

並列入出力機構を用いた可視化システムの提案

沼 寿 隆† 松 原 正 純†
板 倉 憲 一†† 朴 泰 祐†

数値解析を並列計算機で行ない、計算の終了後に全データをまとめてフロントエンドのマシンにファイル転送して可視化を行なおうとすると、データ転送や可視化処理にかかるコストが膨大になり、これらの処理が困難になってしまう。そこで我々は、このような問題を解消する並列可視化システムの開発を行なっている。本稿では、その第一段階として実装したシステムについて報告する。

このシステムは、並列計算機の生成するデータを並列ネットワークを用いて随時オンラインでグラフィック・ワークステーションに転送し、連続的に可視化するものである。並列ネットワークを用いることによってデータ転送を高速化し、また、可視化部分を業界標準の可視化ソフトウェアである AVS/Express を基本に構築し、その上に実装した並列ボリュームレンダリング・モジュールによって画像生成を高速化している。超並列計算機 CP-PACS と並列ワークステーション Origin2000 及び Onyx2 を用いて本システムを実装した結果、並列データ転送及び並列ボリュームレンダリング処理により、AVS/Express の枠組の中で高速な処理が実現できることが分かった。

Developing Parallel Network-based Visualization System

HISATAKA NUMA,[†] MASAZUMI MATSUBARA,[†]
KEN'ICHI ITAKURA^{††} and TAISUKE BOKU[†]

In this paper we propose a parallel visualization system for large volume of data as a result of numerical calculation. In ordinary way, these data are once completely stored in a file, then processed for visualization. However these processes stress the data transfer and visualization costs to make them hard to process.

This system, based on a de facto standard AVS/Express, handles a direct data transfer from a parallel numerical engine to a graphic workstation via parallel network channels and real-time visualization. The advantage of this system is as follows: First, reducing data transfer time with multiple network channels; Second, real-time monitoring which can visualize intermediate results without waiting for the finish of a computation; Third, accelerating creation of image with parallel volume rendering. We implemented this system on CP-PACS, a massively parallel processor, and two parallel workstations, Origin2000 and Onyx2. As a result of basic performance evaluation, the system achieves high processing performance with parallel channels and parallel volume rendering.

1. はじめに

直に観察することのできない自然界の現象を数値データとして計算機に取り込み、解析する方法は計測機器の高度化により一般的なものになっている。また、計算機の処理能力の向上によって、より実世界の問題に近い数値シミュレーションが可能になってきている。これら二つの解析に共通していることは、得られる大規模な数値データから対象の構造や振舞いを効果的に

解析する必要があるということである。そのためには、データを直観的に理解しやすい形にすること、つまりビジュアライゼーション（可視化）が必須であると言える。特に大規模数値シミュレーションによって得られるデータを可視化する場合、数値シミュレーションを並列計算機で行ない、計算の終了後にその結果をグラフィック・ワークステーションにファイル転送して可視化を行なう場合が多い。しかし、この方法では大量の数値データの転送に大きなコストがかかり、またそのデータ・サイズが巨大であるために、可視化処理そのものが困難になってしまう。そこで我々は、超並列計算機 CP-PACS のプロトタイプ機 Pilot-3 と共有メモリ型並列計算機である SGI Onyx2 及び Origin2000 を用いて、大規模数値計算の結果として得られる大量のデータを効率良く可視化するためのシステムを開発

† 筑波大学 電子・情報工学系

Institute of Information Sciences and Electronics, University of Tsukuba

†† 筑波大学 計算物理学研究センター

Center for Computational Physics, University of Tsukuba

している。

本稿では、その第一段階として実装した並列可視化システムについて報告する。このシステムは、Pilot-3の生成するデータを随時オンラインでOnyx2またはOrigin2000に転送し、連続的に可視化するものである。つまり、解析計算と可視化処理を同時並行的に行なうシステムである。また、ボトルネックになると思われるデータ転送部分に関しては、並列入出力機構¹⁾を用いることによって高速化を図っている。なお、可視化部分は業界標準の可視化ソフトウェアであるAVS/Express²⁾を基本としている。

