

モンテカルロ法による輸送計算の Grid コンピューティング

太田 理¹
松澤照男²
奥田洋司³

高精度を要求する原子炉の臨界計算において、モンテカルロ法が必須となるが、多大な計算時間がかかる。そのため、並列計算を行うことになるが、計算ノードを増やすにはコストなどがかかってくるので、容易に計算ノードを増やすことが出来ない。そこで、他機関の計算ノードを利用する事できるような Grid が開発されて来ている。その環境に計算コードを移植し、Grid コンピューティングを行い、パフォーマンスを測定した。移植した計算コードは、動的負荷分散を考慮して開発され、臨界計算でよく使われているモンテカルロ法による粒子輸送コード MORSE を使用した。その結果、容易に既存のコードを移植でき、Cluster を用いた並列計算と同等の速度向上率を得られたことより、Grid の有効性が得られた。

Grid Computing of Transport Calculation by Monte Carlo Method

Osamu OHTA¹,
Teruo MATSUZAWA²,
Hiroshi OKUDA³

High accuracy of critical calculation for the reactor by the Monte Carlo Method takes a long time. Then the calculation is done using parallel. But increasing node cannot do. And so, recent year, parallel calculation does by Grid. Using Code is MORSE code with dynamic load balance. The code is calculated by Grid-computing, and we measured the performance of code. In result, the existing parallel code can be transplanted easily. And Grid is available to provide a lot of effectiveness from results of the speed up ratio with PC-cluster.

1. はじめに

中性子や光子の輸送問題を解く方法として、直接積分法、 S_N 法(離散座標法)、モンテカルロ法などがある。前 2 つは、決定論的手法と呼ばれ、最後の方法は確率論的手法(非決定論的手法)と呼ばれる。決定論的手法はボルツマン輸送方程式を連立代数方程式に変化させ、その変数を離散化することによって、共鳴吸収などの解を得ている。一方、確率論的手法と呼ばれるモンテカルロ法は、共鳴吸収などの解を得る際、擬似乱数を使用して計算機実験により解いていくものである。

確率論的手法であるモンテカルロ法は、決定論的

手法に比べて、複雑な形状において厳密な解を得る事が出来る。しかしながら、モンテカルロ計算の精度は、サンプリングする粒子の数 N の平方根に比例するため、精度の良い結果を得るためには多くの粒子を追跡するために、多大な計算時間が必要となる。

このようにモンテカルロ法では高精度の解に対して計算時間がかかるため、より効率の良い分散低減法が考えられている。 S_N 法で使われている粗メッシュ再釣合法をモンテカルロコードに適用しているものもその一つである。一方、ベクトル・並列化のようなハード的に計算時間の短縮が考慮されている。モンテカルロ法による輸送計算の場合、中性子や光子を生成から消滅まで衝突を考慮しないで追跡する。そのため、分散・並列化がしやすい。近年では、MCNP[1]などにおいて、PVMを使用して分散処理がされているバージョンがある。しかしながら、ベクトル化においては、ループになっているところが少ないため、あまり、考慮されていない。だが、長谷川・樋口らは、Monte-4を使い MORSEコードをベクトル並列処理させた[2]。Monte-4とは、日本原子力研究所がMCの高速処理を実

¹北陸先端科学技術大学院大学 情報科学研究科 情報処理学専攻
Dept. of Information Processing, Japan Advanced Institute of
Science and Technology

²北陸先端科学技術大学院大学 情報科学センター
Center for Information Science, Japan Advanced Institute of
Science and Technology

³東京大学大学院 工学系研究科 システム量子工学専攻
Dept. of Quantum Engineering and System Science, University
of Tokyo

現させるために開発したモンテカルロ・パイプラインと呼ばれるパイプラインを持っている4プロセッサのベクトル演算装置である。

さらに、最近では、計算量の大幅増大により他機関の計算機も利用するグローバルコンピューティングが行われているようになってきている[3]。例えば、データ管理としてデータGrid[4]、計算科学としてGridコンピューティング、さらには、共通の可視化データを用いて遠隔地同士のコラボレーションとしてのVizGrid[5]などがある。Gridコンピューティングとは広域ネットワーク上に分散配置された計算資源を仮想的な高性能計算機と見立てて分散・並列計算を行う計算システムのことを指している。この内、Gridコンピューティングの研究は現在Grid環境を構築・維持するという形で欧米を中心に進められている。そしてGlobus[6]やLegion[7]といったGridコンピューティング環境の基本システムが開発されている。また、実計算への応用研究はH.CasanovaらのNetSolveの研究[8]や佐藤らのnetCFDの研究[9]などが挙げられる。

