

jh170048-NWH

可視化用粒子データを用いた遠隔地からの対話的 In-Situ 可視化

河村拓馬（日本原子力研究開発機構）

概要

In-Situ 可視化は、スーパーコンピュータ上でシミュレーションと同時に可視化を行うことで結果データの I/O を避け、大規模シミュレーションを確実に可視化できる手法である。しかし、従来の In-Situ 可視化では、可視化処理のスケーラビリティが不足しているためシミュレーションのコストを圧迫するという問題がある。加えて、対話的操作が困難であるため可視化の失敗が発生し、シミュレーションの妥当性検証等で再計算が必要になるという問題がある。本課題の目的は、可視化用粒子データを用いた In-Situ 可視化システムを構築し、最新の GPGPU、XeonPhi、および、FX100 環境に対して、シミュレーション性能を劣化させること無く対話的な大規模可視化が可能であることを実証することである。可視化用粒子データの生成は低コストのモンテカルロ法で実行され、高いスケーラビリティを得ることができる。本年度は、FX100 及び KNL 上での開発を完了した。そして、Oaforest-PACS（東大/筑波）を用いて約 10 万コアまでのストロングスケーリングを達成し、結果を国際会議 ISC2017 のワークショップ WOIV で報告した。

1. 共同研究に関する情報

(1) 共同研究を実施した拠点名

- ・ 名古屋大学 情報基盤センター
- ・ 東京工業大学 学術国際情報センター
- ・ 東京大学 情報基盤センター

(2) 共同研究分野

- 超大規模数値計算系応用分野
- 超大規模データ処理系応用分野
- 超大容量ネットワーク技術分野
- 超大規模情報システム関連研究分野

(3) 参加研究者の役割分担

参加者：青木 尊之

役割：TSUBAME3.0 利用技術の提供

参加者：小野寺 直幸

役割：CityLBM への In-Situ PBVR の導入

参加者：山下 晋

役割：Jupiter への In-Situ PBVR の導入

2. 研究の目的と意義

近年のペタスケールシミュレーションの台頭により、結果データを手元の PC に転送してポスト処理を行う従来の可視化手法は、

転送時間の増大やクライアント PC のメモリ容量限界を越え、適用が困難となった。こうした結果データを可視化するため、遠隔地のサーバで可視化処理を行いクライアントで表示する、クライアント・サーバ型の遠隔可視化技術が開発された。

さらに現在では、GPGPU や XeonPhi、FX100 等の計算処理アクセラレータ、あるいは、メニーコアプロセッサによって演算性能が高まり、データ I/O が大きな障害となっている。このような背景から、結果の解析に必要な可視化に関して、結果データの I/O ボトルネックを避けるために、計算実行時に同環境を用いて可視化画像を生成する In-Situ 可視化が重視されている。しかし In-Situ 可視化には、以下に示す二つの課題がある。

1. 領域分割されたシミュレーションデータに対する可視化処理のためのデータ領域構成やデータ探索に伴う大域的通信により、可視化処理のコストがシミュレーション処理のコストを圧迫する。
2. In-Situ 可視化では、バッチ処理投入前に視点位置、色、不透明度等の可視化パ

ラメータを設定するため、可視化の失敗がしばしば発生する。

これらの問題を解決するため、可視化用粒子データを用いた対話的な In-Situ 可視化システム “In-Situ PBVR” を構築し、最新の GPGPU、XeonPhi、および、FX100 環境に対して最適化を行い、シミュレーション性能を劣化させること無く対話的な大規模可視化が可能であることを実証する。

提案システムで利用する可視化用粒子データとは、シミュレーションから得られた結果データ（ボリュームデータ）を可視化用粒子データに変換してボリュームレンダリング画像を生成する可視化手法である、Particle-Based Volume Rendering (PBVR) で利用している画像生成用のデータである。粒子ベースボリュームレンダリングでは伝達関数（色関数と不透明度関数）に物理値をマッピングしてボリュームデータを可視化する。この手法では不透明度は粒子の粗密で表現され、粒子データのサイズは不透明度の分布と画面解像度から計算されるため、元データのサイズに関わらず、1024x1024 ピクセルの画像で十分な画質を得るのに約一千万粒子（約 250MB）の粒子データが必要という結果が得られている。

