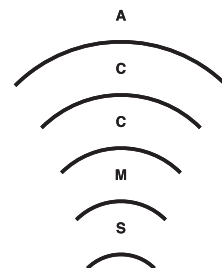


京都大学学術情報メディアセンター

Academic Center for Computing and Media Studies, Kyoto University

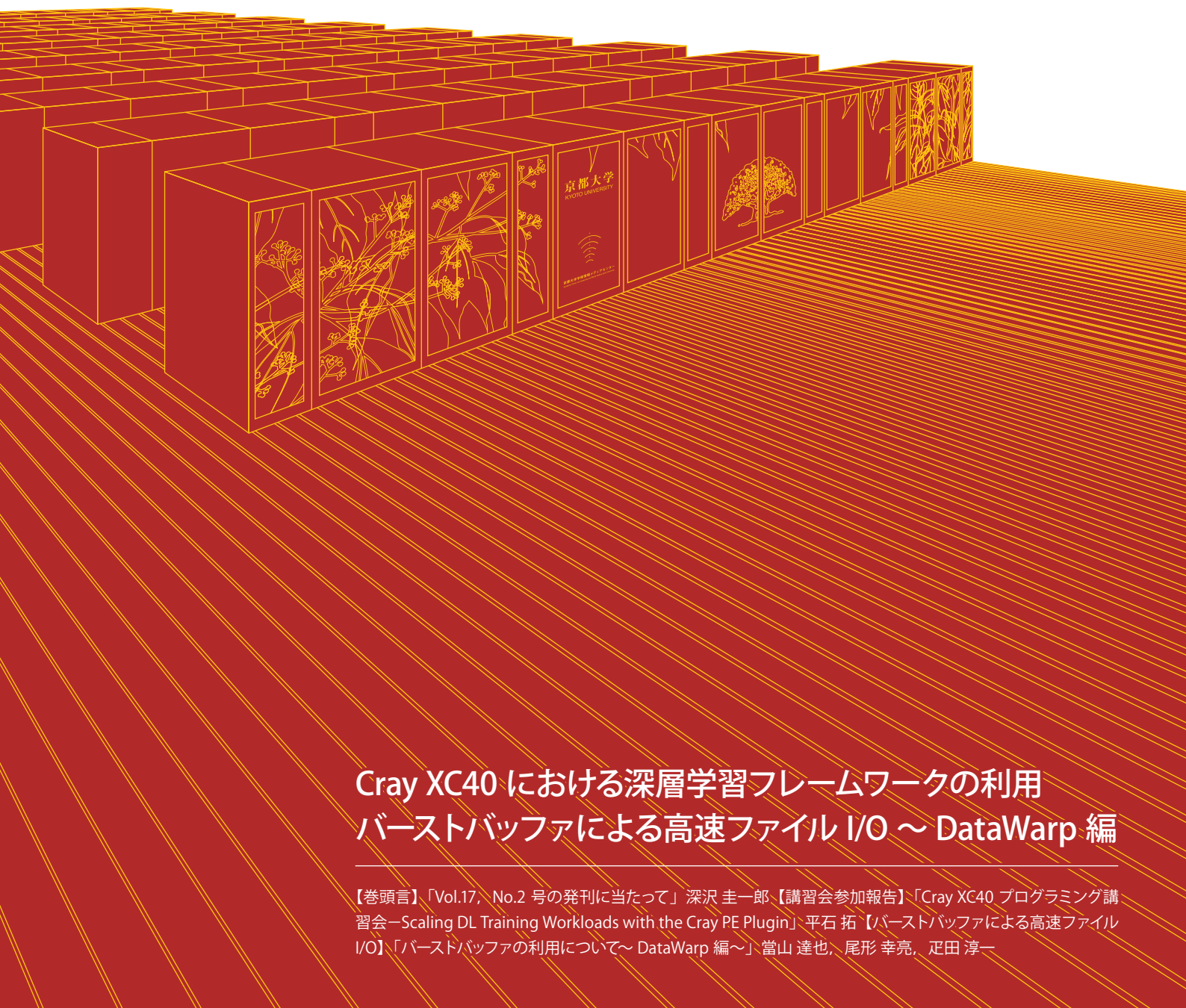


広報

全国共同利用版

Vol.17, No.2 2018

ISSN 1347-3581



Cray XC40 における深層学習フレームワークの利用 バーストバッファによる高速ファイル I/O ～ DataWarp 編

【巻頭言】「Vol.17, No.2 号の発刊に当たって」 深沢 圭一郎 【講習会参加報告】「Cray XC40 プログラミング講習会—Scaling DL Training Workloads with the Cray PE Plugin」 平石 拓 【バーストバッファによる高速ファイル I/O】「バーストバッファの利用について～ DataWarp 編～」 當山 達也, 尾形 幸亮, 疋田 淳一

巻頭言

Vol. 17, No.2 号の発刊に当たって

京都大学学術情報メディアセンター 深沢 圭一郎

近年、機械学習を代表にされるデータ科学が広がりを見せる中、様々な研究分野でデータ科学を活用するためこれまで以上に計算機が利用されてきています。これに伴い、スーパーコンピュータを利用して機械学習や統計解析を行うユーザも増えてきました。このようなユーザに対応するために当センターでもデータ科学分野で多く利用される Python や機械学習フレームワークなどをスーパーコンピュータシステムへ導入するなど環境整備を進めてきています。当センターのシステムは、いわゆる x86 アーキテクチャであり、パソコンの CPU と同じアーキテクチャになります。そのため、オープンソースソフトウェアなどの導入が比較的容易であり、多くのアプリケーションをスーパーコンピュータシステムへ導入してきています。更新情報を逐次メール等でお知らせはしていませんが、利用方法の Web ページ (<https://web.kudpc.kyoto-u.ac.jp/manual/>) は随時更新していますので、ご確認ください。また、希望されるアプリケーションも可能な限り導入いたしますので、オンラインプログラム相談室までお知らせください。

