

Windows マシンに仮想 CUDA 環境を与える PluginGPU の開発

電気通信大学 情報・通信工学科
コンピュータサイエンスコース 成見研究室
1011105 嶋田貴行

1 はじめに

元々画像処理目的で開発された GPU が、より一般的な目的で使われるようになってきている。GPGPU として知られるこのトレンドは研究用途での広まりが目立っていたが、動画像編集などのソフトウェアで利用されるようになり、最近になって、一般ユーザーへも浸透し始めている。また、並列コンピューティングが一般的な技術になってきており、教育の需要が高まることも予想される。しかし教育現場や家庭の中で、PC の台数分と同じだけの GPU を設置するコストは大きい。

そこで本研究では、GPGPU を行ないたいという要求に対して、複数の PC で GPU を共有することでコストを抑え、普段から使っている PC がそのまま GPGPU の実行開発環境として使えるような仕組みとして、PluginGPU を開発することを目標とした。

PluginGPU は LAN 上に設置するサーバーマシンであり、GPU を内蔵している。実行開発環境も搭載しており、同じネットワーク内に存在する PC からこの環境を利用して、GPGPU プログラムを開発・実行することができる。ルーターに挿す、すなわち「プラグイン」するだけで使えるという特徴を捉えて、名称を付けた。設置例を図 1 に示した。

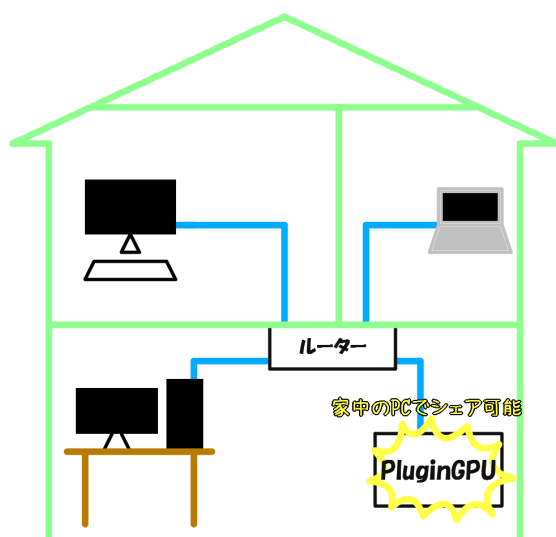


図 1 PluginGPU の設置例

2 既存技術

GPU を共有する技術には、NVIDIA 社の NVIDIA GRID[1] というものがある。GPU を搭載したクラウドマシン上でアプリケーションを実行し、その実行画面を映像として返送する形で GPU の仮想化を実現しているため、提供者によって定められたアプリケーションを高速に動作させることができる一方、自ら作成した GPGPU アプリケーションを実行するような使い方をすることはできない。

3 PluginGPU の特徴

3.1 Windows で使用可能

一般に普及する Windows マシンで使えるようにすることで、普段使っている PC の環境にすぐに GPGPU 環境を付与することができる。

3.2 GPGPU 環境として CUDA を提供

GPGPU 実行開発環境としては NVIDIA 社の提供する CUDA[2] を採用し、CUDA の仕様に沿う形で GPGPU 実行開発を行なうことができる。

3.3 新たにインストールするソフトウェアは不必要

動作に関わる要素を全て PluginGPU のサーバーに載せることで、ユーザーが自らの PC に新たに導入すべきソフトウェアなどはない。

4 PluginGPU の実装

4.1 GPU 実行環境の仮想化・共有化

成見研究室で開発している、GPU 仮想化ソフト「DS-CUDA[3]」を利用した。このソフトウェアは、NVIDIA 社の CUDA を用いて開発するソフトウェアをクラウド環境にあるサーバー上の GPU で実行させることができるツールで、GPU の仮想化・共有化を可能にするものである。しかし PluginGPU で実装する際に問題となった点が 2 つある。表 1 に示した両者の違いに起因する。

1 つ目の問題は、DS-CUDA が Windows に対応していなかった点である。そこで、Windows 上で動作する仮想 UNIX 環境 Cygwin を用いる事で、この問題をクリアした。Cygwin により、ソースコードに対して大きな変更を加えることなく動作させることが可能であった。

