

数値シミュレーション過程の実時間可視化 を支援するハードウェア

原瀬 史靖, 山内 聡[†], 津邑 公暁, 五島 正裕, 森 眞一郎
中島 康彦, 北村俊明^{††}, 富田 眞治 (京都大学)

A Hardware Assist for Scientific Visualization

FUMIYASU HARASE, SATOSHI YAMAUCHI, TOMOAKI TSUMURA,
MASAHIRO GOSHIMA, SHIN-ICHIRO MORI, YASUHIKO NAKASHIMA,
TOSHIAKI KITAMURA and SHINJI TOMITA

ABSTRACT

This paper presents a concept of real-time parallel volume rendering system for time-varying scientific visualization based on the scalable architecture of ReVolter/C40 developed by Kyoto University. A preliminary experimental environment using the rendering pipeline of ReVolter/C40 and newly designed DataProbe which is an additional hardware for parallel volume data transfer is also presented. In terms of the load balancing in parallel volume rendering, a block cyclic data assignment with the early ray termination technique is also discussed.

Keywords: Volume Rendering, Parallel Processing, Hardware Accelerator, Load Balancing

1. はじめに

PC クラスターの普及により、個人レベルで所有する並列計算機環境を利用した数値シミュレーションが現実的なものとなってきた。本稿では、このようなパーソナルな PC クラスター上での数値シミュレーション過程の実時間可視化を支援するハードウェア環境について述べる。

具体的に我々が開発を行ってきたポリウムレンダリング専用並列計算機 ReVolter/C40^{(1),(2)}を用いて実時間可視化システムを構築する。しかしながら、ReVolter/C40 はバックエンド型の可視化システムとして構築してきたために、シミュレーションの途中結果等の時間変化するポリウムデータの転送に十分対応できていなかった。そこで、ReVolter/C40 にポリウムデータの並列転送を支援する機能拡張を行い実時間可視化を実現する。

さらに、ReVolter/C40 のレンダリングパイプラインに帰還パスを設けることで、シミュレーション系でのブロック/サイクリック等の負荷分割形態に対応した。また、サイクリック分割時には Early Ray Termination を併用することで、レンダリング処理自体の更なる高速化の可能性があることが分かった。

以下、2 章で研究の背景となる ReVolter/C40 について延べ、3 章で ReVolter/C40 を用いた実時間可視化システムについて、4 章でサイクリック分割時の性能について、それぞれ考察した後、5 章でまとめる。

2. ReVolter/C40 の概要

ReVolter/C40 は TI 社の DSP TMS320C40 を用いて構成したポリウムレンダリング専用並列計算機 ReVolter⁽¹⁾のプロトタイプシステムであり⁽²⁾、正方格子ポリウムデータに対して、Ray Casting 法でポリウムレンダリングを行うシステムである。

ReVolter/C40 では、主軸等間隔サンプリング法とこれに対応した 3 重化メモリ構造の採用により、可視化の際の視線方向や視点の位置に関する制限を一切設けることなくバンク数に比例するメモリバンド幅を提供することでリアルタイムな可視化を実現している。

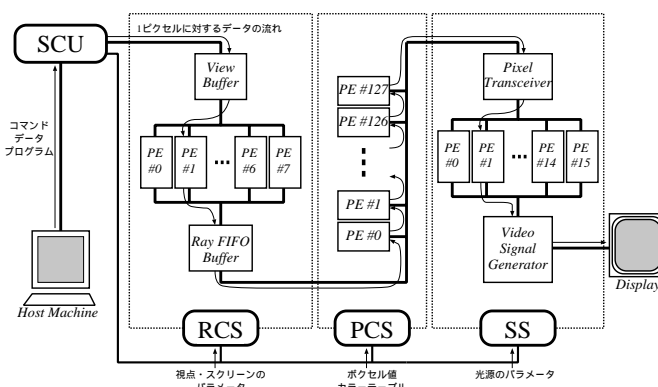


Fig. 1 ReVolter/C40 の全体構成

ReVolter/C40 は、Fig. 1 に示すように、システムコントロールユニット (SCU)、視線生成ステージ (RCS)、ピクセル

[†] 現在: (株) 東芝
^{††} 現在: 広島市立大学

値計算ステージ (PCS), シェーディングステージ (SS) からなる。

ReVolver/C40 の 3 次元メモリでは、1) 与えられたボリュームデータを X, Y および Z の 3 つの主軸に対応して 3 重化し、2) 各々を主軸に添った N 個のサブボリュームに分割する。3) 次に、3 種類のサブボリュームセットから i 番目のサブボリュームを i 番目のメモリバンクに格納する (Fig. 2 参照)。各メモリバンク (VM_i) には 1 つのピクセル値計算ユニット (PCU_i) が対応づけられている (Fig. 1 の PCS を構成する PE_i は VM_i と PCU_i を含んでいる)。

