

JAERI-Data/Code

2000-022



JP0050368



MCNP4B の MPI 化および MCNP4B
を基にした並列 BMT コードの開発

2000 年 3 月

小林 穰

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、お申し越し下さい。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行っております。

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.

MCNP4B の MPI 化および MCNP4B を基にした並列 BMT コードの開発

日本原子力研究所計算科学技術推進センター

小林 穰

(2000 年 2 月 9 日受理)

日本原子力研究所計算科学技術推進センター (CCSE) では、これまで原子力分野の典型的なコードから構成されている並列ベンチマーク・テスト (BMT) コードを開発・整備してきている。これらのコード群は、計算機の性能を評価するのに有効であり、また利用者が性能評価結果を参照しながら類似コードを並列化することも可能である。CCSE で開発・登録されている BMT コードは数値計算法により分類されおり、粒子法、PIC 法、有限差分法および有限要素法コードである。今回、モンテカルロ法のコードが新たに並列 BMT コードに追加された。

モンテカルロ法の並列 BMT コードの基となるコードは原子力分野の粒子輸送問題で広く使用されている MCNP である。最新版である MCNP4B の PVM 版を基にして、MPI 版を作成し、原研内の 4 種類のスカラ並列計算機で性能測定を実施した。さらに、MPI 版 MCNP4B を基に、モンテカルロ法を用いた並列 BMT コードの開発を行なった。本コードは、解析対象よりに臨界計算と固定源問題の 2 種類用意している。

Development of MPI version MCNP4B and Parallel BMT code based on MCNP4B

Minoru KOBAYASHI

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

(Received February 9, 2000)

The Center for Promotion of Computational Science and Engineering(CCSE) at Japan Atomic Energy Research Institute has developed and maintained a set of parallel benchmark test(BMT) codes, which consists of typical codes in nuclear field. They are available for an estimation of computer performance or worked as guidance codes when users parallelize their own codes. BMT codes are classified by numerical method as particle-particle, particle-in-cell, finite difference and finite element codes. Monte Carlo code is newly added to the set of parallel BMT codes. That is MCNP which is widely used as a particle transportation solver in nuclear field. Based on the latest PVM version MCNP4B, the MPI version was developed and its performance was measured with use of 4 scalar parallel computers at JAERI. The code has two versions. The one is the criticality calculation and the other is the fixed source problem.

Keywords : MCNP4B, Benchmark Test, Parallel Processing, MPI

目 次

1. はじめに	1
2. モンテカルロ・コードの概要	2
2.1 モンテカルロ法の概要	2
2.2 MCNP4B の特徴	2
2.3 MCNP4B の計算フロー	2
3. 逐次版の動作確認および性能測定	4
3.1 逐次版 MCNP4B の各スカラー並列計算機への移植	4
3.2 実行結果の確認方法	7
3.3 実行結果の確認	9
3.4 性能比較	10
4. MCNP4B の MPI 化	13
4.1 PVM 版 MCNP4B の構造	13
4.2 負荷分散方法について	14
4.3 MCNP4B の MPI 化	15
4.4 MPI 化の基本方針	15
4.5 MPI 化の方法	15
5. MPI 版の動作確認および性能測定	20
5.1 各スカラー計算機への移植	20
5.2 実行結果の確認	20
5.3 実行環境	20
5.4 実行結果の確認方法	20
5.5 並列版 MCNP4B の実行時間の計測	21
6. 並列 BMT コード開発のコンセプト	25
6.1 コンセプト	25
6.2 開発の手順	25
7. MCNP4BK-BMT (臨界計算)	26
7.1 入力データの概要	26
7.2 並列 BMT コード開発	26
7.3 ベンチマークテストの結果	27
8. MCNP4BF-BMT (固定源問題)	34
8.1 入力データ概要	34
8.2 固定源問題の問題点	34
9. 負荷分散手法の改良	36
9.1 MCNP4B の負荷分散手法	36

9.2	武宮の動的負荷分散法	36
9.3	MCNP4B コードにおける武宮の手法の効果	36
9.4	MCNP4BF-BMT コードにおける武宮の手法の効果	37
9.5	ベンチマークの結果	42
10.	性能評価モデルへの構築	47
10.1	ベンチマークテスト・システム	47
10.2	レベル2ベンチマーク仕様の問題点	47
10.3	性能評価のための基礎データ	48
10.4	性能評価モデルの開発	51
10.5	SP2 の各処理時間のモデル係数の決定	52
10.6	SR2201 の各処理時間のモデル係数の決定	57
11.	固定源問題の性能評価モデル	63
11.1	SP2 の各処理時間のモデル係数の決定	63
11.2	SR2201 の各処理時間のモデル係数の決定	67
12.	留意事項	71
12.1	SR2201 が遅い原因	71
12.2	I/O 処理が遅い原因	74
13.	まとめ	75
	謝辞	75
	参考文献	76
	付録	77
1.1	IBM SP2 の測定結果	77
1.2	日立 SR2201 の測定結果	77
1.3	Intel Paragon の測定結果	77
1.4	富士通 AP3000 の測定結果	77
2.1	IBM SP2 の測定結果	82
2.2	日立 SR2201 の測定結果	82
2.3	Intel Paragon の測定結果	82
2.4	富士通 AP3000 の測定結果	82

Contents

1. Preface	1
2. Outline of Monte Carlo Code	2
2.1 Outline of Monte Carlo Method	2
2.2 Feature of MCNP4B	2
2.3 Calculation Flow of MCNP4B	2
3. Validation and Performance Measurement of Serial Version	4
3.1 Transportaion of Serial Version MCNP4B onto Scalar Parallel Computers	4
3.2 Method for Validation of Computational Results	7
3.3 Validation of Computational Results	9
3.4 Performance Comparision	10
4. Method for Constructing MPI Version MCNP4B	13
4.1 Configuration of PVM Version MCNP4B	13
4.2 Load Balancing Method	14
4.3 Constructing MPI Version MCNP4B	15
4.4 Basic Concept for Constructing MPI Version MCNPT4B	15
4.5 Method for Constructing MPI Version MCNP4B	15
5. Validation and Performance Measurement of MPI Version	20
5.1 Transpotation onto Scalar Parallel Computers	20
5.2 Validation of Computational Results	20
5.3 Execution Environment	20
5.4 Method for Validation of Computational Results	20
5.5 Measurement of Elapsed Time of Parallel MCNP4B	21
6. Concept of Parallel BMT Code	25
6.1 Concept	25
6.2 Development Procedure	25
7. MCNP4BK-BMT (Criticality Calculation)	26
7.1 Outline of Input Data	26
7.2 Development of Parallel BMT Code	26
7.3 Results of Benchmark Test	27
8. MCNP4BF-BMT (Fixed Source Problem)	34
8.1 Outline of Input Data	34
8.2 Problem of Fixed Source Problem	34
9. Improvement of Load Balancing Method	36
9.1 Load Balancing Method of MCNP4B	36

9.2	Takemiya's Dynamic Load Balancing Method	36
9.3	Effect of Takemiya's Method Related with MCNP4B Code	36
9.4	Effect of Takemiya's Method Related with MCNP4BF-BMT Code	37
9.5	Results of Benchmark Test	42
10.	Construction of Performance Evaluation Model	47
10.1	Benchmark Test System	47
10.2	Problem of Level 2 Benchmark Specification	47
10.3	Basic Data of Performance Evaluation	48
10.4	Development of Performance Evaluation Model	51
10.5	Model Parameter Determination for Timing Study on SP2	52
10.6	Model Parameter Determination for Timing Study on SR2201	57
11.	Performance Evaluation Model for Fixed Source Problem	63
11.1	Model Parameter Determination for Timing Study on SP2	63
11.2	Model Parameter Determination for Timing Study on SR2201	67
12.	Notification	71
12.1	Reason for Delay on SR2201	71
12.2	Reason for Delay of I/O Processing	74
13.	Conclusion	75
	Acknowledgements	75
	References	76
	Appendix	77
1.1	Results on IBM SP2	77
1.2	Results on Hitachi SR2201	77
1.3	Results on Intel Paragon	77
1.4	Results on Fujitsu AP3000	77
2.1	Results on IBM SP2	82
2.2	Results on Hitachi SR2201	82
2.3	Results on Intel Paragon	82
2.4	Results on Fujitsu AP3000	82

1. はじめに

日本原子力研究所計算科学技術推進センター (CCSE) では、これまで並列ベンチマークテストコード (並列 BMT コード) を開発・整備してきた (折居 [1])。これまでに開発が完了した並列 BMT コードは、分子動力学コード (MD-BMT)、PIC 法を用いたプラズマ解析コード (GYRO3D-BMT)、有限差分法による 3次元熱流体解析コード (BiCGSTAB2-BMT) および有限要素法による板成形シミュレーション・コード (ITAS-BMT) である。CCSE では、原研に設置されたスーパーコンピュータ (Fujitsu VPP300, Hitachi SR2201, NEC SX-4, IBM SP2, SGI/CRAY T94 および Intel Paragon) 上でこれら BMT コード群の並列化及び性能評価を実施してきた。利用者は、これらのコードを直接利用することも、性能評価結果を参照しながら類似コードを並列化することも可能である。今回は、これまで対応していなかったモンテカルロ法 (MC 法) の並列 BMT コードを開発した。

MC 法の並列 BMT コードの基礎となるコードは、原子力分野の粒子輸送問題で広く使用されている MCNP [6] である。

粒子輸送問題は、固定源問題と固有値問題 (臨界計算) に分類することができる。前者は、多くの粒子 (数百万) を独立に追跡することができるため、負荷分散が比較的容易である。これに対し後者は、少数 (数千個) の粒子を追跡する世代ごとの計算を同期を取りながら逐次的に行なうため、スケジューリングに使用できる粒子数が少なく、負荷分散が困難となっている (樋口ら [4])。

そこで、MC 法の並列 BMT コードは、固定源問題と固有値問題の 2 種類を開発した。固定源問題の並列 BMT コードは、負荷分散手法として、オリジナルに追加して武宮による動的負荷分散手法も用意している。

本報告書では、MCNP パッケージに含まれている PVM 版から MPI 版への移植、並列 BMT コードの開発、各種スカラー並列計算機上での性能評価モデルの構築および性能測定結果について述べる。

2. モンテカルロ・コードの概要

2.1 モンテカルロ法の概要

粒子輸送問題を解くモンテカルロ・シミュレーションにおいては、乱数と断面積データにより個々の粒子の振る舞いがシミュレーションされる。まず、粒子に初期値として位置、エネルギー等の粒子属性を与えた後、追跡を開始する。粒子の飛行パスについては、その方向と平均自由行程単位の距離を乱数によって決める。この飛行パスを追跡し、パスの終点において、断面積データと乱数を用いて非弾性散乱、核分裂といった衝突のシミュレーションを行なう。この間、粒子束、エネルギー蓄積といった物理量について粒子がそれぞれの領域に与える寄与分を計算する。このようなシミュレーションを多数の粒子に対して行ない、系全体の振る舞いを統計的に評価する。

2.2 MCNP4B の特徴

MCNP (A General Monte Carlo N-Particle Transport Code) [6] は、中性子、光子及び電子輸送問題を解くための汎用モンテカルロ・コードであり、米国ロスアラモス国立研究所で開発されている。二次曲面の論理演算によって表現された任意の3次元領域を取り扱うことができる。長期にわたる改良の結果、連続エネルギーの取り扱いや種々の評価法等多くの特長を有しており、このため、原子力分野で広く用いられている。

MCNP コードの統計情報 (Fortran 言語) は、Table 2.1 のとおりである。

Table 2.1 MCNP の統計情報

バージョン	ルーチン数	ライン数	ステップ数	コメント行数
逐次版	348	113,625	92,073	21,552
PVM 版	370	115,309	93,371	21,938

2.3 MCNP4B の計算フロー

MCNP4B は、扱う粒子の種類や統計値として求める物理量によって計算フローは若干異なるが、基本的には、粒子を1個ずつ取り出し、以下の処理を行なう (樋口ら [3])。

- (1) 乱数と分布関数等線源に対する条件を用いて、粒子の初期値 (位置、飛方向、飛程の長さ、エネルギー等) を決定する。
- (2) 粒子の現在位置から飛方向に沿った領域の境界までの距離を計算する。

- (3) 境界までの距離と粒子の飛程の長さを比較し、粒子が領域の境界を越えたか否かを決定する。
- (4) 粒子束や Energy deposition 等興味ある物理量について、粒子のその領域への寄与分を計算する。
- (5) 粒子が領域を越えた場合、粒子が次に進入する領域を探し、ステップ(2)へ戻る。ただし、体系外に出た場合、漏曳したものと判断し、追跡を終了する。その領域にとどまった場合、次のステップにおいて衝突のシミュレーションを行なう。
- (6) 求めた衝突点において、乱数と断面積データを用いて、衝突のシミュレーションを行なう。この結果、粒子が散乱したか、二次粒子を生成した場合、さらに追跡を続けるためにステップ(2)へ戻る。粒子が吸収された場合、追跡を終了する。

3. 逐次版の動作確認および性能測定

3.1 逐次版 MCNP4B の各スカラー並列計算機への移植

MCNP4B の逐次版を 4 種類のスカラー並列計算機 (SR2201, SP2, AP3000 および Paragn) に移植した。相異点は、Table 3.1 のとおりである。

Table 3.1 逐次版の相異点について

ルーチン名	相異点
mcnp	getenv ルーチンの引数の個数が SR2201 のみ異なる。
getexm	変数の引用を開始する位置が SR2201 のみ異なる。
getidt	日付および時刻を得るルーチンが各計算機で異なる。
ttyint	signal ルーチンの引数の個数が Paragon のみ異なる。
mcnpc	Fortran と C との間の手続き呼び出し方法が AP3000 と Paragon で異なる。

具体的な変更点は、Fig.3.1 と 3.2 のとおりである。

```

! --- mcnp.F ----
c
c      get DATAPATH from environmental variable if possible.
cccc   DATAPATH must be upper case.
        hdpth=hdpath
        hdpth=' '
#ifdef SR2201
        call getenv('DATAPATH',8,hdpth,80)
#else
        call getenv('DATAPATH',hdpath)
#endif
        if(hdpth.eq.' ')hdpth=hdpth
c

! ---- getexm.F ----
        subroutine getexm(hm)
c      put the execute line message into hm.
COM$hp9000_800 system on
        character hm*80
c
        hm=' '
        j=0
#ifdef SR2201
        do 10 i=2,iargc()
#else
        do 10 i=1,iargc()
#endif
        call getarg(i,hm(j+1:))
        10 j=j+leng(hm(j+1:))+1
COM$hp9000_800 system off
        return
        end

! ---- ttyint.F ----
        subroutine ttyint
c      set up the tty interrupt.
        external pttyin
c      on controlc call pttyin
#ifdef Paragon
        call signal(2,pttyin,iflag)
#else
        call signal(2,pttyin)
#endif
        return
        end

```

Fig. 3.1 逐次版の修正方法 (その1)

```

! ---- getidt.F ----
      subroutine getidt(hi)
c       put the machine designator, date, and time into hi
c       in the form  x mm/dd/yy hh:mm:ss
      character hi*19
#ifdef SR2201
      character buf*12
#endif
#ifdef SP2
      dimension jd(3),jt(3),it(9)
#endif
#ifdef AP3000
      character buf*24
#endif
#ifdef Paragon
      character buf*24
      character monthe*36
      character monthn(12)*2
      data monthe / 'JanFebMarAprMayJunJulAugSepOctNovDec' /
      data monthn / '01', '02', '03', '04', '05', '06',
1         '07', '08', '09', '10', '11', '12' /
#endif
c
      hi='      / /      : : '
#ifdef SR2201 || defined (AP3000)
      call date(buf)
      hi(3:4)=buf(4:5)
      hi(6:7)=buf(7:8)
      hi(9:10)=buf(1:2)
#endif
#ifdef SR2201
      call clock(buf,1)
      hi(12:13)=buf(1:2)
      hi(15:16)=buf(4:5)
      hi(18:19)=buf(7:8)
#elif AP3000
      call fdate(buf)
      hi(12:13)=buf(12:13)
      hi(15:16)=buf(15:16)
      hi(18:19)=buf(18:19)
#endif
#ifdef Paragon
      call fdate(buf)
      hi(3:4)=buf(23:24)
      month=index(monthe,buf(5:7))/3+1
      hi(6:7)=monthn(month)
      hi(9:10)=buf(9:10)
      hi(12:13)=buf(12:13)
      hi(15:16)=buf(15:16)
      hi(18:19)=buf(18:19)
#endif
#ifdef SP2
      ir=time(ic)
      call getjdt(%val(localtime(ic)),it)
      write(hi,10)it(5)+1,it(4),it(6),it(3),it(2),it(1)
10 format(2x,i2.2,1h/,i2.2,1h/,i2.2,1x,i2.2,1h:,i2.2,1h:,i2.2)
#endif
      return
      end

```

Fig. 3.2 逐次版の修正方法 (その2)

3.2 実行結果の確認方法

逐次版 MCNP4 の動作確認方法は、以下のとおりである。

3.2.1 テスト・データ

動作確認に使用したテスト・データは、以下のとおりである。

- 逐次版の動作確認は、MCNP4B パッケージ中のサンプル・データ (inp01 から inp10 と inp29) および実データ (人体に対する被曝線量計算問題)
- 並列版の並列化効率を見るために、inp04 の粒子数を増加させたデータ

なお、全サンプル・データの中から上記のデータを選択した理由は、inp29 のみが MCNP4B の対応している 3 粒子 (中性子/電子/光子) 全てを含んだ計算である以外は、単純に臨界計算と固定源問題を含んだ 10 種類のデータを最初から順次使用している。

3.2.2 指定のコンパイル・オプション

各種計算機で指定したコンパイル・オプションは、Table 3.2 のとおりである。

Table 3.2 コンパイル・オプション

計算機	指定オプション
SR2201	FFLAGS = -i,E,U,L,BL,CD,CL -W0,'langlvl(darg(1))' \$(FOPTS) FOPTS = -W0,'opt(o(3)),list(e(4)),pvec(diag(0))' COPTS = +O3 -Wc,-T,-hD0
SP2	FFLAGS = -NQ20000 -NA16384 \$(FOPTS) FOPTS = -O3 -qtune=pwr2 -qarch=pwr2 -qstrict COPTS = -O3 -qstrict
AP3000	FFLAGS = -I/usr/openwin/include \$(FOPTS) FOPTS = -O3 -KULTRA,V8PLUS COPTS = -O3
Paragon	FFLAGS = -nx \$(FOPTS) FOPTS = -O4 -Knoieeee -Mvect COPTS = -O4 LDFFLAGS = -L/export/home/PARAGON/paragaon/lib-coll LDLIBS = -lmpi -lnx -L\$(HOME)/local/lib -lX11 -lm

個々のオプションの留意点は、以下のとおりである。

SR2201 で指定したコンパイル・オプションは、パッケージの Makefile に記載されているものを使用した。なお、-W0,'langlvl(darg(1))' のオプションを指定しないとジョブが

異常終了 (bus error が発生) することがあるので, 注意を要する (Version 4A から 4B に変更になった時点で必要となったオプションとのこと)。

SP2 で指定したコンパイル・オプションは, パッケージの Shell Script に記載されているもの (表中の FFLAGS に記載されているオプション) に最適化のオプション (表中の FOPTS に記載されているオプション) を追加した。

AP3000 で使用したコンパイラは, 富士通 Fortran90/C 言語である。指定したコンパイル・オプションは, Web 上で記載されている最適化のオプション (表中の FOPTS に記載されているオプション) を指定し, X 関連のライブラリが存在するディレクトリ (表中の FFLAGS) も指定している。

Paragon で指定したコンパイル・オプションには, 高速化のための `-Knoieee` オプション (除算の高速化) を含んでいる。指定したオプションは, フロント・エンド (SUN4/1000E) でクロス環境を用いた場合の指定方法である (`libX11.a` は, ローカルにコピーしてたライブラリを使用している)。

3.2.3 実行時オプション

指定した実行時オプションは, Table 3.3 のとおりである (csh/tcsh の場合であり, 他の shell を使用する場合は, 適宜変更する必要がある)。

Table 3.3 コンパイル・オプション

計算機	指定オプション
AP3000	<code>setenv FORT90L '-Wl,-Ls'</code>

AP3000 の場合, `signal` でエラーが発生しないように (および経過時間計測用ルーチン `time`, 実行時オプション `-Wl,-Ls` を指定する必要がある。なお, 実行時オプションは, Table 3.3 のように, MCNP4B の入力データと区別するために, `mcnp -Wl,-Ls` と指定しない。

3.3 実行結果の確認

実行結果の確認方法は、以下のとおりである。

- 逐次版 MCNP4B を実行し、サンプル結果（あるいは SR2201 の結果）と比較する。
- 数値誤差による結果の違いに留意して、比較する。
- 標準出力と機番 32 (inp04o の出力) の出力内容を比較し、実行結果を判断する。

3.3.1 サンプル結果との比較

各並列計算機の実行結果とサンプル結果とを比較したのが Table 3.4 である。

Table 3.4 サンプル結果との比較

データ	SR2201	SP2	AP3000	Paragon
inp01	S 1	S 1	S 1	S 1
inp02	S 1	S 1	S 1	S 1
inp03	S	S	S	S
inp04	S 1	S 1	S 1	S 1
inp05	S 1	S 1	S 1	S 1
inp06	S	S	S	S
inp07	S 1	S 1	S 1	S 1
inp08	S 1	S 1	S 1	S 1
inp09	S	S	S	S
inp10	S 1	S 1	S 1	S 1
inp29	S 2	S 2	S 2	S 2

S : 完全に一致

S 1 : 仮数部の小数点第 4 位まで一致

S 2 : tracks の値等が異なっている

3.3.2 SR2201 の実行結果との比較

SR2201 の実行結果とそれ以外の並列計算機との実行結果を比較したのが、Table 3.5 である。

ここで、Paragon の結果の中で、inp01/inp07/inp29 の値が SR2201 と多少異なっているが、これは除算を高速に処理するオプション -Knoieee を指定したことが原因である（除算を高速に処理するために、IEEE に準拠しない値となる）。本オプションを指定しない場合、SR2201 の値と完全に一致することを確認している。

Table 3.5 SR2201 の結果との比較

データ	SP2	AP3000	Paragon
inp01	R	R	R
inp02	R	R	R
inp03	R	R	R
inp04	R 1	R	R
inp05	R	R	R
inp06	R	R	R
inp07	R	R	R 3
inp08	R 1	R	R
inp09	R	R	R
inp10	R	R	R
inp29	R 2	R	R 2

R : 完全に一致

R 1 : 仮数部の小数点第 4 位まで一致

R 2 : tracks の値等が異なっている

R 3 : 一箇所結果が異なっている

3.4 性能比較

MCNP4B の逐次版を各スカラー並列計算機で実行し、その性能を比較した。

3.4.1 測定方法

時間計測方法は、`xclock` 等の各スカラー計算機固有の経過時間計測ルーチンを使用し、経過時間（表中の `elap` に内容）を測定した（参考に、UNIX の `time` コマンドの出力結果表示している）。

3.4.2 逐次版の測定結果

テスト・データの逐次版の時間計測結果は、Table 3.6 のとおりである。表中の `xs (bin)` と `xs (ascii)` は、それぞれ核断面積テーブルを格納しているファイルの形式が、binary 形式か `ascii` 形式かを示している。

