2008－HPC－117 (8)
2008／10／15

Cell/B.E. とDIMMnetを併用した大容量ボリュームレンダリングの予備評価

佐々木愛美 * 田邊昇 † 中條拓伯 ‡ 高田雅美 * 城和貴 *
manami41@ics.nara-wu.ac.jp
* 奈良女子大学 † (株) 東芝 ‡ 東京農工大学

概要

次世代スーパーコンピュータが出力する数値シミュレーション結果に代表される大容量データの可視化をできるシステムが求められている．我々は，Cell Broadband Engine(Cell/B.E.) やSpursEngine のような DMA で主記憶をアクセスする CPU と，大容量メモリ側の Gather 機能を有する DIMMnet を併用した大容量ボリュームレンダリングを行う装置の開発を目標に研究を進めている．本報告では，PLAYSTAION®3 上でのボリュームレンダリングプログラムによる上記方式の予備評価について述べる．

Preliminary Evaluations for Large Volume Rendering using Cell/B.E. and DIMMnet

Manami Sasaki * Noboru Tanabe † Hironori Nakajo‡ Masami Takata * Kazuki Joe *
* Nara Women's University † Toshiba Corporation
‡ Tokyo University of Agriculture and Technology

Abstract

The visualization system for large data such as output result from next generation supercomputer is needed. We are researching and developing about a visualization system for large volume rendering. One of the characteristics of the proposed system architecture is the combination between CPU such as Cell Broadband Engine (Cell/B.E.) and SpursEngine which accesses main memory with DMA and DIMMnet which is a large extended memory with memory side gathering functions. In this report, we describe the preliminary evaluations by the prototype of volume rendering program on PLAYSTAION®3.

1 はじめに

大規模な数値シミュレーション結果や衛星等各種センサーによる大量の観測データを分析するために，可視化は非常に有効な手法である．3次元可視化の一手法であるボリュームレンダリングは，これらの数値データをそのまま利用して可視化を行うことができるため，広く利用されている．しかしながら，ボリュームレンダリングは非常に膨大な計算量を必要とする．そのため，大規模なボリュームデータを扱う可視化では処理時間は大きな課題となっている．その要因として，膨大なデータ量とそれによる頻繁なメモリアクセスが挙げられる．

ボリュームデータが膨大な場合，その全てを1枚のプロセッサ基板のオンボードメモリ上に保持することは困難である．そのため，多数のプロセッサ基板を用意してデータ分割を行って並列処理するか，外部に大容量メモリを用意する必要がある．前者の場合は高コストになり，後者の場合は，ボリュームデータのほとんどはこの外部メモリに保持されることになる．加えて，ボリュームレンダリングにおける視線方向のメモリアクセスは，キャッシュフレンドリーではない．ボリュームデータの再利用性がないことから時間的局所性は非常に低い．また，視線の方向はボリュームデータ配列の並びに沿った方向ではない場合が多く空間的局所性も低い．従って，CPU から見て，計算に必要なボリュームデータは多くの場合，オンボードメモリ上ではなくホストインタフェース越しの外部メモリにある確率が非常に高くなり，外部大容量メモリへのアクセスが頻繁に行われることになる．

このような外部大容量メモリへの頻繁なアクセスによるオーバーヘッドの軽減を目的として，プリフェッチ機能を有するメモリモジュールである DIMMnet が開発されている．DIMMnet を用いることにより，不連続な位置に格納されているボリュームデータをあらかじめ DIMMnet 上のベクトルレジスタに収集し連続化することができる．これによって，外部大容量メモリへのアクセス回数を抑え，バースト長を増加させることが可能となり，実効バンド幅の向上が期待できる．

大容量ボリュームレンダリングを行うための高性能計算エンジンとしては，ハイエンド GPU などが挙げられる．しかしながら，我々は，SWoPP'08 での田邊らによる報告 [1] を受け，演算能力，バンド幅と電力の観点から，ハイエンド GPU よりも有利とされる SpursEngine[2] を採用する．SpursEngine や Cell/B.E.[3][4] はキャッシュの代わりにローカルストア (LS) と呼ばれる小容量のメモリを持ち，DMA 転送により主記憶から LS へデータの転送を行う．

