

宇宙機開発における大規模 CFD 解析結果の遠隔並列可視化

堤 誠司*1, 高木 亮治*1, 嶋 英志*1

Study on Remote and Parallel Visualization of Large-Scale CFD Results for Aerospace Development

Seiji TSUTSUMI*1, Ryoji TAKAKI*1 and Eiji SHIMA*1

Abstract

PC cluster is installed to evaluate the parallel and remote visualization of large-scale computational results produced by the supercomputer in the remote site. It is revealed that more than 60 % of the time for visualization is required for reading the binary data of the computational result. Partitioned file format such as VTK multi-block is desirable for reading large-scale data in parallel. High scalability more than 80 % is obtained by using the parallel rendering and image compositing. Methodology to extract the region of interest (ROI) based on the VTK multi-block format file is developed, and reduction of reading data size is realized for large-scale computational result without losing the parallel I/O performance. The knowledge obtained here will be utilized for the post-processing system of the large-scale computations done by remotely connected supercomputers.

1. 背景と目的

JAXA/ 情報・計算工学 (JEDI) センタでは数値流体シミュレーション (Computational Fluid Dynamics, CFD) を利用して JAXA プロジェクトの設計開発支援を行っている。1-3 これらの解析では突起物を含むロケット全機周りの解析やロケット打上げ時の音響振動に関する非定常解析 (Fig.1) など時空間スケールの大きな解析をスーパーコンピュータを利用して実施している。2009 年に JAXA では統合スーパーコンピュータシステム (JAXA Supercomputer System, JSS) を導入し、旧システムの約 10 倍以上にあたる 130TFLOPS の理論演算性能を有することとなった。計算機性能の増加に比例して計算規模も増加し、例えばロケット打上げ時音響解析では 1 ケース当たりのデータ量が 250~500GB 程度に達した。更に、統合スーパーコンピュータ (JSS) が調布航空宇宙センタに設置されたことにより相模原や筑波といった他地区からは遠隔利用をするようになった。

JSS 本体が設置されている調布航空宇宙センタと作業場所の相模原キャンパスは現在 1GbE の回線で接続されており、bbcp を利用すると 20MB/sec 以下の転送速度しか得られない。従って 500GB のデータを転送するためにおよそ 7 時間を要し、計算データを手元の端末に転送して可視化処理をする従来のやり方では転送時間・ローカルディスク容量の両面から考えて現実的ではない。また結果を解析するという側面から考えると可視化結果をスーパーコンピュータのある遠隔地でバッチ処理により画像を生成するのみでは不十分であり、得られた結果をインタラクティブに表示・操作しながら流れ場の観察を行うことが必須である。しかし大規模なデータに対して手持ちのワークステーションではメモリ・ハードディスク容量の不足や CPU・GPU 処理速

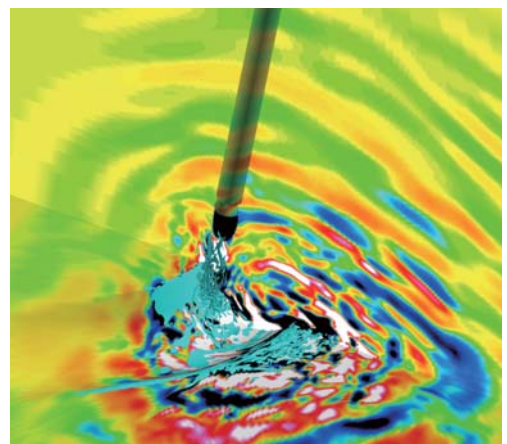


Fig.1 M-V ロケット打上げ時の音響解析。静圧分布と Q 値 (渦構造) を表示。

*1 宇宙航空研究開発機構 情報・計算工学センター (JAXA's Engineering Digital Innovation (JEDI) center, JAXA)

度が不足し、結果データの読み込みや等値面などの表示に多大な時間を要する。その結果、ファイル操作や描画の待ち時間が作業の大半を占めることになるため解析結果の視点変更や等値面変更といったインタラクティブ性が損なわれる。大規模な流体シミュレーションでは計算結果をスーパーコンピュータ上から移動せずに遠隔可視化を行い、かつ可視化プロセスの並列化等の手段を用いてインタラクティブ性を確保することが重要な課題である。

そこで JAXA の調布航空宇宙センタに設置された JSS を利用して計算した数値シミュレーション結果に対し、SINET3 (1GbE、実転送速度は scp で約 7MB/sec) で結ばれた JAXA 相模原キャンパスからステアリング性を損なうことなく可視化解析できることを目的とした可視化専用の PC クラスタ (可視化クラスタ) を構築した。対象とする計算結果は JSS で想定される規模のデータ容量 (格子点数で 1 億から 10 億点規模) とする。本稿では構築した可視化クラスタを紹介し、この可視化クラスタを利用して調べた並列可視化性能や遠隔可視化性能を議論する。次に、ペタフロップス級の次期スーパーコンピュータシステムで取り扱う規模の計算結果 (格子点数で 10 億から 100 億規模) に対してもステアリング性を保持して可視化解析を実施するため、計算結果から必要な空間情報のみを抽出することによって読み込むデータサイズを削減する Region of Interest (ROI) 技術について紹介する。