このようにGridコンピューティングを実現する環境は整いつつある。しかしながら、Gridコンピューティングはヘテロ環境になりやすい。そのため、計算の負荷バランスを考察しなければならない。ところが、このような研究はほとんど行われていないのが現状である。

そこで本研究では、負荷バランスを考慮された実計算のGridへの適用研究として行った。その際の負荷分散を考察した。次章では今回使用した粒子輸送モンテカルロ計算コード・MORSEについて説明をする。また、並列化についても説明をする。第3章ではGrid環境を構築する上で使用したglobus toolkit™について説明をする。また、MPIをglobus toolkit™に適合させられたMPICH-G2について説明をする。第4章では計算を行う上で行った条件を示す。第5章では計算結果を示し、第6章では今回得られた計算結果とClusterでの並列計算との比較を行い、最後にまとめを述べる。

2. 輸送計算コード

2-1 MORSEの特徴

今回使用したモンテカルロ法による粒子輸送計算コードは、MORSE (The Multigroup Oak Ridge Stochastic Experiment)[10]にCombinatorial Geometry(幾何組合せ)法と呼ばれる体形記述を採用したMORSE-CGA[11]を使用した。このため、基本体形(球、円柱、立方体)を利用して、さまざまな形状を取り扱うことが出来、中性子と線の輸送問題を解くことが出来る。多くの臨界・遮蔽問題を解くツールとして設計され、多群断面データを利用することで、中性子、線、中性子と線が組み合わさった問題を線源方向からのforwardモード又は検出器方向からのadjointモードで解くことが出来、さらに、遮蔽と臨界問題の時間依存を扱うことが出来る。また、角度分布のP₁₆展開までの等方又は非等方散乱を考慮できる。また、CGの機能により、原子炉の炉心や多重に仕切られた構造物について矩形格子モデル化することが出来る。

MORSEコードは、大きく分けて、3つのループがあり、一番目はRUNループと呼ばれるループで、さまざまな形状データを持った問題を解くために設けられている。二番目は、Batchループと呼ばれ、統計処理をするためのループとなっている。その統計処理を行った後、次のループの追跡する粒子数を決定するため、世代ループとも呼ばれる。また、最後のループはHistoryループと呼ばれ、粒子自体を追跡するためのループとなっている(Fig1.を参照)。

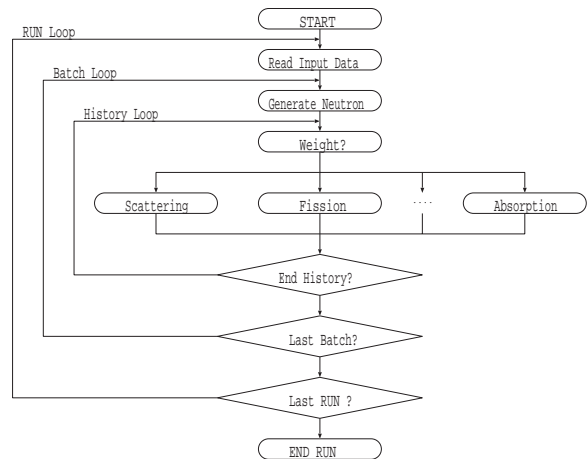


Fig1. Follow of MORSE-CGA

2-2 並列化

モンテカルロ法による粒子輸送計算の場合、粒子を生成から消滅までを追跡する。さらに、粒子同士の衝突などの相互作用は無視しているため、Historyループにおいて、粒子追跡を並列・分散的に処理することが出来る。そのため、計算時間を短縮することが出来る。しかしながら、著者らの過去の研究において、粒子の寿命がさまざまであることより、粒子追跡を均等数振り分ける(静的負荷分散)と、通信時間や待ち時間に遊びが生じて効率が低下してしまうことが確認された[12]。そこで、粒子を追跡する計算部分のみを動的振り分ける(動的負荷分散)手法を導入し効率を向上させ、その分配する粒子の数(以下、分配粒子数と呼ぶ)の予測を可能とした。[13]。なお、今回はあらかじめ分配粒子数を指定する形になっている。また、並列化の際に粒子管理を行えるように、マスタ・スレーブ方式を採用した。つまり、マスタプロセッサ(以下マスタ)において、粒子管理・統計処理などをし、スレーブプロセッサ(以下スレーブ)では粒子追跡のみを行う方式である。