In-Situ PBVR では、In-Situ 環境下で結果データを十分小さな可視化用粒子データに圧縮し、その粒子データをクライアントとなる PC に転送することで、遠隔地からの対話的な可視化を実行する。粒子データの生成は領域分割形状を変更することなく並列化され、デーモンプログラムを介してクライアント/サーバ間での可視化パラメータの変更が可能になる。システム構成を図 1 に示す。

開発システムを XeonPhi アーキテクチャおよび GPGPU アーキテクチャに最適化された燃料溶融複雑系解析コード Jupiter[1]および都市圏気流解析コード CityLBM[2]に適用し、

シミュレーションの性能を劣化させること無く対話的な大規模可視化が可能であることを実証する。

3. 当拠点公募型共同研究として実施した意義

昨今のシミュレーションにおいて結果データの可視化は必須となっており、In-Situ PBVR は、現行あるいは次世代の計算環境に対して汎用的に動作することが求められる。In-Situ PBVR は結合対象となる HPC アプリケーションに合わせてマルチプラットフォームでの動作を想定しており、CPU、GPGPU、そして XeonPhi 等のアーキテクチャに対する実装が必要になる。このため、様々なプラットフォームからなる計算資源を利用可能な当拠点公募型共同研究が、本研究の推進に不可欠である。

加えて、拠点間を結ぶ広帯域なネットワークの存在も本研究にとって重要である。開発する In-Situ PBVR は、シミュレーションと同時に生成した可視化用粒子データをクライアントに転送することで、遠隔地からのリアルタイムな可視化・解析が可能になる。SINET 5 が提供する広帯域ネットワークを利用することで可視化用粒子データを高速に転送でき、可視化作業を円滑に進めることができる。

このシステムは、大規模シミュレーションに対する実時間の In-Situ 可視化・解析が可能であり、膨大な作業時間や待ち時間が必要だった従来のポスト処理可視化、あるいは、可視化失敗に対するトライ・アンド・エラーが必要だった従来の In-Situ 可視化を駆逐する可能性がある。またエクサスケールのメーコアーキテクチャ候補である GPGPU や XeonPhi 等において実行されるステンシル計算のシミュレーションに対して、その性能を劣化させることのない可視化が可能になる

