

Hitachi SR8000/MPP フルノードによる非圧縮性流れの MPI/OpenMP ハイブリッド並列有限要素解析

奥田洋司

okuda@q.t.u-tokyo.ac.jp

東京大学大学院工学系研究科 システム量子工学専攻

1. はじめに

大規模かつ複雑な問題を高精度・短時間で解析するためには並列処理技術の利用が必要不可欠であり，超並列計算機の最大性能を引き出す計算手法が必要とされている．超並列計算機には大きく分けて，共有メモリ型（SMP型），分散メモリ型，分散・共有メモリ型（SMP Clusters）の三方式があるが，近年は SMP Clusters の形式を取るアーキテクチャが増加してきた．実際 2001 年現在で，世界の超並列計算機の性能上位 20 台中 14 台が SMP Clusters の形態を取っている[1]．MPI/OpenMP のハイブリッド，さらにはそこにベクトル化を含めたプログラミングモデルについて，これまでの研究のほとんどは構造格子による解法であり，有限要素法のように非構造格子を用いた場合の知見が求められている．

本稿では SMP Clusters システムである Hitachi SR8000 及び Hitachi SR8000/MPP を用いた並列数値計算手法を検討した結果について報告する．具体的にはバルクシステムの SR8000 および SR8000/MPP（東京大学情報基盤センター）上で MPI（Message Passing Interface）と OpenMP（A Standard Application Program Interface for Shared Memory Programming）[2] を混在させた Element-by-Element 有限要素法のハイブリッド並列コードを開発し，非圧縮粘性流れ問題の解析を通じて，ハイブリッド並列処理のパフォーマンスを計測し，MPI を用いた分散メモリ型の並列処理のみの場合（フラット MPI）と，ハイブリッド並列処理との並列性能比較を行った．計算は規則的な格子からなるキャビティ流れと不規則な格子からなる室内流れの種類について行い，格子の不規則性が演算速度に及ぼす影響についても検討した[3][4]．さらに，キャビティ流れの大規模問題（約 1,000 万節点）について SR8000/MPP フルノード計算を実施した．

2. MPI/OpenMPプログラミングの方法

SMP Clusters 型におけるハイブリッド並列処理とは Fig.1 に示すように，ノード間を MPI などの分散メモリ型並列処理で，ノード内を OpenMP などの SMP 型並列処理で並列処理することである．このように OpenMP のようなスレッド形の並列処理と MPI のような通信を伴う並列処理を混在するときには，ノード内で並列処理を行う適当なブロックを並列実行した後，マスタースレッドのみがノードを代表して通信を行うことにより，安全な通信が可能となるということに特に注意しなくてはならない．

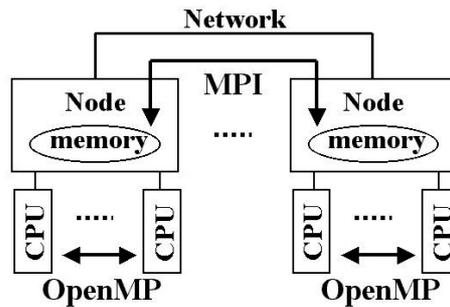


Fig.1 Hybrid parallel programming model

OpenMP は、プログラム中で並列実行の指示があった時点で、複数のスレッドを生成し、並列実行部分が終了した時点でマスタースレッド以外のスレッドは消滅する。したがって、OpenMP ディレクティブで指定される並列に実行される部分領域内に MPI 関数がなければ、MPI 通信は自然にマスタースレッドが行うというスタイルになるため、安全に通信が行える。

MPI と OpenMP によるハイブリッドプログラミングを行う際には、MPI で並列化されているプログラムに OpenMP の並列化ディレクティブを追加することによって並列化を行う。このとき、プログラム中で対象となるループの並列化可能性やプログラムの整合性はユーザーが判断する。また、SMP Clusters がベクトル機の場合は、並列ループ長がパフォーマンスに大きく影響するので、最適なループ長の確保なども考慮に入れる必要がある。

3. 流体解析手法とそのハイブリッド並列化手法の概要 [5]

