

アプリケーションの並列化による効果と作業コストの分析

－ 3次元流体シミュレーションの場合－

長谷部晴美 福井義成

(株) 東芝 総合情報システム部

アプリケーションの立場から、様々な並列処理の適用効果を性能向上に関する効果及び実現のための作業コストとの関係という観点から評価を行った。その例題として、自社開発の3次元流体シミュレーションプログラムを対象として、スーパーコンピュータ上で、オートタスキングとマルチタスキングとで、並列化を行い、それらを比較した。また、問題特性について分析し、その並列化方法についても考察した。その結果、マルチタスキングを利用して空間分割による並列化の場合、最大4倍、平均2.8倍の性能向上が、また、オートタスキングの場合には最大2.3倍、平均1.9倍が実測された。

Analysis of Efficiency and Cost Evaluation on Parallel Computing

Hasebe Harumi Fukui Yoshinari

TOSHIBA Corporation

Total Information and System Division

580-1, Horikawa-cho, Saiwai-ku, Kawasaki, Japan

In order to clear dependency of parallel computing efficient and its cost (man power), evaluated its usefulness on Supercomputer (CRAY Y-MP) by auto-tasking and macro-tasking. As an example, applied parallel to three-dimension fluid simulation program. This paper discusses comparison of these three case, and proposes applicable parallelism for specific of application program.

1. はじめに

近年、並列計算機、並列処理環境の研究が盛んに行われ、それに伴い、並列処理による高速化の事例も多数報告されるようになった。

しかし実際には、全てのアプリケーションがそのような報告例の様に高速化されるとは限らず、多くのコストを費やしても、あまり効果を得られないような例も多くある。現時点の並列化プログラミングには、並列処理の専門性及びアプリケーションの内容に関する専門性が要求されるため、その適用には膨大な知識とコストが要求されることとなる。そのため、対象となるアプリケーションの並列化効果が、そのコストに対し、有用であるという期待がなければ、現場のユーザに普及させることは難しい。すなわち、効果だけではなくそのコストに関する分析も必要となるのである。

アプリケーション側から見た並列処理の実行性能は、分割方法の並列化方針や並列化アルゴリズム、データ配置や分割方法等に依存すると考えられる。また、そのコストについても、対象アプリケーションの特性や並列化に関するそれらの情報があれば、ある程度の予測も可能となる。

本研究では、その様な背景から、スーパーコンピュータ、各種の超並列計算機(MPP)、WSクラスタなどにおける並列化の"コスト"と効果の関係を明確しようという目的のもとで、既存のアプリケーションを例に、スーパーコンピュータやWSクラスタでの並列化効果と適用時の作業コストに関する評価を行うものである。本報告では、その研究の一環として、3次元流体シミュレーションを対象に、スーパーコンピュータ上で並列化に関する評価結果を報告する。

2. 対象アプリケーションと並列化設計

2.1 アプリケーションの紹介

地上付近で破壊的な外出流を起こす強い下流気流であるダウンバーストの3次元シミュレーションを扱う自社開発の流体シミュレーションプログラム[1],[2]を対象として、並列化を行う。このアプリケーションは、地形の影響を考慮できる様な地表面の座標変換を行い、流体方程式を離散化し、陽的解法により解を求めている。シミュレーション空間には直方体を想定し、音波項を含む式

に関しては、計算時間短縮のため、さらに小さい時間刻み幅を適用し、オイラー法、リーブ・フログ法等によって処理している。その処理フローを図1に示す。

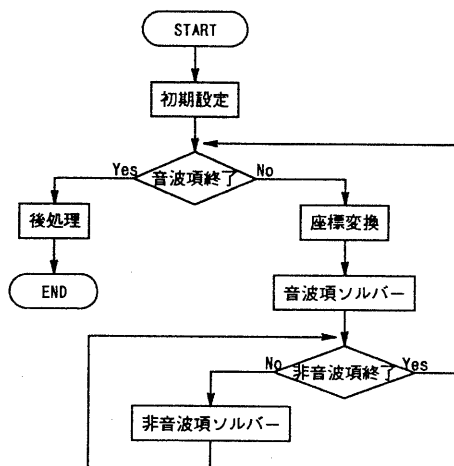


図1. 対象アプリケーション・フロー

2.2 アプリケーション特性の分析

本対象アプリケーションでは、陽的解法による差分法から編微分方程式の解を導くものであり、その時間ステップのループは独立ではないため、並列化の可能性としては、空間分割とループ中の各ソルバーの分割が考えられる。ここでは、その様な並列化の方針を決定するためにアプリケーションの特性を調査する。

各処理過程の計算量を分析するために以下の様な調査を行った。

表2. 処理過程の問題サイズに対する計算量(%)

処理\全格子数	12800	80000	156800
初期設定	0.29	0.23	0.21
座標変換	40.77	44.53	43.76
音波項solver	56.84	53.80	54.56
非音波項solver	1.29	0.96	1.04
後処理	0.81	0.48	0.43