表 1 DS-CUDA と PluginGPU の仕様の違い

仕様	DS-CUDA	Plugin GPU
対応するクライアント OS	Linux 系・Android	Windows
クライアントマシンの CUDA 環境	コンパイル時に必要	不必要

2 つ目の問題は、一部の処理において CUDA のインストールが必要になってしまう点である。CUDA では GPU で動作させるコードを PTX コードと呼んで区別しているが、DS-CUDA でもこの PTX コードを使用しており、PTX コードを生成するためには CUDA に含まれる NVCC というコンパイラが必要になる。ユーザーマシンへの CUDA のインストールを省くために、サーバー上で PTX コードを作成して送りかえすというソフトウェアを作って対応した。

4.2 アプリケーション開発環境の共有化

DS-CUDA の実行環境と共に、アプリケーションの開発環境としても Cygwin を使うことができたため、これをサーバーに搭載し、ユーザーが使える状態にする必要があった。そこでファイルサーバー機能を利用して Cygwin を置くことで、それを利用したアプリケーション開発が可能になった。これにより、CUDA を含めた実行・開発両環境のインストールを省くことができる。

4.3 機能のパッキング

ここまでで PluginGPU に搭載した機能を、DVD からブート可能な OS に組み込んで 1 枚の DVD として発行した。これにより、GPU を搭載したマシンで DVD から OS をブートすることにより、わずかな設定のみで PluginGPU サーバーを構築することができるようになっている。

5 システム評価実験

現状のシステムで、GPGPU 環境としてどの程度の性能があり、どのようなアプリケーションに対して有用なシステムであるかを考察するための実行速度評価実験を全部で 4 つ行なったがここでは最終実験のみ紹介する。速度は、CPU、リモート GPU (PluginGPU)、ローカル GPU の 3 つで比較した。

5.1 実験内容

二つの数字の組み合わせからチェック模様の画像を生成するプログラムを作成した。計算量を増やすために、生成処理を数回繰り返してから画像を出力する仕様になっている。この実験では図 2 に示したように、繰り返しの回数を 100 回以上に増やした場合でほぼローカル GPU と同じような性能を得られた。10 回程度では画像の出力の際にサーバーから転送する時間が重みを持つため少し遅延があるが、CPU

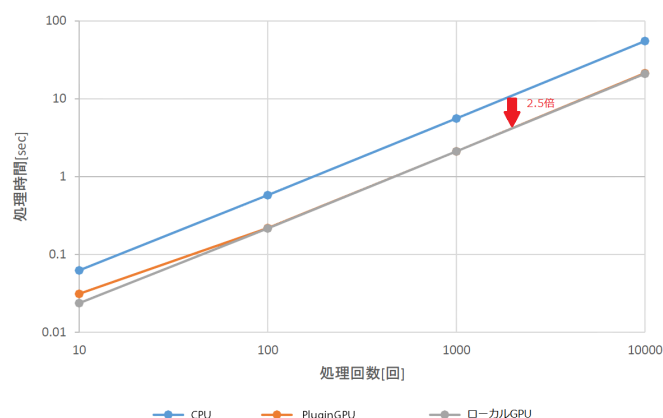


図 2 実験の結果

に比べて 2.5 倍程度の性能上昇があった。全実験を通して、ネットワークを使う以上大容量データの扱いが苦手であることはわかったが、一方 GPGPU に向けた処理でなければ速度の向上が見込まれないため、教育用ツールとしては一定の意味があるようにも思えた。

6 まとめ

6.1 達成できた点

Windows マシンで新たなインストールを行わずに GPGPU を行えるという目標は達成することができた。また、条件次第で PluginGPU が一般用途にも有効であることも示せた。

6.2 今後の課題

プラグインするだけで GPGPU 環境が使える、ということにはまだ足りない部分があるため、その点を改良していく必要がある。さらに、DS-CUDA を今後一般ユーザー向けに使用する場合は考えると、データ転送に対する最適化を施していく必要がある。

参考文献

- [1] Stream Applications and Games On-Demand — NVIDIA GRID — NVIDIA
<http://www.nvidia.com/object/nvidia-grid.html>
- [2] Parallel Programming and Computing Platform — CUDA — NVIDIA — NVIDIA
http://www.nvidia.com/object/cuda_home_new.html
<http://www.yasuoka.mech.keio.ac.jp/cuda/>
- [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.