このような環境の整備だけでなく、利用方法を学べる機会として講習会も開催しています。本号では「Cray XC40 における深層学習フレームワークの利用」という特集として、2018 年 7 月に開催しましたノード間並列を可能とする Cray PE DL Scalability Plugin (DL Plugin) の利用講習会について紹介しています。この講習会では、ノード間並列実行が難しく、並列化効率が高くない機械学習フレームワークに対して、効率的な実装を可能とする DL Plugin について紹介し、またその利用方法を学んでももらいました。当センターは、多くの機械学習で利用される GPU が搭載された計算機が無く、機械学習における学習にあまり向いてはいません。しかし、複数ノードを利用し、並列に学習を行うことで、学習速度の向上が可能となり、元々の大容量メモリが利用できる面と合わせて、機械学習を行う計算機システムとして 1 つの選択肢になると考えられます。その講習の内容を簡単ですが特集としてまとめてありますので、ご参考にしてください。

特集では内容の紹介はしていませんが、センター教員が行う機械学習フレームワーク利用講習会も 2018 年 8 月に開催されました。来年度もこれら 2 つの講習会は開催する予定です。その他に開催希望の講習会などがありましたら、センターまでお知らせください。

また、本号では「DataWarp」と呼ばれるバーストバッファについても特集しています。バーストバッファについては Vol. 16 No.2 で紹介し、その中で DDN 社の IME と呼ばれるバーストバッファについて特集を組みました。同じバーストバッファである Cray 社の DataWarp もセンターで利用可能となり、その利用方法を紹介しています。IME と同様に恩恵を受ける I/O のタイプは限られますが、一度これらのバーストバッファを試していただき、効果のご確認を是非お願いいたします。

現在のスーパーコンピュータシステムが導入され約 2 年が経過しました。この短い間でも情報技術は大きく様変わりし、計算機に求められる計算内容も大きく変わってきました。前述の通り当センターのシステムは汎用的な CPU を搭載しているからこそ、できる限りこのような変化にも対応し、利用者のみなさまに役立つインフラであるよう努めてまいります。今後ともご利用、ご支援のほど、よろしくお願いいたします。

