学生用 GPU 計算サーバの導入とパフォーマンス測定

総合情報基盤センター　講師　山下和也

総合情報基盤センターでは，学生向けの演習用として Web サーバを計算サーバと兼用して開放している。深層学習の普及に伴い TensorFlow 等の機械学習フレームワークの演習環境が必要とされてきている。しかし，現在運用している計算サーバや演習用端末では膨大な計算量を必要とする機械学習の演習に対応できなかった。そこで，新たに GPU 計算サーバを導入したので紹介する。また，パフォーマンス測定した結果を示す。

キーワード：教育用計算機システム，GPU，人工知能，機械学習，深層学習

１．はじめに

2016 年に Google DeepMind が開発した囲碁プログラムの AlphaGo が世界のトップ棋士に勝利した[1]。これを一つの契機として人工知能技術が注目されるようになり，現在は第 3 次人工知能ブームと呼ばれている。

人工知能を実現するための一つの手法に機械学習がある。機械学習には様々な手法があるが，現在よく用いられる手法の一つが深層学習である。深層学習は，学習に膨大な計算量を必要とする。深層学習の演算は主に行列演算であり, CPU よりも行列演算を高速に行うことができる GPU を用いることが多い。

総合情報基盤センターでは，学生向けの演習用として Web サーバを計算サーバと兼用して開放している[2]。しかし，GPU が実装されていないため, 計算量を必要とする TensorFlow[3] 等のような機械学習フレームワークを用いた演習を行うことが困難であった。

そこで, GPU 計算サーバを導入したので紹介する。また，パフォーマンス測定した結果を示す。

２．GPU 計算サーバの基本仕様

NVIDIA Tesla V100 32GB を搭載した GPU 計算サーバを導入した。表 1 に基本仕様を示す。利用資格・方法等については，総合情報基盤センターの Web ページ「学生用 GPU 計算サーバ（教育利用）」[4] に記載されている。

表 1　GPU 計算サーバの基本仕様

CPU	Intel Xeon Gold 5218 1 基
メモリ	64 GB 2933 RDIMM
GPU	NVIDIA Tesla V100 32 GB 1 基
OS	Ubuntu 18.04.4 LTS
Driver	440.64.00 / CUDA10.2

機械学習の演習のために必要と思われるソフトウェアをインストールした。2020 年 4 月現在，インストールされている主なソフトウェアとそのバージョンは以下の通りである。包括ライセンス契約している MATLAB [5] と科学計算などのライブラリが充実している Python を導入している。

- MATLAB R2019b
  - 包括ライセンスに含まれるツールボックス一式
- Python 3.6.9
  - Chainer 7.0.0
  - Keras 2.3.1
  - PyTorch 1.3.1
  - TensorFlow 1.14.0
  - Theano 1.0.4

３．パフォーマンス測定

GPU 計算サーバのパフォーマンスを測定した。測定手法は，MATLAB を用いた GPU パフォーマンスの測定[6], tf_cnn_benchmarks[7], Keras のサンプルコード (mnist_cnn.py) [8]を用いた。

3.1 MATLAB を用いたパフォーマンス測定

MATLAB を用いて，データ転送帯域，メモリ帯域，倍精度 FLOPS を測定した。

まず，関数 gpuArray を用いてローカルワークスペースの配列を GPU へ送信し，関数 gather を用いて GPU 上の配列をローカルワークスペースへ転送することでデータ転送帯域を測定した。配列サイズを2^x $(x = 14, 15, \dots, 28)$としたときのデータ転送帯域をそれぞれ 20 回測定して平均値を求めた。

図 1 に結果を示す。横軸は配列サイズ，縦軸はデータ転送帯域，エラーバーは標準誤差を表す。実線は GPU へのデータ転送帯域, 破線は GPU からのデータ転送帯域を表す。測定範囲内のピーク値は，GPU へのデータ転送帯域は配列サイズ2^{19}のとき平均 9.83 GB/s であり，GPU からのデータ転送帯域は配列サイズ2^{24}のとき平均 5.43 GB/s であった。