Table 3.6 逐次版の時間測定結果

データ		SR2201		SP2		AP3000		Paragon	
		xs (bin)	xs (ascii)	xs (bin)	xs (ascii)	xs (bin)	xs (ascii)	xs (bin)	xs (ascii)
inp01	real	7.00	12.00	7.07	10.35	2.13	3.21	28.00	50.00
	user	5.58	8.35	4.26	6.63	1.81	2.91	19.42	31.40
	sys	0.14	0.19	0.12	0.24	0.03	0.09	0.75	2.00
	elap	6.64	11.99	6.83	10.15	2.09	3.17	28.00	50.00
inp02	real	16.00	18.00	14.72	17.81	5.89	5.69	65.00	88.00
	user	11.15	13.75	8.25	10.46	3.67	4.66	40.79	53.40
	sys	0.37	0.36	0.29	0.53	0.11	0.19	2.60	3.20
	elap	15.90	18.48	14.67	17.75	5.65	5.84	65.00	88.00
inp03	real	12.00	21.00	10.72	15.98	4.93	6.90	38.00	87.00
	user	10.98	17.05	9.59	13.91	4.48	6.47	35.70	58.12
	sys	0.11	0.20	0.09	0.30	0.02	0.12	0.56	1.75
	elap	12.03	20.59	10.69	15.93	4.89	6.86	38.00	87.00
inp04	real	23.00	37.00	22.76	32.55	8.97	14.37	87.00	185.00
	user	21.46	33.53	21.06	30.00	8.46	13.47	82.89	132.15
	sys	0.10	0.36	0.08	0.61	0.04	0.24	0.49	3.27
	elap	22.61	37.03	22.71	32.50	8.93	14.33	87.00	186.00
inp05	real	6.00	10.00	5.02	9.10	1.58	3.26	19.00	51.00
	user	4.14	8.29	3.48	7.45	1.28	2.92	12.97	34.46
	sys	0.25	0.19	0.11	0.26	0.05	0.13	1.70	1.23
	elap	6.01	9.79	4.97	9.04	1.54	3.22	19.00	51.00
inp06	real	9.00	12.00	7.07	9.41	2.70	3.73	37.00	54.00
	user	8.26	10.46	5.86	8.14	2.43	3.48	27.90	38.92
	sys	0.10	0.15	0.02	0.19	0.06	0.07	0.49	1.48
	elap	9.17	11.65	7.02	9.36	2.66	3.69	31.00	54.00
inp07	real	20.00	34.00	21.03	31.96	6.94	11.67	73.00	178.00
	user	17.64	30.38	15.84	26.45	6.28	11.14	59.73	117.74
	sys	0.35	0.37	0.25	0.73	0.16	0.19	1.60	3.43
	elap	20.10	34.05	20.92	31.84	6.89	11.63	73.00	178.00
inp08	real	8.00	24.00	10.93	21.91	4.09	9.03	44.00	147.00
	user	6.12	20.78	8.04	18.38	3.55	8.43	31.42	87.71
	sys	0.26	0.40	0.12	0.70	0.12	0.19	1.74	4.15
	elap	7.98	24.06	10.85	21.85	4.05	8.98	45.00	148.00
inp09	real	10.00	20.00	13.74	25.02	3.70	6.53	39.00	112.00
	user	7.52	16.60	9.13	19.73	2.94	5.92	28.60	60.30
	sys	0.26	0.32	0.13	0.63	0.07	0.15	1.49	2.72
	elap	9.56	19.61	13.52	24.74	3.62	6.45	40.00	112.00
inp10	real	14.00	30.00	18.47	28.32	6.40	11.26	68.00	160.00
	user	12.90	26.64	15.70	24.96	5.88	10.61	54.21	103.54
	sys	0.13	0.36	0.09	0.64	0.06	0.16	2.24	4.17
	elap	14.07	29.64	18.39	28.25	6.36	11.22	68.00	160.00
inp29	real	41.00	107.00	39.99	53.01	14.52	19.64	137.00	160.00
	user	38.21	56.00	36.45	46.94	13.79	18.52	121.22	103.54
	sys	0.32	1.72	0.08	0.80	0.09	0.24	1.64	4.17
	elap	41.32	104.55	39.72	52.58	14.36	19.58	138.00	160.00

テスト・データ inp04 の粒子数 (nps) を 10000 と 20000 に増加させて、時間測定した結果を Table 3.7 に示す。なお、粒子数を増加させた実行結果は、オリジナルの inp04 の場合と同様に SP2 以外の計算機間の実行結果は一致している。この表中の i/o 時間は、(real-user-sys) の値である。

Table 3.7 逐次版の時間計測結果 (その 2)

測定 項目	nps=10000				nps=20000			
	SR2201	SP2	AP3000	Paragon	SR2201	SP2	AP3000	Paragon
real	173.35	164.18	72.49	686.00	340.00	312.63	135.85	1330.00
user	164.76	159.30	66.25	652.73	329.66	307.50	132.50	1287.67
sys	0.44	0.05	0.04	1.95	0.60	0.12	0.09	6.53
i/o	8.25	4.43	6.20	33.22	9.74	5.01	3.26	36.79

実際に使用しているデータである人体に対する被曝線量計算問題 (inp100 と呼ぶ) の計測結果は、以下のとおりである。計算結果については、機番 32 の最後の出力結果である tally の 4 と 6 の値を比較した結果から、精度上問題ないと考えている。すなわち、tally 6 の平均値は、各計算機上で値が異なっているものの、 1σ の範囲で一致している。また、vov(Variance of variance) や slope の値も一致している。

Table 3.8 逐次版の時間計測結果 (その 3)

data		SR2201	SP2	AP3000	Paragon
inp100	real	1352.00	1388.02	590.74	5233.00
	user	1323.05	1362.77	574.05	5077.25
	sys	1.75	0.28	0.21	28.67
	elap	1351.10	1387.95	590.24	5233.00

4. MCNP4B の MPI 化

4.1 PVM 版 MCNP4B の構造

PVM 版のツリー構造は、 Fig.4.1 のとおりである。

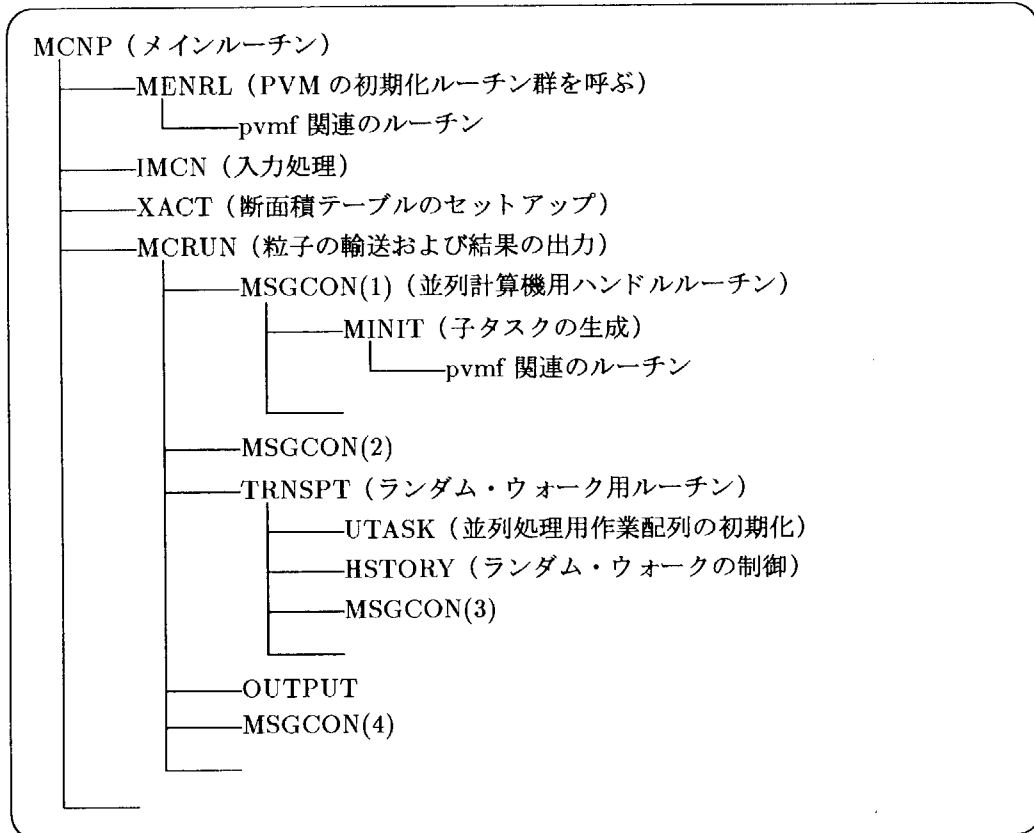


Fig. 4.1 PVM 版 MCNP4B のツリー構造

PVM 版 MCNP4B の各粒子に対するランダム・ウォーク・シミュレーションの並列制御における計算フローは、以下のとおりである（樋口ら [3]）。

- (1) 入力データや断面積の読み込み・内部変換等の前処理（マスタ）
- (2) 体系を構成する領域の幾何学形状データ等の計算に必要なデータの送信（マスタ）
- (3) マスターから送られたデータ類の受信（スレーブ）
- (4) 与えられた数の粒子シミュレーション（マスターおよびスレーブ）
- (5) 統計データの送信（スレーブ）
- (6) 統計データの受信・集計および計算結果の出力（マスタ）

MCNP4B で使用している PVM 関数とその機能は Table 4.1 のとおりである。

Table 4.1 MCNP4B コードで使用している PVM 関数

PVM 関数名	機 能
pvmfkill	指定されたタスクを終了し，削除する。
pvmfspawn	新規のタスクを起動する。
pvmfconfig	現在設定しているバーチャルマシンに関する情報を返す。
pvmfexit	タスクが PVM を終了することを通知する。タスクは削除しない。
pvmfnytid	最初の呼び出しでは，呼び出したタスクを PVM に登録し，呼び出しタスクのタスク識別子を引数に返す。以降の呼び出しでは，呼び出しタスクのタスク識別子を引数に返す。
pvmfparent	指定されたタスクの親のタスク識別子を返す。
pvmfstat	指定されたタスクの状態を返す。
pvmfinitend	デフォルトの送信バッファを作成し，エンコード方法を指定する。
pvmfbuinfo	指定されたメッセージバッファに関する情報を返す。
pvmfmcast	指定されたプロセス群にマルチキャストする。
pvmfpack	与えられたデータ型の配列をデータをメッセージバッファにパックする。
pvmfrecv	メッセージバッファの内容を受信する。
pvmfsend	メッセージバッファの内容を送信する。
pvmfunpack	与えられたデータ型の配列をメッセージバッファからアンパックする。
pvmftasks	バーチャルマシン上で実行しているタスクに関する情報を返す。
pvmfprobe	メッセージが到着したか否かをチェックする。
pvmfsetrbuf	現受信メッセージバッファを切り替え，前のメッセージバッファ識別子を返す。

4.2 負荷分散方法について

粒子輸送モンテカルロ法は，粒子間の相互作用を考慮しないので各粒子の計算を独立に行なうことができるので，MCNP4B は，静的な負荷分散と動的な負荷分散を組み合わせることにより負荷の均衡を図っている。

- (1) 臨界計算の場合は世代毎，固定源問題の場合は PRDMP 単位に負荷分散の手法を切り替える。
- (2) 初回は静的な負荷分散を適用するが，それ以降は粒子の配分結果を考慮しながら，動的な負荷分散を採用する。ただし，連続して静的な負荷分散とならないような処理にしている。

4.3 MCNP4B の MPI 化

ここでは、PVM 版 MCNP4B を基に、MPI 版への移植方法について述べる。

4.4 MPI 化の基本方針

PVM 版は、マスタ・スレーブ型の構造をしている。マスタは、主として、並列制御および収集したデータ処理を行ない、スレーブは、割り当てられた数の粒子シミュレーションを実施している。

MPI 化の基本方針は、以下のとおりである。

- 並列化の方法は、PVM 版と同じ方法を採用する。
 - － マスタ・スレーブ型で並列化する。
 - － PVM 版の負荷分散法を採用する。
- PVM の関数を MPI の関数で置き換える。
- 対応する MPI 関数がない場合は、代替案を考える。

4.5 MPI 化の方法

PVM 版から MPI 化する上で、両メッセージ・パッシング・ライブラリの違いは、次のとおりである。

- 異なったデータ型をパックあるいはアンパックすることにより、1つのメッセージ・バッファで送受信できる。
- 複数のメッセージ・バッファを有することができ、適宜バッファを切り替えることにより、複数のバッファを参照することができる。

PVM 版では、上記の特徴を活用して、以下の方法で採用している。

- (1) データをパック／アンパックすることで、データ型の異なったデータを一括して送信／受信している。
- (2) 複数の受信バッファを適宜切り換えることで (pvmfsetrbuff 関数を使用)、データの処理順序 (粒子の参照順序) を保証している。

上記の相違点に対する MPI 版での対応は、以下のとおりである。

- (1) データ型に応じて、4種類 (単精度実数型、倍精度実数型、整数型および文字型) のメッセージ・バッファを用意する。
- (2) バッファを余分に確保したために、tag を追加した (既存の tag 番号にデータ型に応じて、オフセットを設けた)。

- (3) 必要なデータのみ送信（あるいは受信）する。このため、事前にデータ長を送信しておく。
- (4) データのパック/アンパックは、MPI 関数を使用している（バッファ領域に直接書き込みをしていない）。
- (5) 各バッファへプロセッサ順に受信データを追加書きする。
- (6) 受信バッファを切り換える替りに、各バッファのポジションを変更して参照している。

PVM 関数に対応する MPI 関数あるいは代替案は Table 4.2 のとおりである。

Table 4.2 MCNP4B コードにおける PVM 関数

PVM 関数名	MPI 関数名	移植方法
pvmfkill	なし	コード内で参照していないので、移植していない。
pvmfspawn	なし	mpi_initにて、全プロセスを起動させることにより、対応している。
pvmfconfig	なし	使用していない。
pvmfexit	mpi_finalize	引数を対応させている。
pvmfmytid	mpi_comm_rank	rank をタスク識別子として指定している。
pvmfparent	mpi_comm_rank	rank=0 の場合、PvmNoParent としている。
pvmfpstat	なし	常に、PvmOk(0) としている。
pvmfinit send	なし	バッファのポジションを 0 設定することにより、対応している。
pvmfbuinfo	なし	使用している値のみ rank の値を使用している。
pvmfmcast	mpi_bcast	mpi_send を使用している。
pvmfpack	mpi_pack	データ型により 4 種類用意している。
pvmfrecv	mpi_recv	データ型により 4 種類用意している。
pvmf send	mpi_send	データ型により 4 種類用意している。
pvmfunpack	mpi_unpack	データ型により 4 種類用意している。
pvmftasks	なし	使用していない。
pvmfprobe	mpi_iprobe	引数を対応させている。
pvmfsetrbuf	なし	バッファのポジションを適宜変更することで、代替処理している。

データ送信ルーチン (msend) および PVM の機能を MPI で書き換えた例 (マスターとスレーブ) を、それぞれ、Fig.4.2, Fig. 4.3 および Fig.4.4 に示す。


```

      subroutine msend(mn,mt,in)
c      perform function 'send message'.
      :
#ifdef MPI
      if(mn.lt.0)call pvmfmcst(nsub,itid(2),mt,in)
      if(mn.ge.0)call pvmsend(itid(mn+1),mt,in)
#else
      param(1)=ipos
      param(2)=dpos
      param(3)=rpos
      param(4)=cpos
      if( mn.lt.0 ) then
        do 100 np = 1, size-1
          call mpi_send( param,4,mpi_integer,np,mt+100000,
1             mpi_comm_world,in )
          if( ipos.ne.0 ) then
            call mpi_send( ibuff,ipos/4,mpi_integer,np,mt+110000,
1             mpi_comm_world,in )
          endif
          if( dpos.ne.0 ) then
            call mpi_send( dbuff,dpos/8,mpi_double_precision,np,
1             mt+120000,mpi_comm_world,in )
          endif
          if( rpos.ne.0 ) then
            call mpi_send( rbuff,rpos/4,mpi_real,np,mt+130000,
1             mpi_comm_world,in )
          endif
          if( cpos.ne.0 ) then
            call mpi_send( cbuff,cpos,mpi_character,np,mt+140000,
1             mpi_comm_world,in )
          endif
100      continue
        endif
        if( mn.ge.0 ) then
          call mpi_send( param,4,mpi_integer,itid(mn+2),
1             mt+100000, mpi_comm_world,in )
          if( ipos.ne.0 ) then
            call mpi_send( ibuff,ipos/4,mpi_integer,itid(mn+2),
1             mt+110000,mpi_comm_world,in )
          endif
          if( dpos.ne.0 ) then
            call mpi_send( dbuff,dpos/8,mpi_double_precision,
1             itid(mn+2),mt+120000,mpi_comm_world,in )
          endif
          if( rpos.ne.0 ) then
            call mpi_send( rbuff,rpos/4,mpi_real,itid(mn+2),
1             mt+130000,mpi_comm_world,in )
          endif
          if( cpos.ne.0 ) then
            call mpi_send( cbuff,cpos,mpi_character,itid(mn+2),
1             mt+140000,mpi_comm_world,in )
          endif
        endif
      endif
#endif
      return
      end

```

Fig. 4.2 PVM から MPI への書き換え例

```

c      flag this task as completed and assign the next microtask.
430 call mrecv(30,i)
#ifdef MPI
      call mrcvi(nh)
#else
      nh=i
#endif
      call mgeti(mt,1,i)
      nc(mt,2)=nh
      nk=nk+1
      mt=nk+ntasks
      call msndi
      call mputi(mt,1,i)
      call msend(nh,31,i)
      if(mt.le.nm)nc(mt,2)=-nh
      if(nk.lt.nm)go to 360

c
c      all microtasks complete, send a signal to subtasks.
      call msndi
      call mputi(-ntasks,1,i)
      call msend(-1,31,i)

c
c      collect data from subtasks, first come first serve.
      n0=nps
      do 510 nt=1,ntasks
#ifdef MPI
      call mrecv(33,i)
#else
      call mrecv(33,nt)
#endif
#endif

```

Fig. 4.3 msgcon の修正内容 (マスタ側の処理内容)

```

c      obtain the buffer identifier for each subtask.
#ifdef MPI
    do 520 nt=1,ntasks
        call mrecv(34,i)
        call mrcvi(nh)
        ib(nh)=i
    520 call mrecb(0,ib(nh))
#else
    i=-1
    call mrecv(34,i)
#endif
c
c      step through the microtasks in order - to preserve tracking.
do 580 mt=1,nm
    nh=nc(mt,2)
    ipos=loc(1,nh)
    dpos=loc(2,nh)
    rpos=loc(3,nh)
    cpos=loc(4,nh)
    nc(nh,3)=nc(nh,3)+1
#ifdef MPI
    call mrecb(ib(nh),i)
#endif
c
c      receive output file messages.
530 call mgetc(hf,j)
    if(hf(1:5).eq.'$end$')go to 540
    write(iuo,'(a130)')hf
    go to 530
540 if(issw.eq.0)go to 560
c
c      receive ssw data.
    call mgeti(n,1,i)
    do 550 k=1,n
        call mgetd(rg,nd,j)
    550 write(iusc)(rg(i),i=1,nd)
#ifdef MPI
560 if(nsr.ne.71)go to 580
#endif
560 if(nsr.ne.71)go to 575
#endif
c
c      receive kcode source data.
    call mgeti(n,1,i)
    do 570 k=1,n
        call mgetd(fso(lfso+6*(ist-1)+1),6,i)
        if(ist.eq.msrk)ist=0
    570 ist=ist+1
#ifdef MPI
580 call mrecb(0,ib(nh))
#endif
575 continue
    loc(1,nh)=ipos
    loc(2,nh)=dpos
    loc(3,nh)=rpos
    loc(4,nh)=cpus
580 continue
#endif

```

Fig. 4.4 msgcon の修正内容 (スレーブ側の処理内容)

5. MPI版の動作確認および性能測定

5.1 各スカラー計算機への移植

PVM 版 MCNP4B を基に、SR2201 で移植した MPI 版を他のスカラー並列計算機に移植した。変更した内容は、逐次版から 4 種類のスカラー並列計算機に適用した内容のみであり、MPI に関する部分は基本的に変更する必要はなく、そのまま動作している。

5.2 実行結果の確認

並列版（主として、MPI 版であるが、機種によっては、PVM 版も含む）の動作確認方法は、以下のとおりである。

5.3 実行環境

5.3.1 並列版のテスト・データ

使用したテスト・データは、以下のとおりである。

- MCNP4B に附属サンプル・データ (inp04)
- 並列版の並列化効率を見るために、inp04 の粒子数を増加させたデータ

5.4 実行結果の確認方法

実行結果の確認方法は、以下のとおりである。

- SR2201 で逐次版 MCNP4B を実行し、その結果を基準とする。
- 逐次版の結果を基に、PVM/MPI 版の実行結果を比較する。
- 数値誤差による結果の違いに留意して、比較する。
- 比較する実行結果は、標準出力と機番 32 (inp04o の出力) の出力内容である。

5.4.1 並列版の動作確認

同一スカラー並列計算機内では、並列版の実行結果は、逐次版と基本的に一致している。相違点は、並列版に個々のプロセッサで実行された粒子数が出力される点と、標準出力に以下のようなメッセージが出力されることもある点である。このことは、問題とならないことを確認している。

```
warning. tally not scored beyond last energy bin.
```

並列版は、逐次版と異なって、各プロセッサで実行した粒子数の情報が出力される。MCNP4B のロード・バランスのアルゴリズムにより、各プロセッサで計算する粒子数が、必ずしも一致する訳ではない。

5.5 並列版 MCNP4B の実行時間の計測

MCNP4B の並列版 (PVM 版および MPI 版) を各スカラー並列計算機で実行し、その実行時間を測定した。

5.5.1 測定方法

テスト・データ inp04 の粒子数 (nps) を変更し、並列ジョブ (PVM/MPI) の実行時間を測定した。測定した項目は、以下のとおりである。

- プロセッサ台数 (1PE, 3PE から 16PE まで適宜)
- 粒子数 (10000 と 20000)

なお、コーディング方法により、マスタとスレーブがそれぞれ 1 台となる構成は存在しない (このような指定をした場合、マスタ・プロセッサだけで処理が実行され、基本的にスレーブ・プロセッサは何もしない)。

5.5.2 SR2201 測定結果 (PVM 版)

SR2201 の PVM 版の測定結果は、Table 5.1 のとおりである。ここで、全プロセッサ台数とは、マスタ・プロセッサとスレーブ・プロセッサの合計を意味する。

Table 5.1 SR2201 の PVM 版の経過時間と速度向上率

全プロセッサ 台数	nps=10000		nps=20000	
	経過時間 (秒)	速度向上率	経過時間 (秒)	速度向上率
1	168.18 (170.00)	1.00	342.67 (343.00)	1.00
3	101.27 (103.00)	1.66	194.80 (196.00)	1.76
4	71.97 (74.00)	2.34	135.87 (137.00)	2.52
6	48.42 (50.00)	3.47	91.96 (93.00)	3.73
8	38.64 (40.00)	4.35	68.46 (70.00)	5.01
16	38.79 (40.00)	4.33	46.38 (48.00)	7.39

ここで、経過時間は、コード内でタイム・ルーチンによる計測結果と括弧内の time コマンドの real の値を掲げている。速度向上率は、逐次版の経過時間を並列版の経過時間で割った値である。

5.5.3 SR2201 の測定結果 (MPI 版)

SR2201 の MPI 版の測定結果は, Table 5.2 のとおりである.

Table 5.2 SR2201 の MPI 版の経過時間と速度向上率

全プロセッサ 台数	nps=10000		nps=20000	
	経過時間 (秒)	速度向上率	経過時間 (秒)	速度向上率
1	168.18 (170.00)	1.00	342.67 (343.00)	1.00
3	85.39 (100.00)	1.94	160.20 (160.00)	2.14
4	61.34 (76.00)	2.74	113.64 (129.00)	3.02
6	40.84 (56.00)	4.12	80.92 (89.00)	4.23
8	31.40 (48.00)	5.36	56.28 (72.00)	6.09
16	31.37 (42.00)	5.36	39.05 (57.00)	8.78

5.5.4 SP2 の測定結果 (MPI 版)

SP2 の MPI 版の測定結果は, Table 5.3 のとおりである.