参加報告：Cray XC40 プログラミング講習会—Scaling DL Training Workloads with the Cray PE Plugin

平石 拓*

*京都大学学術情報メディアセンター

1 はじめに

近年、音声・画像処理のみならず医療や創薬、物理学、金融や教育にいたるまで、どの分野の話題をしても耳にするほど深層学習（ディープラーニング）は広く利用されるようになっていきます。本学のスーパーコンピュータも深層学習のために利用したい／しているという話をよく耳にするようになりました。とはいえ、深層学習のためのフレームワークである Tensorflow [1] や Chainer [3] をスーパーコンピュータの性能を生かす形で正しく設定して動かすのは、特にこれまでスパコンに触れることがなかったユーザにとってはやや敷居が高い面もあります。

このような利用者ニーズが高まるなか、本学でも今年度、スパコンのユーザ向けに深層学習フレームワークの利用に関する講習会を以下の通り実施しました。

- 2018 年 7 月 24 日：Cray XC40 プログラミング講習会—Scaling DL Training Workloads with the Cray PE Plugin¹
- 2018 年 8 月 27 日：機械学習フレームワーク利用講習会²

本稿は、このうち前者の講習会について、参加報告という形でその内容を紹介します。この講習会では、実際に本学のスパコンのサブシステム A（Cray XC40）上で機械学習のサンプルプログラムを実行する実習を行いました。読者がその実習内容を再現で

きるようにすることで、スパコン上で機械学習を実行するための一つの足掛かりとなることを目指しています。

2 Cray PE DL Scalability Plugin

本講習で利用した Cray PE DL Scalability Plugin（以下、DL Plugin）は、Cray のスパコン向けに提供されている深層学習向けのプラグインであり、サブシステム A にも導入されています。このプラグインを利用することで、Tensorflow による学習を多数の計算ノードを用いた並列実行で効率良く行うことができるようになります。利用にあたっては、モジュールのロードのほか、以下に述べていく通り学習用の（Python 等の）スクリプトにも少し変更を加える必要があります。

DL Plugin での分散並列処理（複数の計算ノードによる並列処理）は、ミニバッチを各プロセスに分配してそれぞれのプロセス内で勾配計算を行い、同期をとった後それらの結果を集約して重みを更新する、という処理により実現されています。この分配や縮約に必要となる通信処理等が Cray のスパコン向けにチューニングされているため、Horovod [2] などの一般的に知られている Tensorflow の分散並列処理フレームワークと比較して良好な性能が得られるとの主張がなされています。

3 サンプルスクリプト

それでは、実習の流れに沿って DL Plugin を利用した学習処理を実行していきましょう。なお、以下の操作は全て本学スーパーコンピュータのサブシステ

¹<http://www.iimc.kyoto-u.ac.jp/ja/whatsnew/event/detail/180628055104.html>（7 月 23 日から二日間にわたって開催された Cray XC40（サブシステム A）のプログラミングセミナーの一部）

²<http://www.iimc.kyoto-u.ac.jp/ja/whatsnew/event/detail/180528055051.html>

ム A にログインしたシェル上で行います (DL Plugin はサブシステム A にのみ導入されています)。まず、以下の手順で、実習で利用したサンプルプログラムとサンプルデータを取得します。

```
1 # 適当なディレクトリに移動
2 % cd
3 # サンプルが入った書庫ファイルをコピー&展開
4 % cp /opt/app/course/2018/x40/Kyoto_training.tar.gz
5 ./
6 % tar xvfz Kyoto_training.tar.gz
7 # 展開してできたディレクトリに移動
8 % cp Kyoto_training
```

ここで `ls` コマンドを実行すると、以下のファイルが確認できます。

```
1 % ls
2 CrayModifications      go.sh
3 README                 mnist.py
4 README.md              mnist_data/
5 __init__.py            mnist_plugin.py
6 convert_to_records.py  mnist_test.py
```

