粒子ベースボリュームレンダリングを用いた大規模複雑流体データの遠隔可視化

河村　拓馬
日本原子力研究開発機構
システム計算科学センター

粒子ベースボリュームレンダリングを用いた大規模複雑流体データの遠隔可視化

河村　拓馬、井戸村　泰宏、宮村（中村）　浩子、武宮　博

日本原子力研究開発機構・システム計算科学センター
〒277-0871 千葉県柏市若柴 178-4 柏の葉キャンパス 148 街区 4
東京大学柏の葉キャンパス駅前サテライト 4 階・5 階
{kawamura.takuma, idomura.yasuhiro, hiroko.miyamura, hiroshi.takemiya}@jaea.go.jp

遠隔地にあるスーパーコンピュータ上のシミュレーション結果を可視化するために, ユーザ PC にシミュレーション結果データ（ボリュームデータ）を転送して，商用ソフトを利用してポスト処理を行う方法がいまだに広く用いられている．しかしこの手法は，1 ケース当たり数十から数百テラバイトを超えることもある大規模データを可視化するのに，データ転送速度およびクライアント側のメモリ・ストレージ容量の観点で処理性能の限界を超えるという問題がある．そのため，サーバ上で可視化処理を行い，生成した可視化要素をクライアントに転送すること対話的な可視化を行うクライアント／サーバ型可視化が重要な技術になっている．

しかし，多くの商用ソフトで採用されているサーフェスレンダリングはデータの大規模化に比例してポリゴン数が増大するため，サーバからのデータ転送量が問題になる．またサーバ上で描画処理まで行う並列ボリュームレンダリングでは，各ノードで生成した部分画像を合成する画像重畳処理がボトルネックとなってストロングスケーリングの達成が困難である。また視点移動に伴い可視化処理の再計算が発生するため対話的な視点移動という問題がある．

本研究では粒子ベースボリュームレンダリング（Particle-Based Volume Rendering,　PBVR）を利用した並列分散環境向けの遠隔可視化システムを開発している．このシステムは大規模なボリュームデータに対してもデータ転送量が少ない，効率的なクライアント／サーバ可視化システムである．PBVR の重要な特徴として，ポリゴンベースの可視化手法と異なり，可視化用のデータ（粒子データ）のサイズが画像解像度で決定される事が挙げられ，それは元の大規模データのサイズと比べて十分に小さくなる．このシステムは PBVR の粒子生成処理をサーバで，レンダリング処理をクライアントで実行するように処理ステージを分解し，粒子データをソケット通信で転送することでクライアントとサーバ間を接続する．粒子生成処理は要素並列に実行可能なモンテカルロ法による計算で、高い並列化効率が期待される。またレンダリング処理においてサイズの小さい粒子データを利用することで従来手法と比べて十分に高速であり、対話的可視化に必要とされる処理速度が達成可能である．そして画像詳細度制御によりネットワークの帯域に則したサイズの粒子データの転送を行い、対話的なデータ探索が可能である。

サーバ上で行う粒子生成処理はモンテカルロ計算で行われるため計算負荷が高く，従来的な逐次処理では処理速度が不足する．そのためスーパーコンピュータや GPGPU 等の最先端並列環境における並列処理手法を提案し高速化を行った．GPGPU 上では粒子データのインデックス配列を生成することで GPU メモリ上の粒子配列の一括確保と CPU メモリへの転送の隠蔽を可能にし，448GPU コアを用いて 12 倍以上の高速化を実現した．スーパーコンピュータ上では粒子生成処理を負荷分散させ，クライアントサーバモデルによる可視化システムを構築した．このシステムは OpenMP と MPI によるハイブリッド並列で実装され，フィルタ処理により領域分割されたデータをマスタースレーブ方式で負荷分散をする．この手法を約 1 億格子の構造解析結果データや核融合プラズマ乱流シミュレーション結果に適用し，高い並列化効率が実現されることを確認した．本システムを商用のクライアント／サーバ型可視化アプリケーションと比較した結果、処理コストを大幅に削減することができ、30 倍の高速化を実現した。