本報告では，DMA で主記憶をアクセスする CPU と DIMMnet を併用した，CPU 側のオンボードメモリ容量を大幅に超える大容量ボリュームデータをレンダリングする装置の開発を目標とし，予備評価を

PLAYSTAION®3 上で行う．

2 章では，開発目標となる DMA で主記憶をアクセスする CPU と DIMMnet を併用したシステムについて述べる．3 章では，PLAYSTAION®3 を用いた評価実験について述べる．4 章では，関連研究について述べ，5 章でまとめと今後の課題について述べる．

2 開発目標のシステム

我々が開発目標としている，DMA で主記憶をアクセスする CPU と DIMMnet を併用した，CPU 側のオンボードメモリ容量を大幅に超える大容量ボリュームデータをレンダリングする装置の概要について述べる．2.1 節では，システムを開発する目的について述べる．

2.1 開発の目的

平成 18 年 7 月の共用法の施行により，特定先端大型研究施設の利用の機会が様々な研究者に広く与えられることが期待されている．そのため，ユーザはネットワーク越しの遠隔からの利用という形態をとることが主流になると考えられる．

数 GB から数十 GB に及ぶ大規模なボリュームデータを用いた可視化を行う場合でも，計算量の多さから大型計算機を利用することが必要となる．ユーザがデータ分析を行うために，膨大な計算量を必要とする可視化を遠隔から行う場合，インタラクティブな操作性は非常に重要であるが，その実現は極めて困難であるとされている．

そこで我々は，大規模なボリュームデータを実時間可視化し，その可視化結果を H.264 でライブストリーミング配信し，さらに，クライアントからインタラクティブに制御可能であるシステムを目標として研究を進めている．

2.2 提案方式

我々が目標とするシステムを実現させるためには，複数の高性能計算エンジンと大容量メモリからなる並列システムが必要となる．このようなシステムを実現するにあたって，次の点を考慮しなければならない．

- メモリ容量の確保
- メモリバンド幅の確保
- ノード間通信バンド幅の確保
- Zoom レスポンスの向上
- 処理能力と消費電力のバランス
- 遠隔ユーザへの転送フォーマット

これらの設計すべき事項について考察した結果得られた，SWoPP'08 にて田邊により報告されたシステムの提案方式 [1] について紹介する．

計算エンジンとしては，東芝の SpursEngine[2] の利用を提案している．その理由としては次の 2 つが挙げられる．1 つは，SpursEngine はハードウェアによる MPEG2 及び H.264 のエンコーディングをサポートしているため，低消費電力でリアルタイムの高解像度な動画エンコーディングが可能ということである．もう 1 つは，電力あたりの演算能力やメモリバンド幅・コスト的な問題に関して，ハイエンド GPU を利用した場合と比較して SpursEngine の方が有利であるという結論を得たことである．

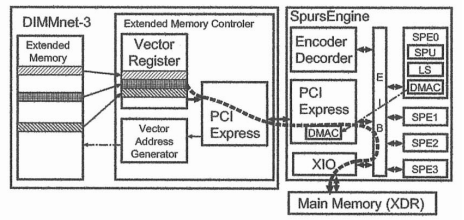

図 1: DIMMnet-3 を介した外部メモリアクセスの手順

次に，メモリ容量の強化・実効メモリバンド幅の向上のために，DIMMnet-3 を用いる．DIMMnet-3 を用いることで，メモリ容量を 1 個あたり最大 28GB 拡張することが可能である．また，DIMMnet-3 の持つベクトル型の不連続アクセスの連続化機能により，DMA を用いたメモリアクセス効率が向上する．DIMMnet-3 と SpursEngine を併用した場合，不連続なアドレスに保持されているデータは DIMMnet-3 の gather 機能を使って収集し，連続化される．その連続化したデータを DMA によって SpursEngine 側に取り込むことでメモリアクセスの効率化をはかることができる．[5] その手順を図 1 に示す．

Zoom 操作におけるレスポンス性の向上のためには，メモリサイドでの解像度調整を行う．これにより，計算エンジンと大容量メモリ間のバンド幅の消費を節約することができる．

