

多種モデルの描画に特化したスケーラブルな 複数ノードによる並列レンダリングの台数効果の検証

石井翔[†] 齋藤豪[†]

[†]東京工業大学 情報理工学院

1 はじめに

描画処理における大きな負荷を分散するために複数のサーバを用いて描画する手法は並列レンダリングとして知られている [1, 2]。巨大なモデルを事前に分割し分散描画する手法 [3] や空間を直方体に分割しそれぞれの空間毎に描画を行う手法 [4] 等が存在するが、各ユーザが各々のアバターのモデルをアップロードして用いるような仮想空間を考えた際には、各モデルは多様性があり、それぞれ複雑な状態を持ち、独立して動作する状況となることから新たな並列描画手法が必要となる。そこで本稿では、1つのサーバプロセスがユーザがアップロードした1つのモデルのみを保持してその描画を担当し、各ユーザ視点に合わせて複数プロセスの描画結果の合成を行い配信する分散レンダリングアーキテクチャを提案する。また、実際にユーザ数やモデルの種類が増えた際の描画性能に対するサーバの台数効果について評価と考察を行う。

2 手法の提案

2.1 アーキテクチャの概要

本提案による並列レンダリングアーキテクチャは我々がすでに提案した「コンテンツノード」、「マージノード」の2層からなるアーキテクチャ [5] を拡張したものであり、図 1a に示す、「コンテンツプロセス」、「マージプロセス」、「多段マージプロセス」の三種類のプロセスから成る。「コンテンツプロセス」はシーン上のモデルそれぞれを担当するサーバプロセスであり、マージプロセスから視点情報やカメラパラメータを受け取り、コンテンツプロセスが管理するモデルの大きさを AABB など近似し考慮して計算された部分視錐台 (図 1b) による射影行列を用いて描画結果及び深度を送信する (図 1a-(1))。「マージプロセス」はコンテンツプロセスから返信された部分的な描画結果を深度を考慮して色及び深度を合成し、マージマージプロセスに送信する (図 1a-(2))。さらに、「マージマージプロセス」がこれらマージプロセスからの画面全体の深度及び色を受け取り最終的な色を合成しクライア

ントに送信する (図 1a-(3))。

コンテンツプロセスは担当モデルのみを描画するため、シーン全体のモデルデータを保持する必要がなく、担当モデルの描画負荷に応じて割り当てるサーバを変える事が可能となる。また、我々の以前のアーキテクチャ [5] ではマージプロセスのスケーリングについて考えられていなかったが、今回導入したマージマージプロセスによって部分描画画像の合成処理のスケーリングを可能にした。

3 実験と評価

提案アーキテクチャを用いて大量のモデルを複数台のサーバに分散して描画する実験を表 1 の条件で行った。この実験では1種類のモデルのみを用いたが、各モデルに対して1コンテンツプロセスを割り当て、各コンテンツプロセス間ではメモリ等を共有していない。そのため、実際には同じ計算負荷の別のモデルをそれぞれのコンテンツプロセスで表示した場合と同一として考えることができる。実験環境として東工大の分散並列型のスーパーコンピュータ TSUBAME3 [6] を用いて、2、4、8、16 台の計算サーバを用いて、32、64、128 体のモデルを描画してそれぞれの種類のプロセスでの1秒間の合計処理フレーム数の平均を計測した (図 2)。コンテンツプロセスは各計算サーバへ均等に割り当てた。また、マージプロセスは2台のサーバに付き1つ割り当て、内一つのサーバでマージマージプロセスとクライアントを1つずつ動作させた。2台における実験ではマージマージプロセスを動作させる必要が無い場合、その場合はマージマージプロセスの結果は載せていない。

表 1: 実験に用いたモデルと描画形式

モデル (1 体あたり)	
ポリゴン数	61666 ポリゴン
関節数	59 関節
アニメーション	57 の関節駆動
描画方式	
描画形式	Torrance-Sparrow model + IBL
アンチエイリアス	4xMSAA
解像度	512x512

サーバ数が増えた場合にいずれの場合においてもコンテンツプロセスとマージプロセスの合計処理フレーム数が上昇し、台数効果が確認できる。2サーバの結果を見ると本稿で指摘した問題の通りコンテンツ数

The number effect of scalable parallel rendering by multiple nodes specialized for drawing various models

[†] Kakeru ISHII

[†] Suguru SAITO

School of Computing, Tokyo Institute of Technology ([†])

