特集 15
研究速報

PVMを用いたトルクコンバータ内翼列の乱流解析

Parallel Numerical Calculation of Turbulent Flow in a Torque Converter with PVM

松田進也*・小林敏雄*・谷口伸行*
大島まり*・田坂知寛**
Shinya MATSUDA, Toshio KOBAYASHI, Nobuyuki TANIGUCHI,
Marie OSHIMA and Tomohiro TASAKA

1. 研究の背景と目的

近年，計算規模の増大や問題の複雑化に伴って，計算時間や必要とされるコンピュータ資源はますます大きくなっており，そのため並列計算が有効な方法として注目されてきている．しかし専用の並列計算機はコストも高く，誰もが利用できるという状況ではない．そこで複数のワークステーションやパーソナルコンピュータをネットワークによって接続し，それによって並列・分散処理環境を構築するツールが用いられるようになってきている．これらのツールは情報処理の分野で開発されたもので，数値流体力学に応用された例はまだ少ない．そこで本研究では，それらの並列処理ツールのひとつであるPVMを用いて，トルクコンバータ内翼列流れを計算し，通常のワークステーションやベクトルコンピュータとの比較を行うことによって，ネットワーク上でツールを用いて行う並列計算の有効性を検証した．

2. PVMによる並列化

PVM（Parallel Virtual Machine）[1]は，ネットワーク上でメッセージパッシングを用いて並列処理環境を構築するツールである．PVMはUNIXワークステーションからWindowsマシンまでさまざまな（heterogeneous）機種，OSのコンピュータを扱うことができ，プログラミング言語はC/C++とFORTRANをサポートしている．

しかしこういった並列処理ツールを用いると，コンパイラによる並列化と比べてプログラムやデバッグの複雑化が問題となる．特にヘテロジニアスな環境では，機種依存となる部分は各機種上にそれぞれ異なったプログラムを置かなければならず，デバッグ時にひとつでも古いプログラムを残してしまうと正常に動かなくなる．また同時に複数のコンピュータで複数のプロセスが走るため，エラーの原因を特定するのが非常に困難である．

この点においても，PVMは情報処理の分野で広く使用され，周辺ツールも多いことが注目される．ソースプログラムを自動的に各コンピュータにコピーするツール[2]や，実行時のメッセージパッシングの様子をモニターするツールなどがあるので，それらの問題点には比較的対処しやすいといえるだろう．

並列化そのものに関しても，各プロセスに固有の機能を割り当てる機能並列と，大きなデータ領域を分割するデータ並列の二種類があるが，PVMはデータ並列はもちろん，プロセスの生成場所を指定することでそれぞれのコンピュータが得意とする機能を活用できるため，このどちらにも対応できるようになっている．今回の計算では，トルクコンバータの各構成部品ごとにひとつの計算プロセスを用いており，境界面補間プロセスも別に走っているので，流れの計算そのものについてはデータ並列，計算部と補間部が独立して走っているという点では機能並列となっている．

プロセスの構成も，代表的なマスター・スレーブ型やSPMD（Single Program Multiple Data）型など，プログラミングによって自由に構築することができる．今回は計算全体の始動とユーザーインターフェースとなるマスタープロセスがあり，それが他の計算プロセスを管理する形をとっている．

このように，プログラミングは複雑になるもののそれぞれの計算にあった並列化を行うことができるという高い自由度がPVMの特徴である．計算途中でのコンピュータの追加や削除も可能であるなど，計算の収束状態や研究室の