2. 大規模可視化に要求される機能

2.1. 遠隔可視化

Figure 2 に遠隔地にあるデータを可視化するための方式をまとめる。SINET 経由で JSS から生データを相模原に転送する方式が最もプリミティブな方式 (Fig.2 の左に示す) であるわけだが、第 1 章で述べたように 500GB 程度の計算結果を転送するには 7 時間程度の時間を要する。一方、計算結果の生データを加工してデータサイズをリダクションした後に転送するのが Fig.2 の中央に示した方式である。つまり、JSS のある調布航空宇宙センタに可視化装置を設置し、JSS のディスクから可視化装置にいったんデータを転送して生データをよりサイズの小さいポリゴンデータ、もしくは画像データに加工する。そして最も通信速度が遅い調布と相模原を結ぶ 1GbE には変換後の容量の小さくなったポリゴンデータか画像データを転送する。例えば Fig.1 に示した M-V ロケットの解析例 (格子点数 2700 万点) の場合、時間 1 ステップ当たりの生データは 1GB 程度であるのに対して可視化結果のポリゴンデータはせいぜい 10MB 程度、画像データは更に小さい 1MB 程度であり、データを加工することで最大 1/1000 倍程度のリダクションが可能である。非定常解析では時間方向にデータ量が数百倍に増える上、JSS を想定した計算規模は更にこの 10 倍程度大きくなることから、転送するデータをリダクションする効果は大きい。一方、Fig.2 の右側に示すようにリモートデスクトップを使うことで可視化クラスタの直接操作も可能とすることも 1 つの方式であり、VNC (Virtual Network Computing) といったソフトウェアの利用や Elsa 社の VIXEL といったハードウェアを利用することが考えられる。

以上のように遠隔可視化は必要不可欠な機能であり、ハードウェア/ソフトウェア環境を構築する上で可視化装置とクライアント端末との通信手段について留意する必要がある。

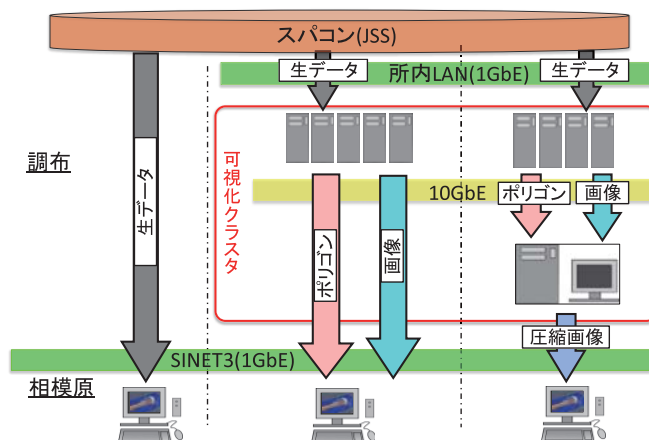


Fig.2 遠隔可視化方式。

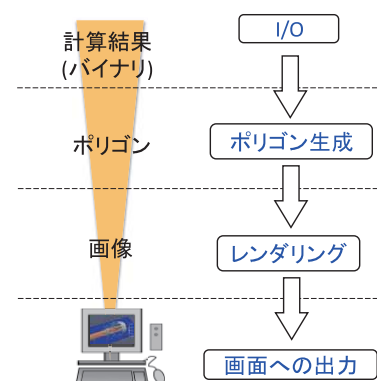


Fig.3 並列可視化プロセス。

2.2. 並列可視化

大規模な計算結果を高速で処理するためには可視化プロセスの並列化が要求される。Figure 3 に示すように可視化プロセスは I/O, ポリゴンの生成, レンダリングの 3 段階に分けられ, それぞれのプロセスで並列化が可能である。2.1 章でも述べたように Fig.1 の M-V ロケットの解析例では時間 1 ステップ当たりの生データが 1GB 程度にあるのに対してポリゴンデータ 10MB 程度, 画像データは 1MB 程度と取り扱うデータ容量は I/O ほど大きく, レンダリングして生成された画像データほど小さい。I/O の並列化に関してはファイルフォーマットや共有ファイルシステムとの関係に留意しなければならない, 高いネットワーク性能が要求される。ポリゴン生成の並列化は複数の CPU やコアを用いることで行われ, 通信ライブラリには一般的に MPI が用いられる。レンダリングの並列化についてはインタラクティブ性を確保するために GPU を用いた並列ハードウェアレンダリングが必須と考えられる。米国ローレンスリバモア国立研究所, 米国サンディア国立研究所などでも既に多数の GPU を用いた並列レンダリングが実施されており, 大規模可視化において成果を上げている。⁴⁵ 従って, I/O, ポリゴンの生成, レンダリングの 3 段階で並列化が可能となるようにハードウェア/ソフトウェア (ファイルフォーマットを含む) を考える必要がある。

