ハイパフォーマンス
コンピューティング 82−31
（2000. 8. 5）

汎用可視化ツール AVS/Express の並列化とその性能評価

板倉憲一† 朴 泰祐†† 松原正純††

本研究では，数値計算と可視化処理を同時並行的に実行するシステムを開発した．実験システムとして，数値計算サーバに超並列計算機 CP-PACS，可視化サーバに共有メモリ型並列計算機 SGI Onyx2 および Origin2000 を用い，CP-PACS と可視化サーバ間のデータ通信に我々が開発しているコモディティネットワークによる高スループットな並列入出力システムを用いている．本システムは，CP-PACS が生成するデータをオンラインで SGI WS に転送し，リアルタイムな可視化を行なう．一方，可視化部分とフロントエンドの GUI は，業界標準の可視化ソフトウェアである AVS/Express 上に構築し，汎用性と実用性を保ったまま並列データ流の取り扱いや並列ボリュームレンダリングを行うモジュールを実装した．実際に本システムを用いた数値計算と可視化処理の同時実行を評価した結果，高スループットな並列入出力と並列可視化ツールが有効に働くことが示された．

Performance evaluation of parallelized visualization tool AVS/Express

KEN'ICHI ITAKURA,† TAISUKE BOKU†† and MASAZUMI MATSUBARA††

In this paper we developed a parallel visualization system for large volume of data as a result of numerical calculation. We implemented this system on CP-PACS, a massively parallel processor, and two parallel workstations, Origin2000 and Onyx2. In order to improve the network bandwidth between CP-PACS and workstations, we utilize our developed parallel I/O system which are based on commodity network. This system handles a direct data transfer from a parallel numerical engine to a graphic workstation via parallel network channels and real-time visualization. We developed some modules on a de facto standard AVS/Express for base system to create a volume rendering image in parallel. As a result of performance evaluation, the system achieves high processing performance with parallel channels and parallel volume rendering.

1. はじめに

大規模並列計算機やワークステーション・クラスタを用いた数値シミュレーションは，宇宙物理学，生物学，構造力学，気象学などの様々な分野で必要不可欠な手法として定着している．特に現在では，プロセッサの処理能力の向上や扱えるメモリ量の増加により，より実世界に近い規模の数値シミュレーションが可能になっている．しかし，その結果として得られる数値データはあまりにも膨大であるために，それらをただ眺めるだけではそこから何の理解も得ることはできない．したがって，このようなデータの洪水から必要な部分を取り出し，そこに潜む構造や振舞いを効果的かつ直観的に理解することは重要な課題であると言える．

† 筑波大学 計算物理学研究センター Center for Computational Physics, University of Tsukuba
†† 筑波大学 電子・情報工学系 Institute of Information Sciences and Electronics, University of Tsukuba

数値シミュレーションとそのデータの可視化処理は独立しているため，従来は数値シミュレーションを並列計算機などで行ない，その全工程終了後に，数値データをフロントエンドのグラフィックワークステーションに一括転送して可視化処理を行なう場合がほとんどであった．今後も計算機の性能は向上し続けると考えられ，これまで以上に大規模な数値データを解析しなければならなくなる．その時，すべての数値計算の終了後に可視化処理をまとめて行なっていたのでは，大規模データの一括ファイル転送にかかるコストが大きい，巨大なデータを一度に可視化しなければならない，数値計算が破綻していた場合にまた始めからやり直さなければならない，などあらゆる面で効率が悪い．そこで，数値シミュレーションと可視化処理を統合した環境を作り，これら二つを同時実行することによって，可視化の結果から得られた知見をすぐさま数値シミュレーションに反映させることが，大規模データをより効果的に解析するためには必要であると思われる．

本研究では，数値計算と可視化処理を同時並行的に

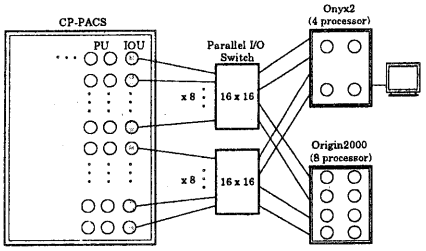

図1　可視化システムのハードウェア構成

実行するシステムを開発する[1]．数値計算と可視化処理を同時並行的に行なうためには，データ生成から最終描画までを高速に行なう必要がある．そのため本システムでは，数値計算サーバが生成したデータを並列ネットワークを用いて可視化サーバへ並列転送する．さらに，可視化処理もできる限り並列化することで全体の処理を高速化する．

