5H-03

PCI Express 拡張 Box と仮想 GPU との計算性能比較

瀬戸口 幸寿† 成見 哲‡
電気通信大学 情報・通信工学専攻

1 はじめに

近年,GPU(Graphics Processing Unit) が科学シミュレーションなどの汎用計算に用いられている. この技術は GPGPU(General-Purpose computing on GPU) と呼ばれており, HPC の分野に大きく貢献している. GPU は本来, 画像処理に特化したプロセッサであるが, 近年の GPU は大規模なデータに対し単純な演算を並列に行うことが可能である.

我々は, ソフトウェアを書き換えることなくネットワークを介して複数の GPU を使用できるようにする GPU 仮想化ツールである DS-CUDA[1] を提案している. GPU の仮想化により, 場所の制限なしに複数の GPU を使用するプログラムを簡単に記述することが出来る. しかし, クライアントノード-GPU サーバ間の通信がボトルネックになる.

本研究では複数の GPU を用いた場合に, DS-CUDA による仮想 GPU とネイティブな GPU との性能を比較し, 性能モデルの構築を行った.

2 システム

本研究では, 以下の手法で複数の GPU を用いた計算性能の比較を行った.

- DS-CUDA:Infiniband ネットワークによる仮想 GPU の利用
- ネイティブな CUDA:PCI Express 拡張 Box による GPU の直接利用

2.1 GPU

本研究では両手法にて用いる GPU として, ELSA GeForce GTX 780[2] を最大 8 台まで用いた.

2.2 CUDA, DS-CUDA

CUDA(Compute Unified Device Architecture)[3] は NVIDIA 社が開発した GPGPU の為の統合開発環境であり, 拡張言語, コンパイラ,C 言語のライブラリから構成されている.

DS-CUDA(Distributed-Shared CUDA)[1] はネットワークを介してリモート GPU を用いる為のミドルウェアである.DS-CUDA のシステムは DS-CUDA コンパイラ,client ノード,(複数の)server ノードから構成される. DS-CUDA コンパイラは client ノード上で, client-server 間通信の為の命令を付加してコンパイルを実行する. これによって DS-CUDA は,CUDA プログラムをローカルな GPU(仮想 GPU) を用いて実行しているように見せる (図 1 左). また, プログラマはネットワークを介した GPU の利用を気にせずに, 同一の CUDA ソースコードを利用できる. 本研究では,8 台の GPU に対して Infiniband ネットワークを介した 8 つの server ノードを構築し実験を行った.

2.3 PCI Express 拡張 box

DS-CUDA と比較の為に,client ノードから直接複数の GPU を繋ぐ仕組みが必要である. 本研究では PCI Express 拡張 box[4] を利用した.PCI Express 拡張 box は PCI-Express バスに挿すホストカード, GPU コンテナ box, 両者を繋ぐケーブルから構成される (図 1 右). 本研究では 4 台格納可能な拡張 box を 2 台用いて, 最大 8 台の GPU を接続する.

2.4 Claret

Clret は古石によって作られた融解塩の分子動力学 (MD) シミュレーションプログラムである [5]. 本研究では Claret を用いて性能比較, モデル構築を行う. オリジナルコードに対し, 複数 GPU への対応を行い, OpenGL による画面描画機能は単純化の為無効化した. 粒子数を n とすると 1 回の MDstep に対して CPU-GPU 間の通信量は $O(n)$,GPU での計算量は $O(n^2)$ である.

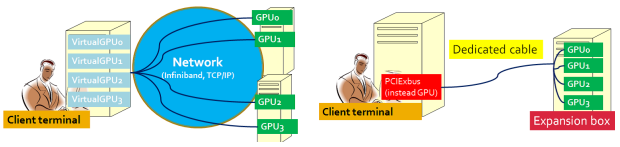

図 1 (左)DS-CUDA 使用時の GPU の接続
(右)PCI Express 拡張 Box 使用時の GPU の接続

Performance Comparison of GPGPU between PCI Expansion Box and Virtualization via DS-CUDA
† Setoguchi Yukitoshi
The University of Electro-Communications
‡ Narumi Tetsu
The University of Electro-Communications

