ハイパフォーマンス
コンピューティング 52−17
（１９９４．７．２３）

# ＰＶＭのアプリケーションへの適用効果

## −モンテカルロ法プログラムＭＣＮＰの場合−

長谷部晴美　福井義成

（株）東芝　総合情報システム部

アプリケーションの立場からのＷＳクラスタを利用したＰＶＭによる並列処理の適用効果を、性能向上に関する効果及び実現のための作業コストとの関係という観点から評価を行った。その例題として、モンテカルロ法プログラムＭＣＮＰを対象アプリケーションとして、スーパーコンピュータでのマルチタスキングとＷＳクラスタでのＰＶＭ適用による並列化を行い、それらを比較した。

その結果、ＰＶＭを適用した場合には、作業コストは大きくなるが、３台程度のＷＳを利用して、スーパーコンピュータの並列化性能の0.44倍が実現されたこと，異機種との互換性もあることから発展性が期待できることなどからも、有効であることが確認できた。

## An Efficeincy Evaluation on Parallel Computing with PVM

Hasebe Harumi　Fukui Yoshinari

TOSHIBA Corporation
Total Information and System Division
580-1, Horikawa-cho, saiwai-ku, Kawasaki , Japan

In order to clear dependncy of parallel computing efficient and its cost (man power), evaluated its usefulness on Work Station cluster by PVM and on Supercomputer by auto-tasking and macro-tasking. As an example, applied parallel to Monte Calro Simulation Program MCNP. This paper discusses comparison of these three case, and proposes appicatable parallelism for specific of appication program.

## 1．はじめに

近年、ワークステーション(ＷＳ)の性能は、著しく向上してきており、その性能を活かし、ＷＳクラスタによってスーパーコンピュータと同等、または、それ以上の性能を実現した様々な事例が報告されている。当社においてもＷＳは普及してきており、ＷＳクラスタによる並列処理を行う環境は身近にも存在している。しかし、実際にアプリケーションをそのような環境に移行するかどうかを判断するために必要な並列化の効果及びそのコストについて定量的な指標を示すことは難しい。本報告では、その様な背景から、スーパーコンピュータ，各種の超並列計算機(ＭＰＰ)，ＷＳクラスタなどにおける並列化の"コスト"と効果の関係を明確にしようとする研究の一環として、既存のアプリケーションを例に、スーパーコンピュータとＷＳクラスタでの並列化効果と適用時の作業コストに関する評価結果を報告する。

## 2．評価環境

対象とするモンテカルロ法プログラムにはLos Aloamos National Lab.から提供されているＭＣＮＰを例として使用した。また、スーパーコンピュータにはCRAY Y-MPを、ＷＳクラスタの構成にはSUNのSPARC Station10(SS10)３台を使用した。

表1．評価マシン

| 評価マシン | CPU数 | 主メモリ | キャシュメモリ | ピーク性能 |
|---|---|---|---|---|
| CRAY Y-MP | 4 | 1GB | — | 333MFLOPS |
| SUN SS10 | 1 | 96MB | 1MB | 40MFLOPS |

並列方法には、スーパーコンピュータではマルチタスキングを、ＷＳクラスタではＰＶＭを利用した。ここで、マルチタスキングはＣＲＡＹに実装されている並列化のための機能である。ここではコンパイラによってＤＯループレベルの自動並列化が行われるオートタスキング，及びユーザが並列化用ライブラリを利用してサブルーチンレベルの並列化を行うマクロタスキングを適用した。

また、ＰＶＭは、複数の異機種をＨ／Ｗリソースとしてクラスタマシンを構成し、メッセージパッシングによって並列処理を実現するライブラリとその実行環境である。ここではＰＶＭ３．１を使用した。

## 3．ＭＣＮＰの並列化モデル

ＭＣＮＰは、粒子(中性子，光子，電子)輸送問題などの解析に用いられるモンテカルロ法輸送プログラムであり、その規模は３万ステップ程度のものである。問題性質が並列化に適していることから、Los Alamosでも並列化は行われており、CRAY,VAX等のマルチタスク版やＰＶＭ２．４でインプリメントされたRS/6000版，SUN版などＷＳクラスタ対応版もリリースされている。さらに、RS/6000(100MFLOPS)１６台による性能評価で１３倍の性能向上率を計測した例も報告されている。