README に記載されている通り、このファイル形式は、GitHub で配布されている Tensorflow のサンプルスクリプト MNIST in Tensorflow³ をベースに、DL Plugin を用いて分散並列処理を行えるよう改変を加えたものになっています。MNIST⁴は、画像認識のサンプルとしてよく用いられる手書き数字画像のデータセットであり、MNIST in Tensorflow は、Tensorflow を用いた学習に基づいてこれらの画像を分類するためのスクリプトのサンプルです。

主なファイルの構成は以下の通りです。

- README.md: オリジナルの MNIST in Tensorflow の README.
- README: この改変版サンプルの README.
- mnist_data/: MNIST のデータセット.
- mnist.py: オリジナルの MNIST in Tensorflow のサンプルスクリプト.
- mnist_plugin.py: mnist.py に改変を加えて、DL Plugin を利用するようにしたもの.
- CrayModifications: mnist_plugin.py の mnist.py からの差分を説明したドキュメント.
- go.sh: 上記の Python スクリプトをスパコン上で実行するためのジョブスクリプトの雛形 (図 1)。

³<https://github.com/tensorflow/models/tree/master/official/mnist>

⁴<http://yann.lecun.com/exdb/mnist/>

```
1 #!/bin/bash
2
3 #QSUB -q ca
4 #QSUB -A p=1:c=62:t=62:m=10G
5 #QSUB -N mnist
6
7 export PYTHONUSERBASE=/opt/app/course/2018/x40/
8 tensorflow
9
10 module use /opt/app/course/2018/x40/pe/modulefiles
11 module load cray-python/2.7.13.1
12 module load craype-ml-plugin-py2
13
14 aprun -n 1 -N 1 -cc none -b python ./mnist.py --
15 enable_ml_comm --data_dir=$PWD/mnist_data/ --
16 model_dir=./train --train_epochs=1
```

図 1: go.sh

4 オリジナルの学習スクリプトの実行

まず、DL Plugin を使用しない `mnist.py` を実行してみます。基本的には図 1 のジョブスクリプトを `qsub` すればよいのですが、ご自身の利用可能なキューに合わせて、3 行目の `-q` のパラメータを変更し、さらにグループ名を `#QSUB -ug grnnnnn` の形で指定する行を追加する必要があります⁵ので注意してください。この書き換えを行った後、

```
1 % qsub go.sh
```

によりジョブを投入すると `mnist.py` が実行されます。ジョブが正常に完了すると、標準出力ファイル `mnist.ojjjjjj` および標準エラー出力ファイル `mnist.ejjjjjj` が生成されます (`jjjjjj` はジョブ ID)。このうち、標準出力ファイルの中身を確認しましょう。

```
1 % cat mnist.ojjjjjj
2 Evaluation results:
3   {'loss': 0.10947487, 'global_step': 600, '
4   accuracy': 0.9679}
5 Application 4820024 resources: utime ~10445s, stime
6 ~383s, Rss ~451108, inblocks ~4446, outblocks
7 ~156160
8
9 ...
10 resources_used.walltime = 00:03:47
11 ...
```

`global_step` が実施した学習のステップ数、`loss` が損失値 (小さいほどよい)、`accuracy` が分類の精度 (大きいほどよい) を示しています。`walltime` が示す通り、ジョブの実行時間は 4 分弱程度です。

なお、この状態からもう一度 `qsub go.sh` を実行し、結果を確認すると、

⁵詳しくは、利用者マニュアル <https://web.kudpc.kyoto-u.ac.jp/manual/ja/run/systema> をご確認ください

```

1 #!/bin/bash
2
3 #QSUB -q ca
4 #QSUB -A p=8:c=62:t=62:m=10G
5 #QSUB -N mnist_par
6
7 export PYTHONUSERBASE=/opt/app/course/2018/xc40/
8 tensorflow
9
10 module use /opt/app/course/2018/xc40/pe/modulefiles
11 module load cray-python/2.7.13.1
12 module load craype-ml-plugin-py2
13
14 aprun -n $QSUB_PROCS -N $QSUB_PPN -cc none -b python
15 ./mnist_plugin.py --enable_ml_comm --data_dir=$PWD/
16 mnist_data/ --model_dir=./train-par --train_epochs=1

