TSUBAME 共同利用 平成 24 年度 学術利用 成果報告書

利用課題名 FMM を利用した分子動力学シミュレーションコードの開発
英文：Development of a Software for Molecular Dynamics Simulations with the Fast Multipole Method(FMM)

利用課題責任者 泰岡 顕治
First name Surname Kenji Yasuoka

所属 慶應義塾大学
Affiliation Keio University
URL http://www.keio.ac.jp/

邦文抄録（300 字程度）
分散コンピューティング環境において分子動力学アプリケーションを効率的に開発するために、GPU 仮想化ミドルウェアを開発し TSUBAME2.0 上で性能評価を実施した。この GPU 仮想化ミドルウェアを、シングルノード用の複数 GPU に対応した既存のレプリカ交換法分子動力学アプリケーションに適用して，同一ソースコードでシングルノード環境と分散コンピューティング環境の両方でシミュレーションを実行できることを確認した。

英文抄録（100 words 程度）
We have developed a GPU virtualization middleware to simplify the development of Molecular Dynamics simulation program on distributed computing system, and evaluated their performance on TSUBAME 2.0 supercomputer. In addition, we applied this middleware to existing Replica-Exchange Molecular Dynamics simulation program, which had been developed for a single node computer, and confirmed that the middleware enables us to use the same source code both for single- and multi-node computing systems.

Keywords: **分子動力学シミュレーション**, **高速化**, **大規模並列計算**, GPU, FMM

背景と目的

従来、分子動力学における長距離力の計算には Ewald 和をベースとする手法が用いられてきた。 これらの手法には計算の分散並列化による高速化を行いにくい FFT 演算が含まれていることから、結果として大規模並列計算による高い並列化効率を達成することは難しい。 一方、同様の計算に Fast Multipole Method (FMM)アルゴリズムを適用することで、高い並列化効率を得られることが近年の研究例で報告されている。 本課題の目的は、この FMM を用いた分子動力学シミュレーションコードの開発を行うことである。 その開発過程において、分子動力学シミュレーションの大規模並列計算で重要になる長距離力の計算の高速化，並列化に関する手法を提案し，詳細な性能評価，検討を行う。計算の高速化が達成されれば、従来の手法では大規模並列計算機を用いてもスケーラビリティの観点から解析が困難であった問題の解析が可能になる。 この開発は今後様々な分子動力学アプリケーションソフトウェアが次世代の超並列計算機上で性能を発揮する上でも重要な役割を果たすと考える。 国外では次世代の Exascale ソフトウェアを念頭においた FMM に関する様々な研究が既に始まっており，国際競争の観点からも FMM の高性能化は重要な課題であるといえる。

また、近年のスーパーコンピュータのシステム構成は多数のプロセッサコア(CPU/GPU)を高速なインターコネクトで相互接続したクラスタ・タイプが主流となっている[1]。 数 10 万のプロセッサコアと深く階層化されたメモリ構成を含む現状の分散コンピューティング環境において計算速度パフォーマンスを十分に引き出すことは難しく、OpenMP・MPI・CUDA といった複数の開発言語基盤に通じたスキルを要求される。 このことは、市販のパーソナルコンピュータ上で動作している既存のアプリケーションをスパコン上で高速化させようとする場合に分子動力学とは直接的に関係しない開発コストが要求されることを意味する。 それらの開発コストを低減させて分散コンピューティング環境下での分子動力学アプリケーション開発のプログラマビリティを向上させる効率的な手法を提案することも本課題の目的である。

概要

TSUBAME2.0 のような分散 GPU コンピューティング環境での高いプログラマビリティを確保し、分子動力学アプリケーション開発を効率化するために、GPU 仮想化ミドルウェアを開発し性能評価を行った[2]。 また既存のレプリカ交換分子動力学アプリケーションに適用し、分散コンピューティング環境における並列化効率の評価を行った。

FMM アルゴリズムの適用に関しては、遠距離力の1つであるクーロン力計算のみを分子動力学アプリケーションの1つである”CHARMM”と組み合わせてテストを行った。 また、独自の分子動力学シミュレーションコードに FMM アルゴリズムを組み込み、開発した GPU 仮想化ミドルウェア上で試験的に実行した。

結果および考察

複数の計算ノード上に多数の NVIDIA 社製 GPU デバイスがインストールされている分散コンピューティング環境において、分子動力学等のアプリケーションから多数の GPU をシームレスに利用するための GPU 仮想化ミドルウェア”DS-CUDA”(Distributed-Shared CUDA)を開発し，性能評価を TSUBAME2.0 上で実施した[2]。

DS-CUDA は、図 1 に示すように TSUBAME2.0 のような GPU クラスタシステム（**図 1** 左半分に示す）上で動作させることで、あたかもローカルマシン上に多数の GPU がインストールされているかのように（**図 1** 右半分に示す）扱うことを可能にする。