Table 5.3 SP2 の MPI 版の経過時間と速度向上率

全プロセッサ 台数	nps=10000		nps=20000	
	経過時間 (秒)	速度向上率	経過時間 (秒)	速度向上率
1	160.18 (160.40)	1.00	312.42 (312.63)	1.00
3	87.73 (91.67)	1.83	165.22 (168.13)	1.89
4	61.08 (64.39)	2.62	114.30 (117.46)	2.74
6	43.85 (48.02)	3.65	78.57 (85.31)	3.66
8	33.06 (37.68)	4.85	60.23 (65.13)	5.19
16	21.88 (41.20)	7.32	34.74 (41.82)	9.00

5.5.5 AP3000 の測定結果 (MPI 版)

AP3000 にはノード間通信を行なう手順に, ユーザレベル通信 (ULT) と システムレベル通信 (SLT) の 2 種類がある. ULT 通信が AP-Net Native のプロトコルを使用しているため, 通信速度が速い. MPI 版は, ULT 通信と SLT 通信の両方を測定し, 結果をそれぞれ Table 5.4 と Table 5.5 に示している. ただし, ULT 通信の場合, aprun: ULT exception occurred in a parallel process エラーが発生し, 正常に動作しないことがあるため, mterm ルーチンの中で同期処理 (mpi_barrier ルーチン) を挿入している. なお, 現状, 東海研の AP3000 で並列ジョブとして使用できるのは, 最大 4 CPU までである.

Table 5.4 AP3000 の MPI 版の経過時間と速度向上率 (ULT)

全プロセッサ 台数	nps=10000 (ULT)		nps=20000 (ULT)	
	経過時間 (秒)	速度向上率	経過時間 (秒)	速度向上率
1	72.10 (72.49)	1.00	135.79 (135.85)	1.00
3	35.09 (37.09)	2.05	66.94 (68.94)	2.03
4	25.96 (27.86)	2.78	47.00 (49.25)	2.89

Table 5.5 AP3000 の MPI 版の経過時間と速度向上率 (SLT)

全プロセッサ 台数	nps=10000		nps=20000	
	経過時間 (秒)	速度向上率	経過時間 (秒)	速度向上率
1	72.10 (72.49)	1.00	135.79 (135.85)	1.00
3	38.02 (39.75)	1.90	69.68 (71.80)	1.95
4	31.20 (33.10)	2.31	58.87 (60.62)	2.31

5.5.6 AP3000 の測定結果 (PVM 版)

PVM 版 MCNP4 を AP3000 で実行したところ、以下のメッセージを出力し正常に動作しなかった。

```
libpvm[t0]:pvm_config():Not implemented
```

```
libpvm[t0]:pvm_tasks():Not implemented
```

PVM/AP で実行可能なように pvmfcong と pvmftasks を使用しないようにソースを修正し (暫定的な方法であり、正しい回避方法ではない)、実行した測定結果が Fig.5.6 である。

Table 5.6 AP3000 の PVM 版の経過時間と速度向上率 (ULT)

全プロセッサ 台数	nps=10000		nps=20000	
	経過時間 (秒)	速度向上率	経過時間 (秒)	速度向上率
1	72.10 (72.49)	1.00	135.79 (135.85)	1.00
3	35.71 (37.59)	2.02	68.50 (68.33)	1.98
4	24.44 (26.45)	2.95	46.17 (48.35)	2.94

5.5.7 Paragon の測定結果 (MPI 版)

Paragon の MPI 版の測定結果は、Table 5.7 のとおりである。

5.5.8 速度向上率の比較

各スカラー計算機の MPI 版の性能を比較したのが、Fig.5.1 である。inp04 で粒子数を 20000 にしたデータを使用した結果である。

Table 5.7 Paragon の MPI 版の経過時間と速度向上率

全プロセッサ 台数	nps=10000		nps=20000	
	経過時間 (秒)	速度向上率	経過時間 (秒)	速度向上率
1	664.00 (686.00)	1.00	1322.00 (1330.00)	1.00
8	128.73 (135.00)	5.16	226.99 (253.00)	5.82
16	92.08 (104.00)	7.21	148.50 (156.00)	8.90
32	91.61 (125.00)	7.25	141.87 (151.00)	9.32

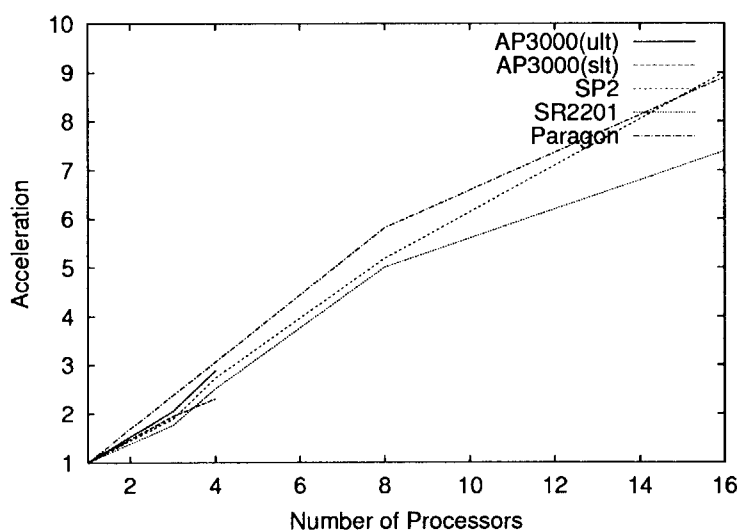


Fig. 5.1 スカラー計算機の実行性能比較 (MPI 版)

Fig.5.1 から以下のことが明らかとなった。

- 性能が良いのは、プロセッサが8台までで、それ以上になると、性能は低下する。
- SP2 と Paragon が速度向上率は比較的良いが、SR2201 は悪い。
- AP3000 はプロセッサ4台しか無く、性能の優劣は不明である。

速度向上率があまり良くないのは、データ転送量と比較して粒子シミュレーションにかかる CPU 時間が少ないことが原因の1つと考えられる。定量的なチェックはしていないが、粒子により演算量が異なるために、負荷にアンバランスが発生したことも要因と考えられる。また、プロセッサ台数が増大したことにとともに、逐次処理（入出力処理）の割合が増加し、並列化効率が低下したことも原因と考えられる。

6. 並列BMTコード開発のコンセプト

6.1 コンセプト

これまで述べてきたように、オリジナルの PVM 版を基に MPI 版を開発してきた。しかしながら、本コードは非常に巨大であり、計算結果を得るために多くの時間がかかる。このため、ユーザは計算したい問題の種類や規模に適した計算機を選択することが困難である。そこで、MCNP ユーザに性能測定結果および有用な情報を提供するために、並列BMTコードを開発することにした。そこで、開発の目的は、効果的な計算機が選択できるように MCNP ユーザを支援することである。

従って、本BMTコードは、以下のことを可能としている。

- 利用可能な計算機に対して性能評価の結果を提供すること。
- 問題の規模や種類に関連して、計算機の経過時間を見積るために支援を行なうこと。

この結果として、本コードは、問題の規模や種類に適した計算機が選択できるように支援することが可能である。

6.2 開発の手順

開発の手順は以下のとおりである。

- オリジナルの PVM バージョンと同じ並列化手法を採用すること。
 - － マスタ/スレーブ・モデルを採用する
 - － 負荷不均衡のレベルの応じた動的と静的な負荷バランス・モデルを採用する
- メッセージ・パッシング・ライブラリとして MPI を採用する。
- オリジナル版から解析対象としている幾何形状や使用している粒子に関連する処理のみ抽出し、種々のアクセサリ・ルーチンは削除すること。

7. MCNP4BK-BMT (臨界計算)

7.1 入力データの概要

7.1.1 入力データ

BMT 用に選択した入力データは、「3 連クラスタ」実験と呼ばれる、有名な臨界実験である [8]。19 × 16 の正方格子状に配列した UO₂ 燃料棒の集合 (クラスタ) 3 体を水中に 1 列に並べた体系である [7]。性能評価においては、1 世代あたり、500、1000 及び 2000 個の中性を 50 世代にわたって追跡した (文献 [7] に記述された入力モデルの場合、1500 個の中性を 110 世代にわたって追跡している)。

7.2 並列 BMT コード開発

7.2.1 並列 BMT コードのチューニング

並列版 MCNP4B は、コードの主要部分である粒子追跡を、マスタ・スレーブ方式で並列処理している。負荷分散については、主として動的スケジューリングにより粒子を各プロセッサに割り当てるが、場合により静的スケジューリングを用いている。前世代の計算における負荷不均衡の度合がある基準内の時、次の世代の計算では静的スケジューリングが選択される。前世代の計算で静的スケジューリングを使用した場合、動的スケジューリングが選択される。粒子輸送計算における並列版モンテカルロ・コードの処理は、以下のとおりである。

i) 初期化 (In: Initialization)

マスタ・プロセッサで入力ファイル (形状データ、核断面積データなど) を読み込み、入力データをコードの内部表現に変換する。

ii) 粒子分配 (PD: Particle distribution)

上記データ及び粒子属性をマスタ・プロセッサからスレーブ・プロセッサに分配することで、スレーブ・プロセッサに粒子を割当る。

iii) 粒子追跡と統計データの計算 (PT: Particle tracking)

スレーブ・プロセッサで 3 次元空間内で粒子を追跡し、個々の粒子が領域の境界を横切るごとに、粒子束、エネルギー蓄積といった統計データを計算する。

iv) 統計データの集計 (TS: Totaling statistic data)

統計データをスレーブ・プロセッサからマスタ・プロセッサに送信する。

v) 統計データの出力 (Op: Output)

マスタ・プロセッサで計算結果や粒子追跡情報などをファイルに出力する。

SP2 上でプロセッサを 1 から 32 台まで変化させた場合の上記処理に対する実行時間を Fig. 7.1 に示す. 初期化は逐次処理のため, その実行時間 (E) は一定であり, プロセッサ台数 (N) とは無関係である. 粒子分配の E は N に比例して増加する. 以下に述べるように, ブロードキャストの方式を改良することにより, 実行時間はおおよそ $\log_2(N)$ に比例するところまで改善できる. 粒子追跡の E は, N に比例して減少する. 並列処理が有効であるが, 後で述べるようにより重要な要素は並列化効率である. データ集計の E は, N にとも増加する. しかしながら, 全実行時間と比較してそれほど多くない. 出力処理の E は, 初期化と同じように一定で, N には無関係である.

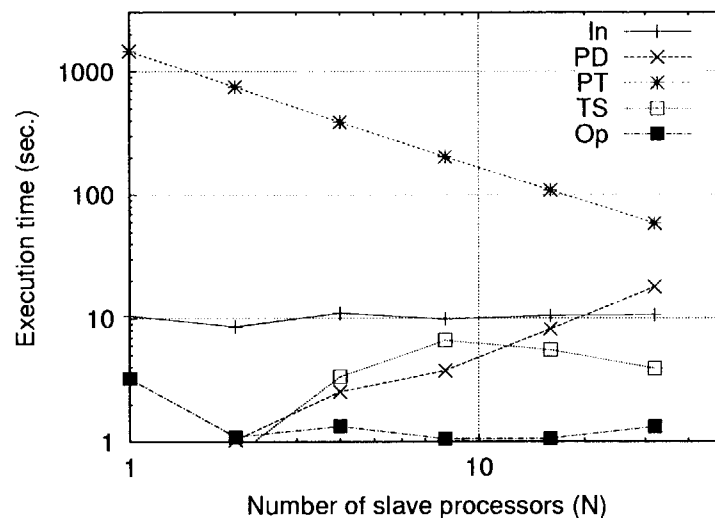


Fig. 7.1 SP2 上での処理別実行時間 (従来方式)

PVM 関数から MPI 関数への書き換えに加えて, プロセッサ間の通信方式を改良した. プロセッサが 8 台以上になると, PD と TS 実行時間が N とともに増大し, コードの性能を支配するようになる. この通信は, 各世代での計算を開始するたびに, マスタ・プロセッサから全スレーブ・プロセッサに 200KB のデータを転送する処理を行なっている. ブロードキャストの通信オーバーヘッドを削減するために, ブロードキャストの通信方式を逐次的にプロセッサ間通信を行なう従来方式からバイナリ・ツリー方式に変更した (樋口ら [3]). Table 7.1 に示すように, この改善により, PD の実行時間は, 改良前の $O(N)$ から $O(\log_2 N)$ となった.

7.3 ベンチマークテストの結果

7.3.1 コード全体の結果

Figure 7.2, 7.3 と 7.4 にそれぞれ SP2, Paragon 及び SR2201 上でプロセッサ数を 1 から 32 台まで変化させた場合の MCNP4BK-BMT の実行時間を示す. 世代数は 50, 1 世代あたりの粒子数は 500, 1000 及び 2000 個 (それぞれ, case-1, case-2 及び case-3 と呼ぶ) で

Table 7.1 粒子追跡処理の並列化効率

スレーブ・プロセッサ数	01	02	04	08	16	32
PD ^a の実行時間 (秒)	0.0	1.03	2.54	3.77	8.18	17.94
PD ^b の実行時間 (秒)	0.0	1.32	2.68	2.80	2.86	3.37
コード全対 ^a の実行時間 (秒)	1474.05	759.99	406.44	224.59	134.61	92.70
コード全対 ^b の実行時間 (秒)	1474.05	760.28	406.58	223.62	129.29	78.13
速度向上率 ^a	1.00	1.94	3.63	6.56	10.95	15.90
速度向上率 ^b	1.00	1.94	3.63	6.59	11.40	18.87

^a 従来方式

^b バイナリ・ツリー方式

ある。

全ての計算機に関する結果は同じ傾向、すなわち同程度の速度向上率を示している。コード全体の実行時間はプロセッサ台数 (N) とともに減少している。 N が大きくなった時、速度向上率が鈍 (ナマ) る理由は、In と Op が逐次処理、すなわち実行時間が一定で、 N に無関係のためであることおよび PD と TS が N とともに増加することである。

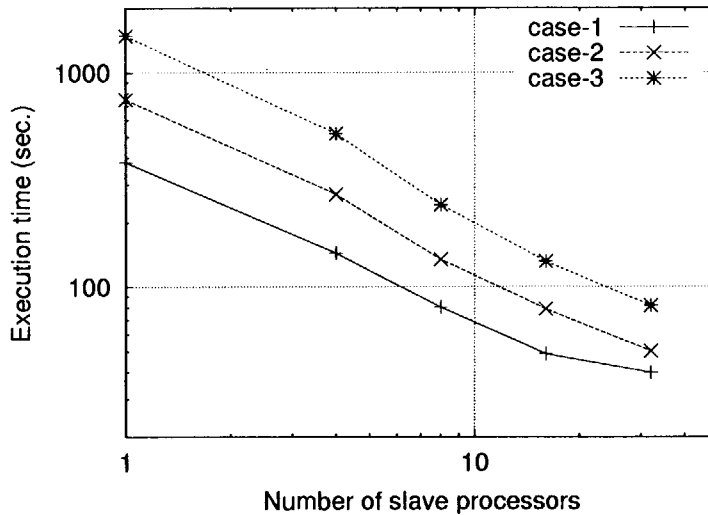


Fig. 7.2 SP2 上での BMT 測定結果

モンテカルロ・コードの主要な処理で完全に並列化可能な粒子追跡処理 (PT) の実行時間は、全計算機上で N におおむね比例して減少している。しかしながら、並列化効率 (P) は、 N とともに減少している。ここで、 P は、速度向上率を N で割った値と定義した。Table 7.2 は、case-3 で N を 1 から 32 まで変化させた時の PT に対する P を示したものである。 N が増加するにつれて、スレーブ・プロセッサに割り当てられる粒子数が減少し、バラツキが大きくなった結果、負荷不均衡の割合が増加し、性能が低下する (樋口ら [4])。

Figure 7.5 は case-3 でプロセッサを 32 台にした場合の SP2, Paragon, SR2201 におけ

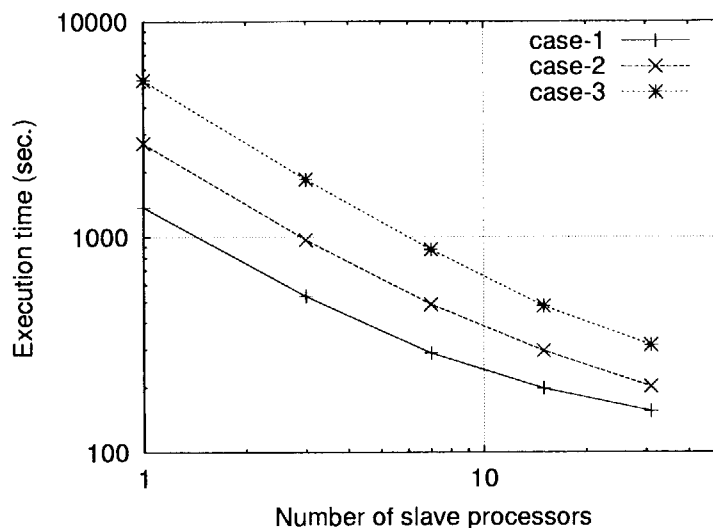


Fig. 7.3 Paragon 上での BMT 測定結果

Table 7.2 粒子追跡処理の並列化効率

スレーブ プロセッサ数	実行時間	並列化効率
01	1460.31 秒	100.0 %
02	749.92 秒	97.4 %
04	381.28 秒	95.7 %
08	198.33 秒	92.0 %
16	107.41 秒	85.0 %
32	59.15 秒	77.2 %

る PT の負荷不均衡を示したものである。負荷不均衡は、式 (7.1) で定義した F_{im} により測定した。全計算機上で F_{im} は同じ傾向を示している。すなわち、前述した入力データを用いた場合、ほとんどの値は 1.1 から 1.5 の範囲に入っている。この結果、実行時間が 20-30 % 増加した。

$$F_{im} = \frac{\text{実際の並列実行時間}}{\text{理想的な並列実行時間}} = \frac{\text{最大実行時間}}{\text{平均実行時間}} \quad (7.1)$$

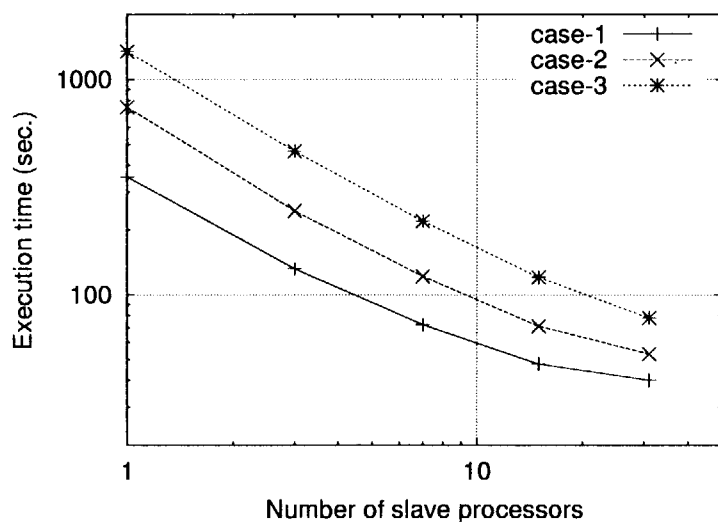


Fig. 7.4 SR2201 上での BMT 測定結果

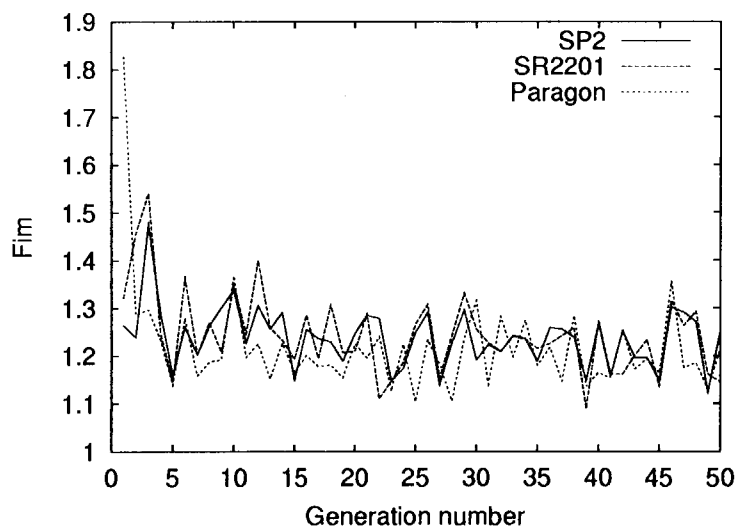


Fig. 7.5 負荷不均衡による性能低下

7.3.2 処理別の性能

ここでは、SP2, SR2201 および Paragon について、処理別の性能を示す。

SP2 の実行結果 (Fig. 7.6) は、バイナリ・ツリー方式に変更したことにより、従来方式と比較して、PD の処理時間が低減した。

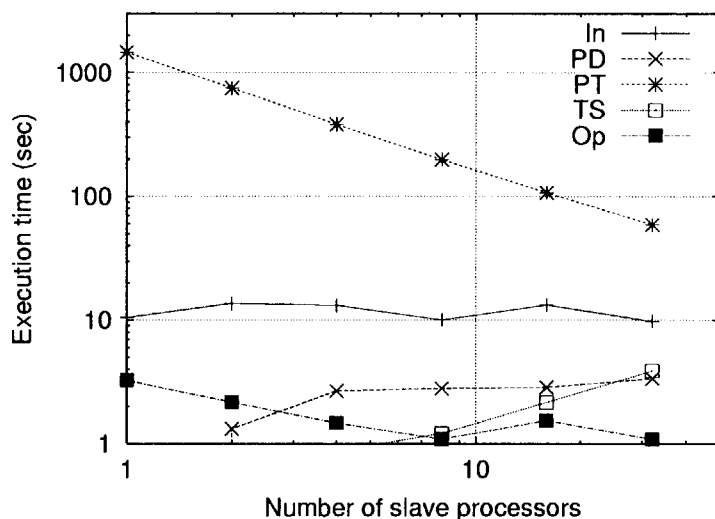


Fig. 7.6 SP2 上での処理別実行時間 (バイナリ・ツリー)

SR2201 の実行結果 (Fig. 7.7) と SP2 との比較結果は以下のとおりである。

- In の処理時間は、同程度である。
- PD の処理時間は、SR2201 の方が速い。
- PT の処理時間は、並列化され、同程度の性能である。
- TS の処理時間は、同程度であり、高速なネットワークの効果が得られていない。
- Op の処理時間は、SR2201 の方が速い。

Paragon の実行結果は、Fig. 7.8 のとおりである。PT 処理は他の 2 種類の並列計算機と同様の傾向を示しているが、CPU 性能が劣っているため、絶対性能は良くない結果となっている。

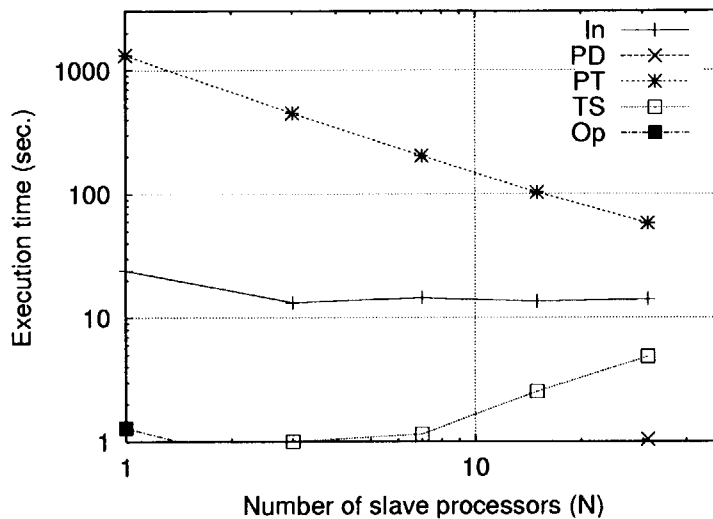


Fig. 7.7 SR2201 上での処理別実行時間 (バイナリ・ツリー)

