

平成 24 年度 HPCI システム利用研究課題 利用報告書

【概要報告】

課題番号	hp130034		
課題名	フォトリアリスティックレンダリングエンジンを使用した高速レンダリング環境の構築		
利用枠	<p>■一般利用枠</p> <p>■「京」</p> <p><input type="checkbox"/>一般利用 <input type="checkbox"/>若手人材育成利用 <input type="checkbox"/>一般利用(条件付選定)</p> <p>■産業利用(<input type="checkbox"/>実証利用 <input checked="" type="checkbox"/>トライアル・ユース)</p> <p><input type="checkbox"/>「京」を除くHPCIシステム利用</p> <p><input type="checkbox"/>北大 <input type="checkbox"/>東北大 <input type="checkbox"/>筑波大 <input type="checkbox"/>東大 <input type="checkbox"/>東工大 <input type="checkbox"/>名大 <input type="checkbox"/>京大 <input type="checkbox"/>阪大 <input type="checkbox"/>九大</p> <p><input type="checkbox"/>一般利用 <input type="checkbox"/>産業利用(<input type="checkbox"/>実証利用 <input type="checkbox"/>トライアル・ユース)</p> <p><input type="checkbox"/>HPCI戦略プログラム利用枠</p> <p><input type="checkbox"/>優先課題(<input type="checkbox"/>分野1 <input type="checkbox"/>分野2 <input type="checkbox"/>分野3 <input type="checkbox"/>分野4 <input type="checkbox"/>分野5)</p> <p><input type="checkbox"/>利用研究課題(<input type="checkbox"/>分野1 <input type="checkbox"/>分野2 <input type="checkbox"/>分野3 <input type="checkbox"/>分野4 <input type="checkbox"/>分野5)</p>		
課題代表者	氏名	スーリエ クリストフ	国名(組織の)
	組織・部門	株式会社 FORUM8	
利用目的	<p>アンバイアスなレンダリングエンジンである LuxRender を用いて多数のフレームを含む CG レンダリングが可能か検証します。</p> <p>LuxRender は非常に低速な CG レンダリングエンジンであり、1枚の静止画をレンダリングするにも相当な時間が必要です。そのエンジンを用いて数百フレームのアニメーションをレンダリングするには大規模なコンピュータ資源が必要です。</p> <p>本トライアルではオープンソースである LuxRender を京に移植し、多数のフレームを同時にレンダリング可能か検証します。</p>		
成果概要	京上でオープンソースである LuxRender の移植に成功し、多数のフレームの同時レンダリングに成功しました。		
実施期間	2013年5月17日～2013年11月16日		
キーワード	可視化,バーチャルリアリティ,CG,フォトリアリスティックレンダリング,アンバイアス		

【利用計算資源情報(システム情報)】

【ソフトウェア利用情報】

※上記については、別紙 Excel シートに記載してください。

【詳細報告】

※以下の項目立て、様式に基づき pdf で作成し提出して下さい。

1.1 課題番号

hp130034

1.2 課題代表者

スーリエ クリストフ

1.3 研究の背景と目的

オープンソースの CG レンダリングエンジンである LuxRender は CG レンダリングとしては非常に時間がかかり、1枚の静止画のレンダリングにもレンダリングの要素数にもよりますが、数日掛かる事も珍しくありません。

静止画のレンダリングに主に使用される LuxRender においてアニメーションの製作までを行おうとすると現実的な時間内では終了する事は非常に困難です。

京においてはノード数が数万ノードとなり実現可能と見られるのでその可能性を検証しました。

本トライアルにおいては、計算自体はオープンソースの LuxRender で行い、計算式も組み込まれたものを使用し、京へのインストール、複数同時動作における問題点の検証が主な内容となります。

1.4 計算モデル

LuxRender 組み込みのアンバイアスな MLT 法と物理方程式によるレンダリング

1.5 並列計算の方法と効果(性能)