2. 並列可視化システムの構成

システム構成を図1に示す．本システムは，数値計算を行なう超並列計算機 CP-PACS[2]，可視化を行なう並列グラフィックワークステーション SGI Onyx2，ファイルサーバである SGI Origin2000[3]，及びこれらを接続する並列ネットワークで構成されている．

これまでは超並列計算機 CP-PACS で数値計算を実行し，すべての計算の終了後に全データを SGI WS にファイル転送をして可視化処理を行なわなければならなかった．これに対して，本システムでは計算と可視化を同時に行い，その途中のデータ転送及び可視化処理をできるだけ並列化する．

2.1 並列入出力システム (PIO システム)

CP-PACS の 128 台の入出力用プロセッサのうち，16 台が 100Base-TX Ethernet インタフェースを備え，Onyx2 と Origin2000 も 100Base-TX Ethernet インタフェースをそれぞれ 4 チャネル備えている．これらを 2 台の 16 ポートスイッチでトランク接続し，高スループットネットワークとして利用する．この並列ネットワークを最大限有効に活用するため，CP-PACS と SGI WS の間のデータ通信には我々の研究室で開発している並列入出力システム（PIO と呼ぶ）[4] を用いる．PIO 上ではユーザはチャネルの本数や負荷分散など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．

2.2 可視化ソフトウェア AVS/Express

可視化処理の各ステップをそれぞれモジュールとして実装し，このモジュールをノードとするネットワークで可視化処理全体を表現することができる．この考えに基づいて設計されたのが，MVE （Modular Visualization Environments）[5] と呼ばれる汎用可視化ソフトウェアである．処理のモジュールは多数用意されており，画面上でネットワークを作るヴィジュアルプログラミングによって可視化処理の方法を決定する．代表的な MVE としては，AVS/Express[6),7)] があり，本研究で実装する並列可視化システムにこれを用いる．AVS/Express では，標準モジュールの他に Fortran, C, C++などの言語を使って新しいモジュールを開発することが可能である．

2.3 並列入出力システムと AVS/Express の融合

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 の標準モジュールには単一のファイルから入力データを読み込むものしかない．そこで，PIO の並列チャネルからのデータを直接読み込むことのできるリードモジュールを AVS/Express 上に実装した．このモジュールは，従来のファイルからデータを読み込む標準モジュールとの互換性を保つようにし，それ以降に続く処理は標準のモジュールが使用できる．

リードモジュールの具体的な機能は，PIO を利用した並列なデータの受け取りとそれを AVS/Express の内部データ構造に変換することである．ここで，AVS/Express は全体で一つのプロセスとして動作するため，この 2 つの機能がパイプライン的に処理できない．そこで，リードモジュールでは UNIX SystemV 共有メモリのデータを内部データ構造に変換する部分だけを担当し，PIO 並列チャネルからのデータを共有メモリに書く処理は，AVS/Express とは別のプロセス（以下 Binder と呼ぶ）として実装した．Binder はマルチスレッド化されており，並列にデータ流を受け取り，そのまま並列にデータバッファである共有メモリに書き込む．これにより，AVS/Express の可視化処理と Binder のデータ読み込み処理がパイプライン的に処理される．

3. 可視化処理の高速化

ボリュームレンダリングには様々な手法が提案されているが，本研究ではレイキャスティング[8] と呼ばれるアルゴリズムを採用した．レイキャスティングは，高画質な画像を得られるアルゴリズムとして知られており，また，最も多く使われているアルゴリズムである．

ここでは，並列ボリュームレンダリングモジュールに実装した 2 つの高速化手法について説明する．

3.1 Spatial Data Structure

レイキャスティングのアルゴリズムでは，不透明度が 0（完全に透明）のサンプル点はピクセルのカラー決定に際して何の影響も与えない．つまり，透明な点をサンプリングせずに済ませば大幅に計算量を削減することができる．そこでボクセルにスキップフラグを用意し，不透明度が 0 のときにフラグを立てる．さらに，このボクセルをリーフとした八分木データ構造

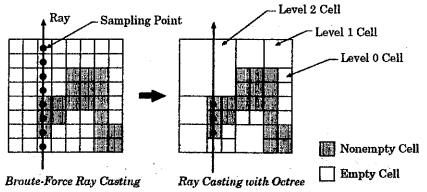

図2 Octreeを用いたレイキャスティング

を作成し，再帰的に子ノードのスキップフラグの論理積をノードのスキップフラグとしてセットする．レイキャスティングの際には，8分木のトップノードからスキャンの対象とし，スキップフラッグがセットされているノード以下のボクセルはサンプリングに意味が無いのでレイを一気にジャンプさせる．