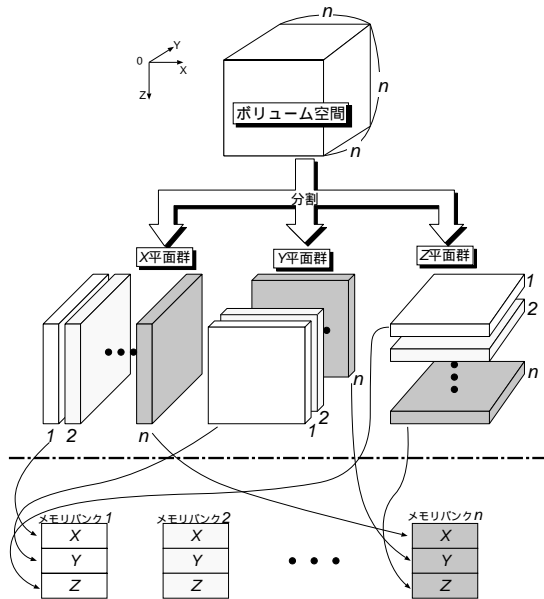


Fig. 2 Volume Data の三重化

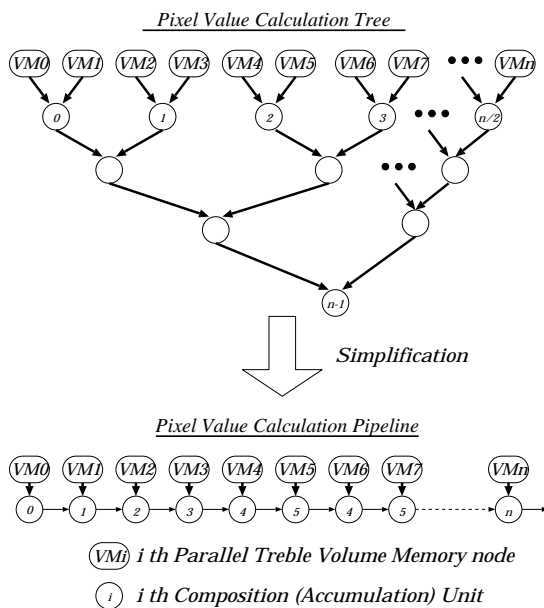


Fig. 3 Composition Network

PCU_{i-1} または RCS から、視線情報とピクセル値計算の途中結果が与えられると PCU_i は自分のメモリバンク VM_i

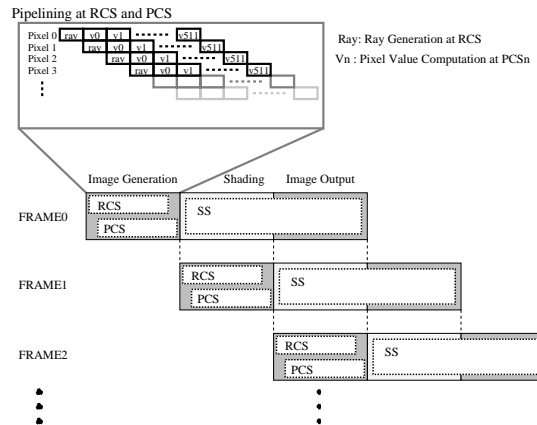


Fig. 4 Rendering Pipeline の動作

内で視線と交差するボリュームデータをサンプリングしピクセル値の計算を行う。N 個の PCU でこの処理をパイプライン処理することで、ボリュームレンダリング処理を高速化する²⁾。

従来、このピクセル値計算を木構造のコンポジションネットワークを用いて $O(\log N)$ 時間で計算するハードウェアが提案されているが^{4), 6)}, ReVolver/C40 ではこれをリング構造のネットワークを用いて $O(N)$ 時間で計算する (Fig. 3)。これにより、最初の 1 ピクセルが表示されるまでの遅延時間は増加するが、この増加は $O(N^2)$ ピクセルの画像表示に要する時間に比べて十部小さい。また、描画速度自体への影響は存在しない。一方で、ネットワークを構成するハードウェアが $1/2$ に軽減され、かつ、構造がシンプルになるため高速化ならびに拡張性の面で優れているといえる³⁾。

