5E－7

粒子輸送モンテカルロコードの高並列処理
－EGS4コードの並列化－

武宮 博，本間 一朗，太田 浩史
日本原子力研究所　計算科学技術推進センター

1. はじめに

粒子輸送モンテカルロ・コード（以下，MCコードと呼ぶ）は，十分に精度の良い解を得るために多大の計算時間を必要とするため，並列処理による効率的な計算が望まれている．しかし，効率的な並列化を実現するためには，通信コストの低減，ロードインバランスの解消など，検討すべき課題は多い．

本稿では粒子輸送モンテカルロ・コードEGS4コードを対象として，分散メモリ型スカラ並列計算機上での効率的な並列実行を実現するための並列化手法及び性能評価結果について述べる．

2. 粒子輸送MCコードのスカラ並列処理

EGS(Electron Gamma Shower)4コードはSLAC(Stanford Linear Accelerator Center)において開発されてきた電磁カスケードシミュレーション用の汎用MCコードである[1]．我々は，EGSコードを種々の分散メモリ型並列計算機上で効率的に並列実行させることを目的としている．

粒子輸送問題を解くモンテカルロ計算では，各粒子に対する計算が独立に行われるため，スカラ並列計算の場合，あらかじめ各プロセッサに粒子を分配し(静的粒子分配)，計算を行うことが考えられる．しかし，

(1)粒子輸送MCシミュレーションには，各粒子に対する計算時間が大きく異なる問題が多い．

(2)計算性能の異なる計算機が結合したWSクラスタ上で実行される場合も想定される．

そのため，静的粒子分配手法では必ずしも効率的な並列実行が実現されるとは限らない．従って，各プロセッサ間の計算時間の違いに基づく全体の計算時間の延長(ロードインバランスコスト (H_{LI}) と呼ぶ) を低減するために，マスタースレーブモデルに基づく動的粒子分配手法の採用が望ましい．その場合，粒子の分配を行うマスタプロセスと粒子計算を行うスレーブプロセス間に通信が発生するため，通信コスト (H_{COM}) も低減する必要がある．

H_{LI}，H_{COM}の低減には，次の2点が重要である．

(1)H_{LI}，H_{COM}は，一般にプログラムにおける1粒子あたりの計算時間，計算機の演算性能，通信性能に依存するため，プログラムや実行環境で個々に異なる．従って，粒子分配手法には，これらの要因に応じて自動的にH_{LI}，H_{COM}を調節する機構が含まれることが望ましい．

(2)H_{LI}，H_{COM}を独立に低減することはできない．H_{LI}を低減するためには，分配粒子数(タスクサイズと呼ぶ)を小さくする必要があるが，タスクサイズを小さくすると分配回数が増大し，H_{COM}が増大するためである．従って，$H_{LI}+H_{COM}$を低減するようにタスクサイズを決定する必要がある．

上記2点を考慮したタスクサイズ決定手法を以下に述べる．

3. タスクサイズ決定手法

(1), (2)を解決するために，今回考案した手法では，各スレーブプロセスが粒子計算途中における通信時間，計算時間を用いて自動的にタスクサイズを決定する．すなわち，各スレーブプロセスはタスクの処理毎に通信時間と計算時間を測定し，そのデータを基に予想されるH_{LI}とH_{COM}からコスト関数$H=H_{LI}+H_{COM}$を最小とするような粒子数を決定する．

ここで，図1のようにHの評価に必要な幾つかの量を定義する．まず，j番のスレーブプロセスにおいてk回目の最適タスクサイズの評価に利用されるタスクはN_{jk}個の粒子計算から構成されるものとし，そのタスク処理に要した計算時間をT_{jk}^{calc}とする．また，粒子計算に用いられる粒子数は全体でN^{total}個であるとする．マスタプロセスとの通信コストはT_{jk}^{com}とする．

H_{LI}は動的粒子分配手法を採用していることから高々1タスク計算時間程度である．各スレーブプロセスは，自分が最終計算終了プロセスになると仮定した場合に，どの程度のばらつきが生じるかを見積もる．最終計算におけるタスクサイズを$N_{j\ last}$とすれば，

H_{LI}～最終タスク計算時間

$$\sim N_{j\ last}\cdot\frac{\sum_k T_{jk}^{calc}}{\sum_k N_{jk}}$$

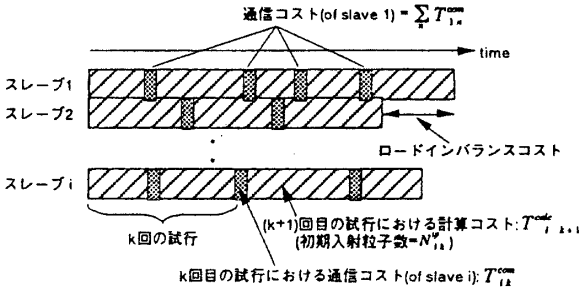

図1. タスクサイズ決定に用いられる諸量

Parallel Processing of Monte Carlo Code for Particle Transport Problem -Parallelization of EGS4 Code-
Hiroshi TAKEMIYA, Ichirou HONMA, Hirofumi OHTA
Center for Promotion of Computational Science and Engineering Japan Atomic Energy Research Institute

一方，H_{COM}は，それまでに要した通信コスト(H^1_{COM})と今後計算終了までに見込まれる通信コスト(H^2_{COM})の和であり，

$$H^1_{COM}=\sum_k T^{com}_{jk}$$

H^2_{COM}〜通信1回あたりの平均通信コスト・通信回数と表される．k回目の評価時点における自分が今後計算終了までに処理する粒子数 = N^{rest}_{jk}とすれば，通信回数は，