ホスト～SpursEngine 間，ホスト～DIMMnet-3 間，SpursEngine～SpursEngine 間それぞれの結合網には，PCI express スイッチ [7] の活用を採用している．PCI express スイッチの活用により，スロット数の制約を回避し，ホスト PC に対する接続可能な SpursEngine の個数を確保可能にする．また，ホスト PC の CPU タイムや主記憶バンド幅を消費せずに低遅延かつスイッチポート間で並行して通信が実行されうるため，低コストで高バンド幅な通信を行うことが期待される．

2.3 システムの全体構成

以上の提案方式を採用した可視化システムの全体構成例を図 2 に示す．

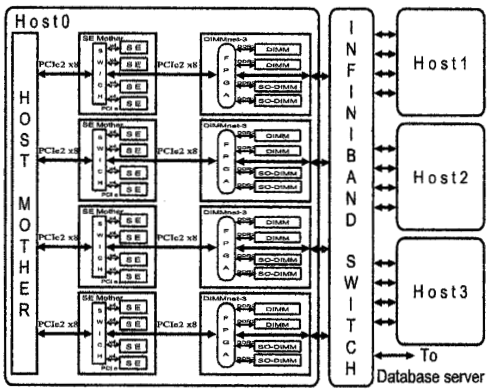

図 2: 可視化システムの全体構成

原案では各ホスト PC(パーソナルコンピュータ) の PCI exprsss スロットに装着される SE マザーボードと，PCI Express インタフェースに改造した DIMMnet-3 を組みとしたものを 4 組，ホスト PC に装着する．各 SE マザーボードには SpursEngine ボードが 4 枚ずつ装着されるので，ホスト PC あたり 16 枚の SpursEngine ボードが実装される．そのホスト PC を原案では 4 台作成し，その間を DIMMnet-3 が有する Infiniband[6] インタフェースを介して Infiniband スイッチによって接続することで全体システムを構成することとなっている．ただし，これらの物量やバンド幅に関しては十分な裏づけのもとに決められたものではないため，アプリケーションに基づく定量的な評価検討が望ましい．

3 評価実験

SWoPP'08 における田邊の論文 [1] の段階では，SpursEngine や DIMMnet-3 の使用台数の案は一応提示されているものの，実際にそれらを何台ずつ用いた時にどれくらいの時間でボリュームレンダリングが可能なのか不明である．さらに，SpursEngine から DIMMnet-3 をアクセスする際の遅延時間も不明である．また，SpursEngine の PCI express x4 インタフェースがバンド幅的にこの用途に足りるのかどうかも不明である．本実験ではそのような物量や遅延時間と性能の関係や，バンド幅に関する知見を得ることを目標に，簡素なボリュームレンダリングプログラムのプロトタイプを作成し，その上での予備性能評価実験を行った．

3.1 実験環境

実験環境を表 1 に示す．最終的なターゲットは前述のように SpursEngine を用いたシステムであるが，SpursEngine の環境は現段階では我々は入手できていない．一方，PLAYSTAION®3 に用いられているのは Cell/B.E. である．表 2 に Cell/B.E. は SpursEngine の仕様上の主な違いを示す．Cell/B.E. はレイアウトや周波数が異なるものの SpursEngine と機能的には同等の SPU を 4 個有する．このため，予備性能評価実験として実施される PLAYSTAION®3 上の測定は，SpursEngine ベースのシステム上の性能予測を行うための参考になると考えられる．

表 1: 実験環境

機器	PlayStation3(Cell B.E. 3.2GHz)
OS	Fedora 7
ソフトウェア開発環境	IBM Cell Broadband Engine SDK 3.0.0-1.0 libspe2-2.2.0 ppu-gcc-4.1.1 spu-gcc-4.1.1
時間計測	SPU Decrementer(精度 12.5ns)

3.2 実験内容

最終的なターゲットは，SpursEngine に直接接続される形で実装される外部メモリである XDR DRAM ベースのオンボードメモリ上に入りきらない数 GB から数十 GB クラスのボリュームデータに対するレンダリングを行うことにある．本実験はそのような状況を想定して，ボリュームデータは全て SpursEngine とは別に設置された DIMMnet-3 上に置かれているものとする．