ナビエ・ストークス方程式と連続の式を、速度に関して 1 次、圧力に関して 0 次の基本関数を用いてガラーキン法で離散化する。また時間方向に速度修正法を用い、速度について陽、圧力について陰に扱う。ソルバーには記憶容量の削減が容易な Element-by-Element CG 法を用いた [6]。Element-by-Element 法では、マトリクス演算を要素単位で行い、それらを足し合わせて全体の結果を得るため全体マトリクスを作成する必要がない。ここで扱った有限要素法非圧縮粘性流れ解析は、圧力ポアソン方程式の解法とナビエ・ストークスの時間進展にかかる計算時間が約 5 割ずつとなっている。この特に計算時間の大半を占める部分を並列化することにより、計算時間の大幅な短縮を図っている。また、圧力ポアソン方程式の解法部分に多くの通信が発生し OpenMP と MPI が多く混在する。

解析領域は、計算に用いるノード数と同じ領域数に領域分割し、ノード間（部分領域間）では MPI を用いた並列計算、ノード内（各部分領域内）の演算は OpenMP による並列計算が行われている。後者は Do ループに OpenMP ディレクティブを挿入して並列に計算している部分であるが、最内側ループのベクトル化を考慮に入れたインライン展開や、メモリの同時アクセスを防ぐ処理も行っている。

4. 解析結果および考察

(1) 小規模ノードによる効率評価

まず、キャビティ流れおよび室内流れの解析を通してパフォーマンスを検討した。また、

比較のために MPI のみを用いたプログラム（以下フラット MPI と呼ぶ）の結果も示す。フラット MPI の場合，同一ノード内のプロセス間でもメッセージパッシングによる通信を行う。ここでの計算は，SR8000 によるものである。SR8000 は 8CPU の SMP 形アーキテクチャを 1 ノードとし，128 ノードを多次元クロスバネットワークで繋いだ SMP Clusters である。演算プロセッサは擬似ベクトル処理機構をもち，1 ノードあたり，8GFLOPS のピーク性能を有する。

キャビティ流れの解析モデルは次の 2 種類（small と large）である：

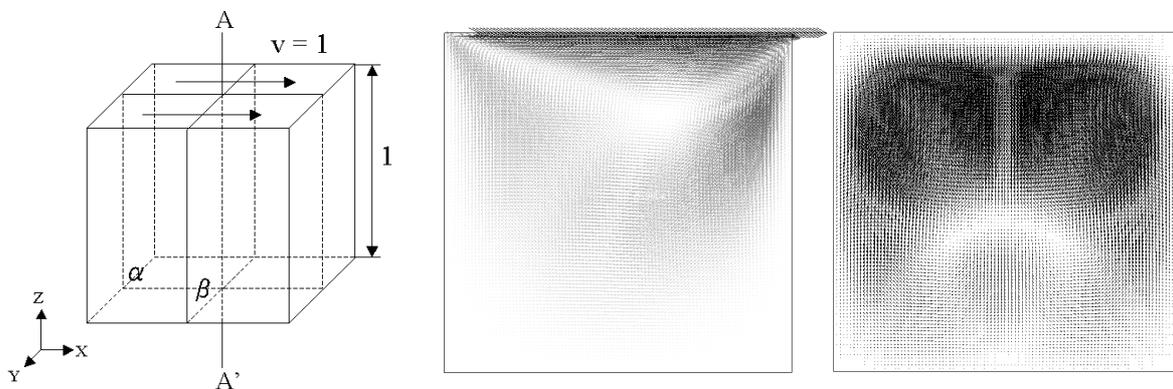
small : 125,000 (50×50×50) 要素 large : 1,000,000 (100×100×100) 要素

キャビティ流れとは Fig.2 に示すように，立方容器内の流体が上壁の動きにより容器内で回転するものである。要素は規則的に分割されている。

また，室内流れの解析モデルは次の 2 種類（small と large）である：

small : 118,504 要素 large : 948,032 要素

Fig.3 に解析形状，境界条件，流れの様子を示す。天井のクーラーからの噴出しと一方の入り口からの流入が，パソコンの載った机や棚が置かれている室内（大学の研究室を想定）を流れる様子を計算している。

