

画像合成装置を用いたシミュレーションと可視化の同時処理性能比較

片野 康生[†] 梶原 景範[†] 劉 学振[†]
緒方正人[†] 村木 茂^{††}

我々はこれまで開発した画像合成装置を用いることによって、PC クラスタシステムにおける画像生成を高速化する手法を開発してきた。本発表ではその PC クラスタシステムの並列処理機能を用いてシミュレーションと可視化を同時に行い、その性能の比較・検討をおこなった。シミュレーション対象として軟組織をバネモデルでモデル化し、セルラーオートマトンを用いて実現した。24 × 24 × 17 格子点 (9792 voxel) を 16PC (32CPU Pentium III × 16, Xeon × 16) で動作した場合、シミュレーションと可視化の同時処理を約 22.7 Hz の更新レートで動作することを確認した。

Performance comparison of simultaneous processing with simulation and visualization by using Image Composition Hardware

YASUO KATANO,[†] KAGENORI KAJIHARA,[†]
LIU XUEZHEN,[†] MASATO OGATA[†] and SHIGERU MURAKI^{††}

We have developed the method that speeds up the image generation in the PC cluster system, by using Image Composition Hardware. In this announcement, we compared a performance of simultaneous processing with PC Cluster system. We simulated soft tissue (24 × 24 × 17 mesh : 9792 voxel) with spring model using Cellular Automaton (CA). When the soft tissue was simulated with 16 PCs (32CPU Pentium III × 16, Xeon × 16), we confirmed that the refresh rate is 22.7 Hz.

1. はじめに

近年パーソナルコンピュータ (以下 PC) やネットワーク技術が半導体製造技術、光通信技術などの進歩により高速化・低価格化したことにより、複数の PC を高速ネットワークで並列化する PC クラスタの技術が登場してきた。これにより高価なワークステーションを使用せず、PC の並列化だけで高度なシミュレーションを行うシステムを構成することができるようになった。PC クラスタは安価な市販の PC 部品を使いシステムを構築するため、ワークステーションと比較して安価に同等のシステムを構築することが可能となる。

しかし、計算結果の可視化を行う場合、結果をディスクに保存し、別のグラフィクスワークステーションにデータを移行して可視化するといった複数のステップ

を経ることが多い。これは結果の直感的な把握という観点からは遠く、インタラクティブなシミュレーション環境の実現を阻害する。またシミュレーション用と可視化用の計算機が別々で必要となり、トータルコストにも影響がでる。

そこで我々は PC クラスタ技術と並列映像発生技術を組み合わせ、画像合成に専用ハードウェアを用いて高速化する VG クラスタシステムを開発してきた^{1)~3)}。本発表はシミュレーションや設計などの分野でインタラクティブなシミュレーション環境を提供することを目的とし、セルラーオートマトンによるシミュレーションと画像合成装置を用いた可視化とを同時処理する PC クラスタシステム⁴⁾ の実時間性能評価を行ったので報告する。

1.1 シミュレーションにおける並列処理

最初にシミュレーションと可視化を同時処理する上で大きな問題となるネットワーク転送負荷について述べる。大規模な科学技術計算やシミュレーションの分野では、大規模並列の PC クラスタシステムがワークステーションを凌駕しつつある。そのようなシミュレーションの並列処理を考える上で重要なのはネット

[†]三菱プレジジョン株式会社
Mitsubishi Precision Co., Ltd.