3. グローバルコンピューティング

3-1 globus toolkit™[14]

Gridコンピューティングを行うにあたっては、遠隔計算機の利用、通信、ユーザー認証などさまざまな技術が必要になる。そこでこれらをサポートするツールキットとして、globus toolkit™がある。これはソースが公開されているフリーのツールキットである。

3-2 MPICH-G2 [15][16]

MPICH-G2はglobus2 deviceを用いて実装されたMPICHである。このためにMPI-1に準拠したプロ

グラムであれば、移植は簡単に行える。なお、プログラムを走らせる際、RSL (Resource Specification Language) ファイルを作成する必要がある。RSL ファイル自体は、自動生成する事が可能である。なお、Fig2. に、192.168.0.1 にある globus サーバーに対して PI_mpi (MPI を使用して を計算させる) という実行ファイルを 2 個の PE (Prosser Element) を使用して実行させるための RSL ファイルを示す。また、9PE を使用した例の machinefile を Fig3. に示す。また、実行ファイルは、現在、全ての計算資源の同じディレクトリにあらかじめ配置しておく必要がある。

```

+
( (&(resourceManagerContact="192.168.0.1")
(count=2)
(label="subjob 0")
(environment=(GLOBUS_DUROC_SUBJOB_INDEX 0)
(LD_LIBRARY_PATH /usr/local/globus/lib/)
(arguments= "1000000")
(directory="/home/ohta/MPI/PI_mpi")
(executable="/home/ohta/MPI/PI_mpi/pi")
)

```

Fig2. RSL (Resource Specification Language) file

```

"ume1f.hpcc.jp" 1
"ume00.hpcc.jp" 2
"ume01.hpcc.jp" 2
"koume00.hpcc.jp" 2
"koume01.hpcc.jp" 2

```

Fig3. machinefile (Master 1CPU,Slave 8CPU)

これらのサービスを利用して Grid コンピューティングを行うが、その際、いくつかの気がついた注意点を述べる。

1. Grid コンピューティングを行う際の計算資源は、現在認証を他機関におく場合、全てグローバル IP を持たないといけない(現在、産業技術総合研究所の田中氏・平野氏らによって通信レイヤーである Nexus を中継できる Nexus_Proxy を作成し、プライベート IP が割り当てられている計算資源も使用できるようにされつつある^[17]。)
2. 自ら認証局を立てることが出来るが、認証は Grid コンピューティングにおける研究となるため、Grid への参加者を集うという意味より、なるべく他機関を利用した方がよい。
3. MPICH-G2 を利用する際、計算資源の把握が必須となる。

4. 計算条件

計算条件は、臨界計算とし10000粒子×100世代とした。中性子問題に対して、forward計算として取り扱った。また、散乱角はP₄展開をさせた。核断面積ライブラリーは、15群の中性子と5群の線からなる EURLIB-LWR-15/5^[18]を使用した。なお、エネルギー群はTable1. に示すとおりである。計算に使用した計算コードは、MORSE-CGAを基にMPIで動的に負荷分散するように開発されたコードを使用した。

Table1. Energy group of EURLIB-LWR-15/5

群	エネルギー範囲	平均速度(m/s)
1	14.9 ~ 5.49MeV	4.0 × 10 ⁷
2	3.68	3.0
3	3.33	2.6
4	2.23	2.3
5	2.02	2.0
6	1.35	1.79
7	0.672	1.31
8	0.369	0.95
9	369 ~ 67.4KeV	5.7 × 10 ⁶
10	40.9	3.1
11	7.10	1.87
12	1.58	0.81
13	1580 ~ 8.32eV	0.32
14	0.414	2.4 × 10 ⁴
15	0	2.48 × 10 ³

計算環境は、産総研の PC クラスタ(ume,koume クラスタ)を使用した。Table2. に Cluster の性能を示す。Cluster の各 PE は、グローバル IP で接続され、Cluster 同士はインターネットを介してデータの通信がされる。