2.3. 結果情報の抽出機能

将来的に超大規模な計算結果を可視化することを考えると得られた結果すべてを読み込み, そして可視化することは処理時間やメモリ消費量といった観点から不可能となると考えられる。そのため, 時空間の両方向に関してデータを間引く機能, 着目する領域 (Region of Interest, ROI) のみを選択的に読み込む機能, 着目する現象を支配する特徴的な物理量のみを抽出する機能といった結果情報の必要なもののみを抽出する機能が不可欠となる。本機能は主にソフトウェアに対して実装されるべき項目である。

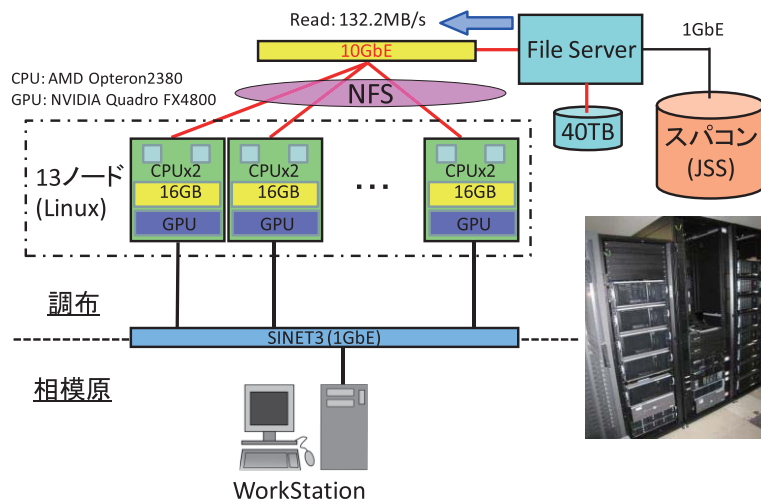


Fig.4 遠隔並列可視化クラスタシステム

3. ハードウェア/ソフトウェア環境

3.1. 可視化クラスタシステムの構成

本研究で構築した遠隔並列可視化クラスタシステム (以下, 可視化クラスタ) の概要を Fig.4 に示す。可視化クラスタは 13 ノードのワークステーションから構成される。各ノードは 2 台の CPU と 1 台の GPU を搭載しており, メインメモリは 16GB である。(CPU, GPU の諸言は Fig.4 に併記) 各ノード, 及びファイルサーバは 10G Ethernet で接続し, 共有ファイルシステムには NFS を使うことで全ノードがファイルサーバの RAID ディスクをマウントしている。こうすることによって 2.2 章で述べた I/O の並列化ができるようにした。また MPI ライブラリをインストールすることでポリゴンの生成, 及び (ハードウェア) レンダリングの並列化も 10GbE 経由でできるようにした。4.4 章で議論するように JSS で想定される最大規模の計算データ (格子点数 7 億点) に対して合計 5 ノード (計算ノード 4 台 + フロントエンド 1 台) あれば十分であることが分かっており, 本システムは 2, 3 人の研究者が同時に解析できる規模の計算機リソースを有している。なお, ファイルサーバのディスク装置からフロントエンドへの転送速度は Bonnie++ を用いた実測で 132.2 MB/sec である。一方, 2.1

章で述べたポリゴン、もしくは画像データを転送する遠隔可視化を実現するため可視化クラスタの全ノードは SINET を経由して相模原のクライアント端末と ssh を利用して相互に通信可能となるようにした。更に、可視化クラスタでは Elsa 社の VIXEL を導入しており、ハードウェアを利用したりリモートデスクトップも使用できるようにした。もちろん、JAXA 機構内のファイアーウォールに関する設定も別途行う必要があった。本来、可視化クラスタから JSS のディスクを直接マウントできるようにすることが望ましいが、運用上の制約から JSS のディスクをマウントすることはできなかった。そのため、可視化クラスタは独自にファイルサーバと RAID ディスクを有しており、調布航空宇宙センタの所内 LAN 経由で JSS の計算結果を一旦可視化クラスタのディスクにコピーするようにした。

3.2. 可視化ソフトウェアの概要と特徴

本稿で使用した遠隔可視化、及び並列可視化が可能な可視化ソフトウェアを Table 1 に列挙する。FIELDVIEW は Intelligent Light 社、EnSight は CEI 社の商用ソフトであり、ParaView はオープンソース⁶である。

Table 1 可視化ソフト一覧

	並列化機能			Server-Client間 転送データ形式	長所/短所
	I/O#	ポリゴン 生成	レンダリ ング		
FieldView	Plot3d形式 (ブロック単位)	○	×	・ポリゴン	シンプルで使い易 い/高価!
EnSight	Plot3d形式 (自動分割)	○	○	・ポリゴン ・画像+イメージ 重畳	多機能/複雑な仕 様, 高価!
ParaView	・VTK MultiBlock形 式 ・XDMF形式 (HDF5)	○	○	・ポリゴン ・画像+イメージ 重畳	無料, カスタマイ ズ可能/マニュアル が殆どない, 不 完全な機能が多 い。

