_c)]+t_{0sum}\}$とnで括ることができる．ここに$t_{0c}'=t_{0c}/n$である．そこで[・・・]内の係数をTable 2から計算し順にならべると$17.2\cdot p : 656\cdot(p-1)/p : 98$となる．従って，プロセッサ数が少ない場合は第2項のバンド巾に制限された通信時間と第3項の総和計算が効いている．プロセッサ数が増すと，第1項の立ち上がり時間t_{0c}がプロセッサ数pに比例しているため効くようになる．

t_{0c}がnとpに比例し，特にnに比例するということは，本ベンチマークの結果として観測されたことである．この現象が今回の測定条件により生じたことなのか，SP2システム固有の現象か，並列計算機の通信で共通した問題かを調べることが今後の課題である．なぜなら，十分なデータ量をブロック転送すれば立ち上がり時間が無視できるという，通常の性能評価方法で用いる仮定が使えなくなるためである．

t_{0c}がpに比例するということは，パケットヘッダ作成等の準備がプロセッサ数pに比例すると捉えることができる．一方今回観測されたt_{0c}がnに比例するということは，生成されるパケットの量が多い，1つのパケットの生成に時間がかかる等が考えられる．

この現象は，プロセッサ数が48台であったために検出された．レベル2ベンチマークは，十分なプロセッサ数で実施する必要があると考える．

これらの現象が生じたシステムの測定上の問題点は，LoadLevelerで運用しているシステムで測定を実施したため，ジョブを特定ノードに割り付けられないことにある．SP2システムの通信は多段結合であるため，本来ノードの組み合わせにより通信性能が異なるはずである．これらの効果を調べるためには，ジョブを特定ノードに割り付けて測定する必要がある．しかしLoadLevelerで運用しているシステムではこのような測定はできない．またFig.3に示した各フレーム毎に存在するスレーブサーバのノード1，17，33に並列ジョブが割り当てられると経過時間が10%長くなる場合があるが，これらのノードの回避した測定もLoadLevelerで運用しているシステムでは容易でない．

これらの問題は，POE(Parallel Operating Environment)でシステムを運用することにより，解決できる．POE ではホストリストファイルを用いてジョブを実行するノードを決めることができる．今後，POE による測定が必要と考える．

force の do270 は do271 とネストしている外側の do ループである．Table 1 に示す $t_u = t_{0u270} + (n-1)\cdot\{t_{0u271} + C_1 \cdot n/2 \cdot [(5+13\cdot C_2)/(r_a \cdot a_{u271})]\}$ の各項のオーダを，Table 2 の t_{0u270}=-0.0143，a_{u271}=0.071，t_{ou271}=0 を用いて比較すると，順に-0.0143 : 0 : $2.4\cdot10^{-5}$ n となる．粒子数 n が 1000 であると-0.0143 : 0 : 0.024 となり計算時間 t_u が短縮される．一方問題の規模が大きい n=100000 の場合，t_{0u} が負値である効果は無視できる．従って，モデルが，問題の規模が小さくなるに従い flop/s 性能が向上する現象を，立ち上がり時間が負になる形で捉えたと言える．この現象は，問題の規模が小さい場合キャッシュラインの有効利用が強調されるために生じていると捉えると，説明しやすい．t_{0u} が負値となるこの現象は，すべてのサブルーチンで生じており，今後，個々の do ループのデータアクセスパターンと問題の規模 n を関連付けた現象の解析が必要と考える．

8.まとめ

数値計算を行う並列計算機の性能評価のため，「レベル１・２並列ベンチマーク仕様」を提唱した．このベンチマーク仕様に基づいたベンチマークテストを SP2 で実施した．ベンチマークコードは，粒子分割法で並列化された分子動力学法を用いた．その結果，全対全転送がプロセッサ数と粒子数に比例していることが検出された．また問題の規模が小さくなるに従い flop/s 性能が向上する現象を，負の立ち上がり時間として記述できることを示した．更に，レベル２並列ベンチマークのモデル係数決定方法の手順を示した．

謝辞

日本原子力研究所計算科学技術推進センターの渡辺正氏には，MD コードを使用させていただいたことを感謝いたします．ウッドランド株式会社の滝川好夫氏には，本報告書に使用した測定データとそれに至るまで試行錯誤で数多くの測定をしていただいたことに，深く感謝いたします．

参考文献

[1]折居茂夫，松山雄次，大田敏郎，久米悦雄，相川裕史，熊倉利昌，滝川好夫：並列計算機用ベンチマークテスト・システムの研究開発，計算工学講演会論文集，Vol. 2，No. 1，pp.117-120，1997

[2]折居茂夫：数値計算のための並列計算機性能評価方法，情報処理学会論文誌，Vol.39，No.3，pp.529-541(1998).

[3]Bailey, D., et al. : The NAS Parallel Benchmarks 2.0, NAS-95-020, 1995.

[4]Dongarra, J. J. : Performance of Various Computers Using Standard Linear Equations Software in a Fortran Environment, Computer Architecture News, Vol. 16, pp.47-69, 1988.

[5]松山雄次，折居茂夫，大田敏郎，久米悦夫，相川裕史：プログラム並列化支援解析ツール kpx，JAERI-Tech 97-017，1997.

[6]渡部弘，折居茂夫，熊倉利昌，滝川好夫：プログラム並列化支援解析ツール kpx(その 2)，JAERI-Data/Code 98-016，1998.

[7]渡辺正，蕪木英雄，町田昌彦：第 9 回数値流体力学シンポジウム講演論文集，235(1995).

[8]折居茂夫：分子動力学ｺｰﾄﾞの段階的並列化方法，JAERI-Data/Code 96-023，1996.

[9]青山幸也：RS/6000 SP 並列プログラミング虎の巻(MPI 版)，1996 年 3 月 6 日版.

[10]Stunkel, C. B., et al. : The SP2 High-Performance Switch, IBM Systems Journal Vol.34, No.2, pp.185-204, 1995.

[11]Synergy Software : KaleidaGraph, http://www.synergy.com

付録A モデル係数決定方法

モデル係数は，処理時間モデルと測定値をフィッティングして求める．ここでは各々のサブルーチンに対して実施した解析方法を示す．横軸をプロセッサ数 p 或いは粒子数 n，縦軸を時間に関係した量で記述し，Table 1 のモデル式を変形してフィッティングを行った．以後，測定値は$[\cdots]_M$と表記する．ツールとしては KaleidaGraph[11]を使用した．

フィッティングを行う際，Table 1 のモデル式をそのまま用いると，値の大きな現象に合うように係数が決定され，オーダー的に無視できないが有意な係数が零になる場合がある．これを防ぐため，時にモデル式を変形してフィッティングする必要が生じる．

また，決定した係数が更にプロセッサ数依存或いは粒子数依存をする場合がある．この場合，それが有意であるかを確認し原因を把握する必要がある．原因は，計算機の特性，モデルの精度，大きな現象に隠れた小さな現象等々がある．大きな現象に隠れた小さな現象は，既存のモデル係数に繰り込み可能な場合もある．これらの原因のため，フィッティングによる係数決定は，場合により手段を変える必要がある．

A.1 force

A.1.1 t_{0p270}, e_{p270}, t_{0n270}, a_{u271} 決定

(1) t_{0p270}

・通信時間を引いた force の時間$[\tau_p(p,n)]_M$ - $[(n_{sample}\cdot I_s+1)\cdot\sigma_{270}]_M$ vs. p の曲線作図

ここに，$[\tau_p(p,n)]_M$：force の時間，$[(n_{sample}\cdot I_s+1)\cdot\sigma_{270}]_M$：通信時間（Fig.4 の処理）．

・作図毎（n=800,3200,…，96800）に線形回帰分析を実施し，各々の図に対し $t_{0p270}(n)$ と $12001\cdot t_u(n)/e_{p270}$ を求める．

ここに，$(n_{sample}\cdot I_s+1)\cdot[t_{0p270}(n) + t_u(n)/e_{p270}]=[\tau_p(p,n)]_M-[(n_{sample}\cdot I_s+1)\cdot\sigma_{270}]_M$，$n_{sample}\cdot I_s+1=12001$ を用いた．

・求めた $t_{0p270}(n)$で，$t_{0p270}(n)$ vs.n 曲線を作図し線形回帰分析，

結果：$t_{0p270}(n) =0.00119+2.671\cdot10^{-7}\cdot n$(sec)

(2) e_{p270}

・(1)の回帰分析で求めた $12001\cdot t_u(n)/e_{p270}$ vs.n 曲線作図し線形回帰分析

・$12001\cdot t_u(n)/e_{p270}=-171.82+0.33895\cdot n$ を得る

・プロセッサ数 p=1 における$[\tau_p(1,n)]_M$ vs. n 曲線を作図し線形回帰分析

・$12001\cdot t_u(n) =-133.54+0.34154\cdot n(=T_p-12001\cdot t_{0p270}(n))$ を得る(Fig.11).