```

図 2: go_par.sh (8 ノード実行)

```

1 % cat mnist.okkkkk
2 Evaluation results:
3   {'loss': 0.07666545, 'global_step': 1201, '
4     accuracy': 0.977}
5 Application 4820066 resources: utime ~10240s, stime
6   ~372s, Rss ~440136, inblocks ~4422, outblocks
7   ~156160
8 ...
9 resources_used.walltime = 00:03:43
10 ...

```

のように、`global_step` が追加され、分類の精度も向上していることが確認できます。これは、前回の学習結果の続きからさらに学習を行ったためです。学習結果は図 1 の 13 行目の `--model_dir` で指定したディレクトリ `./train` に蓄積されていきますので、最初から学習をやり直す場合はこのディレクトリを削除するか、このオプションで指定するディレクトリ名を変更してください。

5 DL Plugin を利用した学習スクリプトの実行

それでは、DL Plugin を利用して Tensorflow の分散並列実行を行いましょう。

`mnist_plugin.py` を複数ノードを使って走らせるためには、ジョブスクリプトを図 2 のように書き換える必要があります。 `go.sh` と同様 `-q` と `-ug` はご自身の利用可能キューに合わせて設定してください。図 1 の `go.sh` からの重要な変更点は、4 行目の MPI プロセス数の指定 (`p=8`)、および 13 行目の `aprun` の `-n` および `-N` の設定値⁶と実行スクリプト名

⁶なお、`-n` と `-N` の設定は `go.sh` でも図 2 の記述で問題ありません (図 1 は配付版のサンプルスクリプトに合わせているのでこのような記述になっています)。

(`./mnist_plugin.py`) です。それ以外に、必須の変更ではないですが、ジョブ名 (5 行目) および学習結果を保存するディレクトリ名 (15 行目の `--model_dir` の値) も変更しています。

変更版のジョブスクリプトを `go_par.sh` として保存したとすると、ジョブは先ほどと同様以下のように投入できます。

```

1 % qsub go_par.sh

```

ジョブが正常に完了すると、標準出力ファイル `mnist.par.o111111` および標準エラー出力ファイル `mnist.par.e111111` が生成されます (111111 はジョブ ID)。先ほどと同様、標準出力ファイルの中身を確認しましょう。

```

1 % cat mnist_par.e111111
2 ...
3 Evaluation results:
4   {'loss': 0.061739106, 'global_step': 600, '
5     accuracy': 0.9809}
6
7 Evaluation results:
8   {'loss': 0.061739106, 'global_step': 600, '
9     accuracy': 0.9809}
10
11 Evaluation results:
12   {'loss': 0.061739106, 'global_step': 600, '
13     accuracy': 0.9809}
14
15 Evaluation results:
16   {'loss': 0.061739106, 'global_step': 600, '
17     accuracy': 0.9809}
18
19 Evaluation results:
20   {'loss': 0.061739106, 'global_step': 600, '
21     accuracy': 0.9809}
22
23 Evaluation results:
24   {'loss': 0.061739106, 'global_step': 600, '
25     accuracy': 0.9809}
26 Application 4820196 resources: utime ~103318s, stime
27   ~3869s, Rss ~536364, inblocks ~36290, outblocks
28   ~1249600
29 ...
30 resources_used.walltime = 00:03:47
31 ...

