領域分割法による MAC 法の効率的な並列計算アルゴリズム

An Efficient Parallel Computation Algorithm of MAC Method Using Domain Decomposition Method

○ 黒川 原佳，北陸先端大院, 石川県能美郡辰口町旭台 1-1, E-mail : kurokawa@jaist.ac.jp
松澤 照男，北陸先端大，石川県能美郡辰口町旭台 1-1，E-mail : matuzawa@jaist.ac.jp
姫野 龍太郎，理研，埼玉県和光市広沢 2-1，E-mail : himeno@postman.riken.go.jp
重谷 隆之，理研，埼玉県和光市広沢 2-1，E-mail : shige@postman.riken.go.jp
Motoyoshi Kurokawa, JAIST, 1-1 Asahi-dai Tatsunokuchi Ishikawa, JAPAN
Teruo Matsuzawa, JAIST, 1-1 Asahi-dai Tatsunokuchi Ishikawa, JAPAN
Ryutaro Himeno, RIKEN, 2-1 Hirosawa Wako Saitama, JAPAN
Takayuki Shigetani, RIKEN, 2-1 Hirosawa Wako Saitama, JAPAN

The performance of a parallel Computational Fluid Dynamics(CFD) MAC solver using domain decomposition method is strongly dependent on the partitioning patterns. It is diffecult to select the better partitioning pattern. In this paper, we show an evaluation method for selecting the better partitioning pattern. Experimental results are in good agreement with evaluated values.

1. はじめに

静的領域分割法を用いて並列計算を行う場合，格子形状や使用する Processing Element(PE) 数そして用いるハードウェアによってどのように領域分割を行うかが問題となる．これらの問題は，従来プログラマの経験によって行ってきたのが現状である．しかし，並列計算に不馴れなプログラマでも，効率が良い領域分割パターンが機械的に分かる方法が必要である．また，分割次元が一次元以外では，分割過程が煩雑になる問題がある．分割が一次元でも性能が得られるような場合には，分割次元を落とすことが可能となる．

本稿では，IBM RS/6000 SP (PowerPC 332MHz，SP-Switch Network : 150MB/s) を用いて，使用する PE 数が固定の場合に領域分割パターンを容易にかつ機械的に予測する方法を提案する．そして，その方法によって得られた分割パターン毎の性能予測値と実際に流体計算を行って得られたパフォーマンスを示し，予測方法が有効であることを示す．

2. 領域分割法

領域分割法を考える場合，使用する PE 数が多くなればなるほど分割パターンは飛躍的に増加する．そして，分割パターンのみならず，格子の分割数が三次元空間の各方向で大きく異なる場合，どの方向に格子の分割数が多い軸を持ってくるかも問題となる．すなわち，使用する PE 数に対する分割パターンの 3 倍の可能性を考慮する必要がある．

分割に伴う通信手順は，各分割次元に対して 1 組の送受信を 2 回行う必要がある．

2.1 分割パターン

従来使用する PE 数が増え，分割数が増える場合，通信量が少なくなるような分割を用い，通信処理の影響を最小にするという戦略で領域分割を実行することが多く行われてきた[1]．しかし，分割パターンは，データ通信量やループ長の違いによる計算性能の両方から全体性能に影響を及ぼす[2]．三次元領域の分割パターンは，PE 数が増えるにつれて多数存在する．例えば 8 PE 用いるために 8 領域に分割する場合を図 1 に示した．

図 1 中のかっこ内の数字は，各座標軸での分割数である．例えば，図の左の分割パターン (8,1,1) は，i 方向に 8 分割，j 方向に 1 分割 (つまり分割なし)，k 方向に 1 分割したことを示す．分割パターンは，一次元分割で 3 パターン {(8,1,1),(1,8,1),(1,1,8)}，二次元分割で 6 パターン {(4,2,1),(2,4,1),(2,4,1),(1,4,2),(2,1,4),(1,2,4)}，三次元分割で 1 パターン {(2,2,2)} の合計 10 通りの分割パターンが存在する．以下の分割パターン表記は全てこの表記方法に従う．そして，三次元空間において，各方向の格子分割数が異なる場合，図 2 のようになり上の分割パターンのさらに 3 倍を考慮する必要がある．