ここに t_u は p=1 の処理時間である．$(n_{sample}\cdot Is+1)=12001$ である．

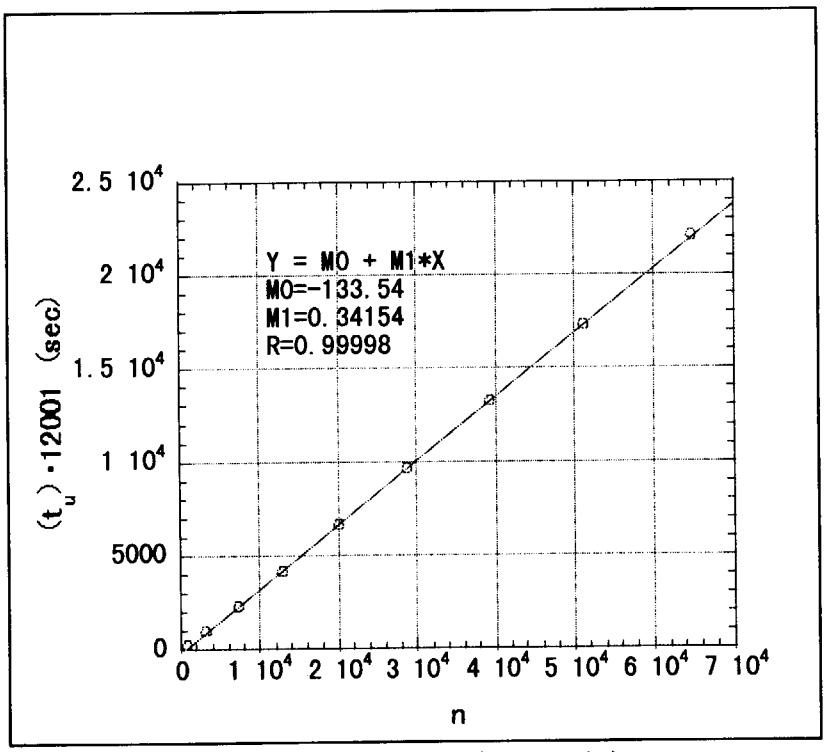

Fig.11 Linear least square fitting of $12001\cdot(t_u)$ vs. n

・これを 12001・$t_u(n)/e_{p270}$ と比較して e_{p270} を次に得る

結果：e_{p270}～1(=0.34154/0.33895)　(n>>1000)

(3) t_{0n270}, a_{u271}

・(1)で求めた 12001・$t_u(n)/e_{p270}$=-171.82+0.33895・n に e_{p270}=1 を代入

・ Table 1 より t_{0u270}+(n-1) ・{ $t_{0u271}+C_1 \cdot n/2 \cdot [(5+13 \cdot C_2)/(r_a \cdot a_{u271})]$ }～ $t_{0u270}+n \cdot 2.02 \cdot 10^{-6}/a_{u271}$

・これと(-171.82+0.33895・n) /12001 を比較し以下を得る

結果：t_{0u270}=-0.0143，a_{u271}=0.071，t_{0u271}～0

Fig.12a に，force の計算部分（σ_{270} の効果を引いた部分）の測定した全てのデータに対する相対誤差を，粒子数に対して示す．相対誤差は$([t_p]_{Measured} - [t_p]_{Estimated}) / [t_p]_{Measured}$ と定義した．ここに$[t_p]_{Measured}$ が実測値$[\tau_p(p,n)]_M$，$[t_p]_{Estimated}$ がモデルからの見積値に対応する．図は粒子数 12800 付近から，モデルが約 20%の精度で処理時間を記述することを示す．

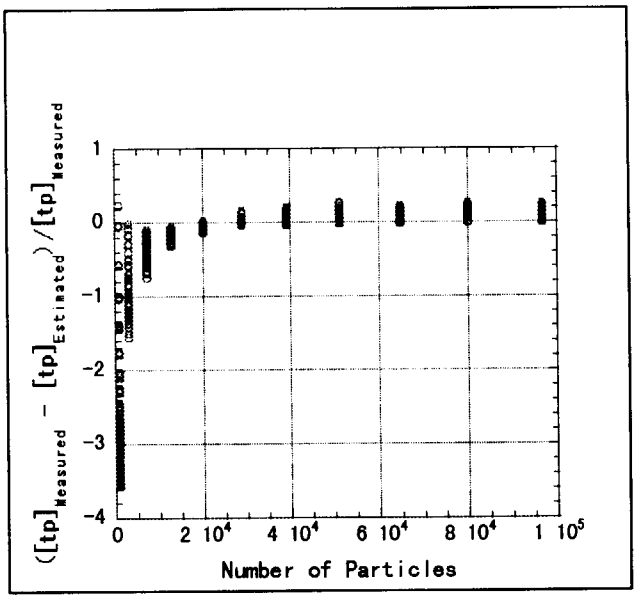

Fig.12a Accuracy of timing model of force with p.

Fig.12b に，force の計算部分（σ_{270} の効果を引いた部分）の測定した全てのデータに対する誤差を，プロセッサ数に対して示す．図は，モデルが粒子数 7200 付近から一定の精度でプロセッサ数に対する処理時間を表すことを示す．尚， n=7200 の上限値を図中に線で示した．

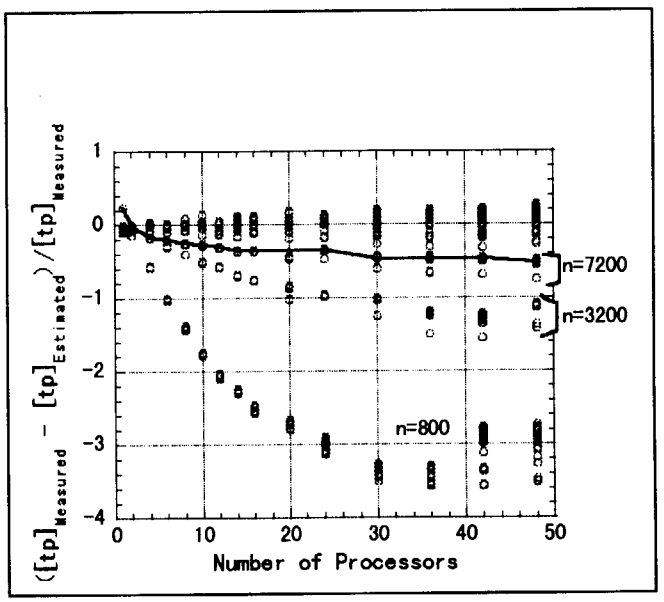

Fig.12b Accuracy of timing model of force with n.