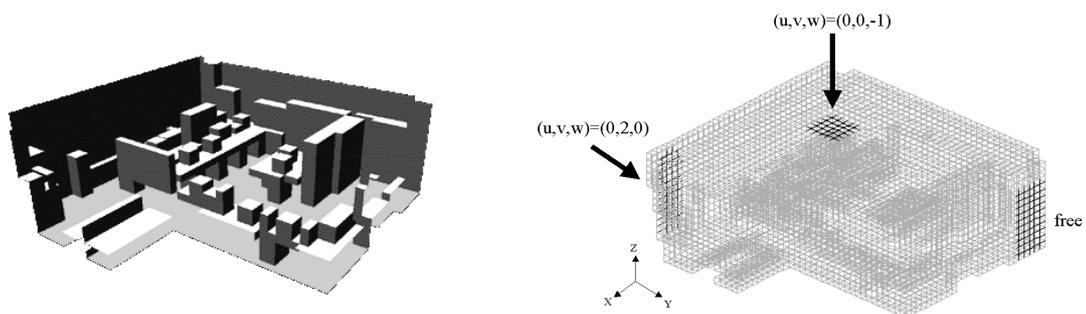


(a) Analysis domain

(b) Velocity on α -plane

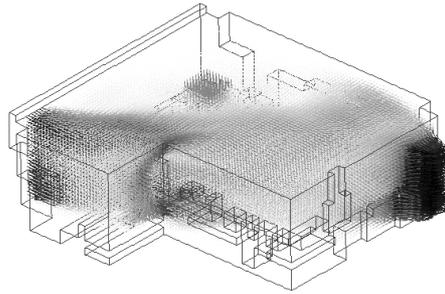
(c) Velocity on β -plane

Fig.2 Cavity flow (Re=100)



(a) Analysis domain

(b) Boundary conditions



(c)Velocity vectors

Fig.3 Flow in a room

Table 1 に 2 ノード 16PE によるハイブリッド並列 (2 ノード間は MPI によるメッセージパッシング, ノード内 8CPU は OpenMP による並列処理) の演算速度を示す. FLOPS 値は両モデルとも約 2GFLOPS 出ており, 2 ノードでの SR8000 のピーク性能は 16GFLOPS であるので, 最大 15% 程度の性能が出ている. メッシュのトポロジーが複雑な室内流れでは, パフォーマンスは多少悪化している. これは, メモリの連続アクセスや負荷バランスがキャビティ流れに比べて劣るためと考えられる. しかしながら, 非構造格子の解法である有限要素法としては, SR8000 の比較的高いパフォーマンスを引き出していると言える.

Table 1 Hybrid parallel using 2 nodes (16 PEs, 16 GFLOPS/peak)

Model		GFLOPS (ratio to peak)
Cavity flow	small (125,000 elements)	2.23 (13.9%)
	large (1,000,000 elements)	2.44 (15.3%)
Flow in a room	small (118,504 elements)	1.32 (8.25%)
	large (948,032 elements)	1.95 (12.2%)

次に, キャビティ流れについて, small, large モデルともハイブリッド並列とフラット MPI で並列性能比較を行った結果を示す. SR8000 の 1 ノード(8PE)~4 ノード(32PE) ノードを用いて 100step を計算する際に要したソルバーの計算時間を用いて並列化の効率比較を行った. その結果を Fig.4 に示す.

small, large モデル共に, フラット MPI とハイブリッドで近い値になっているのが分かる. ここでは SMP Cluster として SR8000 という, いわゆるスーパーコンピュータを使用したので, ハイブリッド, フラット MPI 共に性能差はなかったが, これが SMP PC Clusters 上での計算となるとフラット MPI の場合はノード内の全プロセッサを使うと性能が向上しないことという報告がある[7]. また, MPI による並列計算では入力データを領域分割法によって分割するが, このことにより, 前処理の効果が薄れることが分かっており, 並列計算をするときは領域分割数が少なければいいということが言える. 2 ノード, ノード内 8CPU での並列計算では, フラット MPI の場合は 16 領域に分割しなければな

らないが、ハイブリッドの場合は 2 領域ですむ。これらのことから、SMP Clusters 上での並列計算を考える場合、ハイブリッド並列の形をとることが性能面で最適な並列形態ということが言える。