構造格子のみに着目

いずれのソフトウェアも可視化クラスタの各ノード上で並列処理にてポリゴン生成を行い、クライアント側にポリゴンを転送して表示を行う機能を有している。これを本稿では“ポリゴンモード”と呼称する。一方、EnSight と ParaView に関しては各ノードの GPU を使用した並列レンダリングによる画像転送・重畳の機能を有しており、これを“ビットマップモード”と呼称する。ポリゴンモードではクライアントへいったんデータを転送すれば視点変換等はクライアント側の作業となる。一方、ビットマップモードは可視化画像の視点変換をするたびに画像データ（ビットマップデータ）の転送が必要となるが、ポリゴンよりもデータサイズが小さいためネットワークへの負荷が軽いことが特徴である。なお、FIELDVIEW に関してはビットマップモードこそ実装していないものの並列化に要するセットアップが容易であり、遠隔可視化を行う際にもこれまでとほとんど変わらない感覚で操作でき、他のソフトに比べユーザーフレンドリーである点の特徴である。FIELDVIEW と EnSight が Plot3D フォーマットを使用しているのに対し、ParaView では VTKMultiBlock フォーマットを採用している。格子生成の都合や並列計算を実施するために計算格子は予め分割された領域（ブロック）から構成されているわけだが、Plot3D フォーマットは単一ファイルとしてブロックを取り扱っているのに対して VTKMultiBlock はブロックごとのデータを分散ファイルとして取り扱っている点異なる。

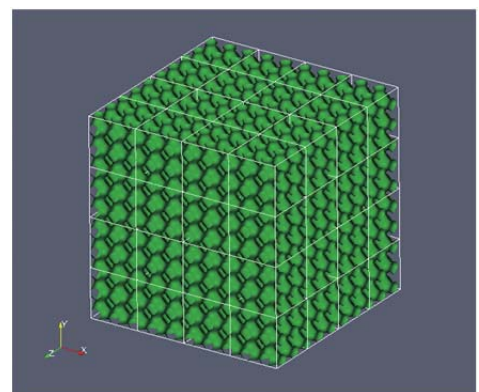


Fig.5 評価対象データ
(1.3億メッシュ, 4.2GB, 1900万ポリゴン)

4. 可視化クラスタの性能評価

4.1. 評価対象データ

Figure 5 に本研究で評価のために用いたデータを示す。格子サイズは $512 \times 512 \times 512$ (格子点数 1.3 億点)、64 個のブロックからなるマルチブロック構造格子である。sin 関数からなるダミーデータを与えており、Fig.5 に示した関数値 $f(x, y, z) = 0$

の等値面描画（ポリゴン数は1900万）をレンダリング性能測定に用いた。Figure 1に示したM-Vロケットの解析例では格子点数が2700万点でジェットの流れ構造を示すのに要するポリゴン数は100万である。またJSSを使った格子点数が約1億点のイプシロンロケットの音響解析における典型的なポリゴン数は540万程度であり、Fig.5の評価対象はレンダリング性能をやや重視した比較となる。なお、本稿では大規模な非定常シミュレーションで主として使用している構造格子のみを対象としており非構造格子は対象としない。

Table 2 並列度に対するCPU・コア数の内訳（CPU数×コア数）

並列度	クライアント	サーバ1	サーバ2	サーバ3	サーバ4
1	1x1				
4	1x1	1x1	1x1	1x1	1x1
8	1x1	2x1	2x1	2x1	2x1
16	1x1	2x2	2x2	2x2	2x2
32	1x1	2x4	2x4	2x4	2x4

4.2. 並列可視化性能の評価

本研究では可視化クラスタのフロントエンド上でクライアントを立ち上げ、1~32並列時におけるファイルの読み込みや等値面の描画に関する並列化効率を得た。並列時に使用されるCPU数及びコア数の内訳はTable.2に示した通りである。これらの評価に当たっては毎回メモリキャッシュをクリアし、計3回の計測を実施してその平均を使用した。