データに依存するが，このアルゴリズを適用した場合には，単純なレイキャスティングよりも5倍以上高速化することが報告されている[9]．

3.2 Shading Lookup Table

輝度値を求めるためには，ボクセル点での gradient を使用する．この gradient は3次元のベクトルであるが，これを高々 $2^{13} = 8192$ 通りに分類し，整数値でエンコードする．そして gradient に対する計算を予め行い，整数値をインデックスとしてこの計算値を保持する lookup table を作成する．

レイキャスティング実行時にボクセルの輝度値を計算する時にはボクセル点での gradient をエンコードし lookup table から計算に必要な値を取り出して使用する．

3.3 並列化手法

2次元のスクリーンをスキャンライン単位で分割し，それをサイクリックにプロセッサに割り当てる．つまり，プロセッサは自分が担当するスキャンラインを描画することになる．これは，隣り合うスキャンラインはほとんど同じ画像になる，という仮定に基づいた方法であり，サイクリックに割り当てることによりプロセッサ間の負荷分散が行えると考えている．また，スクリーンを分割するこの手法では，各プロセッサが行なう処理は完全に独立しており，さらに，ボクセルデータに対しては読み出し処理しか起こらないので，レンダリング実行時にプロセッサ同士の同期による待ち時間は発生せず，高速な処理が可能である．

以上の方法は，静的に負荷分散を行なう方法であるが，dynamic task stealing[10] のように動的負荷分散を行なう手法も提案されている．動的負荷分散は静的負荷分散に比べると，処理は複雑になるが負荷のバランスは良いため，大規模並列化には適している．しかし，本研究で可視化処理に利用できるのは Origin2000 の8プロセッサが最大なので，静的負荷分散でも充分性能が出せると考え，これを採用した．なお，上記のアルゴリズムは C 言語とスレッドプログラミング・ライブラリである pthread を用いて実装している．

以上の機能を付け加えた並列ボリュームレンダリングモジュールを AVS/Express に実装し，標準モジュールと同様な操作を可能にした．また，レンダリングに必要なパラメータについては，そのすべてをシミュレーションプログラムを止めることなく，GUI を介して動的に設定・変更することが可能である．

4. 性能評価

4.1 並列ボリュームレンダリング単体

本節では，本研究で実装した並列ボリュームレンダリングのアルゴリズムの性能評価を行なう．この評価は AVS/Express 上に実装した並列ボリュームレンダリングモジュールとしての評価ではなく，アルゴリズムそのものの評価である．

Onyx2（4CPU, 250MHz）と Origin2000（8CPU, 195MHz）それぞれで，ボリュームデータをレンダリングし，その処理時間を測定する．時間計測範囲は，データをファイルから読み込んだ直後から，ボリュームレンダリング処理によって生成したピクセル列を2次元配列に書き込むまでである．画像サイズは 256×256 ピクセル，使用したデータは，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（128×128×128 ボクセル，図3）である．このデータは不均一に分布する粒子群のようなデータであり，空間コヒーレンスが低い．

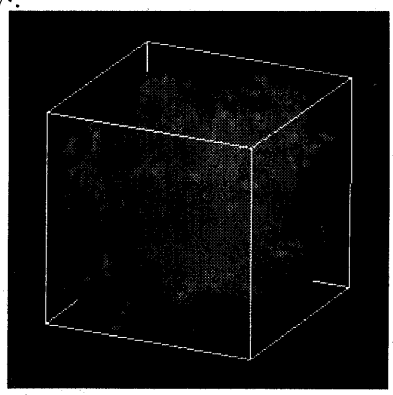

図3 宇宙初期の水素ガスの電離状態

測定結果を図4, 5に示す．グラフの横軸はレンダリングに使用した CPU 数，縦軸はそれぞれ，レンダリング時間と速度向上率を表わしている．速度向上率は Onyx2 と Origin2000 それぞれの 1CPU の結果を基準にしている．グラフから分かるように，ほぼ線形に近い速度向上率を示しており，並列化の効果は充分であると言える．CPU 数が同じでも，Onyx2 と Origin2000 のレンダリング時間に差があるのは，CPU のクロック周波数が違うためである．