A.1.2 t_{0sum}, t_{0c270}, e_{c270} 決定

(1) t_{0sum}

・$[(n_{sample}\cdot I_s+1)\cdot \sigma_{270}(p=1,n)]_M$ vs. n 曲線作図

・線形回帰分析で t_{0sum} 決定

結果：t_{0sum} =98(nsec)

(2) $[(n_{sample}\cdot I_s+1)\cdot \sigma_{270}(p,n)]_M - (n_{sample}\cdot I_s+1)\cdot t_{0sum}$ vs. x （$=p^{-1}$） 曲線作図

(3)縦軸を 2·2·n·p で除し，評価関数 $t_{0c}{}'_{270}+(x-x^2)/(r_c\cdot e_{c270})$を用いて回帰分析し t_{0c270}，e_{c270} を決定．これを Fig.13 に示す．

結果：$t_{0c270}(=t_{0c}{}'_{270}\cdot n)=8.6\cdot n$ [nsec]，$e_{c270}=0.61$

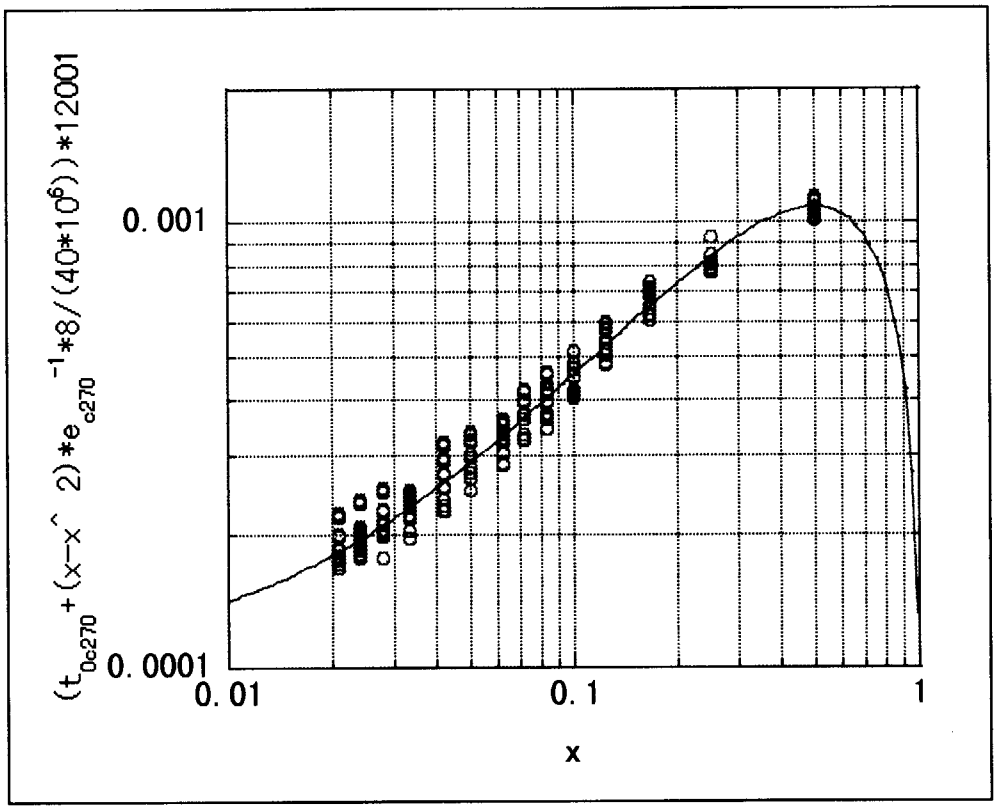

Fig.13 Determination of t_{0c270} and e_{c270} with all measured data excluding case of n=800, n=3200 and p=1.

Fig.14a に，σ_{270} の測定した全てのデータに対する誤差を，粒子数に対して示す．図中の実線は p=1 の下限値を示す．問題の大きさとメモリサイズの関係から，粒子数 80000 と 96800 での p=1 の測定はできなかった．図は粒子数 800，3200 とプロセッサ数 p=1 の場合に誤差が大きく，それ以外では±20%内の精度を持つことを示す．p=1 における誤差が大きいが，σ_{270} の値は p>1 に比べて一桁小さい．また，誤差が粒子数に対して負の傾きを持っているが，現在のところ有意か否かは判然としない．より多くの粒子での実測が必要である．

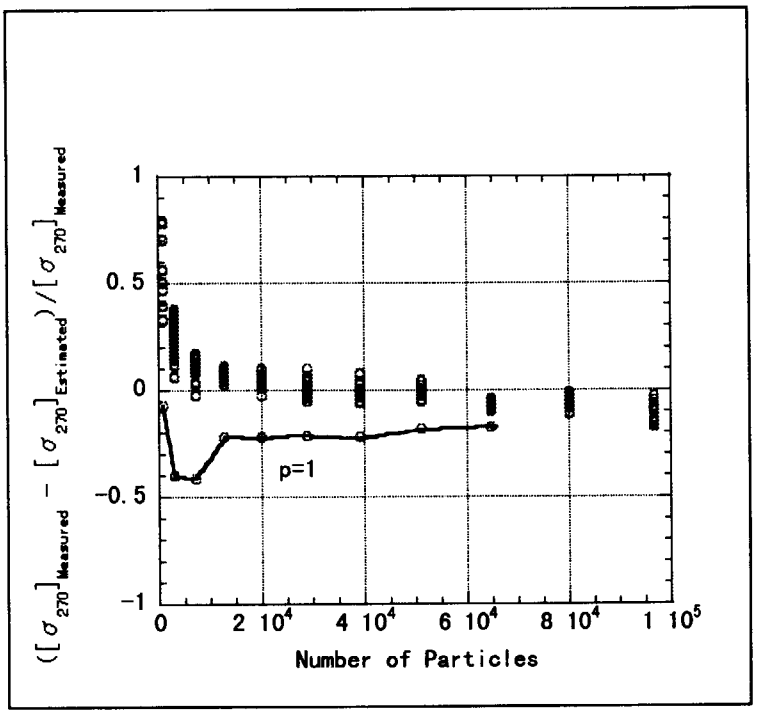

Fig.14a Accuracy of timing model of σ_{270} with p.

Fig.14b に，σ_{270}の測定した全てのデータに対する誤差を，プロセッサ数に対して示す．図中の実線は n=3200 の下限値を示す．図は粒子数 800，3200 とプロセッサ数 p=1 の場合に誤差を取り除くと，±20%内で誤差データの分布がほぼ均等になることを示す．従ってモデルはプロセッサに依存した部分はほぼ現象を捉えていると考えることができる．

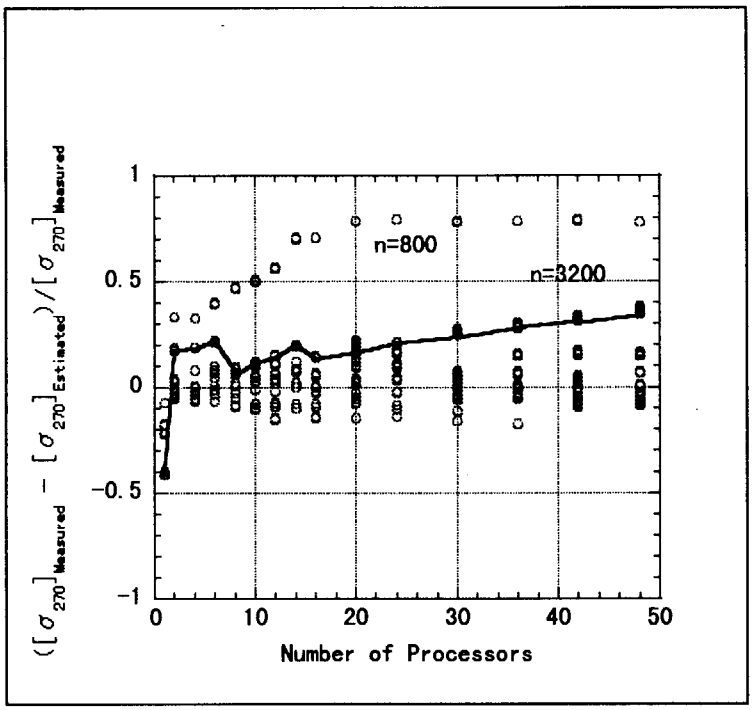

Fig.14b Accuracy of timing model of σ_{270} with n.