2. 実装の方針

始めに、これまで我々が行なっていた可視化の手順とその問題点を挙げ、次にそれらを解消するためには実装するシステムにどのような機能が必要であるかを述べる。

2.1 これまでの可視化手順とその問題点

これまで我々は、画像生成までの可視化処理を以下に示す手順で行なっていた。

- 数値シミュレーションを計算サーバである超並列計算機 CP-PACS、もしくはそのプロトタイプ機 Pilot-3で行なう。
- 計算終了後に、その解析結果(数値データ)を一度ひとつに束ねてから、Onyx2にファイル転送する。
- Onyx2上のAVS/Expressでそのファイルを読み込み、可視化を行なう。

以上のような手順でデータの可視化を行なっていたため、次のような問題点が指摘されていた。

- (1) 計算の途中で中間結果を取り出して可視化することができないため、計算の妥当性をリアルタイムに検証することができない。
- (2) 計算終了後、膨大なデータの可視化を一度に行なうため、数値データの一括ファイル転送にかかるコストが大きい。また、その大量の数値データを可視化するための処理も膨大になる。
- (3) 並列計算されたデータを内蔵のハードディスクに格納し、それらを束ねて外部に転送するため、そこで逐次処理が生じデータの並列性を活かさない。

1については、もし計算が破綻した場合に中間結果をリアルタイムにモニターできていれば、その破綻の原因を分析しやすくなるといった利点があり、このような機能が求められる。また、プロセッサの性能向上や扱えるメモリ量の増加により、複雑なシミュレーションができるようになったが、その反面、出力される計算結果が膨大になり、結果として2のような問題が出てきている。3については、並列チャンネルを用いて並列にデータを外部に転送することで、この問題を解消するシステムの開発が同時に進められている。このシ

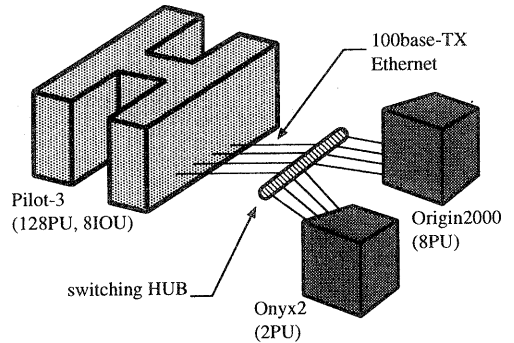


図1 システムの全体構成

ステムについては後で概略を述べ、これを取り入れた可視化システムの構築を行なう。

2.2 実装方針

先に挙げた3つの問題点を考慮すると、我々が開発するシステムに必要な機能は主に次の3つであると言える。

- シミュレーション結果の転送に関して、中間ファイルを生成せずに、オンラインでグラフィック・ワークステーションに転送する。
- 中間結果を随時読み込み、それらを連続的に可視化する。
- 計算データの並列性を活かすため、データの転送と可視化処理もできる限り並列に行ない、画像表示までの時間を短縮する。

これらを考慮した上で、システム的设计・実装を行なった。その詳細については次節以降で述べる。なお、今回実装したシステムはAVS/Expressを基本としている。これは、AVS/Expressが業界標準の可視化ソフトウェアであり、また、これまでのユーザが今回実装する可視化システムへの移行がしやすいとの理由からである。

3. 全体構成

システムの全体構成を図1に示す。解析計算サーバは、超並列計算機 CP-PACS のプロトタイプ機 Pilot-3 であり、128 台の演算プロセッサと 8 台の入出力プロセッサから成る。そして、Pilot-3 の生成するデータを SGI Origin2000 (8PU) と Onyx2 (2PU) で処理する。可視化は、並列ビジュアライゼーション・サーバである Onyx2 で行なうが、並列ファイル・サーバである Origin2000 を用いることも可能である。また、これらのマシンはそれぞれ 4 本の 100base-TX で接続されており、並列なデータ転送が可能となっている。