Table2. Performance of PC Cluster

ume クラスタ	Pentium	Dual 1.4GHz
		1GB memory × 32 ノード
koume クラスタ	Pentium	Dual 1.4GHz
		1GB memory × 4 ノード

使用した PE はスレーブプロセッサが一對一になるようにした。つまり、ume クラスタで2PE 使用した場合、koume クラスタでも2PE 使用した。なお、最大 PE は、koume クラスタより、マスタ1PE スレーブ16PE とした。今回取り扱った体系は、京都大学臨界集合体(KUCA)の B 架台を直方体・均一炉心として近似した体系である。これを Fig4. に示す。中心に炉心があり、その上に制御棒が6本引抜かれた状態で存在し、それらの周りに反射体が存在する。

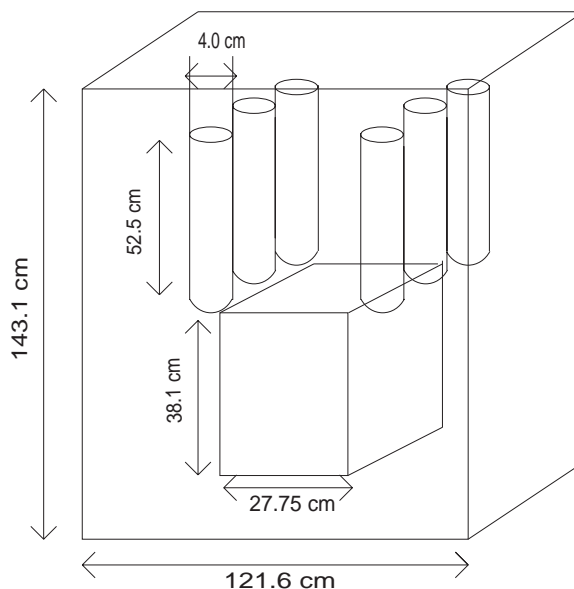


Fig4. Calculation system

また、動的負荷分散時の分配粒子数は、100 粒子分配から 500 粒子分配まで 100 粒子ずつ増やして計測した。

5. 結果

実時間と PE の関係を Fig5. に示す。

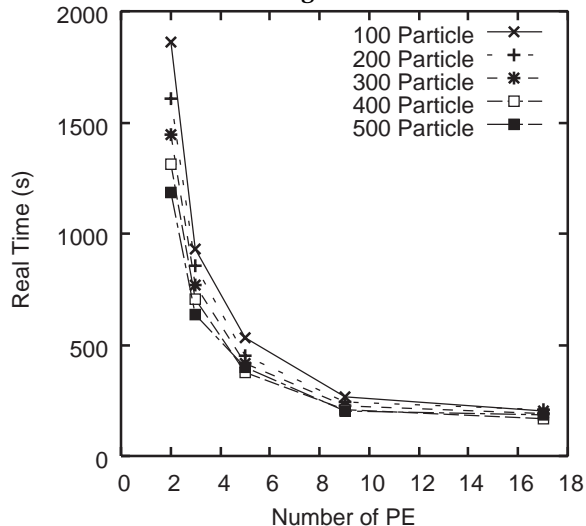


Fig5. Real Time for Grid

また、その時の追跡時間を Fig6. に示し、通信時間を Fig7. に、待ち時間を Fig8. に示す。また、Fig9. に速度向上率を示す。なお、追跡時間・通信時間・待ち時間はスレーブでの計測時間の平均である。

なお、通信時間とは、

1. マスタから、スレーブに粒子データを送信している時間
2. マスタから、スレーブに追跡する粒子の範囲のデータを受け取っている時間
3. スレーブで粒子の追跡計算が終了した後に行われる衝突結果などの追跡結果をマスタに送信している時間

