JAIST Repository
https://dspace.jaist.ac.jp/

Title	並列CFD計算における非同期通信-計算重複法
Author(s)	黒川，原佳；松澤，照男；姫野，龍太郎；重谷，隆之
Citation	情報処理学会論文誌 ： ハイパフォーマンスコンピューティングシステム，42(SIG9(HPS3))：54-63
Issue Date	2001-08-15
Type	Journal Article
Text version	publisher
URL	http://hdl.handle.net/10119/4552
Rights	社団法人　情報処理学会，黒川原佳、松澤照男、姫野龍太郎、重谷隆之，情報処理学会論文誌 ： ハイパフォーマンスコンピューティングシステム，42(SIG9(HPS3))，2001，54-63．　ここに掲載した著作物の利用に関する注意: 本著作物の著作権は（社）情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。 Notice for the use of this material: The copyright of this material is retained by the Information Processing Society of Japan (IPSJ). This material is published on this web site with the agreement of the author (s) and the IPSJ. Please be complied with Copyright Law of Japan and the Code of Ethics of the IPSJ if any users wish to reproduce, make derivative work, distribute or make available to the public any part or whole thereof. All Rights Reserved, Copyright (C) Information Processing Society of Japan.
Description	

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

Vol. 42 No. SIG 9(HPS 3) 情報処理学会論文誌：ハイパフォーマンスコンピューティングシステム Aug. 2001

並列CFD計算における非同期通信–計算重複法

黒川原佳[†] 松澤照男[††]
姫野龍太郎[†††] 重谷隆之[†††]

大規模な Computational Fluid Dynamics（CFD）計算を実用時間内に行うために，並列計算の必要性はきわめて高い．並列 CFD 計算は，計算時間と同時に通信時間も費やす．そのため，通信性能の低い分散メモリ型並列計算機上で流体計算を効率良く行うには，データ通信処理の方法が重要である．効率的な通信処理方法の 1 つは，通信処理を計算処理で隠蔽し，通信処理時間を擬似的に短くする通信隠蔽処理法がある．通信隠蔽処理法には，パイプライン処理法と非同期通信–計算重複法がある．本研究では，領域分割法を用いた Maker And Cell（MAC）法の並列 CFD 計算に，非同期通信–計算重複法による通信隠蔽を適用して並列計算を行った．また，領域の分割パターンは，最も並列計算効率の高いものを用いた．そして，非同期通信–計算重複法の適用による総経過時間への効果を検討した．非同期通信–計算重複法の適用において，PE 数が少ない場合，非同期通信の待ち処理（MPI_Wait）時間は，総経過時間に対して無視できない長さだった．しかし，PE 数の増加に応じて待ち処理時間は短くなった．待ち処理時間は，通信データ量に依存した．また，非常に小さな問題サイズで非同期通信–計算重複を行った場合にも過剰な性能飽和は見られなかった．通信隠蔽を行わない CFD の実装方法に比べ，非同期通信–計算重複法を適用した場合の速度向上比は，RS/6000 SP で最大 14%程度，PC クラスタで最大 31%程度向上した．

A Systolic Communication-Computation Overlap Method for Parallel CFD

Motoyoshi Kurokawa,[†] Teruo Matsuzawa,[††]
Ryutaro Himeno[†††] and Takayuki Shigetani[†††]

A parallel computation for a large-scale Computational Fluid Dynamics (CFD) simulation is important for the real-world simulations. A parallel CFD computation time consists of the computation time and the communication time. The communication processing is important to compute CFD efficiently on a distributed memory based parallel computer with low-speed communication system. The overlapping of communication with computation is a method of pseudo-shortening communication time. The overlapping of communication with computation by the pipeline method and the systolic communication-computation overlap method. In this research, we executed the parallel CFD simulation by the Maker And Cell (MAC) method and the domain decomposition method using the systolic communication-computation overlap. We used a most efficient partitioning pattern for the domain decomposition. We discussed the effect of the systolic communication-computation overlap for a parallel CFD on a total elapsed time. In result, $Wait$ processing (MPI_Wait) of the asynchronous communication using the systolic communication-computation overlap by a few number of PE had an elapsed time, which was not able to be disregarded. However, the elapsed time of $Wait$ processing became short when the number of PE increased. The elapsed time of $Wait$ processing depended on the amount of the data communication. The elapsed time of the systolic communication-computation overlap does not have excessive performance saturation on the very small problem size. We show the speed up ratio of the parallel CFD that the systolic communication-computation overlap implementation improved the maximum performance by 14% on the RS/6000 SP and 31% on the PC Cluster against the normal implementation (no overlapping communication with computation).

† 北陸先端科学技術大学院大学情報科学研究科博士後期課程
School of Information Science, Japan Advanced Institute of Science and Technology
†† 北陸先端科学技術大学院大学情報科学センター
Center for Information Science, Japan Advanced Institute of Science and Technology
††† 理化学研究所情報環境室
Advanced Computing Center, The Institute of Physical and Chemical Research (RIKEN)

1. はじめに

超並列計算機上で Computational Fluid Dynamics

（CFD）の大規模計算を行うためには，効率的な並列アルゴリズムと並列プログラミングが重要な課題となっている．効率的な並列計算プログラムは，通信のオーバヘッドをいかに少なくするかが非常に重要である．通信オーバヘッドを少なくする手段として，計算処理時間に通信処理時間を重ね合わせて，総経過時間に対する通信処理時間を仮想的に減少させる通信隠蔽処理法がある．通信隠蔽処理法は，通信処理と計算処理が独立に行える場合に適用できる．

通信隠蔽処理法には，パイプライン処理法と非同期通信–計算重複法の 2 つが考えられる．パイプライン処理は NASA Ames Research Center で開発された NAS Parallel Benchmark（NPB）のアプリケーション LU で Symmetric Successive Over Relaxation（SSOR）法に用いられている[1]．しかし，通信機構の性能が低い場合や計算粒度が非常に小さい場合，並列計算性能が低下する[2]．