ファイルの読み込み時間と高速化率をFigs.6と7に、等値面描画に関する並列化効率をFig.8に示す。なお、高速化率・並列化効率はそれぞれ以下のように定義する。

$$\text{高速化率} = \frac{\text{非並列時の処理時間}}{n \text{ 並列時の処理時間}}$$

$$\text{並列化効率} = \frac{\text{高速化率}}{\text{プロセス数}}$$

Figure 6より、非並列時の読み込み時間はいずれのソフトウェアも80~110 (Sec)であるが並列度に応じて読み込み時間が短縮する。32並列時ではVTKMultiblockフォーマットを用いているParaViewが11秒と最も早い結果となったが、非並列時の速度はFIELDVIEWとほぼ同等であることからファイルの読み込み時間の並列化は各ブロックをあらかじめ分散配置したフォーマットがより有効であることが分かる。EnSightは他のソフトウェアよりも若干読み込みに時間を要しているが、FIELDVIEWやParaViewはブロックを分散させることによってロードバランスを取るのに対しEnSightはPlot3Dデータを自動分割処理を行った上で読み込みを行う仕様になっているためと考えられる。Figure 7よりいずれのソフトウェアも並列度に応じて読み込み時間は高速化するものの、概ね8~16並列で高速化率は限界に達している。特にParaViewに関しては16並列と32並列で処理時間がほぼ一致しており、ノード間の通信速度がボトルネックになっていると推測される。

等値面の描画に関する並列化効率をFig.8にて比較する。ビットマップモードを利用したEnSight、ParaViewの両ソフトウェアでは8並列までの並列化効率がおおよそ80%程度となっており、比較的高い値を維持している。これに対してポリゴンモード時の並列化効率はビットマップモードと比較して総じて低く、並列度の増大に伴い並列化効率が大きく低下する。特にParaViewの場合、ポリゴンモードのみの状態では8並列時でも10%程度の効率しか得られていない。本稿で評価に用いた等値面のポリゴンデータ（1900万ポリゴン）ではポリゴン転送時にネットワークがボトルネックとなり性能が伸びていないと考えられる。従って、大規模解析の可視化において大量のポリゴン転送を必要とする場合、ビットマップモードを使用した方がより高いスケーラビリティを確保できる。ただし、ビットマップモードを用いた場合でも高い並列化効率を得られているのは8並列時までで、それ以上にプロセス数を増やすと効率が急低下する。その傾向はEnSightの場合で特に顕著で、8並列時80%程度だった並列化効率が16並列では50%程度まで低下している。並列度に対するCPU数とコア数の内訳はTable 2に示すとおりであり、並列化効率が低下する16並列以降では1CPU内で複数コアを使用していることからCPUのメモリバンド幅の影響を受けていると考えられる。つまり並列可視化の場合はむやみにスレッド数を稼いでも並列化効率は上がらず、コア数よりもCPU数を増やした方が費用対効果の点で有利である。

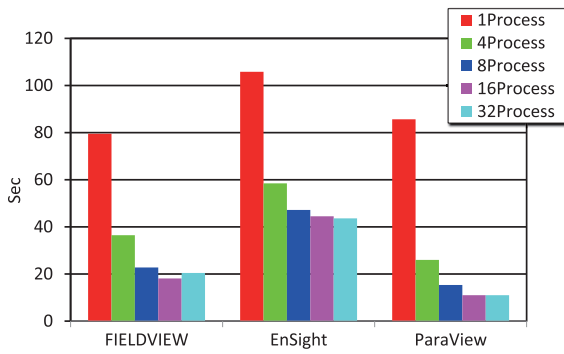


Fig.6 I/O の所要時間.

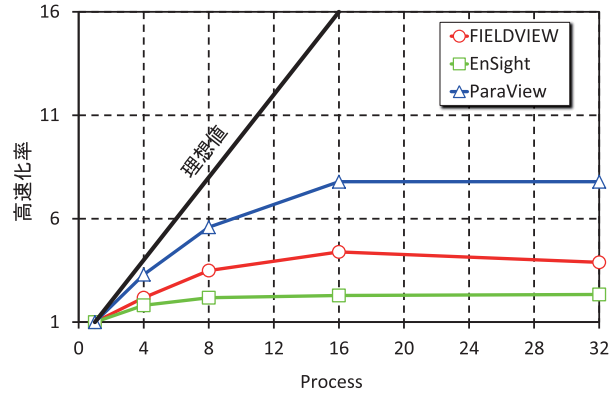


Fig.7 I/O の高速化率.

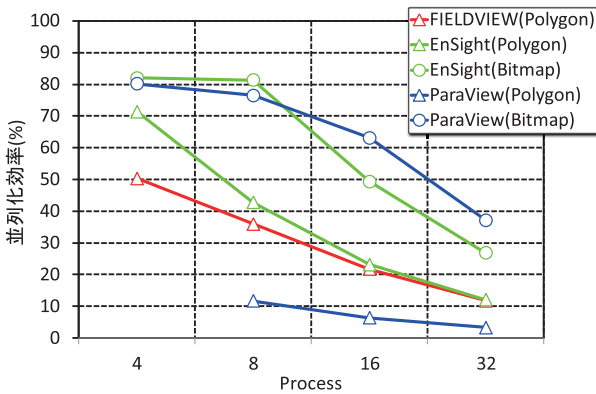


Fig.8 描画性能.