粒子ベースボリュームレンダリングを用いた大規模複雑流体データの遠隔可視化

河村拓馬, 井戸村泰宏, 宮村（中村）浩子, 武宮博

日本原子力研究開発機構

第２６回CCSEワークショップ

発表の流れ

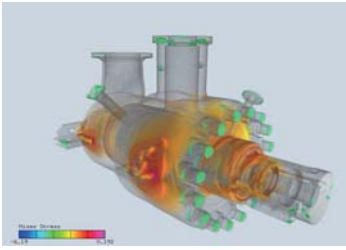
1. 大規模データにおける可視化の課題
2. 粒子ベースボリュームレンダリングによる クライアント・サーバモデル
3. スーパーコンピュータ上での超並列処理
4. グラフィックカードを利用したレンダリング
5. 画像詳細度制御を利用した対話的探索
6. 実験と考察
7. GPGPUクラスタでの実装と最適化
8. 多変量解析の可視化
9. 結言

大規模データの遠隔可視化

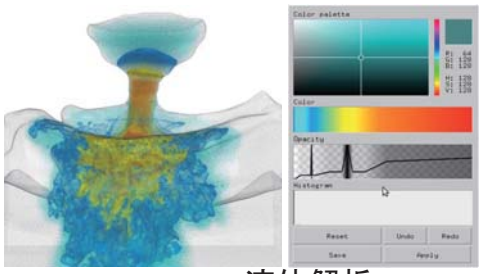
ボリュームレンダリング

- 様々な分野での利用

医療



構造解析



流体解析

並列ボリュームレンダリング (Ray-casting, Splatting, etc...)[1,2]

- 並列環境上の大規模可視化手法
- ストロングスケーリングが困難(画像重畳フェーズ)
- 固定的な視点位置

クライアント／サーバ型ボリュームレンダリング (eg. EnSight)

- 自由な視点移動
- 可視化要素(ポリゴン, カーネル)の爆発的増加

粒子ベースボリュームレンダリングによるクライアント／サーバシステム

1. Mark, et al. “Hybrid parallelism for volume rendering on large, multi-core systems”,2012
2. Peterka, et al. “Parallel Volume Rendering on the IBM Blue Gene/P”,2008