ボリュームレンダリングの需要としては，ある固定

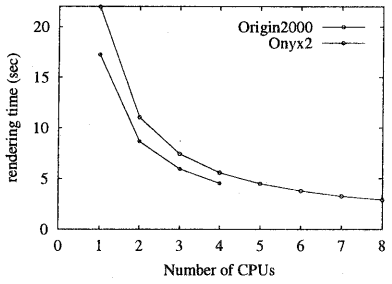

図4　ボリュームレンダリング単体の描画時間

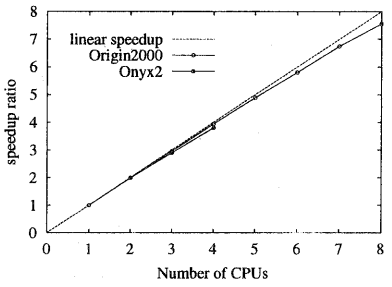

図5　ボリュームレンダリング単体の速度向上率

データ（時系列ではないデータ，例えば医療で使用されるような人体のCTデータ）に対して，あらゆる角度で見たり，不透明度を変えたりして内部を観察したりする場合が多い．このような静的なデータに対しては前処理を一度だけ行なえば良いので，前処理の段階でその後のレンダリングを高速に実行できるようなデータをあらかじめ作成する手法がよく用いられる．例えば，前処理に60秒以上かかるが，レンダリングそのものは1秒以下で行なえる逐次アルゴリズムも存在する[11]．それに対して本システムが対象とするのは，シミュレーションプログラムから送られてくる時々刻々と変わる時系列データである．このような動的なデータを対象とする場合には1データ1レンダリングが基本であり，前処理に時間をかけることはできない．そのため，本システムでは全体のバランスを考えて前処理に時間をかけないアルゴリズムを採用した．

なお，今回は使用できるCPU数を考慮し静的負荷分散を行なうアルゴリズムを実装したが，より大規模なシステムにおいても充分な並列化効率が得られるようにするためには，動的負荷分散を行なうアルゴリズムを実装して行くことが必要であると考えられる．

4.2 AVS/Express モジュール化した性能

本システムは数値計算の結果をリアルタイムに可視化するだけではなく，ファイルからのデータ入力も可能であるので，ここではあらかじめ計算しておいた時系列データを読み込みながら，連続で可視化（ボリュームレンダリング）し，その全実行時間を測定する．したがって，ファイル読み込みや描画にかかる時間も含めることになる．評価に使用するデータは，宇宙初期の再電離過程の3次元輻射輸送シミュレーション・データ（$128 \times 128 \times 128$ ボクセル $\times$ 60ステップ）[12]で，CP-PACSの2048PUすべてを使って計算したものである．画像サイズは 512×512 ピクセルとした．

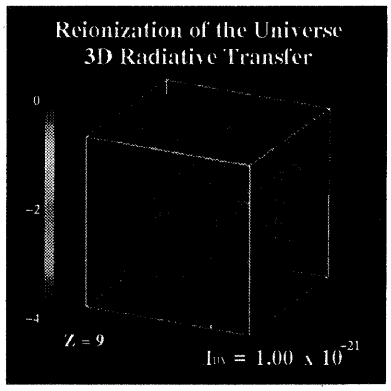

図6　並列ボリュームレンダリングモジュールによる画像

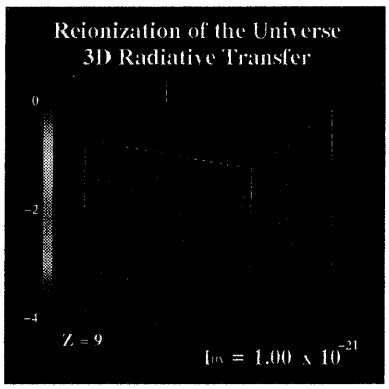

図7　標準のボリュームレンダリングモジュールによる画像

AVS/Express標準のボリュームレンダリングモジュールと並列ボリュームレンダリングモジュールを使って同様の可視化処理を行なう．AVS/Expressのモジュールのソースコードは公開されていないため，レンダリングのパラメータなどを完全に一致させた評価を行なうことはできないが，スクリーンに描画される画像がほぼ同じになるようにパラメータを調整し，そのパラメータの基で評価を行なった．図6と図7がそれぞれのモジュールを用いて生成した画像のスナップショットである．