4. 並列可視化システムの実装と評価

可視化システムの構成を図2に示す。計算サーバは

Pilot-3 (Data Generation)

Onyx2 (Visualization)

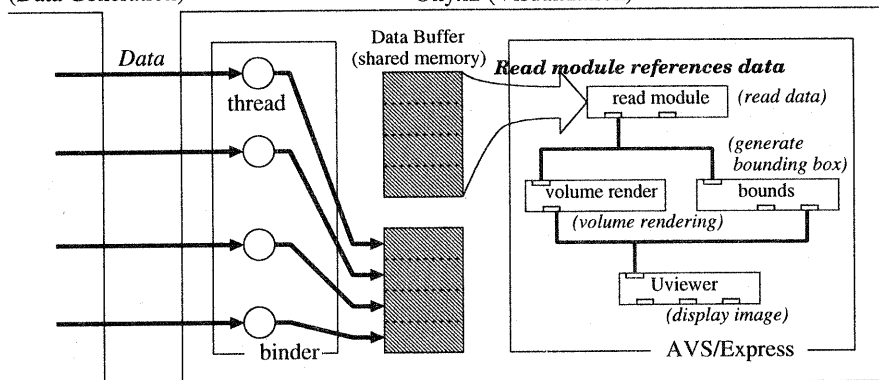


図2 並列可視化システムの構成

Pilot-3 であり、数値シミュレーションの結果を Onyx2 に随時転送しながら可視化を行なう。本システムは、Pilot-3 上での数値シミュレーションとその結果の可視化を同時並行的に行ない、並列に計算されたデータを並列に転送し、可視化処理もできる限り並列に行なうことを目標としている。

まず始めに並列なデータ転送を実現する並列入出力システム (PIO) の概要を述べ、次に AVS/Express から PIO を利用するために実装したモジュールについて説明する。また、AVS/Express 上に実装した並列ボリュームレンダリングについても述べる。

4.1 PIO：並列入出力システム

並列システム間で何らかの処理を行なう場合、その間のネットワークが単一であると、並列に処理されたデータが一度ひとつにまとめられることになる。このような場合、データをひとつに束ねるといった処理が全体のボトルネックとなり、データの並列性を最大限に利用することができない。また、単一の入出力プロセッサにデータが集中するため、複数の入出力プロセッサを備えるマシンでは、それらを有効に活用できないといった問題も挙げられる。

そこで、Pilot-3、Origin2000、Onyx2 間を並列ネットワークで接続し、並列なデータ通信を可能にする並列入出力システム (PIO)¹⁾ の開発が進められている。PIO の提供する API を用いることにより、アプリケーション・プログラムから直接外部のマシンへ並列なデータ転送が可能になる。PIO 上では、ユーザはチャンネルの本数や負荷分散等を特に意識することなく、これを利用することが可能である。

4.2 PIO + AVS/Express

可視化部分は AVS/Express を基本に構築した。AVS/Express は MVE (Modular Visualization Environments)³⁾ と呼ばれる汎用可視化ソフトウェアの一つであり、可視化の各ステップに相当する処理が

モジュールとして独立しており、それらを組み合わせることによって一連の可視化処理を行なうものである。また、標準モジュールの提供する機能以上の処理を行ないたい場合には、ユーザ独自のモジュールを作ることができる。