^{††}産業総合研究所
National Institute of Advanced Industrial Science
Technology

ワークにかかる負荷である。

例えばセルラーオートマトンを用いたシミュレーションにおいては、隣接するセル同士の相互作用によってその状態を決定する^{5)~6)}ため、シミュレーション領域を分割することによって計算負荷を分散した場合、PC間で境界領域におけるデータの交換が必要で通常は境界領域にゴーストセルと呼ばれる重複したセル領域を設定し、その領域の計算を接する両方のPCで行い、計算結果を互いに交換する。そのためPCクラスタの並列度が增大するに従いPC間で重複するセル領域が増え、そのデータ交換のネットワーク転送負荷により、並列効率は低下する。

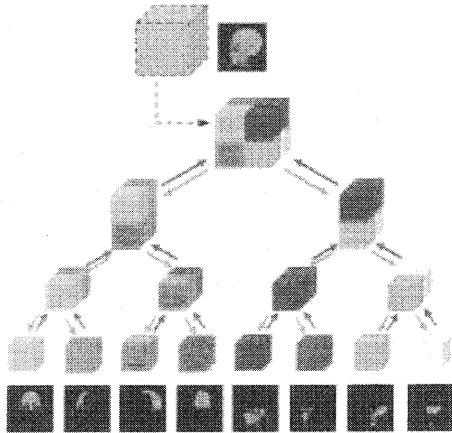


図1 画像合成の手法

1.2 可視化における並列処理

対して可視化を並列処理する場合には、担当したシミュレーション領域それぞれの可視化画像を各PCが作成し、その画像を合成することによって最終映像を生成する手法にはいくつかの方法がある^{7)~8)}。

今回の画像合成手法の概要を図1に示す。可視化処理を並列で行う場合、もっとも大きな負荷は画像データの転送負荷である。各自が持つデータ領域を可視化することによって各PCには最終画像と同じ大きさの画像が生成される(図中最下層)。そのため、それらをすべて合成するにはサーバPCにすべての画像データを転送する必要がある。そのデータは並列するPCの数に比例し、画像が大きくなるほど転送負荷も増大する。このような負荷がネットワークにおける処理速度を大きく低下するため、シミュレーションと可視化の同時処理を難しいものとしている。

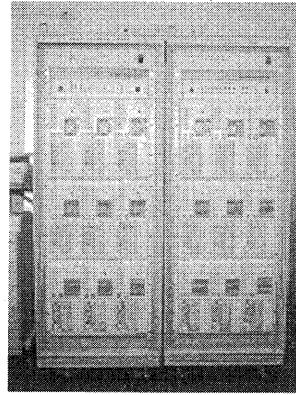


図2 PCクラスタシステム

2. システム構成

2.1 PCクラスタ構成

図2に今回使用したPCクラスタシステムの概観を示す。システムは2台のキャビネットで作成され、1キャビネットあたりに8台のノードPCと1台のサーバPCが搭載されている。CPUにはそれぞれPentium III, XeonプロセッサをDualで搭載し、グラフィックボードにはnVidiaのGeforceシリーズを採用している。

表1にPCクラスタシステムのスペックを、図3にPCクラスタの構成を示す。各ノードPCでは受け持ちの分割領域単位でのシミュレーション、レンダリングおよび合成処理を行う(図1中最下層参照)。各PC間の通信は物理的通信速度の高速化をはかるために、PC間接続にMyrinet(転送速度1.28 Gbit/sec)を採用している。

2.2 画像合成装置

続いて画像合成装置の構成を図4に示す。このPCクラスタシステムを構築する場合に重要な要素がネットワークのトラフィックである。並列度の増加に伴いプロセッサ間通信がボトルネックとなりやすく、ソフトウェアのみによる並列化ではビデオレートによる映像表示が難しくなる。我々はこの問題を解決するために、8入力を1単位とする画像合成装置を開発し、可

表1 PCクラスタのスペック

	Cabinet 1	Cabinet 2
PC数	9(node 8, server 1)	9(node 8, server 1)
CPU	Pentium III800MHz × 2	Xeon 2GHz × 2
GPU	Geforce 3	Geforce 4
Memory	512MB	1024MB
Network	Myrinet	Myrinet
OS	Linux 7.2, SCore5.0.1	Linux 7.2, SCore5.0.1

