

並列計算機を利用した遮蔽安全評価用  
モンテカルロコードMCACEの高速化

川添 明美、南 多善 (富士通)  
高野 誠、増川 史洋、内藤 倣孝 (原研)

原子力分野の複雑な幾何形状を扱う遮蔽計算には、モンテカルロ法による解析計算が有効である。しかし、遮蔽計算においては非常に多くの試行回数を要求され、処理能力の高いコンピュータを必要としていた。モンテカルロ法は各々の試行は互いに独立しており並列処理向きであると考えられ、近年の並列計算機の発達に対して、遮蔽計算の高速化への期待が集まっている。本論文では、遮蔽解析コードMCACEを富士通の分散メモリ型高並列計算機AP1000上へのインプリメントし及び、その性能評価について述べる。

Parallelization of MCACE,  
A Monte carlo for shielding analysis code.

Parallelization of Monte Carlo Shielding Analysis Code MCACE and Estimation of Parallel Efficiency By Fujitsu's shared memory processing highly parallel computer AP1000.

Akemi Kawazoe, Kazuyoshi Minami  
Nuclear Energy System Section. Scientific System Dept. Fujitsu Limited  
17-25, Sinkamata 1-Chome, Ota-ku, Tokyo 144, Japan

Makoto Takano, Fumihiro Masukawa, Yoshitaka Naito  
Fuel Safery Division. Japan Atomic Energy Research Institute  
Tokai-mura, Ibaraki-ken 319-11, Japan

The Monte Carlo Method is effective in shieling analysis of complicated geometrical shape in the field of nuclear energy. However, the shielding analysis requires so many random walk processes, and high performance computer. In these days, highly parallel processor have improved in it's performance and the shileding analysis using Monte Calro Method which calculates random walk processe independently are expected to be efficiently in there. This paper describes performance evaluations of MCACE, a shielding analysis code, which is implemented on a Fujitsu's distributed memory experimental highly parallel computer AP1000.

## 1. はじめに

遮蔽計算モンテカルロコードMCACE は任意の放射線量測定地点での放射線量を算出する為に、各放射線粒子について放射線源から空気・コンクリート壁等の遮蔽物中をランダムウォークする様子をモンテカルロ法を用いて追跡する。

遮蔽計算は測定値が信頼おける統計値に達するまで、非常に多くの粒子の試行を要する為、長時間の計算時間を必要とする。今回、遮蔽計算モンテカルロコードMCACE の並列計算機上で並列化を行い、プログラムの高速化を図った。

本報告では、分散メモリ型高並列計算実験機AP1000上で遮蔽計算モンテカルロコードMCACE の並列化と、その評価を行う。

## 2. 並列計算機AP1000の概要

AP1000は、数値計算等の高速実行と、並列ソフトウェアの研究環境を実現することを目的とした、分散メモリ型の並列計算機である。

AP1000のシステム構成を図2.1 に示す。AP1000は、最大1024台の汎用マイクロプロセッサ(セル)を使用したMIMD方式の並列計算機である。

AP1000は高速計算を実現する3種類のネットワークを持つ。第1にホストと全セルは、毎秒50メガバイトの転送能力を持つブロードキャストネットワークで接続され、ホストからセルへの放送を基本とした効率の良い通信が可能である。第2にセルとセルの間は毎秒25メガバイトの転送能力を持つ2次元トラスネットワークで接続され自動ルーティング機能により、隣接セルだけでなく遠隔セルとも高速に通信できる。第3に高速に同期を行うためのネットワークを持っている。

各セルは、25MHzで動作する32ビットRISCアーキテクチャのInteger UnitとFloating Point Unitを使用しており、セル単体でピーク性能15MIPS, 8.3メガFLOPS, メモリ16MBを持つ。

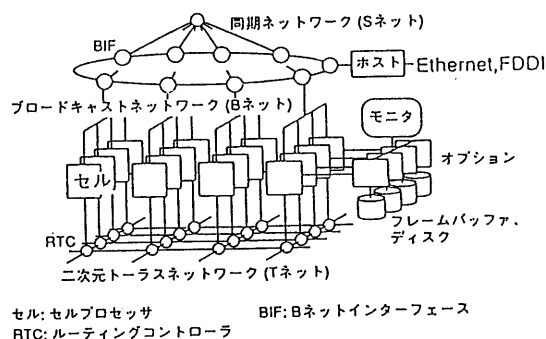


図2.1 AP1000 のシステム構成

## 3. 並列化方針

遮蔽計算モンテカルロコードMCACE は、各粒子のランダムウォーク追跡処理を幾つかまとめた『バッチ』と呼ばれる単位を作っている。『バッチ』は統計量が十分か確認する為の、標準偏差を

