

使用コア数の動的変更による仮想環境上での In-Situ Visualization の高速化に関する検討

樋口 耀[†] 和田 康孝[†] 山中 脩也[†]

[†]明星大学情報学部

1 はじめに

計算機の並列度向上により、In-Situ Visualization のようなシミュレーション計算と可視化を同時に実行する新しい可視化手法が注目されている。システム導入コストの観点から、クラウド環境上でこのような HPC 向けアプリケーションを実行する要求が高まっているが、クラウドシステムが前提とする仮想環境上では負荷状況が刻一刻と変わっており、そのため安定した性能が得られないことが考えられる。そこで本稿では、同じ物理ノードにおいてシミュレーションおよび可視化処理と並行して別のアプリケーションが動作する際の性能の変動を評価し、その結果から、負荷の状況に応じてシミュレーションと可視化それぞれに割り振るコア数を動的に変更することにより仮想環境上で In-Situ Visualization の高速化を行う方針について検討する。

2 In-Situ Visualization

従来の可視化手法においては、あらかじめ対象とするシミュレーション全体を実行した上で、その出力結果を別途可視化ツールを用いて処理していた。しかしながら、この方法では可視化された結果の確認や、その結果に基づいて可視化のパラメータを調整することに多大な時間を要する。

その問題を解決するために、現在では In-Situ Visualization と呼ばれる手法が用いられるようになってきている。In-Situ Visualization では、シミュレーション計算と同時に可視化を行うことにより、リアルタイムに結果の確認やパラメータ変更が可能となり、上記の問題を大きく改善することができる。ただし、そのためには可視化に関わる処理を並行に行うためのリソースが必要となる。

Performance Improvement of In-Situ Visualization on a Virtual Environment with Damaris Core Assignment
Yo HIGUCHI[†], Yasutaka WADA[†], and Naoya YAMANAKA[†]
[†] Dept. of Information Science, Meisei University
2-1-1 Hodokubo, Hino-shi, Tokyo 191-8506 JAPAN
mail:14j5109@stu.meisei-u.ac.jp

2.1 Damaris/Viz

In-Situ Visualization を実現する手法およびソフトウェアは複数提案・実装されているが、本稿では Inria によって開発・公開されている Damaris[2,3]を対象とする。Damaris では、計算機のプロセス数増加により性能をだす上でボトルネックとなっている I/O ジッタを削減する目的で考案された、各ノード上に I/O 専用コアを置くアプローチ方法である。Damaris/Viz では、この Damaris を用いて In-Situ Visualization 環境を実現している。また、VisIt[4]等のツールにオンラインでデータを送信し可視化することが可能である。

3 他アプリケーションの影響による

シミュレーション性能低下に関する予備評価

In-Situ Visualization 環境では、実際にシミュレーションを行うプロセスと並行して可視化に関わるプロセス（データの集約とデータの可視化を担当する）が同時に動作するため、プロセス毎に負荷特性が大きく異なることが考えられる。そのため、同じ物理ノード上で同時に動作するアプリケーションの影響が状況によって異なることが予想される。

本予備評価では、シミュレーションおよび可視化を担うプロセスの他に別のアプリケーションを動作させることで、実際のシミュレーション実行性能への影響を調査する。

3.1 評価環境および評価方法

本予備評価では、表 1 に示す環境を用いて評価を行う。表 1 に示す評価環境では、物理コア数が 4 となっており、CPU インテンシブな処理においては、4 プロセス以上で性能に大きな影響を及ぼしあうことが予想される。

本評価では、理想的な状況を想定し、シミュレーションおよび可視化以外のアプリケーションが動作しない状況において、Damaris/Viz を用いた In-Situ Visualization を 500 ステップ実行し、実行時間を計測する。その後シミュレーションとは別のプログラムを並行して実行させ、

人工的コアのタイムスライスを発生させた。シミュレーションとは別のプログラムで使用するコア数と変化させて同じく 500 ステップのシミュレーションおよび可視化に要する時間を計測する。

本予備評価では、シミュレーションとして 3D のマンデルブロ集合演算を、他プロセスとしてフィボナッチ数列計算させるプログラムを用いた。なお、この 3D マンデルブロ集合演算は Damaris/Viz に付属するものを拡張して用いた。