*東京大学生産技術研究所　第2部
**(株)EXEDY

コンピュータ環境にあわせた対応ができるようになっている．

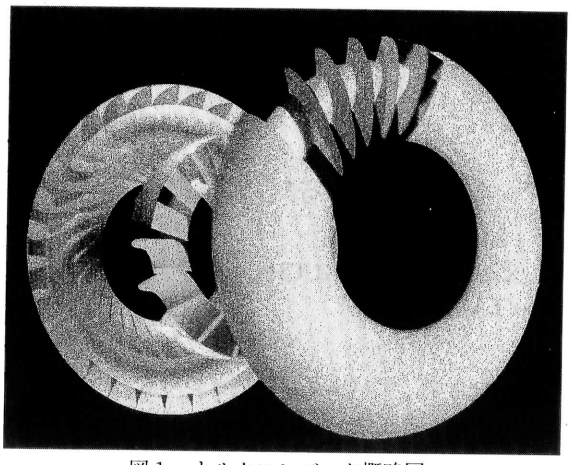

図1　トルクコンバータ概略図

3. 計算対象と手法

今回取り上げたトルクコンバータの概略図を図1，グリッドを図2，計算条件を表1に示す．並列化を行わない

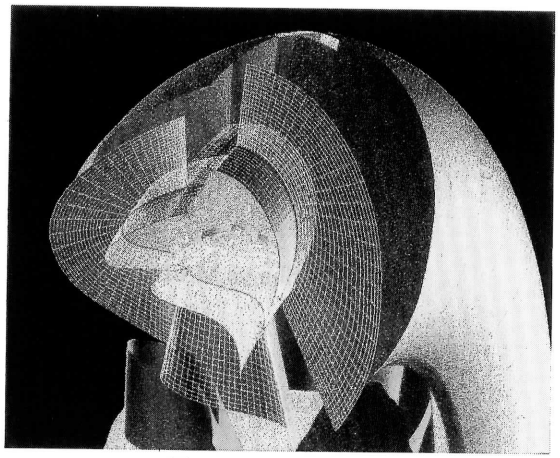

図2　グリッド概略図

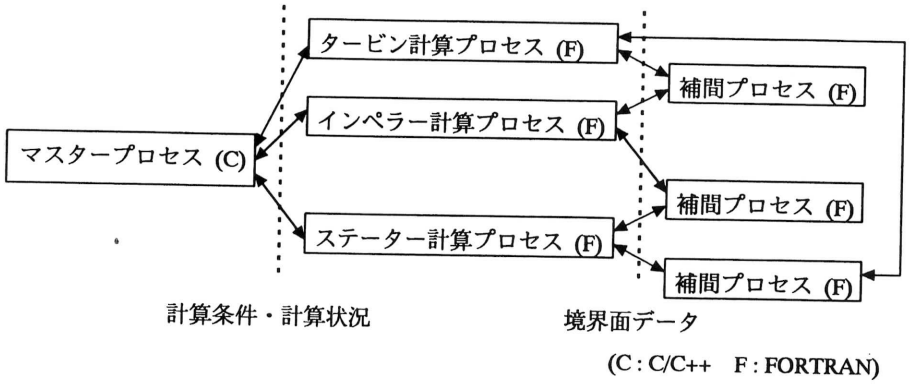

図3　プロセス構成

表1　計算条件

モデル	k－ε モデル
モデル定数	C_μ=1.0, σ_k=1.0, σ_ε=1.3, C_1=1.44, C_2=1.92
アルゴリズム	SIMPLE 法
対流・拡散項	Hybrid スキーム
格子分割数	タービン　60x24x23 インペラー　58x24x23 ステーター　45x24x23

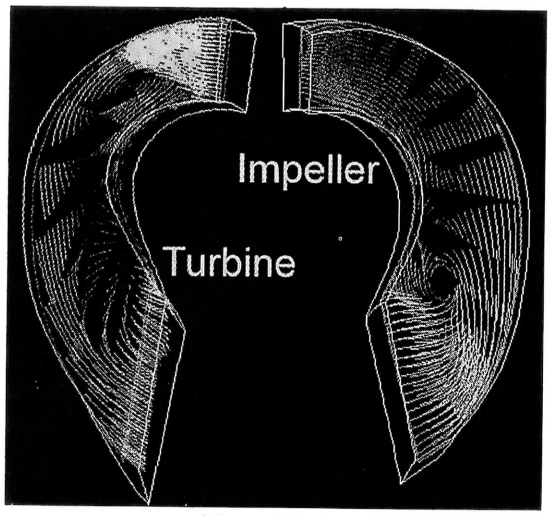

図4　計算結果

場合は，1台のコンピュータ上でトルクコンバータの各部分を順番に計算し，そのつど境界面補間プログラムを実行して境界条件の設定を行う．その際ジオメトリデータなどの各ループごとに再計算される部分が非効率的になっている．