図 1. GPU 仮想化ミドルウェア”DS-CUDA”の動作概念図

図 2 および**図 3** に、開発した GPU 仮想化ミドルウェア DS-CUDA の構成図を示す。 本ミドルウェアはコンパイラプリプロセッサ、ライブラリ(**図 3** 中の”CUDA API Wrapper”) およびサーバプログラム（**図 2** 中の ”Server0”/”Server1”、 **図 3** 中の ”DS-CUDA Server”)から構成される。 コンパイラプリプロセッサは、通常の CUDA に対応した(MPI 通信処理に関する記述を含まない)C/C++ソースコードを事前処理することで、ローカル PC にインストールされている GPU へのアクセス記述を、インターコネクトで接続されている別ノード上の GPU へのアクセス記述に差し替える。 ライブラリは、専用プリプロセッサが差し替えた部分のネットワーク呼び出し処理ルーチンを提供し、InfiniBand と Gigabit Ethernet の2種類のインターコネクトを使ったクライアント-サーバ通信処理ルーチンを提供する。 サーバプログラム(**図 2** 中の”Server0/1”)は、クライアント側(**図 2** 中の”Client”)から送信された命令を解釈して、サーバノード上にインストールされている GPU の制御を行う。

本ミドルウェアの特徴として、単一ノード上の複数 GPU にしか対応していない既存の分子動力学アプリケーションに適用することで、ソースコード上で明示的にノード間通信のためのコードを書くことなく、ソースコードの変更なしで TSUBAME2.0 分散コンピューティング環境に対応させられることが挙げられる。

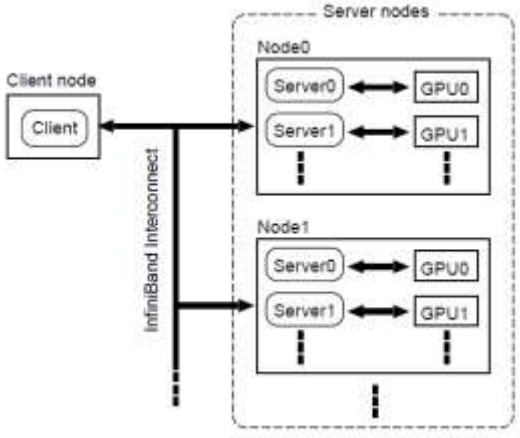

図 2. GPU 仮想化ミドルウェア”DS-CUDA”の構成図（参考文献[2]より抜粋）

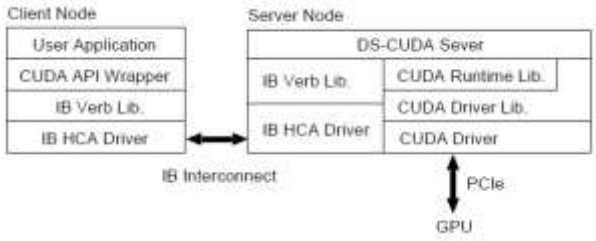

図 3. GPU 仮想化ミドルウェア"DS-CUDA"のソフトウェアレイヤスタック（参考文献[2]より抜粋）

図 4 に、TSUBAME2.0 の 22 ノード(64GPU デバイス)を使用して測定した DS-CUDA の計算速度の実測結果を示す。 評価に利用したアプリケーションプログラムは、NVIDIA 社から配布されている GPU アプリケーション開発環境”CUDA Toolkit”に含まれているサンプルプログラムの1つ、モンテカルロ法によるオプション・プライシングを計算する”MonteCarloMultiGPU”を利用した。グラフ中の “OPT_N”は計算する銘柄数を表し、weak スケーリング評価用として GPU 並列数の 256 倍に、strong スケーリング評価用として{256, 2048}にそれぞれ設定した。