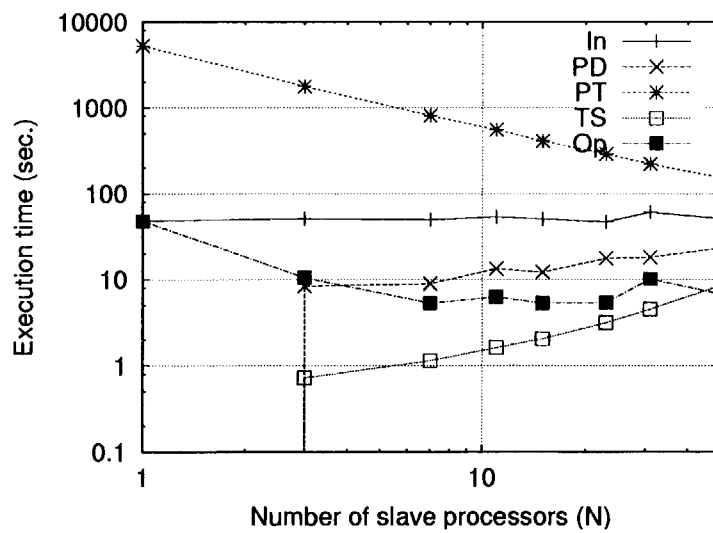


Fig. 7.8 Paragon 上での処理別実行時間 (バイナリ・ツリー)

7.3.3 参考：富士通 AP3000 の結果

富士通 AP3000 は運用の関係で使用可能なプロセッサ台数が少ない（最大 4 プロセッサ）が，動作確認をしたので測定結果を以下に示す。

コード全体の実行時間（Fig. 7.9 および付録 1 の測定結果の表を参照のこと）は他の 3 種類の並列計算機よりも良い結果が得られている。これは，AP3000 の単体 CPU 性能が一番良いことが原因と考えられる。しかしながら，速度向上率（グラフの傾き）は他の並列計算機と同等の傾向を示している。

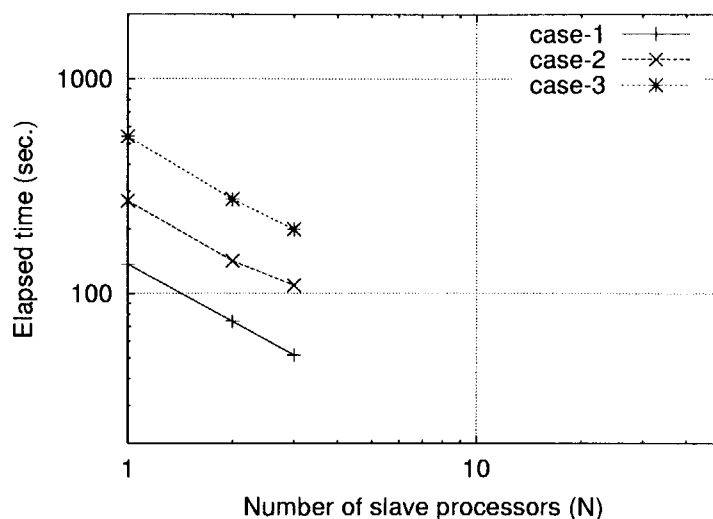


Fig. 7.9 AP3000 上での実行時間 (バイナリ・ツリー)

1 世代あたりの粒子数が 2000 個の場合における処理別実行時間（Fig. 7.10 参照）はプロセッサ台数が少ないが，PT の処理時間は他の並列計算機と同様に N の増加とともに減少する傾向がある。

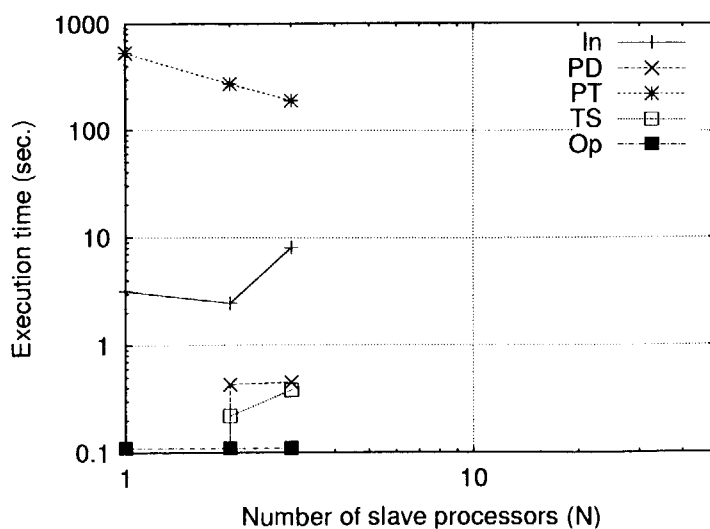


Fig. 7.10 AP3000 上での処理別実行時間 (バイナリ・ツリー)

8. MCNP4BF-BMT (固定源問題)

8.1 入力データ概要

8.1.1 入力データについて

今回ベンチマーク・テスト用に使用した計算モデルは、人体に対する線量等量計算である。人体模型は、米国オークリッジ国立研究所で開発された Cristy Phantom を用いた。人体模型は、肺、心臓、肝臓等の臓器と背骨、肋骨等の骨および骨髄、軟組織によって構成され、約 2000 枚の 2 次曲面の論理演算によって記述されていた約 100 の 3 次元領域によって表現されている。中性子の並行ビームを人体模型に照射した時の各臓器に対する線量等量を求めるために、0.1 MeV の中性子を追跡している (樋口ら [3])。

8.2 固定源問題の問題点

固定源問題の特長は、以下のとおりである。

- 取り扱う粒子数が多く (数百万個)、負荷分散が比較的容易である。

まず、100 万粒子の SP2 上でのコード全体の実行時間を Table 8.1 に示す。この結果から臨界計算と同様に SP2 と SR2201 とで同じ傾向を示すことがわかる。

Table 8.1 SP2 と SR2201 の測定結果の比較 (固定源問題)

スレーブ プロセッサ数	SP2 実行時間	SR2201 実行時間
01	1310.19 秒	1327.84 秒
03	491.82 秒	481.42 秒
07	251.14 秒	235.56 秒
15	149.54 秒	140.28 秒
31	104.02 秒	92.39 秒

次に、モンテカルロ法の主要な処理である粒子追跡の性能を Table 8.2 に示す。

Table 8.2 SP2 上での粒子追跡処理の測定結果 (固定源問題)

スレーブ プロセッサ数	実行時間	速度向上率	並列化効率
01	1266.99 秒	1.00	100.0 %
03	450.70 秒	2.81	93.7 %
07	208.75 秒	6.07	86.7 %
15	105.96 秒	11.96	79.7 %
31	58.67 秒	21.60	69.7 %

負荷バランスの向上を図る方法として、武宮の手法を検討しその有効性を確認した。本手法を適用した効果を次章で述べる。

9. 負荷分散手法の改良

9.1 MCNP4B の負荷分散手法

現行の MCNP4B が採用している負荷分散手法は、動的負荷分散と静的負荷分散手法を組合せた方法である。

固定源問題における負荷分散手法の判定は、PRDMP で指定された DMMP 回（デフォルトの指定値で 10 回）毎に実施される。負荷分散手法の決定方法は、以下のとおりである。

- i) 初回は静的な負荷分散手法を採用する（スレーブ・プロセッサ台数に分割する）。
- ii) 2 回目は、連続して静的な負荷分散手法を採用しない判定基準により、動的な負荷分散手法を採用する。この場合、分配する粒子をスレーブ・プロセッサ台数の 5 倍（ここでは、分配係数と呼ぶ）に分割し、処理が終了したプロセッサから順次粒子を分配する。
- iii) 各スレーブ・プロセッサに分配した回数のバラツキにより、次回の負荷分散手法を決定する。
- iv) 分配した回数の最大値と最小値の比率（整数値に変換し、最大は 3）の 3 乗した値を分配係数とする。
- v) 同様の処理を最後まで繰返す。

9.2 武宮の動的負荷分散法

本手法は、マスタースレーブ方式により粒子輸送モンテカルロ計算を並列処理した際、各スレーブ・プロセッサへ動的に分配するタスクサイズ（粒子数）を自動的に決定するアルゴリズムである。並列処理における性能低下の度合は、 H^L （動的な粒子分配による負荷分散の不均衡による性能低下）と H^C （通信オーバーヘッドによる性能低下）によって定量的に評価することができる。負荷不均衡のコストと通信のコストの和（ $H = H^L + H^C$ ）が小さいほど並列化効率が高い。そこで、本アルゴリズムは、各スレーブ・プロセッサごとに、各タスク時点までの計算時間および通信時間を用いてコスト関係 H が小さくなるように次のタスクの分配粒子数を決定する。具体的な方法は、武宮ら [5] を参照のこと。

9.3 MCNP4B コードにおける武宮の手法の効果

9.3.1 MPI 化

武宮の手法は、PVM を用いて Paragon 上で開発されたため、MPI 版の MCNP4B 用に移植した。変更の対象となったルーチンは、以下のとおりである。

- msgcon
- trnspt
- advijk
- getexm

また、FORTRAN 以外に C の関数群も使用しているが、必要最小限なもののみ使用している。

9.3.2 性能比較

オリジナルの MCNP4B 版ではスレーブ・プロセッサの I/O 関連の影響により、SR2201 および Paragon では、プロセッサ数が 8 台以上になると並列化効率が極端に悪くなることがわかっている。そこで、MPI 版 MCNP4B に武宮の手法を組み込んだ版を SP2 上で実行時間を測定し、両負荷分散手法の性能比較を実施した。

SP2 で従来の負荷分散手法と武宮の手法における実行時間を比較したのが、Table 9.1 である。表中の実行時間は、コード全体の数値である。この表から武宮の手法は、従来の手法より効率が良く（負荷不均衡が解消されている）、性能が改善されていることが明らかとなった。

Table 9.1 SP2 における MCNP4B の負荷分散法の違いによる性能比較

スレーブ プロセッサ数	従来の手法 実行時間	武宮の手法 実行時間
01	1325.95 秒	1325.95 秒
03	493.18 秒	493.32 秒
07	249.53 秒	233.09 秒
15	153.73 秒	142.51 秒
31	103.98 秒	91.81 秒

9.4 MCNP4BF-BMT コードにおける武宮の手法の効果

MCNP4B を基にした並列 BMT コード MCNP4BF-BMT を用い、従来の手法と武宮の手法での性能測定を実施し、その効果を検証した。本コードの場合、スレーブ・プロセッサへの I/O 関連の処理を削除しているので、両負荷分散手法の効果が直接比較することが可能である。

SP2 で従来の負荷分散手法と武宮の手法における実行時間を比較したのが、Fig. 9.1 である。図中の実行時間は、PT (Particle Tracking) 処理の値である。

SP2 上での各負荷分散手法における PT 処理の並列化効率を比較したのが、Table 9.2 である。この表から武宮の手法が非常に有効であると判断できる。

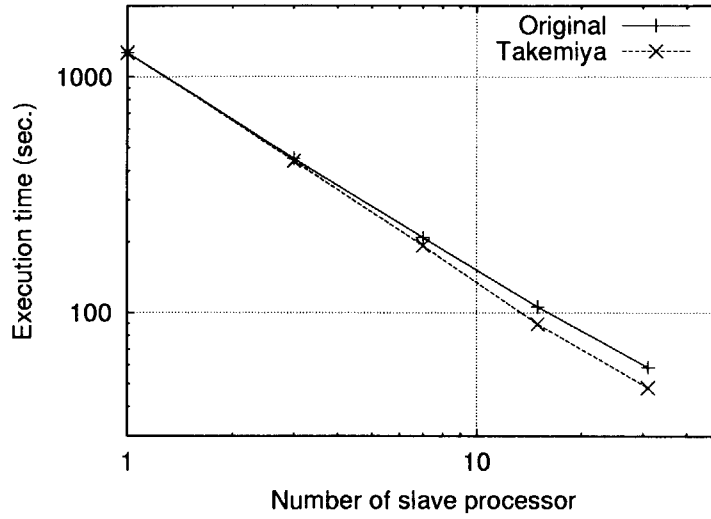


Fig. 9.1 SP2 における負荷分散法の違いによる性能比較

Table 9.2 SP2 上での両負荷分散手法における並列化効率の比較

スレーブ プロセッサ数	従来的手法			武宮の手法		
	実行時間	速度向上率	並列化効率	実行時間	速度向上率	並列化効率
01	1266.99	1.00	100.00 %	1266.99	1.00	100.00 %
03	450.70	2.81	93.69 %	440.59	2.88	95.86 %
07	208.75	6.07	86.71 %	192.55	6.58	94.00 %
15	105.96	11.96	79.71 %	89.54	14.15	94.33 %
31	58.67	21.60	69.66 %	48.08	26.35	85.01 %

次に、両負荷分散手法の負荷不均衡割合 F_{im} を比較したのが Fig. 9.2 である。比較したデータは、粒子数を 100 万個に固定し、プロセッサ台数が 16 と 32 の場合の結果を示している。

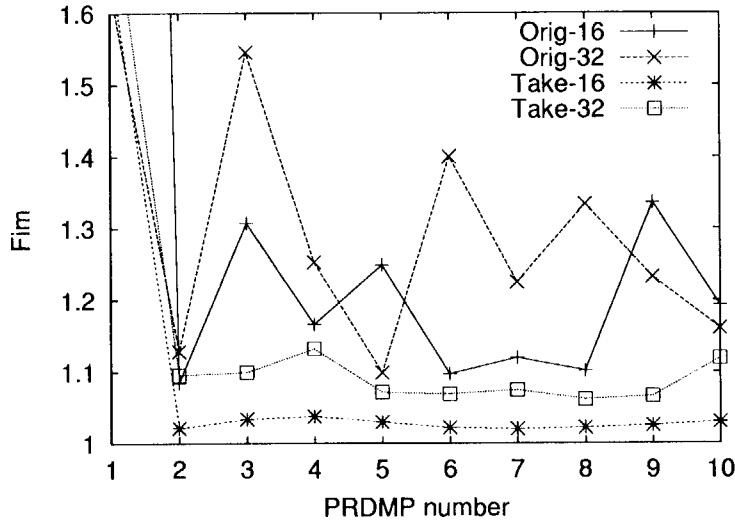


Fig. 9.2 SP2 上での両負荷分散手法における負荷不均衡割合の比較

Table 9.3 は、両負荷分散手法における分配された粒子の平均個数と標準偏差を示している。両者で標準偏差が倍以上も異なっており、この結果は武宮の手法による動的な負荷分散が効率よく行なわれていることを示している。

Table 9.3 SP2 上での両負荷分散手法における粒子分配の比較

スレーブ プロセッサ数	平均値	標準偏差	
		従来の手法	武宮の手法
03	333333.33 個	1.25 個	2839.98 個
07	142857.14 個	1528.74 個	3936.02 個
15	66666.66 個	1190.57 個	2440.37 個
31	32258.06 個	1468.86 個	2115.03 個

SP2 と同様に SR2201 における PT 処理の実行時間および並列化効率について両負荷分散手法の結果を比較したものが Fig. 9.3 および Table 9.4 である。なお、ベンチマーク・コードには、スレーブ・プロセッサの I/O 関連の処理を含んでいないため、プロセッサが 8 台以上の場合でも、性能は低下しない。

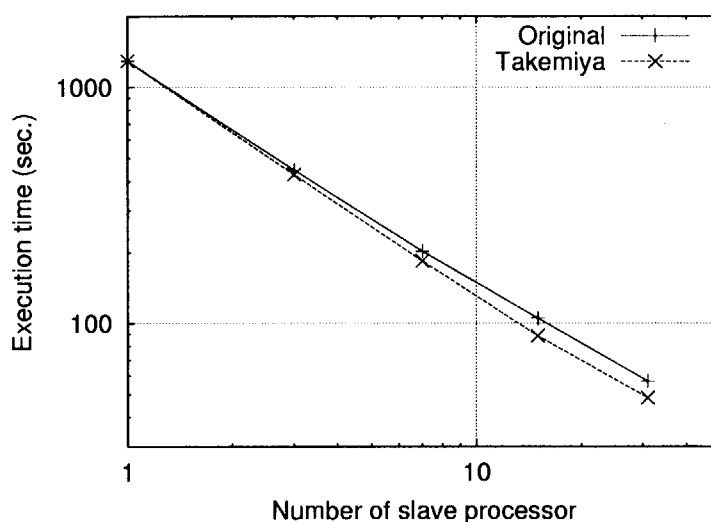


Fig. 9.3 SR2201 における負荷分散法の違いによる性能比較

Table 9.4 において、スレーブ・プロセッサ数が 7 以下の並列化効率が 100 % を超えているのは、通信と演算のオーバーラップを考慮せずに実行時間の測定していることが影響していると考えている。

Table 9.4 SR2201 上での両負荷分散手法における並列化効率の比較

スレーブ プロセッサ数	従来手法			武宮手法		
	実行時間	速度向上率	並列化効率	実行時間	速度向上率	並列化効率
01	1292.99	1.00	100.00 %	1292.99	1.00	100.00 %
03	446.58	2.90	96.51 %	426.67	3.03	101.01 %
07	202.20	6.39	91.35 %	183.83	7.03	100.48 %
15	105.12	12.30	82.00 %	88.45	14.62	97.46 %
31	56.79	22.77	73.44 %	48.31	26.76	86.34 %

SP2 と同様に Paragon における PT 処理の実行時間および並列化効率について両負荷分散手法の結果を比較したものが Fig. 9.4 および Table 9.5 である。

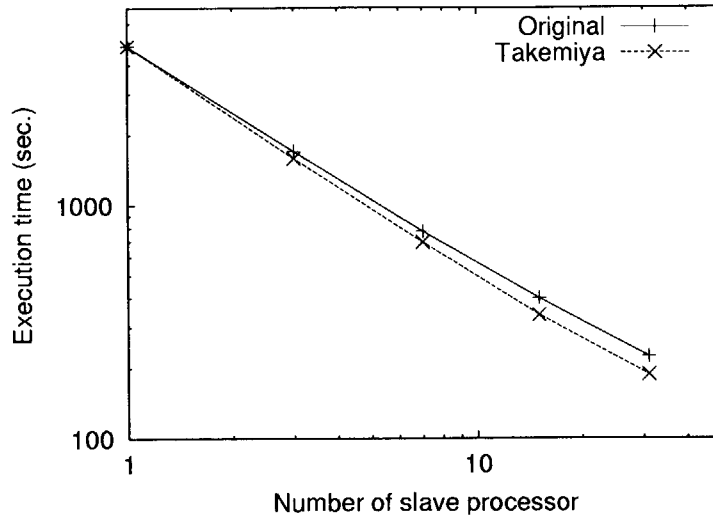


Fig. 9.4 Paragon における負荷分散法の違いによる性能比較

Table 9.5 Paragon 上での両負荷分散手法における並列化効率の比較

スレーブ プロセッサ数	従来的手法			武宮の手法		
	実行時間	速度向上率	並列化効率	実行時間	速度向上率	並列化効率
01	4806.21	1.00	100.00 %	4806.21	1.00	100.00 %
03	1706.42	2.82	93.88 %	1589.19	3.02	100.81 %
07	774.11	6.21	88.70 %	695.83	6.91	98.67 %
15	399.52	12.03	80.20 %	338.85	14.18	94.56 %
31	224.71	21.39	69.00 %	187.89	25.58	82.52 %

9.5 ベンチマークの結果

9.5.1 コード全体の結果

Fig. 9.5, 9.6 および 9.7 にそれぞれ SP2, SR2201 および Paragon 上でプロセッサ数を 1 から 32 台まで変化させた場合の実行時間を示す. 粒子数は 100 万個, 200 万個および 400 万個 (それぞれ, case-1, case-2 および case-3 と呼ぶ) である.

全ての計算機に関する結果は同じ傾向, すなわち同程度の速度向上率を示している. コード全体の実行時間はプロセッサ台数 (N) とともに減少している. N が大きくなった時, 速度向上率が鈍る理由は, In と Op が逐次処理であり, PD と TS が N とともに増大するからである.

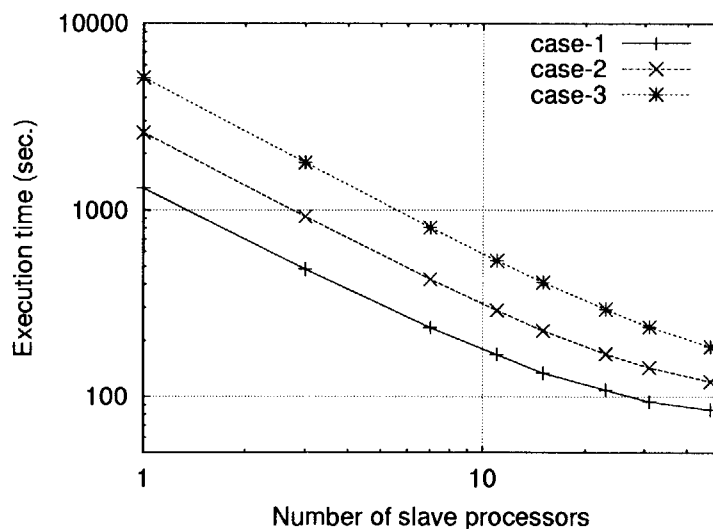


Fig. 9.5 SP2 上での BMT 測定結果

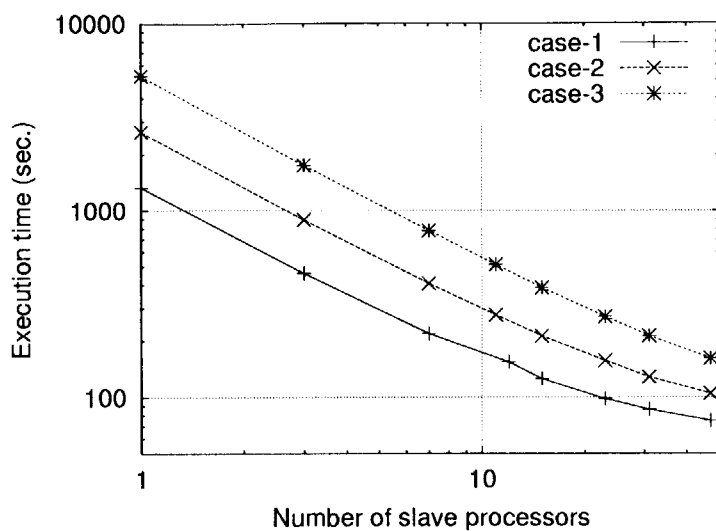


Fig. 9.6 SR2201 上でのBMT測定結果

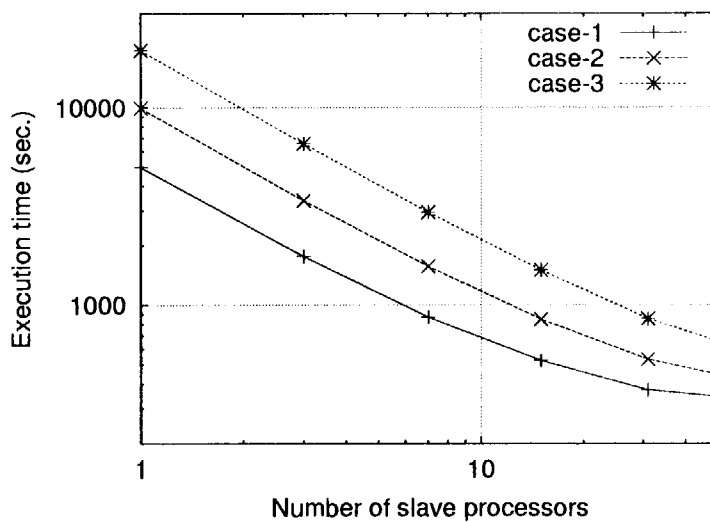


Fig. 9.7 Paragon 上でのBMT測定結果

9.5.2 処理別の性能

粒子数が 200 万個における SP2, SR2201 および Paragon の処理別の実行時間をグラフにしたものが、それぞれ Fig. 9.8, 9.9 および 9.10 である。個々の処理内容 (In, PD, PT, TS および Op) は臨界計算と同じである。