A.2 table

A.2.1 t_{0p100}, e_{p100}, t_{0n101}, a_{u101} 決定

(1)t_{0p101}

・$[\tau_p(p,n)]_M$ vs. p の曲線作図

ここに，$[\tau_p(p,n)]_M=((n_{sample}\cdot I_s/n_{table}+1)\cdot t_{p101})=72\cdot t_{p101}$.

・作図毎（n=800,3200,…，96800）に線形回帰分析を実施し，各々の図に対し $t_{0p101}(n)$ を求める.

・求めた $t_{0p100}(n)$で，$t_{0p101}(n)$ vs.n 曲線を作図し線形回帰分析，

結果：$t_{0p101}(n)=0.120\cdot10^{-9}\cdot n^2$ (sec)

(2)e_{p101}

・$([\tau_p(p,n)]_M - 72\cdot t_{0p101}(n))\cdot p$ vs.n 曲線作図し 2 次多項式で回帰分析

・$72\cdot t_u(n)/e_{p101}=33-0.0066\cdot n+2.63\cdot10^{-6}\cdot n^2$ を得る(Fig.15).

ここに t_u は p=1 の処理時間である．Table 1 より $72=(n_{sample}\cdot Is/n_{table}+1)$である.

・プロセッサ数 p=1 における$[\tau_p(1,n)]_M$ vs. n 曲線を作図し線形回帰分析

・$72\cdot t_{0p101}(n)=16 - 0.0054\cdot n + 2.60\cdot10^{-6}\cdot n^2$ を得る．これを Fig.15 の結果と比較する.

結果：$e_{p101}=0.989$

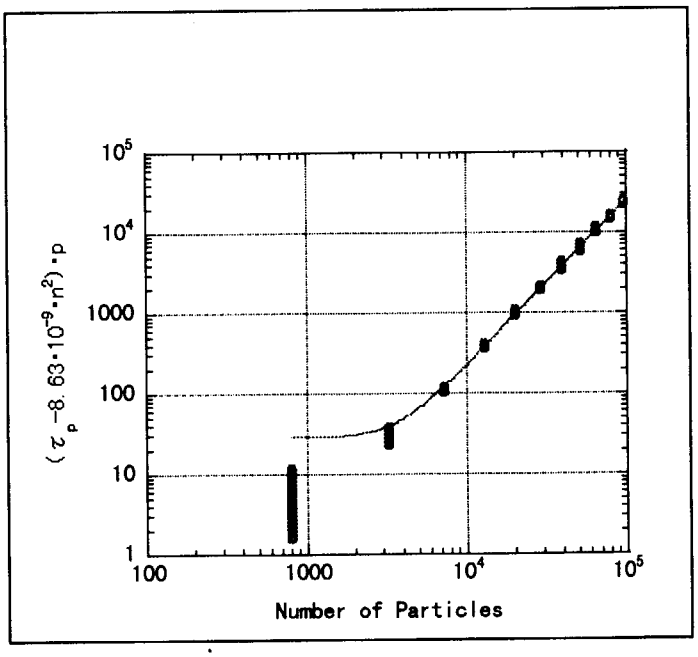

Fig.15 Least square fitting of $[\tau_p(p,n)]_M - 72\cdot t_{0p101}(n)$ vs. n.

(3) t_{0n101}, t_{0n100}, a_{u100}

・ $72 \cdot t_u(n)/e_{p101}=33-0.0066 \cdot n+2.63 \cdot 10^{-6} \cdot n^2$ に e_{p101} を掛けて 72 で割り，Table 1 のモデルと比較することにより，右辺の係数から t_{0u101}, a_{u101} を得ることができる.

結果：$t_{0u101}=0.45$, $t_{0u100}=-97 \cdot 10^{-6}$, $a_{u100}=0.260$

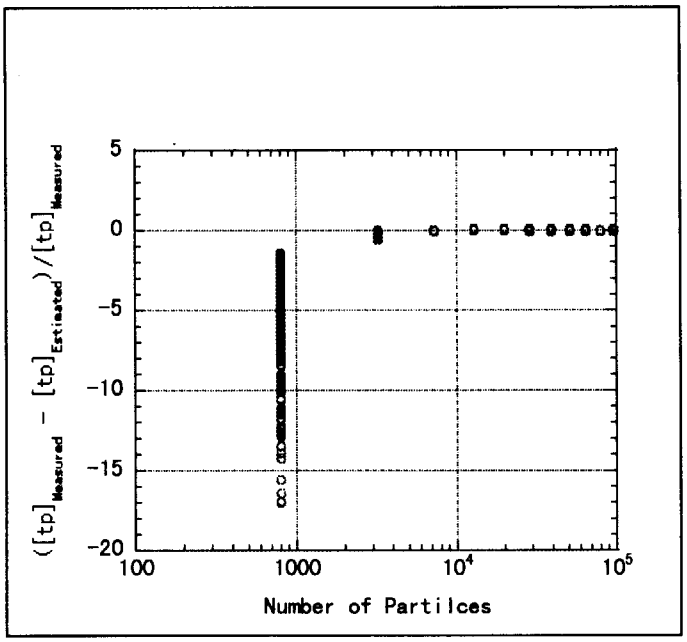

Fig.16a Accuracy of timing model of table with n.

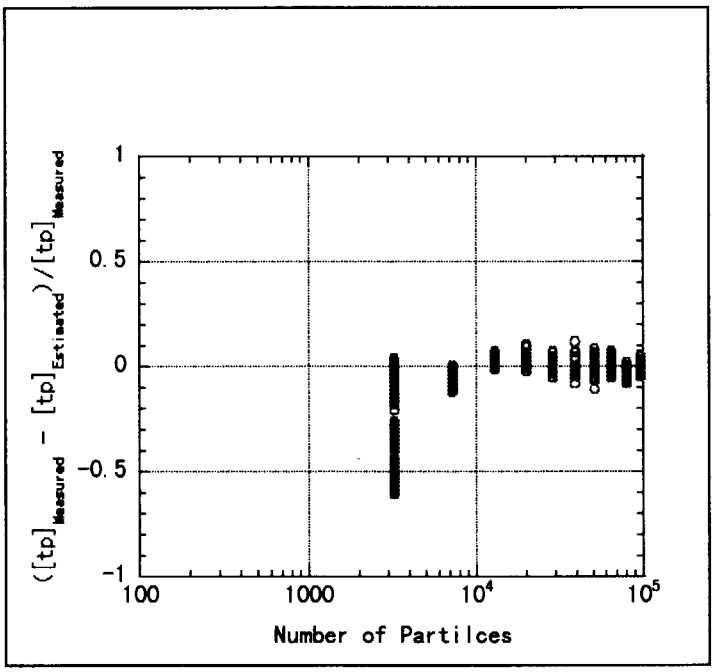

Fig.16b Accuracy of timing model of table with n.