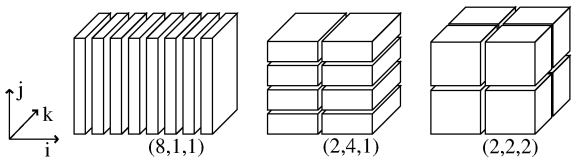

Fig. 1: For example, Partitioning patterns for domain decomposition method

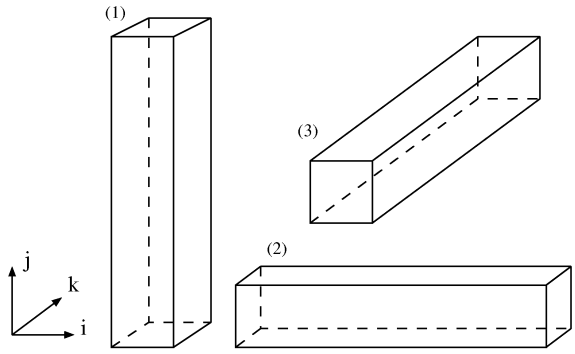

Fig. 2: For example, grid style

並列プログラムの経験があるプログラマは，通常領域分割を行う場合，通信量が最小で最内ループ (FORTRAN では，配列の最左インデックスが連続アクセスされるループ) がある程度長くなるように，プログラミングすることが多い．

例えば，図 1 のような立方体の三次元空間を考慮すると，三次元領域で格子分割数が各次元ともに等しく PE 数が多い場合，通信量が小さくなるため，三次元の分割

パターンが用いられる．通信帯域が低い場合では，通信量が最小であることの意味は大きいが，通信帯域がある程度広い場合やベクトル並列機の場合，分割次元数を増やすことは，最内ループを短くし計算性能を落とす事がある[3]．分割パターンの違いによる性能の変化を予測する方法は，このような現象を予測する必要がある．

2.2 通信手順

分割した各小領域すなわち各 PE における通信処理は，

1. 隣接領域の送信データ領域を自領域の受信データ領域に受信
2. 自領域の送信データ領域を隣接領域の受信データ領域に送信

の二つの通信で構成される．この通信が各 PE で実行される．図 3 に二次元領域を二次元に分割した場合の通信方法を示した．communication 1 と 2 で 1 回目，3 と 4 で 2 回目の通信が行われる．送信データは送信データ領域 (Send data region) の 1 辺全てを受信データ領域 (Overlap region) に送信する．これらの通信操作には，全て同期通信を用いた．

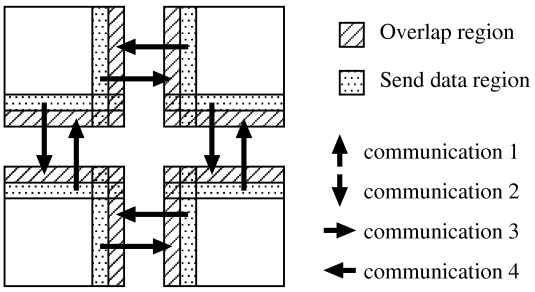

Fig. 3: Communication style

3. 性能予測

逐次処理時の MAC 法ベースの CFD ソルバの計算負荷は，物理量の計算部分によってかなりの片寄りが見られる．通常，圧力解法部分の計算負荷が非常に高くなる傾向がある．そのため，計算負荷の見積もりを行う場合，ソルバのカーネル部分である Poisson 方程式を解くことで CFD ソルバ全体の計算負荷をある程度見積もる事が可能であり，ベンチマークプログラムとして良く用いられる[6]．この考え方は，並列 CFD においても十分に通用するものと考えられる．