定性的な結果は、臨界計算の場合と同じである。個々の処理の状況は、以下のとおりである。

- In は逐次処理であり、プロセッサ台数に無関係で一定である。ただし、臨界計算と比較して、時間がかかっている。
- PD はプロセッサ台数に比例して増加している。臨界計算と比較して処理時間は少ない。
- PT はプロセッサ台数に反比例して、減少している。この傾向は臨界計算と同じである。
- TS はプロセッサ台数に比例して増加している。処理時間は臨界計算と同程度である。
- Op は逐次処理のため、プロセッサ台数に無関係で一定である。ただし、逐次処理の場合、余分な出力があるために、見かけ上、増加している。

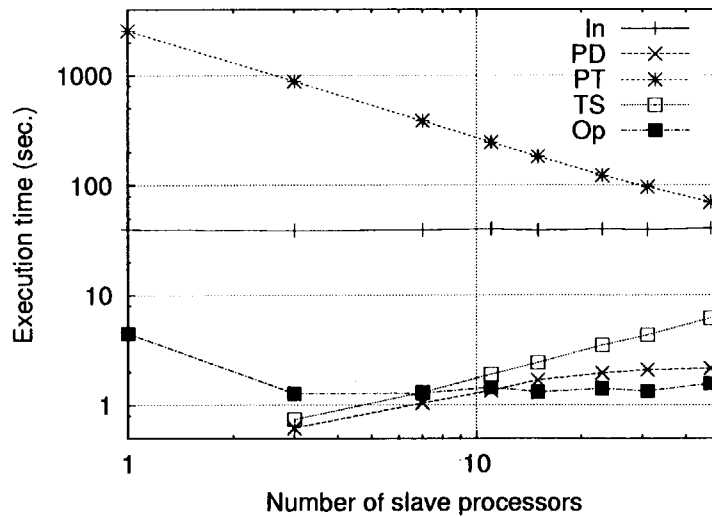


Fig. 9.8 SP2 上での処理別実行時間

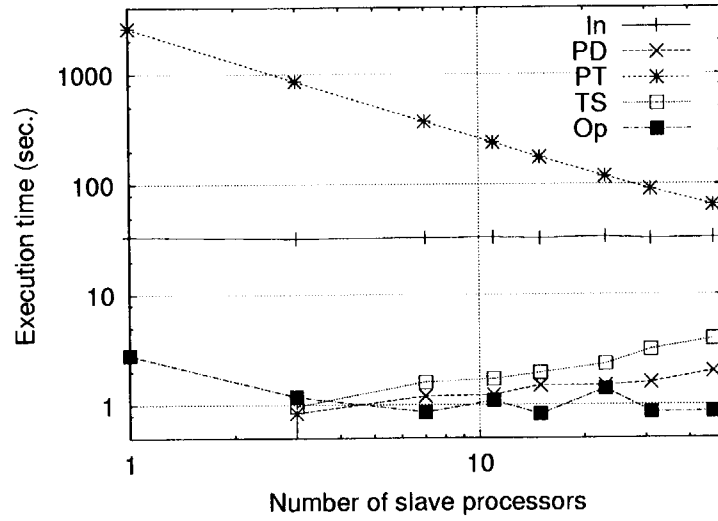


Fig. 9.9 SR2201 上での処理別実行時間

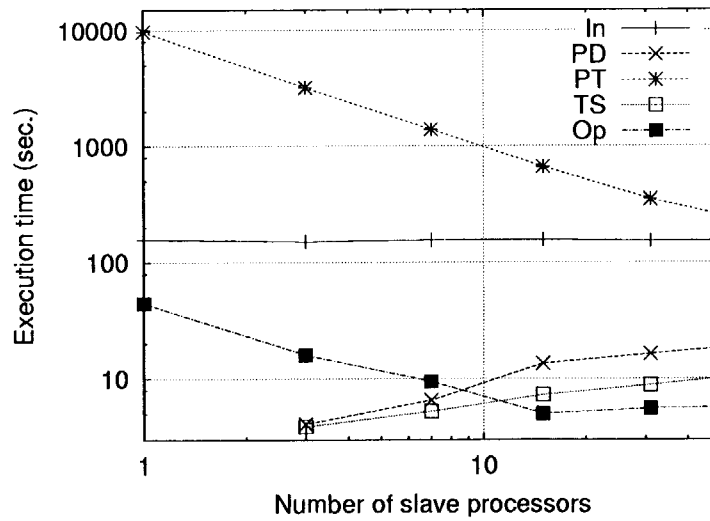


Fig. 9.10 Paragon 上での処理別実行時間

9.5.3 参考：富士通 AP3000 の結果

富士通 AP3000 は運用の関係で並列で使用可能なプロセッサ台数が少ない（最大4プロセッサ）が、動作確認をしたので測定結果を以下に示す。

コード全体の実行時間（Fig. 9.11 を参照のこと）は他の3種類の並列計算機よりも良い結果が得られている。これは、AP3000 の単体性能が一番良いことが原因と考えている。速度向上率（グラフの傾き）は他の並列計算機と同様の傾向を示している。

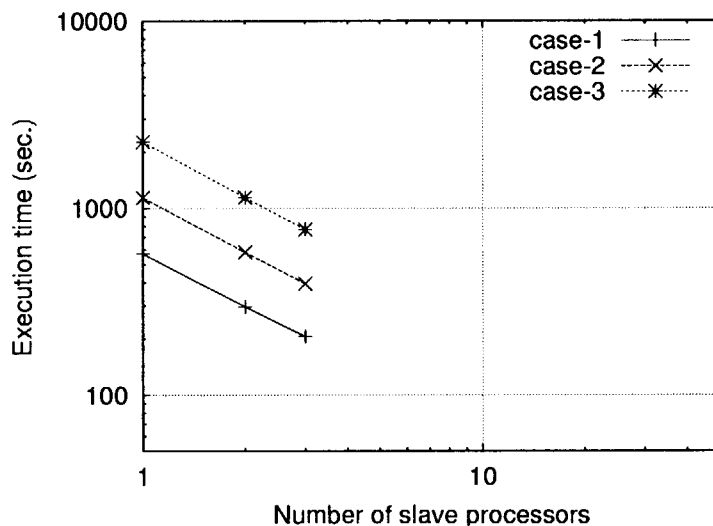


Fig. 9.11 AP3000 上での処理別実行時間

粒子数 200 万個における処理別の実行時間を Fig. 9.12 に示している。

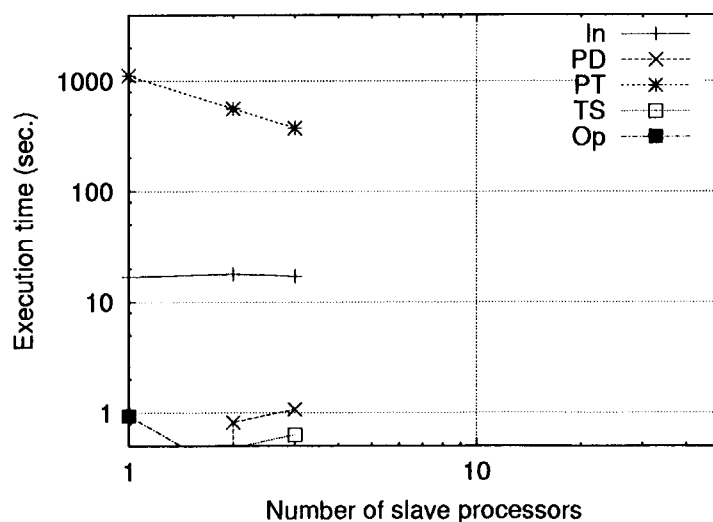


Fig. 9.12 AP3000 上での処理別実行時間

10. 性能評価モデルへの構築

10.1 ベンチマークテスト・システム

これまで、計算科学推進センターでは、並列 BMT コードおよび BMT 結果を蓄積・整備していく一環として、様々な性能を有する並列計算機の性能評価を実施するための性能評価方法を開発してきた。性能評価方法は、ユーザ・コード全体の性能を評価するレベル 1 ベンチマーク仕様とユーザ・コードのループおよび通信ライブラリ等の性能を評価するレベル 2 ベンチマーク仕様から構成されている。

レベル 1 ベンチマーク仕様を基に、プロセッサ台数の増加に比例して処理時間が減少するかという台数効果を評価したり、ある規模の問題を計算するのに要する時間を推定することが可能となる。この結果からコードに適した並列計算機を選択することができる。

10.2 レベル 2 ベンチマーク仕様の問題点

現行のレベル 2 ベンチマーク仕様では、(1) 各ループおよび通信ライブラリの実行時間を測定し、(2) 測定値と処理時間モデルからモデル係数を決定し、(3) 性能指標を各ループおよび通信ライブラリに対して求めることになっている。

現行の性能評価方法はプログラム中のループ処理単位にモデル化し、モデル化したループの処理時間を合成してプログラム全体の処理時間をモデル化している。この考え方は、原則としてループ単位の逐次実行処理時間が、四則演算実行回数に比例すること前提条件として仮定している。

これに対して、今回性能評価対象の MCNP4BK-BMT (および MCNP4BF-BMT) コードは、ある粒子の誕生から消滅までの一連のシミュレーションを単一プロセッサに割り当てる history based なアプローチをしているため (樋口ら [4])、以下の理由により現行の性能評価モデルをそのまま適用することは困難である。

- 四則演算を有するループが少ない。
粒子追跡を行なう 1 つのカーネルループから構成されており、コードがループ単位に細分化できることを前提としたレベル 2 の解析手法に不適である。
- 複雑な IF 文が多い。
- マスタ/スレーブ型の並列化手法を採用しているので、通信と演算のオーバーラップを考慮する必要がある。
- 動的な負荷分散を採用しているので、ジョブ毎に各プロセッサに分配される粒子数が異なる。

10.3 性能評価のための基礎データ

MCNP4BK-BMT コードのようなスカラー並列計算機向きの性能を評価するために必要となる基礎データは、以下のような数値を考えている。ここでは、ループ単位の処理時間でなく、一連の処理単位を基礎にモデル化することを前提として考えている。

- 処理別実行時間結果
- 負荷不均衡の割合（あるいは、並列化効率）

処理別実行時間を計測するにあたって、I/O 処理、通信処理、逐次処理のみの演算および並列処理可能な演算を明確にしておく。処理別実行時間に関して、今回は以下のような処理単位（詳細な内容は、7 章を参照のこと）に分割して、実行時間を測定している。

- (1) 初期化 (In: Initialization)
核断面積テーブル等のファイルを読み込む逐次処理。
- (2) 粒子分配 (PD: Particle distribution)
マスタ・プロセッサからスレーブ・プロセッサへのデータ通信処理（ブロードキャスト通信が主体）。
- (3) 粒子追跡と統計データの計算 (PT: Particle tracking)
スレーブ・プロセッサでの演算処理。
- (4) 統計データの集計 (TS: Totaling statistic data)
主として、スレーブ・プロセッサからマスタ・プロセッサへのデータ通信処理。
- (5) 統計データの出力 (Op: Output)
計算結果を出力する逐次処理。

上記の処理別に測定した実行時間を SP2 上で測定した case-3 の結果は、Fig. 10.1 のとおりである。

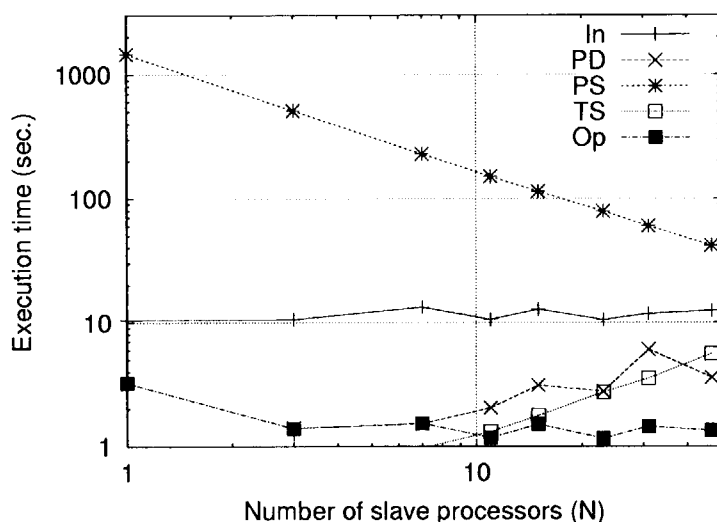


Fig. 10.1 SP2 上での処理別実行時間（臨界計算問題）

処理別の時間の中で PD, PT および TS 処理について（In と Op 処理はプロセッサ台数に依存しない逐次処理なので、対象から除外する）、プロセッサに対する関係を示したのが、Fig. 10.2, 10.3 および 10.4 である。ここで、case-4 は、1 世代あたりの粒子数が 1500 個のケースである。残りの 3 ケースは、7 章と同じである。

PD 処理の実行時間は、理論上 $\log_2(n)$ に比例して増加する。Fig. 10.2 によるとバラツキが大きい個所も存在するが、ここでは、理論式が満足することを前提に評価式を作成する。

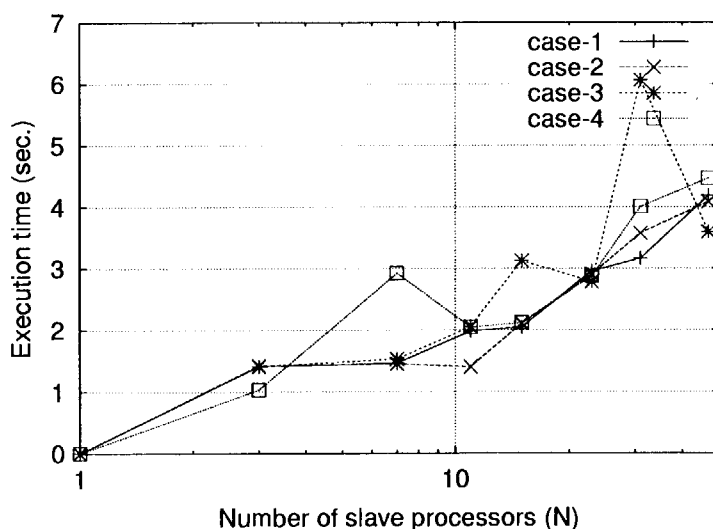


Fig. 10.2 SP2 上の PD 処理に対するプロセッサ台数との関係

PT 処理時間は, Fig. 10.3 のようように, どの粒子数の場合でも, プロセッサ台数の増加とともに減少する傾向がある. 第 1 次近似としては, 1 次関数で近似しても問題無いと考えられる.

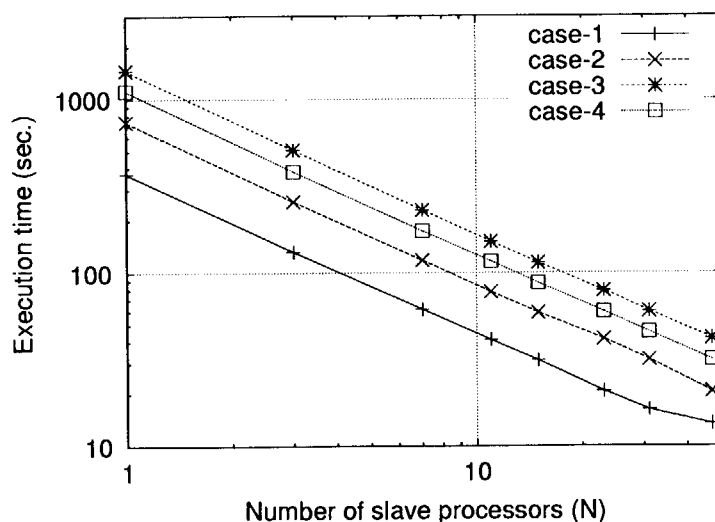


Fig. 10.3 SP2 上の PT 処理に対するプロセッサ台数との関係

TS 処理時間は, Fig. 10.4 のようにプロセッサ台数が増加するとともに増加する傾向がある. ソースから, 通信量はプロセッサに依存する部分と非依存の部分から構成されている. 両者を考慮して, 評価式を作成する. なお, 図中の一部のデータがバラついていて, これは測定誤差と考えている (再測定すると正常な値となっている).

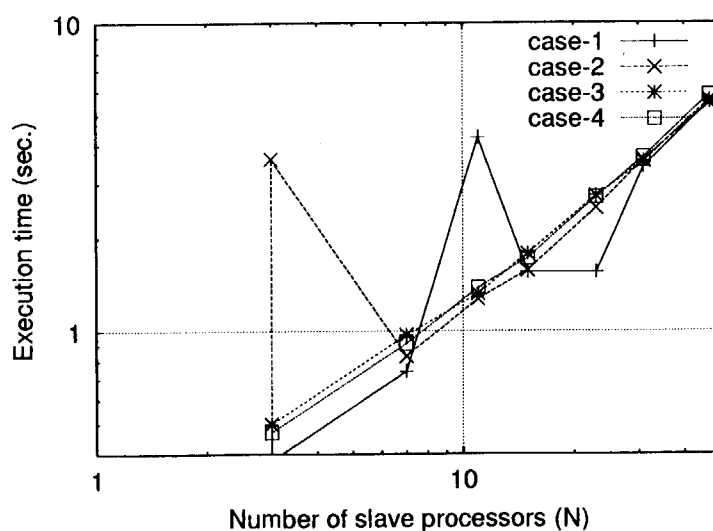


Fig. 10.4 SP2 上の TS 処理に対するプロセッサ台数との関係

10.4 性能評価モデルの開発

逐次処理における全実行時間 T_{tot}^s は、式 (10.1) で記述できる。

$$T_{tot}^s = T_{In}^s + T_{PT}^s + T_{Op}^s \quad (10.1)$$

ここで、 T_{In}^s, T_{PT}^s および T_{Op}^s は、それぞれ In, PT および Op の逐次処理実行時間である。

これに対し、並列処理における全実行時間 T_{tot}^p は、式 (10.2) で記述できる。

$$T_{tot}^p = T_{In}^s + T_{PD}^p + T_{PT}^p + T_{TS}^p + T_{Op}^s \quad (10.2)$$

ここで、 T_{PD}^p, T_{PT}^p および T_{TS}^p は、それぞれ PD, PT および TS の並列処理実行時間である。見積りの精度を高めるためには、演算と通信のオーバーラップを考慮する必要があるが、ここでは、オーバーラップ部分は通信時間とみなしている。この結果、演算部分の処理時間は過小評価している。

次に、並列処理に関連する処理時間 T_{PD}^p, T_{PT}^p および T_{TS}^p を見積る。

T_{PD}^p 時間は、主要な処理がブロードキャスト通信で、バイナリ・ツリー方式を採用しているので $\log_2 N_{pe}$ に比例すると仮定している。 T_{TS}^p 時間は、主要な処理が、スレーブ・プロセッサからマスタ・プロセッサへの通信処理のため、 N_{pe} に比例すると仮定する。 T_{PT}^p 時間は、最小二乗法で得られた 1 次式を採用する。なお、ここで、 N_{pe} はスレーブ・プロセッサ台数のことである。

以上のことから、式 (10.2) は、式 (10.3) のように記述できる。

$$T_{tot}^p = T_{In}^s + (a \cdot \log_2 N_{pe} + b) + (c \cdot \frac{1}{N_{pe}} + d) + (e \cdot N_{pe} + f) + T_{Op}^s \quad (10.3)$$

ここで、 a, b, c, d, e および f は、定数または問題規模（計算する総粒子数）の関数である。

10.5 SP2 の各処理時間のモデル係数の決定

まず最初に、SP2 に関する各処理のモデル係数を決定する。

PT 処理時間を 1 次式によるフィッティングの係数は、オリジナルのモデル係数決定方法を採用する。ただし、本コードは、DO ループ内の四則演算から算出して、演算量を見積ることは不可能なので、各並列計算機のピーク演算性能値に対する効率は考えないことにする。

なお、ここで算出する PT の処理時間モデルは、以下の前提条件を満足し、1 世代あたりの粒子数と世代数をパラメータで記述できるものと仮定する。それ以外のケースは、今後の課題とする。

- 変更可能な入力データは粒子数と世代数である。
- モデル構築に使用した結果から極端に逸脱した振舞いをしない。

すなわち、以下の手順で係数を決定する（詳細は、折居 [2] を参照のこと）。

- (1) 粒子数毎の逐次実行結果を基に、回帰分析を実施する。
- (2) 回帰分析結果から $(T_{PT}^s)_0$ を推定する。
- (3) 粒子数およびプロセッサ台数毎の並列結果を基に、回帰分析（最小二乗法）を実施する。
- (4) 回帰分析の結果から $(T_{PT}^p)_0$ および並列化効率 E_p を推定する。

逐次実行の回帰分析の結果 T_{PT}^s は、式 (10.4) のとおりである。

$$T_{PT}^s = 9.855 + 0.01457 \cdot N_P \cdot N_G \quad (10.4)$$

ここで、 N_G と N_P は世代数と 1 世代あたりの粒子数である。

次に、並列実行の回帰分析の結果 $(T_{PT}^p)_0$ および E_p は、式 (10.5) と (10.6) のとおりである。

$$(T_{PT}^p)_0 = 2.966 + 1.439 \cdot 10^{-4} \cdot N_P \cdot N_G \quad (10.5)$$

$$E_p = 1.0 \quad (10.6)$$

以上のことから、 T_{PT}^p のモデル式は、式 (10.7) となる。

$$T_{PT}^p = 2.966 + 1.439 \cdot 10^{-4} \cdot N_P \cdot N_G + \frac{9.855 + 0.01457 \cdot N_P \cdot N_G}{N_{pe}} \quad (10.7)$$

PT 処理における測定値と予測値とをプロットしたものが Fig. 10.5 であり、以下のことが明らかとなった。

- プロセッサ台数が 32 台程度までよく一致している。

- 1 世代あたりの粒子数が多い方がよく一致している。
- プロセッサ台数が 20 以上になると、推定曲線は実測値と比較して過大評価する傾向にある。

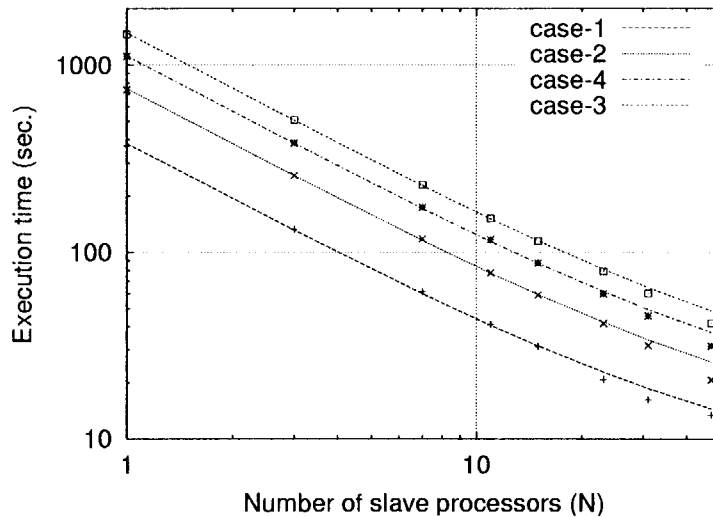


Fig. 10.5 PT 処理における測定値と予測値

PT 処理における相対誤差をプロットしたものが、Fig. 10.6 であり、以下のことが明らかとなった。

- case-1 以外の 3 ケースはほぼ同じ曲線を描いている。
- 1 次式による近似のズレから発生する系統的な誤差と考えられる。

測定値と予測値ズレは、プロセッサ台数が増加するとともに負荷不均衡の割合が上昇することが原因であると考えている。このことは、case-2 と case-3 の負荷不均衡の割合 F_{im} をプロットした Fig. 10.7 から確認している。また、この図から、1 プロセッサ内で計算する粒子数が少ない程、負荷不均衡の割合が高くなっていることも明かである。

1 世代あたりの粒子数を 4000 個にした場合の測定結果（最小二乗法で推定する際に使用していないデータ）と予測値をプロットしたのが、Fig. 10.8 である。