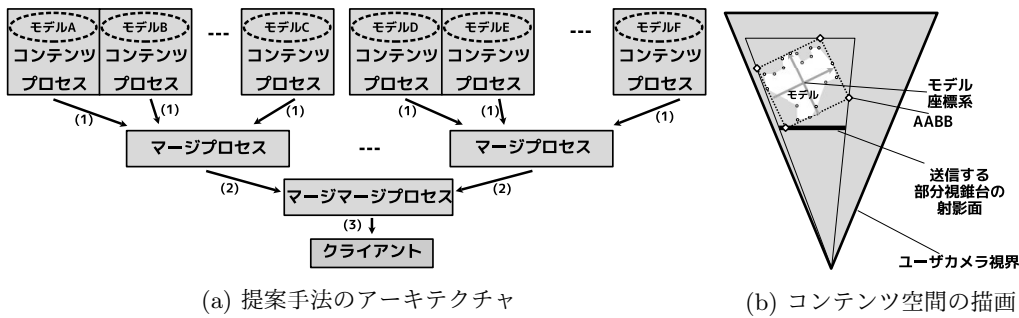


図 1: 提案手法の概要

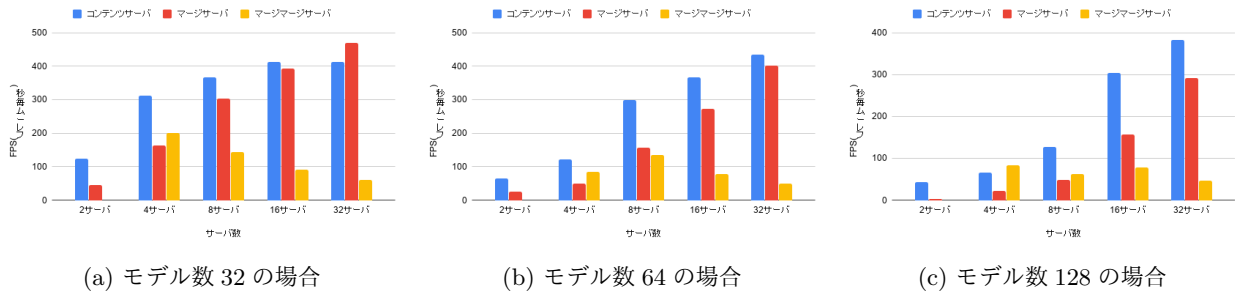


図 2: サーバ数の変化に伴う各種のプロセスの 1 秒間の合計処理フレーム数

が増えるとマージプロセスの描画能力が大きく低下してしまう事がわかる。マージマージプロセスの導入によってプロセス数が増えても 1 マージプロセスあたりが接続するコンテンツプロセスを分散して削減することができるため、合計処理フレーム数が頭打ちにならずマージサーバの描画速度が改善したと考えられる。しかし、一方でモデル数やサーバ数の多い状態では、各種プロセスの合計処理フレーム数に大きな差が見られ余計なフレーム待ちなどが生じてしまっていると考えられる。実際にはモデル数や各モデルの描画負荷、サーバの性能によってコンテンツプロセスに対するマージプロセスの比率や多段化の階層を最適に調整する必要があると思われる。

4 結論と今後の課題

我々の以前の提案手法ではマージプロセスに負荷が集中し、台数効果に限界があった。今回は多段マージプロセスを導入することでその問題を解決した。今後は複数クライアントへの同時送信時の負荷の調査とその負荷の低減手法について検討してゆきたいと考えている。

参考文献

[1] S. Molnar, M. Cox, D. Ellsworth, and H. Fuchs. A sorting classification of parallel rendering. *IEEE CG&A*, Vol. 14, No. 4, pp. 23-32, July 1994.

[2] Stefan Eilemann, David Steiner, and Renato Pajarola. Equalizer 2.0 - convergence of a parallel rendering framework. *IEEE TVCG*, Vol. 26, No. 2, pp. 1292-1307, 2020.

[3] Rudrajit Samanta, Thomas Funkhouser, Kai Li, and Jaswinder Pal Singh. Hybrid sort-first and sort-last

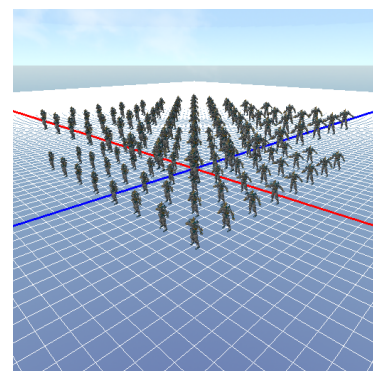


図 3: 実験に用いたシーン (128 モデルの場合)

parallel rendering with a cluster of pcs. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware, HWWS '00*, pp. 97-108, New York, NY, USA, 2000. Association for Computing Machinery.

[4] 渡部雅人, 齋藤豪, 中嶋正之. 空間領域分割による分散レンダリングにおける画像合成並列化手法. 電子情報通信学会 2008 総合大会 D-11-104, 2008.

[5] 石井翔, 齋藤豪. 多種多様なコンテンツへのスケーリングに特化したパラレルレンダリングを用いたサーバーサイドレンダリングアーキテクチャによる合成 cg 作成法. 情報処理学会研究報告 (CG) 2019-CG-173,5, pp. 1-7, 2019.

[6] S Matsuoka, T Endo, A Nukada, S Miura, A Nomura, H Sato, H Jitsumoto, and A Drozd. Overview of tsubame3.0 green cloud supercomputer for convergence of hpc ai and big-data. *e-Science Journal*, Vol. 16, pp. 2-9, 2017.