計算粒度が非常に小さな場合や通信機構の性能が低く，パイプライン処理では並列化効率が低下する場合でも，並列化効率の低下を防ぎ通信隠蔽を行う方法として非同期通信–計算重複法がある．非同期通信–計算重複法は，領域分割が可能で領域間のデータ依存性が少ない場合やデータ依存が領域境界部分に集中する場合に特に有効である．Quinn ら[3] は，二次元モデルについて自動並列コンパイラ技術を用いた非同期通信–計算重複法の有効性や性能限界を理論的に示した．さらに，二次元モデルについて，Fink ら[4] や田中ら[5] は，SMP クラスタを用いた研究を行っている．しかし，三次元モデルにおける非同期通信–計算重複法の有効性や問題点は，明らかにされていない．

また，通信隠蔽処理の CFD への応用研究は，以下のような研究がある．Evans ら[6] は，自動並列化ツールを用いたパイプライン処理による通信隠蔽のプログラミングを示し，NPB LU 等で良好な並列化効率を示した．しかし，非同期通信–計算重複法を実際の CFD に応用した例はほとんどない．

多くの CFD 計算において，最も計算に時間を要する部分は方程式解法部分である．代表的な CFD の計算方法の 1 つである Maker And Cell（MAC）法は，圧力と速度の分離解法である．次時間ステップの圧力と速度は，以下のように導かれる．

(1) 連続の式と Navier-Stokes（NS）方程式から導かれた Poisson 方程式を陰的に解くことで圧力を求める．
(2) 既値の速度と方程式から得られた圧力を NS 方程式に代入して，速度を求める．

この過程で (1) の圧力を得るための Poisson 方程式を解く部分の計算負荷が圧倒的に大きく，この部分の通信処理を隠蔽することで，並列計算性能全体が向上する．

MAC 法は非定常問題の解法としてよく用いられる．本稿では定常流れを扱うが，定常流れも時間ステップの初期段階では擬似的な非定常計算と考えられる．よって，定常流れの初期段階の性能評価により一般的な非定常流れの性能が類推できる．

非同期通信–計算重複法は，Successive Over Relaxation（SOR）法等の計算順序に依存する反復法に適用したとき，計算順序を変更するため，収束に影響を与える場合がある．一方で収束性は，流体の物理状態や計算領域によっても大きく変化する．このため本稿では，収束性を考慮せず並列計算性能のみを議論するため Jacobi 法を用いた．Jacobi 法は，計算順序の変更や領域の分割パターンによる収束性の変化はない．また，この方法は緩和法の基本的な形態であり，収束を早めた様々な緩和法の並列性能を類推できる．

本稿では，MAC 法による並列 CFD 計算に対する非同期通信–計算重複法の実装方法を提案する．実装には，高速通信機能を追加したクラスタシステムである RS/6000 SP と通常の 100 Base イーサネットを用いた PC クラスタシステムを用いた．非同期通信–計算重複法による性能向上が，並列 CFD 計算の全体性能に与える影響を異なる 2 つのシステム上で 2 種類の問題サイズにおいて非同期通信–計算重複法の効果を検討する．

2. CFD 計算

2.1 支配方程式

非圧縮粘性流体解析の支配方程式は，速度ベクトルを $\mathbf{V}$，圧力を p として，ベクトル形式で表記すると質量保存を表す連続の式

$$\nabla \cdot \mathbf{V} = 0 \tag{1}$$

と運動量保存を表す NS 方程式

$$\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla)\mathbf{V} = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{V} \tag{2}$$

$$\nabla = \mathbf{i}\frac{\partial}{\partial \mathbf{x}} + \mathbf{j}\frac{\partial}{\partial \mathbf{y}} + \mathbf{k}\frac{\partial}{\partial \mathbf{z}}$$

ただし，i，j，k は各軸方向の単位ベクトル（Re は，$(LU)/\nu$ で定義する無次元量のレイノルズ数である．ただし，L は代表長さ，U は代表速度，ν は動粘性係数）となる．式 (2) の左辺第 2 項が非線形であること，速度$\mathbf{V}$ が時間発展であるのに対し圧力 p は時間発展ではない等によって，安定な数値解を得る

ために様々なスキームが開発されている．これらの中で MAC 法は，代表的なスキームの 1 つである．

表記を簡略化するために式 (2) を

$$\frac{\partial \mathbf{V}}{\partial t} = f(\mathbf{V}) - \nabla p \tag{3}$$
$$f(\mathbf{V}) = \frac{1}{Re}\nabla^2\mathbf{V} - (\mathbf{V}\cdot\nabla)\mathbf{V}$$

とする．式 (3) を時間に関して前進差分し，さらに両辺の発散をとると，

$$\frac{\nabla\cdot\mathbf{V}^{n+1} - \nabla\cdot\mathbf{V}^{n}}{\Delta t} = \nabla\cdot f(\mathbf{V}^n) - \nabla^2 p^{n+1} \tag{4}$$

が得られる．さらに時刻 $n+1$ では連続の式を満たす必要があるため，$\nabla\cdot\mathbf{V}^{n+1} = 0$ である必要がある．よって，

$$\nabla^2 p^{n+1} = \frac{\nabla\cdot\mathbf{V}^{n}}{\Delta t} + \nabla\cdot f(\mathbf{V}^n) \tag{5}$$

のような圧力に関する Poisson 方程式が得られる．次に，式 (3) を時間で前進差分した式に式 (5) から得た p^{n+1} を代入すると最終的に速度の時間発展を表す次式が得られる．

$$\mathbf{V}^{n+1} = \mathbf{V}^n + (f(\mathbf{V}^n) - \nabla p^{n+1})\cdot\Delta t \tag{6}$$

式 (5)，(6) が MAC 法の基礎方程式である[7)~9)]．

CFD 計算では，計算対象の物体表面近傍で物理量の変化が大きい．計算精度を向上させるため計算格子点を物体表面近傍に多く配置する[7)~10)]．また，任意形状の計算対象を差分法で離散化する場合，直交格子では，境界条件の扱いが非常に複雑で，プログラミングが煩雑である．各方程式の一般座標変換は，物体表面に格子点を集中させることや境界条件の扱いを容易にする．