3. ReVolver/C40 を用いた実時間可視化

ボリュームデータの並列転送を支援するハードウェア (Data Probe と呼ぶ) を作成し、ReVolver/C40 の機能拡張を行い時変ボリュームデータの実時間可視化を実現する。

予備評価環境としては、P 台の PC からなる PC クラスタと N 個のピクセル値計算ユニット (PCU) からなるピクセル値計算ステージ (PCS) を用いて実時間可視化を行うシステムを構築する。このとき、 N/P 個の PCU 毎に、Data Probe を介したボリュームデータ転送経路を設けることで、データ転送時間を $1/P$ にすることを旨とする。

3.1 システム構成

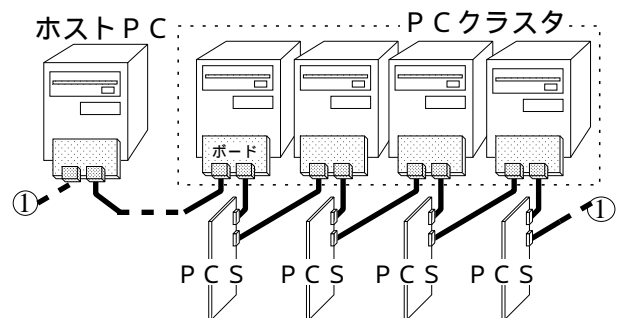


Fig. 5 プロトタイプシステムの構成

全体の構成は Fig. 5 のように、PCI バス に DataProbe が挿さったパソコンと PCS 基板 (各基板に 8 個の PCU) からなる。PC クラスタでシミュレーションを実行し、その結果を PCS でレンダリングし、最終結果を Host PC で取り込みシェーディング等の処理を行ったのちディスプレイに表示する。

Host PC は、この他にシミュレーションの実行開始・終了、データ転送の開始・終了、ポリウムレンダリングの開始・終了の全体の制御と、ポリウムレンダリング実行時の視線生成を行う。

3.2 DataProbe

DataProbe は、PCI スロットに挿入する PCI カードで、PC の主記憶上に置かれたシミュレーションの途中結果 (ポリウムデータ) を PCS 内のポリウムメモリに転送するためのインタフェースカードである。従来 PCS 基板間を直結していた通信リンク (ComPort) に直列に挿入する形で PCS と接続する。

DataProbe の基本動作モードは、1) シミュレーションのデータ等を PC クラスタから PCS へ転送するモード、と 2) 可視化処理実行時に PCS 間の通信を中継する二つのモードがある。

現在は、Xilinx 社の FPGA (XC4062XLA) を用いて構成しており、基本的な通信機能のみが実装を完了している。

3.3 シミュレーションと可視化処理の連携

Host PC での PC クラスタのシミュレーションの制御は以下の順序で行う (Fig. 6 参照)。

- (1) 各 PC は一定間隔シミュレーションを実行し、その結果を PCS に転送する。
- (2) 転送が終了したら Host PC に知らせる。全 PC が転送を終了した時点で Host PC は各 PC の Data Probe を、可視化のためのデータ転送モードにする。
- (3) ポリウムデータの三重化が必要であれば、全ての PCS 間でデータを一巡させる。各 PCU は送られてきたデータが自分のポリウムメモリに格納すべきデータであれば、そのコピーをつくって格納する。
- (4) Host PC から視線データを送信し、PCS でレンダリング処理を行う。最終結果は Host PC が受け取り、自身のディスプレイに表示する。
- (5) 可視化条件を変更して可視化を続ける場合はそのままくり返し、可視化処理を終了する場合には Data Probe を PCS へのデータ転送モードに変更し、1 に戻って各 PC のシミュレーション終了をまつ。

3.4 プロトタイプシステムの評価

3.4.1 評価環境

PC クラスタは、Intel Pentium III 1GHz 主記憶 512MB の PC が 5 台、OS は Vine Linux 2.1.5 (kernel 2.2.18) で構成している。ネットワークには 100BASE-TX のイーサネットを用いており、与えられた境界条件の下で熱拡散方程式を解く簡単なプログラムを MPI 通信ライブラリを用いて実装した。データの分割は、 256^3 の 3 次元データを $256^2 \times 64$ のデータに四分割し、それぞれの PC に割りあてている。この時、各 PC が生成するポリウムデータのサイズは各々 4MB となる。