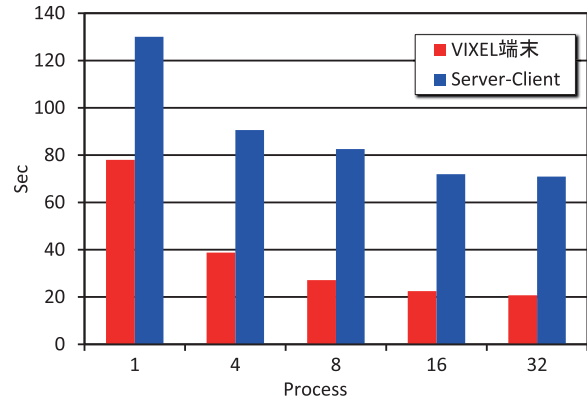


Fig.9 遠隔可視化における等値面描画時間の比較.

4.3. 遠隔可視化性能の評価

調布に設置した可視化クラスタから相模原キャンパスにあるクライアント端末への遠隔可視化について評価する。可視化システムのフロントエンドを VIXEL(Elsa 社) のリモートデスクトップ機能を用いて操作した場合と、FIELDVIEW を用いて相模原キャンパスのクライアント端末にポリゴンデータを送付した場合の等値面描画時間を比較したのが Fig.9 である。VIXEL と比較して FIELDVIEW を用いてポリゴンを送付する場合はおよそ 2~3 倍程度の時間を要しており、調布 - 相模原間の転送速度がネックになっていることが分かる。一方、調布 - 相模原間の SINET3 の環境化で実際に可視化解析を実施すると、VIXEL では見かけ上のフレームレートが回線速度に応じて低下し通信量が増える日中はマウスの操作にストレスを感じたが、FIELDVIEW ではそれほどストレスを感じなかった。

Table 1 に示したように可視化ソフトウェアによってサーバとクライアント間で転送するデータが異なる。FIELDVIEW はポリゴンデータを転送するためサーバには GPU を必要とせず、クライアントへいったんポリゴンデータを転送すれば視点変換等はクライアント側の作業となる。一方、EnSight や ParaView はポリゴンデータに比べて容量が小さいビットマップデータを転送し、可視化画像の視点変換をするたびにビットマップデータの転送が必要となる。定量的に比較することは難しいが、実際に使用して可視化解析を実施するとビットマップモードの方が遠隔性能に優れていることが分かった。

4.4. 大規模可視化性能の評価

JSS で想定される最大規模の計算データに対しても対応できるかを確認するため、32 並列の下で格子点数 896×896×896 (7 億点) の格子を用いて Fig.6 と同様の等値面描画を実施し、メモリ量と I/O・描画速度を調べた。可視化クラスタの 4 ノードを使用した 32 並列時のシステム最大のメモリ量は 64GB であるが、Fig.10 に示すようにメモリ量という観点からはいずれのソフトウェアも可視化処理を十分行える。Figure 11 には I/O と描画に要した時間を比較する。分散ファイルフォーマットである VTKMultiBlock フォーマットを使用し、更に各ノードの GPU を使ってレンダリングも並列化する ParaView は等値面の描画までに約 60 秒程度であり、7 億点規模の計算結果に対してもステアリング性を保って可視化解析が可能である。また、Fig.11 に示した結果から可視化全体に要する時間の 60~80% は I/O が占めており、I/O がホットスポットであ

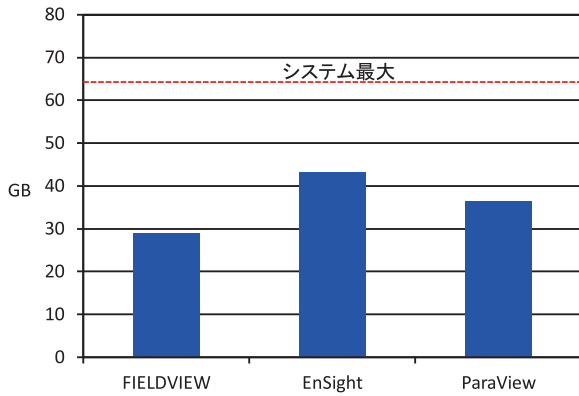


Fig.10 メモリ消費量.

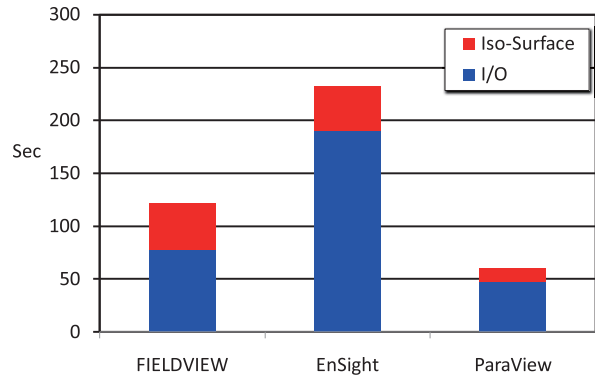


Fig.11 I/O+ 描画時間.