```

ジョブの実行時間 (`walltime`) が `mnist.py` の時とほとんど変化がないことに注意してください。これは、先ほどと同じステップ数を 8 ノードを用いて実行したわけではなく、先ほどの 8 倍のステップ数を 8 ノードで並列実行したためです。実際、先ほどより小さい `loss` 値と大きい `accuracy` 値が得られていることが確認できます。また、8 倍の量の学習が先ほどとほぼ同じ計算時間で行えているので、ほぼ

理想的な性能向上が分散並列処理によって得られていることになります。

図2のMPIプロセス数は、利用するキューの利用資源量が許す限りもっと大きな値に変更してもかまいません。実際、 $p=8$ を $p=16$ に変更すると、以下のように、同じ所要時間でより良い学習結果が得られます。

```
1 # 最初から学習をやり直すため、先ほどの学習結果を別の場所に退避
2 % mv train-par train-par-p8
3
4 # 図2のp=8をp=16に書き換えたのち、以下を実行
5 % qsub go_par.sh
6
7 # ジョブ終了後、出力結果を確認 (nnnnnnnn はジョブ ID)
8 % cat mnist_par.annnnnnnn
9 ... (以下の Evaluation results は 16 回出力) ...
10 Evaluation results:
11     {'loss': 0.055892173, 'global_step': 600, '
12       accuracy': 0.982}
13 Application 4820290 resources: utime ~223132s, stime
14 ~8441s, Rss ~521048, inblocks ~74584, outblocks
15 ~2499328
16
17 ...
18         resources_used.walltime = 00:03:54
19 ...
```

なお、実際の講習会では、講習会スライド資料を参考にしながら `mnist.py` を `mnist_plugin.py` 相当に自分で書き換えるという実習を行ったのですが、本稿では完成後の `mnist_plugin.py` を直接使用するという流れで記載しました。スライド資料で説明された両者のスクリプトの差分については、`CrayModifications` にもほぼ同等の説明があり、これを参考にしつつ同様の書き換えを行うことで、MNIST 以外の学習スクリプトでも DL Plugin が利用可能だということです。

6 おわりに

本稿では、2018年7月に実施されたプログラミング講習会の内容を紹介する形で、スーパーコンピュータのサブシステム A 上で Cray PE DL Scalability Plugin を利用した機械学習の分散並列処理を実行する方法を紹介しました。ごく基本的な部分のみの紹介となりましたが、本稿が機械学習のためにスパコンを利用しようとされている方、あるいはすでに利用されている方の一助になれば幸いです。

謝辞

講習会の内容、および講習会で用いたサンプルコード等の本稿への掲載についてご快諾いただいたクレイ・ジャパン・インク社に感謝いたします。

参考文献

- [1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. and Zheng, X.: TensorFlow: A system for large-scale machine learning, *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283 (2016).
- [2] Sergeev, A. and Del Balso, M.: Horovod: fast and easy distributed deep learning in TensorFlow, *arXiv preprint arXiv:1802.05799* (2018).
- [3] Tokui, S. and Oono, K.: Chainer: a Next-Generation Open Source Framework for Deep Learning, *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, Vol. 5 (2015).

バーストバッファの利用について ～DataWarp 編～

當山 達也

尾形 幸亮

疋田 淳一

京都大学 企画・情報部

1 はじめに

京都大学 学術情報メディアセンター(以下、本センターという)が提供するスーパーコンピュータは、ストレージシステムとして DataDirect Networks 社の EXAScaler を採用しており、24PB の容量と転送性能 150GB/sec を実現しています。EXAScaler は並列ファイルシステムであることから、大きなサイズのファイルを多数の HDD に並列転送することによって高い性能を発揮する一方、ランダムアクセスや小さい単位での I/O については、やや苦手としています。この課題を改善するために、バーストバッファと呼ばれるランダムアクセスや小さい I/O に強い SSD を用いて構築されたストレージシステムを、図 1 のように配置しています。ストレージシステムへのファイルの読み込み及び書き込みの際に、バーストバッファがキャッシュ領域として機能することで、高速なファイルアクセスを実現します。

本センターのバーストバッファシステムとしては、2018 年 6 月からシステム A 向けに Cray 社の DataWarp を、システム B/C 向けには 2017 年 10 月から DataDirect Networks 社の IME を、それぞれ提供開始しています。