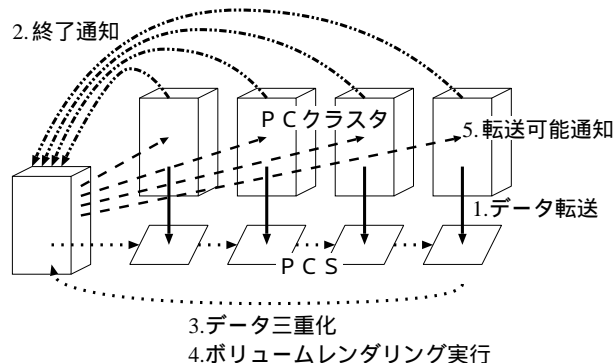


Fig. 6 ポリウムデータの転送と可視化処理

3.4.2 ポリウムデータ転送時間の評価と考察

PCS への転送を並列に行う場合と行わない場合を実装し比較を行った。なお、可視化時間は両者とも同じであるので、可視化処理開始までに必要となる時間を実測した。Table 1 は、データ転送時間の内訳である。

Gather 処理は、逐次転送において PC クラスタ側のネットワークでポリウムデータを Host PC に集める時間であり、PCS 内巡回転送は並列転送時に、ポリウムデータの 3 重化を行う際に必要となる転送である²⁾。逐次転送の場合は、PCS へのポリウム転送と並行して 3 重化の処理が行われるため、別途 PCS 内巡回転送を行う必要はない。

Table 1 ポリウムデータ転送時間の内訳

	逐次転送	並列転送
Gather 処理	4.7	none
PC→PCS 転送	19.8	5.0
PCS 内巡回転送	none	9.9
合計	24.5	14.9

PC から PCS への転送時間が 1/4 になっていることが分かる。視点と画角に若干の制約をつけてポリウムデータの 3 重化を行わずに可視化を行う場合は⁹⁾、約 5 秒で全ての処理が完了する。この時間は DataProbe を挿入した PC の台数に反比例して単純に減少する。一方、逐次転送に要する時間は、PC の台数に関わらずほぼ一定である。ポリウムデータの 3 重化を行う場合においても、並列転送の有効性が確認できた。

現在、PC 内巡回転送時には、PCU 内の DSP がプログラム転送を行っており、実効転送速度が 1.5MB/s 程度しか得られていないが、PCU 内の専用ハードウェア (FPGA) に専用の転送エンジンを実装すれば、容易に数倍の高速化が期待できる。

さらに、描画時の詳細度制御¹³⁾を併用して、3 重化処理と描画処理を同時進行させることで、可視化システムの描画停止時間を最小にすることが可能と考えられる。

4. サイクリック分割への対応

ReVolver/C40 のレンダリングパイプラインは、当初は視線生成ステージからシェーディングステージまで一方向のリンク構造であったが、ポリウムデータの循環転送やポリウムレンダリング処理自体の負荷分散を支援するため、ピクセル値計算ステージの構成を若干変更し、PCU_i と PCU_{(i+1) mod N} の間に双方向のリンクを設けることで、レン

本来は AGP が望ましい。
現在は DSP TMS320C40 の通信リンクに直結 (20MB/s peak)

ダリングパイプラインの帰還パスを実現した。

2章のアーキテクチャの説明では、PCU 毎に1組のサブボリュームを対応づけていたが、3章の例のように1つのPCUにM組のサブボリュームを割り当ててみる場合を考えてみる。PCU_{N-1}からPCU₀に帰還パスが存在することから、サイクリックなサブボリューム割り当てが可能となる。このように、シミュレーション系でのブロック/サイクリック等の負荷分割形態に対応することが可能となった。しかしながら、ボリュームデータをサイクリックに割り当てた場合、ピクセル値計算時にネットワークの負荷が上がると言う問題が発生する。今、M=1の時、1つのPCUでの1フレーム分の処理に要する時間を通信時間Cと計算時間Tの和として表現すると、幅 $\frac{M}{m}$ のブロックサイクリック分割時に描画に要する時間 $T_{M,m}$ は、

$$T_{M,m} = (C + \frac{M}{m} \times T) \times m \quad (1)$$

となり、ブロック分解 (m=1に相当) に比べて $C \times (m-1)$ だけ処理時間が増加する。

この問題に対しては、サイクリック分割であることを利用して、Early Ray Terminationの手法を適用することが考えられる。例えば、m=2の場合で、E%の視線がボリューム空間の半分までのサンプリングで Terminateした場合、描画時間 $T_{M,m,E}$ は