の三つが含まれている。

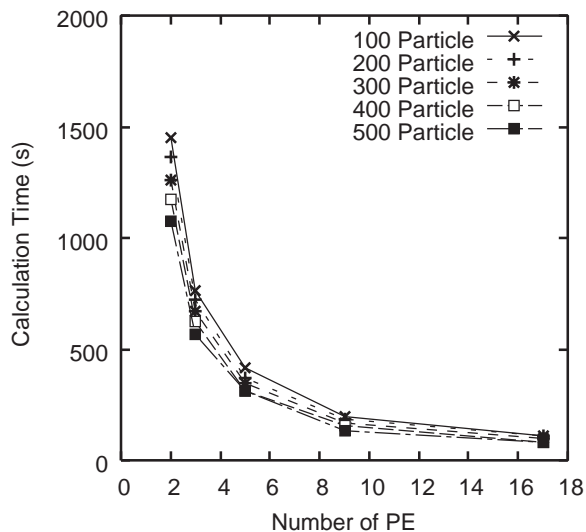


Fig6. Calculation Time for Grid

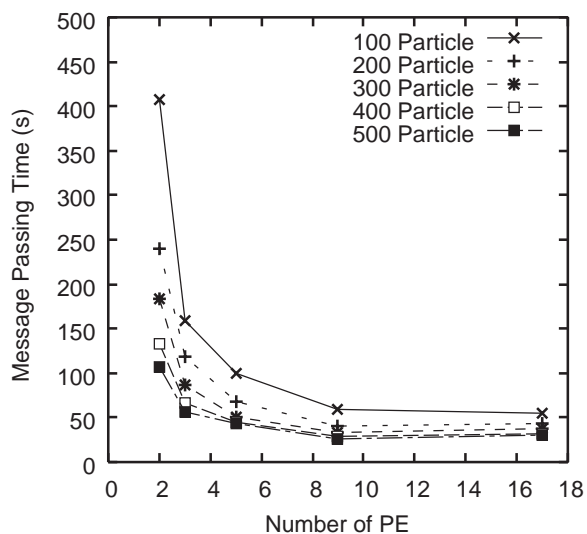


Fig7. Message Passing Time for Grid

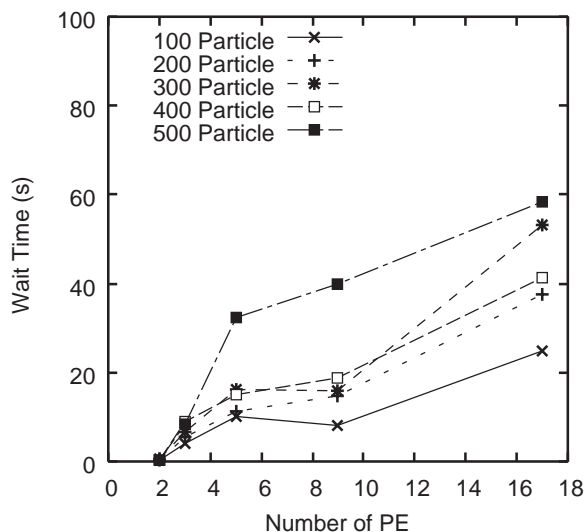


Fig8. Wait Time for Grid

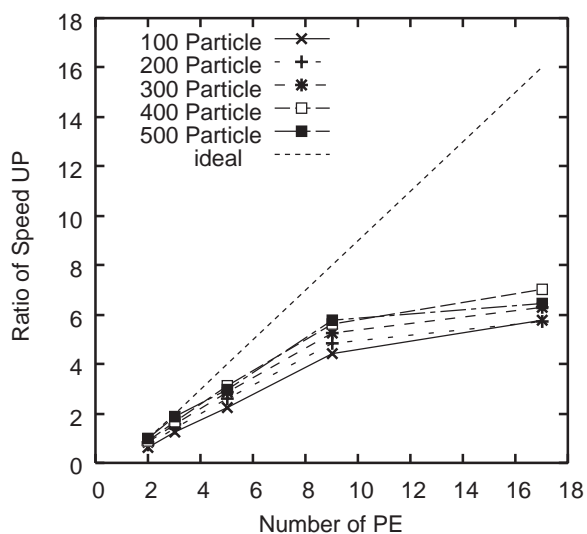


Fig9. Ratio of Speed UP for Grid

また、待ち時間は以下の二つが含まれている。

1. マスタが粒子データを生成している間の待ち時間
2. 一世代が終了するごとに行うマスタでの統計処理を行っている間のスレーブの待ち時間

PE が増加するにつれて実時間が減少していることが Fig5. より解る。しかしながら、9PE を過ぎたところで、伸びが少なくなっている。反対に、待ち時間は 9PE を境に増加していることが Fig8. より見て取れる。一方、動的負荷分散を行った時の分配数の変化は、100 粒子分配よりも 200 粒子分配、200 粒子分配よりも 300 粒子分配と、若干であるが分配粒子数を大きく取る方が高い効率になっている。Fig9. の速度向上率からも、9PE 過ぎから理想より離れていっている。