上図の結果は、外挿した予測値が実測値との間に極端な差がないことを示している。

次に PD 処理時間の見積りを行なう。

主要な PD の処理内容は、 $6 * msrk$ 個の倍精度実数型の変数をブロードキャストすることである。ここで、 $msrk$ は、RKPL 配列に格納する k_{eff} 値の個数であり、 $max(4500, 2.0 * N_P)$ で定義されている。case-1, case-2 および case-3 の場合、 $msrk=4500$ となるので、理論上 PD の処理時間 T_{PD}^p は、式 (10.8) と記述できる。

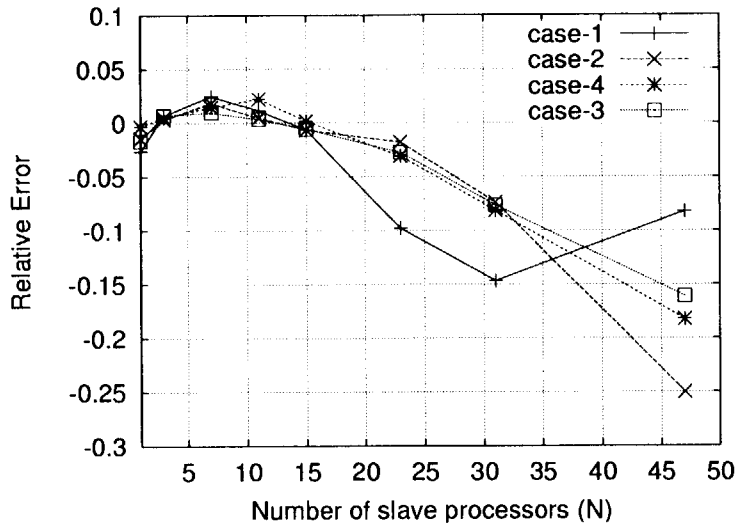


Fig. 10.6 PT 処理における相対誤差 (対プロセッサ台数)

$$T_{PD}^p = \frac{6 \cdot 4500 \cdot 8 \cdot \log_2 N_{pe} \cdot N_G}{40 \cdot 10^6 \cdot E_c} \quad (10.8)$$

ここで、 $40 \cdot 10^6$ および E_c は、SP2 のネットワーク転送速度 (40 MB/sec) と通信効率である。

Fig. 10.9 に PD の実測値と予測曲線 (式 (10.8) に 4 種類の通信効率を適用したものをプロットしている。この図から通信効率 E_c は 0.4 が適していると判断している。

最後に、TS 処理のモデル係数を決定方法について述べる。

データ転送量は、以下のとおりである。

$$C_{TS} = (ltskcm + ltskpt + 42) \cdot 4 \cdot N_c + (ntskcm + 9) \cdot 8 \cdot N_c + 6 \cdot 8 \cdot N_P \cdot N_G \quad (10.9)$$

ここで、 $ltskcm$ 、 $ltskpt$ および $ntskcm$ は、それぞれ、コモン領域内の変数の個数であり、今回のデータでは、2244、39 および 1898 となっている。 N_c は、データの転送回数を示している。

処理時間は、1 回のデータ転送量およびデータ転送回数に関係している。しかしながら、静的と動的な負荷分散手法を採用しているので、実際のデータ転送回数をプロセッサ毎に見積ることは不可能である。そこで、今回は、全データ転送量のみ考慮する。静的と動的な負荷分散手法を交互に採用し、動的な場合における分散係数を 5 と仮定すると、 N_c は、式 (10.10) と記述できる。

$$\begin{aligned} N_c &= (N_{pe} + 5 \cdot N_{pe}) \cdot \frac{N_G}{2} \\ &= 3 \cdot N_{pe} \cdot N_G \end{aligned} \quad (10.10)$$

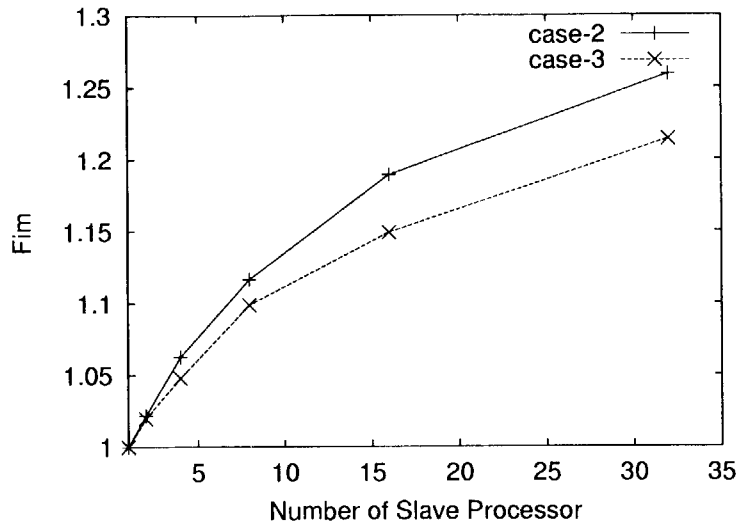


Fig. 10.7 SP2 上での負荷不均衡の割合

したがって、TS 処理時間 T_{TS}^p は、式 (10.11) となる。

$$T_{TS}^p = \frac{(25056 \cdot 3 \cdot N_{pe} + 48 \cdot N_P) \cdot N_G}{4.0 \cdot 10^7 \cdot E_c} \quad (10.11)$$

Fig. 10.10 に TS の実測値と予測曲線 (式 (10.11) に 4 種類の通信効率を適用したもの) をプロットしている。なお、粒子数として 2000 の値を使用している。この図から通信効率 E_c は 0.9 が適していると判断している。

以上のことから、SP2 における臨界計算の処理時間モデルは、式 (10.13) となる。

$$T_{tot}^p = 2.966 + 1.439 \cdot 10^{-4} \cdot N_P \cdot N_G + \frac{9.855 + 0.01457 \cdot N_P \cdot N_G}{N_{pe}} \quad (10.12)$$

$$+ \frac{2.16 \cdot \log_2 N_{pe} \cdot N_G}{1.6 \cdot 10^2} + \frac{(75168 \cdot N_{pe} + 48 \cdot N_P) \cdot N_G}{3.6 \cdot 10^8} + 14.0$$

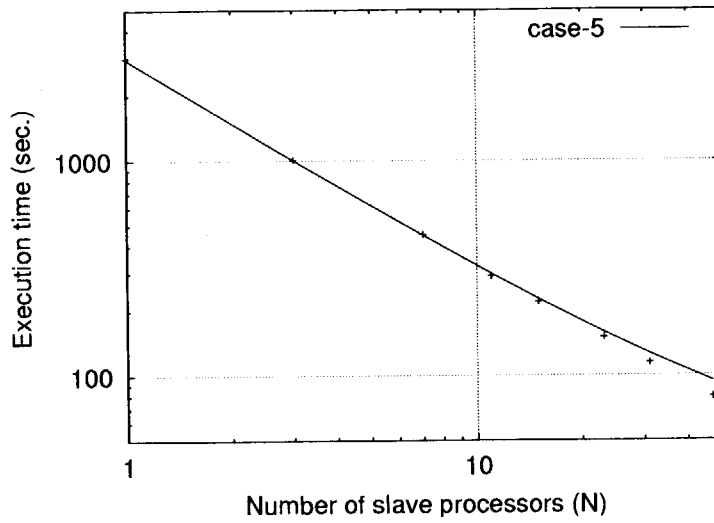


Fig. 10.8 PT 処理における推定値と実測値（外挿結果）

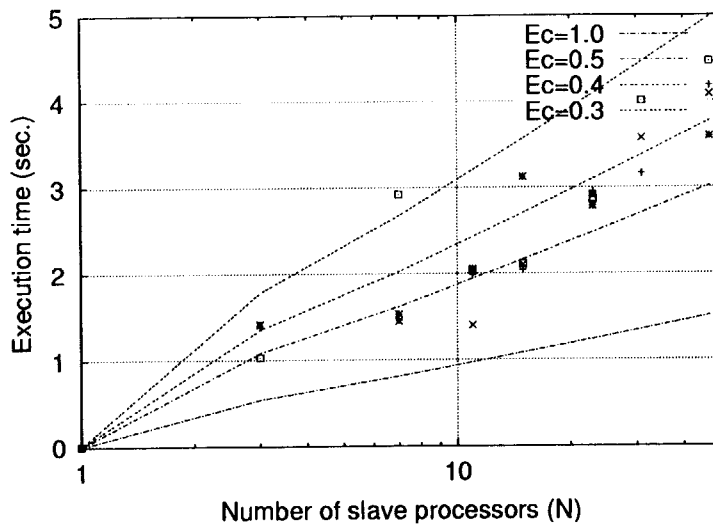


Fig. 10.9 SP2 上での PD の実測値と通信効率

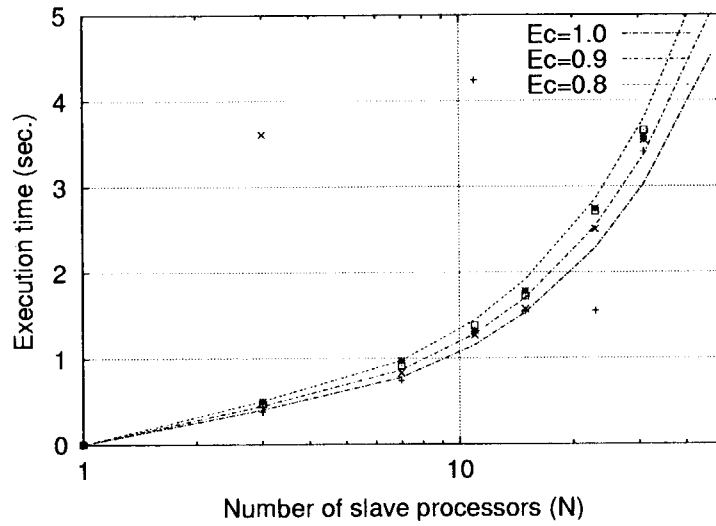


Fig. 10.10 SP2 上での TS の実測値と通信効率

10.6 SR2201 の各処理時間のモデル係数の決定

次に、SR2201 について SP2 と同様のモデル係数を決定し、両者のモデル係数を比較した。

処理別に測定した実行時間を SR2201 上で測定した case-3 の結果は、Fig. 10.11 のとおりである。

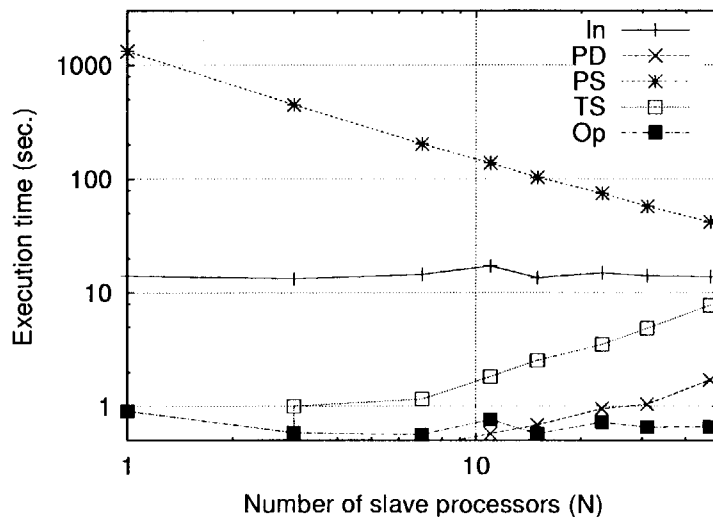


Fig. 10.11 SR2201 上での処理別実行時間（臨界計算問題）

処理別の時間の中で PD, PT および TS 処理について、プロセッサに対する関係を示したのが、Fig. 10.12, 10.13 および 10.14 である。

PD 処理の実行時間は、SP2 と同様に理論上 $\log_2(n)$ に比例して増加することを前提に、評価式を作成する (Fig. 10.12 を参照のこと)。

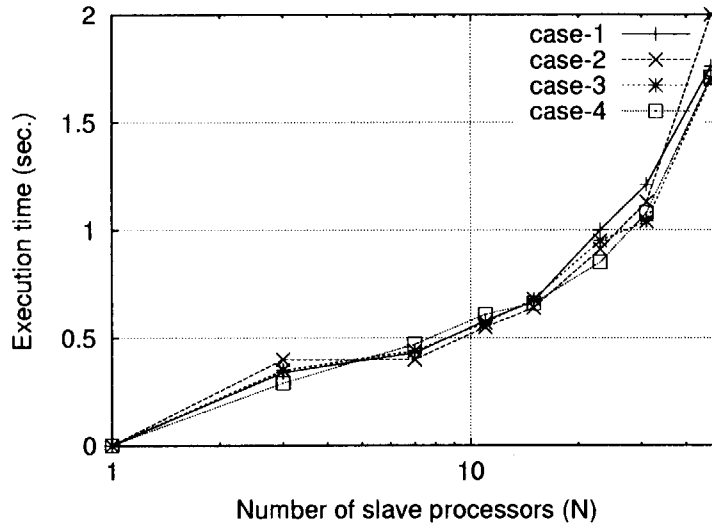


Fig. 10.12 SR2201 上の PD 処理に対するプロセッサ台数との関係

PT 処理時間は、Fig. 10.13 のように、どの粒子数の場合でも、プロセッサ台数の増加とともに減少する傾向がある。第 1 次近似としては、1 次関数で近似しても問題無いと判断している。

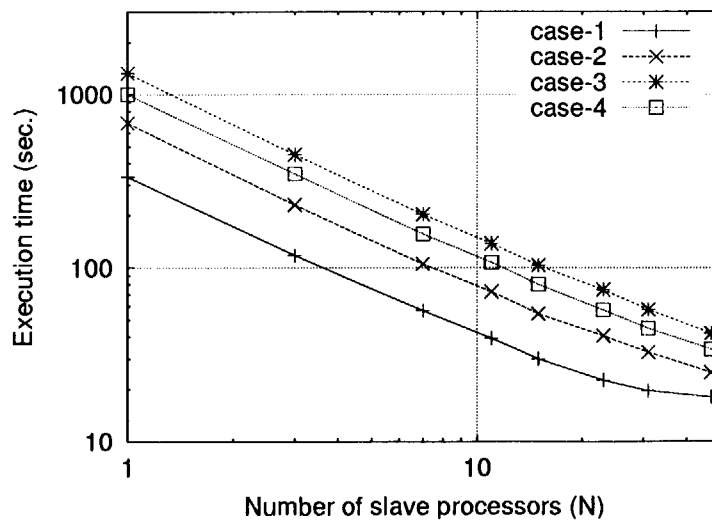


Fig. 10.13 SR2201 上の PT 処理に対するプロセッサ台数との関係

TS 処理時間は、Fig. 10.14 のようにプロセッサ台数が増加するとも増加する傾向がある。ソースから、通信量はプロセッサに依存する部分と非依存の部分から構成されている。両者を考慮して、評価式を作成する。

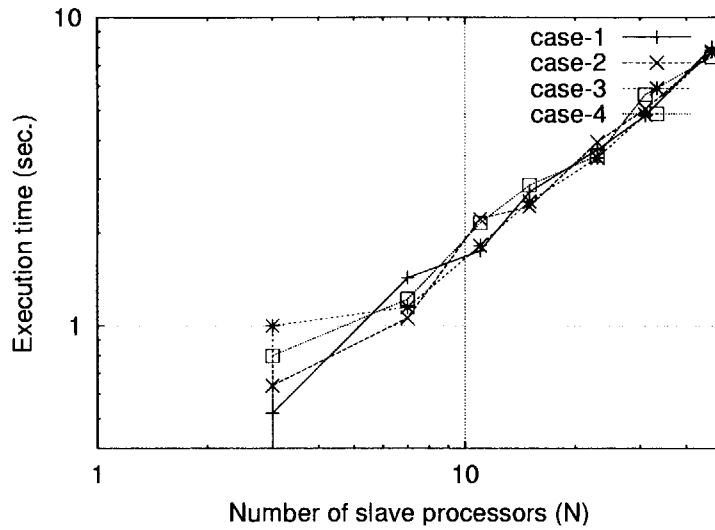


Fig. 10.14 SR2201 上の TS 処理に対するプロセッサ台数との関係

逐次実行の回帰分析の結果 T_{PT}^s は、式 (10.13) のとおりである。

$$T_{PT}^s = 13.755 + 0.01315 \cdot N_P \cdot N_G \quad (10.13)$$

ここで、 N_G と N_P は世代数と 1 世代あたりの粒子数である。

次に、並列実行の回帰分析の結果 $(T_{PT}^p)_0$ および E_c は、式 (10.14) および (10.15) のとおりである。

$$(T_{PT}^p)_0 = 6.555 + 0.9651 \cdot 10^{-4} \cdot N_P \cdot N_G \quad (10.14)$$

$$E_c = 1.0 \quad (10.15)$$

以上のことから、 T_{PT}^p のモデル式は、式 (10.16) となる。

$$T_{PT}^p = 6.555 + 0.9651 \cdot 10^{-4} \cdot N_P \cdot N_G + \frac{13.755 + 0.01315 \cdot N_P \cdot N_G}{N_{pe}} \quad (10.16)$$

PT 処理における測定値と予測値とをプロットしたものが Fig. 10.15 であり、以下のことが明かとなった。

- プロセッサ台数が 48 台程度まで非常によく一致している。
- 1 世代あたりの粒子数が多いほどよく一致している。
- SP2 と比較して、実測値を予測値がよく一致している。

PT 処理における相対誤差をプロットしたものが、Fig. 10.16 であり、以下のことが明かとなった。

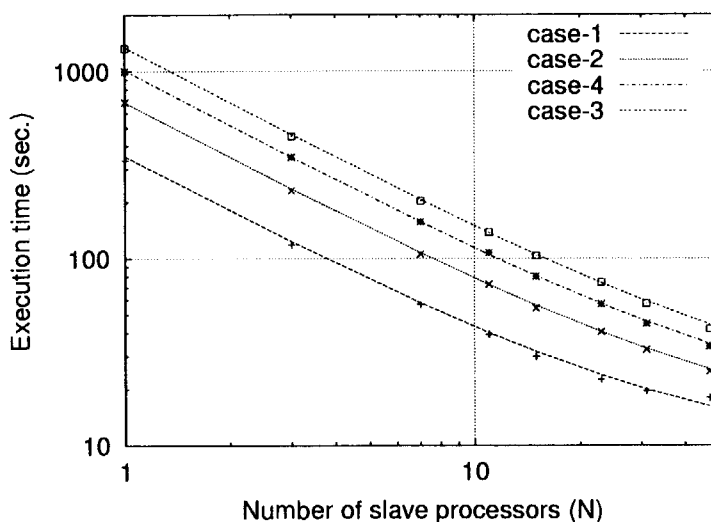


Fig. 10.15 PT 処理における測定値と予測値 (SR2201)

- case-1 以外の 3 ケースはほぼ同じ曲線を描いている。
- 1 次式による近似のズレから発生する系統的な誤差は少ない。

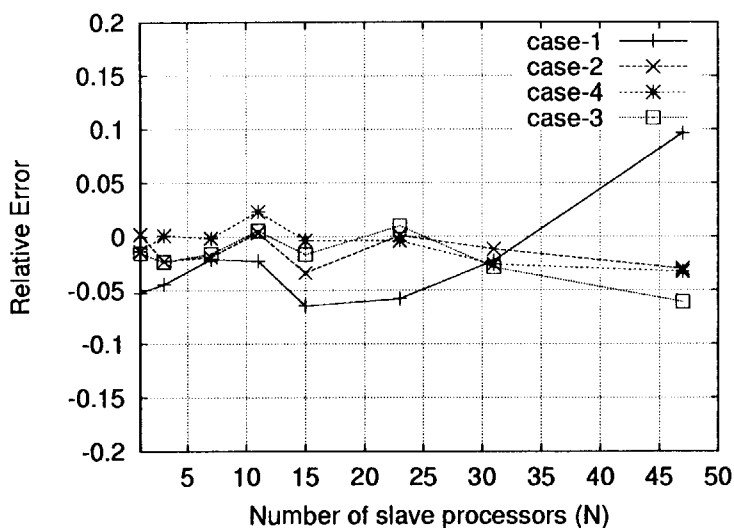


Fig. 10.16 PT 処理における相対誤差 (対プロセッサ台数)

SP2 と同様に SR2201 における理論上 PD の処理時間 T_{PD}^p は、式 (10.17) と記述できる。

$$T_{PD}^p = \frac{6 \cdot 4500 \cdot 8 \cdot \log_2 N_{pe} \cdot N_G}{200 \cdot 10^6 \cdot E_c} \quad (10.17)$$

ここで、 $200 \cdot 10^6$ は、SR2201 のネットワーク転送速度 (200 MB/sec) である。

Fig. 10.17 に PD の実測値と予測曲線 (式 (10.17) に 5 種類の通信効率を適用したもの) をプロットしている. この図から通信効率 E_c は 0.25 が適していると判断している. SP2 と比較して, プロセッサ台数が 32 台以上になると実測値と予測値とのズレが大きい.

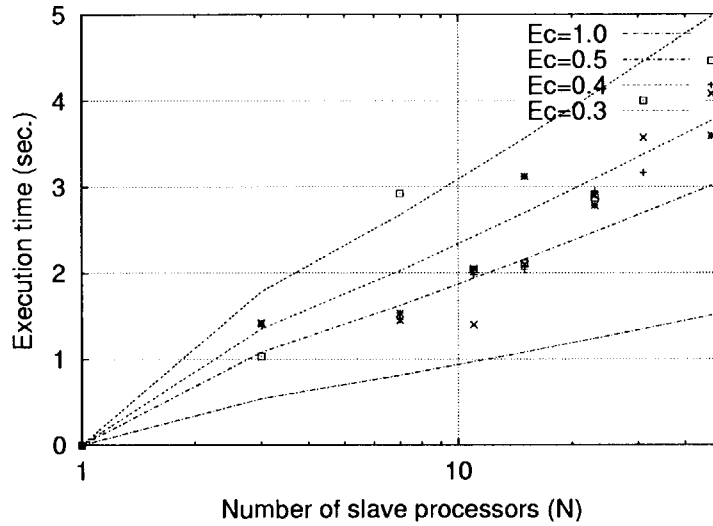


Fig. 10.17 SR2201 上での PD の実測値と通信効率

最後に, TS 処理のモデル係数を決定方法について述べる.

データ転送量は, SP2 と同じである. したがって, TS 処理時間 T_{TS}^p は, 式 (10.18) となる.

$$T_{TS}^p = \frac{(25056 \cdot 3 \cdot N_{pe} + 48 \cdot N_P) \cdot N_G}{2.0 \cdot 10^8 \cdot E_c} \quad (10.18)$$

Fig. 10.18 に TS の実測値と予測曲線 (式 (10.18) に 5 種類の通信効率を適用したもの) をプロットしている. この図から通信効率 E_c は 0.5 が適していると判断している.