求める為にあり、物理的意味はない。オリジナルのMCACE の処理の流れ図を図3.1に示す。

オリジナルのMCACE の中で、どの部分に処理時間が消費されているか測定した結果、1バッチ200粒子で64バッチの時、全処理時間の98パーセントをバッチ処理が占めていた。各バッチは全く独立であり、各バッチ間でデータの依存性も無いためバッチ単位で各セルへ処理を動的に振り分ける方針で並列化を進めた。

並列化したMCACE の処理の流れを説明していく。流れ図を図3.2に示す。

[ホスト1： 初期処理]

まず、最初にホストは入力データの読み込み・各変数の初期化・各セルの乱数初期値生成等の初期処理を行った後、全セルに初期値データを送信する。

[セル： バッチ処理]

セルは初期値データを受信か、バッチ処理指示を受信すると、1バッチ計算をして計算結果をホストへ送信し、ホストからの次の指示を待つ。バッチ処理指示を受信すると1バッチ計算を行い、タスク終了指示を受信すると自タスクを終了させる。

[ホスト2： バッチ管理]

ホストはセルから計算結果を受信すると、まだ処理すべきバッチが残っている場合は、計算結果を送信してきたセルへバッチ処理指示を送信し、バッチ毎の集計や印刷を行う。バッチが残っていない場合は、ホストは、タスク終了指示を送信する。

[ホスト3： 後処理]

セルが終了した後、FLUX等の計算・印刷を行ってプログラムを終了する。

しかし、この方法ではセルはホストへ計算結果を送信した後必ずホストからの指示待ちが発生する。しかも、セル台数が多い時、[ホスト2：バッチ管理]でのホストのバッチ毎の集計や印刷が、セル側との通信にアイドルを生じさせる可能性が高い事が予想される。

そこで、ホスト側はバッチ数がセル台数の2倍以上ある時、最初に各セルへ2つのバッチ処理指示を与えておくアルゴリズムとした。これにより、セルは計算結果データをホストへ送信した後、自分の受信バッファにバッチ処理指示が入っていれば、次のホストからの指示を待つ事なく次のバッチ処理を開始でき、アイドル時間の大幅な減少が期待出来る。

[ホスト1： 初期処理 (追加)]

全バッチ数がセル数の2倍以上ある場合、初期値をセルへ送信した後に、無条件にセルへバッチ処理指示を送信する。

乱数は合同乗算法を使用している。乱数列の発生順に各セル番号目を各セルの種とし、セル台数読み飛ばしている。

図3.1 オリジナル MCACEの流れ図

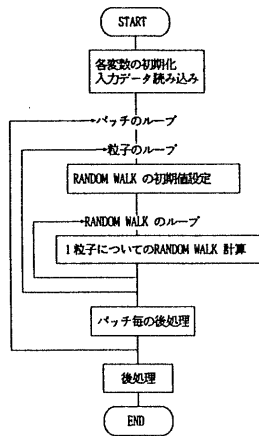
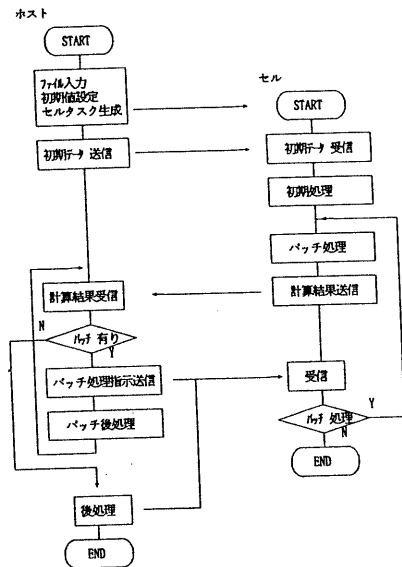