6. 考察

今回、さらに、同じインプットで ume クラスタのみを使用して並列計算を行った。その結果の速度向上率を Fig10. に示す。

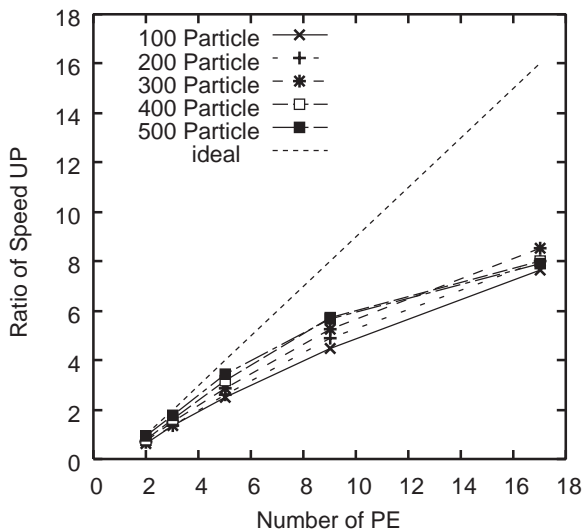


Fig10. Ratio of Speed UP for Cluster

また、Fig11. に Cluster 計算での待ち時間、Fig12. に通信時間を示す。Fig9. と Fig10. を比較すると、同等の速度向上率を得ていることが解る。これは、コード自体が動的負荷分散になっているためだと考えられる。しかしながら、Fig8. より、高並列時に、Grid 計算の方が、待ち時間の増加が多いことがわかる。これは、統計処理のための同期が、ネックになっていることが考えられる。なぜなら、待ち時間には、統計処理のための同期のほかに、粒子データの生成ための待ち時間があるが、追跡している粒子数は同じであることより、Grid 計算にしても Cluster 計算にしても、同じ時間だけ待つことになるため、残りの統計処理のための同期がネックとなる。また、通信時間の減少率は、双方の計算において、横倍になっている。

7. おわりに

本稿では、並列計算機上で開発された MPI コードを Grid 環境への移植を施したが、MPICH-G2 によって、容易に行うことが出来た。また、速度向上率などにより Grid 環境の有効性を見ることが出来た。しかしながら、Cluster 計算における分配粒子数の決定することが出来たが、通信の評価が複雑になることが予想できるので、本稿では追求をしなかった。また、臨界計算であることため、世代ごとに統計処理のための同期を取る必要性があったが、世代数を増やすことより一世代での追跡する粒子を増やした方が良い遮蔽計算などでは、効率がより出ると思われる。また、モンテカルロ計算であるならば、一粒子を生成から消滅までを追跡し、この部分を並列化することが多い。そのため、計算資源がヘテロな環境などにおいてもスレーブ同士の通信が無いので、より効率のよい計算が出来る。今後は、この分配粒子数を追跡計算中に変化することが出来るようにコードを改良する予定である。