ただし、本評価では、並列化の適用過程も評価対象としているため、並列化が行われていないオリジナル版のＭＣＮＰ(図１)を元に適用を行った。

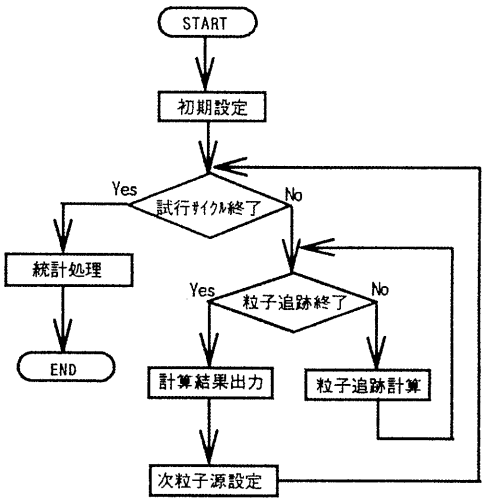


図１．オリジナル版ＭＣＮＰのフロー

ＭＣＮＰは粒子の飛行や衝突などを追跡し、そのサイクルを複数回試行して、最後にそれらの結果を統計処理するものである。図１で示したフローから、オリジナル版では粒子追跡の過程には独立性があり、並列化が可能であることが分かる。

また、本評価では、ＭＣＮＰの例題として、臨界集合体TRX1とTCAの単位セルモデルを使用した。この問題は各サイクル毎に粒子源を設定する固有値問題であり、その処理は逐次的に行われる。しかし、処理開始時に粒子源を与えるようなソース

問題を扱う場合には、各サイクル終了毎に次の粒子源を設定する必要がないため、全試行サイクルの粒子追跡過程を並列化することも可能となる。その場合には、各サイクルで発生する同期処理が不要となり、並列化率はさらに向上する。

3.1 並列化粒度の調査

次に最適な並列化を行うため、並列化の対象となる問題サイズと計算量及び処理時間から、並列化粒度について調べた。問題サイズは、追跡する全粒子数(以下ヒストリーと呼ぶ)で示す。

ここで、並列化粒度とは、一般的に、並列化のために分割されたタスクの計算量(ＣＰＵ量)を意味する。その調査のため、初期設定,粒子追跡,結果処理の各処理を対象として、ＣＰＵ時間から算出した各処理の計算量の全計算量に対する割合を表２に、実行処理時間の全実行時間に対する割合を表３に示す。

表２．問題サイズと計算量（%）

| /ヒストリー | 1000 | 10000 | 50000 | 100000 | 150000 |
|---|---|---|---|---|---|
| 初期設定 | 45.5 | 13.9 | 16.3 | 0.8 | 0.5 |
| 粒子追跡 | 54.3 | 86.1 | 98.3 | 99.1 | 99.4 |
| 結果処理 | 0.3 | 0.1 | 0.1 | 0.0 | 0.0 |

表３．問題サイズと処理時間（%）

| /ヒストリー | 1000 | 10000 | 50000 | 100000 | 150000 |
|---|---|---|---|---|---|
| 初期設定 | 53.0 | 16.7 | 9.6 | 2.5 | 1.0 |
| 粒子追跡 | 47.0 | 83.3 | 90.4 | 97.5 | 99.0 |
| 結果処理 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

以上の結果から粒子追跡の処理割合が、問題サイズに依存して増加していることが分かる。さらに、この粒子追跡処理部のＣＰＵ時間は、全サイクルのTotalを示していることからその並列化粒度は全体の

(粒子追跡の全計算量)／(サイクル数)　%

となり、ここでは２０サイクルでの試行を行っているので、最大で５%程度となることが分かる。

一般に、オートタスクのようなDOループレベルの並列化やデータパラレルは細粒度な問題に適しており、ＰＶＭやマクロタスキングのようなタスクパラレルでは並列化のオーバーヘッドが大きいため粗粒度な問題に適していると言える。しかし、本問題ではサイズが大きくなれば計算量は絶対的に大きくなるが、本問題におけるそのオーバーヘッドはサイズによる影響が小さいため、並列化効果の向上が期待できる。最適な並列化方法の判断には、並列化粒度と共に対象とする問題サイズ,絶対的な計算量とそのオーバーヘッドの影響度などの考慮も重要となる。