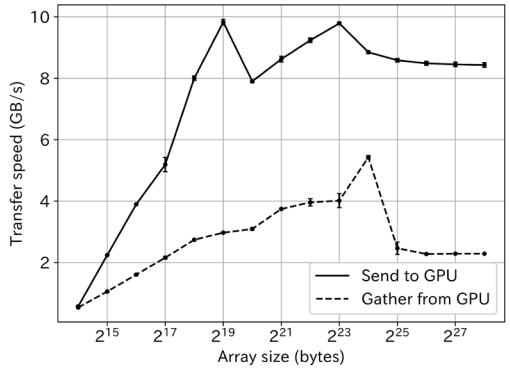

図 1　データ転送帯域と配列サイズの関係

次に, 関数 plus を用いて倍精度浮動小数点演算のメモリ書き込みと読み込みを行うことでメモリ帯域を測定した。配列サイズを 2^x $(x = 14, 15, \dots, 28)$としたときのメモリ帯域をそれぞれ 20 回測定して平均値を求めた。

図 2 に結果を示す。横軸は配列サイズ，縦軸はメモリ帯域，エラーバーは標準誤差を表す。実線は GPU のメモリ帯域，破線は CPU のメモリ帯域を表す。測定範囲内のピーク値は，GPU の場合は配列サイズ2^{28}のとき平均 792.03 GB/s であり，CPU の場合は配列サイズ2^{22}のとき平均 125.79 GB/s であった。

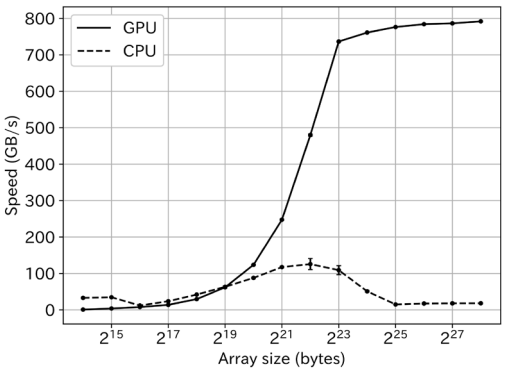

図 2　メモリ帯域と配列サイズの関係

最後に，行列の乗算の処理時間から倍精度 FLOPS を求めた。行列サイズを 2^x $(x = 12, 14, \dots, 26)$としたときの倍精度 FLOPS をそれぞれ 20 回測定して平均値を求めた。

図 3 に結果を示す。横軸は行列サイズ，縦軸は倍精度 FLOPS, エラーバーは標準誤差を表す。実線は GPU の倍精度 FLOPS，破線は CPU の倍精度 FLOPS を表す。行列サイズ2^{14}までは大きな差はないが, 行列サイズが大きくなるにつれて GPU の方が速く計算できることが分かる。

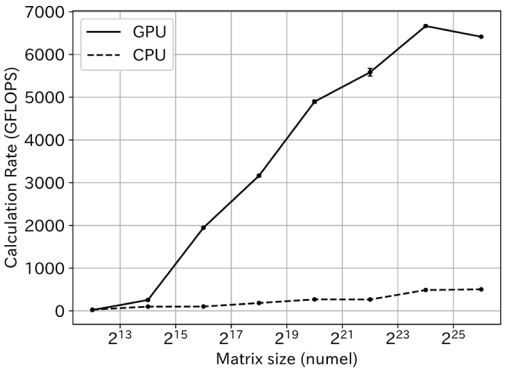

図 3　倍精度 FLOPS と行列サイズの関係

3.2 tf_cnn_benchmarks

tf_cnn_benchmarks は TensorFlow が公開しているベンチマークテストである。画像分類問題に対して畳み込みニューラルネットワークの学習モデルやバッチサイズ等を指定して実行すると，1 秒あたりに処理された画像枚数を出力する。

測定する学習モデルは，AlexNet, InceptionV3, ResNet50, VGG16 の 4 種類を用いた。バッチサ

イズには，32, 64 の 2 種類を用いた。これらの全 8 通りの組み合わせに対して, それぞれ 10 回ずつ測定して平均値を求めた。

比較対象は，TITAN RTX，TITAN V，GeForce RTX 2080Ti，GeForce RTX 2080，GeForce GTX 1080Ti とした。いずれも NVIDIA 社製の GPU である。比較対象の測定結果は文献[9]より引用した。いずれも GPU を 1 基用いた場合の結果であるが，システム構成が異なるため, GPU のみの比較ではないことに注意されたい。