PBVRによる遠隔可視化システム

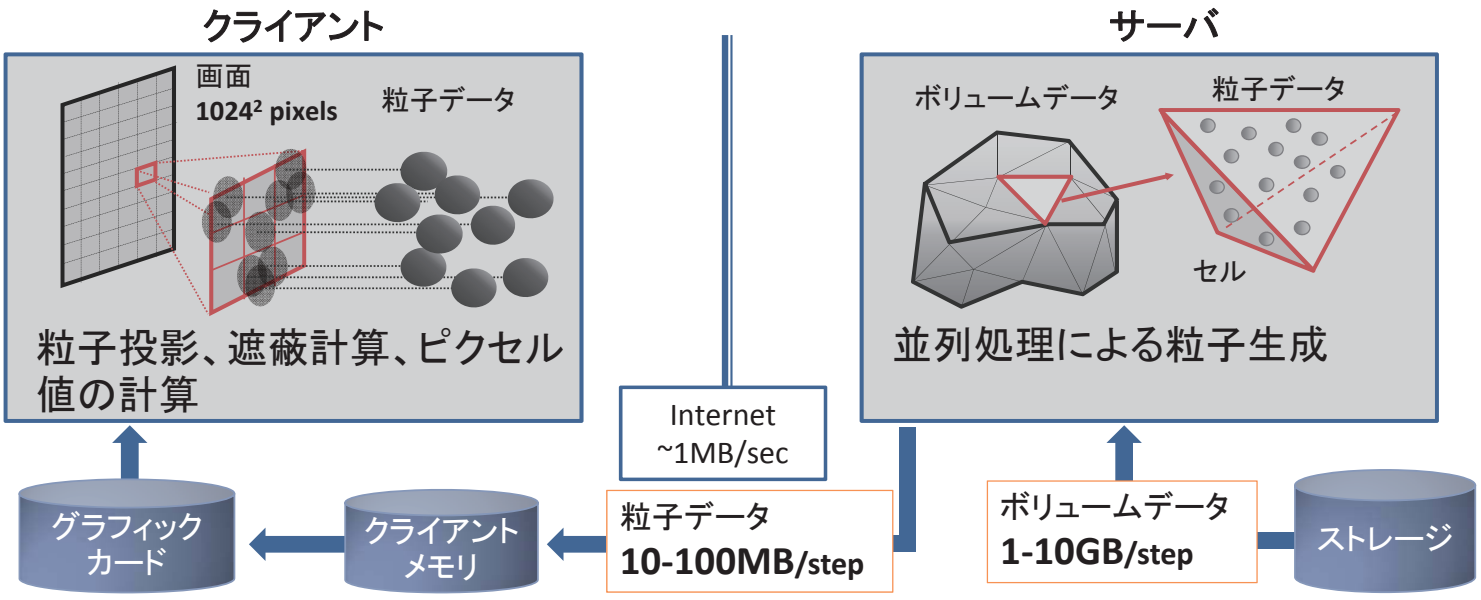
PBVR　不透明で自己発光する微小な粒子

– 不透明度は粒子による光の遮蔽率 (Sakamoto, et.al. 2010)