分子動力学の計算モデルを用いて，非常に精巧な性能予測モデルから性能評価が行われている[7]．しかし，性能評価モデルでは，用いるパラメータが簡単であるほど良い．パラメータが複雑であれば，そのモデルを使いこなすには，かなりの技術と知識が必要となるため，容易で機械的な性能評価は行えない．本稿で用いるパラメータは，一般に良く知られているハードウェア性能のパラメータのみを用い，その他のパラメータは，容易な方法で得られる値を用いる．

本稿で用いた並列計算アルゴリズムは，通信隠蔽等を行わず，計算部と通信部は分離できる．そして，計算性能の予測に用いたカーネル部分の並列計算の流れは図 4 となる．そのため，計算モデルは計算性能と通信性能をそれぞれ分離して考慮して良い．

4. 並列 CFD の計算性能予測

4.1 計算性能

通常，計算量を元にした計算性能モデルの場合，分割数が固定で，分割パターンが違う小領域内の計算時間は，計算量が変化しないため，計算性能はどの分割数でも同じとして扱われることが多い．しかし，現実には同一ではあり得ない．

計算性能は，各分割パターンでの 1 PE の計算領域の計算性能で見積もる．ただし，上述のように流体コード全体の計算性能ではなく，流体コードの計算負荷の高い

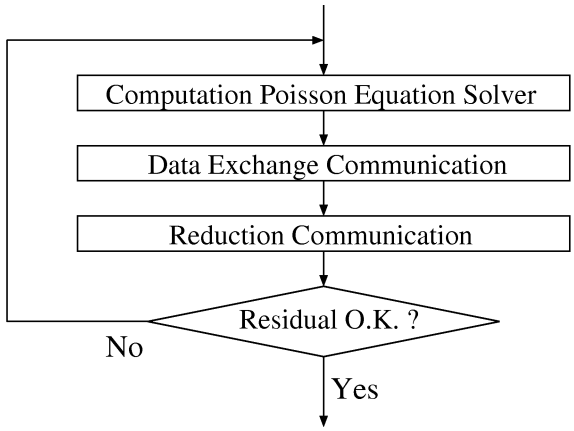

Fig. 4: Overview of Parallel Computation

部分だけを領域分割から得られる各分割パターンによる分割形状を総当りで計測する．計算負荷が高い部分のみに限定することで，計測時間はすべての分割パターンを総当たりで計測しても大きくない．そして，1 PE のみを利用するため，計算資源も少なくすむ．

ここで重要なのは，計算量に変化がなく，計算ループの長さが変化する場合には，ベクトル計算機では当然のことであるが，スカラー計算機でも計算性能にかなりの影響を与えることを前提とすることである．

そのため，領域分割によって得られる 1 PE の領域の計算性能が明らかになれば，n PE での CFD の全体の計算性能をかなり正確に予測することが可能となる．また，計算性能や通信性能の両者の性能予測に大きな影響があるため，データ移動性能を予測することも可能になる．

4.2 通信性能

並列計算機の通信性能の予測モデルは，データ通信量，通信帯域，レイテンシそして通信オーダーモデルによって決定される．通信帯域とレイテンシは，ハードウェア理論性能を用いることで簡単に決定できる．また，メモリ移動時間も上記の計算性能を用いることで予測が可能である．そして，通信オーダーモデルは，用いる計算機のトポロジーやルーティングから割り出すことが可能である．

IBM RS/6000 SP の場合，スイッチネットワークを用い，ルーティングの方法の違いによって，クロスバールーティング[4]や Tree (Fat Tree) ルーティング [5]が可能である．