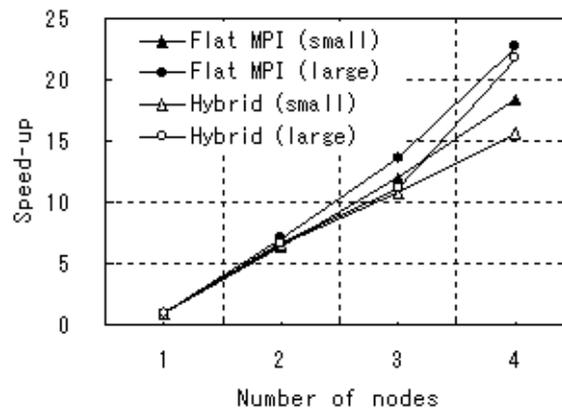


Fig.4 Speed-up : Hybrid vs. Flat-MPI (Cavity flow)

(2) フルノードによるキャビティ流れの大規模計算

Re=1,000, 3,200 のキャビティ流れについて、SR8000/MPP フルノード (128 ノード) による大規模計算を行った。SR8000/MPP のピーク性能は 1 ノードあたり 14.4GFLOPS、フルノードで 1,843GFLOPS である。ここでの解析規模は、10,077,697 (216 x 216 x 216) 要素, 10,218,313 (217 x 217 x 217) 節点 である。解析の諸条件を Table 2 に、解析に要した計算時間や演算速度を Table 3 に示す。

Table 2 Analysis conditions of cavity flow (10,077,697 elements & 10,218,313 nodes)

	Δt	Steps (simulation time)
Re=1,000	0.004	10,000 (t=40)
Re=3,200	0.004	20,000 (t=80)

Table 3 Hybrid parallel cavity flow analysis using 128 nodes
(1,024 PEs, 1,843 GFLOPS/peak)

	Elapse time [s]	Memory [GB]	GFLOPS (ratio to peak)
Re=1,000	2.90x10 ³	105.7	167 (9.05%)
Re=3,200	5.73x10 ³	107.5	169 (9.15%)

Table 3 からわかるように、フルノード計算では対ピーク性能が約 9% となっている。Table 1 に示した対ピーク性能は SR8000 (8GFLOPS/ノード) の 2 ノード計算に対するも

ので直接的な比較は難しい. なお, ここで用いている有限要素法では, 記憶容量削減の工夫を施した手法[5]を用いており, 1,000 万点規模の問題でも 100GB 程度のメモリで済んでいる. SR8000/MPP はノードあたり 16GB の主記憶容量を有しており (SR8000 は 8GB/ノード), 問題規模としてはまだ余裕がある. Fig.5 に $Re=3,200$ の α 面上における流速ベクトルの時間発展を示す. $Re=3,200$ では定常解は得られず, 流れの様子は変化を続ける.