ることが分かる。なお、4.1 章でも述べたように今回の評価データはポリゴン数が多く、実際の解析では I/O の占める割合はさらに大きくなる。本研究で構築した可視化クラスタは 10G の Ethernet でノード間を接続し共有ファイルシステムとして NFS を使用しているわけだが、可視化クラスタを構築する際は例えば Lustre といった並列性能に着目した共有ファイルシステムを利用して如何に並列 I/O 性能を向上させるかがキーポイントであると分かる。

5. VTK based ROI 機能の構築と性能評価

4.4 章から可視化処理のホットスポットは I/O であることが判明したわけだが、格子点数が 100 億点規模（データ容量は 300GB 程度）を超えるような将来的に実施される大規模な計算結果を取り扱うことを考えると、本研究で紹介した可視化クラスタの並列度を単純に上げるだけではステアリング性能を保持し続けるには限界がある。そのため、得られた計算結果から必要となる時間・空間情報のみを抽出し可視化クラスタが読込むデータ容量を削減することが必要となる。本研究では GUI からマウス操作で、もしくはテキストファイルに記述することによって着目する空間 (ROI) を指定し、その空間に対応するデータのみを並列 I/O 性能を損なうことなく読み込みが可能な ROI 機能を作成した。

本研究で作成した ROI 機能 (VTK based ROI と呼ぶ) は ParaView にプラグインとして実装した。VTK based ROI の主要機能は以下の 4 つである。

1. ParaView 上で計算領域内の予め分割されたブロックのブロック番号表示とマウスを使った ROI 領域の選択機能
2. ROI 領域の領域指定ファイル (roi.txt) と ROI 領域を抽出した計算データ (ROI データ) の出力
3. 別途テキストエディタで作成した領域指定ファイル (roi.txt) から ROI データの作成機能
4. 作成した ROI データを元データのデータ構造 (ブロック番号、境界面情報など) を保ちつつ表示する機能

VTK based ROI は VTKMultiBlock フォーマットデータのみを対象としている。VTKMultiBlock フォーマットは Fig.12 に示すように格子・結果を記述したデータファイル (hoge_XXX.vts) をブロック単位で保存する分散ファイルフォーマットであり、vts ファイルは XML ファイル (hoge.vtmb) で管理されている。VTK based ROI ではデータの実態 (hoge_XXX.vts) には手を付けず、上層の XML ファイル (hoge.vtmb) を編集することで容易にブロック単位の ROI 抽出が可能である。

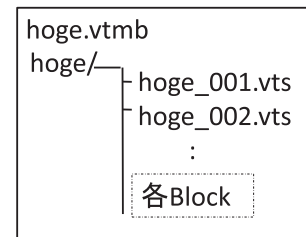


Fig.12 VTKMultiBlock フォーマット.

M-V ロケット打上げ時の音響解析²を例に VTK based ROI のプラグインを評価した。CFD 解析では一般的に計算領域外縁 (境界条件) の影響を極小化するため着目する領域よりも大きな計算領域をとる。M-V ロケットの解析では Fig.13(a) に示すように平面方向に 1000m、高さ方向に 500m 程度の広い空間を計算している。一方、実際に音の発生や伝播に着目したい領域は Fig.13(b) に示した 1 段モータ出口と火炎偏光板近傍のたかだか平面方向に 20m、高さ方向に 10m 程度の領域である。今回使用したプラグインを使用しない場合、計算格子を生成するソフトウェア (本研究では Pointwise 社の Gridgen を使用) で抽出すべきブロック番号を予め調べ、そして vtmb ファイルをテキストエディタで直接編集する必要があった。本研究で作成したプラグインを使用する場合、主要機能 1. と 2. を使ってマ

ウス操作で領域指定ファイルを作成するかもしれない。Gridgen とそのスクリプト機能を利用して領域指定ファイルを作成し、次に主要機能 3. を利用して vtmb ファイルを自動作成できるため、ROI 抽出が簡便になるとともに結果表示までの時間を大幅に短縮可能となった。以上のように元データのデータ構造を保ったまま（主要機能 4.）着目領域の抽出が非常に簡単に早くできるため、将来的な大規模データを取扱う一つの手法として期待される。

なお、本研究では VTK based ROI だけでなく VTKMultiBlock データリーダーや ParaView の利便性向上に向けたプラグインを開発しており、全て無償で配布している。