1枚の静止画を部分的に分割し、計算後合成することによる並列化と、複数のレンダリングデータを同時にレンダリングする事により、並列化を行います。

部分分割と複数シーンの同時レンダリングに関しては相互依存性がなく、通信もありません。理論上は線形的に分割数、同時シーンレンダリング数に比例して計算速度が上がります。ただし、同時実行時の I/O の性能に依存します。

1.6 研究成果(特に「京」の課題の場合は、「京」でこそ出せた成果を明記)

LuxRender のソースを京に移植する部分から始めましたが、x86 アーキテクチャと gcc コンパイラがベースのソースであり、京は sparc CPU で、またコンパイラも gcc ベースのものではなかったので、コンパイルして動作させる事が出来るか検証する事が目的の1つでしたが、無事 g++ でコンパイルできコンパイルに必要なライブラリーも京に移植する事ができました。SSE 等の命令は単純にエミュレートするコードに置き換えることで対応しました。また、シーンデータから京の JOB スクリプトを自動生成する環境を作成し、半自動化を行う事ができました。

図 A におけるサンプルシーンにおいて測定を行いました。図 A は京コンピュータで1フレームあたり 240 分 レンダリングした総計 240 フレーム(10秒)のアニメーションの1フレームです。

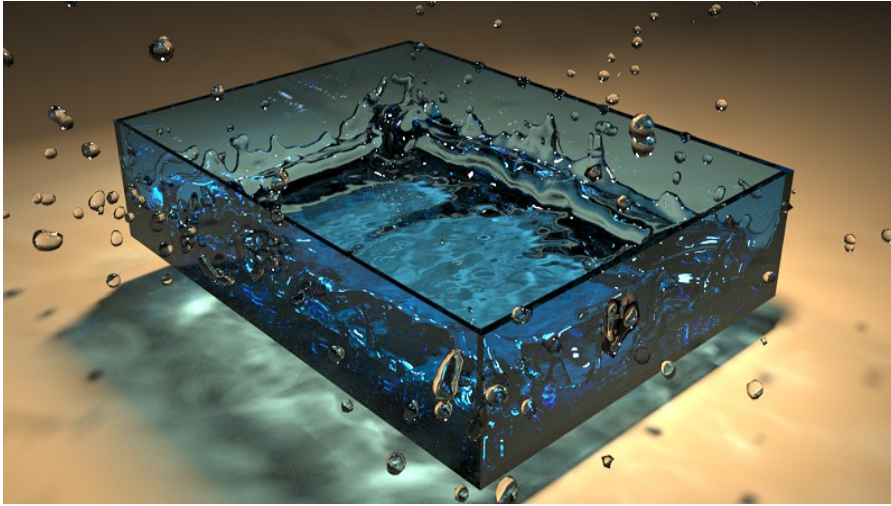


図 A レンダリング結果

図 B は京コンピュータにおける画像分割数に対する一秒間の光線サンプリング数(S/sec)です。1分割に1ノード、8スレッドを割り当てています。これによると分割数が増えても S/sec の値はあまり増減しておらず、分割による同時実行プロセス数によって速度は影響を受けない事がわかりました。