一般座標系で定式化を行うために，式 (5) の三次元物理空間 (x, y, z) での圧力 p に関する Poisson 方程式を次のように整理する．

$$\frac{\partial p^2}{\partial^2 x} + \frac{\partial p^2}{\partial^2 y} + \frac{\partial p^2}{\partial^2 z} = -g(x,y,z) \tag{7}$$
$$-g(x,y,z) = \frac{\nabla\cdot\mathbf{V}^{n}}{\Delta t} + \nabla\cdot f(\mathbf{V}^n) - \nabla p$$

三次元物理空間座標から一般座標 (ξ, η, ζ) の矩形計算空間への写像関数を次のように表すと，

$$\begin{cases} x = x(\xi,\eta,\zeta) \\ y = y(\xi,\eta,\zeta) \\ z = z(\xi,\eta,\zeta) \end{cases} \tag{8}$$

式 (8) を式 (7) に代入し，係数および右辺を簡略化した X, Y, Z, G で表すと，Poisson 方程式は，次のように整理できる．

$$X(\xi,\eta,\zeta)\frac{\partial p^2}{\partial^2\xi} + Y(\xi,\eta,\zeta)\frac{\partial p^2}{\partial^2\eta} + Z(\xi,\eta,\zeta)\frac{\partial p^2}{\partial^2\zeta} = -G(\xi,\eta,\zeta) \tag{9}$$

式 (6) の NS 方程式も同様の操作を行うことで一般座標に変換することができる．

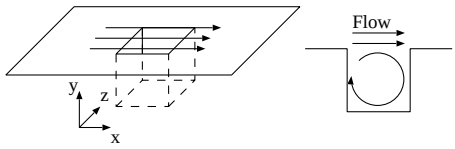

図 1 キャビティー流れの概略
Fig. 1 Overview of flow in a driven cabity.

表 1 計算格子サイズ
Table 1 Grid system size.

	Grid size	Number of grid points
Type A	$26 \times 26 \times 26$	17,576
Type B	$66 \times 66 \times 66$	287,496

2.2 計算対象

非圧縮粘性流体の計算モデルは，**図 1** に示す三次元正方キャビティー流れとした．Re 数は 100 である．ただし，L を正方キャビティーの一辺の長さ 1，U を 1，そして ν を 0.01 とした．計測対象は非定常性が強い計算初期の 100 ステップまでとした．

計算格子は境界付近で格子点間隔を細かくとる．計算格子の設定は直交不等間隔格子である．方程式は直交不等間隔の定式化ではなく，一般座標変換（2.1 節）をともなった定式化を行った．

検討に用いた計算格子サイズは，**表 1** に示す 2 種類である．また，CFD 計算を安定に解くにはクーラン数 $(= (u\Delta t)/(\Delta x_{\min}))$ を考慮する必要がある．速度計算に陽解法を用いたため，クーラン数は 0.2 とした．よって，時間ステップは最小格子点間隔の影響で大きな値をとれない．どちらの計算格子も最小格子点間隔 $(\Delta x_{\min})$ は同じとし，時間ステップ幅 Δt も同一である．

2.3 計算条件

式 (5) の Poisson 方程式の離散化は，二次精度中心差分を用いた．式 (6) の NS 方程式の離散化は，左辺第 2 項の移流項に三次精度風上差分を用い，その他の空間微分項はすべて二次精度中心差分を用いた．また，速度の時間微分項は，一次精度前進差分を用いた．

図 1 において，x 方向の流速を u，y 方向の流速を v，z 方向の流速を w とした．速度の境界条件は，上

の境界が $u = 1.0$，$v, w = 0.0$ とし，その他の境界は $u, v, w = 0.0$ とした．圧力の境界条件は，すべての境界で圧力勾配を 0.0 とした．

プログラミングは，Fortran 77 と Message Passing Interface（MPI）ライブラリを用い，計算精度は倍精度とした．三次元計算空間の格子点における物理量を直接三次元の配列に割り当てた．また並列化に際して，Processing Element（PE）内で計算に必要なデータのみを確保し，計算に用いない部分のデータは確保しない．

Jacobi 法の収束判定は，各格子点での残差の最大値が 1.0×10^{-5} になるまでとした．

3. 領域分割法による CFD 計算の並列化

領域分割法を用いた並列計算は，全計算領域を小領域に分割し，小領域を各 PE に割り当て，全計算領域の計算結果は，小領域間でデータ交換を行い，個別に計算して得る．本稿では，有限差分法の並列化でよく用いられる重複領域を持つ領域分割法を用いた．この方法は，他領域の境界領域（図 3 の *Boundary region*）を自領域の重複領域（図 3 の *Overlap region*）のデータとして用いる．自領域の重複領域の値が決定されれば，自領域内部の計算は他領域と独立に解くことができる．重複領域は空間差分精度によって異なり，本稿の場合，圧力計算は二次精度の中心差分であり，重複領域は一格子点分である．流速計算は最も高次精度が三次の風上差分のため，重複領域は二格子点分である．

領域分割に用いる分割パターンは，通信データ量や計算のループ長の違いから全体性能に影響を及ぼす[11]．最適な分割パターンは，計算にかかわる要素（ハードウェア，ソフトウェア，計算スキームや問題サイズ等）が複雑に関連し，並列計算を実行する前に決定することは非常に難しい．非同期通信–計算重複を行わない領域分割法を用いた並列 CFD 計算において，プログラマは 1 回の通信（4 章）あたりの通信データ量が少なくなる多次元の分割パターンを用いることが多いが，最適な分割パターンではない[12]．本稿では，並列 CFD 計算の全体性能を検討することが目的であるため，各 PE 数で最も計算性能が高い分割パターンを用いた．