この結果から、各処理に対する問題サイズの影響は、比較的小さいことが分かった。このことから格子数が12800(20×20×32)程度の小さな問題でも高速化の効果が期待できることが予想できる。

また、並列化の対象も座標変換及び音波項solverの処理が全計算量の大部分を占めていることから、これらを対象として並列化を行えば良いことが分かった。次に使用メモリ量についても調べた。

表3. 問題サイズに対する処理時間とメモリ量

調査\全格子数	12800	80000	156800
処理時間 (Sec)	514	1598	2581
使用メモリ (MW)	1.5	7.4	14.2

この結果から、問題サイズにメモリ量が比例して増加することが分かった。粒子追跡法などのシミュレーションプログラムでは、各粒子の追跡時に膨大な粒子データを必要とし、それは粒子数には依存しなかった[3]。その様なケースでは、問題を小さくしても、メモリ量は小さくならない。そこで、その様な問題に対し、並列化処理を行う場合には、並列実行数に比例したメモリ量が要求されることとなる。しかし、今回のケースでは問題サイズがメモリ量に影響していることが明らかであるため、空間分割法で並列化を行う場合に各タスクのメモリ量が分割数に依存して小さくなることが予想できる。そのため、タスク・スケジューリング時の優先度が、アプリケーション全体の処理時間 (TAT) に効いてくる可能性もある。そこで、本評価では空間分割を採用することとした。

2.3 並列化設計と実現方法

シミュレーション空間の部分分割については、Z方向のみの分割とし、図2の様にした。その理由として、次の点を考慮した。

- ・鉛直方向の解像度を上げるため、格子間隔をX、Y方向よりもZ方向は小さく設定する点
- ・鉛直方向は、地形の影響を考慮できる様に各ソルバーを呼び出す前に、座標変換が行われるため、X、Y方向より計算量が多い点

以上の様な並列化方針を元に、モデル化し設計を行っていく。モデル化はオリジナルプログラムを最低限改良することで実現することとする。具体的にはソルバー全体をタスク化し、X-Y境界面でのデータ交換は各時間ステップのループにおいて行うようにする。並列化モデルを適用したフ

ローを図3に示す。

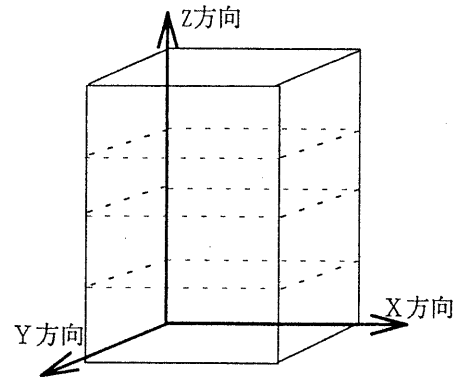


図2. シミュレーション空間の分割

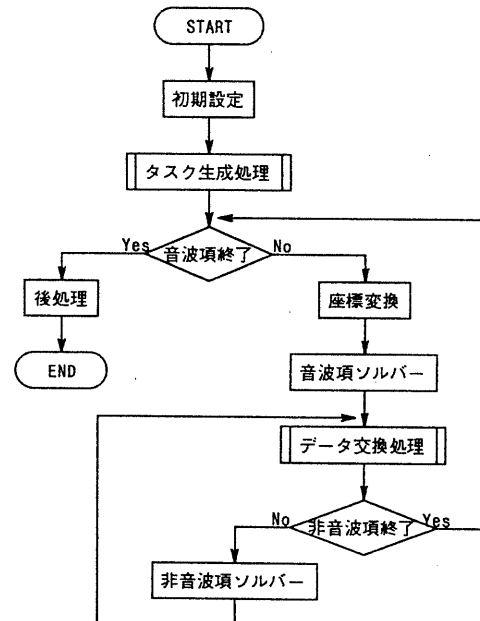


図3. 並列化モデルによるフロー

また、各タスク発生から境界面でのデータ交換の処理をデータフローで図4に示す。

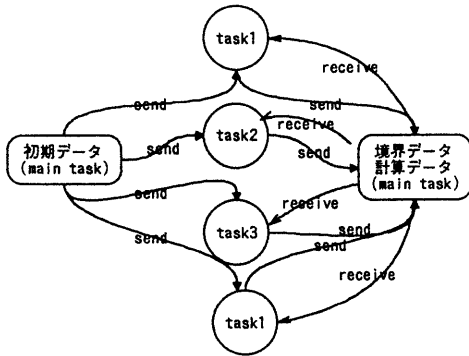


図4. 並列化データフローモデル

3. 性能評価

3.1 評価環境

性能を評価は以下のような環境で行った。実測は、LINPACK値を計測した同等程度の負荷状態でを行った。

表1. 評価マシン

機種名	CRAY Y-MP
メモリ	1GB
CPU数	4
LINPACK値 (100×100)	45.8 MFLOPS (実測)
LINPACK値 (1000×1000)	168.2 MFLOPS (実測)