表 2: Cell/B.E. と SpursEngine の仕様上の主な違い

	Cell/B.E.	SpursEngine
PPU	あり	なし
SPU	8 個	4 個
エンコーダ	なし	H.264, MPEG2
主記憶バンド幅	25.6GB/s	12.8GB/s
周波数	3.2GHz	1.5GHz
入出力 I/F	独自 (FlexIO)	PCI express x4
入出力バンド幅	76.8GB/s	1GB/s

しかし，実験環境である PLAYSTAION®3 には 256MB の XDR DRAM ベースのオンボードメモリしか実装されておらず，DIMMnet-3 を拡張する手段もない．現状の DIMMnet-3 には PCI express インタフェースもない．さらに，DIMMnet-3 と SpursEngine を接続する PCI express スイッチもない．また，それらの正確な遅延時間も未知であり，今後の最適化設計項目の一つである．そこで，SpursEngine の SPU から PCI express スイッチを介して DIMMnet をアクセスする環境を，Cell/B.E. の SPU 上の空の for ループの回数を変化させて人工的に生成され挿入される遅延時間に置き換えて模擬する．

ループ回数とその遅延時間の関係はあらかじめ SPU decrementer を用いた測定によって求めておき，挿入すべき遅延時間に相当するループ回数を指定して，適切な遅延時間を挿入する．この遅延時間を実機上で概ね予想される値が入る適当な幅で変化させた場合の，ボリュームレンダリングプログラムの実行時間の変動を観察する．

DIMMnet と SpersEngine の組み合わせというこれまでの実行環境とは全く異なる環境での実現を目標としており，従来環境上での最適化手法は必ずしも適合するとは限らない．また，その環境上でのアルゴリズムの最適化自体が今後の研究対象でもある．このため，測定に用いるボリュームレンダリングプログラムは現段階では極力シンプルなものをベースに用いる．

ボリュームレンダリングでは RayCasting 法 [9] が通常用いられる．RayCasting 法では，視点からスクリーンの各ピクセルにベクトルを伸ばし，そのベクトル上にボリュームデータのサンプリングポイントをとり，不透明度の累積計算を行う．本予備評価実験で使用しているプログラムは簡素化のためこれとは少し異なり，スクリーンに垂直なベクトルを各ピクセルから伸ばして計算している．つまり，厳密な RayCasting 法ではないが，不連続アクセスとなる不透明度の累積計算の部分に関しては RayCasting 法と同じである．よって今回の実験の範囲においては問題ないと考えている．

ベースとして使用しているボリュームレンダリングプログラムのフローは以下の通りである．

Step1
ボリュームデータを読み込み配列 alpha に格納
Step2
スクリーンの左上の座標
スクリーンを右方向に走査するベクトル
スクリーンを下方向に走査するベクトル
スクリーンに垂直なベクトル　などの設定
Step3
スクリーンのピクセルに垂直な Ray を設定
Step4
Ray を伸ばしてサンプリングポイントをとり，
ボリュームデータ内にあるかどうか判定
Step5
ボリュームデータ内であれば，alpha 値から輝度値を累積計算 (α ブレンディング) する
Step4 に戻る
Step6
Ray を伸ばし終わったら求めた輝度値を配列に保持
Step3 に戻る
Step7
計算結果を PPM 形式の画像として出力

上記のフローのプログラムをベースに Cell/B.E. 上に以下の三つの手法での移植を行った．

(1) alpha 値すべてを最初にローカルメモリ (LS) へ転送後，計算
(2) alpha 値を使用するときに使用する要素のデータだけを単純 DMA で LS へ転送し，計算
(3) あらかじめ使用する要素のアドレスを計算して配列に格納し，DMA リスト [8] 転送で LS へ転送後，計算