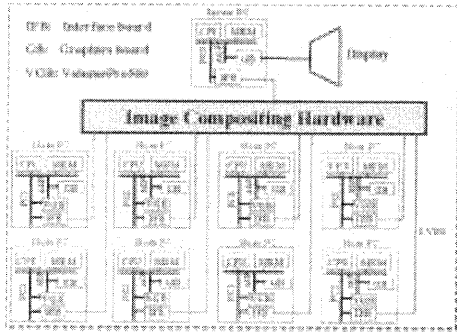


図3 PC クラスタの構成 (9PC: 8 並列)

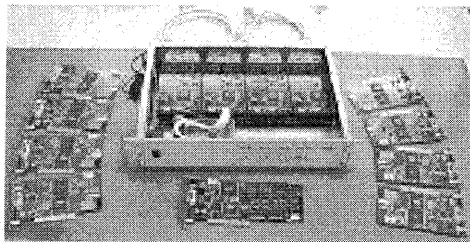


図4 画像合成装置の構成

視覚処理のパイプライン化を行ってきた。

各ノード PC は Binary-swap⁷⁾ によって求められた画面の前後関係とサブイメージの全画素の色情報 (RGBA) を画像合成装置の入力ポートに転送する。PCIバスに挿入されたインタフェースカードを経由して画像合成装置に送られた各 PC 上のサブイメージはそこで与えられた前後関係と各画素ごと与えられた透明度 (α 値) の値をもとに装置内の FPGA によって合成処理を施される。画像合成装置内で合成された画像はインタフェースカードを経由してサーバ PC のグラフィックボードのフレームメモリに書き込まれ、最終映像としてディスプレイに表示される。各フレームの画像合成装置での遅れ時間は PCI ポートの転送速度 33MHz の 21 クロック分 (約 $0.636 \mu \text{ sec}$) で並列処理で使用するネットワークに負荷を与えないためシミュレーションと可視化の同時処理を可能にする。この画像合成装置は階層的に接続することにより最大 512 PC の並列化まで対応している。

2.3 ソフトウェア構成

OS には Linux 7.2 を採用し、Score 5.0.1 をインストールしてある。グラフィック系のプログラムには OpenGL を、通信関係には MPI (Message Passing Interface) を採用することにより汎用性、移植性を高めている。

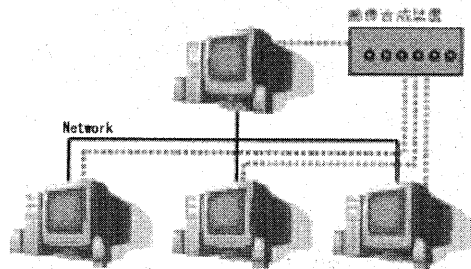


図5 実験条件の概念図

3. 性能評価実験

3.1 実験概要

続いてこの PC クラスタシステムを用いて、シミュレーションと可視化の同時処理性能を評価する実験について述べる。実験には、前章で述べた 16 PC (32 CPU) の PC クラスタを使用し、シミュレーション実験の結果を同時に表示する。その同時処理の更新レートから性能を評価する。実験条件は 2 つあり、条件ではノード PC の数を 1 → 16 まで変化させ、その並列化の効率を検証する。条件は画像合成装置の有無で図 5 に 2 つの条件の概念図を示す。画像合成装置を使用しない条件 (以下 without 条件) では従来の方法と同じでシミュレーションに伴う通信負荷と可視化処理に伴う通信負荷はいずれも Myrinet による通信が負担する (図中実線のみ構成)。画像合成装置を使用する条件 (以下 with 条件) では画像合成装置を使用することで可視化処理に伴う負荷を画像合成装置の PCI バス経路で負担する (図中実線と点線の構成) 評価に使用するシミュレーションには図 6 に示すバネモデルを用いた軟組織変形シミュレーションを採用し、並列処理手法にはセルラーオートマトンを用いて変形計算を行い、各セル間の運動方程式を、

$$\vec{f}_i = -r_i \vec{x}_i - m_i G - \sum_{j \ni \text{近傍}} k_{ij} (|\vec{x}_{ij}| - L_{ij}) \frac{\vec{x}_{ij}}{|\vec{x}_{ij}|}$$