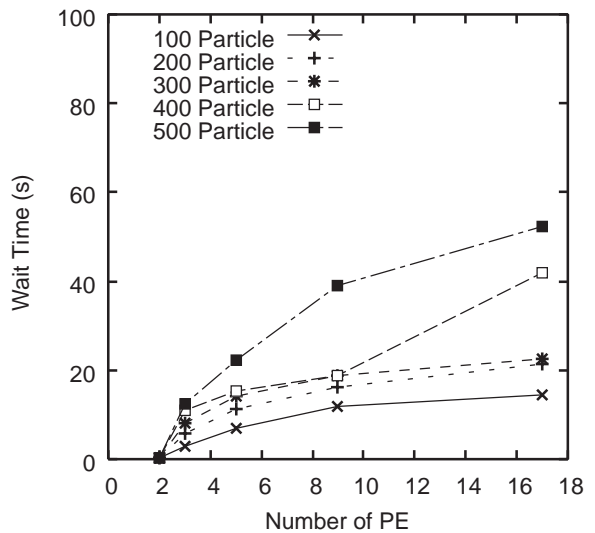


Fig11. Wait Time for Cluster

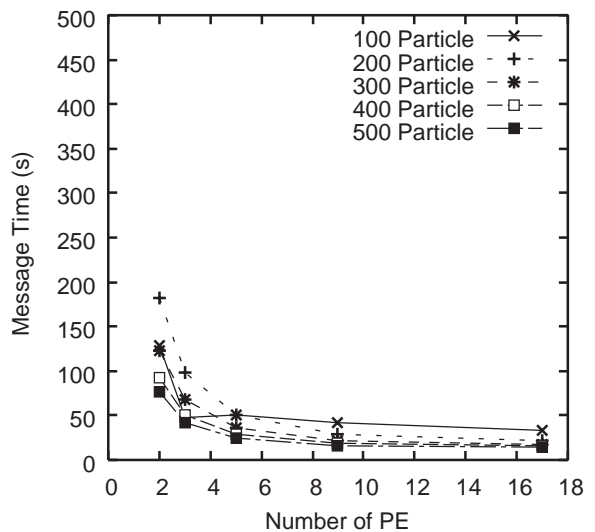


Fig12. Message Passing Time for Cluster

謝辞

東海大学の朝岡卓見教授には MORSE コードについて多大なる助言を頂きました。ここに感謝いたします。また、日本原子力研究所計算科学技術推進センター並列処理基本システム開発グループの樋口氏、長谷川氏には並列処理の手法について多大なる助言を頂きました。ここに感謝いたします。なお、本研究の一部は Asia-Pacific Grid Partnership (ApGrid) の一環として実施いたしました。特に実験に当たって利用方法等についてご協力頂いた産業技術総合研究所グリッド研究センターの皆様には感謝いたします。

¹ <http://laws.lanl.gov/x5/MCNP/index.html>

² 長谷川、樋口, “高速モンテカルロ装置上における MORSE コードのベクトル並列”, JAERI-Tech 95-051, 日本原子力研究所, 1995

³ I. Foster and Carl Kesselman, “The GRID: Blueprint for a New Computing Infrastructure”, Morgan Kaufmann Publishers, 1998

⁴ http://resaerch.nii.ac.jp/kaken-johogaku/reports/H13_overview/A05-09.pdf

⁵ <http://www.vizgrid.org/>

⁶ I. Foster, C. Kesselman. “The Globus Project: A Status Report.”, Proc. IPPS/SPDP '98 Heterogeneous Computing Workshop, pp. 4-18, 1998

⁷ A.S. Grimshaw, W.A. Wulf, the Legion team, “The Legion vision of a worldwide virtual computer”, Communications of the ACM, 1997

⁸ H. Casanova, J. Dongarra “NetSolve: A Network Server for Solving Computational Science Problems ” SUPERCOMPUTING '96 Pittsburgh USA, pp.17-22, 1996

⁹ 佐藤、草野、中田、関口、松岡, “グローバルコンピューティングシステム Ninf を用いた数値流体解析コンポーネント netCFD” 第 10 回計算流体シンポジウム, 1999

¹⁰ M.B.Emmett, “The MORSE Monte Carlo Radiation Transport Code System.” ORNL-TM-4972, Oak Ridge National Laboratory, Oak Ridge, TN, February 1975

¹¹ M.B.Emmett, “MORSE-CGA. A Monte Carlo Radiation Transport Code with Array Geometry Capability”, Oak Ridge National Laboratory, Report ORNL-6174, 1985

¹² 太田、長谷川、樋口, “中性子輸送モンテカルロ・コード MORSE による臨界計算の並列処理”, 日本原子力学会, 秋の年会, 札幌, 2001

¹³ 太田, “中性子輸送モンテカルロ・コード MORSE による臨界計算の並列処理 - 分配粒子数の決定 - ”, 日本原子力学会, 春の年会, 神戸, 2002

¹⁴ <http://www.globus.org>

¹⁵ I. Foster and Nicholas T. Karonis, “A Grid-Enabled MPI: Message Passing in Heterogeneous Distributed Computing Systems”, Proc Supercomputing, 1998

¹⁶ <http://www.globus.org/mpi>

¹⁷ 田中、平野、佐藤、中田、関口, “Globus を用いたグローバルコンピューティング環境の構築とその評価”, インターネットカンファレンス'99, pp.97-106, 1999

¹⁸ <http://www.nea.fr/abs/html/nea-0794.html>