共同利用 成果報告書 平成25年度 課題種別

仮想 GPU を用いた分子動力学シミュレーションコードの開発と評価
Development of Molecular Dynamics Simulation Program using Virtualized GPU

泰岡 顕治
Kenji Yasuoka

慶應義塾大学 機械工学科
Department of Mechanical Engineering,
Keio University
http://www.yasuoka.mech.keio.ac.jp/

邦文抄録　本課題では，スパコン TSUBAME に搭載されている GPU のうち最大 1,024 台を使用して分子動力学シミュレーションの計算を並列化し，その計算速度のスケーラビリティの評価を行った．本研究グループで開発している GPU 仮想化ソフトウェア「DS-CUDA」を使用することで，CUDA 言語によるシングルノード上のマルチ GPU 向けに記述されたソースコードから改変することなくマルチノード GPU クラスタ向けの実行コードを生成している．1,024 台の GPU 並列時の実測結果からは最大で 61%の並列化効率を得られ，算出した期待値と誤差 4%以下で一致し，概ね期待した通りの結果が得られることを確認できた．

英文抄録　We parallelized the molecular dynamics simulation with up to 1,024 GPUs in the TSUBAME supercomputer, and evaluated the scalability of the calculation speed. We utilized the software named “DS-CUDA” which we are developing for virtualizing CUDA APIs for GPUs. It enables us to use the same source code for a single node as for multiple node GPUs without any modification. We obtained 61% parallel efficiency with 1,024 GPUs, and it matched with the estimation model of the performance within 4% error.

Keywords:　virtualization, Graphics Processing Unit, CUDA, molecular dynamics simulation

背景と目的

近年ではスマートフォン，コンシューマ向けの PC からスーパーコンピュータに至るまで，複数の演算コアを1パッケージに封入したマルチコアプロセッサが広く利用されるようになった．とりわけ多くのプロセッサコアを搭載したプロセッサの一つに GPU（Graphics Processing Unit)がある．その名称が示すように画像データ処理専用に開発されたデバイスであるが，汎用的なマルチコアプロセッサに搭載されるコア数が多い場合で数 10 個であるのに対して GPU は数 1,000 個を搭載し，特に科学技術計算の分野においてその高い並列演算速度性能は広く利用されている．現在の高性能計算機構成では，それらのマルチコアプロセッサ・GPU 等のハードウェアアクセラレータを搭載した計算ノードを高速なインターコネクトで接続することにより，数 10,000 個のプロセッサコアを使って並列処理を行い演算処理速度を得ているタイプが多く見られる [1].

多数のプロセッサコアが階層的に構成された計算機環境で効率的にソフトウェアを開発することは，並列処理記述をサポートする拡張フレームワークが普及しているとはいえ単一のコアを対象としたソフトウェアを作成することに比べて困難さを伴うことが多い．

多数の異種プロセッサが混在するヘテロジニアス環境向けのアプリケーションを開発するには，例えばマルチコアプロセッサ向けには OpenMP，GPU 向けに CUDA(Compute Unified Device Architecture)に準じた記述を行い，分散的かつ階層的に構成されたメモリ空間を参照するために MPI（Message Passing Interface）に関するプログラミング技術が必要とされる．また CPU と GPU 間あるいはノード間の通信時間はプロセッサの処理速度にくらべて遅く，通信処理コストを無視できないため効率的な処理のために特殊な並列化技術が必要とされる．そのため特定の計算機環境に特化してチューニングが施されたアプリケーションソースコードは，

別の計算機環境では必ずしも有効に機能するとは限らず，改めてソースコードの見直しが必要になることがある．

本課題では GPU 仮想化ミドルウェアを使用することで，プログラムソースコードを改変することなく，単一ノードの GPU から複数ノードに分散した多数の GPU での並列化に対応した実行プログラムを生成することを目的とし，実用に耐えうる処理速度パフォーマンスを得ることを目指した． その評価対象として特に分子動力学シミュレーションコードを対象とした．その結果，OpenMP や MPI を使うことなく単一プロセスから 1,024 台の GPU を使用して実用に耐えうる性能が発揮できることが分かった．

