数 値 解 析 37－3
（１９９１．７．１７）

ＤＳＭＣ法における並列処理の適用

渡辺 健二，鈴木 孝一郎（富士通）

横川 三津夫，山本 浩康，蕪木 英雄（日本原子力研究所）

Abstract

直接シミュレーション・モンテカルロ法（ＤＳＭＣ法）は，希薄気体から連続流体に近い流れまで，幅広い領域の流れのシミュレーションに有効な手法である．しかし，多数の模擬分子を取り扱う場合には，非常に長い計算時間とメモリを必要とする．本報告では，ＤＳＭＣ法における効率的な並列処理の手法とその性能評価の結果について述べる．分散メモリ型高並列計算機ＡＰ１０００でのキャビティ流れの解析において，プロセッサ６４台を使用した場合，１台での処理時間と比較して約４２倍の速度向上が得られた．

Parallel processing for the Direct Simulation Monte Carlo method

Kenji Watanabe, Koichiro Suzuki
Fujitsu Limited
Shinkamata, Ota-ku, Tokyo 144, Japan

Mitsuo Yokokawa, Hiroyasu Yamamoto, Hideo Kaburaki
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan

Abstract

The Direct Simulation Monte Carlo (DSMC) method is a powerful technique for simulating a wide range of flows from rarefied to near continuum region. Since a large number of simulated particles have to be treated in the DSMC method, a lot of computer resources, CPU time and memories, are required in the simulation. In this note, an effective way of parallelization for the DSMC method and its evaluation results are presented. For the cavity flow analysis in a rarefied region, a speedup of 42 times using 64 processors is achieved on a highly parallel computer Fujitsu AP-1000 with distributed memories.

1．はじめに

直接シミュレーション・モンテカルロ法（Direct Simulation Monte Carlo method：以下DSMC法）は，実在気体の流れを計算機上の模擬分子によりモデル化する数値シミュレーション手法であり，希薄気体流れの解析に有効である．近年の計算機処理能力の向上に伴い，ＤＳＭＣ法は連続体に近い流れに対しても有力な解析手法となってきた．しかしながら，ＤＳＭＣ法は多くの模擬分子を取り扱う場合，非常に長い計算時間を必要とするためプログラムの高速化が必要となる．ここでは，ＤＳＭＣ法の並列計算機上での高速処理を目指して，並列化のための処理手順を検討し，プログラムの高速化を図った．

本報告では，ＤＳＭＣ法における並列処理の手法を検討し，分散メモリ型高並列計算機ＡＰ１０００を用いた性能評価の結果について述べる．

2．ＤＳＭＣ法

近年の飛躍的な計算機能力の向上に伴い，乱数を用いて確率的に問題を解くモンテカルロ法が有力なシミュレーション手段となっている．希薄気体流れの解析においても，乱数によって確率的に問題を解くＤＳＭＣ法が用いられている[1,2]．この方法では，気体を構成する各分子の運動を模擬した分子を追跡することによって流れ場を決定する．ここでは，アボガドロ数個のオーダーの分子の運動を，数千から数万個の模擬分子で代表させ，流れ場における物理量の統計平均を求める．例えば，ある微小体積にある模擬分子の速度の平均がその位置における流体の速度を与える．

ＤＳＭＣ法は，“分離の原理”に基礎を置いている．この原理に基づけば，シミュレーションの時間ステップΔｔを，平均自由時間（分子が平均自由行程を走るのに要する時間）より十分小さくとれば，分子の移動と衝突を分けて扱うことが出来る．通常のＤＳＭＣ法では，分子の衝突を行わせる衝突セルは，均一な大きさに固定されている．しかし，本来衝突セルはその場における平均自由行程λ程度の大きさに取ることが理想的である．そこで，今回用いたプログラムでは，蕪木らによって開発された，衝突セルの大きさを自動的にλ程度に設定する手法を用いている[3]．

初期設定では，模擬分子の初期位置，初期速度，境界条件，衝突を行わせる衝突セル，物理定数，乱数の初期値等を設定する．衝突セルの大きさは，平均自由行程程度になるようにプログラムが自動的に設定する．

時間発展ループでは，以下の手順に従って処理を行う．まず各計算領域で平均自由行程を計算し，それに基づいて衝突セルを設定する．与えられた各分子は，その座標に応じて衝突セルに割り当てる．次に各衝突セル内で，分子を確率的に衝突させる．この衝突過程は，各衝突セルで独立に計算することが出来る．それから，各分子の持っている速度でΔｔだけ移動させる．移動の際，境界と干渉した分子については，境界条件に従って速度と座標を更新する．