Fig.16a に，table の測定した全てのデータに対する相対誤差を，粒子数に対して示す．図は粒子数 800 の相対誤差が大きいことを示す．Fig.16b に，粒子数 800 を除いた相対誤差を示す．図は粒子数 7200 付近から，モデルが約 20%の精度で処理時間を記述することを示す．

Fig.16c に，table の全てのデータに対する相対誤差を，プロセッサ数に対して示す．図は，モデルが粒子数 800 においてプロセッサのロードが均一でない様子を示している．table は do100 と do101 の２重ループで構成されている．外側のループのループ変数が内側のループの初期値になるが，外側のループ do100 をブロックサイクリック分割して並列化してある．このため粒子数 800 ではロードバランスが悪い場合が生じる．実際，800 を分割して余りが出ないプロセッサ数 10 では，他と比べるとロードバランスがよい．しかし 20 では悪いため，ロードバランスを悪くする複数の要因があると推測する．

尚，　n=3200 の上限値を図中に線で示した．

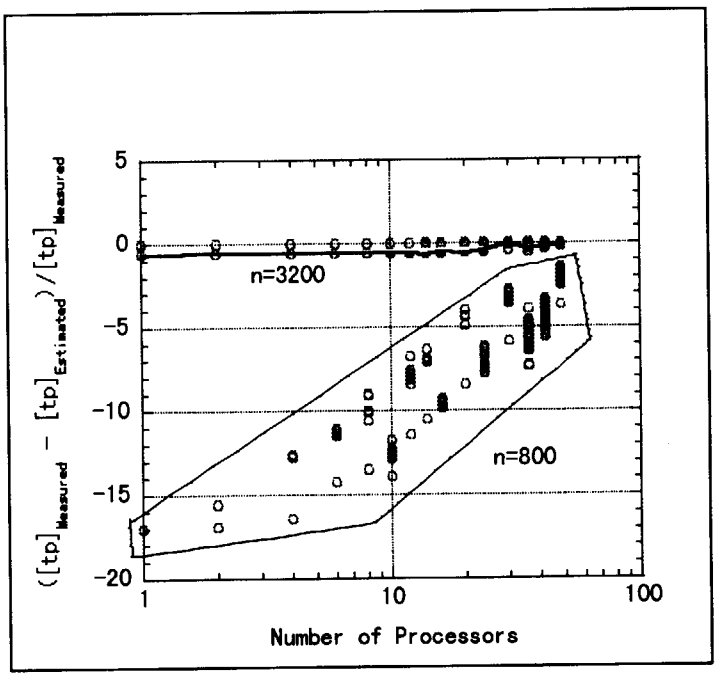

Fig.16c Accuracy of timing model of table with p.

A. 3 propcel

A. 3. 1 t_{0p302}, e_{p302}, t_{0n302}, a_{u302}決定

(1)t_{0p302}

・$[(n_{sample}\cdot I_s)\cdot t_{p302}(p,n)]_M$ vs. p の曲線作図

・作図毎（n=800,3200,…, 96800）に線形回帰分析を実施し，各々の図に対し $t_{0p302}(n)$ を求める.

・求めた $t_{0p302}(n)$で，$t_{0p101}(n)$ vs.n 曲線を作図し線形回帰分析，

結果：$t_{0p302}(n)$ ~0

(2)e_{p302}

・$([\tau_p(p,n)]_M - 12000\cdot t_{0p302}(n))\cdot p$ vs.n 曲線作図し線形回帰分析

・ $12000\cdot t_u(n)/e_{p302}=0.690+0.00287\cdot n$ を得る(Fig.17).

・プロセッサ数 p=1 における$[\tau_p(1,n)]_M$ vs. n 曲線を作図し線形回帰分析

・ $12000\cdot t_u(n)/e_{p302}=0.132+0.00280\cdot n$ を得るこれを Fig.17 の結果と比較する.

結果：$e_{p302}=0.976$

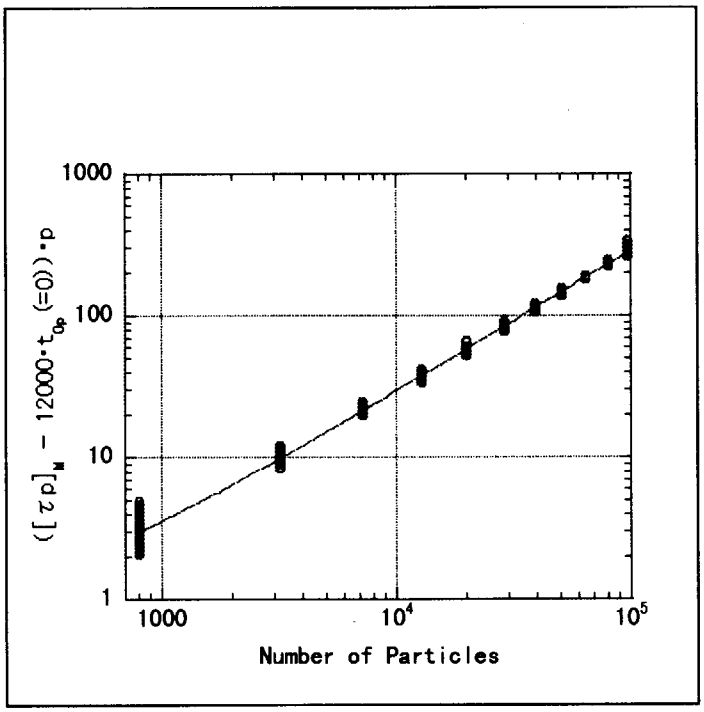

Fig.17 Least square fitting of $[\tau_p(p,n)]_M - 12000\cdot t_{0p302}$ vs. n.

(3) t_{0n302}, a_{u302}

・ $12000\cdot t_u(n)/e_{p302}=0.690+0.00287\cdot n$ に e_{p302} を掛け，12000 で割ると t_{0n302}，$10/a_{u302}$ を得る.

結果：$t_{0n302}=56\,\mu$ sec, $a_{u302}=0.161$

Fig.18a に，do302 の測定した全てのデータに対する相対誤差を，粒子数に対して示す．
Fig.18b に，測定した全てのデータに対する相対誤差を，プロセッサ数に対して示す．

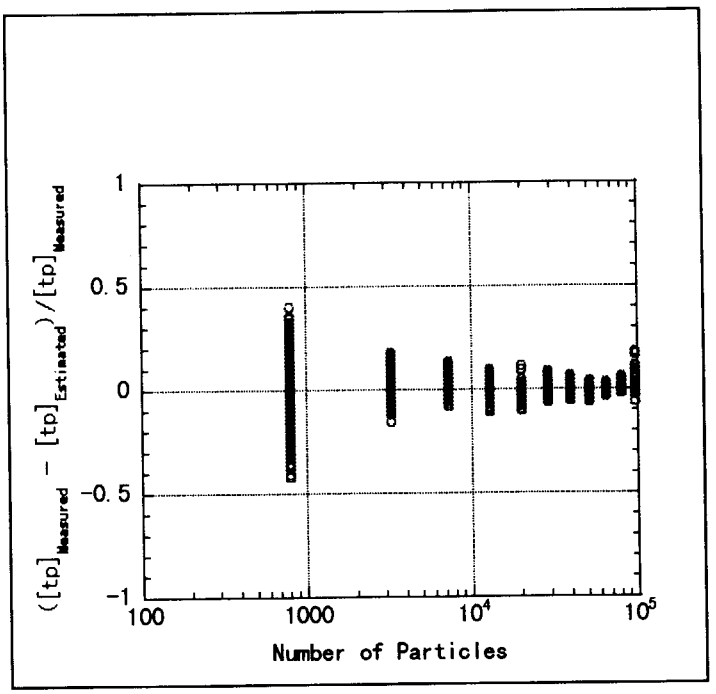

Fig.18a Accuracy of timing model of do302 of propcel with n.