3.2 並列処理適用へのモデル化

3.1の結果を元に粒子追跡処理部を対象として、並列化モデルをマスタ／スレーブに分けて図２，３に示す。マスタタスク(マスタ)は、スレーブタスク(スレーブ)を発生し、データを送信する。それらを受信したスレーブは粒子追跡計算を開始し、各サイクル終了まで各タスクは並列実行が可能となる。その間、マスタは各タスクからの送信待ちとなる。スレーブは、１サイクルの粒子追跡終了後、マスタに対して追跡結果を送信する。マスタはそれを受信して、結果を出力する。さらに次のサイクルの粒子源データを設定し、スレーブへ送信する。スレーブではそれらのデータを受信して、次サイクルの粒子追跡計算を開始する。これを繰り返し、全試行サイクル終了後、結果を処理してシミュレーションは終了する。

以上のようなモデルに基づいて、マルチタスキング及びＰＶＭでの並列化適用を行うこととする。

## ４．作業コストと並列化効果

並列化適用の際に要した作業コストとその結果得られた並列化効果について、マルチタスク適用の場合とＰＶＭ適用の場合について比較する。

4.1 マルチタスクの場合

(1)並列化適用の作業コスト

マルチタスキングの内、オートタスキングではオプションの利用によって、コンパイラが自動的に並列化を行うため、作業コストはほとんど発生しない。一方、マクロタスクでは、ユーザが提供されているサブルーチンを使用して、マクロタスキング用にプログラムを変更する必要がある。しかし、共有メモリ方式を使用するため、各タスク間でメモリ領域を共有することができるため、図２，３にあるようなタスク間での通信は必要ない。そのため、基本的にはタスクを制御するルーチンを挿入するだけで実現できる。ただし、各タスク間で同一のメモリ領域を更新する場合には、メモ