4. 通 信 処 理

各 PE におけるデータ交換は，

(1) 自領域の境界領域を隣接領域の重複領域に送信
(2) 隣接領域の境界領域を自領域の重複領域に受信

の 2 つの通信処理で構成される．**図 2** は，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．上で示した処理を 1 回の通信とすると，通信処理は，communication phase 1 と 2 で上下，3 と 4 で左右の 2 回である．座標変換をともなった差分計算は，通常計算に用いる格子点だけでなく，斜め方向の点も用いる．二次元以上の分割での通信処理は，斜め方向の通信処理が必要となる．しかし，Evans ら[13] の方法は，斜め通信を行わずに済み，通信処理手順が簡略化できる．二次元領域を二次元で分割した図 2 の受信領域（*Receive region*）は，隣接領域の存在する側の重複領域と定義する．また，送信領域（*Send region*）は，受信領域と平行な境界領域と隣接する受信領域の一部と定義する．送信領域と受信領域を順次送受信することで，図 2 の黒抜き部は，破線矢印の方向に移動するため，斜め方向の通信処理を行った場合と等価になる．通信データ量は冗長になるが，斜め方向通信を行う必要がないため，通信処理手順が簡略化され，プログラミングが容易になる．圧力計算部分の通信はこの手順を反復回数分行い，速度計算部分の通信は各時間ステップに 1 回行う．

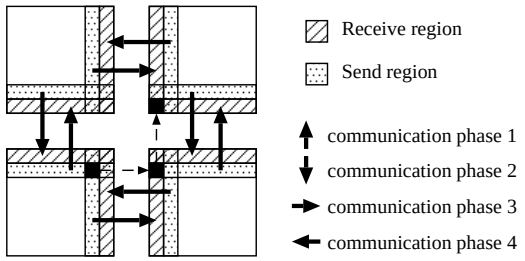

図 2 通信方法
Fig. 2 Communication style.

5. 非同期通信–計算重複法

5.1 計 算 手 順

並列 CFD 計算に適用した非同期通信–計算重複法は，領域の内部を**図 3** 左に示す 3 つの部分に分けて考える．小領域の境界領域（*Boundary region*）の計算は，重複領域（*Overlap region*）と領域内部（*Non-boundary region*）を用いる．境界領域の値が確定したとき，領域内部の計算は，小領域間の依存関係がなくなるため小領域で個別に実行できる．

本稿では，圧力計算の Jacobi 法に非同期通信–計算重複法を適用したが，速度計算の通信回数は，圧力計算の通信回数より非常に少ないため，非同期通信–計算重複法を適用しなかった．

非同期通信–計算重複法の手順は，最初に境界領域の一対の面（たとえば左右の面）を計算し，その結果を

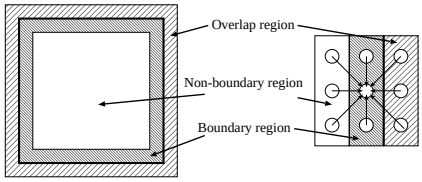

図 3 小領域境界付近の概要
Fig. 3 Neighboring computational boundary.

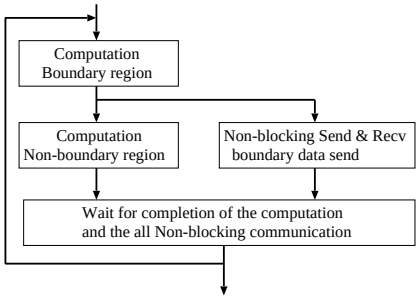

図 4 非同期通信–計算重複法の概要
Fig. 4 The systolic communication-computation overlap method.

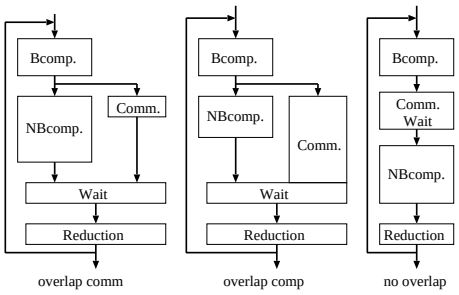

図 5 通信圧縮（左）と計算圧縮（中）および非圧縮（右）
Fig. 5 The overlap communication (left), the overlap computation (middle) and the no overlap (right).

表 2 RS/6000 SP，PC クラスタでの使用 OS とコンパイラ
Table 2 OS and compiler on the RS/6000 SP, the PC cluster.

	OS	コンパイラ
RS/6000 SP	AIX 4.3	XL Fortran 5.1
PC クラスタ	Linux 2.2.14	富士通 Linux Compiler V2

隣接領域へノンブロッキング通信する．この通信開始直後に，二次元以上の分割では，次の面（たとえば上下の面）を計算し，その結果を隣接領域にノンブロッキング通信する．この通信処理と並行に領域内部を計算する．領域内部の計算終了後，通信が終了していれば次反復の境界領域の計算を開始し，通信が終了していなければ待ち状態となる．**図 4** に計算の流れの概要を示した．

5.2 非同期通信–計算重複法の 2 つのケース

理想的な非同期通信–計算重複は，通信処理時間のすべてが計算処理時間に覆い隠される．非同期通信–計算重複法の効果は，以下のモデル式が成り立つことで明らかになる．モデル式は，MPI の待ち処理（MPI_Wait）の影響を仮定した．

非同期通信–計算重複による総経過時間の短縮には 2 つの場合がある．内部領域の計算時間が通信時間よりも長い場合（通信圧縮，**図 5** 左 *overlap comm*）と通信時間が内部領域の計算時間よりも長い場合（計算圧縮，図 5 中 *overlap comp*）である．

通信圧縮と計算圧縮の判定は，非同期通信–計算重複法と同一の計算手順を用いるが，非同期通信–計算重複を行わない並列計算方法（非圧縮，図 5 右 *no overlap*）を用いる．非圧縮の総経過時間 (T_n) を次式に示す．

$$T_n = C + M + R \tag{10}$$

ただし，C は計算時間を示し，$C = C_B + C_{NB}$，C_B は境界領域の計算時間（図 5 Bcomp.），C_{NB} は領域内部の計算時間（図 5 NBcomp.）である．M は通信時間（図 5 Comm.+Wait）を示し，R は，誤差評価の縮約処理時間（図 5 Reduction）を示す．