概要

TSUBAME 上で GPU 仮想ミドルウェアを利用することで，分子動力学シミュレーションの大規模並列計算において重要となる長距離力の計算の高速化と並列化に関する手法を提案し，以下の2つアプローチで性能評価を行った．

1つめとして，本グループで開発を進めている GPU 仮想化ミドルウェア「DS-CUDA」[1,2]を使用し TSUBAME にある多くの GPU を仮想化して分子動力学シミュレーションを対象としてスケーラビリティを評価した． 図 1 に DS-CUDA の構成図を示す． 本ミドルウェアは、単一計算ノードの GPU 向けに開発されたアプリケーションのソースコードをそのまま TSUBAME のような GPU を搭載した大規模分散コンピューティング環境でも利用可能にする機能を有し、分散コンピューティング環境でのプログラム開発を容易にする特長をもつ．

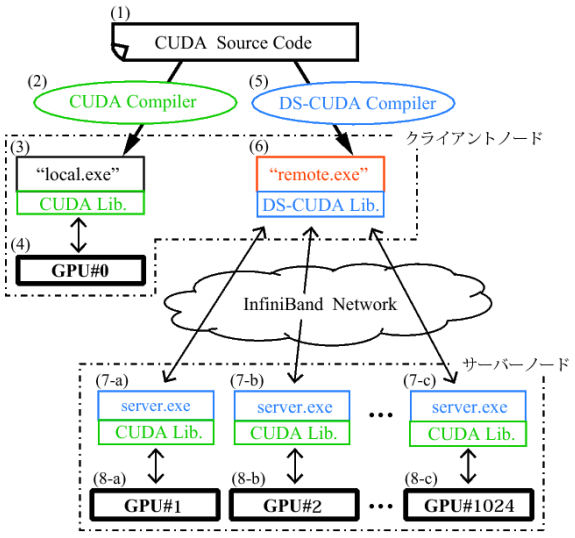

図 1 GPU 仮想化ソフトウェア「DS-CUDA」の概略

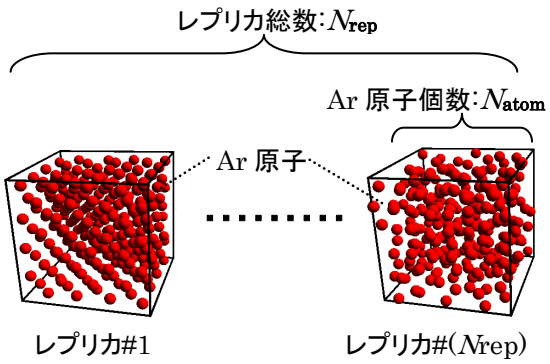

図 2 レプリカ交換分子動力学シミュレーション手法

図 2 に，実施したレプリカ交換分子動力学(Replica-Exchange Molecular Dynamics: REMD)シミュレーションの計算概要を示す． レプリカ総数 Nrep は 14,366，Ar 原子個数 Natom は最大 2,048 個を扱った．並列化効率のモデル式を作成し，計算速度の期待値を算出した．使用した GPU の台数は，最大 1,024 台である．

2つめは，Fast Multipole Method (FMM)を用いた分子動力学シミュレーション用の並列処理コードの開発と評価である． 既存の Ewald 和を用いた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計算規模の問題を解析可能にする．今年度は，開発過程において FMM ライブラリを GPU 上で並列化した場合に，Ewald 和の計算した場合に比べて計算精度が劣化する場合が存在することが分かり，その解析に時間が懸かってしまったため大規模計算を行うには至れなかった． 来年度も開発を継続する予

定である.

結果および考察

図 3 に, 16 台から 1,024 台までの GPU を仮想化して REMD シミュレーションを並列化させたときの並列化効率の期待値(図中の破線)および実測結果(図中のシンボル▲および■)を示す. 期待値は, 別途導出したモデル式とTSUBAME2.5 環境で計測した通信速度パラメタ(バンド幅およびレイテンシ)から算出した. 2 つのグラフ(i) および(ii) はそれぞれ図 2 に示した Ar 原子個数 N_{atom} が 256 個の場合と 2,048 個の場合で表す.