図 4 は AlexNet，図 5 は InceptionV3，図 6 は ResNet50，図 7 は VGG16 をそれぞれ学習モデルに用いた結果を示す。Tesla V100 32GB が今回導入した GPU 計算サーバの測定結果である。縦軸が 1 秒あたりに処理された画像枚数であり，この値が大きいほど高速に処理できることを表している。どの学習モデルでも，今回導入した GPU 計算サーバが高速に処理できることが分かる。

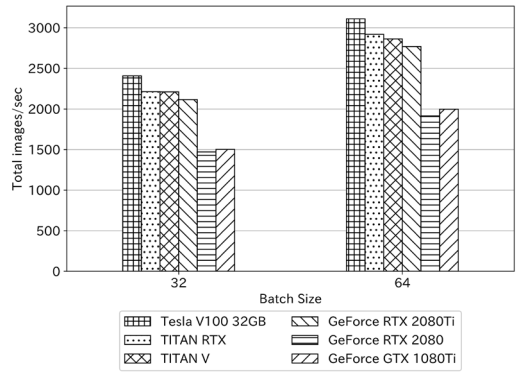

図 4　AlexNet の結果

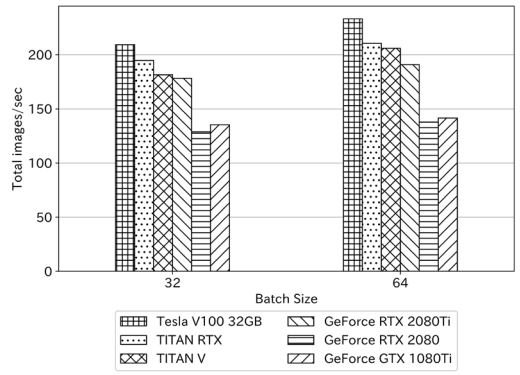

図 5　InceptionV3 の結果

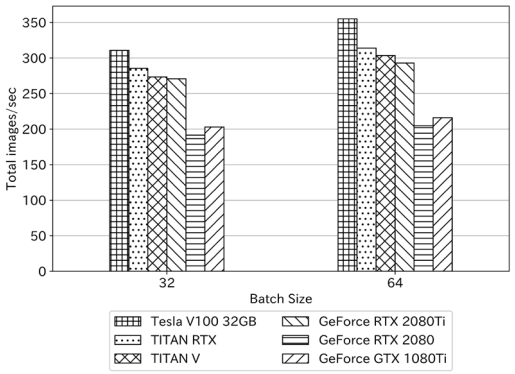

図 6　ResNet50 の結果

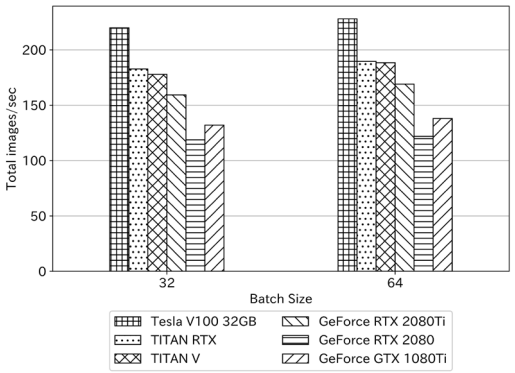

図 7　VGG16 の結果

3.3 Keras を用いた学習時間の測定

Keras のサンプルコードを用いて，機械学習の学習に掛かる時間を測定した。MNIST データセットに対して畳み込みニューラルネットワークを用いて画像分類するコードを用いた。

比較対象は, 演習用端末と計算サーバ兼 Web サーバとした。表 2 に演習用端末の基本仕様を示す。演習用端末には，Unix の演習環境として Hyper-V 上に Ubuntu の仮想マシンを作成している。表 3 に仮想マシンの設定を示す。表 4 に Web サーバの基本仕様を示す。