定常状態の解析においては，場の状態が定常に達した後，適当な時間間隔でサンプリングを行い，場の速度や数密度などを求める．サンプリングの時間間隔は，前後のサンプリング値に相関がなくなる程度に大きく取ることが理想的である．得られたサンプリング値の時間平均を取ることによって定常解が得られる．

ＤＳＭＣ法による定常計算のシミュレーションのアルゴリズムを Fig. 2.1 に示す．

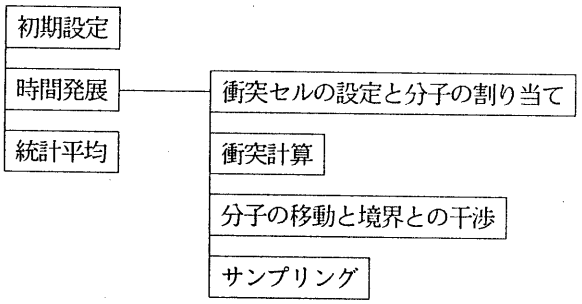

Fig. 2.1　ＤＳＭＣ法の基本アルゴリズム

3．並列計算機ＡＰ１０００

ＡＰ１０００は，富士通研究所によって開発された，分散メモリ型・ＭＩＭＤ方式の高並列計算実験機である[4,5]．ＡＰ１０００のシステム構成を Fig. 3.1 に示す．ＡＰ１０００は，１台のフロントエンド・プロセッサ（ホストと呼ぶ），最大１０２４台の汎用マイクロプロセッサ（セルと呼ぶ），そして３種類の高速ネットワークから構成されている．

ホスト上では，並列化プログラムの開発，ホストプロセスの実行等を行う．

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 を使用している．セル単体の仕様は，ピーク性能１５ＭＩＰＳ，１２．５ＭＦＬＯＰＳ．メモリ１６ＭＢである．ただし，今回測定に使用した計算機のピーク性能は，１５ＭＩＰＳ，８．３ＭＦＬＯＰＳである．

ネットワークには，ホスト－セル間を接続するブロードキャスト・ネットワーク，セル－セル間を接続する２次元トーラス状ネットワーク，そして同期をとるためのシンクロナイゼーション・ネットワークの３種類がある．転送能力は，ホスト－セル間では毎秒５０ﾒｶﾞﾊﾞｲﾄ，セル－セル間では毎秒２５ﾒｶﾞﾊﾞｲﾄである．また，セル－セル間では，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することが可能である．

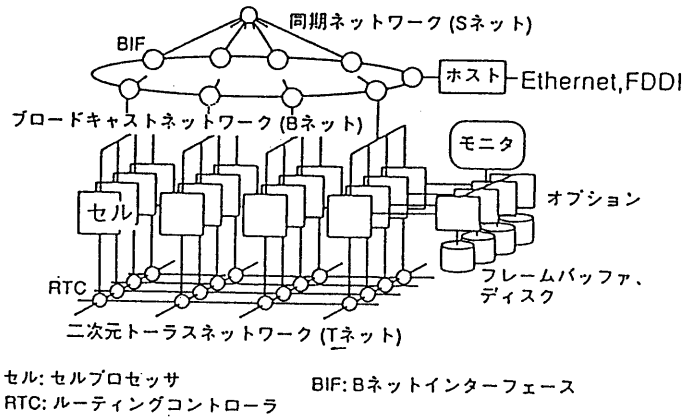

Fig. 3.1　ＡＰ１０００のシステム構成

4．並列処理の手法

アプリケーション・プログラムの並列処理を行う場合には，そのプログラムに内在する並列性を引き出すことが重要となる．また，使用する並列計算機のアーキテクチャを有効に利用するようなプログラミングを行うことが必要となる．ＤＳＭＣ法の大きな特徴は，分子間衝突の計算が各衝突セルで独立に行えることである．すなわち，ＤＳＭＣ法には衝突セルを単位とする並列性が存在する．また，分散メモリ型計算機の場合，メモリ間のデータの転送量を出来るだけ小さくする方が効率が良い．すなわち，同じ衝突セルに属する分子に関するデータは，同じプロセッサのメモリ上に格納されている方が効率が良い．これらのことから，ＤＳＭＣ法の並列処理を行う場合には，衝突セルを並列処理の単位として各プロセッサに割り当てる手法が有効であると考えられる．以下に具体的な処理の流れを示す．