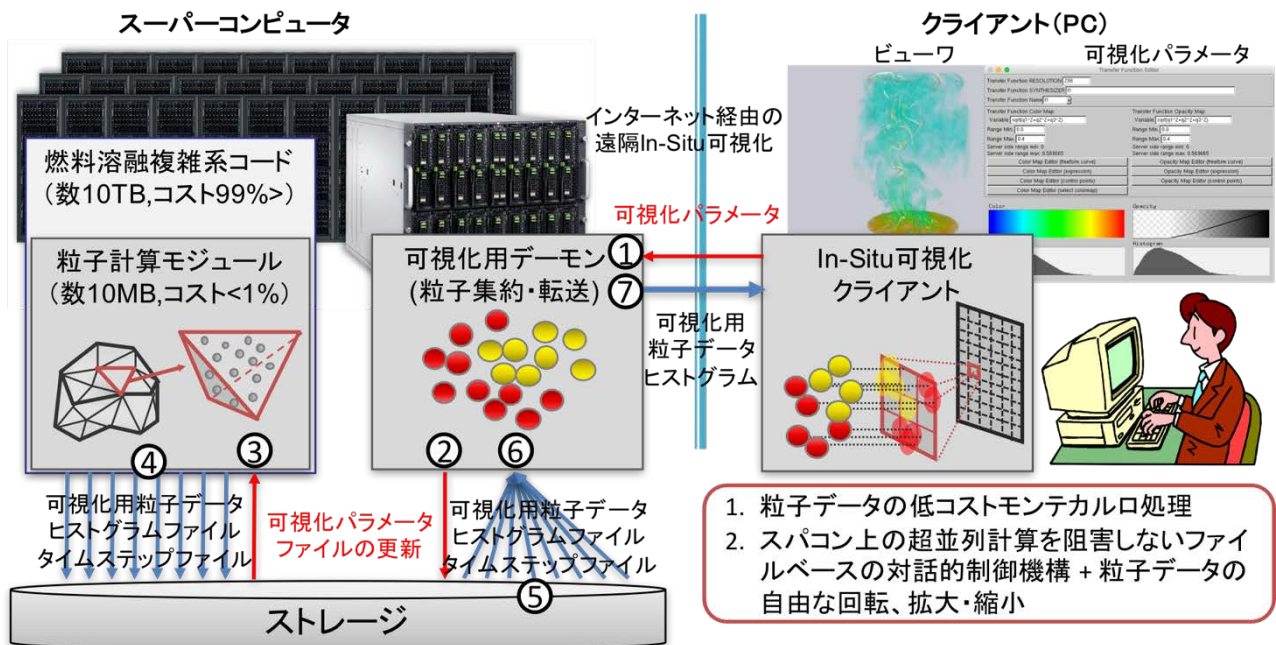


図 1 In-Situ PBVR のシステム構成

ことで、将来のエクサスケール計算への応用が期待される。

4. 前年度までに得られた研究成果の概要

新規課題なので特になし。

5. 今年度の研究成果の詳細

本課題では、原子力機構が所有するスーパーコンピュータ ICEX (Xeon) 上で開発してきた In-Situ PBVR のシステムを、FX100 および Oakforest-PACS (PFP) 上に移植し、可視化用粒子データの出力に関して新たな機能の開発を行なった。そして、In-Situ 可視化が対話的にできるかどうかを検証する動作試験を実施した。また、大規模シミュレーションを阻害しないかどうかを検証するため、問題規模を変えずに物理コア数を増加させるストロングスケーリング試験と、使用メモリ量の計測を実施した。

In-Situ PBVR

開発した In-Situ PBVR は、可視化手法として、流体分野で有効性を知られているボリュームレンダリングを採用しており、可視化用粒子データを用いてボリュームレンダリ

ングを実現するためのアルゴリズムとして、PBVR [3]を利用している。PBVR では、物理値に伝達関数（色関数と不透明度関数）をマッピング、計算結果の格子中に可視化用粒子データばら撒き、それを画像面に投影することでボリュームレンダリング画像を生成する。この手法では粒子データのサイズは不透明度の分布と画面解像度から計算される（図 2）。粒子の疎密は不透明度に比例し、モンテカルロサンプリングによって粒子の配置が決定される。色の値は、粒子位置から物理値を補間して計算される。投影された粒子データは、画像バッファに色情報が、デプスバッファに深度情報が溜め込まれ、それらの値を用いて最終的なピクセルの色が計算される。粒子データは元データのサイズに関わらず、十分な画質を得るのに約 250MB の粒子データが必要という結果が得られており、逆に、画質を荒くすることで 3 MB 程度の少ないデータサイズで可視化が可能である。

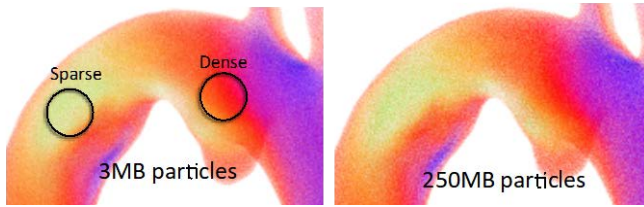


図 2 粒子データのサイズと画質

上記の粒子生成と画像生成の処理を分割することで、遠隔地での実行が可能なシステムとして In-Situ PBVR のシステムを構築した。遠隔地で実行される大規模シミュレーションのデータを小さな可視化用粒子データに変換し、手元のクライアント PC でボリュームレンダリングにより可視化する。そして、バッチ処理されるシミュレーションの実行時における対話的な In-Situ 可視化を実現するために、ファイルベースによる可視化パラメータの制御機構を開発した。これらの機能は、シミュレーションに結合される粒子計算モジュール、クライアント PC 上で画像を表示するための In-Situ 可視化クライアント、そしてファイルベースの制御を実行する可視化用デーモンにより構成されている (図 1)。In-Situ 可視化用クライアントと可視化用デーモンは、インターネットを介したソケット通信により接続され、可視化用デーモンはストレージを介したファイルのやり取りにより粒子計算モジュールを制御する。