と記述する。初項が粘性項、第 2 項が重力項、第 3 項がバネによる弾性変形項を示す。評価はシミュレーションと可視化の同時処理する際の更新レートで行い、各条件による結果は 400 回の平均値から求めた。出力した画像は 512×512 のフルカラー画像である。

3.2 実験結果

実験結果を図 7 に示す。横軸はノード PC の数を示し、右に行くほど並列度が高くなることを表す。縦軸は更新レートで上に行くほど高速に処理ができてい

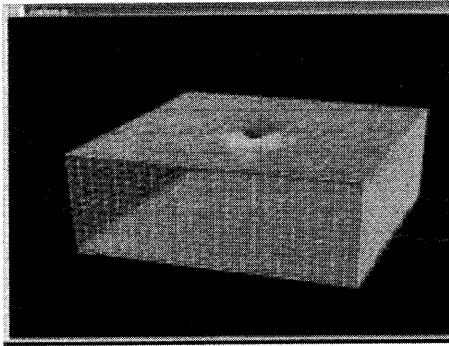


図6 軟組織変形シミュレーションの実行例

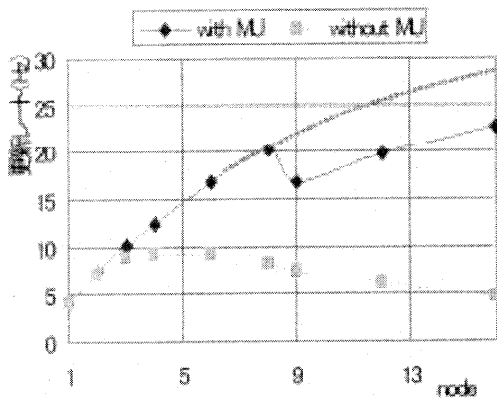


図7 実験結果

ことがわかる。通常ビデオレートで30~60 Hzを目安とするため、30 Hzを実時間処理の目標値とした。

すべてのノード数でwith条件>without条件となり、画像合成装置を使用することにより同時処理性能が向上していることを示す結果となった。それぞれの条件における最高更新レートを比較するとwithout条件で4ノード時に9.11 Hz、with条件では16ノード時に22.73 Hzと2.49倍、また最高並列16ノード時と比較すると4.80倍もの差となった。

また、グラフをからwith条件で8ノードと9ノードの間で大きく更新レートが落ちていることがわかる。これは表1に示すようにキャビネット1とキャビネット2でCPU構成が異なることが原因である。キャビネット1のみで計算を行っているときにはXeonのクラスタで計算を行っているが、キャビネット2も計算に加えると遅いPentium IIIの処理に同期して更新レートが遅くなっているものと考えられる。よってす

べてのPCがXeonを採用していると仮定した場合の予想曲線を点線で表すと、16ノード時には約28 Hzとビデオレートに匹敵する数値を達成していることがわかる。

またwithout条件においてキャビネット間で性能差が発生しないのは、それ以上にネットワークの負荷がボトルネックになっているものと考えられる。

3.3 withとwithoutとの性能比較

without条件は4ノード目をピークに並列度が上がるにつれて更新レートが落ちる。それに対してwith条件はノード数にほぼ比例して更新レートの向上がみられる結果となった。without条件で並列性能が向上しない理由はネットワークのボトルネックにあるものと考えられる。without条件におけるネットワーク負荷には主に並列シミュレーションにおけるゴーストセルのデータ交換負荷、サブイメージの画像転送負荷が挙げられる。