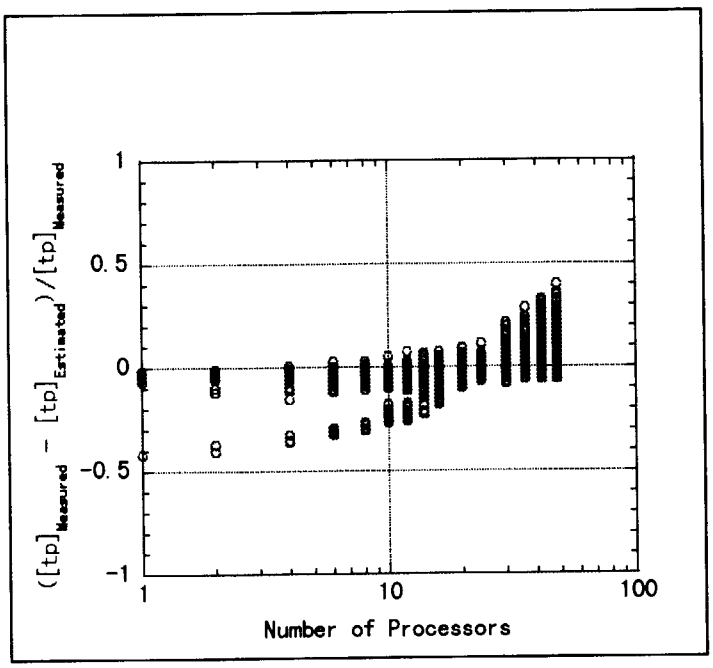

Fig.18b Accuracy of timing model of do302 of propcel with p.

A. 3. 2 t_{0p303}, e_{p303}, t_{0n303}, a_{u303} 決定

(1)t_{0p303}

・$[(n_{sample}\cdot I_s)\cdot t_{p303}(p,n)]_M$ vs. p の曲線作図

・作図毎（n=800,3200,…, 96800）に線形回帰分析を実施し，各々の図に対し $t_{0p303}(n)$ を求める.

・求めた $t_{0p303}(n)$で，$t_{0p101}(n)$ vs.n 曲線を作図し線形回帰分析,

結果：$t_{0p303}(n)$ ~0

(2)e_{p303}

・$([\tau_p(p,n)]_M - 12000\cdot t_{0p303}(n))\cdot p$ vs.n 曲線作図し線形回帰分析

・ $12000\cdot t_u(n)/e_{p303}=-0.383+0.00259\cdot n$ を得る(Fig.19).

ここに t_u は p=1 の処理時間である.

・プロセッサ数 p=1 における$[\tau_p(1,n)]_M$ vs. n 曲線を作図し線形回帰分析

・ $12000\cdot t_u(n)/e_{p302}=-0.0553+0.00261\cdot n$ を得るこれを Fig.19 の結果と比較する.

結果：$e_{p303}=1.0$

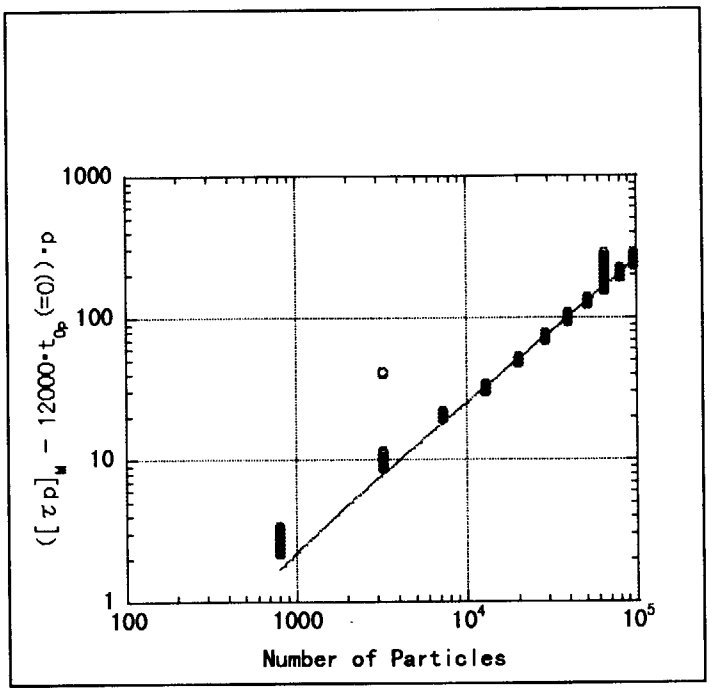

Fig.19 Least square fitting of $[\tau_p(p,n)]_M - 12000\cdot t_{0p303}$ vs. n.(3) t_{0n303}, a_{u303}

・ $12000\cdot t_u(n)/e_{p303}=-0.383+0.00259\cdot n$ に e_{p303} を掛け 12000 で割り，Table 1 のモデルと比較することにより，右辺の係数から t_{0u303}，a_{u303} を得ることができる.

結果：$t_{0n303}=-32\,\mu$ sec, $a_{u303}=0.122$

Fig.20a に，do303 の測定した全てのデータに対する相対誤差を，粒子数に対して示す．
Fig.20b に，測定した全てのデータに対する相対誤差を，プロセッサ数に対して示す．

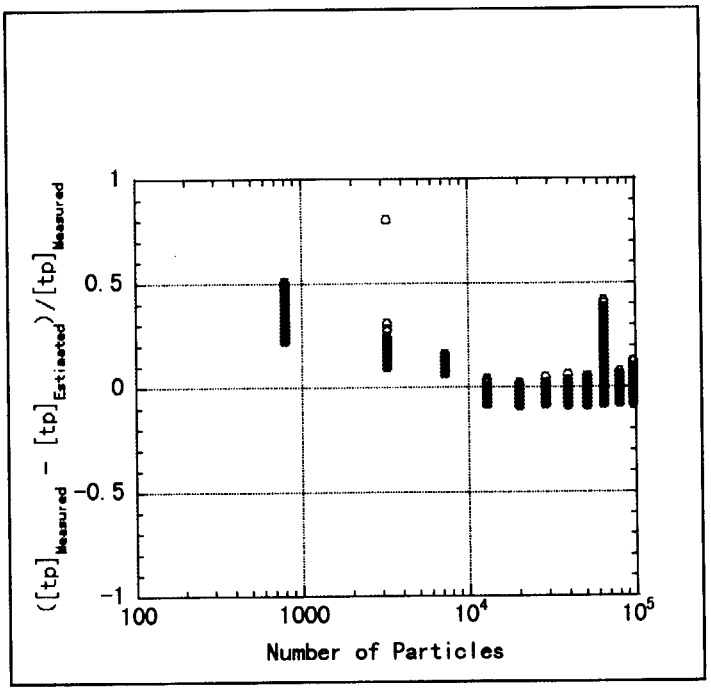

Fig.20a Accuracy of timing model of do303 of propcel with n.

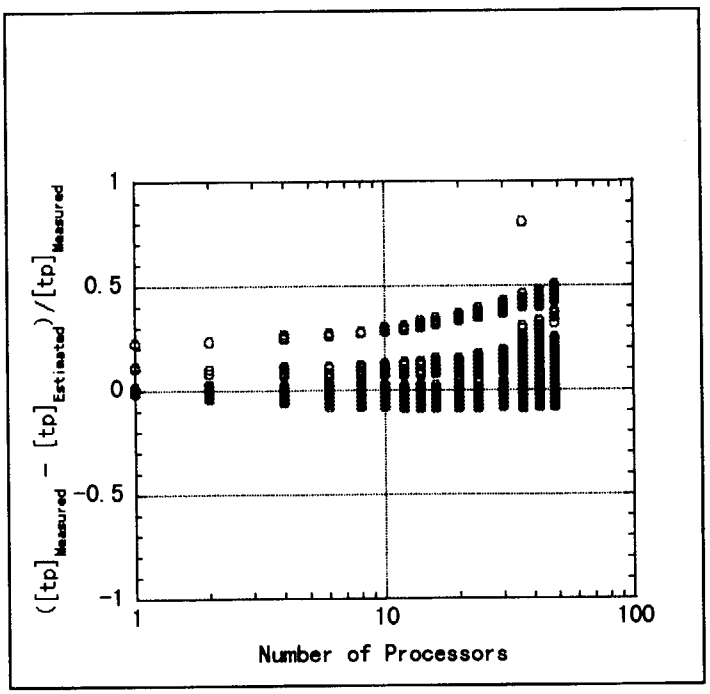

Fig.20b Accuracy of timing model of do303 of propcel with p.