以上のことから, SR2201 における臨界計算の処理時間モデルは, 式 (10.20) となる.

$$T_{tot}^p = 6.555 + 0.9651 \cdot 10^{-4} \cdot N_P \cdot N_G + \frac{13.755 + 0.01315 \cdot N_P \cdot N_G}{N_{pe}} + 0.432 \cdot 10^{-2} \cdot \log_2 N_{pe} \cdot N_G + (75168 \cdot N_{pe} + 48 \cdot N_P) \cdot N_G \cdot 10^{-8} + 14.0 \quad (10.19)$$

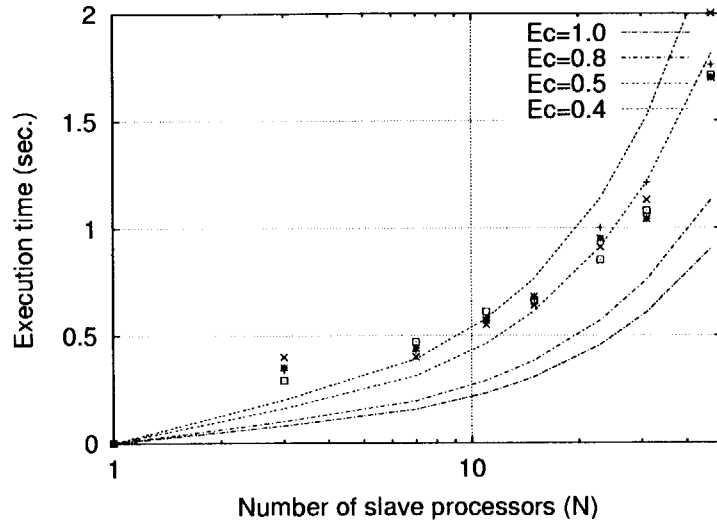


Fig. 10.18 SR2201 上での TS の実測値と通信効率

11. 固定源問題の性能評価モデル

臨界計算問題と同様に固定源問題の性能評価モデルのモデル係数を決定する。基本となるモデル式は、臨界計算問題と同じである。

11.1 SP2 の各処理時間のモデル係数の決定

まず最初に、SP2 に関する各処理のモデル係数を決定する。

逐次実行の回帰分析の結果 T_{PT}^s は、式 (11.1) のとおりである。

$$T_{PT}^s = 5.0800 + 0.0012685 \cdot N_P \quad (11.1)$$

ここで、 N_P は粒子数である。

次に、並列実行の回帰分析の結果 $(T_{PT}^p)_0$ および E_p は、式 (11.2) および (11.3) のとおりである。

$$(T_{PT}^p)_0 = 2.6743 + 7.0089 \cdot 10^{-6} \cdot N_P \quad (11.2)$$

$$E_p = 1.0 \quad (11.3)$$

以上のことから、 T_{PT}^p のモデル式は、式 (11.4) となる。

$$T_{PT}^p = 2.6743 + 7.0089 \cdot 10^{-6} \cdot N_P + \frac{5.0800 + 1.2685 \cdot 10^{-3} \cdot N_P}{N_{pe}} \quad (11.4)$$

PT 処理における測定値と予測値とをプロットしたものが Fig. 11.1 であり、以下のことが明らかとなった。

- プロセッサ台数が 48 台程度までよく一致している。
- 粒子数の多寡に関係無く、よく一致している。

PT 処理における相対誤差をプロットしたものが、Fig. 11.2 であり、以下のことが明らかとなった。

- case-3 以外の 3 ケースはほぼ同じ曲線を描いている。
- 1 次式による近似のズレから発生する系統的な誤差と考えられる。

主要な PD の処理内容は、必要な変数をブロードキャストすることである。PD の処理時間 T_{PD}^p は、式 (11.5) と記述できる。

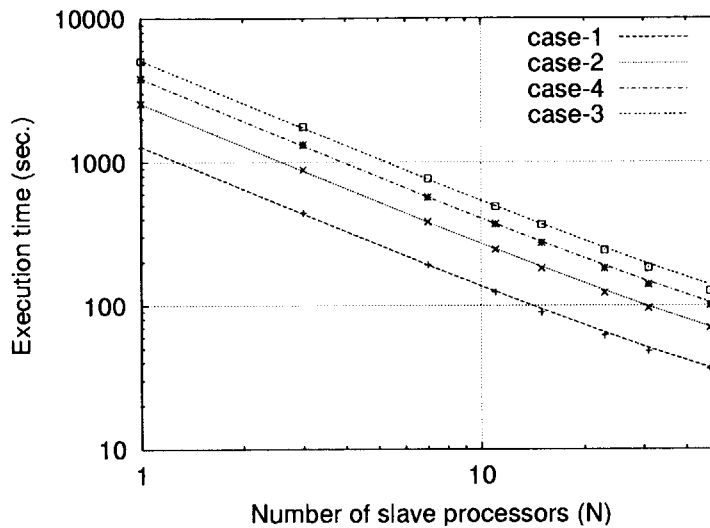


Fig. 11.1 PT 処理における測定値と予測値

$$T_{PD}^p = \frac{(1385 + (N_G + 656) \cdot \log_2 N_{pe}) \cdot 4 + (1563 + 453 \cdot \log_2 N_{pe}) \cdot 8}{40 \cdot 10^6 \cdot E_c} \quad (11.5)$$

ここで、 $40 \cdot 10^6$ および E_c は、SP2 のネットワーク転送速度 (40 MB/sec) と通信効率である。

Fig. 11.3 に PD の実測値と予測曲線 (式 (11.5) に 4 種類の通信効率を適用したもの) をプロットしている。この図から通信効率 E_c は 0.01 程度と判断している。

最後に、TS 処理のモデル係数を決定方法について述べる。

データ転送量は、以下のとおりである。

$$C_{TS} = (ltskcm + ltskpt + 42) \cdot 4 \cdot N_c + (ntskcm + 9) \cdot 8 \cdot N_c + 265880 \cdot N_e \quad (11.6)$$

ここで、 $ltskcm$ 、 $ltskpt$ および $ntskcm$ は、それぞれ、コモン領域内の変数の個数であり、今回のデータでは、2244、39 および 1898 となっている。 N_c は、データ転送回数である。

処理時間は、データ転送量とデータ転送回数に依存する。しかしながら、武宮の動的な負荷分散手法を採用しているので、実際のデータ転送回数をプロセッサ毎に見積ることは困難である。そこで、今回は、静的と動的な負荷分散手法を交互に採用し、動的な分散係数が 5 と仮定している (データ転送回数を過小評価していることになる)。したがって、 N_c は、式 (11.7) となる。

$$\begin{aligned} N_c &= (N_{pe} + 5 \cdot N_{pe}) \cdot \frac{N_{prdmp}}{2} \\ &= 3 \cdot N_{pe} \cdot N_{prdmp} \end{aligned} \quad (11.7)$$

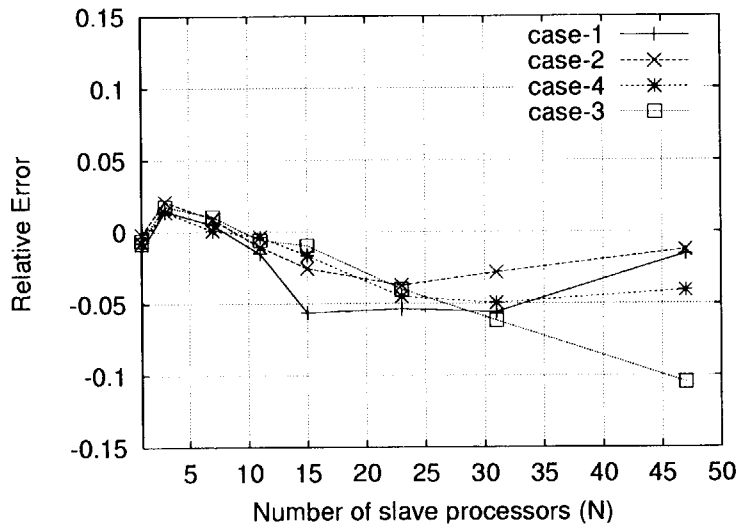


Fig. 11.2 PT 処理における相対誤差 (対プロセッサ台数)

以上のことから TS 処理時間 T_{TS}^p は、式 (11.8) となる。

$$T_{TS}^p = \frac{(25056 \cdot 3 + 265880) \cdot N_{pe} \cdot N_{prdmp}}{4.0 \cdot 10^7 \cdot E_c} \quad (11.8)$$

Fig. 11.4 に TS の実測値と予測曲線 (式 (11.8) に 4 種類の通信効率を適用したもの) をプロットしている。なお、粒子数として 400 百万個の値を使用している。この図から通信効率 E_c は 0.5 が適していると判断している。なお、 N_{prdmp} の値は 10 (デフォルトの値) である。

以上のことから、SP2 における固定源問題の処理時間モデル式は、式 (11.9) となる。

$$T_{tot}^p = 2.6743 + 7.0089 \cdot 10^{-6} \cdot N_P + \frac{5.0800 + 1.2685 \cdot 10^{-3} \cdot N_P}{N_{pe}} + (4511 + (N_G + 1562) \cdot \log_2 N_{pe}) \cdot 10^{-5} + 0.0170524 \cdot N_{pe} \cdot N_{prdmp} + 40.0 \quad (11.9)$$

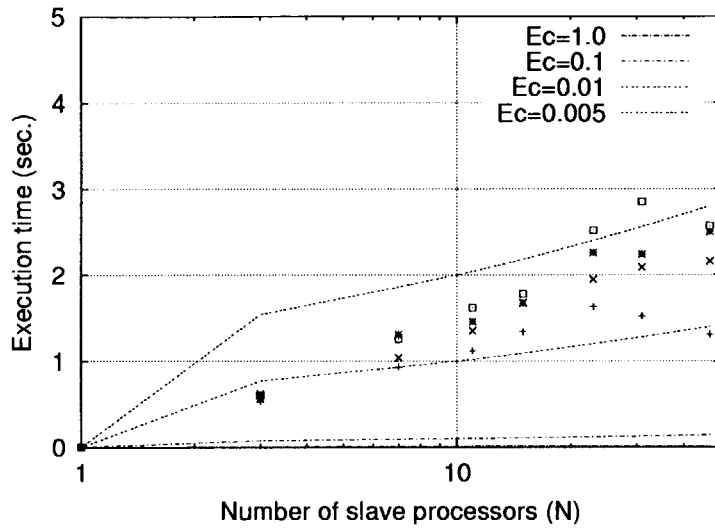


Fig. 11.3 SP2 上での PD の実測値と通信効率

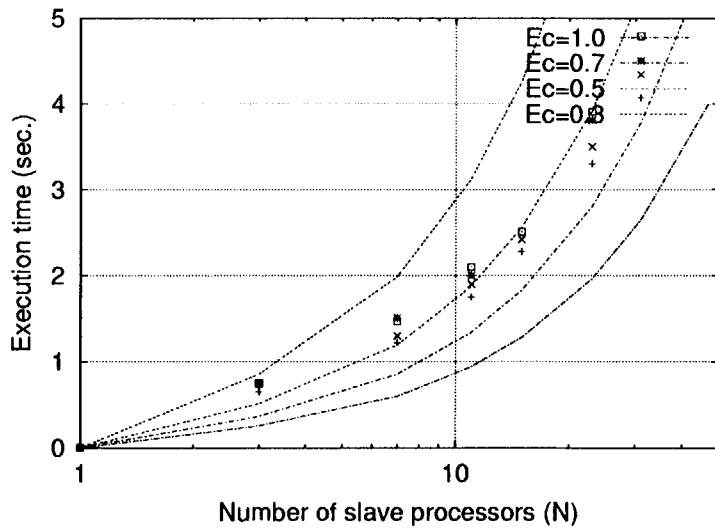


Fig. 11.4 SP2 上での TS の実測値と通信効率

11.2 SR2201 の各処理時間のモデル係数の決定

次に、SR2201 に関する各処理のモデル係数を決定する。

逐次実行の回帰分析の結果 T_{PT}^s は、式 (11.10) のとおりである。

$$T_{PT}^s = -23.435 + 0.0013146 \cdot N_P \quad (11.10)$$

ここで、 N_P は粒子数である。

次に、並列実行の回帰分析の結果 $(T_{PT}^p)_0$ および E_p は、式 (11.11) および (11.12) のとおりである。

$$(T_{PT}^p)_0 = 6.0565 - 1.8182 \cdot 10^{-6} \cdot N_P \quad (11.11)$$

$$E_p = 1.0 \quad (11.12)$$

ここで、問題は、式 (11.11) で係数が負になることである。並列化効率が 100 % 以上となっていることから判断して、原因は通信と演算のオーバーラップを考慮せずに、PT 処理時間を測定したことであると推定している。すなわち、PT 処理時間を過小評価したため、並列実行結果を 1 次式で回帰分析すると粒子数が多いケースが急傾斜となり、外挿した切片の値が粒子数が増大するとともに小さい値を取るようになった。このことは、例え問題のある測定結果を利用しているとしても（データ測定範囲内では問題無い結果は得られている）、外挿することにより値を決定する本方式の限界であると考えている（改善案は今のところ検討していない）。

以上のことから、 T_{PT}^p のモデル式は、式 (11.13) となる。

$$T_{PT}^p = 6.0565 - 1.8182 \cdot 10^{-6} \cdot N_P + \frac{-23.435 + 1.3146 \cdot 10^{-3} \cdot N_P}{N_{pe}} \quad (11.13)$$

PT 処理における測定値と予測値とをプロットしたものが Fig. 11.5 であり、以下のことが明かとなった。

- プロセッサ台数が 32 台程度までよく一致している。
- プロセッサ台数が 48 台の結果はあまり一致していない。
- 粒子数の多寡に関係無く、よく一致している。

PT 処理における相対誤差をプロットしたものが、Fig. 11.6 であり、以下のことが明かとなった。

- 4 ケースともほぼ同じ曲線を描いている。
- 1 次式による近似のズレから発生する系統的な誤差と考えられる。

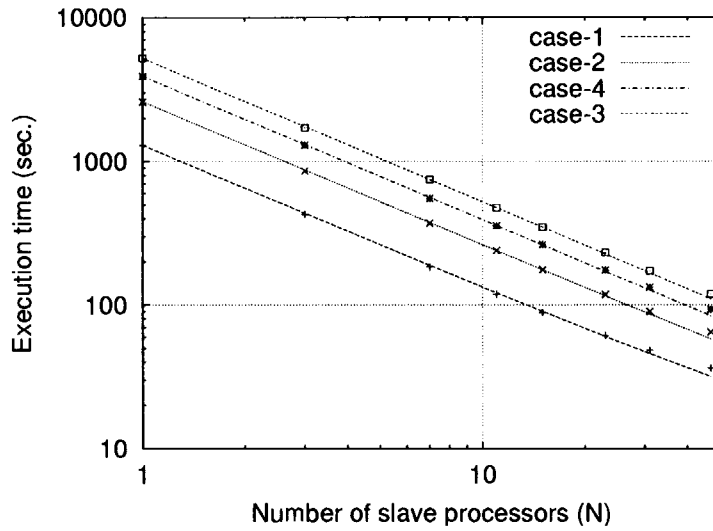


Fig. 11.5 PT 処理における測定値と予測値

主要な PD の処理内容は、必要な変数をブロードキャストすることである。PD の処理時間 T_{PD}^p は、式 (11.14) と記述できる。

$$T_{PD}^p = \frac{(1385 + (N_G + 656) \cdot \log_2 N_{pe}) \cdot 4 + (1563 + 453 \cdot \log_2 N_{pe}) \cdot 8}{200 \cdot 10^6 \cdot E_c} \quad (11.14)$$

ここで、 $200 \cdot 10^6$ および E_c は、SR2201 のネットワーク転送速度 (200 MB/sec) と通信効率である。

Fig. 11.7 に PD の実測値と予測曲線 (式 (11.14) に 4 種類の通信効率を適用したもの) をプロットしている。この図から通信効率 E_c は 0.01 程度と判断している。

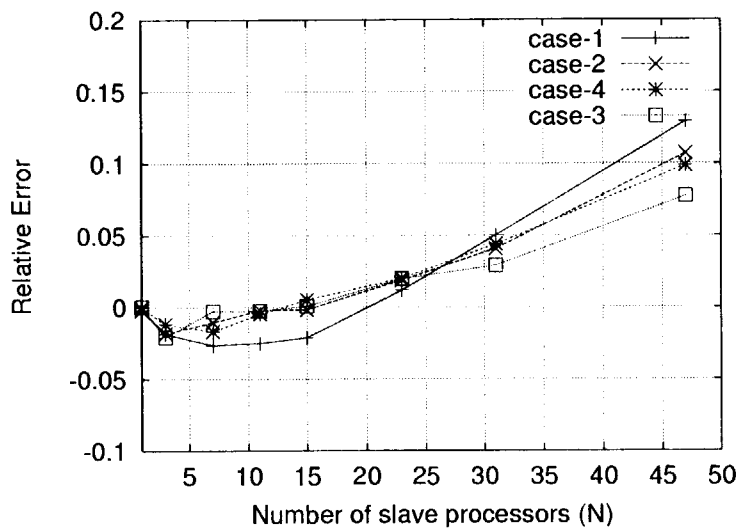


Fig. 11.6 PT 処理における相対誤差 (対プロセッサ台数)

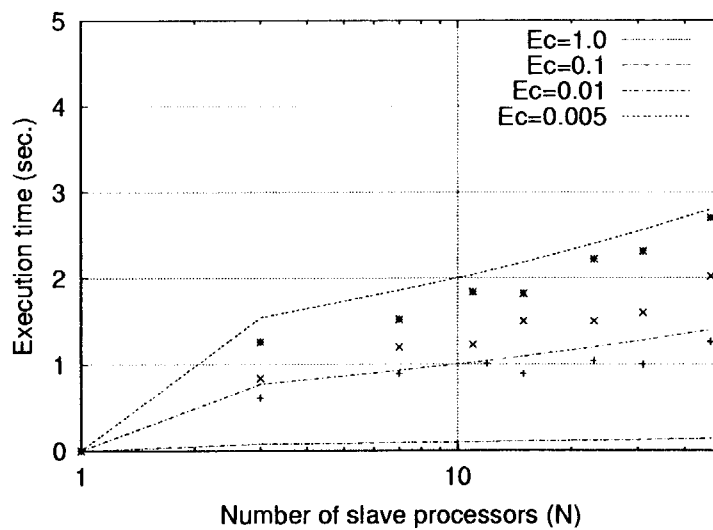


Fig. 11.7 SR2201 上での PD の実測値と通信効率

最後に、TS 処理のモデル係数を決定方法について述べる。

データ転送量は SP2 と同じなので、TS 処理時間 T_{TS}^p は、式 (11.8) となる。

$$T_{TS}^p = \frac{(25056 \cdot 3 + 265880) \cdot N_{pe} \cdot N_{prdmp}}{200.0 \cdot 10^6 \cdot E_c} \quad (11.15)$$

Fig. 11.8 に PD の実測値と予測曲線 (式 (11.15) に 4 種類の通信効率を適用したもの) をプロットしている。なお、粒子数として 400 百万個の値を使用している。この図から通信効率 E_c は 0.7 が適していると判断している。なお、 N_{prdmp} の値は 10 (デフォルトの値) である。

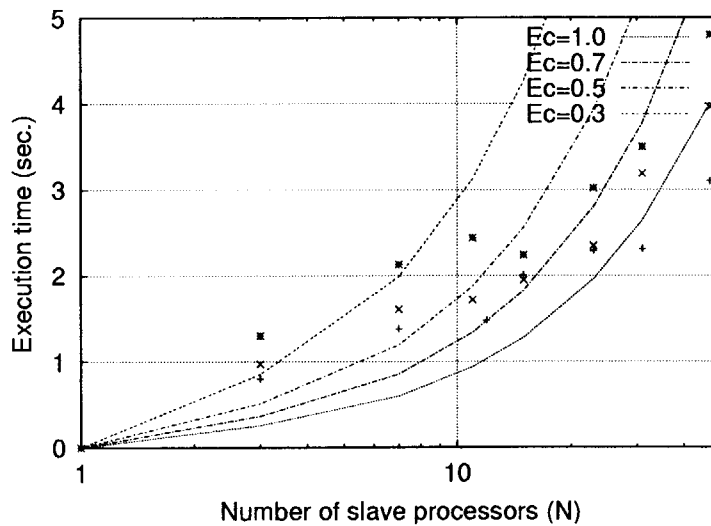


Fig. 11.8 SR2201 上での TS の実測値と通信効率

以上のことから、SR2201 における固定源問題の処理時間モデル式は、式 (11.16) となる。

$$T_{PT}^p = 6.0565 - 1.8182 \cdot 10^{-6} \cdot N_P + \frac{-23.435 + 1.3146 \cdot 10^{-3} \cdot N_P}{N_{pe}} + (4511 + (N_G + 1562) \cdot \log_2 N_{pe}) \cdot 2.0 \cdot 10^{-6} + 2.4361 \cdot 10^3 \cdot N_{pe} \cdot N_{prdmp} + 35.0 \quad (11.16)$$

12. 留意事項

12.1 SR2201 が遅い原因

並列BMTコードを開発中に、SR2201 (Paragon も) の測定結果が 8 プロセッサ以上になると性能が低下する現象が発生した (Fig.12.1 を参照のこと)。その原因が判明し SP2 と同等の性能が得られたので以下にその内容を留意事項として記載する。

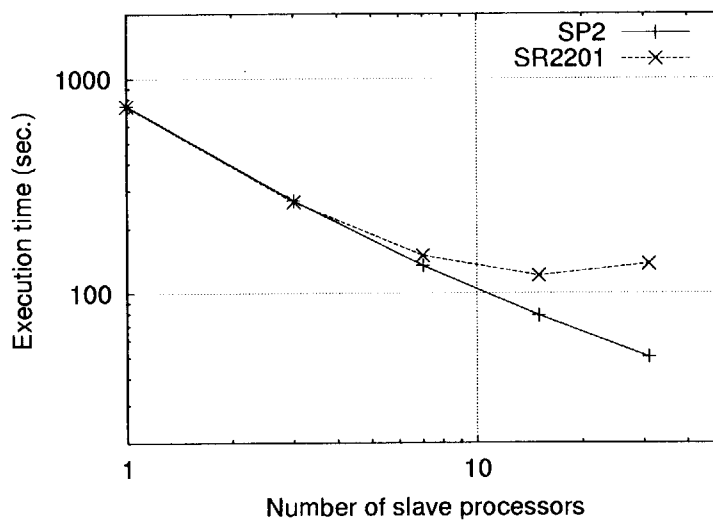


Fig. 12.1 SP2 と SR2201 の測定結果の比較 (ソース修正前)

SR2201 が性能を発揮できないのは、データの受信関連処理で時間がかかっていることが原因であった。マスタ・プロセッサの受信処理は、Fig.12.2 中の `mrecv(33,nt)` と `mrecv(34,-1)` で行なっている。両者の処理で問題となり、受信に時間がかかっている処理は、`mrecv(34,-1)` である。本処理は、単に全スレーブからデータを順次受信するだけの処理であり、特に問題となるような処理を含んでいない (ようにプログラミングしている)。

次にスレーブ・プロセッサでの処理 (ソース) を Fig.12.3 に示す。マスタ・プロセッサでの `mrecv(33,nt)` と `mrecv(34,-1)` に対応するスレーブ・プロセッサでの処理は、それぞれ、`msend(0,33,i)` と `msend(0,34,i)` である。最初、スレーブ・プロセッサからマスタ・プロセッサへの送信時間を測定したが、実行時間は短かく、データ転送量から算出したデータ転送速度は正常であった。次に、`do 840` ループの実行時間を測定した。この結果、本処理で非常に実行時間がかかっていることが判明した。問題の処理は、`read(iuo,..)` の入力処理である。これは粒子追跡処理における途中経過の情報をファイル経由で処理している。本データで出力している内容は重要でないと判断し、BMT コードでは本処理に関連した I/O 処理を全て削除した。

```

do 510 nt=1,ntasks
c
c   receive data from slave processor (1)
c
c   call mrecv(33,nt)
c       :
c       :
510 continue
c
c   receive data from slave processor (2)
c
c   call mrecv(34,-1)
c
c   do 580 mt=1,nm
c       nh=nc(mt,2)
c       ipos=loc(1,nh)
c       dpos=loc(2,nh)
c       rpos=loc(3,nh)
c       cpos=loc(4,nh)
c       nc(nh,3)=nc(nh,3)+1
c
c   530 call mgetc(hf,j)
c       if(hf(1:5).eq.'$end$')go to 540
c       write(iuo,'(a130)')hf
c       go to 530
c
c   540 continue
c       call mgeti(n,1,i)
c       do 570 k=1,n
c           call mgetd(fso(lfso+6*(ist-1)+1),6,i)
c           if(ist.eq.msrk)ist=0
570   ist=ist+1
c
c       loc(1,nh)=ipos
c       loc(2,nh)=dpos
c       loc(3,nh)=rpos
c       loc(4,nh)=cpos
580 continue