$$T_{M,2,E} = (C + \frac{M}{2} \times T) \times (2 - \frac{E}{100}) \quad (2)$$

に短縮される。256³のボリュームデータを256²スクリーンに描画する場合のReVolter/C40demoでの実測データC=164msec, T=207msecを適用すると、M=2ではE=45程度、M=32ではE=5程度でブロック分割より高速になることが分かった。なお、PCクラスタ等での並列ボリュームレンダリングで同様のサイクリック分割を行うと、一般に $C \gg T$ であるため、M, Eを相当大きくとって始めて有効性が確認できることが分かっている¹³⁾。

5. おわりに

本稿では、これまで我々が開発してきたReVolter/C40を応用し、パーソナルなPCクラスタ上での数値シミュレーション過程の実時間実時間可視化を支援するハードウェア環境について述べた。

ReVolter/C40プロトタイプシステムが約7年前のデバイス技術をベースに設計されているため、実測データそのものは「実時間」には達していないが、現在のデバイス技術に基づいて外挿すると十分実時間性が達成できているものと考えられる。

現在、これらの成果を踏まえて、最新のデバイス技術を用いた場合の大規模シミュレーションと実時間可視化システムを統合した仮想実験環境の構築に着手しているところである¹²⁾。

謝 辞

日頃より御討論いただく京都大学大学院情報学研究科富田研究室の諸氏に感謝します。なお、本研究の一部は文部省科学研究費補助金(基盤研究(B)(2)課題番号13480083な

らびに特定領域研究(C)(2)「情報学」課題番号13224050)による。

参 考 文 献

- 1) 對馬 雄次 他, “ボリューム・レンダリング専用並列計算機 ReVolter のアーキテクチャ”, 情報処理学会論文誌, 第36巻, 第7号, pp.1709-1718, 1995.
- 2) 吉谷直樹 他, “ボリュームレンダリング専用並列計算機 ReVolter/C40 の性能評価”, 情処研報告, 99-ARC-132, pp.79-84, 1999.
- 3) John Eyles ,et.al., “PixelFlow: The Realization”, Proc. SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware, pp.57-68, 1997.
- 4) Hanspeter Pfister, et al., “Cube-3: A Real-time Architecture for High-resolution Volume Visualization”, Proc. of 1994 Symp. on Volume Visualization, pp. 75-82, 1994.
- 5) Hanspeter Pfister, et. al., “The VolumePro Real-Time Ray-Casting System”, ACM SIGGRAPH 99, pp.251-260, 1999.
- 6) 村木 茂 他, “VG クラスタ:スケーラブルビジュアルコンピューティングシステム”, Visual Computing グラフィクスとCAD 合同シンポジウム 2001 予稿集, pp.85-90,2001.
- 7) 小西, 小笠, “リアルタイム可視化ツール VisLink の紹介”, 京都大学大型計算機センター広報, Vol.34, No.4, pp.209-222(2001).
- 8) 助村 俊一 他, “バーチャルマイクロスコプの制御手法の開発”, 情報研報 99-HPC-66(HOKKE'97), Vol.97, No.37, pp.25-30,1997.
- 9) 金 喜都 他, “ピクセル並列処理によるボリューム・レンダリング向けの超高速専用計算機アーキテクチャ”, 情報処理学会論文誌, 第38巻, 第9号, pp.1668-1680, 1997.
- 10) 藤原雅宏 他, “階層格子ボリュームデータの実時間可視化”, 情処研報, 98-ARC-128, pp.7-12, 1998.
- 11) 山内, 他, “透視投影ボリュームレンダリングにおけるサンプリング方式の評価”, 第29回画像電子学会年次大会予稿集, pp.33-34, 2001年6月.
- 12) 山内, 他, “アクティブボリュームレンダリングに基づくシミュレーションステアリング”, 信学技報 CPSY2001-35, pp.1-8, 2001年8月.
- 13) 原瀬 史靖, “並列ボリュームレンダリング処理の高速化”, 京都大学工学部情報学科 特別研究報告書, Feb. 2002.
- 14) Kevin Kreeger, et. al., “Adaptive Perspective Ray Casting”, Proc. of 1998 Symp. on Volume Visualization, pp.55-62, Oct. 1998.

32PE 構成のデモ機で、専用の FPGA や高速リンクを使わず DSP とその通信リンクのみを使用するもの