クロスバールーティングであれば，どのような分割パターンでも 1 組の通信に要する通信コストは，$O(1)$ になる．また，Tree (Fat Tree) ルーティング (図 5) であれば，分割パターンによって通信コストは異なる．各次元の分割数を N_i，N_j，N_k とすると $O(log_2(N_i))$, $O(log_2(N_j))$, $O(log_2(N_k))$ となる．例えば，8 PE で (1,1,8) の一次元分割の場合 $O(log_2 8)$，(1,2,4) の二次元分割の場合 $O(log_2 2)$ と $O(log_2 4)$ となる．本稿で用いた SP システムは，Tree ルーティングであるため，通信コストは，Tree (Fat Tree) ルーティングモデルを用いた．

通信処理に対するデータ移動性能をモデル化するために，計算性能で得られた実測値を用いる．計算処理性能の実測値は，本来理論性能と同値になるはずであるが，様々な理由からほぼ理論性能は得られない．そのため，理論性能が得られない理由をデータ移動性能の影響であると仮定し，通信性能の理論性能が得られない理由も同様であると考える．通信性能の予測にも計算性能の実測値を用いることで有効な通信性能予測を行えるものと考えられる．

以上のことから，分割パターンによる通信時間を算出することが可能となる．この通信時間と 1 PE が計算する小領域の計算量から 1 秒辺りの浮動小数点演算量 (Mega Floating point operation per second : MFlop/s) 値として評価する．

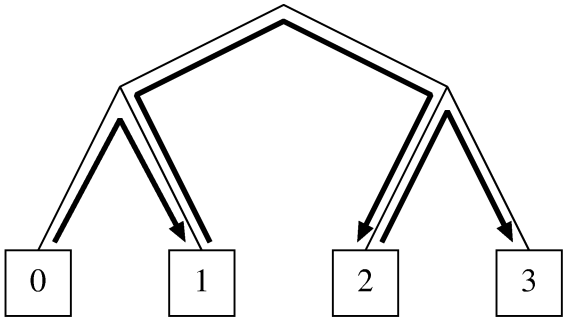

Fig. 5: For example, Communication pattern using Tree Topology

最近，低コスト並列計算機として注目されている PC クラスタの場合，一般にスイッチ Hub (インテリジェンス or ブリッジ) を用いて，全ての PC を接続するか，スイッチ Hub を数台用いてカスケード接続するかである．本稿で用いた通信処理を単一スイッチ Hub 接続の PC クラスタで実行することを考えると，図 6 のように 2 分割の場合 $O(1)$ となるが，4 分割以上では $O(2)$ となる．そのため，各方向の分割数を 2 分割にする意味は大きい．しかし，スイッチ Hub の性能は，用いるスイッチ本体の性能によってかなり大きな開きがあり，かならずしもこのような通信コストになるとは限らないため注意が必要である．

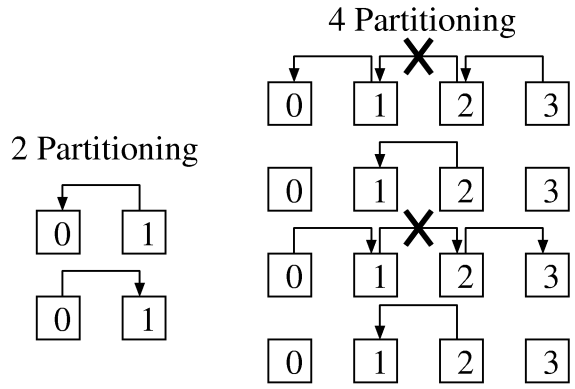

Fig. 6: For example, Communication pattern using Switching Hub

また，並列 CFD 計算時間に占める通信時間として大きな影響がある反復制御のリダクション通信 (総和や最大値を求める通信，例：MPI_Reduce 等) は，用いる PE 数に依存するため，本稿のような PE 数を固定した場合の分割パターンを選択する場合，PE 数に依存するリダクション通信は考慮する必要はない．

5. 並列処理コストモデル

上述の点を踏まえて，並列 CFD の計算性能予測モデルを計算カーネル部 (Poisson 方程式解法部) の性能モデルとして作成する．モデルから得られる値は，全て MFlop/s に換算し，最大値を 1 として評価する．