1. 元データと比較して小さい粒子データ
2. ソート不要のアルゴリズム、モンテカルロ計算
3. グラフィックカードの利用
4. 粒子数に依存する画質

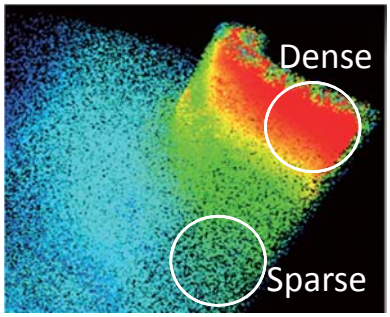
⇒

1. クライアント／サーバ
2. 高並列化(GPGPU、スパコン)
3. 対話的操作
4. 柔軟なLoD制御



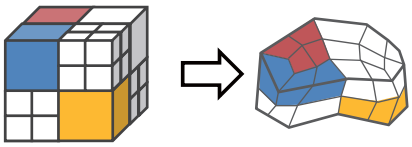
マルチノード環境における並列化

MPI-OpenMPによるハイブリッドプログラミングモデルを利用

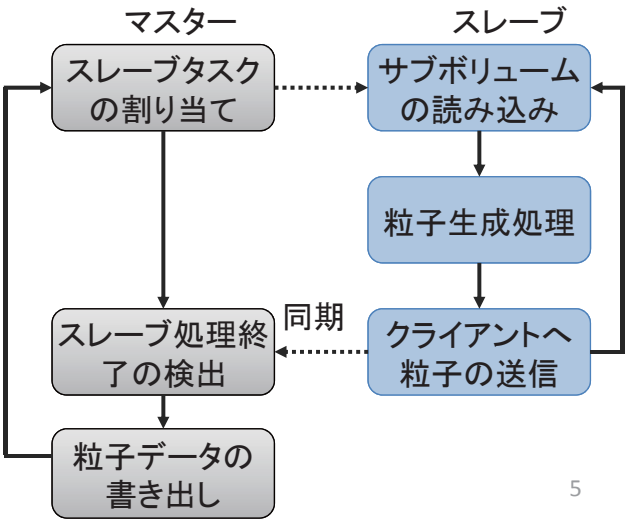


不均一な粒子分布によるロードバランスの低下

八分木による空間分割



マスタースレーブMPIモデル



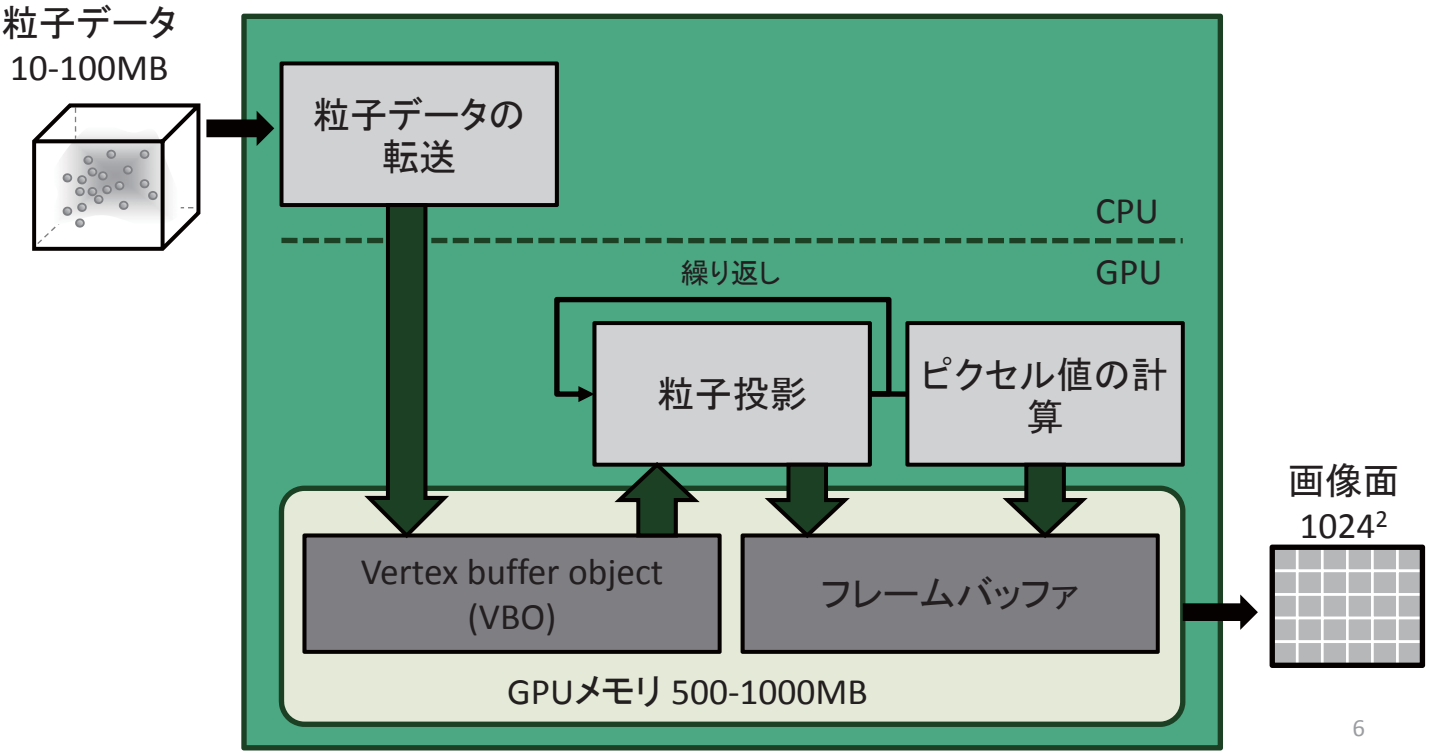
動的負荷分散

- 八分木を利用した時空間データの分割(フィルター処理)
- マスタースレーブモデルを利用した各ノードに対するサブボリュームの割り当て

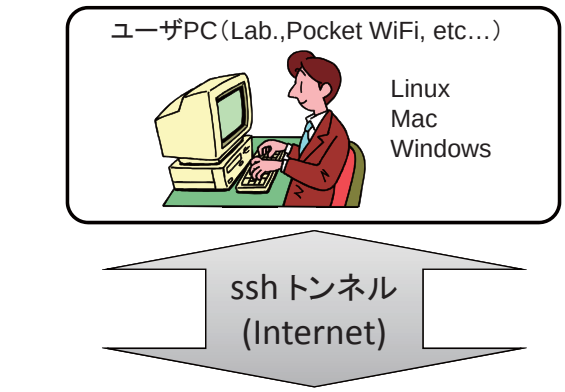
GPUによる粒子レンダリング

GPUを利用した高速な描画処理

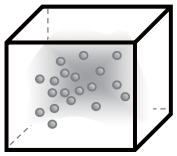