測定に使用したデータサイズは以下の通りである．

- ボリュームデータ：16*16*16
- スクリーンサイズ：64*64

上記のサイズは上記の三つのメモリアクセス方法の違いに伴う性能比較を行うことと，プログラムの簡単化のために小さいものとしている．このサイズ自体は PLAYSTAION ®3 のオンボードメモリはもちろん，(1) の実行が可能であるように 1 個の SPE に内蔵される 256KB の LS 上に入りきる小さいものである．このようなサイズであれば本来，(2)(3) のようなアクセスは不要である．しかし，オンボードメモリにも乗り切らないサイズに対する処理の挙動を模擬する評価なので，本実験ではあえて (2)(3) のようなアクセスも測定し，(1) との差異を確認する．

なお，本原稿執筆段階での評価プログラムの SIMD 化や並列化は作業途中であり，これらは行われていない段階の測定に関する報告である．

3.3 実験結果と考察

3.3.1 メモリアクセス手法による性能差

前述の三つのメモリアクセス方法 (1)〜(3) の α ブレンディングを Ray4096 本分行うループの実行時間を計測した．なお，(3) については DMA コマンドのリストを作成するだけで，そのメモリアクセス実行時間 (4) は入っていない．その結果を以下に示す．

(1) 全て LS をアクセスさせる場合：24.04[ms]
(2) 単純 DMA 方式：37.70[ms]
(3) DMA リスト (メモリアクセス以外)：23.53[ms]
(4) DMA リスト (DMA リスト作成)：5.00[ms]
(5) DMA リスト (メモリアクセス)：2.51[ms]

(1) と (2) では，LS へのアクセスと細かい DMA 転送での主記憶へのアクセスの違いなので，10〜100 倍の速度差を予想していたが，予想していたより差が見られなかった．この理由は，この段階のプログラム実装においてはまだ並列化や SIMD 化が行われていないため，処理時間におけるメモリアクセスへの負担がハードウェア能力の限界近くまでは達しておらず，メモリアクセス時間がボトルネックになっていないためであると考えられる．

DMA リストによるメモリアクセス時間 (5) は全体の 10.7%にすぎず，DIMMnet-3 によって拡張される大容量の外部メモリに対するアクセス遅延がこの数倍の範囲で増加したとしても，実行時間にあまり大きな影響が出ないことが予想される．

3.3.2 Ray1 本あたりの処理時間とその内訳

単純 DMA を用いる場合の Ray1 本あたりの処理時間は以下の通りである．

$37.70\text{ms} \div 4096 = 9.2\mu\text{s}$

一方，DMA リストを用いる場合の Ray1 本あたりの処理時間は以下の通りである．

$23.53\text{ms} \div 4096 = 5.4\mu\text{s}$

単純 DMA を用いる場合の 2 倍弱の性能向上に留まっている．この倍率は Wisconsin ベンチマークによる評価 [5] における 3.9 倍と比べると半分程度であり，SIMD 化も並列化もしていない現状のプログラムにおけるメモリアクセスの性能への寄与度は Wisconsin ベンチマークより低いことが判った．

DMA リスト転送を用いた場合は Ray あたりの平均メモリアクセス時間は，以下の通りである．

$(5.00\text{ms} + 2.51\text{ms}) \div 4096 = 1.83\mu\text{s}$

DMA リスト作成は，Ray によってリストの個数が 16 個の場合と 17 個の場合がある．これを考慮すると，DMA リスト作成 1 回の処理時間は，以下の通りである．この値はボリュームサイズ変更時の性能に影響を与える数値として将来重要となると考えられる．

$5.00\text{ms} \div (16 \times 2016\text{本} + 17 \times 2016\text{本}) = 75\text{ns}$

DMA リスト転送 1 回の処理時間と実効バンド幅は，以下の通りである．この値は主記憶を 16 個程度のリストからなる DMA リスト転送の遅延やバンド幅を与えるものとして重要である．

$2.51\text{ms} \div 4096 = 613\text{ns}$

$4\text{B} \times 16/0.613\mu\text{s} = 104\text{MB/s}$

上記の 104MB/s という値は Cell/B.E. の主記憶の 25.6GB/s というメモリバンド幅からはかけ離れた値を示しており，不連続アクセスに伴う効率が悪い状態でのメモリアクセスになっていることがその原因である．大容量を実現するために導入される DIMMnet-3 をアクセスする場合でも，この程度の実効バンド幅を実現できれば，性能低下を防止できる可能性があることを示唆している．