5.1 並列計算コスト

並列計算コストは，1 PE の処理時間を並列処理の計算コストと考える．そして，1 PE の計算時間を MFlop/s として評価する．1 反復中の総浮動小数点演算量を $FLOPs$ と計算時間 $Time$ すると，計算コスト $Cost_{comp}$ は，

$$Cost_{comp} = \frac{FLOPs}{Time(\mu s)} \tag{1}$$

となる．実際に測定した $Cost_{comp}$ は，

$$Cost_{comp} = \frac{FLOPs \times ITR}{Time(\mu s)} \tag{2}$$

$$ITR : Number\ of\ Iterations$$

となるが，式 (1) の $Cost_{comp}$ は 1 iteration での $Cost_{comp}$ となるため，式 (2) と違いはない．本稿では $ITR = 200$ として計測した．

5.2 通信処理コスト

カーネル計算部分の 1 反復に必要な通信処理時間 (s) を理論性能からモデル化すると

$$\text{Comm Time} = BW \times \sum_{n=1}^{N} (DS_n \times log_2 DD_n) + L \times N \tag{3}$$

となり，N は分割次元数，DS は各次元での通信データ量 (Byte)，BW は理論通信帯域幅 (Byte/s)，DD は各分割次元での分割数，L は通信立ち上がり時間 (レイテンシー) である．このままの状態でもある程度の通信処理コストの検討は可能であるが，本稿では，並列計算コストの値を用いて，データ移動性能 DT を

$$DT = \frac{CF}{Cost_{comp}} \tag{4}$$

として評価する．CF にどの値を用いるかはハードウェアにある程度依存するが，本稿では CPU の駆動周波数 (MHz) とした．式 (3)，(4) を考慮すると通信処理時間は，

$$\text{Comm Time} = BW \times DT \times \sum_{n=1}^{N} \Big(DS_n \times log_2 DD_n \Big) + L \times N \tag{5}$$

となる．式 (5) と $FLOPs$ から通信コスト $Cost_{comm}$ は，

$$Cost_{comm} = \frac{FLOPs}{\text{Comm Tine} \times 1.0 \times 10^{-6}} \tag{6}$$

となる．

5.3 モデル

以上のことから，並列処理コスト $Cost_{Total}$ は，

$$Cost_{Total} = Cost_{comp} + Cost_{comm} \tag{7}$$

となり，並列処理の評価値は，$Cost_{Total}$ の最大値 $Cost_{Total_{max}}$ で正規化したものを用いた．IBM RS/6000 SP の通信性能は，文献 [8] に示されているが，本稿では，容易に得られる値に重点を置き，理論性能である BW を 150 MB/s，レイテンシ L は，理論値として 1.2×10^{-6} とした．

6. 計算対象

180 度曲がり管内流れの計算 (非圧縮粘性流れ:総格子点数約 15 万点，$130 \times 34 \times 34$) を 8 並列で用い，性能予測値との比較を行なう．計算形状モデルおよび結果を図 7 に示す．

プログラムは一般座標系によって定式化した三次元非圧縮 Navier-Stokes (NS) 方程式を差分法で離散化した．解法は MAC 法に準じる．空間の離散化精度は，移流項は三次精度風上差分，その他の部分は二次精度中心差分とし

た．方程式解法には，分割パターンの変化による収束性の変化の無い Jacobi 法を用いた．Jacobi 法の収束条件は，1.0×10^{-4} とした．レイノルズ数は 200，時間ステップは，50 とした．境界条件は，流入口で圧力 $P = 1.0$，速度 $\frac{\partial V}{\partial X} = 0.0$ とし，流出口で圧力 $P = 0.0$，速度 $\frac{\partial V}{\partial X} = 0.0$，滑り無し壁上で圧力 $\frac{\partial P}{\partial X} = 0.0$，速度 $V = 0.0$ とした．