$\sum_{i=1}^{n^{com}} N_{j\,k+i}=N^{rest}_{jk}$を満足する$n^{com}$である．

ここで，$N_{j\,k+i}$を$N_{j\,last}$に重み付けをしたもの，すなわち$N_{j\,k+i}=g_{k+i}\cdot N_{j\,last}$とすることによって，通信回数は$N_{j\,last}$の関数として計算可能である．$g_{k+i}$の与え方には様々な戦略が可能であるが，

(1)要求粒子数の計算はタスク計算毎に行われるため，通信回数の算出にあまり複雑な計算が含まれると要求粒子数の計算コスト自体が大きくなってしまう．

(2)n^{com}はあくまでも予想であり不定性が存在するため，予測と実際にずれが生じた場合，重み付けはより大きなロードインバランスを生んでしまう，

この二点を考慮して，本手法では$g_{k+i}=1$として通信回数を評価することにした．

この場合，通信回数は$(N^{rest}_{jk}/N_{j\,k+i})$で表される．$N^{rest}_{jk}$はk回目の評価における全体処理粒子数と自分がそれまでに処理した粒子数を基に予測する．k回目の評価の時点で，粒子は全体として$\sum_j\sum_k N_{jk}$個処理されており，自分は$\sum_k N_{jk}$個処理していることから，

$$N^{rest}_{jk}=\frac{N^{total}\cdot\sum_k N_{jk}}{\sum_j\sum_k N_{jk}}-\sum_k N_{jk}$$

よって，

$$H^2_{COM}\sim\frac{\sum_k T^{com}_{jk}}{k}\cdot\left(\frac{N^{total}\cdot\sum_k N_{jk}}{\sum_j\sum_k N_{jk}}-\sum_k N_{jk}\right)$$

以上の議論より，コスト関数Hは

$$H=A\cdot N_{j\,k+1}+B+C/N_{j\,k+1}$$

$$A=\frac{\sum_k T^{calc}_{jk}}{\sum_k N_{jk}},\quad B=\sum_k T^{com}_{jk},$$

$$C=\frac{\sum_k T^{com}_{jk}}{k}\cdot\left(\frac{N^{total}\cdot\sum_k N_{jk}}{\sum_j\sum_k N_{jk}}-\sum_k N_{jk}\right)$$

この関数を最小とするように$N_{j\,k+1}$を決定すればよい．これより，P =B/2A, Q = C/Aとすれば求めるタスクサイズは

$$N_{j\,k+1}=P\cdot\left(1+\sqrt{1+Q/P}\right)$$

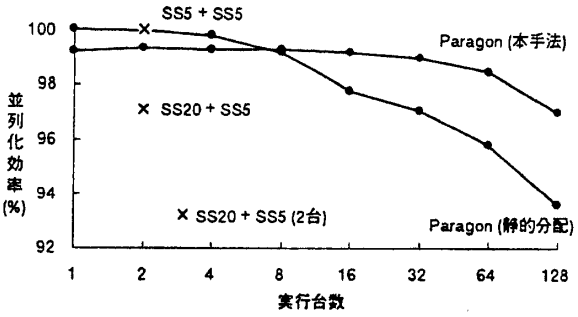

図2. Paragon及びWSクラスタにおける並列実行効率

仮に1粒子計算時間及び通信時間が常に一定であったとすれば，要求粒子数 = (平均計算時間/通信コスト) = (計算終了時点におけるP) が成立する．

4. 性能評価

性能評価のために，上記のタスクサイズ決定手法及び静的均等粒子配分手法をIntel Paragon及びWSクラスタ上に実装し，試走を行った．対象には平行ウラン平板に対する電子の入射問題を用いた．電子の入射エネルギーは0.1MeV, 粒子数は10,000個である．

図2にIntel Paragon及びWSクラスタを用いて並列実行を行った場合の並列化効率を示す．WSクラスタはSUN SS20(1台), SS5(2台)によって構成されている．WSクラスタでは，並列実行効率の算出において基準となる1台での実行時間として，各WSでの逐次実行時間の調和平均を用いた．

Paragonにおいては，プロセッサ台数≧8において本提案手法が静的均等粒子分配手法よりも並列化効率において上回っている．これは，台数が増加したために，静的均等粒子分配におけるロードインバランスが増加したことが原因である(128台使用時に，静的均等分配の場合実行時間の12%．本手法の場合，2.7%)．

WSクラスタにおいては，異なるWSの組み合わせに対しても90%以上の並列化効率が実現されている．ただし，SS20が含まれる並列実行に関しては，並列実行の低下が激しい．WSクラスタの場合，通信コストとロードインバランスコストの和が計算時間全体の1%以下であったことから，予想を大きく上回る低下である．これは，SS20のキャッシュが大きく(1MB)，逐次計算時にはキャッシュ上にあった粒子計算用データが，並列実行の際は要求粒子数計算の際にパージされてしまったためである．

5. まとめ

粒子輸送MCコードの局所メモリ型スカラ並列計算機上での高並列実行に関して，動的なタスク分配手法を考案し，Intel Paragon, WSクラスタ上で性能評価を行った．

参考文献

[1] Nelson, W. et al.; The EGS4 Code System, SLAC-265 (1985)