図8は全実行時間，図9はそれぞれの1CPUの結果を基準にした速度向上率を表わしている．横軸はボリュームレンダリングに使用したCPU数である．なお，AVS/Express標準のボリュームレンダリングモジュールを用いた測定は，Onyx2上で行なっている．

並列ボリュームレンダリング単体性能の評価結果と

は異なり，Onyx2の4CPUで3.5倍弱，Origin2000の8CPUで約4.5倍ほどの速度向上に留まっている．この原因は，AVS/Expressに組み込んだことにより，スクリーンへの描画に時間がかかり，そこがボトルネックになっているためである．また，Origin2000の方がOnyx2よりも速度向上率が低いのは，本システムのOrigin2000が画像表示用のディスプレイを備えていないため，Onyx2のディスプレイに画像を送らなければならないことに因る．

次に処理時間を見てみると，Onyx2の4CPUとOrigin2000の8CPUを使用した場合の実行時間がほとんど同じになっている．この原因は先ほどと同じく，Origin2000には画像をOnyx2のディスプレイに転送するオーバヘッドがあるからである．しかし，AVS/Expressの標準のボリュームレンダリングモジュールと比べると，1CPUの時でも25%以上高速である．

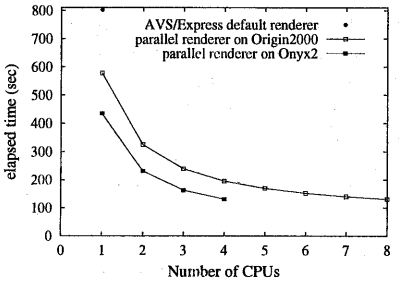

図8 AVS/Expressに組み込んだ時の実行時間

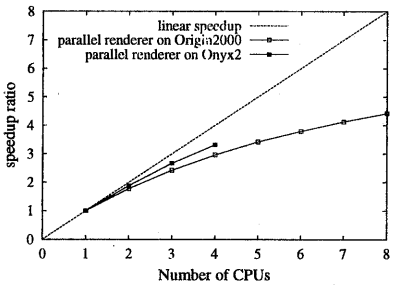

図9 AVS/Expressに組み込んだ時の速度向上率

4.3 PIOとの連動協調動作性能

最後に数値計算の結果のリアルタイムな可視化を行った時の性能評価を示す．ここでは，CP-PACSの256CPUを使用し分子動力学法(MD)シミュレーションプログラム[13]を用いた．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16384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48KB．
- 総ステップ数は20.
- 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（16^3格子）に変換し，ボリュームレンダリングを行なう．
- 512×512ピクセルのスクリーンに描画する．

図10にMDシミュレーションと可視化に要した時間を示す．この図よりAVS/Expressオリジナルのボリュームレンダリングによる可視化の時間に対しては，8CPUで3.58倍のスピードアップが達成されている．このうち，並列ボリュームレンダリングに要した時間も図10に示した．これによると，4.1節で行ったレンダリング単体の結果に比べ，やや劣るものの8CPUで6.55倍のスピードアップを示している．

また，処理時間の内訳を表1に示す．この時間はOrigin2000上のbinderで測定しており，PIOにデータを要求してから受けとり終るまでの時間（データ読込）とAVS/Expressへ渡す共有メモリへの書き込みが完了するまでの時間（可視化合計）を測定している．更に可視化合計は並列ボリュームレンダリングモジュールに要する時間（レンダリング）とその他表示等にかかる時間（その他）に分類した．この実験ではシミュレーションによるデータ生成のスピードが速く，可視化処理がボトルネックとなる．このため共有メモリはダブルバッファリングとしているが，full-empty処理によって書き込みに待ちが生じる．また，使用CPU数に対するレンダリング時間の短縮効率が悪い．この理由は，1ステップ辺りの処理時間のうち並列ボリュームレンダリングの時間ではなく，その後のスクリーンに対する描画時間によって大部分を占められているからである．これは，4.2節で行なったモジュール化した性能の評価結果にも現れているが，MDのデータは規模が小さいために相対的にボリュームレンダリングにかかる時間が小さくなり，表示のオーバヘッドがより顕著に現れている．より大規模データによる可視化ではデータ転送時間やレンダリング時間が支配的になると考えられるので，この表示にかかるオーバーヘッドはそれほど問題にならないと考えられる．