京コンピュータ Samples/sec

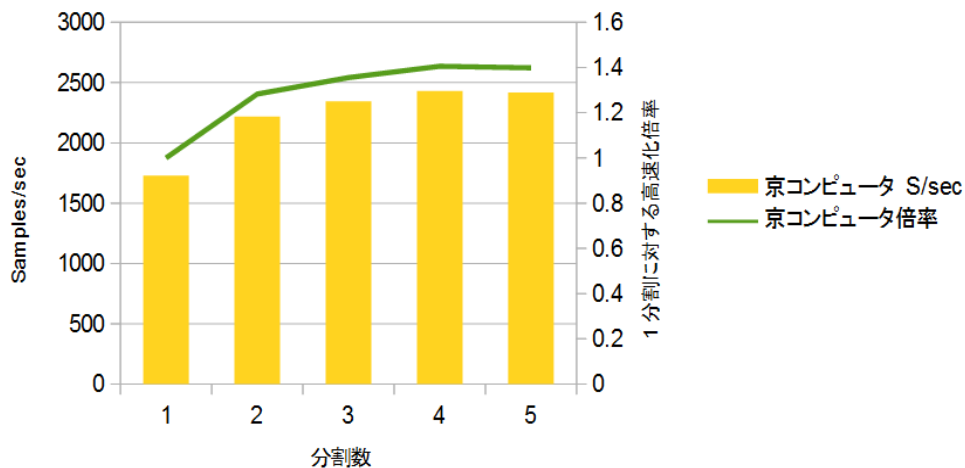


図 B 分割数に対する S/sec

図 C は画像分割に対する10分間のレンダリング中に行われたピクセルあたりのサンプリング数(S/pixel)です。S/pixel は描画の進行度合いを示します。

分割数が増えると線形に S/pixel が増えていき、分割数に応じて良好に高速化が図られている事がわかりました。

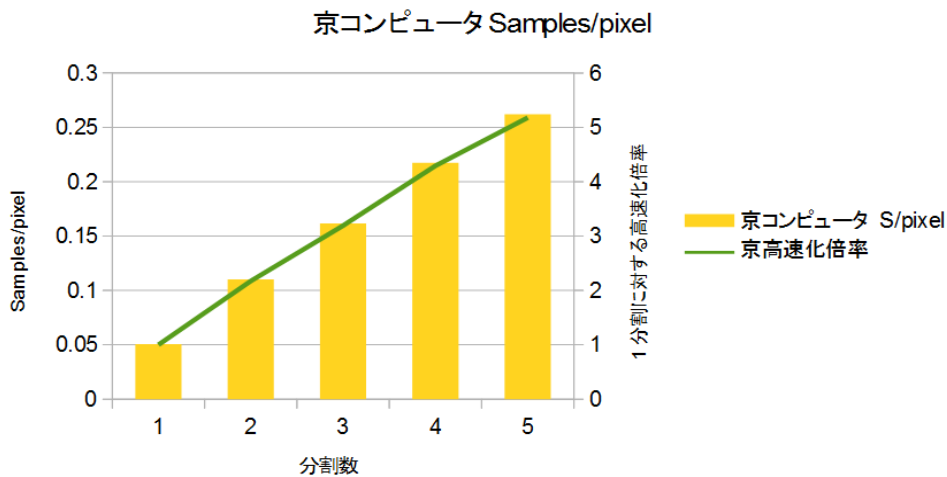


図 C 分割数に対する S/pixel

図 D は40フレームのシーンをノード分散した場合の S/pixel を示します。
 多数のノードに分散した場合 I/O の影響などから速度が遅くなる事が懸念されましたが、実測結果は良好な線形を示しました。