図3.2 並列化 MCACEの流れ図



4. 測定結果

AP1000の64セル構成を使用で測定を行った。問題は『<sup>60</sup>Coガンマー線源によるスカイシャイン問題』を使用した。粒子数の違うモデルの、セル数の増加による台数効果の変化を調べる為に、1バッチ200 粒子で128 バッチのモデルと、1バッチ400 粒子で128 バッチのモデルをセル台数 1, 2, 4, 8, 16, 32, 64 で測定した。

測定結果を表4.1、表4.2に示す。ここで、ホストの全処理時間と実行時間の差は、ホストの通信とアイドル時間である。

表4.1 計算時間 (バッチ数128 粒子数200)  
(単位: 秒)

| セル | ホスト    |          | セル (全セルの平均値) |        |          |
|----|--------|----------|--------------|--------|----------|
|    | 実行時間   | 全処理時間    | 実行時間         | 通信アイドル | 全処理時間    |
| 1  | 14.646 | 1460.531 | 1457.979     | 0.419  | 1458.398 |
| 2  | 14.675 | 733.688  | 726.312      | 5.263  | 731.575  |
| 4  | 14.630 | 367.591  | 363.398      | 2.089  | 365.487  |
| 8  | 14.663 | 187.772  | 182.501      | 4.245  | 186.746  |
| 16 | 14.648 | 95.781   | 91.376       | 2.282  | 93.658   |
| 32 | 14.550 | 51.454   | 45.843       | 3.511  | 49.354   |
| 64 | 14.851 | 29.142   | 22.229       | 4.242  | 26.572   |

表4.2 計算時間 (バッチ数128 粒子数400)  
(単位: 秒)

| セル | ホスト    |          | セル (全セルの平均値) |        |          |
|----|--------|----------|--------------|--------|----------|
|    | 実行時間   | 全処理時間    | 実行時間         | 通信アイドル | 全処理時間    |
| 1  | 14.911 | 2904.262 | 2901.449     | 0.124  | 2901.573 |
| 2  | 14.609 | 1453.872 | 1451.203     | 0.569  | 1451.772 |
| 4  | 14.848 | 732.517  | 723.119      | 7.020  | 730.139  |
| 8  | 14.642 | 369.682  | 363.585      | 3.957  | 367.542  |
| 16 | 14.400 | 188.979  | 181.883      | 4.981  | 186.864  |
| 32 | 14.984 | 100.169  | 91.401       | 6.662  | 98.063   |
| 64 | 14.289 | 55.286   | 45.842       | 7.358  | 53.200   |

## 5. 評価

並列化されたMCACE の高速化を評価する為、4章の結果から並列化効率の向上・抑制の原因について考察する。

### 5.1. 並列化効率

台数効果と並列化効率は次式で定義し、結果を図5.1、表5.1に示す。

$$\begin{aligned} \text{台数効果:} & S_p \equiv T_1 / T_p \\ \text{並列化効率:} & U_p \equiv S_p / P \times 100 \\ T_1: & \text{単一加セッサでの実行時間 (セル台数1)} \\ T_p: & P \text{ 個加セッサでの実行時間} \end{aligned}$$

200 粒子の時、セル64台の全処理時間が 1台の場合の50.13 倍で、並列化効率は78.3%であった  
また、400粒子の場合は、セル台数64の全処理時間は 1台の時の52.53 倍で、並列化効率は82.1 %であった。