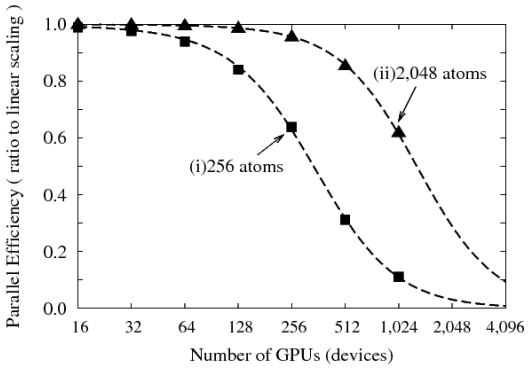

図 3 並列化効率のモデル式と実測結果

(i)より(ii)の方がシミュレーション時間中に占める GPU での処理時間の割合が多く, 通信時間の占める割合は小さい. 1,024 台並列時の並列化効率は, (i)の場合が 11%, (ii)の場合が 61%であり, 全測定ポイントで期待値と誤差 4%以内で一致することを確認した.

並列化効率の期待値 R について説明する. ローカルホストにインストールされた 1 台の GPU を使った場合に費やしたシミュレーション時間を T_{local}, DS-CUDA を利用し n 台の GPU を並列化させて使った場合に費やしたシミュレーション時間を $T(n)$ で表したときの並列化効率 $R = T_{\mathrm{local}} / (n \cdot T(n))$ から算出した. R はストロングスケールした場合の理想的な計算速度増加に対する実効速度の比を表す.

算出されたシミュレーションの消費時間における, 計算時間と通信時間の各要因の割合を図 4 に示す.

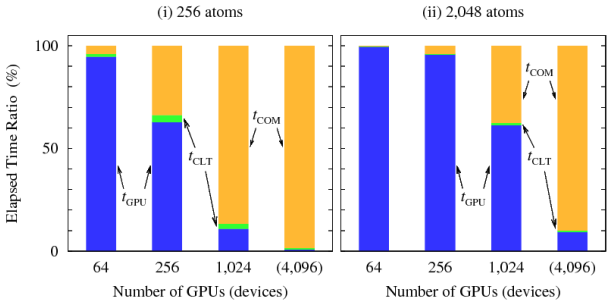

図 4 シミュレーション消費時間の内訳

左右2つのヒストグラム(i)および(ii)は, 図 3 のグラフ(i)および(ii)にそれぞれ対応する. 内訳は, $T(n) = t_{\mathrm{GPU}}(n) + t_{\mathrm{COM}}(n) + t_{\mathrm{CLT}}$ で表し, ヒストグラム中の青色で示した t_{GPU} は仮想化された GPU での計算処理時間, 緑色の t_{COM} はクライアントノードとサーバノードの通信時間, 橙色で示した t_{CLT} はクライアントノードでの処理時間であり, それぞれの合計を 100%に正規化した割合いを示す.

GPU の並列台数が多い場合ほど, シミュレーション時間 $T(n)$ を占める通信時間の割合い t_{COM} は増加し, 特に GPU 並列台数が多い場合には, 計算時間よりも通信時間 t_{COM} の方で大半が消費されている(グラフ(i)の 1,024 並列時ではシミュレーション時間の 87%)ことを示している.

まとめ、今後の課題

開発中の GPU 仮想化ソフトウェア「DS-CUDA」を利用して TSUBAME2.5 上の GPU を 1,024 台まで使った, レプリカ交換分子動力学シミュレーションの並列化を行った. 計測された並列化効率は, 1,024 台並列時に最大で 61%を示し, 算出した期待値と誤差 4%以内で一致した. 1,024 台までの GPU を DS-CUDA を使って仮想化し, 概ね予想通りのスケーリング効果を得ることを確認できた.

今後は, GPU 仮想化の機能を利用して, GPU クラスタで発生する動作異常発生を想定した耐故障性機能を拡充し, 本課題と同様に分子動力学シミュレーションを使った性能評価を行うことを予定している.

参考文献

[1] TOP500, http://www.top500.org, 2014.3

[2] DS-CUDA, http://narumi.cs.uec.ac.jp/dscuda, 2014.3

[3] 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.

[4] 老川稔，野村昴太郎，川井敦，泰岡顕治，成見哲「256GPU を用いたレプリカ交換分子動力学シミュレーションの高速化」，第 18 回計算工学講演会,2013.8