の負荷はシミュレーションを行う上での通信負荷であるためwith条件、without条件の両方に存在する負荷である。対してwithout条件のみに存在する負荷で、各ノードPCから $512 \times 512 \times 4$ byte (RGBA) \approx 1 MByteのデータが転送されるため並列度nにおける転送量はn Mbyteになる。そのため実時間で処理しようとするとき並列化による計算効率の向上効果を画像の転送負荷が上回り、並列化の効率が下がることになるものと考えられる。

with条件にはの負荷はあるが、の負荷を画像合成装置を用いてPCIバス経由で転送しているため画像転送に伴うネットワーク上の負荷は存在しない。そのため可視化を伴わない並列シミュレーションとほぼ同等の効果が現れた。以上のことから画像合成装置を使用することでシミュレーションと可視化を実時間同時処理できる可能性を示す結果を得た。

現在、多くの並列計算手法が提案されており、それらの最適化によって条件間の差は縮まるものと予想される。しかしwith条件における同時処理は可視化における負荷をネットワークにかけないため通常のグラフィック処理を行わないPCクラスタとほぼ同等の性能を示すことができるものと考えられ、シミュレーションと可視化の同時処理を行う上で画像合成装置は大きな効果をもたらすものと期待できる。

4. まとめ

インタラクティブなシミュレーション環境を実現するために、シミュレーションと可視化を同時に処理することが可能なシステムを目標に検討、評価を行った。

その結果、画像合成装置を使用することによって 16 ノードの Pentium III CPU による構成で 22.73Hz, Xeon CPU 搭載の PC クラスタシステムのレベルで約 28 Hz の処理性能を示す結果を得た。画像合成装置の設計上、その階層接続においてほとんど時間遅れを生じないことから高い並列度のシステムでも画像転送の負荷を除去でき、さらに大きなデータの実時間可視化が可能になると考えられる。また市販 PC を使う利点を生かし、今後登場するであろうさらに早い CPU への換装やグラフィックボードの機能を積極的に活用することによって安価に性能を向上させることも期待できる。

シミュレーションと可視化を同時に行うことが可能なシステムは、シミュレーション分野の研究者にとって非常に有益な技術となる。この画像合成装置によってさらに科学技術計算、設計などの分野で使用できるシステムへと発展させていきたい。

謝辞 本研究の一部は科学技術振興事業団 計算科学技術活用型特定研究開発推進事業の助成を受けた。

参 考 文 献

- 1) S. Muraki, K. Shimokawa, M. Ogata, K. Kajihara, K. Ma and Y. Ishikawa: "VG Cluster: Large Scale Visual Computing System for Volumetric Simulations", SC2000 Research Gems(Poster), November 2000
- 2) S. Muraki, M. Ogata, K. Ma, K. Koshizuka, K. Kajihara, X. Liu, Y. Nagano and K. Shimokawa: "Next-Generation Visual Super-Computing using PC Cluster with Volume Graphics Hardware Devices", will be appear on SC2001: Super Computer Conference, Denver, Colorad, November 2001
- 3) 緒方, 梶原, 村木, 下川, マ, 土肥: "マルチプルシア重畳法を用いた並列ボリュームレンダリングシステム", 映像情報メディア学会誌, Vol.55, No.7, pp.1011-1018, 2001
- 4) 緒方, 村木, マ, 劉: "高並列ボリュームレンダリングを可能にするパイプライン画像重畳装置の設計と評価"
- 5) J. von Neumann: "Theory of Self-Reproducing Automata", Univ. of Illinois Press. (1966)
- 6) S. Wolfram : "Cellar automata and complexity", collected papers, Addison-Wesley, 1994
- 7) Kwan-Liu Ma: "Parallel Volume Rendering Using Binary-Swap Computing", In IEEE Computer Graphics and Applications, 14, 4, pp.59-68, 1994
- 8) G. Knittel: "The UltraVis System", In

Hewlette-Packard Laboratories, Usual Computing Department, knittel@hpl.hp.com