分割パターンは，10 パターン：(8,1,1)，(1,8,1)，(1,1,8)，(4,1,2)，(4,2,1)，(1,4,2)，(2,4,1)，(1,2,4)，(2,1,4)，(2,2,2)，格子の取り方が 3 パターン：Type-A：130 × 34 × 34，Type-B：34 × 130 × 34，Type-C：34 × 34 × 130 の合計 30 パターン存在する．

この問題を並列プログラム経験者が領域分割法で並列化する場合，Type-C，(1,1,8) を用いると考えられる．なぜなら，通信量は最小であり，分割次元も一次元となり領域分割が行いやすい．そのため，この格子の取り方を分割パターンが基準と考えて良い．この分割パターンを特に *base pattern* とする．

7. 結　果

7.1 流れ場

時間ステップを進めた場合の結果として流跡線を図 7 に示す．

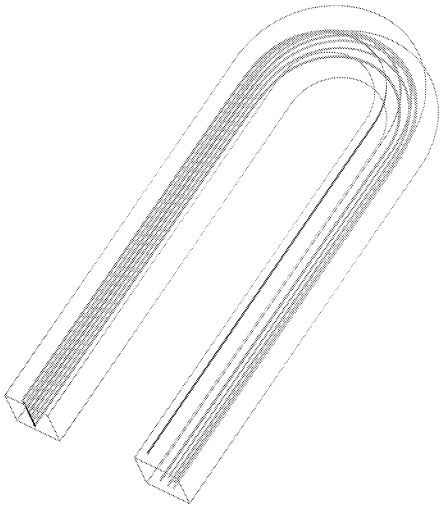

Fig. 7: Overview Pipe Flow

管内の二次流れ影響による，流跡線の旋回が見られる．

7.2 並列計算性能予測

並列 CFD 計算では，以下の各時間を計測した．並列処理の総経過時間 $Etime$，並列処理時の総計算処理時間

$$Ctime = \left(\sum_{n=1}^{NPE} Ctime_n \right) /NPE,$$

並列処理時の通信時間 $Ptime = Etime - Ctime$ である．NPE は使用 PE 数とする．

図 8 に $Etime$ と性能予測値の相関，図 9 に $Etime$ と $Cost_{comp}$ の相関，図 10 に $Ptime$ と $Cost_{comm}$ の相関および $Ptime$ と $Cost_{comm}$ 値の算出に DT 値を用いない場合の相関をそれぞれ示す．図 8 は縦軸に並列 CFD の $Etime$，横軸に評価値をとり，図 9 は縦軸に $Etime$，横軸に $Cost_{comp}$ をとり，図 10 は縦軸に $Ptime$，横軸に $Cost_{comm}$ を取った．また，表 1 に 予測最良分割パターン，*base pattern* と実測値の最良分割パターンおよび $Cost_{comp}$ と $Cost_{comm}$ の最大値での分割パターンと総経過時間を示す．

図 8 から実際の流体計算のパフォーマンスとの比較から，性能予測値と実経過時間は，相関係数 -0.94 と良好な結果が得られた．また，図 9 から $Cost_{comp}$ のみの評価