〔ステップ０：初期設定〕

まず計算領域をプロセッサ数に等分する．分子もやはりプロセッサ数に等分し，各プロセッサに割り当てる．各分子の座標や速度の初期値は，各プロセッサで並列に計算する．

〔ステップ１：衝突セルの設定と分子の割り当て〕

次に，各プロセッサでローカルな平均自由行程λを計算し，ちょうどλ程度の大きさになるように衝突セルを設定する．各分子の座標から，その分子がどの衝突セルに所属するかを計算し，衝突セルの順序に並べ変える．

〔ステップ２：衝突計算〕

各衝突セルにおいて衝突計算を行う．衝突の計算にはいくつかの手法が提案されているが，今回のプログラムでは修正南部法を用いている[6]．この手法では，各分子についてまず衝突のパートナーを乱数を用いて決定し，相対速度と衝突確率を計算する．得られた衝突確率と乱数を比較し，衝突するかどうかを判定する．衝突する場合には，運動量とエネルギーが保存するように衝突を行わせ，注目している分子についてのみ速度を更新する．

〔ステップ３：分子の移動と境界との干渉〕

分子の速度が決まったら，まずΔｔだけ直線的に移動させる．その際，境界と干渉した分子については，境界条件に従って速度を更新し，再び移動させる．

〔ステップ４：プロセッサ番号の計算とデータの転送〕

分子の座標が決まったら，所属すべきプロセッサ番号を計算する．もし，現在所属しているプロセッサ以外のプロセッサに所属すべき座標にある分子については，その分子に関するデータを所属すべきプロセッサに送信する．転送すべき全てのデータの送信が終わったら，今度は自分宛てに他のプロセッサから送られてくるデータを受信する．送信と受信の際には，それぞれ全てのプロセッサの処理が終わるまで同期を取る．全ての通信が終わったら，ステップ１に戻り，次のタイムステップの計算を行う．

〔ステップ５：サンプリング〕

決められたタイムステップ毎に，サンプリングを行う．サンプリングを行う物理量は，分子数，数密度，速度ベクトルなどである．この処理も各プロセッサで並列に行う．

〔ステップ６：統計平均〕

全てのタイムステップの計算が終わったら，サンプリングしておいた物理量の時間平均を取る．この処理も各プロセッサで並列に行う．

5．並列処理の結果

高並列計算機ＡＰ１０００を用いてＤＳＭＣ法プログラムの並列処理を行い，並列処理の効率等の評価を行った．今回，解析の対象とした問題は，クヌーセン数（λ／Ｌ；Ｌはキャビティの一辺の長さ）０．０４２のキャビティ流れで，初期条件は温度２８３〔Ｋ〕，圧力０．００２〔Ｐａ〕，壁の速度は最確速度の０．６倍である．また，全体の模擬分子数４４８００個，サンプリング回数２０回，全体のタイムステップ数１０００ステップである．並列処理の効率をみるため，プロセッサ数を１台から６４台まで変えて処理時間を測定した．プロセッサ台数（ＰＥ数）と並列処理効率との関係を次の３つの式によって評価する．

$$P_n = Tp / Tt$$
$$S_n = T_1 / T_n$$
$$U_n = (T_1 / n) / T_n$$

ここで，

P_n ： 並列化率
S_n ： 処理速度向上率
U_n ： 並列化効率
Ｔｐ： 並列処理を行っている部分の処理時間
Ｔｔ： 全体の処理時間
T_1 ： プロセッサ１台のときの処理時間
T_n ： プロセッサｎ台のときの処理時間
ｎ ： プロセッサ台数

である．測定および評価の結果を Table 5.1に示す．

Table 5.1 ＡＰ１０００による測定結果

PE数	CPU 時間 (sec)	Pn (%)	Sn (倍)	Un (%)
1	2951.0072	99.90	1.00	100.00
2	1486.9601	99.78	1.98	99.23
4	764.5946	99.68	3.86	96.49
8	390.2234	99.35	7.56	94.53
16	204.5960	98.61	14.42	90.15
32	111.2066	97.49	26.54	82.93
64	70.3504	94.10	41.95	65.54