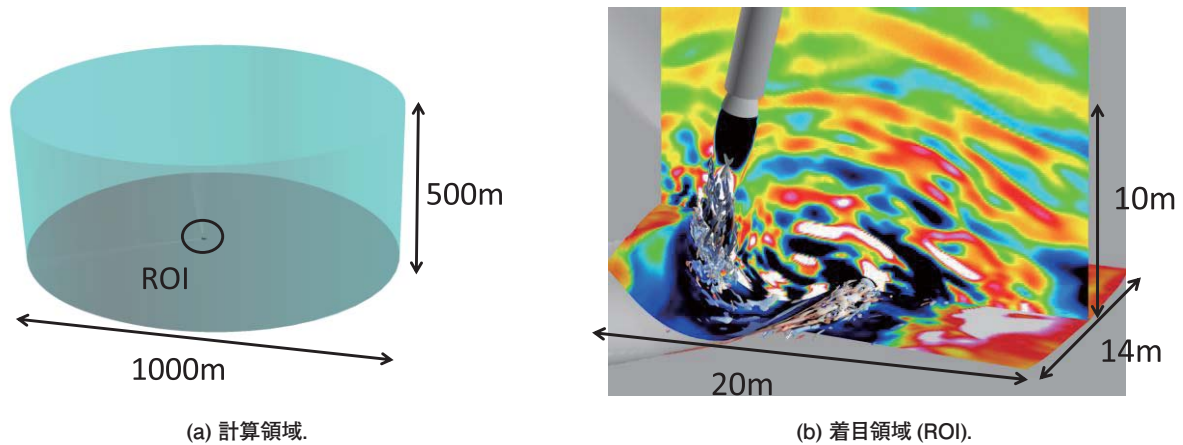


Fig.13 計算領域全体と着目領域.

6. まとめ

調布航空宇宙センタに設置された JAXA の統合スーパーコンピュータ (JSS) を用いて計算した数値流体シミュレーション (CFD) 結果を、相模原キャンパスからステアリングよく可視化解析作業を実施するために必要となる遠隔並列可視化技術に関して検討を行った。JSS のある調布航空宇宙センタ内に可視化専用の PC クラスタを導入した。10G の Ethernet で接続されたノードはハードウェアレンダリングを可能とするために GPU を搭載しており、I/O・ポリゴン生成・レンダリングの各段階での並列化が可能とした。また、可視化クラスタは SINET3 を経由して相模原の端末に接続しており、ポリゴン・ビットマップ画像を転送する遠隔可視化ができるようにした。更に、本研究では Elsa 社の VIXEL を導入し、ハードウェアを利用したリモートデスクトップも使用できるようにした。一方、本研究で用いたソフトウェアは遠隔可視化、及び並列可視化が可能な FIELDVIEW (Intelligent Light 社)、EnSight (CEI 社)、及び ParaView (オープンソース) の 3 つとした。並列性能について調べた結果、I/O の並列化に関しては VTKMultiBlock 形式に代表される分散ファイルフォーマットが高い読み込み性能を示した。次に描画性能を比べた結果、各ノードでレンダリングして画像データ (ビットマップデータ) を転送してフロントエンドでイメージ重畳する並列化方式が高い並列化効率を示した。I/O と描画に要する時間を比較すると I/O がホットスポットであり、並列可視化においては I/O 性能をいかに向上させるかが重要であることが分かった。遠隔化性能に関しては今回試行したソフトウェアの機能でもそれほどストレスなく可視化解析が可能であったが、ポリゴンデータよりもビットマップデータを転送する方式がより遠隔性能に優れていることが分かった。JSS で想定される最大規模の計算データ (格子点数 7 億点) に関して本可視化システムの 4 ノード (合計 64GB メモリ) を利用すればステアリング性を維持して可視化解析が可能であることが分かった。最後に、格子点数が 100 億点を超えるような次期スパコンで実施される数値解析を見据え、ホットスポットである I/O 性能を確保するために得られた計算結果から着目する領域 (ROI) のみを並列化性能を低下させることなく抽出可能な VTK based ROI 技術を ParaView のプラグインとして作成した。その結果、プラグインなしの場合と比べて ROI の抽出が 1/6 程度まで時間を短縮できた。今後ますます増えると考えられる遠隔地の大型計算機を利用する環境において、本研究で得られた知見や作成したプラグインはステアリング性を維持して可視化解析を実施するために重要はであり、JAXA の次期スーパーコンピュータシステムに対しても積極的に導入していきたいと考えている。

参考文献

- [1] Tsutsumi, S., Fukuda, K., Takaki, Shima, E., Fujii, K., and Arita, M., "Generation and Propagation of Pressure waves form H-IIA Launch Vehicle at Lift-off", AIAA Paper 2008-390, Jan.2008.

- [2] Tsutsumi, S., Fukuda, K., Takaki, R., Shima, E., Fujii, K., and Ui, K., “Numerical Study on Acoustic Radiation for Designing Launch-pad of Advanced Solid Rocket,” AIAA Paper 2008-5148, 2008.
- [3] 北村圭一, 藤本圭一郎, 葛生和人, 野中聡, 入門朋子, 福添森康, 嶋英志, “次期固体ロケット空力特性についての数値解析,” 航空宇宙技術, Vol.10, 2011, pp.1-10.
- [4] http://www.nvidia.com/object/IO_27539.html
- [5] https://publicaffairs.llnl.gov/news/news_releases/2005/NR-05-11-04p.html
- [6] Andy Ceilnik, BerkGeveci, Kenneth Moreland James Ahrens, and Jean Favre, “Remote Large Data Visualization in the ParaView Framework”, EGPGV2006.