通信圧縮の総経過時間 (T_m) は次のようになる．

$$T_m = C + W + R \tag{11}$$

W は非同期通信のための待ち処理（MPI_Wait）の時間（図 5 Wait）を示す．W は通常明確にならないが，C_{NB} が M より十分に大きい場合は待ち処理のみの時間として扱える．計算圧縮の総経過時間 (T_p) は次のようになる．

$$T_p = C_B + M + R \tag{12}$$

式 (11)，(12) の関係が成り立つ場合，非同期通信–計算重複法によってデータ通信時間または内部領域の計算処理時間が圧縮され，総経過時間が短縮する．

モデル式の適用は，非圧縮を計測し，通信圧縮と計算圧縮を判断する．非同期通信–計算重複法を適用した並列 CFD 計算プログラムの各成分の経過時間を計測した．

6. 結　　果

使用した並列計算機は，IBM 製 RS/6000 SP（PowerPC 604e 332 MHz，SP-Switch Network，理論性能 150 MByte/s），PC クラスタ[11]（Intel Pentium III 450 MHz，100 Mbps Switch-Hub Network，理論性能 12.8 MByte/s）である．RS/6000 SP，PC クラスタの OS とコンパイラを**表 2** に示した．

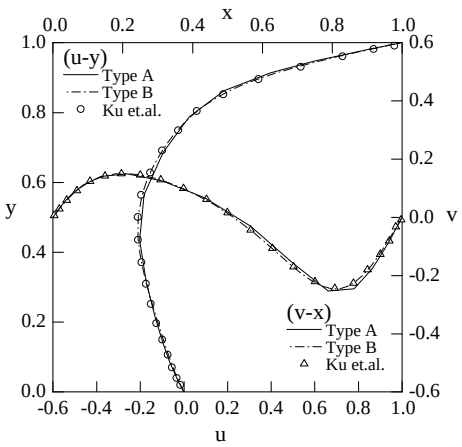

図 6 三次元キャビティー流れの速度プロファイル
Fig. 6 Flow velocity profiles of cubic cavity on vertical and horizontal centerline.

RS/6000 SP では 32 PE，PC クラスタでは 8 PE を用い，コンパイルの最適化オプションは，両者とも最適化レベルが高い-O3 である．

また，並列計算結果とした時間は，以下の方法で選択した．PE 間で総経過時間が最大である PE を選択し，その PE の総経過時間と各成分の経過時間を代表値として得る．同一プログラムを 3 回実行し，総経過時間が最小である PE の代表値を結果として採用した．

6.1 流 れ 場

本稿で用いたプログラムを検証するため，定常状態まで計算した結果を示した．**図 6** に Ku らの結果[14]と表 1 の Type A，B の垂直，水平の中心線上の速度プロファイルを示した．Type A では，格子点数が少ないため精度低下が見られた．また，定常状態における三次元キャビティー内の流跡線を**図 7** に示した．図は流れ方向の上流側から下流側へ斜め下に見下ろしたものである．下流側上面部の中心付近の流れが，上流側の壁に近づくに従って，側壁方向に移動している様子が見られ，三次元キャビティー流れの特性がよく現れている．

6.2 計算負荷率

本稿の CFD 計算を 1 PE で実行した場合の圧力計算部分と速度計算部分の計算負荷比率を**表 3** に示した．

CFD 計算の総経過時間は，圧力計算時間でほぼ 100%を消費している．速度計算時間は 1.0%前後である．速度計算に非同期通信–計算重複を行わなくても総経過時間にはほとんど影響ない．

6.3 分割パターン

計算量が均一となる分割パターンで並列 CFD の計算性能を調べ，最も並列性能が良い分割パターンを**表 4**（RS/6000 SP）と**表 5**（PC クラスタ）に示した．

表 4，表 5 の数字は i，j，k 各方向の分割数である．たとえば 1,1,2 は，i，j 方向では分割なし，k 方向に 2 分割することを意味する．また，非同期通信–計算

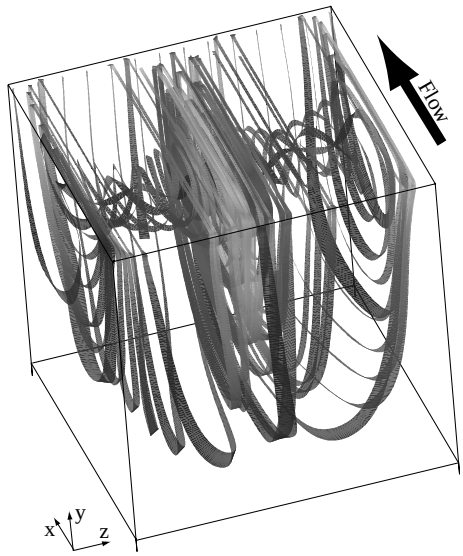

図 7 三次元キャビティー流れ
Fig. 7 The cubic cavity flow.

表 3 計算負荷比率（%）
Table 3 Percentages of total elapsed time contributed by the major processes.

Process	RS/6000 SP		PC Cluster	
	Type A	Type B	Type A	Type B
Pressure Comp.	99.0	99.2	98.9	98.8
Velocity Comp.	1.0	0.8	1.1	1.2
Total	100.0	100.0	100.0	100.0

表 4 RS/6000 SP での領域分割パターン
Table 4 Domain partitioning patterns on the RS/6000 SP.

PE	Type A		Type B	
	normal	*overlap*	*normal*	*overlap*
2	1,1,2	←	1,2,1	←
4	1,2,2	←	1,2,2	1,4,1
8	1,4,2	1,2,4	1,4,2	←
16	1,4,4	1,2,8	2,4,2	1,8,2
32	1,4,8	←	2,4,4	1,8,4

表 5 PC クラスタでの領域分割パターン
Table 5 Domain partitioning patterns on the PC cluster.

PE	Type A		Type B	
	normal	*overlap*	*normal*	*overlap*
2	1,1,2	←	1,2,1	←
4	1,1,4	1,2,2	1,2,2	←
8	1,1,8	1,2,4	2,2,2	←

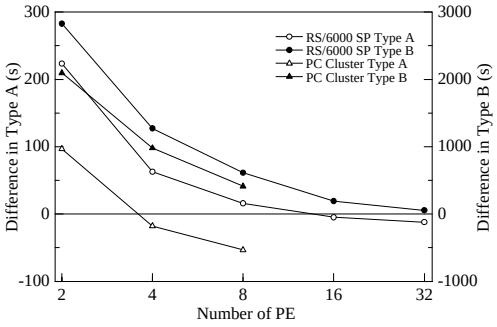

図 8 計算時間–通信時間差
Fig. 8 Difference between computation time and communication time.

