

課題名 (タイトル) :

大規模データ可視化システムの分散ファイルロード機能およびユーザ運用機能の開発

利用者氏名 :

○ 小野 謙二 川鍋 友宏
藤田 将洋 瀧内 元気

所属 :

社会知創成事業 次世代計算科学研究開発プログラム
次世代生命体統合シミュレーション研究推進グループ
生命体基盤ソフトウェア開発・高度化チーム

1. 本課題の研究の背景、目的、関係するプロジェクトとの関係

次世代計算機を利用し得られた大規模な計算結果を可視化するシステムの開発を行っている。このシステムは大規模なデータを分割し、各々を並列プロセスでロードする。個々の並列プロセスで生成した可視化画面はネットワークを經由し、重ね合わせられ、一枚の作画を行う。この作画処理を行うクライアントプログラムはネットワーク上の並列計算器とは別にあり、クライアントサーバシステムとして構築する。

このようなシステムを大規模計算機システムで動作し、効率を上げる事により、今後の大規模計算ユーザが利用する可視化システムの構築を目指す。

2. 具体的な利用内容、計算方法と結果

シミュレーションが作り出す大規模なデータに対して、インタラクティブ、バッチ、リモート、ローカルなど多様な利用形態を想定した可視化システム (図 1) の開発を実施している。可視化システムの論理的な設計は、分散並列動作による部分画像の生成、ソートラスト方式による画像重畳、クライアント・サーバー型アプリケーション、CLI/GUI クライアント、HW/SW レンダラーなどの機能構成により、ユーザーの多様な利用シナリオに対応するように設計されている。各機能はフレームワークに対して、拡張的に機能追加が可能なように実装できる。現時点では、対応するデータ形式は、直交等間隔、非構造格子、点群データに限られるが、今後順次追加予定である。

この開発における各課題について以下に報告する。

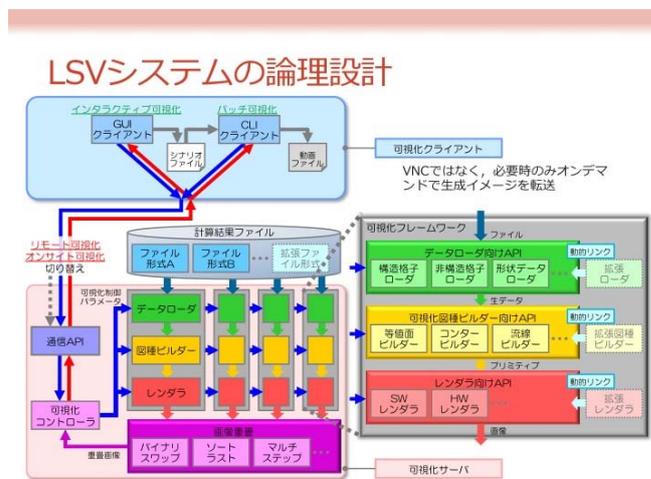


図 1 LSV システムの概要

2.1. LSV システムの機能開発

2.1.1. はじめに

開発中の大規模データ可視化システム(LSV)に対し、バッチ可視化機能、アニメーション動画作成機能を追加する機能開発を実施した。また、RICC 環境で並列動作の検証を実施した。

事前に作成したアニメーション動画作成の Python バッチスクリプトを使用し、RICC 上で LSV をバッチジョブとして動作させる。

2.1.2. 結果

RICC の upc および mpc において、LSV のバッチ可視化機能によりアニメーション動画作成が行えることを確認した。図 2 にクライアントでの表示イメージを示す。

UPC において、GPU を使った並列 H/W レンダリングの動作を確認。100 ノードにて大規模データ(100GB 程度)のインタラクティブな可視化が可能であることを確認。またデータロードにおいて、7 割程度のスケーラビリティ