In-Situ 可視化クライアント

In-Situ 可視化クライアントは、クライアント PC 上で起動され、可視化結果を表示するためのビューワ、そしてヒストグラムを表示し伝達関数を編集するための GUI から構成されている。この GUI は、多変量可視化を実現するための機能である、“伝達関数合成器” [4] として実装されている。伝達関数合成器では、結果データに含まれる多数の変量を組み合わせることで色や不透明度を設定することで、変量間の相関や特徴を形状として抽出する。そのために、複数の伝達関数をユーザ定義の

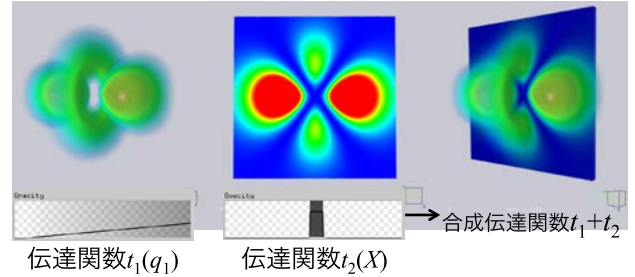


図 3 伝達関数の合成

代数式によって自在に組み合わせることによって多変量データに対する多次元伝達関数を設計することが可能になっている (図 3)。

粒子計算モジュール

シミュレーションに結合される粒子計算モジュールは、C++で記述され、Cによるラッパーによって Fortran プログラムにも適用可能である。粒子計算モジュールはライブラリとしてまとめられ、シミュレーションの各タイムステップにおける MPI プロセスから関数を一つ呼び出すだけで結合が可能のように設計されている。

粒子計算モジュールには、シミュレーションの領域分割に対応した MPI 並列化、そして各部分領域に対しては OpenMP による要素並列の粒子生成が実装されている。各要素では、不透明度関数を参照して要素中心のスカラー値を粒子密度に変換し、生成粒子数を計算する。そして、要素内部にモンテカルロ法で粒子分布を生成する。本計算では棄却法を採用しており、乱数で生成した粒子位置に対して規格化した粒子密度を計算し、粒子の採用/不採用を決定している。この粒子生成の計算は要素独立に実行することが可能であるが、生成粒子数は要素毎に異なるため、動的スケジューリングによる負荷分散を行っている。粒子生成後、各 MPI プロセスは配下のスレッドの粒子データとヒストグラムデータを集約する。そして、各タイムステップで生成した粒子データや計算結果のヒストグラム、そしてタイムステップ情報をファイル

としてストレージ上に出力する。これらのデータは各プロセスが独立に出力することができるため、コストの高い同期や集団通信を避けることが可能である。

しかし、粒子データファイルの出力がプロセス毎に実行される場合、高並列環境下ではファイル数が増加しボトルネックとなりうる。そのため、同一ノード内のプロセスで粒子データを集約する機能を実装した。この機能は MPI のコミュニケータ分割によりノード毎に独立に実装され、少ない通信コストで出力ファイル数をプロセス数からノード数に減らすことができる。

可視化用デーモン

可視化用データは対話ノード上で、あるいは、対話ジョブとして起動される。そしてストレージ上の粒子ファイル、ヒストグラムファイル、タイムステップファイルを収集し、In-Situ 可視化クライアントに送信する役割がある。図 1 に可視化用デーモンを中心とする In-Situ PBVR の動作メカニズムを示す。

1. 可視化用デーモンは、In-Situ 可視化クライアントから可視化パラメータ（伝達関数合成器で生成した伝達関数や画面解像度）を受信する。
2. 可視化用デーモンはストレージ上の可視化パラメータファイルを更新する。
3. 粒子計算モジュールはストレージ上の可視化パラメータファイルの更新の有無をチェックし、更新があった場合、可視化パラメータファイルの情報を各プロセスにブロードキャストする。
4. 粒子計算モジュールは、粒子データファイル、ヒストグラムファイル、タイムステップファイルを出力する。
5. 可視化用デーモンがタイムステップファイルの更新を検出する。
6. 可視化用デーモンは、全プロセス分の粒子データファイルが全て揃っているかど