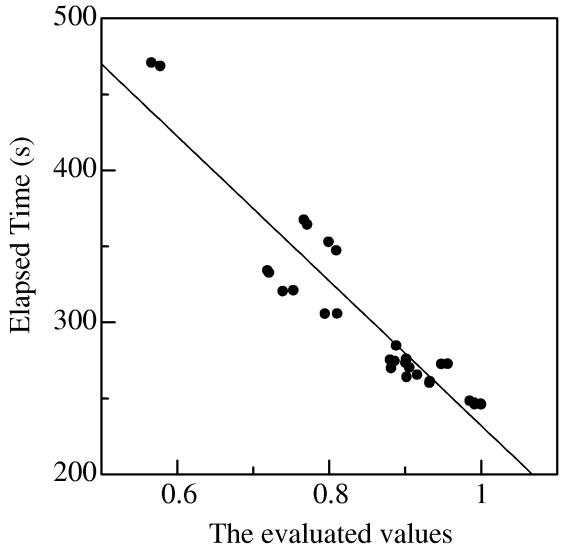

Fig. 8: Correlation between experiment and evaluation

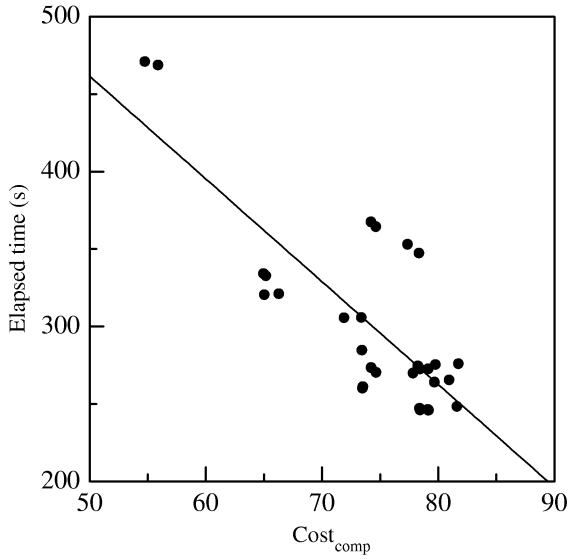

Fig. 9: Correlation between $Cost_{comp}$ and $Etime$

では，相関係数 -0.79 と十分な精度の性能予測は得られない．また，図 10 から $Cost_{comm}$ と実際の通信処理時間の相関係数は -0.86，DT 値を用いない場合では，-0.79 となり，DT 値は予測値をより実際の性能に近づける効果があった．

表 1 から並列計算性能予測値と実際の最良の並列計算性能との結果が一致したことが分かった．また，*base pattern* と最適分割パターンでは，30 (s) 程度の差が見られた．そして，$Cost_{comp}$ と $Cost_{comm}$ の個々の最大値は，最良の並列計算性能とは一致しなかった．

8. 考察

逐次処理の場合，CFD コードのカーネル部分をベンチマークとして用いれば，実際の CFD コードの性能をかなり正確に評価することが出来る．また，本稿の結果から，その考え方は，並列 CFD コードにも適用できることが分かった．本稿では，予測値と実際の最適な CFD 計算時間の分割パターンが一致した．IBM RS/6000 SP 以外の並列計算機を用いた場合の検討も必要であるが，この方法を用いることで，分割パターンを選択する際に最適な分割パターンに近いパターンが得られると考えられる．

図 9 から明らかなように，実際に並列 CFD の性能を評価するには，計算性能だけを評価したのでは正確な評

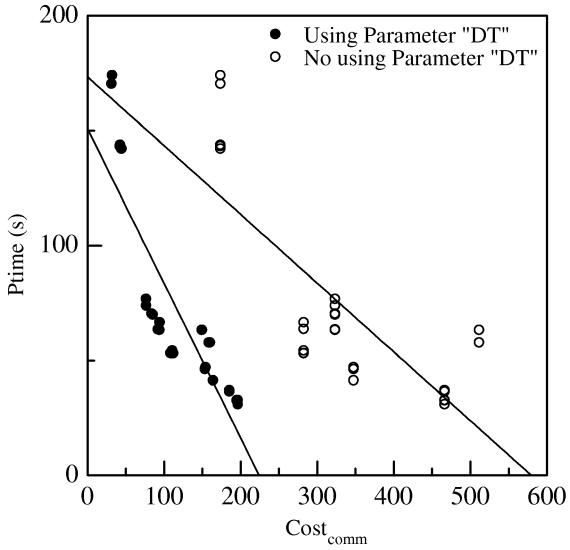

Fig. 10: Correlation between $Cost_{comm}$ and $Ptime$

Tab. 1: Result

	Grid Type & Partition pattern	Elapsed Time (s)
Evaluate better pattern	Type-B, (1,4,2)	246.002
Elapsed best pattern	Type-B, (1,4,2)	246.002
base pattern	Type-C, (1,1,8)	272.677
Maximum $Cost_{comp}$	Type-A, (1,2,4)	275.968
Maximum $Cost_{comm}$	Type-C, (1,2,4)	247.169