SpersEngine の主記憶バンド幅は Cell/B.E. の半分 (12.8GB/s) に設定されており，上記の 104MB/s という値は SpersEngine 上では得られず，性能低下が生じる可能性が高い．さらに，複数の SPU から主記憶をア

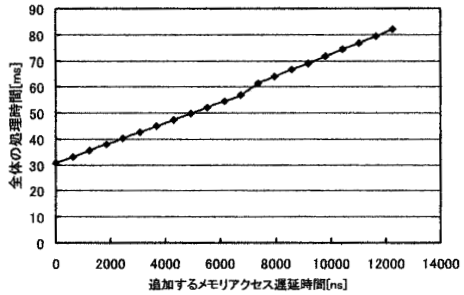

図 3: メモリアクセス遅延と実行時間の関係

クセスさせた場合は，さらにバンド幅低下が発生する可能性がある．これらの点は今後 SpursEngine 上で，プログラムの並列化をすることで定量的に評価にすべきであり，今後の課題である．

一方，DIMMnet-3 を SpersEngine からアクセスさせる場合，PCI express x4(片方向 1GB/s) の経路を通過するが，DIMMnet-3 によって連続化されたアクセスで上記の 104MB/s というバンド幅を 1GB/s の PCI express x4 上で実現することは困難ではない．SpersEngine 内部の 1 個の SPU にプリフェッチスレッドを動作させ，残りの 3 個の SPU で不透明度の累積計算を行わせることを想定すると，104MB/s × 3 程度のバンド幅の供給に，PCI express x4 自体がボトルネックにはならないと考えられる．

3.3.3 メモリアクセス遅延と実行時間の関係

ターゲットシステムにおいて，SpursEngine から DIMMnet-3 へのリモートベクトルリストロードの遅延時間は現段階では不明である．この遅延は，今後の設計項目の一つである．このコマンドを実行する場合の CPU 側の準備として必要な処理は上記の (3)DMA リスト (メモリアクセス以外) とほぼ同等である．この処理の後で行われる (4)DMA リスト (メモリアクセス) の実行時間が上記のターゲットシステムと PLAYSTAION®3 の間で大きくなる．そのメモリアクセス実行時間を (4) の 1 倍 (613ns) から 20 倍の範囲で変化させた場合の全実行時間の変化を図 3 に示す．この結果からわかるように，SIMD 化や並列化を行っていないボリュームレンダリングプログラムにおいてはメモリアクセス遅延が 10 倍に増加して，その隠蔽を行わずに実効バンド幅が 1/10 に減少したとしても，性能低下は高々2 倍程度であるに過ぎない．

なお，DIMMnet-3 の前身である DIMMnet-2 上でのベクトルリストロード時間は実機上での測定結果があり，8 バイトのデータ 64 語を DIMMnet-2 上のメモリからホスト CPU にロードし終わるまでにリストの内容によって 1.3〜6.5μs の遅延がかかる．これに対して DIMMnet-3 上では，動作周波数やバンク数はその 2 倍以上で，本アプリケーションでの転送語数が上記の測定の 1/4(16 語) である一方，本システム上に適用するために PCI express インタフェース (数百 ns) や PCI express スイッチ (数百 ns) を往復で通過する遅延時間が余計にかかることを考慮するとやや少ない遅延時間で済むことが予想される．つまり，概ね予想される遅延時間は図 3 の中央 (6μs) より左側の領域にあり，性能劣化は利用する SPU の個数を 2 倍に増やす程度でカバーできる範囲に留まっていることがわかる．

また、今後、SIMD 化や並列化を進めることによって，図 3 のグラフの切片の部分が短縮していくことになるが，SpursEngine 上でそのような最適化を進めた場合，SIMD 化で最大 4 倍，並列化で最大 3 倍 (SPU1 個はデータ転送に使用する場合) の処理速度となり，メモリアクセス遅延の増加の影響が顕著になることが予想される．ただし，今回の評価ではダブルバッファリングやソフトウェアパイプライニングの技法を用いた遅延隠蔽や無駄な計算を行わない最適化は行っていないので，更なる改善を行う余地は残されている．