イを確保できることを確認した。

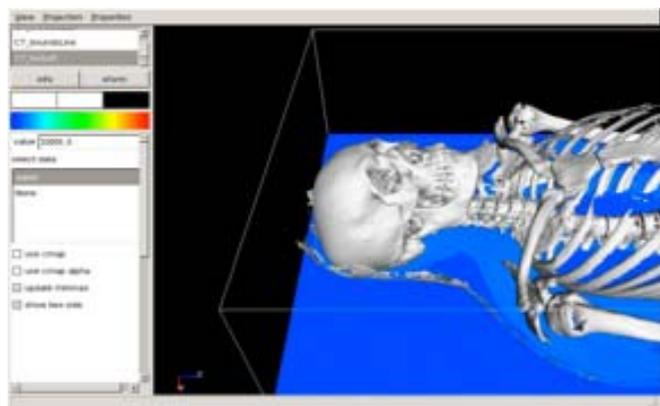
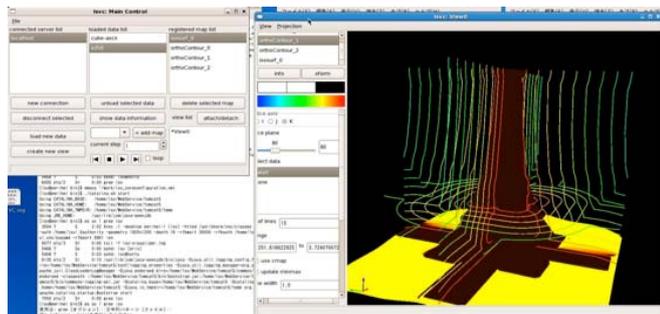
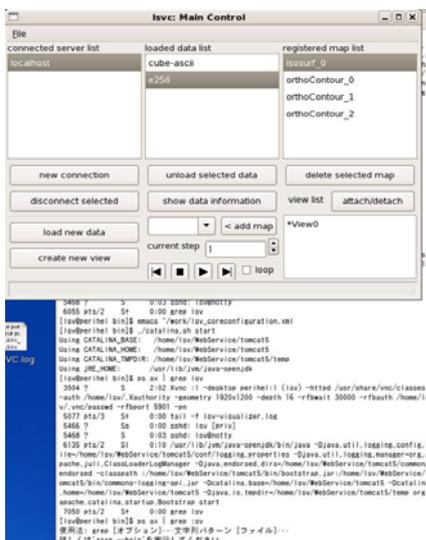
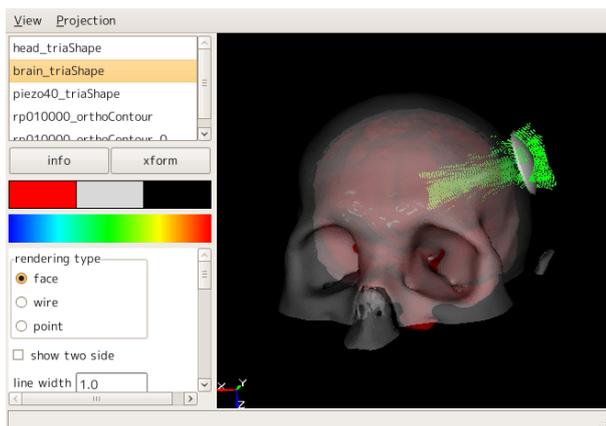


図 2 LSV クライアントイメージ

2.2. 性能評価

2.2.1. はじめに

本年度の開発を進めるにあたり、昨年度（FY09）で開発したシステムの性能評価を実施した。

2.2.2. 評価方法

利用したデータは HIFU の解析データ（600x800x600, 200step）および 100GB の人体ボクセルデータである。並列数は RICC の GPU 搭載のノードを 100 ノード利用した。HIFU データの場合 80 領域に分割し、読み込み時間と描画時間を計測した。描画図種にはコンターを用いた。

2.2.3. 結果

測定結果を図 3 に示す。16 並列まではスケールしていることがわかる。視点変更や可視化パラメータの変更操作では、おおむねインタラクティブな操作が可能であることがわかった。また、現時点ではデータロードがボトルネックになっていることがわかった。

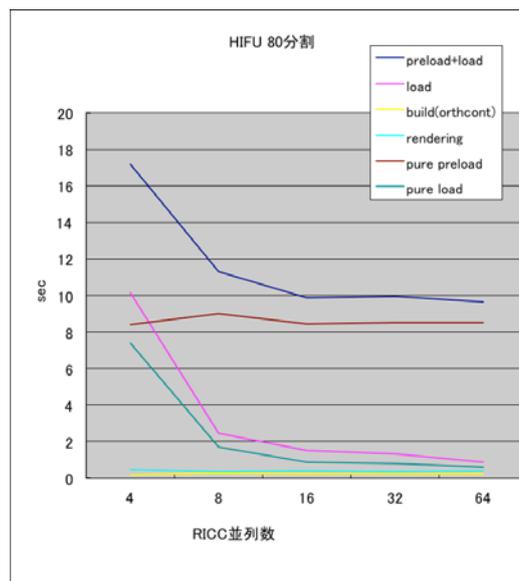


図 3 並列可視化の計時結果

2.2. 高品位レンダリング機能の開発

2.2.1. はじめに

描画機能について、GPU を搭載していない計算機でも可視化が可能ないように、レイトラサーによる可視化機能を開発している。CPU を用いたレンダリングではインタラクティブ性よりも高品位な画像を得る方がメリットがあるため、特に品質を重視して開発を実施した。

2.2.2. 計算方法

レイトレーサを基本にパストレースやアンビエントオクルージョンなどにより品質を向上する。サンプルの事例を図 4 に示す。



図 4 グローバルイルミネーション効果を用いたレンダリング結果

(Data: Sponza atrium copyright by@Marko Dabrovi
<http://hdri.cgtechniques.com/~sponza/files/>)