リロックが必要になる。ＭＣＮＰでは、メモリ領域のほとんどを共有／結合して使用しているため、その作業に特に負荷を要した。

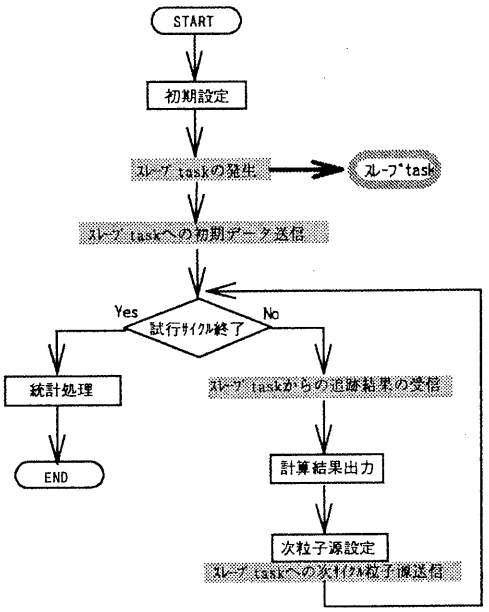


図２．並列化ＭＣＮＰのマスタモデル

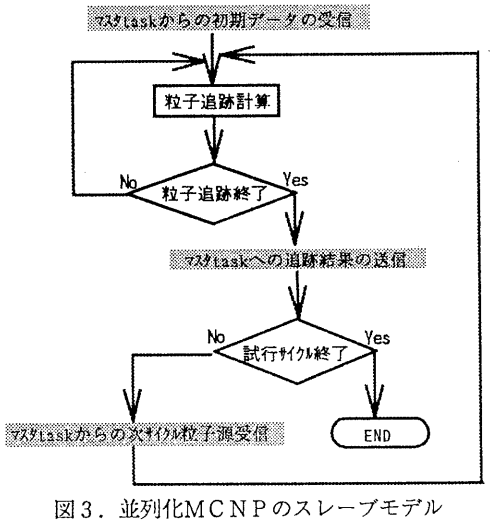


図３．並列化ＭＣＮＰのスレーブモデル

(2)並列処理の効果

以下にＣＲＡＹ上で測定した理論最大性能向上率とオートタスク適用及びマクロタスク適用時の性能向上率を図４にグラフで示す。

ここで、グラフ中に示す理論値とは、Amdahl法則のGustafson修正モデルにより、以下の方法で算出した理論最大性能向上率を意味している。

$$SpeedUp(p) = (Ts+pTp)/(Ts+Tp) \qquad [5]$$

p ：プロセッサ数

Ts：非並列実行部の処理時間

Tp：並列実行部の処理時間

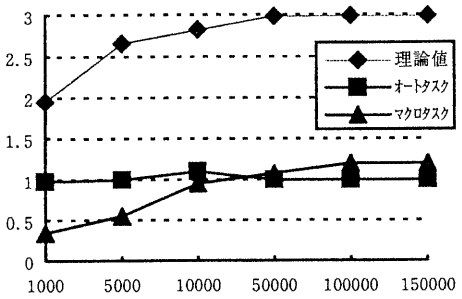


図４．マルチタスク版ＭＣＮＰの性能向上率

図４のグラフから、オートタスキングの効果は、１０％程度に留まっているのに対し、マクロタスキングの効果は、問題サイズが大きくなるとそれに伴って、性能向上率も向上していることが分かる。マクロタスキングでは、最大で約１．２倍の性能向上が計測された。また、マクロタスキングの処理時間が、オートタスキングの処理時間を50000ヒストリー程度で逆転していることから、3.1で述べたように並列処理部分の計算量に対するそのオーバーヘッドの割合が相対的に小さくなり、本問題粒度でもタスクパラレルの並列化効果が得られることを示している。

4.2 ＰＶＭの場合

(1)並列化適用の作業コスト

ＰＶＭを適用する場合には、まず、マスタ／スレーブのタスク分割の作業が発生する。さらに、分散メモリ環境へ移行するための変更を行うと共に、相互で参照／更新するデータをメッセージとして、各タスク間で通信する必要がある。そのため、その作業コストは、通信の発生しないマルチタスクの場合と比較すると非常に大きくなる。また、ＭＣＮＰの特殊なメモリ構造も、分散メモリで使用するための不都合な点があり、その修正にも負荷を要した。しかし、分散メモリ方式では、タスク間で共有するメモリ領域の更新によって起

こる問題は排除されるため、適用時の発生する問題の絞り込みが行い易いというメリットもある。

(2) 並列処理の効果

最大理論性能向上率とＰＶM版の性能向上率を図５に示す。

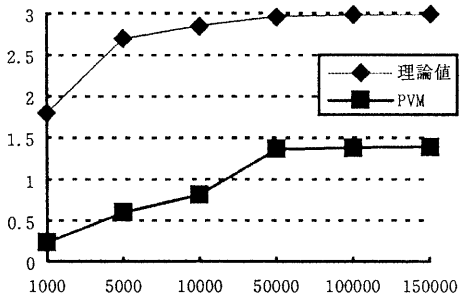


図５．ＰＶM版ＭＣＮＰの性能向上率

図５からも問題サイズが10000ヒストリーまでは、むしろオリジナル版の方が速く、50000ヒストリーからはＰＶM版の方が速くなっていることが分かる。このことから、数千オーダーの小さな問題を対象とする場合には、ＰＶMのオーバーヘッドのために並列化の効果は得られないが、数万オーダー以上の問題を扱う場合には、その効果が期待できると言える。ＰＶMによる並列化の効果は最大で約１．４倍の性能向上が計測された。

ただし、実際のシミュレーションでは、数十万オーダー以上のヒストリーを扱うこと，並列実行部である粒子追跡過程の計算量が増加するような問題が対象となることから並列化の効果が、十分期待できることが分かった。ここで、グラフ中の理論最大性能向上率と実測性能向上率との差が並列化に関するオーバーヘッドであると考えられる。しかし、使用した例題で発生するの同期・通信処理間は、理論最大性能向上率には反映されていない。本評価のＰＶMに関する並列化のオーバーヘッドとしては以下のものが考えられる。

・ＰＶMタスクのエントリ手続き時間
・タスク発生時間
・メッセージバッファ作成／初期化時間
・メッセージのパック／アンパックに関する時間
・メッセージの通信／同期処理とその待ち時間
・ＰＶMタスクの終了手続き時間