表 2　演習用端末の基本仕様

CPU	Intel Core i5-8500
メモリ	16 GB
GPU	Intel UHD Graphics 630
OS	Windows 10 Education 1903

表 3　仮想マシンの設定

CPU	仮想プロセッサ 2 個
メモリ	動的割り当て
OS	Ubuntu 18.04.4 LTS

表 4　Web サーバの基本仕様

CPU	仮想プロセッサ 4 個 （Intel Xeon Platinum 8168）
メモリ	7.5 GB
OS	Red Hat Enterprise Linux 7.7

12 epoch を 1 試行として 5 回測定し，1 epoch 当たりの平均値を求めた。

図 8 に結果を示す。図中の VM が演習用端末上の Ubuntu， WEB が Web サーバ，CPU が GPU 計算サーバで CPU のみを用いた場合, GPU が GPU 計算サーバで GPU を用いた場合の結果である。横軸は 1 epoch 当たりの学習時間の平均値，エラーバーは標準偏差である。演習用端末は平均 55.6 秒，Web サーバは平均 37.8 秒，GPU 計算サーバで CPU のみを用いた場合は平均 16.4 秒，GPU を用いた場合は平均 4.1 秒であった。

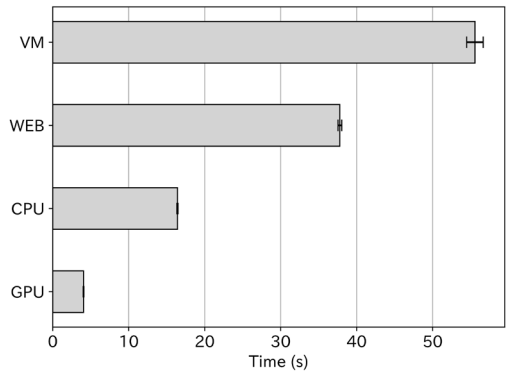

図 8　1 epoch 当たりの学習時間

4．おわりに

GPU 計算サーバを導入し, パフォーマンスを測定した。Keras を用いた学習時間の測定の結果から，演習用端末上で実行する場合に比べて，GPU 計算サーバで実行すると学習時間を約 1/14 に短縮できることが分かった。また，演習用として開放している Web サーバに比べても，学習時間を約 1/9 に短縮できることが分かった。仮に 10 epoch 学習させると，演習用端末で約 9 分，Web サーバでも約 6 分も掛かる。GPU 計算サーバであれば約 40 秒で終わるため，学生演習の進行に影響がない時間で結果が得られると考えられる。

複数人が同時利用した場合のパフォーマンスを測定できていないが, GPU 計算機を用いることでプログラム実行時間を短縮でき，演習を円滑に進行できると考えられる。

参考文献

[1] D. Silver et al., “Mastering the game of Go with deep neural networks and tree search”, nature, 529, 484-489 (2016).

[2] 富山大学総合情報基盤センター，“学生用計算サーバ（教育利用）”，https://www.itc.u-toyama.ac.jp/service/compute_server.html（参照 2020/03/23）.

[3] M. Abadi, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems”, https://www.tensorflow.org/ (2015)

[4] 富山大学総合情報基盤センター，“学生用 GPU 計算サーバ（教育利用）”，https://www.itc.u-toyama.ac.jp/service/gpu_server/gpu_server.html（参照 2020/05/08）.

[5] 富山大学総合情報基盤センター，“MATLAB 使用可能なライセンス”，https://www.itc.u-toyama.ac.jp/service/pdf/matlab.pdf（参照 2020/03/24）

[6] MathWorks，“GPU パフォーマンスの測定”，https://jp.mathworks.com/help/parallel-computing/examples/measuring-gpu-performance.html（参照 2020/03/23）.

[7] TensorFlow, “TensorFlow benchmarks”, https://github.com/tensorflow/benchmarks（参照 2020/03/23）.

[8] Keras, “Keras Documentation Mnist cnn”, https://keras.io/examples/mnist_cnn/（参照 2020/03/23）.

[9] HPC テクノロジーズ株式会社，“Tensorflow Benchmarks”，https://www.hpc-technologies.co.jp/tensorflow-benchmark-2（参照 2020/03/23）.