PIO の性能を最大限に活かすため、AVS/Express へのデータ入力は、中間ファイルを生成せずに直接並列チャンネルからのデータ入力を可能にしなければならない。しかし、AVS/Express はデータの入力が同じシステムのファイルからしか行なえない閉じたシステムである。よって PIO を用いたとしても、一度中間ファイルを生成して、そこからデータを読み込まなくてはならない。そこで、この問題を解消するため、並列チャンネルからのデータを直接読み込むことのできるモジュール (リード・モジュール) を AVS/Express 上に実装した。今回新たに作成したモジュールの具体的な機能は、PIO システムを利用することによって並列にデータを受け取り、それを AVS/Express の内部データ構造にセットすることである。これにより、計算サーバからの並列なデータ入力が可能になる。このモジュールは、それ以降に続く処理を行なうモジュールに対し、従来のファイルからのデータ読み込みモジュールとの互換性を保っている*。

しかし、並列チャンネルからのデータの読み込みと、内部構造へのセットという二つの機能を一つのリード・モジュールに持たせるとある問題が生じる。それは、AVS/Express がマルチ・プロセッサに対応していないため、リード・モジュールが並列チャンネルからデータを読み込んでいる間は、レンダリング等の AVS/Express 内部の画像生成のための処理が行なえなくなることで

* ただし、完全な互換性は保っていない。今回実装したモジュールに備わっている機能に関しては互換性があり、従来モジュールとの置き替えが可能である。

ある（その逆も同じである）。これは、時系列データを連続的に読み込み、それらを連続的に可視化する場合に処理時間の面で大きなマイナスになってしまう。

そこで以上のような問題を解消するため、並列チャンネルからのデータ入力部分をリード・モジュールの機能から分離させ、AVS/Expressとは別のプロセス（binderと呼ぶ）として実装した。これらの動作は以下になる。

- binderはマルチスレッド化されており、並列に入ってくるデータを並列に受け取り、それらをデータ・バッファに書き込む。
- リード・モジュールは、binderが書き込んだデータを読み込み、AVS/Expressの内部構造にセットする。

このように、データの読み込み処理を二つのプロセスに分担させ、これらをOnyx2上で並列実行することによって、AVS/Expressがあるステップのデータを処理しているのと並行して、binderが次のステップのデータを読み込むことが可能になる。また、binderとリード・モジュール間のデータ受け渡しは共有メモリを介して行なわれるため、余計なデータ・コピーは生じない。さらに、この二つのプロセス間で共有しているデータ・バッファは多重化されていて、リード・モジュールがデータを読み込んでいる間に、別のデータ・バッファにbinderがデータを書き込めるといった処理のオーバーラップが可能になっている。

4.3 モニタリング機能

計算サーバの出力するデータを連続的に読み込み、連続的に可視化する機能を持たせた。これにより、解析計算の中間結果をリアルタイムに監視することが可能になった。また、計算サーバから送られてくるデータを一定の手順に従って連続的に可視化するだけではなく、視点の移動や断面の表示などといった各種表示法を利用することもできる。これについては、シミュレーション・プログラムを止めることなく、可視化システムのGUIを介して随時切り替えることが可能である。

4.4 並列ポリウムレンダリング

ポリウムレンダリングは、3次元スカラー・データを半透明表示することによって、データの内部構造を直観的に見せる有効な可視化手法であるが、多くの計算時間を必要とする。特に、ポリウムレンダリングを用いて時系列データを連続的に可視化する場合、レンダリング処理がボトルネックになってしまい、並列にデータを入力している利点を活かすことができない場合が考えられる。そこで、このような問題を避けるために並列ポリウムレンダリング・モジュールをAVS/Express上に実装した。アルゴリズムはレイキャスティングを用いている。また、並列化にはpthreadを使い、投影するスクリーンのスキャンラインをサイクリックにスレッドに割り当てて、レンダリングを行