また，処理速度向上率S_nをグラフ化したものを Fig. 5.1 に示す．

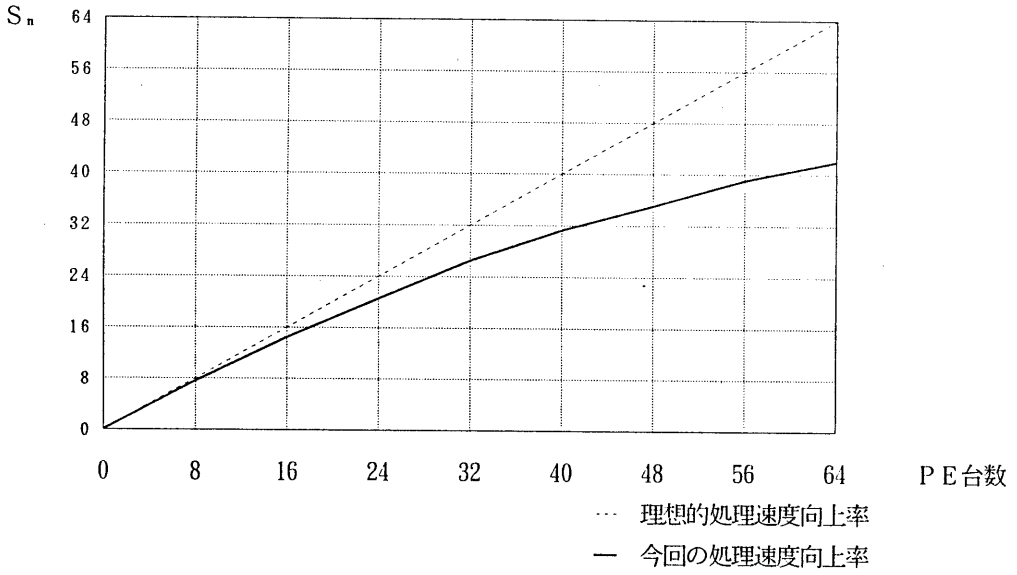

Fig. 5.1　処理速度向上率S_n

6．結果の検討

5節の結果より，プロセッサ台数が16台程度までは90%以上の並列処理効率が得られている．しかし，プロセッサ数がそれ以上になると，並列処理効率が低下することがわかる．

並列処理を行っている各プロセッサの処理時間は，大別して，演算時間，通信時間，アイドル時間の3つの要素から成り立っている．今回のDSMCコードにおいて，それぞれがどのような分布になっているかを調べた．測定結果を Table 6.1に，また，分布の様子を Fig 6.1 に示す．ただし，ここでの通信時間とは，通信のための前処理，後処理を含んだ時間である．

これらの結果より，プロセッサ数が増加したとき，演算時間は十分に高速化されているが，そのために逆に全体に占める通信およびアイドル時間の比率がふくらんで見えてきていることがわかる．通信およびアイドル時間の比率の増加には，種々の要因が考えられるが，最も大きなものは負荷のばらつきであると考えられる．これは，現在のプログラムが領域分割による並列化を行っているため生じる問題である．すなわち，領域分割による並列化の場合，各プロセッサが担当する粒子数にはいくらかのばらつきが生じる．並列度が高くなると，プロセッサ1台あたりの粒子数が減少するため，ばらつきの影響が大きくなってくると考えられる．実際に，粒子数のばらつきを測定してみたところ，64プロセッサの場合，40%程度のばらつきのあることがわかった．これを回避するためには，並列度が高くなっても負荷が均一に保たれるような並列化の手法，例えば粒子分割による並列化などが必要になるであろう．

Table 6.1　演算・通信・アイドル時間の測定結果

PE数	演算時間 (sec)	通信時間 (sec)	ｱｲﾄﾞﾙ時間 (sec)
1	2911.8645 (98.02%)	58.3596 (1.96%)	0.6092 (0.02%)
2	1441.2873 (96.39%)	33.1106 (2.22%)	20.7994 (1.39%)
4	710.2369 (91.58%)	19.1980 (2.48%)	46.0713 (5.94%)
8	346.6751 (86.62%)	13.3968 (3.35%)	40.1334 (10.03%)
16	167.8373 (83.42%)	10.1755 (5.06%)	23.1778 (11.52%)
32	81.0625 (75.94%)	9.8438 (9.22%)	15.8438 (14.84%)
64	40.3169 (59.50%)	14.9013 (21.99%)	12.5387 (18.51%)