A.3.3 t_{0sum}, t_{0c303}, e_{c303}決定

(1) t_{0sum}

・$[(n_{sample}\cdot I_s+1)\cdot \sigma_{303}(p=1,n)]_M$ vs. n 曲線作図

・線形回帰分析で t_{0sum} 決定

結果：t_{0sum} ~0

(2)$[(n_{sample}\cdot I_s)\cdot \sigma_{303}(p,n)]_M$ vs. x （$=p^{-1}$） 曲線作図

(3)縦軸を $2\cdot 2\cdot n_x\cdot n_y\cdot p$ で除し，評価関数 $t_{0c}{}'_{303}+(x-x^2)/(r_c\cdot e_{c270})$を用いて回帰分析する．これを Fig.21 に示す．ここに，n_x=40，n_y=20 である．

結果：$t_{0c303}(=t_{0c}{}'_{303}\cdot n_x\cdot n_y)=61\cdot n_x\cdot n_y$ [nsec]，e_{c303}=0.34

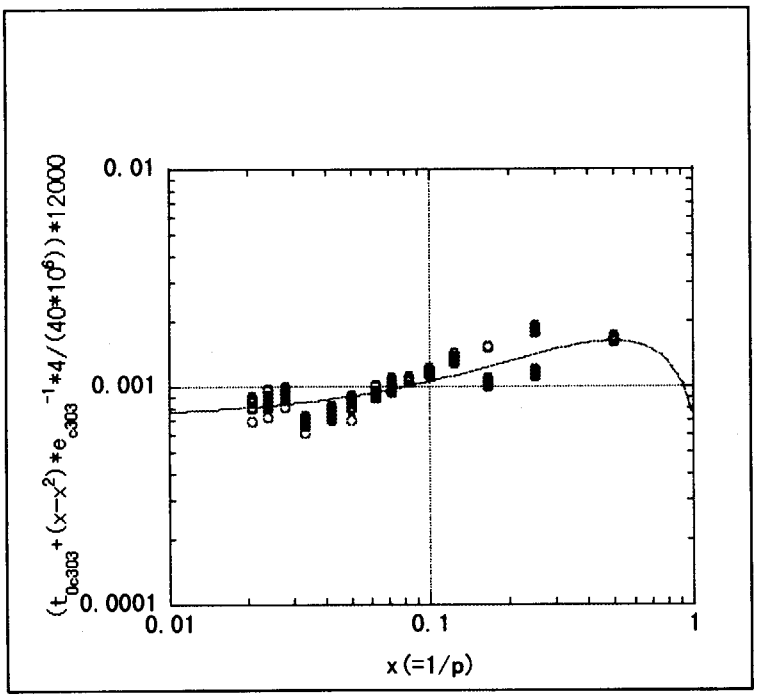

Fig.21 Determination of t_{0c303} and e_{c303} with all measured data excluding case of p=1.

Fig.22 に，σ_{303} の測定した全てのデータに対する相対誤差を，プロセッサ数に対して示す．この BMT での問題の規模 $n_x \cdot n_y$ は一定である．

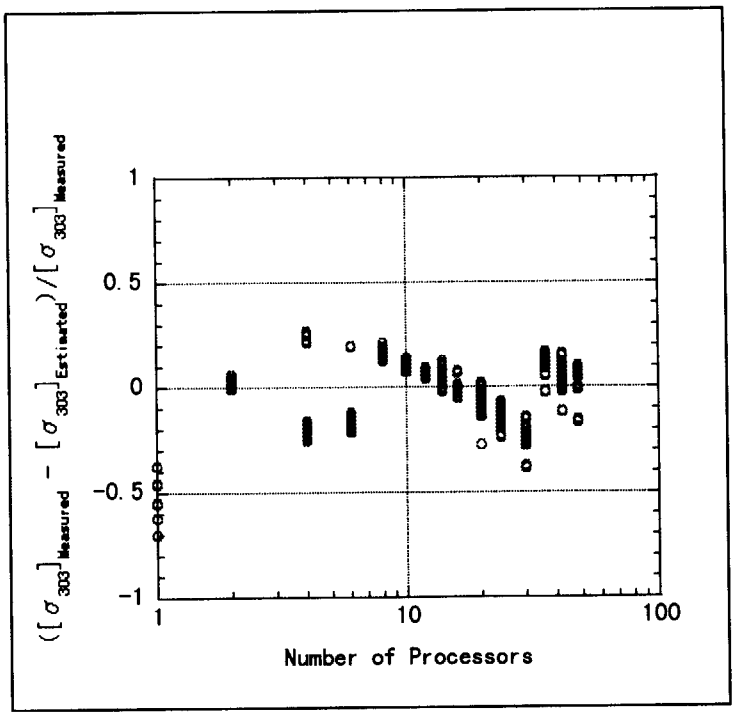

Fig.22 Accuracy of timing model of σ_{303} with p.

A.4 pcross の t_{0u10}, a_{u10} 決定

- $[(n_{sample} \cdot I_s) \cdot t_{u10}(n)]_M$ vs. n の曲線作図し線形回帰分析
- $12000 \cdot t_u(n) = -0.383 + 0.00259 \cdot n$ を 12000 で割り，Table 1 のモデルと比較することにより，右辺の係数から t_{0u10}, a_{u10} を得ることができる.

結果：t_{0u10}= -0.00025sec, a_{u10}=0.0450

Fig.23 に，do10 の測定した全てのデータに対する相対誤差を，粒子数に対して示す.

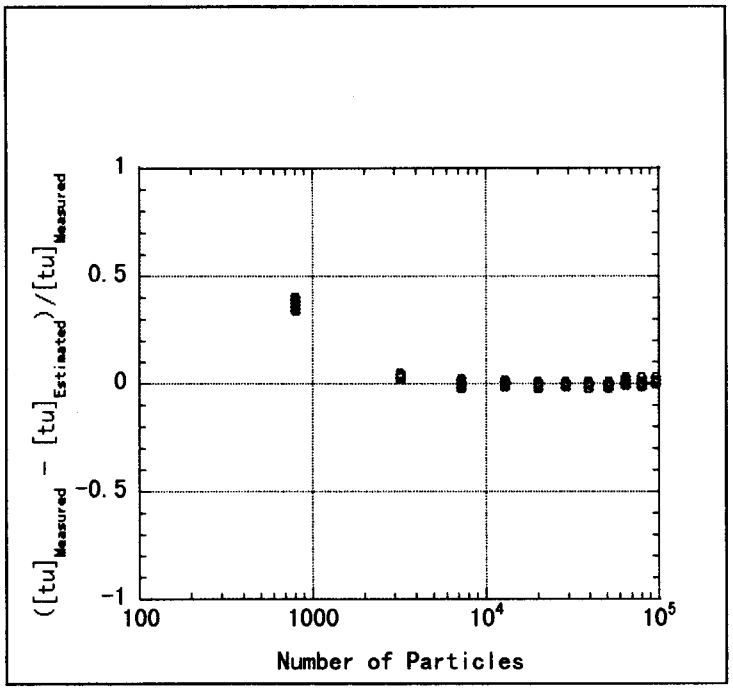

Fig.23 Accuracy of timing of pcross with n.

A. 5 pmoves の t_{0u300}, a_{u300} 決定

- $[(n_{sample} \cdot I_s) \cdot t_{u300}(n)]_M$ vs. n の曲線作図し線形回帰分析
- $12000 \cdot t_u(n) = -7.908 + 0.00346 \cdot n$ を 12000 で割り，Table 1 のモデルと比較することにより，右辺の係数から t_{0u300}, a_{u300} を得ることができる.