評価する Engine データは 1 個およそ 40 万ポリゴンとなっている。図 5 に一例を示す。この場合、N 個のノードにジオメトリを分散することで、1 node ではメモリに入りきれないシーンでもレンダリングできることを考慮する。今回の結果は 40 万ポリゴン x N のデータを分散レイトレーシングで処理できることを示している。たとえば N が 10 であれば、400 万ポリゴンを分散レイトレーシングによりレンダリングできるということになる。Engine データを N 個のノードに複製して、N ノードで分散レイトレーシング(シェーディングはアンビエントオクルージョン)した結果になる。Engine モデルは、空間内に $M \times M (= N)$ となるように配置している。

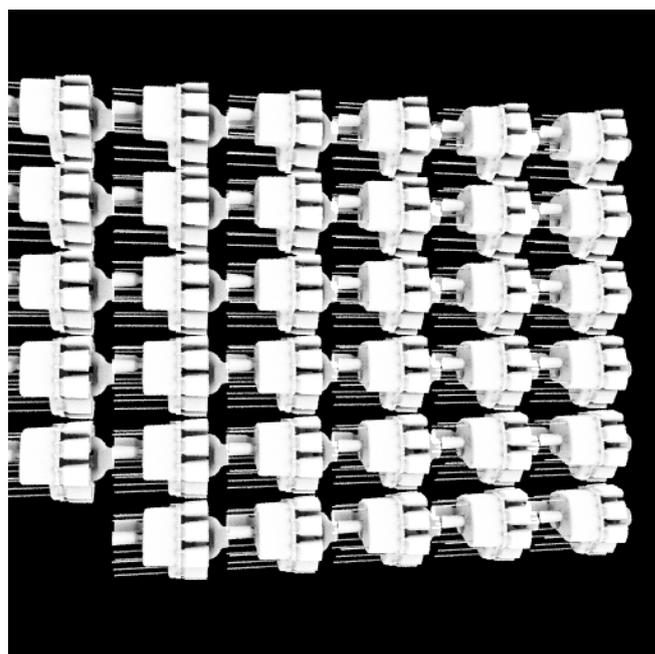


図 5 評価に用いたエンジン部品 (36 並列)

2.2.3. 結果

レンダリング時間の結果を図 6 に示す。レンダリング速度がノード数が増えるごとに高速になっているが、これはノード数の増加も影響している。加えて、今回の場合はノード数が増えると物体数が増え、隙間のピクセル(背景ピクセル)が増えることにより全体でのレイトレーシング処理量が減っているというのも影響として考えられる。

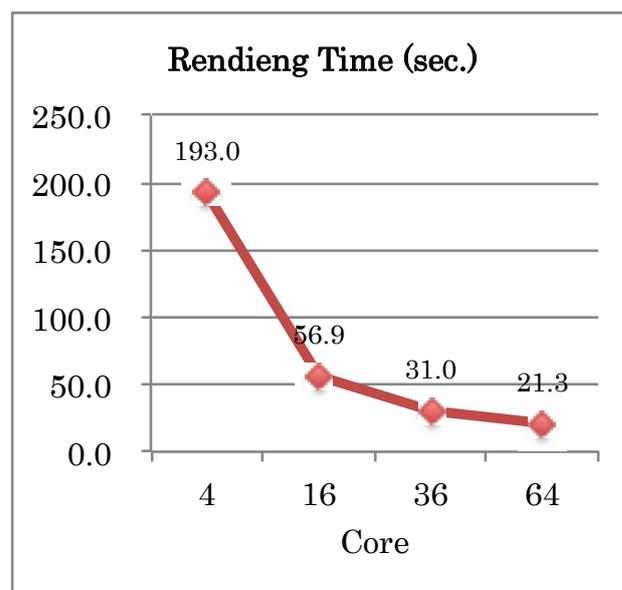


図 6 レイトレーサーの並列性能

大規模なポリゴンに対しても分散並列性を活かして性能が向上することが実証できた。

3. まとめ

大規模データ可視化システムの機能開発を実施した。本年度の開発項目が終了すると、基本的な可視化機能とデータ構造に対する分散並列可視化が可能になり、ユーザ評価の準備ができるようになる。今後、引き続き機能追加や性能評価を実施し、大規模解析の後処理をサポートできるようにする。

4. 今後の計画・展望

GPU を持たない汎用 CPU クラスタで、大規模なバッチジョブの可視化を検討中。Mesa による S/W レンダリングへの対応を行い、MPC による大規模並列バッチジョブで動作の確認と性能の調査を実施予定である。

LSV の H/W レンダリングにおいて、100 並列までの性能評価が実施できた。100 並列以上の並列実行は GPU を持たない MPC で実施する必要がある、現時点では未実施。今後は S/W レンダリングを用いて更なる並列化を行い、MPC による性能評価を実施予定。

5. 利用研究成果が無かった場合の理由

公表できるまでの機能実装、本年度開発システムの性能評価に至っていないため。