汎用可視化ソフト AVS/Express の並列化

平澤 和之¹⁾, 吉川 慈人²⁾, 宮地 英生³⁾, 宇川 彰⁴⁾, 朴 泰祐⁴⁾, 梅村 雅之⁴⁾

Parallelization of AVS/Express

Kazuyuki Hirasawa, Yoshihito Kikkawa, Hideo Miyachi,

Akira Ukawa, Taisuke Boku, and Masayuki Umemura

ABSTRACT

To visualize large data at high speed, parallelization is required. We developed parallelized volume rendering module in the visualization software AVS/Express. In general, it is not easy to change existing source codes. In this paper, the volume rendering module is parallelized in three ways: auto-parallelization, OpenMP and pthread. We compare these methods in parallelized performance and difficulty of the coding.

Keywords: Visualization, Parallelization, Volumerendering

1. はじめに

近年, マルチ CPU や大規模並列コンピュータの普及により大規模データによる数値シミュレーションが行われるようになった。これに伴い並列化による可視化処理の高速化が望まれている。また, 並列化のための環境も整いつつある。そこで, 汎用可視化ソフト AVS/Express の並列化をいくつかの手法で行い処理速度の向上やコーディングの容易さについての比較検討を行う。今回は計算速度がかかり並列化をすることにより計算時間の向上が見込めるボリューム・レンダリングを例に取り上げる。また, オリジナルのソースコードになるべく手を入れないでいかに並列化するかという点に主眼をおく。

2. 並列化の手法

並列化の手法としてコンパイラによる自動並列化, OpenMP 指示文を使用した並列処理, スレッドプログラム・ライブラリ Pthread を使用した並列処理¹⁾の3つの手法を用いて比較した。自動並列化とはコンパイラのオプションを指定することによりコンパイラ自身に並列化を行わせる, 手法としては一番簡単な方法である。OpenMP と Pthread は並列化可能なループに命令文を挿入することにより並列化を行う。特に, Pthread は並列化のためのメモリ領域の確保²⁾など行わなければならない

ないので今回取り上げた手法の中では一番難しい手法である。

3. 評価

今回の報告では各手法で並列化されたボリューム・レンダリングの計算速度の比較をした。計算機は SGI Origin-2000 の 8 CPU³⁾ を使用し, データは AVS/Express のサンプルデータである“lobster”(120×120×32のボクセルデータ)を使用した。計算速度を Fig.1 に示す。この計算速度はボリューム・レンダリングのルーチンがコールされる直前とコールされた直後の時間である。また, 表示結果を Fig.2 に示す。

Fig.1 Rendering Time.

Fig.1よりPthreadの場合1CPUから2CPUになると計算速度は半分になる。しかし, CPUの数が増えるにしたがってCPUの数に対する計算処理速度の効率が悪くなる傾向があるが, 最終的には8CPUで約1/7.1とほぼ

* 原稿受付 2001年5月11日

1) 正会員 (株) ケイ・ジー・ティー テクニカルサービスセンター(〒160-0022 東京都新宿区新宿 2-8-8 とみん新宿ビル, E-mail: k-hirasa@kubota.co.jp)

2) (株) ケイ・ジー・ティー テクニカルサービスセンター

3) (株) ケイ・ジー・ティー ビジュアライゼーション事業部

4) 筑波大学 計算物理学研究センター

期待どおりの性能が得られた。

OpenMP の場合でも 8CPU で約 1/6.9 と Pthread と同様な傾向が見られた。OpenMP の場合 Pthread の時と比べて計算時間自体が約 3 倍掛かるという結果になった。また、1～3CPU の時の計算速度が並列化を行っていない場合より計算時間が掛かってしまうのでかなりパフォーマンスが悪くなってしまっている。

自動並列化では期待していたほどの効果が現れず、並列化を行わないときよりも計算時間がかかってしまった。

以上をまとめとして処理速度とコーディングの関係について Table.1 に示す。

Table.1 Speedup and Ease of coding.

	Auto Parallel	OpenMP	Pthread
Calculation speed	Slow	Medium	Fast
Coding	Easily	Little Difficult	Difficult

Fig.2 Volume rendering of volume data of lobster.

4. 考察

比較検討の結果、Pthread による並列化では、CPU の台数に見合うだけの効果が得られた。

OpenMP に関しては 4CPU 以上の時に並列化の効果を発揮することができた。しかし、3CPU 以下では期待していたほどの並列化の効果は得られなかった。原因として、CPU の数が少ない場合、OpenMP のオーバーヘッドは、おそらく「OpenMP そのもののコード実行準備やスレッド準備を行うオーバーヘッドが、並列化によって得られるゲインを上回っている」ということが推察できる。また、OpenMP では一般的に、Pthread よりも余分な同期操作が含まれることが多い（特に大きなループの中で複数のループが個別に並列化されている場合）ため、その余計なコストが計算時間に反映された可能性が高いと推察される。この件については調査を行う必要がある。

自動並列化に関しては、今回ソースコードのチューニングを行わずコンパイルオプションのみで並列化を行ったが、それだけでは並列化の効果は受けられず、かえって並列化のためのオーバーヘッド分、余計な計算時間が掛かる結果となった。この件についても原因の調査を行う必要がある。また、調査の結果次第ではソースコードを並列化しやすいようにチューニングすることも視野にされる。

5. おわりに

ボリューム・レンダリングのソースコードの変更をなるべく行わず並列化により計算処理速度の向上を図ったが、現段階としてソースコードの変更が必要な Pthread が、並列化の効果が一番あった。OpenMP による並列化や自動並列化の場合に関してはパフォーマンスが出ていない部分について調査をおこない、その原因を明らかにする事である。

今後の課題として、今回は 8CPU のマシンでの検証であったが、16CPU や 32CPU など CPU 数が増えた場合どのぐらいでパフォーマンスが出るかの見極めを行うことが必要である。また、ボリューム・レンダリングと同様に処理時間が非常にかかる Isosurface（等数値面生成）についても並列化を行う予定である。

謝 辞

今回の報告に関して種々のご検討をいただいた筑波大学計算物理学研究センターの中本泰史氏、板倉憲一氏（現 原子力研究所）、農業生物資源研究所の沼寿隆氏、および筑波大学計算物理学研究センターの関係者各位に厚く御礼を申し上げます。

参 考 文 献

- 1) 沼寿隆, 松原正純, 板倉憲一, 朴泰祐: 並列入出力機構を用いた可視化システムの提案, 情報処理学会研究報告, 99-HPC-77-10(1999), pp.53-58.
- 2) 板倉憲一, 朴泰祐, 松原正純: 汎用可視化ツール AVS/Express の並列化とその性能評価, 情報処理学会研究報告, 2000-HPC-82-31(2000), pp.179-184.
- 3) 板倉憲一, 朴泰祐, 松原正純: AVS/Express におけるボリューム・レンダリングおよびデータ入力の並列化, 第 6 回ビジュアライゼーションカンファレンス(2000), http://www.kgt.co.jp/avs_conso/event/vc6/data/1-2.pdf