表 1 実験環境

CPU	Intel®core™ i5-6400 CPU @2.70GHz 4 コア
メモリ	16GB
コンパイラ	gcc 5.4.0
可視化ツール	VisIt (ver 2.10.3)

3.2 評価結果

本予備評価の結果を図 1 および 2 に示す。図 1 はシミュレーションの並列度を 1, 可視化を担当するプロセス数が 1 とした場合の評価結果であり、図 2 はそれぞれ 2 および 1 とした場合の評価結果である。

4 予備評価に基づく検討

予備評価前の想定としては、プロセス数に対して不足する物理コア数が 1 つであれば可視化プロセスと外部アプリケーションが競合することによって性能低下が大きくなることが予想される。それよりも競合するコア数が増えた場合には、シミュレーションのプロセスにも影響が及ぶためさらに大幅な性能低下が起こる事が予想された。

図 1 及び図 2 から、競合コア数が 0 から 1 に変わる場合の性能低下、1 から 2 になった場合の性能低下を比較した結果、性能の低下に大きな差が無い事がわかった。本評価ではプロセスのコアへの割り当てを OS が行っているため、割り当てが流動的であることが原因であると考えられる。プロセスのコアへの配置を固定させることで性能低下への影響を限定させることができると考えられる。

予備評価の結果から、競合が起こっているコアに可視化とシミュレーションどちらのプロセスを動的に配置できるかをさらに詳細に検討する必要がある。プロセスとコアの配置、つまり競合するプロセスの関係を固定した状態での評

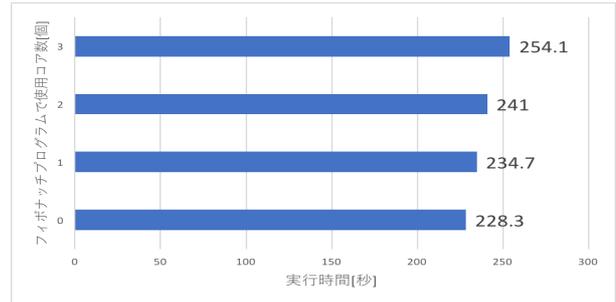


図 1 シミュレーションコア 1 可視化コア 1

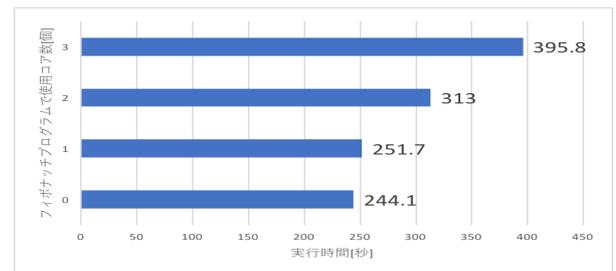


図 2 シミュレーションコア 2 可視化コア 1

価、またさらにコア数の多い環境における評価を行うことによって、プロセス数をどのように変化させるか、プロセスの競合の状態をどのように保つかを検討する必要がある。

5 まとめ

本稿では、仮想環境上で In-Situ Visualization を高速化することを想定し、シミュレーションおよび可視化と並行して同じノード上で実行されるアプリケーションの影響について評価を行った。また、その結果に基づいて、全体のスループット低下を抑制するためには、どのようにシミュレーションの並列度や可視化を担うプロセスの配置を調整すれば良いかについて検討を行った。

今後の課題としては、上述のような詳細な評価、およびその評価結果に基づく検討内容を実際に Damaris 上に反映させ、動的にこれを実現することなどが挙げられる。

参考文献

- [1] M. Rivi, L. Calori, G. Muscianisi, and V. Slavnic. "In-Situ Visualization: State-of-the-art and Some Use Cases".
- [2] M. Dorier, R. Sisneros, T. Peterka, G. Antoniu, D. Semeraro, "Damaris/Viz: a Non-intrusive, Adaptable and User-Friendly In Situ Visualization Framework", in Proc. of the IEEE LDAV 2013 conference. October 2013.
- [3] Damaris: <https://project.inria.fr/damaris/>
- [4] VisIt: <https://wci.llnl.gov/codes/visit/>