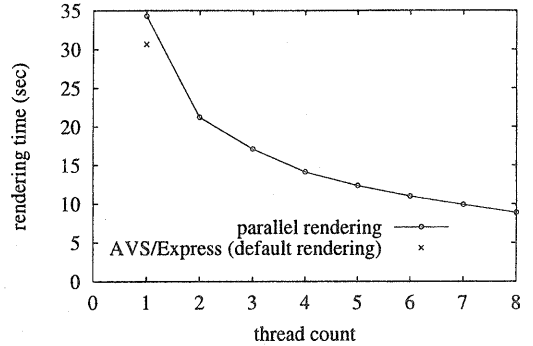


図3 レンダリング時間

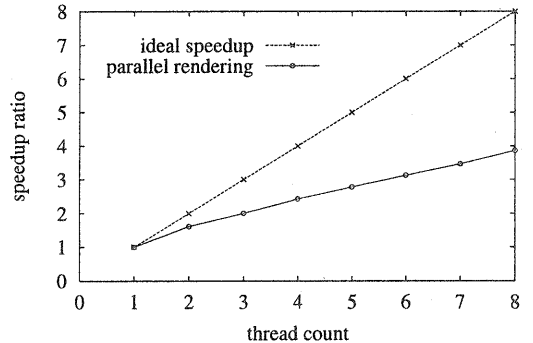


図4 速度向上率

なっている。

今回実装した並列ポリウムレンダリング・モジュールとAVS/Expressの標準のポリウムレンダリング・モジュールの性能評価を行う。評価はOrigin2000(8PU)上で行ない、CP-PACSで計算した宇宙初期の水素ガスの電離状態を表わすデータ⁴⁾(128^3 格子)を 500×500 サイズのスクリーンにレンダリングする。結果を図3および図4に示す。スレッド数を1から8まで変化した場合の評価を行なっている。また、表1はスレッド数が1, 2, 4, 8のときのレンダリング時間の内訳を示している。表中の“shading & classification”は、ボクセル(raw data)をカラーと不透明度に変換する処理を表わし、“ray casting”は変換されたデータをサンプリングして画像を生成する処理を表わしている。

図3から分かるように、並列化していない場合にはレンダリングに約30秒ほどかかっている。これは、今回用いたデータは空間コヒーレンスが低いため、階層的ボクセル集合⁵⁾を用いた高速化手法が効きにくいためであると思われる。

8スレッドの場合、AVS/Express標準のポリウム

表1 レンダリング時間の内訳(単位は秒)

thread count	1	2	4	8
shading & classification	8.6	4.2	2.2	1.1
ray casting	24.5	15.5	11.4	6.3

レンダリングと比べて実行時間が約3分の1になっており、並列化した効果は出ていると言える。しかし、図4を見ると速度向上率はそれほど良くない。これは、表1から分かるように、ray castingに時間がかかっているためである。shadingとclassificationは配列データに連続アクセスを行ないながら処理を進めて行くが、ray castingは配列データにランダムアクセスをするような処理を行なう。特に今回使用したデータのように、ほとんどすべてのボクセルをサンプリングしなければならない場合には、キャッシュ・ミスの頻発によりメモリネックとなり、それが性能を低下させている原因になっていると思われる。また、プログラムのチューニングを充分に行っていないことも理由の一つだと思われるので、この改善は今後の課題である。

4.5 ケース・スタディ

本節では実際のアプリケーションを想定し、本システムを用いて可視化を行なう場合の有効性と問題点を、特に速度の面から検証して行く。

アプリケーションは、分子動力学法(MD)シミュレーション・プログラム⁶⁾を対象とした。検証は、このプログラムをPilot-3の128ノードを用いて実行し、本システムを用いてシミュレーションと同時に並行的に各ステップのデータを連続で可視化する場合を想定する。なお、シミュレーションおよび可視化の条件は以下のように設定した。

- 分子数131072の液体アルゴンのシミュレーション。
- 1ステップ計算する度に分子の位置座標を転送し、可視化を行なう。1ステップ当たりのデータ・サイズは3MB。
- 可視化側のマシンはOrigin2000。
- 可視化側は分子の位置座標を密度分布(64³格子)に変換し、ボリュームレンダリングを行なう。
- 500×500サイズのスクリーンに描画する。