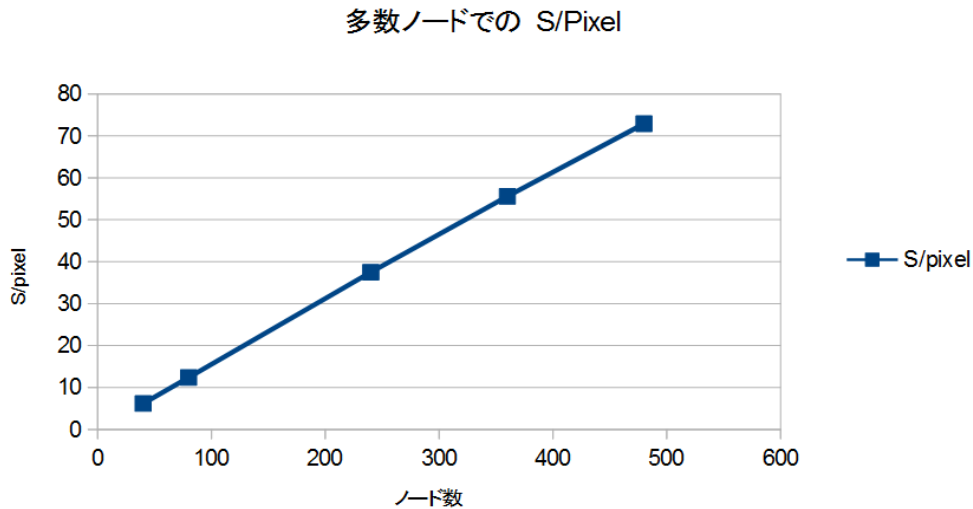


図 D ノード分散における S/pixel

図 E は実際のシーン(120~240フレームのアニメーションシーン 1920x1080 pixel)をレンダリングした結果論理必要時間(フレーム数*分割数*1フレームあたりのレンダリング時間)より何倍速になったかのグラフです。

JOB の制限などから複数の JOB に分割するシステムとなっており、JOB の実行スケジュールが影響しますが、実時間の計測という意味から速度を計測しました。

このグラフによると現在のシステムでは、大体 1000 倍程度の高速化が見込め、LuxRender のようなアンバイアスなレンダリングエンジンを使用してアニメーションを数時間程度で得る事が可能な事が実証できました。

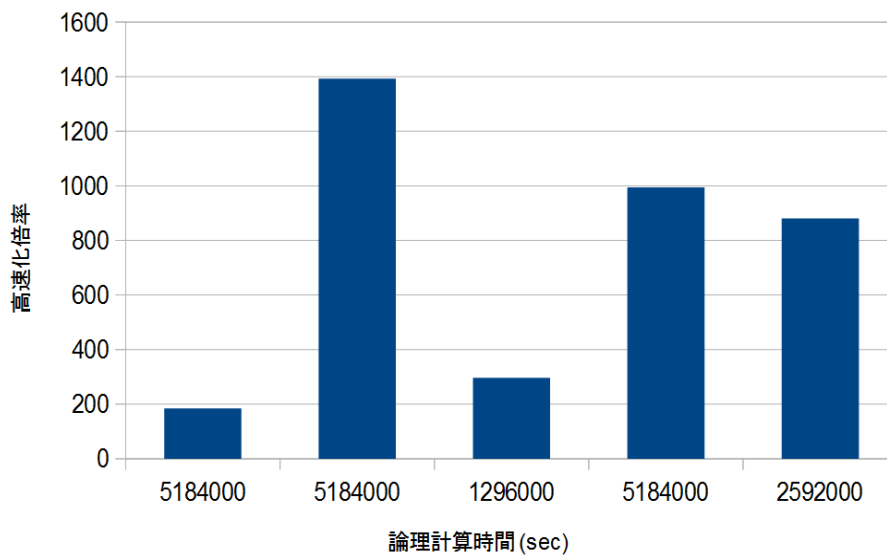


図 E サンプルシーンのレンダリング高速化率

1.7 まとめと今後の課題

今回のトライアルでオープンソースのレンダリングエンジンである LuxRender を京コンピュータ上で動作できる事が確認されました。

また、懸念していた多数ノードによる I/O による速度低下も実測により影響が殆どない事が判明しました。

市販のオープンソースの CAD で作成したアニメーションシーンデータよりレンダリングスクリプトを自動生成する環境も構築できそうなことが判明し、一般的には時間的に実現できないであろう写実的、高精細なレンダリングによるアニメーションを作成する基盤が京コンピュータにおいて実現できる事が実証されました。

実際にアニメーションシーンをレンダリングし、通常の可視化環境であれば 1ヶ月掛かるレンダリングがわずか数時間で終わる事を確認できました。

一方、sparc やコンパイラによる最適化の問題が露呈し、図 F に示すとおり x86 ベースの PC に対して現在は 0.1 倍速程度と最適化の余地が残されています。