- プログラマブルシェーダを利用
- GPUメモリに収まるサイズの粒子データ



クライアント／サーバ通信とLoD制御



粒子データ10-100MB



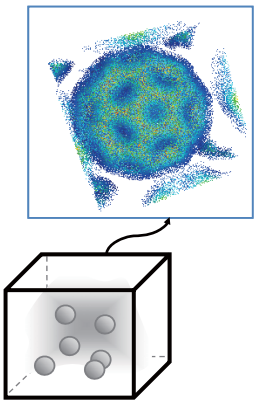
通信速度
1-10MB/sec

対話的データ探索

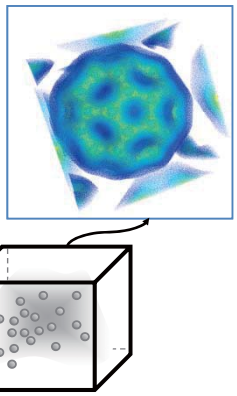
概要解析 ⇄ 詳細解析

興味領域の発見(ROI)
画像詳細度制御 (LOD)

粗い / 高速
（数MBの粒子）



精細 / 低速
(数百MBの粒子)



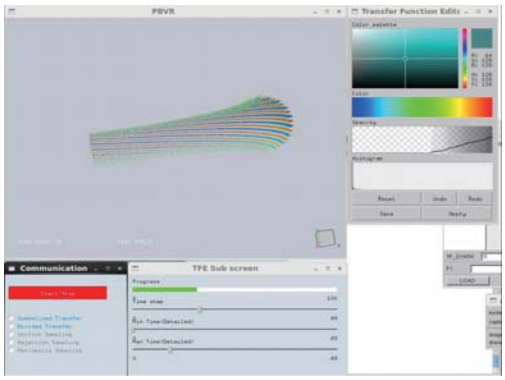
スーパーコンピュータ上の並列性能

サーバ (各ノード)	
K	SPARC64VIIIfx (8 コア), 128GFLOPS
	6D Torus interconnect
Bx900	2 x Xeon x5570 (4コア), 94GFLOPS
	Infiniband QDR(Fat tree)

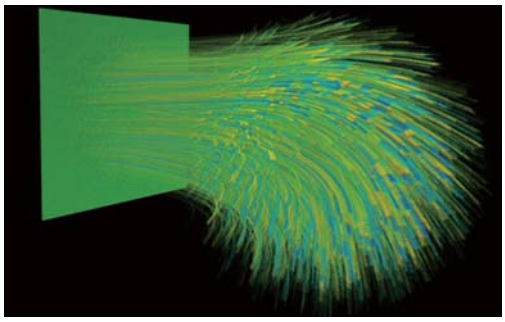
- ITERサイズの核融合プラズマシミュレーションにおける乱流変動の静電ポテンシャルを可視化
- 曲線座標で表現されたトーラス型データ