以上のような可視化を行なう場合、データ生成から描画までがパイプライン的に流れて行くので、最も処理に時間のかかる部分が全体の速度を決定する。ここでは描画までの処理を(1)データ生成、(2)データ転送、(3)ボリュームレンダリングの3つに分けて考える。1ステップ当たりのデータ生成には0.5秒かかり、データ転送に関しては、PIO(4チャンネル)を使うことによって10MB/sのスループットを得ることができる。それに対して、AVS/Expressの標準のボリュームレンダリング・モジュールでは、今回用いたMD計算データにおいて、1ステップのレンダリングに35秒かかり、ここがボトルネックになるのは明らか

である。よって、このようにデータ生成間隔の短いアプリケーションに関しては、今回実装した並列ボリュームレンダリング・モジュールが有効であると言える。このモジュールを使って、8スレッドでレンダリングを行なった場合には、1ステップのレンダリングは7秒で終了する。標準モジュールに比べ大幅な時間短縮ができるが、これでもなおデータの生成スピードには追い付けない。したがって、並列ボリュームレンダリング・モジュールのさらなる高速化が必要である。

5. 今後の展望

今回実装した可視化システムは、必要最小限の機能しか備えていない。したがって、機能の拡充はもちろん、使い易いインタフェースの開発が急務である。また、PIOを超並列計算機CP-PACSに実装する計画が進められており、これまで以上にデータの生成速度およびPIOの転送速度に見合った高速な可視化処理が必要となってくる。これらを考慮した上で、以下に示すような機能拡張を検討している。

- 並列ボリュームレンダリング・モジュールの高速化: データの生成間隔が比較的長いアプリケーションに対しては現在のままで充分だが、前章で検討したようなデータ生成間隔の短いアプリケーションに対応するためにも、並列ボリュームレンダリング・モジュールのさらなる高速化を行なう。
- データの領域抽出機能: 本システムでは、1ステップのデータ全体をPilot-3からOnyx2に転送して可視化を行なっている。したがって、データの一部分を取り出し、その部分だけを可視化する場合に画像生成に寄与しないデータも転送することになる。容量の大きいデータに対しては、このような無駄なデータ転送がボトルネックになる場合が考えられるので、Pilot-3側で必要なデータだけを抽出して、余計なデータを転送しないようにする機能が必要である。そして、このような操作をAVS/ExpressのGUIから動的に変更できるようにする。
- 可視化パラメータの保存: 可視化を行なうためには、対象の構造や振舞いが的確に見えるようにパラメータを調整しなければならない。そこで、一度目の可視化の時に第何ステップ目でどのパラメータを変更したか、などといった情報を保存しておき、再び同種の解析計算の結果を可視化する場合にそれを利用できるようにする。また、現在描画しているデータよりも前のステップのデータを幾つか保存しておき、プレイバックをできるようにする機能も検討している。

6. 関連研究

AVSを用いた並列可視化の例を挙げる。いずれの例

も本システムのような並列入出力は用いておらず、この点が最も大きな差異であると言える。また、多くのシステムでは、モジュールの分散化が比較的容易だった一世代前の AVS5 を用いているのに対し、我々のシステムは最新の AVS/Express を用いており、厳しい制約の中で分散化・並列化を実現している点が大きく異なる。

AVS/PVM (Syracuse Univ.USA, PARMA Univ.Italy) ⁷⁾

並列計算機やワークステーション・クラスタなどの計算用のノードとファイル・サーバからなる分散環境で数値解析とその結果の可視化を行ない、ホーム・マシン上の AVS でそれらの制御と画像の表示を行なう。

AVS/PCM (Australian National Univ.) ⁸⁾

AVS における特定の可視化処理をバックエンドの並列計算機で実行する。AVS モジュールとして実装されているので、通常のモジュールと同様な操作で扱える。

VisLink (富士通) ⁹⁾

解析プログラムに埋め込むライブラリとそれに対応する AVS モジュールから成り、TCP/IP で解析計算の中間結果をオンラインで AVS に転送し、リアルタイムな可視化を行なう。