4 関連研究

本研究は，田邊らによって SWoPP'08 において提案された可視化装置の構想 [1] に基づくものである．本研究で評価しているアーキテクチャ的要素である DMA によって主記憶をアクセスするタイプの CPU と DIMMnet のようなメモリサイドの Gather 機構を組み合わせることによる不連続アクセスを主体とするアプリケーションの加速については FIT'08 における田邊らの研究 [5] に基づいている．ただし，この研究は等間隔アクセスを主体とする Wisconsin ベンチマークに基づく評価であり，本研究が取り扱っているボリュームレンダリングが多用するリストアクセスとは異なる．ただし，Cell/B.E. や SpursEngine 上では等間隔アクセスもリストアクセスもどちらも DMA リスト [8] と呼ばれるリストアクセスをサポートする機構によって実現されるので，提案アーキテクチャで加速される前のソフトウェア実装の性質は共通するものが多い．ただし，ボリュームレンダリングと Wisconsin ベンチマークとでは不連続アクセス以外の部分の CPU 負荷などに大きな違いがある．このため，本研究による詳細な解析が必要であり，これらは従来研究では明らかになっていなかった．

一方，本年 8 月に Intel 社は Larrabee[10] という Visual computing をターゲットとした x86 ベースの many コアプロセッサのアーキテクチャを発表した．詳細な評価は報告されていないが本研究で取り扱っている Gather 機能も搭載されていることが報告されており，Visual computing 向けのアーキテクチャのトレンドを裏付けるものとして注目に値する．Gather 機能自体はこれまでにも DIMMnet だけでなく，DMA リスト機能のように Cell/B.E. や SpursEngine にも搭載されている．Larrabee の Gather 機能は命令で実現されるという点で DMA リストとは異なるが，DMA リストと同様にプロセッサ側に実装される機能である．これはキャッシュラインサイズとアクセス単位の不整合という性能低下要因から脱却するものでもない．よって，DIMMnet のようなメモリコントローラ側に実装される Gather 機能とは異なり，バースト長が 4 バイトといった本アプリケーションの利用形態においては DIMMnet と Larrabee の Gather 機能の間には大きな転送効率の差が現れると考えられる．

大容量のボリュームデータを扱えるようにする先行研究としては，Castanie[11] は数十から数百 GB class のボリュームデータを扱うことができるメモリシステムのアーキテクチャを発表している。可視化における read only な性質を前提に一貫性維持のオーバーヘッドを排除した簡素な分散共有メモリを構築する点は我々

の研究と共通である。しかし、ディスクアクセスを遠隔メモリアクセスに置き換えることが主眼になっており、brick と呼ばれる固定サイズを有する一種のページをベースにしたメモリ管理が行われる。これはハードウェアとして COTS ベースの PC クラスタを用いているとともに、DIMMnet-3 のような不連続アクセスの連続化による効率化は行わない。

また，Cell/B.E. におけるボリュームレンダリングの高速化の研究には IBM 社が開催したコンテストでの受賞作品である C-Ray[12] がある。C-Ray は 1 個の Cell/B.E. の主記憶に収まる範囲での最適化を施したプログラムであり、本研究のような主記憶容量を超える場合を想定していない．ただし，Empty space skipping と Early termination[13] という無駄な計算を省略する最適化が施してある点で本研究の簡素なプロトタイプより本格的なプログラムになっている．

5 まとめと今後の課題

DMA で主記憶をアクセスする CPU と DIMMnet を併用した可視化システムの提案を紹介し，PLAYSTAION®3 を用いたボリュームレンダリングプログラムの実装による提案アーキテクチャの評価実験について報告した．