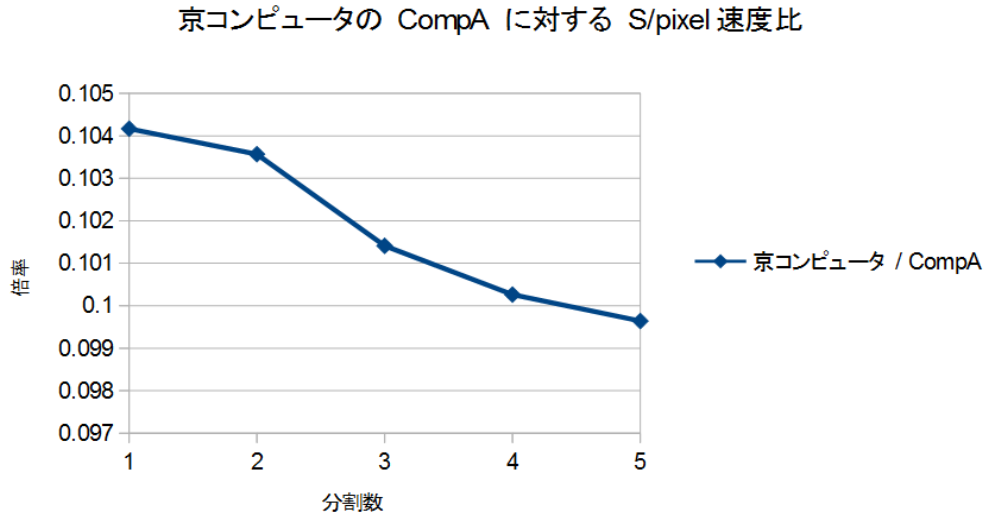


図 F コンピュータ A に対する分割数における速度倍率

また、画像分割による並列化には最終生成画像において限界があり、1枚の画像を高速にレンダリングできる環境にはなっておらず、その為には内部的に MPI による光線計算の並列化が必要です。

1.8 参考文献(本報告文をまとめる上での参考文献であり、網羅的な成果発表リストである必要はない)

[1] http://www.luxrender.net/en_GB/index (LuxRender ホームページ)

[2] http://www.luxrender.net/wiki/Main_Page(LuxRender WIKI ページ)

様式:

- ・図表挟み込み A4 版 2 頁以上 + 参考文献
- ・文字 11 ポイント、2000 字以内 / 頁、MS 明朝、Century、周囲の余白は各 25mm

平成24年度 HPCIシステム利用研究課題 利用報告書 別紙

課題番号	hp130034
------	----------

【利用計算資源情報(システム情報)】

※下表の該当する利用計算資源の欄に、割当資源量、実績資源量をご記入ください。

機関名	資源名	単位	割当資源量 (通期)	実績資源量 (通期)
理化学研究所	「京」	ノード時間	50,000	49,288
北海道大学	SR16000/M1	ノード時間		
		TB		
	BS2000	ノード時間		
		TB		
	RENKEI-VPE	ノード時間		
東北大学	SX-9	ノード時間		
	Express5800	ノード時間		
	共有DISK	TB		
筑波大学	T2K-Tsukuba	ノード時間		
		TB		
東京大学	HA8000	月数		
		TB		
	GPUクラスタシステム	ノード時間		
	データ解析クラスタ 東拠点	ノード時間		
	データ解析クラスタ 西拠点	ノード時間		
東京工業大学	TSUBAME2.0	口数		
		TB		
	RENKEI-VPE	VM		
名古屋大学	M9000	CPUコア時間		
	HX600(一般利用ノード)	ノード時間		
	HX600(占有利用ノード)	ノード時間		
	FX1	ノード時間		
京都大学	CRAY XE6(32ノード通期利用)	時間h		
	CRAY XE6(128ノード集中利用)	週		
		TB		
大阪大学	SX-8R	シェア率%		
	SX-9	シェア率%		
	PCクラスタ	シェア率%		
	共有DISK	TB		
九州大学	PRIMEHPC FX10	口数		
		ノード数		
	PRIMERGY CX400S1	口数		
		ノード数		
RENKEI-VPE	ノード時間			
東京大学	共用ストレージ	TB		

【ソフトウェア利用情報】

※以下に、課題実施において利用したソフトウェア(プログラム)名をご記入ください。

ソフトウェア名
LuxRender,ruby,g++,cmake,tar,独自ツール