3.2 実測性能

実測は格子数が12,800から、156,800までの問題を6段階に分けて行った。

また、並列化の効果を正確に把握するため、ベクトル化による効果、ベクトル化及びオートタスクによる効果についても測定した。オリジナル及びベクトル化の測定では1CPUを、オートタスク及びマルチタスクでは4CPUが使用可能であるように設定した。図3にその処理時間の比較を示す。

この実測結果から、ベクトル化とオートタスクでも最大2.3倍程度の高速化が実現されており、細粒度並列化の効果にも期待できることが分かった。また、マルチタスクを適用した場合には、最大4倍、平均でも2.8倍程度の高速化が実現されている。

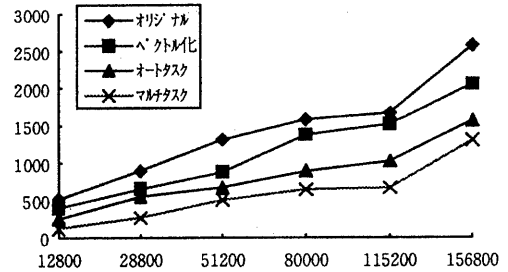


図3. CRAY Y-MPにおける実測性能

また、マルチタスクではメモリ量の最も少ない格子数12800で4倍の性能向上比を実測しており、その向上比は、メモリ量の増大に従って鈍化している。このことから、前述したタスクスケジューリングにおけるメモリ低減化の効果が現れていると言える。また、このことは問題サイズが増大しても、タスク分割を行うことでメモリ量を小さくし、多くのCPUに割り当てることで、より性能向上が期待できることを示唆している。

3.3 コスト分析

3.2の結果から、本対象アプリケーションではベクトル化及びオートタスクを適用するだけでも、平均1.9倍程度の高速化が実現されている。スーパーコンピュータに実装されている自動ベクトル化/自動並列化コンパイラは、高速化に限界があるものの、低コストで実現できる高速化方法としては、大変有効である。特に今回のようにベクトル化及び細粒度並列化に向いているアプリケーションでは、その効果が十分に発揮された。

マルチタスクの適用に関しても、高速化の効果は期待できることが分かったが、領域分割によって並列化を行うには、やはり問題に対する専門性が要求される。そのため、プログラム解析や概要の理解に関する机上の調査にかかるコストがネックとなる。この問題は、HPFのような並列化コンパイラにおいても解決できる問題ではない。そのため、今後もこのような境界データ等のようにタスク間でのデータ交換が必要となる場合には、単純に作業工数だけでは見積もれない。作業者の問題に対する理解度などがそれを左右する。

客観的な見地から、本適用に関する作業コストの分析結果を表4にまとめる。

表4. コスト分析結果

コスト分析項目	分析値
プログラム解析	1.0人月
並列化設計	1.5人月
コーディング	1.5人月
平均速度向上比	2.8
対オートタスク向上比	1.5
全作業コスト	5.0人月

ここでは、作業コストを定量的に示すために、実際に必要となった工数を利用した。全体の作業コストには表4には示されていない作業を含んでいる。この結果から、得られる速度向上比が、このコストとトレードオフになるかどうかを類似アプリケーションを所持したユーザが、判断することになる。

4. 結論

今回、対象としたアプリケーションは並列化の効果だけではなく、ベクトル化による効果も高いプログラムであったため、比較的、高速化の実現は容易であった。また、低コストに実現可能なオートタスク適用の効果も有用であることが分かった。

しかし、このような3次元流体シミュレーションにおいて、もっと多くの計算量を必要とする場合には、それに応じたプロセッサ資源を使用して、その効率を最大限に活かせる様な並列処理を行う必要がある。そのため、プログラム解析のコストが増大しても、ソルバーの並列化を行うことよりも、空間分割を行って領域の計算を並列化させる方がもっと多くの計算量を必要とする場合には有効となる。本評価の結果でも、この問題ケースが空間分割で十分に高速化可能であり、メモリ量の最適化によって、さらに性能向上が可能であること等が実証できた。

また、このアプリケーションについては、WSクラスタでの並列処理についても評価を行っている。その場合には、特に通信量とそのタイミングを考慮してモデル化し、最適な分割方法についても考察する予定である。

5. 謝辞

本研究に対して、アプリケーションのご提供及びご指導、ご支援頂いた機械・エネルギー研究所の堀端主任研究員に感謝致します。

参考文献

- [1] 堀端康義 (株)東芝 機械・エネルギー研究所
"積雲対流の一生のラジ・エディシミュレーション"
日本流体力学会誌, 1989
- [2] 堀端康義 (株)東芝 機械・エネルギー研究所
"ダウンバーストの数値シミュレーション - 擬似圧縮法とタイム・スプリッティング法 -"
第7回数値流体シミュレーション講演論文集, 1993
- [3] 長谷部晴美, 福井義成 (株)東芝
"PVMのアプリケーションへの適用効果 -モンテカルロプログラムMCNPの場合"
Swapp琉球'94 HPC-52-17, 1994
- [4] エリ・ラングウ, イェリブツ 東京図書
流体力学 1, 2