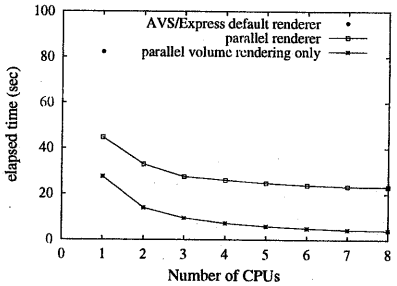

図10 MDシミュレーションと同時実行した時の処理時間

表1　MDシミュレーションと同時実行した時の処理時間の内訳 (秒)

	AVS /Express	parallel rendering (Number of CPUs)							
		1	2	3	4	5	6	7	8
データ読込	0.754	0.733	0.773	0.797	0.779	0.800	0.797	0.799	1.074
レンダリング		27.589	13.936	9.438	7.159	5.830	4.936	4.305	4.021
その他		16.476	18.257	17.272	18.092	18.127	18.051	18.070	17.916
可視化合計	81.606	44.065	32.193	26.710	25.251	23.957	22.987	22.375	21.937
総処理時間	82.360	44.798	32.966	27.507	26.030	24.757	23.784	23.174	23.011

5. おわりに

本研究では, 数値シミュレーションと可視化処理を統合した並列可視化システムについて述べ, 数値計算サーバに超並列計算機 CP-PACS, 可視化サーバに共有メモリ型並列計算機 SGI Onyx2 および Origin2000 を用いて実装し, その性能評価を行った.

CP-PACS と可視化サーバ間のデータ通信には, 我々が開発しているコモディティネットワークによる高スループットな並列入出力システムを用いており, CP-PACS で生成するデータをオンラインで転送し, リアルタイムな可視化を行った. 可視化サーバ側では, この並列入出力からのデータ受信処理と可視化処理をパイプライン的に処理するために, 可視化処理とは独立した受信専用のプロセスを生成し共有メモリを用いてこれらを協調動作させた. このような数値計算と可視化処理の同時実行処理により, 可視化結果から得られた知見を即座に数値計算にフィードバックすることで, 効果的な解析が可能になる.

また, 可視化部分とフロントエンドの GUI は, 業界標準の可視化ソフトウェアである AVS/Express を基本に構築することで, 汎用性と実用性を高めた. オリジナルのモジュールでは逐次な可視化処理しか行なうことができない AVS/Express に並列ボリュームレンダリングモジュールを実装し, 可視化処理の高速化を図った. これにより, AVS/Express に備わっている標準のモジュールを使用した時に比べ, 2 から 8 倍程度高速な処理が可能になった. 以上のように, 本システムを使う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, 大規模データによる性能検証, 並列ボリュームレンダリングモジュールの高速化, 機能の拡充などが挙げられる.

謝辞　本システムの基礎的な研究を行った農業生物資源研究所 沼寿隆氏, ならびに, 種々の御討論を頂いた筑波大学計算物理学研究センターの関係者各位に感謝致します. 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」(Project No. JSPS-RFTF 97P01102) によるものである.

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, pp.53-58, 1999.
2) 岩崎 洋一, 中澤 喜三郎 他, 「計算物理学と超並列計算機 – CP-PACS 計画 –」, 情報処理, Vol.37, No.1, pp.10-42, 1996.
3) http://www.sgi.co.jp/products/
4) 松原 正純, 沼 寿隆, 板倉 憲一, 朴 泰祐, 「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, 99-HPC-76-1, pp.1-6, 1999.
5) H.D.Lord, "Improving the Application Development Process with Modular Visualization Environments", Computer Graphics, vol.29, no.2, pp.10-12, May 1995.
6) http://www.kgt.co.jp/avs/index.html
7) 藤代 一成, 竹島 由里子, 「AVS – ビジュアライゼーションソフトウェア」bit 別冊 「インターネット時代の数学」(戸川, 中嶋, 杉原, 野寺 編), pp.236-245, 1997.
8) M.Levoy, "Display of Surfaces from Volume Data", IEEE, Computer Graphics & Applications, Vol.8, No.5, pp.29-37, 1988.
9) M.Levoy, "Efficient Ray Tracing of Volume Data", ACM Transactions on Graphics, Vol.9, No.3, pp245-261, July 1990.
10) J.Nieh and M.Levoy, "Volume Rendering on Scalable Shared-Memory MIMD Architectures", Proceedings of the Boston Workshop on Volume Visualization, pp.17-24, 1992.
11) P.Lacroute and M.Levoy, "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation", Computer Graphics (SIGGRAPH '94), pp.451-458, 1994.
12) 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
13) 松原 正純, 板倉 憲一, 朴 泰祐, 「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.2172-2182, May 1999.