並列計算機 SR2201 によるトラッキングステアリングの計算力学への適用 (電力中央研究所, 日立製作所) ¹⁰⁾

システム構成はネットワーク接続された SR2201 と Onyx2, ビジューライザに AVS を使い, SR2201 による数値解析の結果をリアルタイムに可視化する。可視化処理を Onyx2 側だけで行なうのではなく、並列ボリュームレンダラ、並列レイトレーサが SR2201 上に実装されており、可視化処理の分散を行なっている。また、AVS の GUI から SR2201 上の数値解析の条件やパラメータを動的に変更することができる。

7. おわりに

本稿では、現在我々が開発を進めている並列可視化システムについて述べ、実際のアプリケーションを想定して、本システムを用いて可視化処理を行なう場合の有効性について検討を行なった。

本システムは、並列計算機の生成するデータをオンラインでグラフィック・ワークステーションに転送し、リアルタイムな可視化を行なうことを目的として開発が進められている。その特徴としては、まず並列入出力システムを用いてデータ転送を高速化している点が挙げられる。これにより、並列に計算されたデータを一度ひとつに束ねてから転送するといった無駄がなくなっている。また、可視化部分には業界標準の可視化ソフトウェアである AVS/Express を採用しており、

中間ファイルを生成することなく、ダイレクトに並列データ流を受け取ることができるモジュールを実装した。さらに、並列ボリュームレンダリング・モジュールを AVS/Express モジュールとして実装することで、データ生成から描画までの処理を可能な限り並列に行なうことが可能になった。

今後の課題としては、並列ボリュームレンダリング・モジュールの高速化、ユーザ・インタフェースの整備、機能の拡充などが挙げられる。また、実際のアプリケーションに合わせたシステムのチューニングを施して行く予定である。

謝 辞

本研究において、貴重な御意見を頂いた筑波大学計算物理学研究センター未来開拓プロジェクトのメンバーに感謝致します。なお、本研究の一部は日本学術振興会未来開拓学術研究推進事業「計算科学」による。

参 考 文 献

- 1) 松原 正純 他「コモディティネットワークに基づく並列入出力システム」情報処理学会研究報告, HPC76-1, pp.1-6, (1999) .
- 2) <http://www.kgt.co.jp/kgt/avs/>
- 3) H.D. Lord, "Improving the Application Development Process with Modular Visualization Environments", *Computer Graphics*, vol.29, no.2, pp.10-12, May 1995.
- 4) Umemura, M., Nakamoto, T. & Susa, H. 1999 "3D Radiative Transfer Calculations on the Cosmic Reionization" in *Numerical Astrophysics*, eds. S. M. Miyama, K. Tomisaka, T. Hanawa, (Kluwer Academic Publishers, Dordrecht), pp.43-44
- 5) M. Levoy, "Efficient Ray Tracing of Volume Data", *ACM Trans. on Graphics*, Vol.9, No.3, pp245-261, July, 1990.
- 6) 松原 正純 他「超並列計算機 CP-PACS における大規模分子動力学法シミュレーション」情報処理学会論文誌, Vol.40, No.5, pp.2172-2181, 1999.
- 7) G.Cheng et al. "Developing Interactive PVM-based Parallel Programs on Distributed Computing Systems within AVS Framework," In *Proc. of the 3rd Annual International AVS Conference, AVS'94, Boston, MA, May 2-4.*
- 8) http://www.mcnc.org/HTML/ITD/IAC/AVS95_www/avs95/netnews/sharrott/sharrott.html
- 9) http://www.fujitsu.co.jp/hypertext/Products/Info_process/hpc/topics/vislink/index.htm
- 10) 「並列計算機 SR2201 によるトラッキングステアリングの計算力学への適用」日経サイエンス 1999 年 5 月号 pp.2