重複しない場合を *normal*，非同期通信–計算重複を行う場合を *overlap* とする．ここでの非同期通信–計算重複しない場合は，非圧縮の計算手順ではなく，境界領域と内部領域を分けて考えず，通常の計算順序で計算した後，通信を行うものである．

各 PE 数での分割パターンによる総経過時間の最小値と最大値の差は 2〜4 倍程度になった．RS/6000 SP の場合，問題サイズが大きな Type B の *normal* では，8 PE の場合を除いて 1 回あたりの通信データ量が少なくなる多次元分割である．しかし，問題サイズが小さい Type A や *overlap* では必ずしも 1 回あたりの通信データ量が少ない分割パターンではない．

また，通信処理の影響が少ない *overlap* は通信データ量ではなく内部領域の計算処理性能が全体性能を決めると考えられる．Type A における *normal* も通信データ量の影響が少ないため同様の結果が得られた．

6.4 非同期通信–計算重複

4.2 節で説明した非同期通信–計算重複法の 2 つのケースを調べるため，非圧縮について，領域内部の計算処理時間と通信処理時間を調べた．非圧縮の計算には，表 4，表 5 に示された *overlap* の分割パターンを用いた．

図 8 に，計測によって得られた C_{NB} と T_m の差を示した．問題サイズが小さい Type A で RS/6000 SP の 16，32 PE と PC クラスタの 4，8 PE が計算圧縮になる．また，その他と問題サイズが大きな Type B は通信圧縮になった．

非圧縮の結果から非同期通信–計算重複法を 2 ケースに分け，非同期通信–計算重複法を適用した並列 CFD 計算の各成分の計測を行った．**図 9**，**図 10** は，PE 数を変化させたときの通信圧縮の場合（式 (11)）と計算圧縮の場合（式 (12)）の計測を行い，各成分の経過時間を示した．図には，左側棒グラフに式 (11), (12) の左辺 T_m と T_p を，右側棒グラフに式の右辺の各成分を示した．左側棒グラフは，黒色が T_m であり，白色が T_p である．右側棒グラフは通信圧縮の場合，上から R，W，C_{NB} であり，計算圧縮の場合，上から R，C_B，M である．W は MPI_Wait，R は $MPI_Reduction$ の各サブルーチンの処理時間であり，通信時間 M は，R を除いた総経過時間から計算時間 C を引いた値を用いた．

式 (11)，(12) の非同期通信–計算重複法における仮定である *Wait* を用いることで性能モデル式が図 9，図 10 の各 PE 数における左右棒グラフより成り立つことが分かった．また，通信圧縮と計算圧縮の領域内部の経過時間は，非圧縮の場合と比べ多少遅くなった．非同期通信–計算重複法では，通信処理と計算処理を同時に行うためと考えられえる．

RS/6000 SP，PC クラスタともに待ち処理の時間 *Wait* が大きいことが分かった．待ち処理時間は，RS/6000 SP と PC クラスタともに PE 数の増加にともない減少した．一般に PE 間の同期時間は PE 数の増加に従って長くなると考えられる．しかし，本稿では問題サイズにかかわらず，PE 数が増加すると，待ち処理時間が減少した．

待ち処理（MPI_Wait）のサブルーチンコールが完了する時間を明確にするため追加実験を RS/6000 SP で行った．2 PE 間で一方向の非同期通信処理（MPI_Isend と MPI_Irecv）行い，待ち処理の経過時間の計測を行った．非同期通信のサブルーチンコールと待ち処理のサブルーチンコールの間に CPU 時間を消費する dummy ルーチン（CPU 内部で閉じた計算処理）を加え，dummy のルーチンは，通信処理が完全に終了する以上の経過時間とした．通信処理は完全に dummy の処理に隠蔽されているため，待ち処理は通信データ量を変化させてもほぼ一定であると考えられるが，計測すると待ち処理は通信データ量に応じて増加した．本稿での待ち処理時間の減少の原因は，PE 数の増加とともに減少する通信データ量であると考えられる．

6.5 速度向上比

並列性能評価として速度向上比（*Speed up ratio* = (1 PE の総経過時間)/(並列計算時の総経過時間)）を用いた．**図 11** に RS/6000 SP，**図 12** に PC クラスタの速度向上比を示した．

図 11 に示した RS/6000 SP において，問題量の多い Type B では，*overlap* は *normal* に対して，最大約 14%高速であり，8 PE 以降の並列性能は直線的にのびた．また，問題量の少ない Type A では，性能の

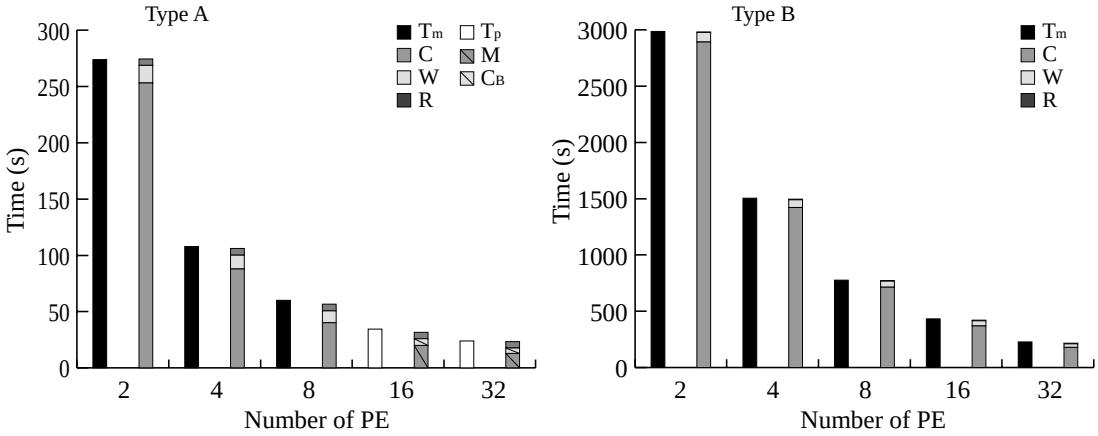

図 9 RS/6000 SP での非同期通信–計算重複の効果
Fig. 9 Effect of the systolic communication-computation overlap on the RS/6000 SP.