	セル数	データサイズ
GT5D	$286*10^6$	1.1 GB * 2000 step

画像解像度	粒子数
1024x1024	$\sim 10^7$



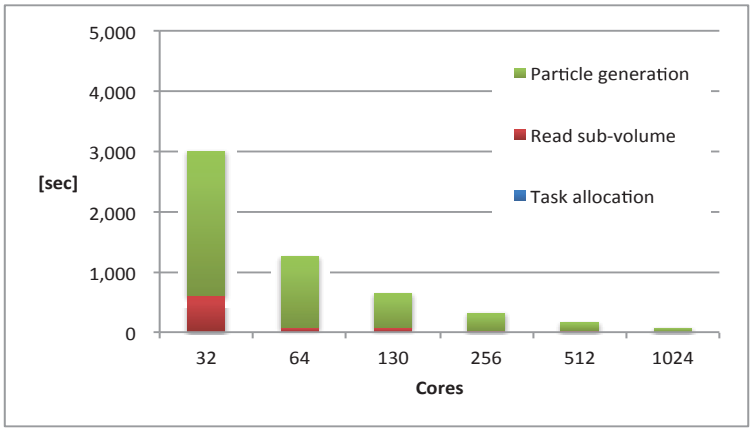
システムインターフェース



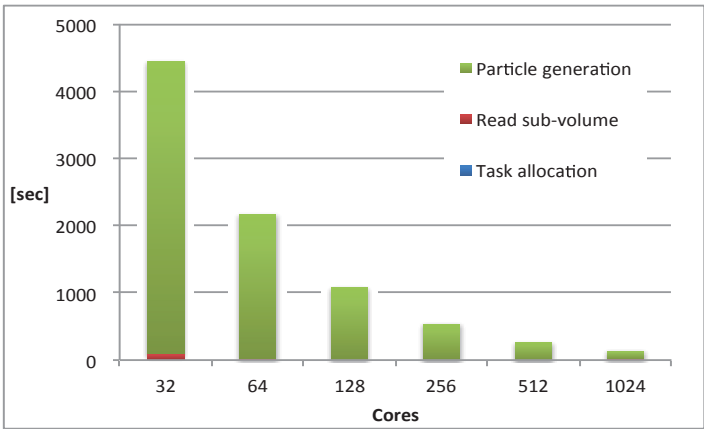
[データ提供: Idomura et al., Nuclear Fusion 49,065029 (2009)]

スーパーコンピュータ上の並列性能

K

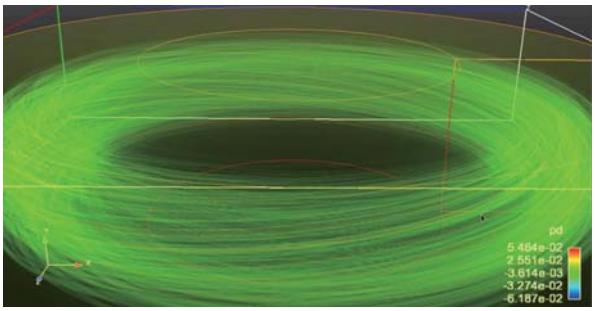


BX900

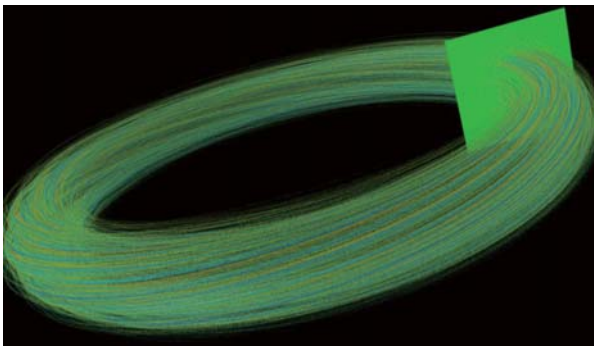


- 1024コアまで使用しストロングスケーリングを達成
  - 並列化効率90%以上
  - BX900よりも1.36倍高速
- Kコンピュータ1024コアで 0.84 [sec/step]

EnSightによるベンチマークテスト



EnSight



PBVR

クライアント／サーバ環境		
クライアント	CPU	Xeon E5
	GPU	Quadro K5000
サーバ	BX900 (48 コア)	
ネットワーク	3.4 [MB/sec]	

sec/step	PBVR	Ensight
フィルタ [sec/step]	0.7	-
粒子生成 [sec/step]	51.4	-
粒子転送 [sec/step]	75.7	-
総計 [sec/step]	127.8	3873
フレームレート [fps]	60.0	2.7
クライアントメモリ [MB]	257	900