最大で64GPUを使用して測定を行った結果、weakスケーリングで 95%、strong スケーリングで 68%の示す並列化効率が得られた。

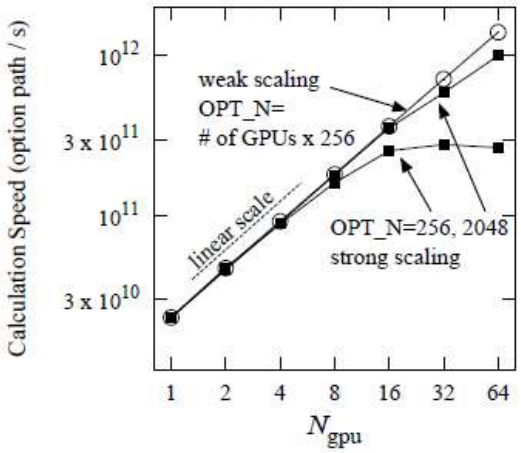

図 4. DS-CUDA を使った計算速度の並列化効率（参考文献[2]より抜粋）

次に、上述の GPU 仮想化ミドルウェアを実際に分子動力学アプリケーションに適用した例を示す。 具体的には”レプリカ交換法”と呼ばれるアルゴリズムを使用して単一アルゴン原子の融点を求める分子動力学シミュレーションに適用した。 レプリカ交換法とは、レプリカと呼ばれる温度のみ異なる系を並列してシミュレーションすることにより、広い温度範囲のエネルギー状態をサンプリングし、系の安定状態を効率的に得ることのできる手法の一つである[3]。 このアルゴリズムの特徴として、レプリカそれぞれの系が独立しており、レプリカ間の分子間力の計算を必要としないことから計算を並列化しやすいことが挙げられる もともと単一ノード上の複数 GPU に対応した CUDA プログラムがあり、そのソースコードを変更することなく DS-CUDA により複数ノード上の GPU に対応して使用している。

図 5 に最大 64GPU を使用した並列化効率の実測結果を示す。 高速・低レイテンシを特徴とする InfiniBand LAN、広く一般に普及している Gigabit-Ethernet LAN、および WAN(商用インターネット網)の3種類のインターコネクトを同一の分子動力学シミュレーションコードを用いて評価した[4]。 InfiniBand LAN を使用した場合は 88%、 Gigabit-Ethernet LAN を使用した場合には 63%、WAN を使用した場合は 5%の並列化効率の実測結果を得た。 これらの結果は，TSUBAME2.0 を使用して得た結果ではないが、現在 TSUBAME2.0 を使い、より多くの 256GPU を並列化させた場合のパフォーマンスを評価した論文をまとめている。

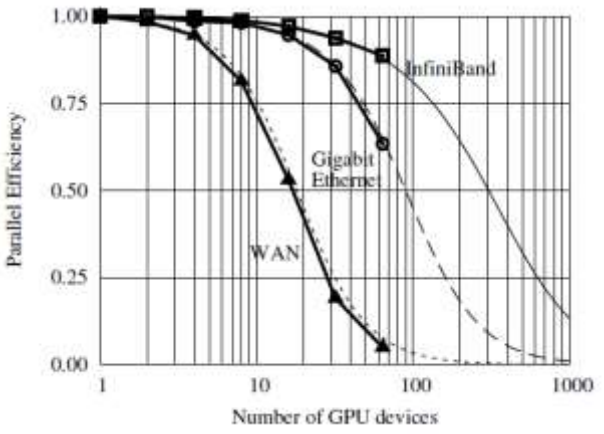

図 5. GPU 仮想化ミドルウェアを分子動力学アプリケーションへ適用した計算速度パフォーマンスの実測例 （参考文献[4]より抜粋）

以上に示したように、開発した GPU 仮想化ミドルウェアの機能を利用することにより、OpenMP や MPI を使わずに分散コンピューティング環境上のアプリケーションを開発することが出来る。 今後、分子動力学シミュレーションの高速化を効率的に行うことが可能と考えられる。

まとめ、今後の課題

分子動力学シミュレーションの高速化を、分散 GPU コンピューティング環境上で効率的に実施するための GPU 仮想化ミドルウェアを開発し、TSUBAME2.0 上で評価を行った。 単一ノード対応の分子動力学アプリケーションに適用し、ソースコードを改変することなしに分散コンピューティング環境にも対応可能であることを確認した。 現在は、256GPU 上の並列化効率を評価中である。

今後の課題は、GPU 仮想化ミドルウェアが、より規模の大きい(目標として Exascale 規模)であっても有効に機能するかを評価することである。 そのためにより多くの GPU を並列化した場合の動作実績を増やし、自動負荷分散機能などを追加実装し、パフォーマンスの評価を行う必要がある。 また FMM アルゴリズムを組み込んだ計算高速化の評価も継続して行っていく予定である。

参考文献

[1] TOP500 Supercomputer Site [Online] http://www.top500.org/, retrieved Mar.2013.

[2] Atsushi Kawai, Kenji Yasuoka, Kazuyuki Yoshikawa, and Tetsu Narumi, “Distributed-Shared CUDA: Virtualization of Large-Scale GPU Systems for Programmability and Reliability”, The Fourth International Conference on Future Computational Technologies and Applications, Nice, France, 2012.

[3] Hukushima,K. and Nemoto,K., J.Phys Soc.Jpn., 65, 6, pp.1604-1608, 1996.

[4] Minoru Oikawa, Atsushi Kawai, Kentaro Nomura, Kenji Yasuoka, Tetsu Narumi, “DS-CUDA: a Middleware to Use Many GPUs in the Cloud Environment”, SHPCloud workshop, Salt Lake City, USA, 2012.