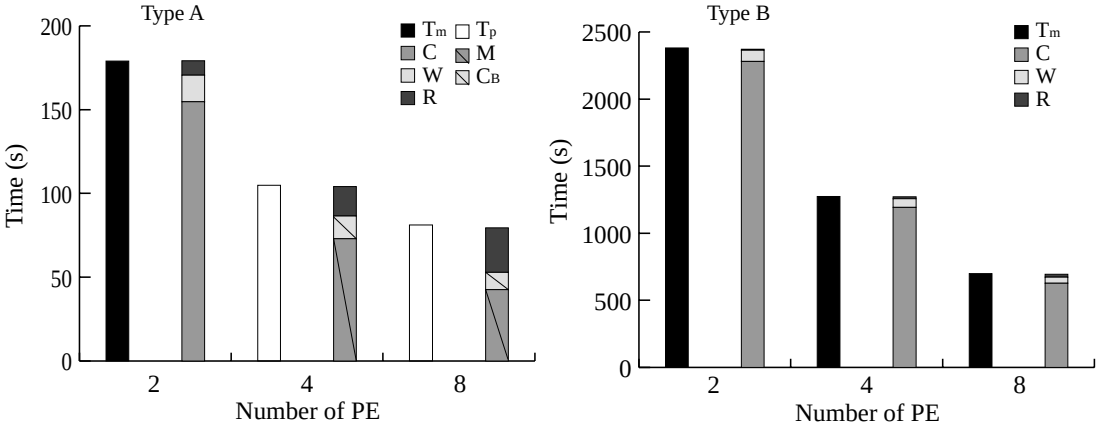

図 10 PC クラスタでの非同期通信–計算重複の効果
Fig. 10 Effect of the systolic communication-computation overlap on the PC cluster.

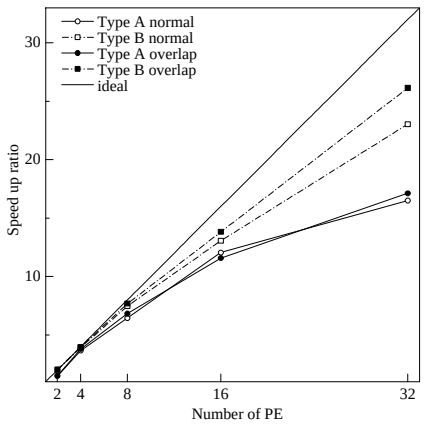

図 11 RS/6000 SP における速度向上比
Fig. 11 Speed up ratio on the RS/6000 SP.

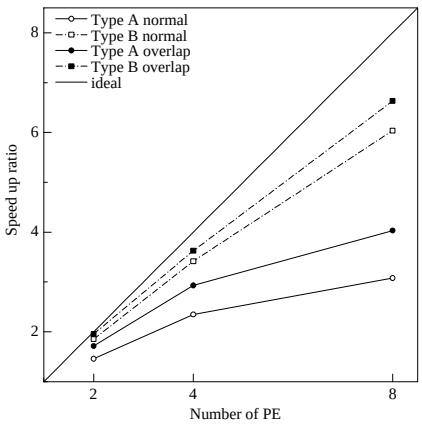

図 12 PC クラスタにおける速度向上比
Fig. 12 Speed up ratio on the PC cluster.

飽和が見られ，*overlap* と *normal* の差が非常に小さい．さらに PE 数が増えた場合，Type B では *normal* と *overlap* の性能差がさらに拡大し，*overlap* の並列性能向上がさらに続くと考えられる．

図 12 に示した PC クラスタでは，Type A で性能の飽和が見られるが，*overlap* は，8 PE においても *normal* ほど大きな性能低下は見られない．Type B では，*overlap* と *normal* ともに並列性能は直線的に推移した．Type A，B ともに *overlap* と *normal* の並列性能差は大きい．Type A では最大 31%，Type B で最大 12%程度の差が見られた．そして，PE 数を増やした場合には，さらに並列性能差が拡大すると考えられる．

7. 考　　察

並列 CFD 計算では，通信負荷の影響をどのように扱うかが重要な問題である．非同期通信–計算重複法はデータ依存性が少ない場合，特に有限差分法の CFD 計算に多く発生する隣接領域のみのデータ依存性を持つ場合，データ通信処理は，計算時間で隠蔽され，通信時間の多くを擬似的に減少できるため非常に有効な手法である．また，本稿で用いた非同期通信–計算重複法は，境界領域の処理を先行して行うという計算順序の変更だけであるためアルゴリズムが単純であり，既存の並列 CFD 計算のプログラムへの適用や新規の並列実装においても有効であると考えられる．

非同期通信–計算重複法は，待ち処理（MPI_Wait）のコストが予想以上に高く，非同期通信–計算重複による効果を抑えている．待ち処理に時間を要する理由は不明であるが，待ち処理の経過時間がなくなれば，非同期通信–計算重複法の効果はさらに高くなる．

図 9，図 10 から非同期通信–計算重複法は，2 ケースの両方の場合において有効である．しかし，図 11 の Type A に見られるように，*normal* と *overlap* の並列計算性能がほとんど変化がない場合がある．原因は，計算処理性能の低下が考えられる．*overlap* は，データアクセスの順番を変更するため，順番にデータアクセスする *normal* に比べて計算処理効率が低下することや通信圧縮によって計算のためのデータ移動と通信のためのデータ移動が同時に発生するため全体的なデータ移動性能が相対的に低下することが考えられる[5]．計算処理効率の低下を抑制できれば，*overlap* の並列計算効率が *normal* より低くならない．また，並列化効率の向上にも役立つ．