図5.1 台数効果 バッチ数128

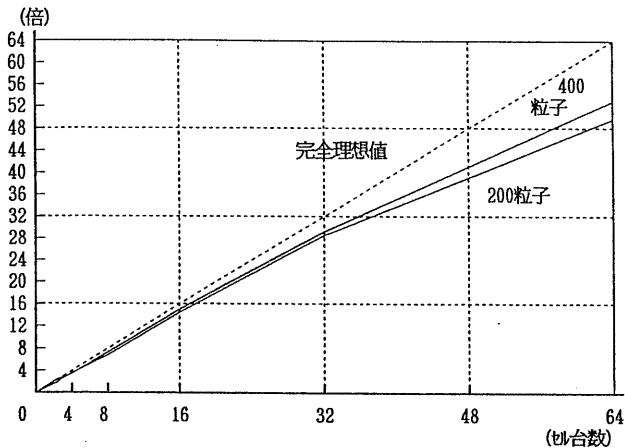


表5.1 台数効果 バッチ数128

(カッコ内は並列化効率) %

| セル | 200粒子        | 400粒         |
|----|--------------|--------------|
| 1  | 1.00 (100.0) | 1.00 (100.0) |
| 2  | 1.99 (99.5)  | 2.00 (99.9)  |
| 4  | 3.98 (99.3)  | 3.96 (99.1)  |
| 8  | 7.78 (97.2)  | 7.86 (98.2)  |
| 16 | 15.25 (95.3) | 15.37 (96.1) |
| 32 | 28.39 (88.7) | 28.99 (90.6) |
| 64 | 50.13 (78.3) | 52.53 (82.1) |

並列化効率を下げるものとして一般的に下記の要因が挙げられる。

- (1) ネットの逐次処理時間の、全処理時間に占める割合の増加
- (2) 並列化に伴う逐次計算にはなかった処理によるセル実行時間の増加
- (3) セルのアイドル時間の増加

これらの要因が MCACEの場合どの様に作用しているか粒子数200 の場合を基に検証した。

(1) ホストの逐次処理時間の、全処理時間に占める割合の増加

表5.2 理想台数効果

ホストの初期処理と全バッチ処理が終わった後の処理は逐次処理部分であり、台数が増えてもこの部分は変わらない。200 粒子 400粒子ともに初期処理部分は約 1.6秒であり、後処理は約 0.2秒で一定である。台数の増加と共にセル側の処理時間が速くなっているため、相対的にセル台数が増加するとホストの実行時間の全処理時間に占める割合は増加し、台数効果を下げる。この部分を差し引いた理想を次式で定義し、表5.2 に示す。

| セル数 | 200 粒子 |
|-----|--------|
| 1   | 1.00   |
| 2   | 2.00   |
| 4   | 3.99   |
| 8   | 7.93   |
| 16  | 15.71  |
| 32  | 30.82  |
| 64  | 59.40  |

1セルでの全処理時間

$$\left( (1 \text{セルでの全処理時間} - \text{ホストの逐次処理時間}) / \text{セル台数} \right) + \text{ホストの逐次処理時間}$$

表5.1と表5.2を比較すると、表5.1は理想の台数効果は出ていない。この要因を更に検証する。

(2) 並列化に伴う逐次計算にはなかった処理によるセル実行時間の増加

並列化する事によって逐次計算では必要無かった処理が必要となり、全セルで冗長計算が発生する。冗長計算は全セルが同じ計算を行うので逐次処理であり、従ってこの処理が大きいと台数効果を低下させる。

MCACE では、ホストから初期値を受信した後、受信データを各変数へセットする冗長計算がある。この部分にかかっている時間を測定すると0.162秒で一定であった。この時間はセル台数64の場合でも全処理時間の0.3%にすぎず、粒子数やセル台数にも左右されない。この事より、セルの逐次処理部分（初期処理部分）は十分に小さいと言える。

(3) セルの通信とアイドル時間の増加

表4.1より、セル台数2の時を除いて、セル台数の増加と共にセルの通信とアイドル時間が増加している。MCACEには、次の2種類のアイドル時間を含む通信がある。

- I セルがホストへ計算結果を送信した後ホストから次のバッチ指示が来るまでの通信時間とアイドル時間
- II セルがホストへ計算結果を送信した後、ホストが全セルの終了を待つて全セルへタスク終了指示を送信し、セルが受信するまでの通信とアイドル時間

上記の2種類の通信とアイドルの最大値を表5.3に示す。セル台数の増加と共にIの通信とアイドル時間が、減少している。これは、セル台数が増加すると

1セル当たりの処理するバッチ数が減少する為である。

一方、IIの通信とアイドルの原因を考える。表5.3では各バッチ処理時間のバラツキを示しているが、これから明らかな様に各バッチのバッチ処理のバラツキは第2の通信とアイドルの大きな原因である。このバラツキはモンテカルロ法の乱数の違いによる計算時間の違いであり、原理的に変える事の出来ないものである。

64セルの場合のみ、IIとアイドルからバッチ処理のバラツキを引いた差が増加しており、この差が全処理時間の5%を占めている。これは、ホストがバッチ毎に行う後処理がセル側のバッチ処理に隠れきれないでアイドル時間に陽に現れた時間であると考えられる。