その結果，ボリュームレンダリングは比較的計算部分が重く，先行研究 [5] で評価した Wisconsin ベンチマークよりもメモリアクセス性能による全体性能への寄与率が半分程度であることが判った．また，Cell/B.E. における DMA リストを用いた本アプリケーションに特徴的なメモリアクセスに対する実効バンド幅は 104MB/s に過ぎなかった．低周波な SpursEngine においてはこれより要求バンド幅が低くなることが必至である．よって，SpursEngine から PCI express x4 経由での DIMMnet-3 へのアクセス時に，PCI express x4 はバンド幅のボトルネックにはなりにくいことが判った．さらに，現状の SIMD 化も並列化も行っていない状態では，DIMMnet-3 を PCI express スイッチ経由でアクセスすることにより，メモリアクセス遅延が 10 倍に増加したとしても，レンダリング性能は 1/2 に落ちないレベルであることも判った．DIMMnet-2 の実機上での遅延時間の実績から，DIMMnet-3 の遅延も同様に，隠蔽しなくてもレンダリング性能が 1/2 に落ちない領域に入ることが予想される．

現状の予備評価プログラムは SIMD 化も並列化も行っていない．無駄な計算も多く，高速なメモリ階層を可能な限り多用するような最適化も行っておらず，処理性能は十分に最適化されていない．扱ったデータサイズも小さい．さらに SpursEngine 上での評価もできていない．それらのプログラム改良と評価や性能モデリングは今後の課題である．性能改善にあたっては C-Ray[12] などに取り入れられている技法を踏襲することで，実用的なボリュームサイズおよび実用的な処理時間での実行に近づけるられると考えられる．ただし，そのような従来の最適化技法の取り込みに当たっては，DIMMnet の有無や SpursEngine において PPU が無いこととの相性を検討した上で，取捨選択や独自の改良を加えるべきである．

謝辞

本研究の一部 (DIMMnet-3 の研究開発) は総務省戦略的情報通信研究開発推進制度 (SCOPE) の一環として行われたものである．

参考文献

[1] 田邊, 佐々木, 中條, 城 : “大容量データ向け対話的実時間遠隔可視化装置の実現性検討”, 電子情報通信学会コンピュータシステム研究会 (CPSY2008-18), pp.43-48 (Aug. 2008)

[2] 東芝 : “メディアストリーミングプロセッサ SpursEngine^{TM} SE1000 のサンプル出荷開始について”, http://www.toshiba.co.jp/about/press/2008_04/pr_j0801.htm

[3] 東芝セミコンダクター社 : “Cell Broadband Engine”, http://www.semicon.toshiba.co.jp/product/micro/cell/index.html

[4] Cell User's Group : “Cell 関連情報”, https://www.cellusersgroup.com/modules/product/

[5] 田邊, 太田, 金, 中條 : “DMA で主記憶をアクセスする CPU における不連続アクセスの連続化”, 第 7 回情報科学技術フォーラム FIT2008，第 1 分冊, pp.31-34, RC-005 (Sep. 2008)

[6] InfiniBand Trade Association, http://www.infinibandta.org/

[7] PLX Technology : “PCI Express 2.0 Switches - PCIe Express Lane I/O Interconnect”, http://www.plxtech.com/products/expresslane/gen2.asp

[8] M. Kistler, M. Perrone, and F. Petrini: ”Cell Multiprocessor Communication Network: Built for Speed” IEEE Micro, vol. 26(3) pp. 10-23, (May-June 2006)

[9] M. Levoy : “Display of surface from volume data”, IEEE Computer Graphics and Applications, Vol.8, No.3, pp.29-37 (May 1988)

[10] L. Seiler et al. : “Larrabee: A Many-Core x86 Architecture for Visual Computing”, ACM Trans. Graph. Vol.27, No.3, Article 18 (Aug. 2008)

[11] Laurent Castanie, Christophe Mion, Xavier Cavin and Bruno Levy : “Distributed Shared Memory for Roaming Large Volumes”, IEEE Trans. on Visualization and Computer Graphics, Vol.12, No.5, pp.1299-1306 (2006)

[12] J. Kim : “C-Ray: Interactive Volume Ray Casting Library for Cell B.E. Architecture”, http://www-304.ibm.com/jct09002c/university/students/contests/cell/r1w3proposal.pdf

[13] M. Levoy : “Efficient Ray tracing of volume data”, ACM Trans. Graphics, Vol.9, No.3, pp.245-261 (1990)