本稿では、システム A 向けに導入した DataWarp の特徴や利用方法について解説します。IME につきましては、本センターの全国共同利用広報 Vol16, No2 にて紹介しておりますのでそちらをご覧ください。

2 DataWarp

2.1 概要

DataWarp は、容量 230TB、転送性能 200GB/sec といった性能諸元であり、ストレージシステム (EXAScaler) と比較すると 1.3 倍の転送性能を実現しています。

バーストバッファ領域として使用可能な領域は、`/LARGE{0,1,2,3}` 及び `/home` です。DataWarp を有効にするには、ジョブスクリプトに `-bb` オプションを記載する必要があります。`-bb` オプションでは、引数としてバーストバッファに確保する一時ファイルの上限値(必須)ならびに、適用するディレクトリ(任意)を指定する事ができます。なお、DataWarp を適用するディレクトリを記載しない場合は、カレントディレクトリが DataWarp を適用するディレクトリとして自動的に設定され実行されます。プログラムの実行終了後、DataWarp にキャッシュされた書き込みデータは、ジョブ終了時に自動で `-bb` オプションで指定されたディレクトリ(指定しない場合はカレントディレクトリ)へ書き戻されます。

また、DataWarp は、ストレージとメタデータを共有していることから、`/LARGE{0,1,2,3}` 及び `/home` のファイルに対して透過的にアクセスすることが可能となっています。

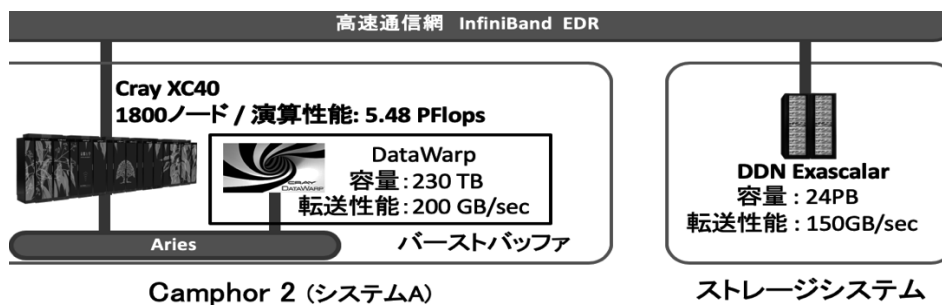


図 1 システム構成図

これにより、DataWarp を適用するディレクトリ配下のすべてのファイルに対して、パスの読み替えをすることなく書き込み/読み込みを行うことが可能となります。

DataWarp の利用には、第 3 章で解説する利用申請が必要となりますのでご注意ください。

2.2 利用イメージ

DataWarp の動作イメージを図 2 に示します。計算ノードで実行するプログラムの入出力先として DataWarp を利用し、一連の処理が終わった後にストレージに書き出すのが基本的な流れとなります。

DataWarp を利用する場合、プログラムの実行時間は、実行中に DataWarp を利用する時間とストレージへ書き出す時間の合計となりますが、DataWarp への I/O が高速に動作することで、実行時間を短縮できる可能性があります。しかし、全てのプログラムに効果があるわけではないので、ご自身のプログラムで性能比較を行い、有効性を確認して頂く必要があります。

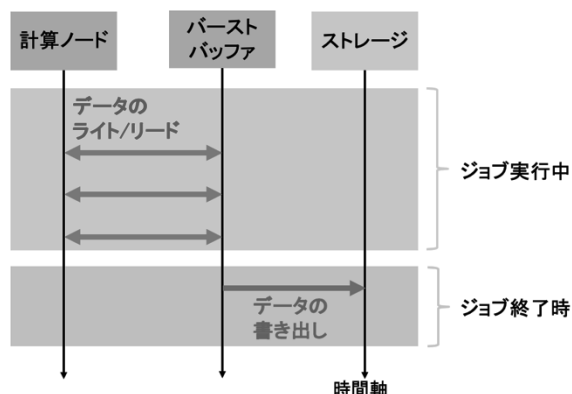


図 2 DataWarp の利用イメージ

2.3 利用方法

ジョブスクリプトに `-bb` オプションが記載されたジョブが投入されると、一時領域として確保する DataWarp の領域(`capacity` オプションの設定値)が、利用可能な上限値未満であることを確認した上で当該ジョブ専用の一時的領域を確保します。