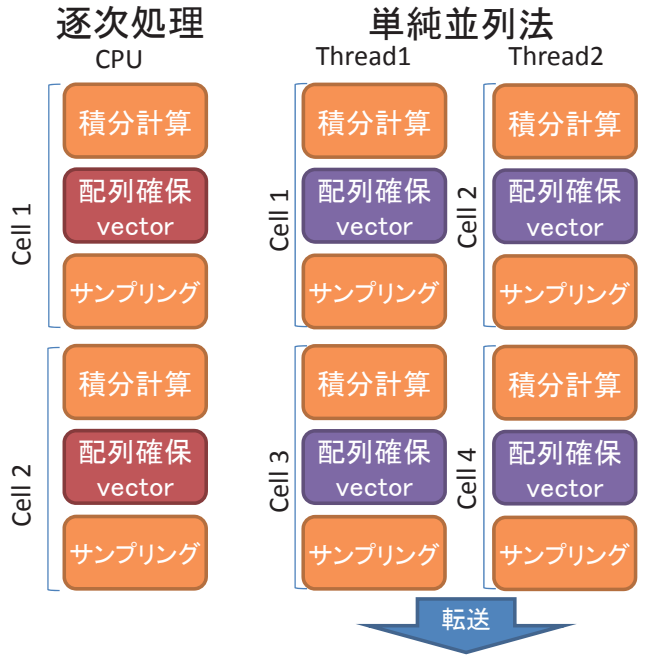
- 全体性能30倍以上の高速化
- クライアント使用メモリ1/3
- フレームレート22倍以上の高速化

先端的並列環境（GPGPU）の利用

粒子生成手順

1. 物理値から生成粒子数を積分計算
2. 粒子配列を確保(malloc/vector関数)
3. 粒子分布をモンテカルロサンプリング

高速化のために要素並列処理



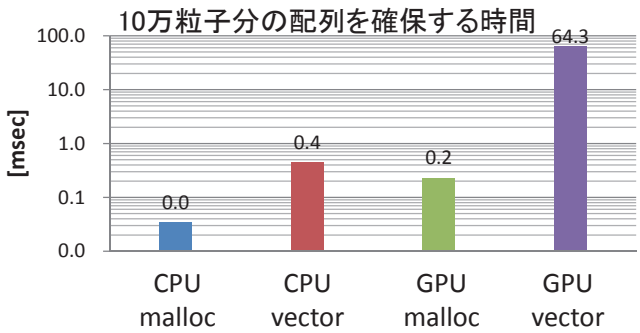
実験環境		
	CPU:1 core (Xeon E5607)	GPU:448 cores (Tesla M2075)
ピーク性能	11.72GFLOPS	1030 GFLOPS
メモリバンド幅	32 Gbyte/sec	150 Gbyte/sec