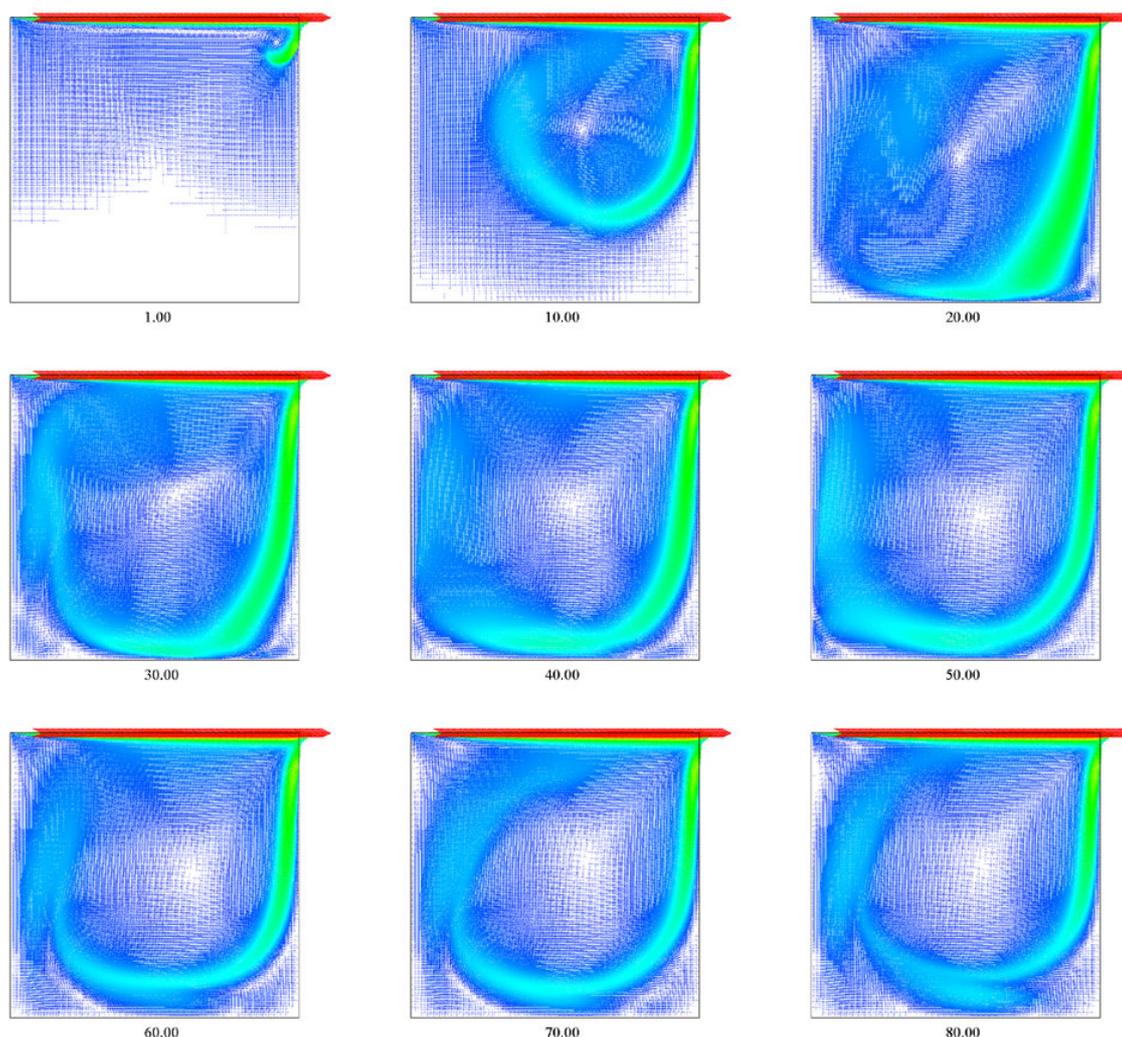


Fig.5 Velocity on α -plane ($Re=3,200$)

5. おわりに

MPI と OpenMP を混在させた Element-by-Element 有限要素法のハイブリッド並列コードを開発し, 非圧縮粘性流れ問題の解析を実施した. まず, SR8000 において, ハイブリッド並列はフラット MPI とほぼ同じ並列性能向上が得られることを確認した. また, SR8000 の 2 ノード計算においてピーク性能の約 15% の演算速度が得られた. 次に, SR8000/MPP において, 1,000 万節点の大規模キャビティ問題のフルノード計算 (128 ノード) を実施した. 演算速度はピーク性能の約 9% であった.

フルノード計算により大規模性、高速性の両面において飛躍的にそのポテンシャルが高まった。対ピーク性能の向上には、PE 単体の演算速度が大きく効いており、今後は単体性能の最適化をさらに追究して全体の速度向上を図る予定である。

謝辞

フルノード計算の機会を提供して頂いた東京大学情報基盤センターならびに有益なアドバイスを頂きました同センタースーパーコンピューティング部門の金田康正教授に、この場をお借りしてお礼申し上げます。

参考文献

- [1] <http://www.top500.org>
- [2] <http://www.openmp.org>
- [3] 奥田洋司, 工藤真吾, 非圧縮性流れの MPI/OpenMP ハイブリッド並列有限要素解析, 第7回日本計算工学講演会, 2002. (発表予定)
- [4] H. Okuda and S. Kudo, Parallel Finite Element Flow Computations for SMP Clusters, 6th Japan US Int. Symp. on Flow Simulation and Modeling, 2002. (発表予定)
- [5] 工藤真吾, 「分散・共有メモリ型並列計算機のための有限要素法の最適化」, 横浜国立大学修士論文, 2002.
- [6] G.Yagawa, Y.Nakabayashi and H.Okuda, Large-scale Finite Element Fluid Analysis by Massively Parallel Processors, Parallel Computing , 23, pp.1365-1377, 1997.
- [7] 吉川ら: SMP-PCクラスタにおける OpenMP + MPI の性能評価, 情報処理学会ハイパフォーマンスコンピューティング研究会 (HOKKE2000), 2000-HPC-80-27, 2000, pp155-160.