図 11，図 12 より，通信圧縮が並列 CFD 計算の全体性能の向上（あるいは性能低下の防止）に有効であることは明らかである．Type B のような問題サイズが大きな場合，さらに PE 数を増やした場合にも高い性能が得られることが予測できる．問題量が少ない Type A でさらに多数の PE を用いる場合，非同期通信–計算重複は計算圧縮となり，通信時間のみが表面化するため通信性能に依存すると考えられる．Type A のような非常に小さな問題サイズを PC クラスタ等の通信性能が低い分散メモリ並列計算機実行しても，並列計算性能の飽和による過剰な効率低下はなかった．

PC クラスタで縮約処理に総和を用いた並列 CFD 計算の結果[15] と比較すると，総和を用いて本稿の Type B の問題量に非同期通信–計算重複法を適用した場合，5.8 倍程度の並列加速率しか得られていない．本稿では，縮約通信に最大値を用いた．最大値を用いた本稿での結果の方が並列計算効率が高くなった．縮約処理コストの抑制を考えた場合，最も良い方法は縮約処理を行わず，収束判定ができることが理想である．しかし，現実的には最大値で収束判定を行い，判定回数を数回に一度の割合に削減することが有効であると考えられる．

8. おわりに

MAC 法による並列 CFD 計算に非同期通信–計算重複法を適用し，並列 CFD 計算の全体性能に対する変化を検討した．比較的大規模な問題に対しては，RS/6000 SP と PC クラスタの両方のハードウェアで非同期通信–計算重複を行うことで十分な効果が得られた．また，非同期通信–計算重複法は 2 つの場合に分けられ，それぞれの場合でモデル式に応じた時間の短縮が行われた．しかし，非同期通信の待ち処理の影響が大きく，全体性能を抑制している．待ち処理の影響は，通信データ量に依存すると考えられ，PE 数が増加した場合，通信データ量は減少し，非同期通信–計算重複法の効果はさらに高くなることが期待できる．

緩和法の基本形である Jacobi 法で十分な結果が得られた．今後，実際に Multi-color SOR 法のようなさらに収束性が良い緩和法に対する検討が重要である．そして，もう一方の通信隠蔽法であるパイプライン処理との計算性能での比較検討が必要であると考えられる．

参 考 文 献

1) Bailey, D., Harris, T., Saphir, W., van der Wijngaart, R., Woo, A. and Yarrow, M.: The NAS Parallel Benchmarks 2.0, NAS Technical Report NAS-95-020, NASA Ames Research

Center (1995).
2) van der Wijngaart, R., Sarukkai, S. and Mehra, P.: Analysis and Optimization of Software Pipeline Performance on MIMD Parallel Computers, NAS Technical Report NAS-97-003, NASA Ames Research Center (1997).
3) Quinn, M. and Hatcher, P.: On the Utility of Communication-Computation Overlap in Data-Parallel Programs, *J. Parallel Distrib.*, Vol.33, No.2, pp.197–204 (1996).
4) Fink, S. and Baden, S.: Non-uniform Partitioning of Finite Difference Methods Running on SMP Clusters. http://now.cs.berkeley.edu/clumps/
5) 田中良夫，松田元彦，久保田和人，佐藤三久：SMP クラスタ上でのリモートメモリ転送を用いた通信と計算のオーバーラップによる性能改善，情報処理学会研究報告，98-HPC-72 (1998).
6) Evans, E., Johnson, S., Legget, P. and Cross, M.: Automatic code generation of overlapped communications in a parallelisation tool, *Parallel Comput.*, Vol.23, No.10, pp.1493–1523 (1997).
7) 荒川忠一：数値流体工学，東京大学出版会 (1994).
8) 川村哲也：流体解析 I，朝倉書店 (1996).
9) 数値流体力学編集委員会（編）：数値流体力学シリーズ 1，非圧縮性流体解析，東京大学出版会 (1995).
10) Thompson, J., Wasri, A. and Mastin, W.: *Numerical Grid Generation: Foundations and Applications*, North Holland (1985).
11) 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，情報処理学会研究報告，2000-HPC-80 (2000).
12) 黒川原佳，松澤照男，姫野龍太郎，重谷隆之：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，第 14 回数値流体力学シンポジウム Web 原稿 (2000). http://wwwsoc.nacsis.ac.jp/jscfd/cfds14/pdf/e08-2.pdf
13) Evans, E., Johnson, S., Legget, P. and Cross, M.: Automatic and effective multi-dimensional parallelisation of structured mesh based codes, *Parallel Comput.*, Vol.26, No.6, pp.677–703 (2000).
14) Ku, H., Hirsh, R. and Taylor, T.: A Pseudospectral Method for Solution of the Three-Dimensional Incompressible Navier-Stokes Equations, *J. Comput. Physics*, Vol.70, pp.739–462 (1987).
15) 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処理隠蔽を行な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 (2000).

(平成 13 年 2 月 8 日受付)
(平成 13 年 5 月 17 日採録)

黒川 原佳

1972 年生．1997 年電気通信大学電気通信学部機械制御工学科卒業．1999 年北陸先端科学技術大学院大学博士前期課程（情報科学）修了．現在同大学博士後期課程在籍．主に並列計算機上での数値流体解析に関する研究に従事．

松澤 照男（正会員）

1948 年生．1973 年信州大学大学院工学研究科修士課程修了．同年信州大学医学部助手．1986 年沼津工業高等専門学校助教授．1991 年北陸先端科学技術大学院大学助教授．1995 年同教授．数値流体力学における並列計算の研究に従事．医学博士．日本機械学会，日本数値流体力学会，日本流体力学会等各会員．

姫野龍太郎（正会員）

1955 年生．1977 年京都大学工学部電気系学科卒業．1979 年同大学院修士課程修了．1979 年から 1997 年まで日産自動車勤務．主に自動車空力の数値解析の研究に従事．1998 年から理化学研究所勤務．現在，情報環境室室長．埼玉大学大学院客員助教授．著書「魔球をつくる」等．工学博士．

重谷 隆之（正会員）

1966 年生．1990 年東京理科大学理学部物理学科卒業．1996 年東京都立大学大学院博士課程修了．1997 年から理化学研究所に勤務．現在，情報環境室技師．理学博士．