```

Fig. 12.2 マスタ側での受信処理

不要な I/O 関連処理 (open, endfile, rewind) を削除し、SP2 と SR2201 で実行した結果は、Fig.12.4 のとおりである。この結果、両計算機でほぼ同等の測定結果が得られた。


```

c
c   send data to master processor (1)
c
c   call msend(0,33,i)
c
c   endfile iuo
c   rewind iuo
c   if(issw.ne.0)rewind iusc
c   ist=is
c   mp=0
c   call msndi
c   do 840 mt=1,nm
c   if(nc(mt,2).eq.0)go to 840
c
c 780 read(iuo,'(a130)',end=790)hf
c   call mputc(hf,i)
c   if(hf.ne.'$end$')go to 780
c
c 790 n=nc(mt,4)-nc(mp,4)
c   call mputi(n,1,i)
c   do 820 k=1,n
c   call mputd(fso(lfso+6*(ist-1)+1),6,i)
c   if(ist.eq.msrk)ist=0
c 820 ist=ist+1
c 830 mp=mt
c 840 continue
c
c   send data to master processor (2)
c
c   call msend(0,34,i)

```

Fig. 12.3 スレーブでの送信処理

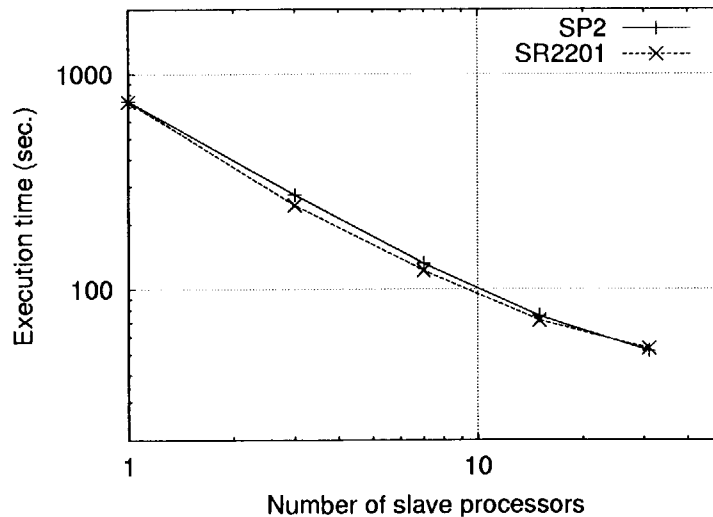


Fig. 12.4 SP2 と SR2201 の測定結果の比較 (ソース修正後)

12.2 I/O 処理が遅い原因

SP2 と SR2201 と I/O 関連のハードウェア構成を比較して、異なる点は、以下のとおりである。

- SP2 : 各プロセッサにローカルにディスクを持っている。
- SR2201 : 各プロセッサにローカルにディスクが無く、IOU に付属しているディスクにネットワークを経由して、アクセスする。

SR2201 のネットワーク性能が高速 (200 MB/sec) であっても、ネットワークを経由してデータをアクセスする限り、ローカル・ディスクを有している SP2 の性能を凌駕することは不可能である。この結果、ハードウェア構成の違いにより、スレーブ・プロセッサがデータを読み書きする場合に、性能差が顕著にあらわれると推定している。

今回は少量のデータのためローカル・ディスクのみで対応することが可能であったが、画像データのような大量データをスレーブ・プロセッサから書き込みを行なうような処理を行なう場合、ローカル・ディスクで対応できないことが考えられる。この場合には、IOU に付属のディスクの方が有利であると考えられる。

13. まとめ

本報告書では日本原子力研究所計算科学技術推進センターが開発・整備しているベンチマークテストシステムの一環として開発した、MCNP4BK-BMT と MCNP4BF-BMT を紹介した。並列BMTコードを開発するにあたって、コードの基礎となった粒子輸送モンテカルロ・コード MCNP4B の MPI 化を実施し、原研内の 4 台のスカラール並列計算機で実施し、PVM 版と同等の性能が得られた。

計算対象（臨界計算および固定源問題）に応じて、2 種類のコードを開発し、それぞれの問題の特長に応じた手法（バイナリ・ツリーを用いたブロードキャスト通信および武宮の手法による負荷分散の均等化）を採用することにより、並列効率の良いコードとなっている。

本報告書で紹介した手法や計測結果が、これから科学技術計算コードの並列化を実施する際の指針として、少しでも役立つことを期待している。

謝辞

日本原子力研究所計算科学推進センターの相川次長には本BMTコードの開発にあたり、多大のご支援を賜りました。また、同センターの樋口健二氏には MCNP4B コードに関する有益な情報やデータを提供していただきました。ここに感謝の意を表します。

参考文献

- [1] 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.
- [2] 折居茂夫, 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, *JAERI-Data/Code*, 98-020, (1998)
- [3] 樋口健二, 川崎琢治, 粒子輸送モンテカルロ・コード MCNP の並列処理, *JAERI-Data/Code*, 96-017, (1996)
- [4] 樋口健二, 武宮博, 川崎琢治, 高並列処理における粒子輸送モンテカルロ・コードの負荷分散, 日本原子力学会誌, Vol.40, No.10, pp. 798-808, 1998.
- [5] 武宮博, 川崎琢治, 本間一朗, 太田浩史, 樋口健二, 粒子輸送モンテカルロ・コードの高並列処理, 計算工学会講演論文集, Vol.1, No.1, pp. 93-96, 1996.
- [6] Judith F. Briesmeister, MCNP - A General Monte Carlo N-Particle Transport Code, Version 4B. LA-12625-M, Manual, 1997.
- [7] NEA Nuclear Science Committee, Internal Handbook of Evaluated Criticality Benchmark Experiments, 4, NEA/NSC/DOC(95)05/IV, 1995.
- [8] 小室雄一, 「国際臨界安全ベンチマーク実験ハンドブック」の紹介, 日本原子力学会誌, Vol.40, No.9, pp. 697-701, 1998.

付録1 臨界計算 (MCNP4BK-BMT) の測定結果

1.1 IBM SP2 の測定結果

臨界計算における IBM SP2 の測定結果を Table A.1 に示す。なお、表中の記号 NPE, Total, In, PD, PT, TS および OP は、それぞれ、プロセッサ台数、初期化、粒子配分、粒子シミュレーション、粒子収集と統計計算および出力処理である。具体的な処理内容は本文を参照のこと。

1.2 日立 SR2201 の測定結果

臨界計算における日立 SR2201 の測定結果を Table A.2 に示す。なお、記号の意味は IBM SP2 と同じである。

1.3 Intel Paragon の測定結果

臨界計算における Intel Paragon 測定結果を Table A.3 に示す。なお、記号の意味は IBM SP2 と同じである。

1.4 富士通 AP3000 の測定結果

臨界計算における富士通 AP3000 の測定結果を Table A.4 に示す。なお、記号の意味は IBM SP2 と同じである。

Table A.1 IBM SP2 の測定結果

NPE	NP	Total	In	PD	PT	TS	Op
01	500	381.51	9.35	0.00	370.97	0.00	1.19
04	500	148.10	12.67	1.41	132.19	0.38	1.45
08	500	77.27	12.08	1.46	61.54	0.74	1.45
12	500	59.69	11.26	1.98	41.05	4.24	1.16
16	500	48.28	12.00	2.04	31.31	1.55	1.38
24	500	38.98	11.53	2.95	20.80	1.55	1.38
32	500	34.61	10.38	3.16	16.25	3.40	1.43
48	500	38.24	13.63	4.18	13.42	5.49	1.53
01	1000	748.51	9.08	0.00	738.08	0.00	1.35
04	1000	274.43	10.78	1.42	257.18	3.61	1.43
08	1000	131.59	10.39	1.45	117.81	0.83	1.10
12	1000	94.89	12.55	1.40	77.65	1.27	1.40
16	1000	74.80	10.90	2.10	59.01	1.57	1.22
24	1000	59.73	11.59	2.90	41.53	2.50	1.20
32	1000	51.70	11.73	3.57	31.65	3.58	1.17
48	1000	43.09	10.90	4.08	20.70	5.47	1.96
01	1500	1125.90	10.65	0.00	1113.24	0.00	1.98
04	1500	402.06	15.73	1.03	383.27	0.47	1.23
08	1500	189.87	11.02	2.92	173.84	0.91	1.23
12	1500	132.79	11.54	2.04	116.60	1.38	1.23
16	1500	103.51	11.08	2.11	87.42	1.72	1.18
24	1500	77.98	11.29	2.87	59.87	2.71	1.24
32	1500	66.15	11.68	4.00	45.62	3.65	1.20
48	1500	54.45	11.41	4.46	31.48	5.81	1.28
01	2000	1474.04	10.49	0.00	1460.31	0.00	3.25
04	2000	523.97	10.61	1.41	510.05	0.50	1.40
08	2000	247.26	13.36	1.53	229.12	0.97	1.54
12	2000	166.42	10.63	2.05	151.23	1.32	1.18
16	2000	132.71	12.81	3.12	114.48	1.78	1.51
24	2000	96.23	10.56	2.78	78.99	2.74	1.16
32	2000	79.77	11.78	6.06	60.09	3.54	1.45
48	2000	64.83	12.47	3.59	41.84	5.59	1.34

Table A.2 日立 SR2201 の測定結果

NPE	NP	Total	In	PD	PT	TS	Op
01	500	348.93	14.00	0.00	334.03	0.00	0.78
04	500	132.13	12.68	0.34	117.90	0.52	0.66
08	500	72.67	13.47	0.43	56.69	1.43	0.62
12	500	60.68	18.16	0.58	39.21	1.75	0.91
16	500	47.67	13.78	0.67	29.87	2.72	0.58
24	500	44.77	16.39	1.00	22.55	3.72	1.04
32	500	40.12	13.79	1.21	19.57	4.80	0.70
48	500	44.24	15.83	1.76	17.99	7.99	0.63
01	1000	700.48	15.18	0.00	684.21	0.00	0.95
04	1000	245.48	13.93	0.40	229.89	0.64	0.58
08	1000	121.91	14.57	0.40	105.26	1.06	0.58
12	1000	96.18	20.01	0.55	72.66	2.22	0.69
16	1000	71.46	13.43	0.64	54.29	2.44	0.58
24	1000	66.29	20.06	0.91	40.61	3.94	0.73
32	1000	53.06	13.62	1.13	32.65	5.05	0.56
48	1000	50.16	14.89	2.00	24.91	7.68	0.63
01	1500	1125.90	10.65	0.00	1113.24	0.00	1.98
04	1500	402.06	15.73	1.03	383.27	0.47	1.23
08	1500	189.87	11.02	2.92	173.84	0.91	1.23
12	1500	132.79	11.54	2.04	116.60	1.38	1.23
16	1500	103.51	11.08	2.11	87.42	1.72	1.18
24	1500	77.98	11.29	2.87	59.87	2.71	1.24
32	1500	66.15	11.68	4.00	45.62	3.65	1.20
48	1500	54.45	11.41	4.46	31.48	5.81	1.28
01	2000	1474.04	10.49	0.00	1460.31	0.00	3.25
04	2000	523.97	10.61	1.41	510.05	0.50	1.40
08	2000	247.26	13.36	1.53	229.12	0.97	1.54
12	2000	166.42	10.63	2.05	151.23	1.32	1.18
16	2000	132.71	12.81	3.12	114.48	1.78	1.51
24	2000	96.23	10.56	2.78	78.99	2.74	1.16
32	2000	79.77	11.78	6.06	60.09	3.54	1.45
48	2000	64.83	12.47	3.59	41.84	5.59	1.34

Table A.3 Intel Paragon の測定結果

NPE	NP	Total	In	PD	PT	TS	Op
01	500	1374.93	51.42	0.00	1314.34	0.00	8.82
04	500	529.65	43.62	8.12	468.73	4.68	4.48
08	500	288.45	49.27	8.99	223.83	1.08	5.27
12	500	233.44	58.15	13.11	154.73	1.54	5.62
16	500	197.97	54.30	12.22	123.52	2.08	5.85
24	500	167.13	52.04	18.07	87.52	3.48	5.98
32	500	155.51	53.18	18.75	72.07	5.14	6.35
64	500	239.36	48.79	26.43	142.19	16.39	5.54
01	1000	2721.86	57.63	0.00	2649.37	0.00	14.72
04	1000	974.76	47.24	8.13	909.61	4.68	5.09
08	1000	486.63	49.75	8.98	421.29	1.10	5.50
12	1000	363.55	53.59	13.34	288.79	1.56	6.25
16	1000	296.09	55.68	12.22	220.03	2.01	6.13
24	1000	239.06	52.97	17.66	159.68	3.10	5.62
32	1000	202.17	49.06	18.97	123.84	4.72	5.57
64	1000	246.59	54.91	27.19	142.37	15.78	6.32
01	1500	4004.99	59.41	0.00	3916.97	0.00	27.66
04	1500	1416.99	49.53	8.35	1345.11	0.72	13.27
08	1500	702.37	47.12	8.98	639.41	1.13	5.70
12	1500	500.54	56.57	13.34	423.28	1.59	5.74
16	1500	510.18	158.56	12.21	331.59	2.05	5.73
24	1500	302.93	50.45	18.73	225.96	3.21	5.55
32	1500	253.91	47.27	18.63	178.21	4.49	5.27
64	1500	274.63	54.05	26.55	172.21	16.50	5.25
01	2000	5355.25	48.27	0.00	5282.10	0.00	48.27
03	2000	1848.86	51.29	8.40	1777.89	0.72	10.54
07	2000	872.86	49.78	8.96	808.26	1.14	5.34
11	2000	628.07	53.62	13.38	554.17	1.62	6.26
15	2000	477.41	50.86	12.20	406.93	2.05	5.36
23	2000	361.79	46.88	17.70	288.63	3.15	5.40
31	2000	314.52	60.89	18.19	220.80	4.51	10.12
63	2000	220.02	47.43	26.32	128.88	11.60	5.66

Table A.4 富士通 AP3000 の測定結果

NPE	NP	Total	In	PD	PT	TS	Op
01	500	136.99	2.18	0.00	134.67	0.00	0.11
03	500	74.11	2.85	0.40	70.61	0.15	0.11
04	500	51.66	1.40	0.43	49.45	0.23	0.13
01	1000	270.33	2.18	0.00	268.00	0.00	0.11
03	1000	141.98	2.58	0.40	138.73	0.17	0.11
04	1000	109.12	8.48	0.45	99.68	0.40	0.11
01	2000	539.62	3.18	0.00	536.29	0.00	0.11
02	2000	275.52	2.44	0.43	272.68	0.22	0.11
03	2000	199.05	8.09	0.45	190.02	0.38	0.11

付録2 固定源問題 (MCNP4BF-BMT) の測定結果

2.1 IBM SP2 の測定結果

固定源問題における IBM SP2 の測定結果を Table B.1 に示す.

2.2 日立 SR2201 の測定結果

固定源問題における日立 SR2201 の測定結果を Table B.2 に示す.

2.3 Intel Paragon の測定結果

固定源問題における Intel Paragon の測定結果を Table B.3 に示す.

2.4 富士通 AP3000 の測定結果

固定源問題における富士通 AP3000 の測定結果を Table B.4 に示す.

Table B.1 IBM SP2 の測定結果

NPE	NP	Total	In	PD	PT	TS	Op
01	1000000	1310.19	39.44	0.00	1266.99	0.00	3.76
04	1000000	481.67	38.65	0.54	440.59	0.65	1.24
08	1000000	234.17	38.23	0.93	192.55	1.22	1.24
12	1000000	167.87	39.99	1.12	123.57	1.75	1.42
16	1000000	133.68	39.19	1.34	89.54	2.28	1.33
24	1000000	108.57	40.35	1.63	61.74	3.30	1.55
32	1000000	93.83	38.93	1.52	48.08	4.07	1.23
48	1000000	84.91	40.29	1.31	36.22	5.66	1.42
01	2000000	2598.34	39.67	0.00	2554.17	0.00	4.50
04	2000000	923.76	38.60	0.62	882.54	0.74	1.27
08	2000000	425.80	38.93	1.04	383.24	1.30	1.28
12	2000000	289.81	39.91	1.35	245.20	1.90	1.45
16	2000000	225.57	38.61	1.68	181.55	2.42	1.31
24	2000000	169.45	39.93	1.95	122.65	3.50	1.41
32	2000000	143.00	39.23	2.09	95.99	4.34	1.34
48	2000000	120.22	40.42	2.16	69.91	6.17	1.57
01	3000000	3857.45	44.18	0.00	3806.21	0.00	7.07
04	3000000	1355.94	40.00	0.56	1312.21	0.74	2.43
07	3000000	612.83	40.01	1.31	568.53	1.51	1.46
12	3000000	412.92	39.19	1.46	368.82	2.00	1.45
16	3000000	322.19	39.88	1.67	273.22	5.94	1.47
24	3000000	232.68	43.92	2.26	181.20	3.81	1.49
32	3000000	187.07	39.24	2.24	139.71	4.50	1.39
48	3000000	151.95	40.67	2.50	100.67	6.54	1.54
01	4000000	5125.32	39.70	0.00	5077.99	0.00	7.63
04	4000000	1797.19	39.29	0.60	1754.52	0.75	2.07
08	4000000	806.79	38.46	1.26	764.35	1.47	1.25
12	4000000	535.02	40.14	1.62	489.65	2.10	1.49
16	4000000	409.69	38.31	1.78	365.84	2.51	1.23
24	4000000	294.39	44.56	2.52	241.88	3.91	1.52
32	4000000	236.23	43.99	2.85	183.23	4.79	1.36
48	4000000	185.41	45.87	2.57	125.62	9.96	1.41

Table B.2 日立 SR2201 の測定結果

NPE	NP	Total	In	PD	PT	TS	Op
01	1000000	1327.84	32.22	0.00	1292.99	0.00	2.61
04	1000000	461.43	32.39	0.61	426.67	0.80	0.91
08	1000000	219.43	32.45	0.89	183.83	1.38	0.85
12	1000000	154.42	32.44	1.01	118.65	1.48	0.80
16	1000000	125.29	33.00	0.89	88.45	2.01	0.90
24	1000000	98.00	32.70	1.04	61.10	2.29	0.83
32	1000000	85.86	33.34	1.00	48.31	2.31	0.85
48	1000000	74.99	33.39	1.26	36.41	3.10	0.79
01	2000000	2639.53	33.71	0.00	2602.94	0.00	2.83
04	2000000	890.73	32.31	0.84	855.39	0.97	1.18
08	2000000	407.33	32.88	1.20	370.75	1.61	0.86
12	2000000	276.11	33.20	1.23	238.82	1.72	1.09
16	2000000	212.60	32.44	1.50	175.84	1.95	0.82
24	2000000	156.80	33.56	1.50	117.94	2.35	1.40
32	2000000	127.92	32.07	1.60	90.15	3.19	0.86
48	2000000	104.58	32.89	2.02	64.79	3.96	0.87
01	3000000	3956.65	32.79	0.00	3920.59	0.00	3.24
04	3000000	1327.23	31.52	1.01	1292.25	1.26	1.15
08	3000000	588.07	32.52	1.53	551.33	1.76	0.89
12	3000000	392.63	32.07	1.77	355.37	2.46	0.93
16	3000000	300.15	31.65	1.90	263.34	2.39	0.84
24	3000000	213.16	33.00	1.77	174.58	2.88	0.87
32	3000000	171.36	32.18	1.92	132.90	3.26	1.05
48	3000000	133.77	33.31	2.22	93.21	4.11	0.87
01	4000000	5271.26	31.74	0.00	5235.79	0.00	3.70
04	4000000	1744.58	32.67	1.26	1708.35	1.30	1.06
08	4000000	780.13	31.03	1.52	744.57	2.13	0.95
12	4000000	513.08	34.39	1.84	473.45	2.44	0.91
16	4000000	385.41	32.44	1.82	348.06	2.24	0.81
24	4000000	269.82	32.71	2.22	230.96	3.02	0.87
32	4000000	212.39	33.06	2.31	172.68	3.50	0.80
48	4000000	161.20	33.32	2.70	119.41	4.80	0.94

Table B.3 Intel Paragon の測定結果

NPE	NP	Total	In	PD	PT	TS	Op
01	1000000	4982.77	152.35	0.00	4806.21	0.00	24.20
04	1000000	1760.03	156.20	3.44	1589.19	2.84	8.34
08	1000000	867.69	156.63	5.66	695.83	3.99	5.58
16	1000000	522.76	160.73	11.92	338.85	5.64	5.62
32	1000000	372.47	157.12	15.08	187.89	6.95	5.41
64	1000000	330.53	156.74	18.00	140.77	9.91	5.10
01	2000000	9933.89	157.36	0.00	9731.87	0.00	44.65
04	2000000	3372.89	150.10	4.08	3198.84	3.88	15.98
08	2000000	1565.34	155.25	6.54	1388.83	5.25	9.48
16	2000000	843.98	155.80	13.47	662.43	7.25	5.01
32	2000000	530.21	152.10	16.32	347.41	8.82	5.55
64	2000000	407.66	153.07	19.33	218.86	10.72	5.67
01	4000000	19497.34	156.99	0.00	19252.23	0.00	88.10
04	4000000	6558.06	150.05	4.95	6376.04	5.27	21.75
08	4000000	2939.66	154.78	7.75	2757.14	7.09	12.88
16	4000000	1496.48	157.35	14.62	1306.58	9.67	8.26
32	4000000	851.31	155.57	18.14	660.77	11.23	5.59
64	4000000	573.75	157.75	19.80	378.14	11.99	6.06

Table B.4 富士通 AP3000 の測定結果

NPE	NP	Total	In	PD	PT	TS	Op
01	500	570.28	16.74	0.00	552.83	0.00	0.71
03	500	296.92	16.88	0.61	278.74	0.43	0.25
04	500	204.99	16.88	0.72	186.62	0.50	0.26
01	1000	1139.15	16.86	0.00	1121.35	0.00	0.94
03	1000	581.56	17.92	0.82	562.06	0.48	0.27
04	1000	394.47	17.07	1.07	375.44	0.63	0.25
01	2000	2265.97	16.54	0.00	2248.01	0.00	1.42
03	2000	1141.39	17.23	1.19	1121.77	0.65	0.54
04	2000	768.39	16.96	1.45	748.85	0.85	0.28

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	ジーメンズ	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	°C	
光束度	ルーメン	lm	cd·sr
照射度	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
線量当量	シーベルト	Sv	J/kg

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	b
バール	bar
ガリ	Gal
キュリー	Ci
レントゲン	R
ラド	rad
レム	rem

1 Å = 0.1 nm = 10⁻¹⁰ m

1 b = 100 fm = 10⁻²⁸ m²

1 bar = 0.1 MPa = 10⁵ Pa

1 Gal = 1 cm/s² = 10⁻² m/s²

1 Ci = 3.7 × 10¹⁰ Bq

1 R = 2.58 × 10⁻⁴ C/kg

1 rad = 1 cGy = 10⁻² Gy

1 rem = 1 cSv = 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)

- 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。
- 表4には海里, ノット, アール, ヘクト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。
- barは, JISでは流体の圧力を表わす場合に限り表2のカテゴリに分類されている。
- 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/m²) = 10 P (ポアズ) (g/(cm·s))

動粘度 1 m²/s = 10⁴ St (ストークス) (cm²/s)

圧	MPa (=10 bar)	kgf/cm ²	atm	mmHg (Torr)	lbf/in ² (psi)
	1	10.1972	9.86923	7.50062 × 10 ³	145.038
力	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 = 4.18605 J (計量法)
 = 4.184 J (熱化学)
 = 4.1855 J (15 °C)
 = 4.1868 J (国際蒸気表)
 仕事率 1 PS (仏馬力)
 = 75 kgf·m/s
 = 735.499 W

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

M C N P 4 B の M P I 化 お よ び M C N P 4 B を 基 に し た 並 列 B M T コ ー ド の 開 発