結果：t_{0n300}= -0.00065sec, a_{u300}=0.117

Fig.24 に，do300 の測定した全てのデータに対する相対誤差を，粒子数に対して示す.

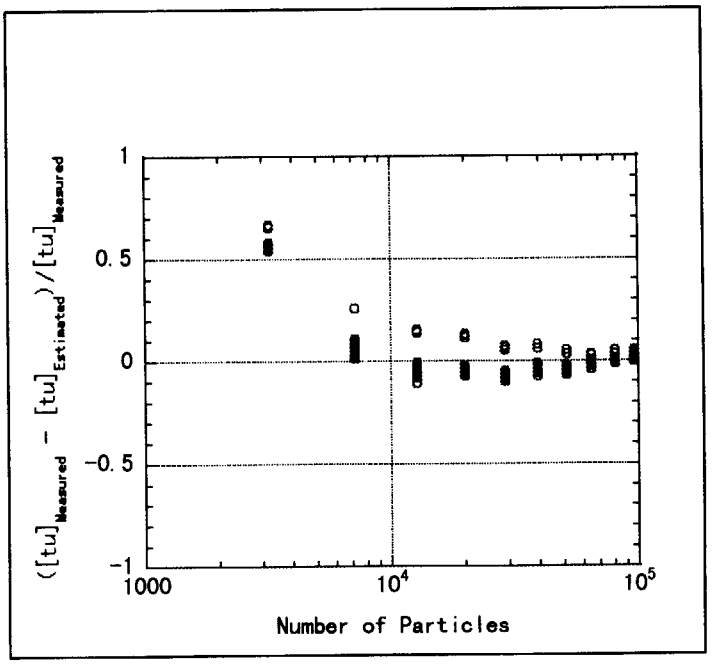

Fig.24 Accuracy of timing of pmoves with n.

This is a blank page.

国際単位系（SI）と換算表

表1　SI基本単位および補助単位

量	名称	記号
長さ	メートル	m
質量	キログラム	kg
時間	秒	s
電流	アンペア	A
熱力学温度	ケルビン	K
物質量	モル	mol
光度	カンデラ	cd
平面角	ラジアン	rad
立体角	ステラジアン	sr

表2　SIと併用される単位

名称	記号
分，時，日	min, h, d
度，分，秒	°, ′, ″
リットル	l, L
トン	t
電子ボルト	eV
原子質量単位	u

1 eV=1.60218×10^{-19}J

1 u=1.66054×10^{-27}kg

表3　固有の名称をもつSI組立単位

量	名称	記号	他のSI単位による表現
周波数	ヘルツ	Hz	s^{-1}
力	ニュートン	N	m·kg/s^2
圧力，応力	パスカル	Pa	N/m^2
エネルギー,仕事,熱量	ジュール	J	N·m
工率，放射束	ワット	W	J/s
電気量，電荷	クーロン	C	A·s
電位，電圧，起電力	ボルト	V	W/A
静電容量	ファラド	F	C/V
電気抵抗	オーム	Ω	V/A
コンダクタンス	ジーメンス	S	A/V
磁束	ウェーバ	Wb	V·s
磁束密度	テスラ	T	Wb/m^2
インダクタンス	ヘンリー	H	Wb/A
セルシウス温度	セルシウス度	℃	
光束	ルーメン	lm	cd·sr
照度	ルクス	lx	lm/m^2
放射能	ベクレル	Bq	s^{-1}
吸収線量	グレイ	Gy	J/kg
線量等量	シーベルト	Sv	J/kg

表4　SIと共に暫定的に維持される単位

名称	記号
オングストローム	Å
バーン	b
バール	bar
ガル	Gal
キュリー	Ci
レントゲン	R
ラド	rad
レム	rem

1 Å=0.1nm=10^{-10}m

1 b=100fm^2=10^{-28}m^2

1 bar=0.1MPa=10^{5}Pa

1 Gal=1cm/s^2=10^{-2}m/s^2

1 Ci=3.7×10^{10}Bq

1 R=2.58×10^{-4}C/kg

1 rad=1cGy=10^{-2}Gy

1 rem=1cSv=10^{-2}Sv

表5　SI接頭語

倍数	接頭語	記号
10^{18}	エクサ	E
10^{15}	ペタ	P
10^{12}	テラ	T
10^9	ギガ	G
10^6	メガ	M
10^3	キロ	k
10^2	ヘクト	h
10^1	デカ	da
10^{-1}	デシ	d
10^{-2}	センチ	c
10^{-3}	ミリ	m
10^{-6}	マイクロ	μ
10^{-9}	ナノ	n
10^{-12}	ピコ	p
10^{-15}	フェムト	f
10^{-18}	アト	a

（注）

1．表1－5は「国際単位系」第5版，国際度量衡局1985年刊行による。ただし，1 eVおよび1 uの値はCODATAの1986年推奨値によった。

2．表4には海里，ノット，アール，ヘクタールも含まれているが日常の単位なのでここでは省略した。

3．barは，JISでは流体の圧力を表わす場合に限り表2のカテゴリーに分類されている。

4．EC閣僚理事会指令ではbar，barnおよび「血圧の単位」mmHgを表2のカテゴリーに入れている。

換算表

力	N(=10^{5}dyn)	kgf	lbf
	1	0.101972	0.224809
	9.80665	1	2.20462
	4.44822	0.453592	1

粘度　1 Pa·s(N·s/m^2)=10 P(ポアズ)(g/(cm·s))

動粘度　1m^2/s=10^{4}St(ストークス)(cm^2/s)

圧力	MPa(=10bar)	kgf/cm^2	atm	mmHg(Torr)	lbf/in^2(psi)
	1	10.1972	9.86923	7.50062×10^3	145.038
	0.0980665	1	0.967841	735.559	14.2233
	0.101325	1.03323	1	760	14.6959
	1.33322×10^{-4}	1.35951×10^{-3}	1.31579×10^{-3}	1	1.93368×10^{-2}
	6.89476×10^{-3}	7.03070×10^{-2}	6.80460×10^{-2}	51.7149	1

エネルギー・仕事・熱量	J(=10^7 erg)	kgf·m	kW·h	cal(計量法)	Btu	ft·lbf	eV
	1	0.101972	2.77778×10^{-7}	0.238889	9.47813×10^{-4}	0.737562	6.24150×10^{18}
	9.80665	1	2.72407×10^{-6}	2.34270	9.29487×10^{-3}	7.23301	6.12082×10^{19}
	3.6×10^6	3.67098×10^5	1	8.59999×10^5	3412.13	2.65522×10^6	2.24694×10^{25}
	4.18605	0.426858	1.16279×10^{-6}	1	3.96759×10^{-3}	3.08747	2.61272×10^{19}
	1055.06	107.586	2.93072×10^{-4}	252.042	1	778.172	6.58515×10^{21}
	1.35582	0.138255	3.76616×10^{-7}	0.323890	1.28506×10^{-3}	1	8.46233×10^{18}
	1.60218×10^{-19}	1.63377×10^{-20}	4.45050×10^{-26}	3.82743×10^{-20}	1.51857×10^{-22}	1.18171×10^{-19}	1

1 cal= 4.18605J（計量法）

= 4.184J（熱化学）

= 4.1855J（15℃）

= 4.1868J（国際蒸気表）

仕事率　1 PS(仏馬力)

= 75 kgf·m/s

= 735.499W

放射能	Bq	Ci
	1	2.70270×10^{-11}
	3.7×10^{10}	1

吸収線量	Gy	rad
	1	100
	0.01	1

照射線量	C/kg	R
	1	3876
	2.58×10^{-4}	1

線量当量	Sv	rem
	1	100
	0.01	1

（86年12月26日現在）

レベル1・2並列ベンチマーク仕様及びそれに基づくスカラ並列計算機SP2のベンチマークテスト