うかを調べる。

7. 粒子データを集約し、クライアントへ転送、クライアント上でレンダリングする。

上記手順において In-Situ 可視化クライアントとの可視化パラメータ、および、粒子データのやり取りはストレージ上のファイル検出によってトリガーされており（ファイルベース制御機構）、可視化用デーモンと粒子計算モジュールは完全に非同期に実行することが可能になっている。

実験内容

In-Situ PBVR を燃料溶融シミュレーションコード JUPITER に結合し、FX100 の約 3000 コア、および OFP の約 10 万コアまでを使用したストロングスケールリング試験を実施した。

JUPITER は原子炉圧力容器内部の燃料溶融物の移行挙動を解析するために開発され、構造格子系における連続の式、Navier-Stokes 方程式、エネルギー方程式、及び界面捕獲方程式で記述される 3 次元多相多成分熱流動シミュレーションである。JUPITER は XeonPhi アーキテクチャ上で動作するよう移植されており、MPI/OpenMP によるハイブリッド並列化モデルを用いて 3 次元領域分割が実装されている。

典型的なシミュレーション時間は約 300 万タイムステップであり、本来は 1000 ステップ毎に可視化されていたが、本実験では毎タイムステップに In-Situ PBVR による可視化を行なった。問題規模は 240x240x1920（約 1 億格子）に固定し、可視化用粒子は約 1000 万粒子（約 250MB）、画面解像度は 1024x1024 を使用した。

OFP での実験結果

本実験では、1 ノードあたり 64 コアを使用し、プロセスあたりのスレッド数を 16、ノードあたりの MPI プロセス数を 4 に固定した。

MCDRAM をキャッシュモードで使用し、I/O システムはラウンドロビン分散の並列ファイルシステムを使用した。

ストロングスケールリング試験の結果を図 4 に、コスト分布を表 1 に示す。コスト分布の値として、全プロセスの最大値を使用している。表 1 において“JUPITER”と“PBVR”がそれぞれ、JUPITER と粒子生成モジュールの 1 タイムステップの処理時間である。また “PBVR” の処理のうち、粒子データの計算時間が “Sampling” に、粒子ファイル出力の処理時間が “Write” に示されている。この結果から、粒子生成モジュールの処理時間は約 10 万コアまでストロングスケールリングしていることが確認でき、更に JUPITER の 8%-28% 以内に抑えられていることが解る。粒子ファイルの書き出しコストは、ノード内ファイル集約の機能およびファイルベース制御機構による非同期出力の効果により、10 万コアまでスケールしていることが確認できる。一方、24 ノードから 1536 ノードの間の理論性能の加速率は 64 であるのに対し、“JUPITER” の加速率は 3.9 にとどまっている。“PBVR” の加速率は 15.4 であり、“JUPITER” よりも高いが、理論性能には届かないのが現状である。

この原因は、モンテカルロ計算で必要とされる補間計算や数式処理のベクトル化がなされておらず、KNL の SIMD 命令を使用できていないことにあると考えられる。

表 1 In-Situ PBVR のコスト分布 (OFF)

ノード数	24	96	384	1,536
コア数	1,536	6,144	24,576	98,304
JUPITER [sec/step]	28.0	9.9	6.0	5.9
PBVR [sec/step]	7.7	2.7	1.0	0.5
Sampling [sec_step]	7.5	2.6	0.9	0.4
Write [sec/step]	1.8	1.1	0.6	0.3

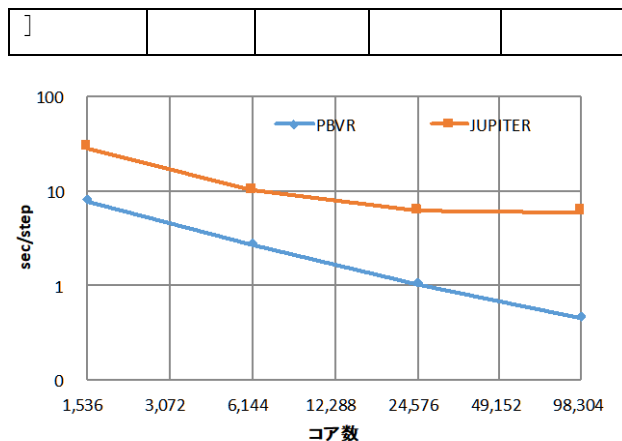


図 4 In-Situ PBVR のスケーラビリティ (OFF)