粒子生成時間[msec] （448コアで計算）

逐次	単純並列
66070.4	25000.3

448コアで約3倍の高速化に留まる

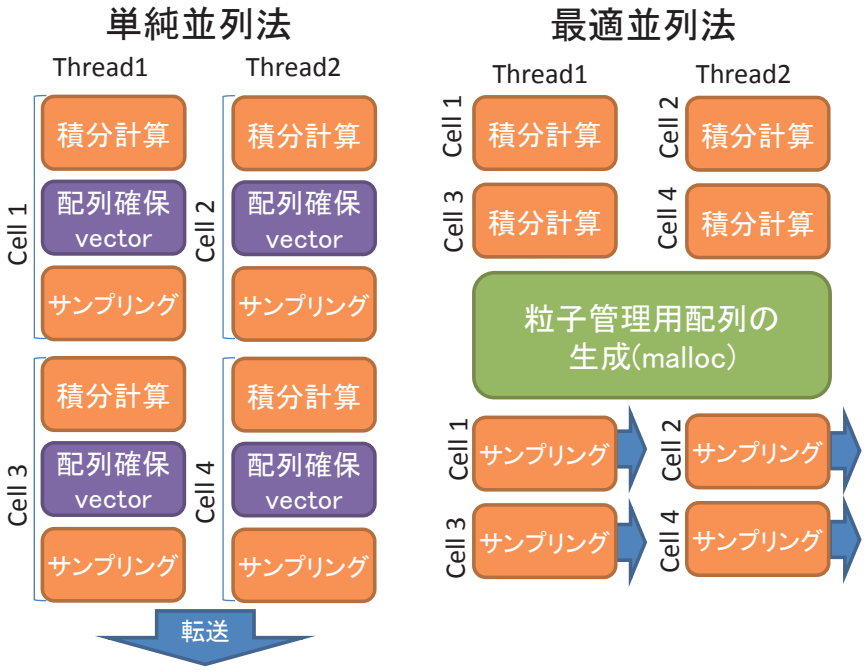
➡ GPU上での配列確保が原因



GPGPU上での粒子生成処理超並列最適化

粒子配列をmallocで確保 ⟹ 粒子管理用配列の生成

- 粒子書き込み位置を指定するためのインデックス配列
- 書き込み位置が判明するため、粒子転送の隠蔽も可能になる



粒子生成時間[msec] （448コアで計算）

逐次	単純並列	最適並列
66070.4	25000.3	3726.4

提案手法は逐次処理手法に比べて平均17倍、単純並列法と比較しても平均6.7倍の高速化を達成している

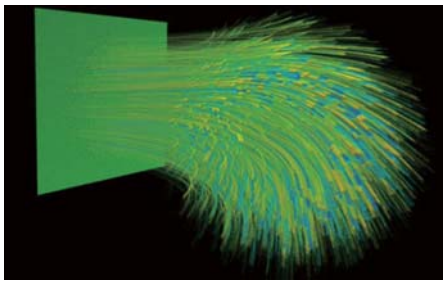
粒子管理用配列による改良並列法が最も高速

GPGPUによる並列処理性能

実験環境
CPU: Intel Xeon E5607 90GB RAM
GPU:Tesla M2075 6GB 6GB

	セル数	データサイズ
GT5D_small	$2.76*10^6$	11 MB * 3200 step

- 実験には1/6トーラスを使用



msec/step	逐次	単純並列	最適並列
I/O, CPU-GPU 転送	9	82	82
粒子管理配列	0	0	53
粒子生成	4010	1499	178
合計	4019	1581	313

- 単純並列:2.5倍
- 最適並列:12.8倍
- メモリ確保時間の削減
- 転送時間の隠蔽

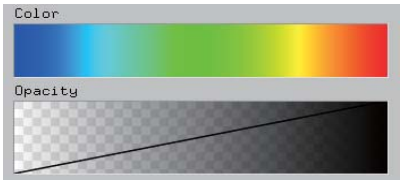
多変量データの可視化

- 同一空間に定義された複数の物理値
- 各物理値の分布を知りたいだけならば、ボリュームレンダリング画像を並べれば良い
- 各変量の空間的な関連性(相関)が知りたい
  - 非空間的な手法→散布図、パラレルコーディネート
- 各変量にどのように色を割り当てるか？独立に与えても、画面上で最終的に表示できる色は一色である

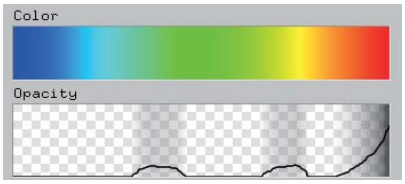
論理的な関連性を任意に入力

⇩

一次元伝達関数を合成して多次元伝達関数を得る



×
(且つ)
＋
(又は)



結言

- 並列ボリュームレンダリング: ストロングスケーリングが困難、固定的視点位置
- クライアント／サーバ可視化: 可視化要素の増大

粒子ベースボリュームレンダリングによるクライアント／サーバ型可視化

□ モンテカルロ法による粒子生成	⇒	■ ストロングスケーリングの並列処理
□ グラフィックカードによるレンダリング		■ 対話的な視点移動
□ 粒子数に依存する画質		■ 小さい粒子データの転送とLoD制御

より高度な可視化のために

先端的並列環境（GPGPU）の利用

□ 粒子管理配列 ⇒ ■ メモリ確保の高速化、転送の隠蔽

多変量データの可視化

□ 多次元伝達関数 ⇒ ■ 物理値の空間的な相関を抽出

- 提案システムのオープンソース化が進行中
- 時間変動を考慮した多変量、マルチスケール可視化技術の開発
- エクサスケール時代を見据えた更なる並列化効率