また、これらのオーバーヘッドは、問題サイズには影響されないため、問題サイズが大きくなり、並列化率が高くなれば相対的に小さくなると予想できる。そのため、問題サイズが大きくなれば並列化の効果はより大きくなると考えられる。

５．おわりに

CRAYでマクロタスクを適用した場合とＷＳクラスタでＰＶMを適用した場合の処理時間を図６に示す。縦軸は処理時間(Sec)，横軸はヒストリーを表す。

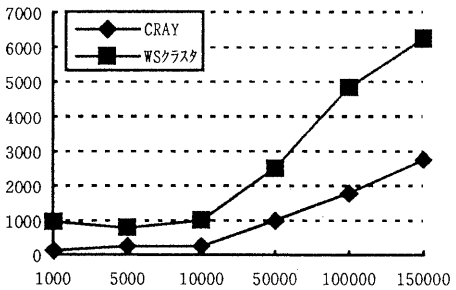


図６．並列化ＭＣＮＰの処理時間

処理性能が、スーパーコンピュータの３６％程度であるＷＳクラスタで、約４４％のパフォーマンス(150000ヒストリーの場合)を実現することができた。また、並列化適用の際の作業コストと並列化の効果を表４にまとめた。作業コストはマルチタスクとＰＶMに関するコストを定量的な指標として、作業人月で示す。並列化効率は、10000ヒストリー以上の全ケースでの平均性能向上率を示す。

表４．並列化の効果と作業コストの比較

| メモリ方式 | 共有メモリ | | 分散メモリ |
|---|---|---|---|
| 並列化方法 | オートタスク | マクロタスク | PVM |
| 並列化効率 | 1.0 | 1.1 | 1.2 |
| 作業コスト | 0.0 | 2.0 | 5.0 |

これらの結果から最適な方法を結論づけることは難しいが、適用の際に重視する事項が分かっていれば、この評価例をその判断材料とすることができる。

また、マクロタスキングとＰＶMを比較すると、ＰＶMで並列化を行ったＳ／Ｗリソースでは、異

機種マシンとの互換性もあることから、プロセッサ数をコントロールすることが簡単であり、最大性能をマシン数に応じて向上させることもできるなどの発展性があることが挙げられる。問題対象についても、同期／通信処理がサイクル中に発生する固有値問題を扱ったケースでもこの程度の効果が得られたことから、ソース問題に適用すればさらに効率は向上すると予想できる。

以上のことから、身近にあるＷＳを利用できる点，並列化効率が良い点，発展性が期待できる点などから、本問題に対するＷＳクラスタによる並列化は相当するコストを要しても有用であるという結論が得られた。また、今後は各種ＭＰＰでも"コスト"と効果の関係を明らかにしていきたい。最終的には、各問題の並列性と様々なアーキテクチャのＭＰＰ(特にプロセッサ間の"ネットワーク")の適合性についての指針を得たいと考えている。これが実現できれば、ＭＰＰのアーキテクトに対し、アプリケーションからの提言が可能となる。

参考文献

[1]Judith F.Briesmeister Los Alamos "MCNP-A General Monte Carlo Code for Neutron and Photon Transport"

[2]Geist,A.,Beguelin,A.,Dongarra,J.,Jiang,W.,Manchek,R.,and Sunderam,V.,"PVM 3.2 user's guide and reference manual"

[3]Gregg W.McKinney Los Alamos "DISTRIBUTED PROCESSOR MONTE CALRO:MCNP RESULTS ON A 16-NODE IBM CLUSTER"

[4]Frank Schmitz, Ulrich Fischer Kernforschungszentrum Karlsruhe Germany "MCNP4,a Parallel Monte Carlo Implementation on a Workstation Network"

[5]佐藤三久,関口智嗣
並列計算機ｼｽﾃﾑのｽｹｰﾗﾋﾞﾘﾃｨについて
情報処理学会研究報告HPC-49-2,1993

[6]津田孝夫　培風館
モンテカルロ法とシミュレーション

[7](株)東芝　原子力技術研究所
"Paralleism in Continuous Energy Monte Calro Method for Neutron Transport"