PVMにより並列化した場合には，実際の計算を行うプロセスが6つ及び管理するマスタープロセスが1つの計7つのプロセスで計算が行われる．計算プロセスは，トルクコンバータを構成する3つの要素（タービン，ステーター，インペラー）それぞれについてひとつずつ，またそれらの境界面データを補間するプロセスが（境界面は3つあるので）3つとなっている．各要素間は定常干渉するという仮定により，各要素内は定常流とする．乱流解析にはk-εモデルを適用している．

ユーザーによって始動されるのはマスタープロセスのみで，ネットワークバーチャルマシンの構成（どのコンピュータを計算に用いるか）や計算パラメータの設定を行い，それ以降はマスタープロセスが計算を管理する．6つの計算プロセスの生成，通信経路の確保，各プロセスの初期設定（繰り返し回数，メッセージ送受信での相手側プロセスIDの通知など）を行い，実行状態をモニターする．

マスタープロセスによって生成された計算プロセスは，それぞれ2つのプロセス（計算プロセスは入口/出口両境界面の補間プロセス，補間プロセスは担当する境界面を共有する2つの部品計算プロセス）と速度ベクトル，圧力，k，ε，νの各データのやり取りをPVMによって行いながら計算を進めていく．実行状況や途中の収束状態はマスタープロセスによってモニターされるので，ユーザーはマスタープロセスの走っているコンピュータからすべての計算をコントロールすることができる．

4. 結果と考察

プロセスの構成は図3のようになっており，必要メモリ量は計算部がそれぞれ約40MB（ステーターのみ30MB）である．計算結果を図4，また他の計算アーキテクチャとの比較を表2に示す．計算時間は一台のワークステーションを用いて計算ときの60％程度になっている．シリアルに計算した結果から，計算時間の50％がタービンにかかっているので，並列化の限界も50％ということになり，この場合のオーバーヘッドは10％程度になっている．また今回は5台のワークステーションを用いているが，比較的負荷の少ないステーターやインペラーを計算しているホストでマスタープロセスや補間プロセスを走らせれば，最低3台あれば計算を行うことができる．その場合，同一ホスト上にあるプロセス間では，ネットワークを用いずに直接プロセス間通信をする，データ転送時に共通エンコーディングは行わなくてよい，などの指定をしておけばパフォーマンス低下を抑えることができるだろう．

表2　計算時間の比較

アーキテクチャ	層流	乱流
PVM (Sun SparcStation x 5)	9時間40分	22時間50分
ベクトルコンピュータ (Stardent TITAN 2 CPU)	12時間30分	31時間10分
ワークステーション (Sun SparcStation)	16時間10分	(未計測)

コンパイラが高度な並列化を行う超並列型のコンピュータとは違い，データ送受信の際のオーバーヘッドや，そもそも並列化を人間が行うことの非効率性を考えなければならないこういった並列化ツールの場合は，時間短縮以上に扱える計算規模が大きくなることによりメリットがあると思われる．収束状況を知るために速度などのデータを計算を中断することなく取り出してグラフィクスワークステーションなどで可視化し，それに応じて計算条件を変更するといったことや，格子生成などのプリ・ポストプロセスの段階にも並列化は応用可能である．

並列化は計算規模の増大に伴って不可欠の要素になりつつあるが，並列マシンでなくとも，PVMのようなツールによってネットワークによる並列計算が行えるようになってきている．それをサポートするベンダーやツール類も増えてきており，今後更にこういった計算が行われる機会は多いと思われる．

（1995年11月28日受理）

参考文献

1) Beguelin, A.L., Dongarra, J.J., Geist, G.A., Jiang, W.C., Manchek, R.J., Moore, B.K., Sunderam, V.S., Parallel Virtual Machine System, Univ. of Tennessee, Oak Ridge National Lab., Emony Univ. (1992).
2) Devaney, J.E., Lipman, R., Lo, M., Mitchell, W.F., Edwards, M., Clark, C.W., The Parallel Applications Development Environment (PADE) User's Manual, Computing and Applied Mathematics Lab., Physics Lab., National Institute of Standards and Technology (1995).
3) 田坂知寛，小林敏雄，谷口伸行，池田誠児，“トルクコンバータ内部流れの粘性解析”，自動車技術会学術講演会，1995-5, pp. 25-28 (1995).