表 2 の “JUPITER” と “PBVR” はそれぞれ、JUPITER と粒子生成モジュールの消費メモリ量を示している。固定した問題規模にも関わらず、袖領域と MPI バッファによって JUPITER のメモリ消費量はコア数に従って増加している。他方、PBVR のメモリ消費量は JUPITER と比べて微増であり、各 MPI プロセスのメモリ消費量は約 3.3MB-92MB に抑えられている。

表 2 メモリ消費量

コア数	1,536	6,144	24,576	98,304
JUPITER[GB]	106.7	135.0	256.7	773.2
PBVR[GB]	8.9	9.5	11.6	20.4

図 5 に可視化用デーモンで粒子ファイルを収集し、クライアント PC 上で対話的な可視化を行った画像を示す。In-Situ 可視化クライアント上では約 10 FPS で対話的な画像の視点変更を実現している。

OFF 上で開発した In-Situ PBVR により、バッチ処理実行時の対話的な In-Situ 可視化が実現できた。本成果を 7 節に示す国際会議発表 1 件、および、学術論文 1 編として発表した。

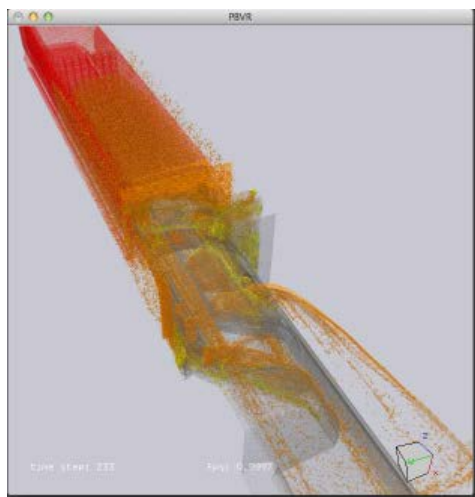


図 5 JUPITER の可視化結果画像

FX100 での実験結果

名大 FX100 を約 3000 コアまで使用してストロングスケーリング試験を実施した。本実験において、1 ノードあたり 32 コアを使用し、プロセスあたりのスレッド数を 4、ノードあたりのプロセス数を 8 に固定した。ノード配置はメッシュモードを利用し、I/O にはデフォルトの共有ファイルシステム (FEFS) を利用した。ストロングスケーリング試験の結果を図 6 に、コスト分布を表 3 に示す。

この実験において、“PBVR” の性能はスケールリングしたが、“JUPITER” の処理時間を上回ってしまった (図 6)。原因としては、粒子データの計算に使用する関数の最適化が不十分であることが考えられる。表 3 のコスト分布は、処理時間の殆どが粒子データの計算時間で占められていることを示している。粒子データの計算にはノード間通信が発生しないため、モンテカルロ計算で利用している並列乱数生成や補間計算、そして数式処理の最適化が不十分だと考えられる。

書き出しコストに関して表 1 と表 3 を比較すると FX100 の結果の方が高速に処理でき、同ノード数 (同じファイル出力数) となる 24 ノード及び 96 ノードでは 5 倍程度高速な結果となった。

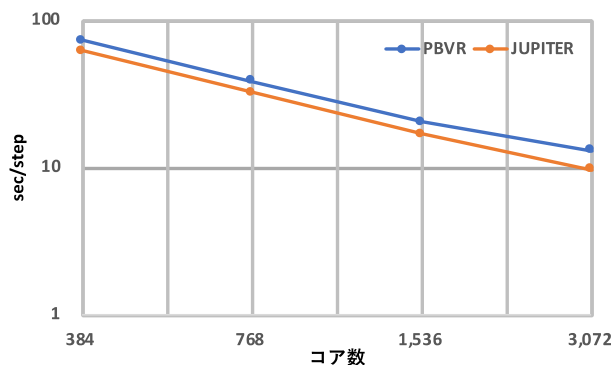


図 6 In-Situ PBVR のスケーラビリティ (FX100)

表 3 In-Situ PBVR のコスト分布 (FX100)