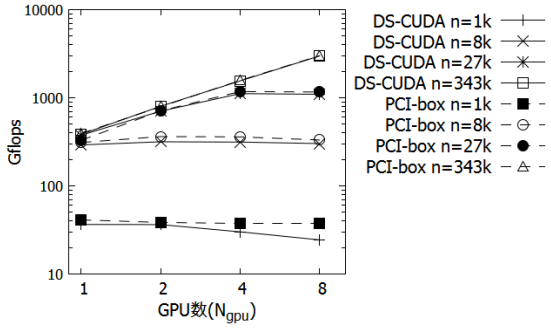

図 2　GPU 数に対する計算性能

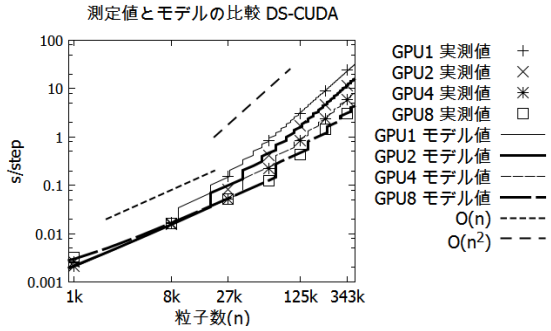

図 3　1step の計算時間の実測値とモデルの比較 (DS-CUDA の場合)

3　実験結果

粒子数,GPU の並列数, そして GPU の利用手法を変えながら MD シミュレーションを実行した. 測定した 1step における計算時間を基に, GPU の並列数に対する計算性能 (Gflops) を図 2 に示す. 粒子数が多い時には GPU 数に応じて性能が向上しているが, 粒子数が少ない時は逆に低下している.

4　モデルの構築

1step の計算時間 T を GPU で要する計算時間 (T_{gpu}),CPU で要する計算時間 (T_{cpu}), server-client 間の通信時間 (T_{com}) の和で以下のようにモデル化する.

$$T_{gpu} = \lceil \frac{n}{kcN_{gpu}} \rceil nkct_{gpu} \quad (1)$$

$$T_{cpu} = t_{cpu} n \quad (2)$$

$$T_{com} = \left(\frac{16nN_{gpu}}{B_{h2d}} + N_{gpu}L_{h2d} \right) + \left(\frac{12n}{B_{d2h}} + N_{gpu}L_{h2d} \right) + t_{kernel}N_{gpu} \quad (3)$$

$$\begin{cases} n & : \text{粒子数} \\ N_{gpu} & : \textit{GPU} \text{ 並列数} \\ t & : \text{測定で決定する各係数 (sec)} \\ k & : \text{測定で決定する整数 (今回は 4)} \\ c & : \textit{GPU} \text{ あたりのコア数 (今回は 2304)} \\ B & : \text{転送スループット (byte/sec)} \\ L & : \text{転送レイテンシ (sec)} \end{cases}$$

ここで,kcN_{gpu} は 1 度に実行出来るスレッド数, $h2d$ は client(host) から server(device) へ, $d2h$ は server から client への転送を表し, t_{kernel} はカーネル実行の遅延を表す. 測定の結果 L,$1/B$,t_{kernel} いずれも DS-CUDA の方が大きく, 小粒子, 並列大領域における PCI Express 拡張 box との差を決定づけていることが分かった. DS-CUDA のモデルと実測値の比較を図 3 に示す. 粒子数の少ない時は粒子数に比例して計算時間が上昇していくが, 粒子数が多い時は 2 乗で増えている. またモデルにおける

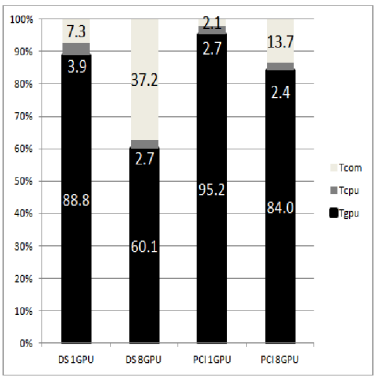

図 4　1000 粒子における T_{gpu},T_{cpu},T_{com} の比率

T_{gpu},T_{cpu},T_{com} の比率を図 4 に示す. GPU 数が増えると通信時間が増えていることが分かる.

5　おわりに

測定とモデル化の結果から, 両者共に小粒子数,GPU 並列数大のときに, 通信時間の割合が大きくなることが確認出来た. 今後の展望として, ピーク時に対する効率に関するパラメータを導き, 最適な仮想 GPU の運用の為の指標を提案したい.

参考文献

[1] A. Kawai, K. Yasuoka, K. Yoshikawa, T. Narumi: “Distributed-Shared CUDA: Virtualization of Large-Scale GPU Systems for Programmability and Reliability”, The Fourth International Conference on Future Computational Technologies and Applications, FUTURE CONPUTING 2012, Nice, France, 2012.
[2] ELSA GeForce GTX 780: http://www.elsa-jp.co.jp/products/products-top/graphicsboard/geforce/ultra_high_end/geforce_gtx780/ (Last access:2015/05/13)
[3] CUDA Zone: https://developer.nvidia.com/cuda-zone (Last access:2015/05/13)
[4] ELSA VRIDGE X100 Dual16: http://www.elsa-jp.co.jp/html/products/pes/vridge_x100_dual16/index.htm (Last access:2015/05/13)
[5] Claret: htpp://atlas.riken.go.jp/~koishi/claret.html (Last access:2015/05/13)