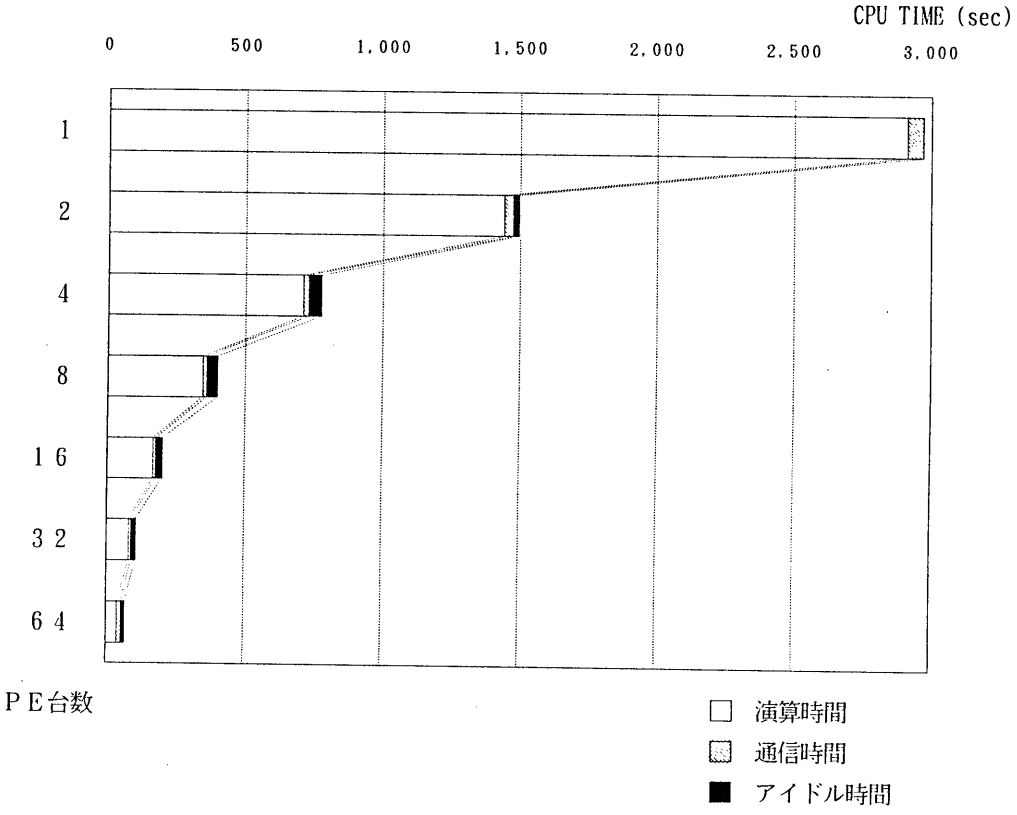

Fig. 6.1　演算・通信・アイドル時間の分布

7．まとめ

ＤＳＭＣ法による希薄流れ解析プログラムについて，並列処理のための手法を検討し，分散メモリ型高並列計算機ＡＰ１０００上で性能評価を行った．キャビティ流れの解析において，プロセッサ６４台を使用した場合，１台での処理時間と比較して，約４２倍の速度向上が図られた．今回用いた領域分割による並列化の手法では，プロセッサ数が１６台程度までの並列処理では，９０％以上の並列化効率が得られるが，それ以上になると並列化効率が低下する．これは，並列度が高くなるとプロセッサ１台あたりの粒子数が減少するため，負荷のばらつきの影響が大きくなるためであると考えられる．

これより，１６台程度の並列処理では，比較的容易な領域分割による並列化で十分な性能が得られるが，１０００台以上の高並列処理を行う場合には，より進んだプログラムの並列化を行わなければ高い性能は得られないことがわかった．

8．謝辞

本研究の機会を与えて頂きました，日本原子力研究所　情報システムセンタ室長　秋元正幸氏に感謝致します．また，富士通（株）の　南多善氏には，今回の作業全般に渡り，貴重な助言を頂きました．感謝致します．

9．参考文献

1）G.A.Bird;　Molecular Gas Dynamics, Oxford University Press (1976)

2）K.Nanbu;　"Theoretical Basis of the Direct Simulation Monte Carlo Method", Rarefied Gas Dynamics, Proc. of the 15th International Symposium, Vol-1, pp.369-383 (1986)

3）H.Kaburaki, H.Yamamoto, M.Yokokawa, and T.Arisawa;　"A New Collision System for the Direct Simulation Monte Carlo Method", Proc. of 1st International Conference on Supercomputing in Nuclear Applications, pp.51-56, March 12-16 (1990)

4）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, pp.13-16, May 9-10 (1991)

5）T.Horie, H.Ishihata, T.Shimizu, S.Kato, and M.Ikesaka; "AP1000 Architecture and Performance of LU Decomposition", International Conference on Parallel Processing (submitted)

6）H.Floss;　"On Simulation Methods for Solving Boltsman Equation", Computing 38, pp.101-115 (1987)