ノード数	12	24	48	96
コア数	384	768	1,536	3,072
JUPITER [sec/step]	62.7	32.4	17.1	9.8
PBVR [sec/step]	73.6	39.0	20.6	13.1
Sampling [sec/step]	73.4	38.8	20.5	13.0
Write [sec/step]	0.19	0.12	0.09	0.24

6. 今年度の進捗状況と今後の展望

本年度は In-Situ PBVR を JUPITER に結合して OFP、および、FX100 に移植し、動作を確認した。OFP 上では、省メモリと高いスケーラビリティを達成したため、シミュレーションを圧迫しない In-Situ 可視化システムを構築することができたと言える。

しかし、In-Situ PBVR を CityLBM コードに結合し、TSUBAME 上に移植する作業に関しては未完了で終わってしまった。原因としては、コード間の言語のバージョンの違いと、階層化された格子への対応に、当初の想定を超えた作業が発生してしまったことが挙げられる。これまで In-Situ PBVR は C++98 ベースで開発してきたが CityLBM は C++11 ベースで開発されており、In-Situ PBVR が利用している可視化ライブラリや粒子生成コード、数式

処理コードにおけるバージョンの違いを吸収する作業に時間を取られた。また、In-Situ PBVR は JUPITER 向けに 3 次元の構造格子に対応して開発を進めていたが、CityLBM は階層型格子を利用していた。そのため、In-Situ PBVR に入力される計算結果データのフォーマットを 4 次元に拡張する必要があった。更にそれに伴い、階層格子向けの補間計算機能や、それに対応した新たな粒子生成アルゴリズムを構築する必要があった。これらの理由により、期間内で作業を完遂することが困難になった。

今後の展望として、CityLBM コードへの結合を完了すること、そして、SIMD 化による性能の向上を目指している。ミュレーションソルバは今後 KNL 向けに最適化され、更に高速になっていくと考えられる。しかし、OFP での計測により、KNL の性能を發揮しきれていないことが判明した。そのため今後は、粒子生成処理を SIMD 化可能なようにアルゴリズムを変更して KNL の性能を十分に引き出し、最新鋭のシミュレーションに対しても十分に高速に動作する対話的 In-Situ 可視化を実現したい。

7. 研究成果リスト

(1) 学術論文

Takuma Kawamura, Tomoyuki Noda, Yasuhiro Idomura, “Performance Evaluation of Runtime Data Exploration Framework based on In-Situ Particle Based Volume Rendering”, Journal of Supercomputing Frontiers and Innovations, vol. 4, No. 3, pp. 43-54, 2017.

(2) 国際会議プロシーディングス

(3) 国際会議発表

Takuma Kawamura, Tomoyuki Noda, Yasuhiro Idomura, “Cross-platform In-Situ

Visualization System for Runtime Data Exploration based on PBVR”, WOIV held on ISC’ 17, Frankfurt, 2017.

(4) 国内会議発表

招待講演：河村拓馬, “PBVR による大規模シミュレーションの対話的遠隔可視化”, 第 7 回計算力学シンポジウム, 日本学術会議, 12 月 7 日, 2017.

(5) その他 (特許, プレス発表, 著書等)

参考文献

- [1] Susumu Yamashita, Takuya Ina, Yasuhiro Idomura, Hiroyuki Yoshida, “A numerical simulation method for molten material behavior in nuclear reactors”, Nuclear Engineering and Design, vol. 322, pp. 301-312, 2017.
- [2] Naoyuki Onodera, Yasuhiro Idomura, “Acceleration of Wind Simulation using Locally Mesh-Refined Lattice Boltzmann Method on GPU-Rich supercomputers”, Supercomputing Frontiers, pp. 128-145. Springer International Publishing, Cham, 2018.
- [3] Takuma Kawamura, Naohisa Sakamoto, Koji Koyamada, “A High Quality Sampling Technique for Particle-based Volume Rendering”, IEEE Visualization(Poster), 2009.
- [4] Takuma Kwamura, Yasuhiro Idomura, Hiroko Miyamura, Hiroshi Takemiya, “Algebraic design of multi-dimensional transfer function using transfer function synthesizer”, Journal of Visualization, vol. 20, No. 1, pp. 151-162, 2016.