価が行えない．また，通信処理に関しても図 10 のように理論性能のみを用いても，良い結果が得られない．また，Ping-Pong ベンチマークを用いて，通信性能に実測値を用いても図 10 の DT 値を用いない場合のデータがグラフ中を平行移動するだけで，実際の値に近づくことはない．そのため，計算性能から得られる DT 値を通信性能の評価に用いることで，有効な結果が得られる．また，DT 値をさらに細かな評価値に置き換えることで，さらに精度が上がるものと考えられる．

並列計算モデル作成で最も複雑だと思われる部分は，通信性能の決定に影響がある．通信オーダーの部分である．並列計算機のネットワークトポロジーは公開され，一般に Cross-bar，Tree，Torus，Switch 等のネットワークがある．これらネットワークの通信オーダーの決定は並列計算性能の評価に大きな影響があり，難しい部分である．このため，今後この部分を容易なパラメータに置き換える可能性を考慮する必要がある．

また，領域分割法を用いた並列計算を境界領域を先行して計算し，通信と内部領域の計算を同時に実行する通信隠蔽処理をおこなう事で並列計算性能を向上させる方法がある[9]．この方法を用いた場合の並列計算性能を評価するばあいにも，本稿の方法を用いれば，計算の評価と通信の評価が MFlop/s で得られるため，通信隠蔽の効果があるかどうかを即座に評価できる．

9. おわりに

従来並列数値計算プログラムの性能評価には，細かなパラメータを用いたものが多い．しかし，主要計算部分だけの逐次処理時の実測値を用いるだけで，非常に有効な性能評価をより容易に行なえる．

そして，主要計算部分 (Poisson 方程式解法部) の性能評価は重要な要素となある．主要計算部分の評価は，計算部分の性能評価のみならず，通信処理部分の評価にも大きな影響を与えることが分かった．

参考文献

1. Crandall P.E., Quinn M.J. : *Three-Dimensional Grid Partitioning for Network Paralle Processing*, In Proceedings of the ACM 1994 Computer Science Conference, (1994)
2. 黒川原佳, 姫野龍太郎, 重谷隆之, 松澤照男:三次元ポアソン方程式に対する領域分割法の分割方法による性能への影響, 情報処理研究会報告, 2000-HPC-80, pp.137-142 (2000)
3. Kurokawa M., Himeno R., Shigetani T., Matsuzawa T.: *A case study of the partitioning patterns for domain decomposition method on VPP700E*, RIKEN HPC Review No.30, pp.30-34 (2000)
4. IBM Corporation : *The RS/6000 SP High-Performance Communication Network*, http://www.rs6000.ibm.com/resource/technology/sp_sw1/spswp1.book_1.html
5. Siegel H.J. : *INTER-CONNECTION NETWORKS FOR LARGE-SCALE PARALLEL PROCESSING*, McGraw-Hill Publising Company, 1990
6. http://w3cic.riken.go.jp/HPC/himenoBMT/
7. 折居茂夫：数値計算のための並列計算機性能評価方法, 情報処理学会論文誌, Vol.39, No.3, pp.529-541, (1998)
8. IBM Corporation : *SP Switch Performance*, Version 3, (1999)
9. 黒川原佳, 松澤照男, 姫野龍太郎, 重谷隆之:境界領域先行計算による通信隠蔽処理を行う並列計算アルゴリズム, 第 13 回計算力学講演会講演論文集, pp.499-500 (2000)