従って、粒子数とバッチ数を一定にしてセル台数を増加させた場合、ホストのバッチ毎の後処理が並列化効率を低下させる要因になる。

表5.3 1バッチの終了待ちによる通信とアイドル時間バッチ処理のバラツキ (単位:秒)

| 200 粒子 128バッチ |                   |        |          |          |         |       |
|---------------|-------------------|--------|----------|----------|---------|-------|
|               | 通信とアイドルの最大値       |        | バッチ処理時間  |          |         |       |
|               | 計算結果送信<br>〜バッチ処理受 | 終了指示待  | 平均値      | 最大値      | 最小値     | 差     |
| 1             | 0.304             | 0.115  | 1457.817 | 1457.821 | ——      | ——    |
| 2             | 0.151             | 10.106 | 726.150  | 731.146  | 721.154 | 9.992 |
| 4             | 0.074             | 3.640  | 363.236  | 365.137  | 361.607 | 3.530 |
| 8             | 0.037             | 5.388  | 182.339  | 185.306  | 180.032 | 5.274 |
| 16            | 0.017             | 5.541  | 91.214   | 93.361   | 89.338  | 4.023 |
| 32            | 0.007             | 7.736  | 45.681   | 48.502   | 41.096  | 7.406 |
| 64            | 0.003             | 7.208  | 22.067   | 25.580   | 19.870  | 5.710 |

## 6. 結論

今回、各粒子のランダムウォーク追跡処理を幾つかまとめた単位である「バッチ」を、各セルへ動的に振り分ける方針で並列化を進めた。

計算時間の評価結果は、400粒子128バッチセル64台で実行した場合、55.29秒で、計算速度向上率がセル1台の時に比べて52.53倍を示した。

各セルへのバッチの振り分けについては、バッチ数がセル台数の2倍以上ある時予め2バッチの処理をセルへ与え、セル側にアイドルが発生しない様なアルゴリズムを採用した。この結果、セルがバッチの計算結果をホストへ送信後、次のバッチの処理に入るまでの通信とアイドル時間が非常に小さく、バッチを各セルへの無駄なく動的に割り当てる事が実現した。

又、セル64台の時のセルのアイドル時間は殆どバッチ処理時間のバラツキに等しく、アイドル時間はバッチ単位でセルに分担させる今の手法ではほぼ極限まで下げる事に成功した。

粒子数を増加させると、各バッチのバラツキが小さくなり、結果としてこのアイドルは小さくなる

。200 粒子の時よりも 400粒子の時の方が、台数効果が大きいのはこの為である。  
従って、今後さらに大きな粒子数のモデルを計算させる時、並列化効率は下がる事はない。

将来、更に高並列構成の計算機でMCACE の検証に行く計画であるが、セル台数が非常に増加した場合、次の2点を考察する必要がある。

- (1) 今回ホスト側での各バッチ単位の処理の殆どはセル側の処理時間に隠れて陽に表れなかった。この部分の処理時間はバッチ数にのみ依存するので、セル台数が増えバッチ数が増えた場合、この部分が台数効果を低下させる事が予想される。
- (2) 小さな問題を計算する場合、バッチ処理のバラツキによるアイドル時間の占める割合が大きくなる。これを防止するには粒子単位でセルに振り分ければよいが、今度はホスト・セル間通信の増加によるアイドル時間が発生すると予想される。

この2つの問題を考慮して、各セルの処理の進捗状況によってセルへの処理分配を動的に行う手法を検討していく計画である。

#### 謝辞

株式会社 富士通研究所 システム研究部 第二研究室 池坂 守夫氏に、並列化手法等について御指導頂きました。深く感謝しております。

#### 参考文献

- 1) N.Yamano, K.Minami, K.Koyama, and Y.naito; "RADHEAT-V4:A Code System to Generate Multigroup Constants and Analyze Radiation Transpot for Shielding Safety Evaluation", Japan Atomic Energy Research Institute 1316, March 1989
- 2) Straker E.A., et.al.; "The MORSE-Code with Combinatory Geometry" DNA-2860T(1972)
- 3) H.Ishihata, T.Horie, S.Inano, T.Shimizu, and S.Kato; "An Architecture of Highly parallel Computer AP1000" IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, May 9-10, 1991