`-bb` オプションの記法は下記のいずれかとなっており、各パラメータの意味は表 1 の通りです。

```
#QSUB -bb capacity=XXXX
または
#QSUB -bb capacity=XXXX:pfs=YYYY
```

表 1 `-bb` オプションのパラメータ

オプション	意味
<code>capacity</code> (必須)	バーストバッファに確保する一時ファイルの上限値。単位指定は G(ギガ)。 例) <code>-bb capacity=200G</code>
<code>pfs</code> (任意)	バーストバッファを適用するディレクトリ。書き込み権限のある既存のディレクトリのみ使用可能で、省略した場合はカレントディレクトリとなります。 例) <code>-bb pfs=/LARGE0/gr19999/</code>

DataWarp によって確保された一時領域は `/home` や `/LARGE{0,1,2,3}` とは異なった領域となるため、DataWarp 上に確保されたディレクトリパスでファイル操作をする必要があります。

また、DataWarp を使う設定としたジョブであっても、ジョブの開始時点では通常のジョブと同様に `/home` や `/LARGE{0,1,2,3}` 配下のパスでジョブスクリプトが実行されます。このため、DataWarp を用いて実行するジョブでは、`aprun` コマンドの後ろに `--wdir` オプションを記載する必要があります。

`--wdir` オプションの引数には DataWarp で一時的に確保されたディレクトリパスを指定する必要がありますが、可変であるため、表 2 に示す環境変数 `$QSUB_BB_DIR` をご利用ください。

表 2 `pfs` オプション設定時に登録される環境変数

環境変数名	変数に登録される内容
<code>\$QSUB_BB_DIR</code>	DataWarp 上に一時的に確保されたディレクトリパス※
<code>\$QSUB_BB_PFS_DIR</code>	<code>-bb</code> オプションの <code>pfs</code> で指定したディレクトリパス

※ `/var/opt/cray/dws/mounts/batch/
${PBS_JOBID}_striped_cache/`

2.3.1 ジョブスクリプト

DataWarp を有効にするジョブスクリプトの記載例を次に示します。この記載例ではバーストバッファの領域として、容量 100GB、対象ディレクトリに `/LARGE0/gr19999/dir` を指定しています。これにより、`/LARGE0/gr19999/dir` 配下のファイルを対象に

100GB 以内の範囲でキャッシュが行われます。

```
#!/bin/bash
#===== PBS Options =====
#QSUB -q gr19999a
#QSUB -ug gr19999
#QSUB -W 2:00
#QSUB -A p=8:t=4:c=4:m=1355M
#QSUB -bb capacity=100G:pfs=/LARGE0/gr19999/dir
#===== Shell Script =====
aprun -n $QSUB_PROCS -d $QSUB_THREADS ¥
-N $QSUB_PPN --wdir $QSUB_BB_DIR ./a.out
```

2.3.2 利用上の注意事項

DataWarp が有効なジョブが実行されると、ファイルの読み書きの際に DataWarp のディレクトリにファイルがキャッシュされます。DataWarp 上にファイルが存在する状態でストレージシステム側のファイルの変更を実施すると、DataWarp からの書き戻しの際に更新データが消失します。また、削除処理はメタデータを共有している都合上、DataWarp で適切に処理できなくなりますので、DataWarp をご利用中のディレクトリ配下に対して原則変更を行わないでください。

DataWarp を有効にしたジョブを複数同時に実行することは可能ですが、複数のジョブで同じディレクトリを DataWarp の対象とした場合、ファイルの整合性が保てない場合があります。原則、別々のディレクトリに分けてご利用ください。

DataWarp の対象ディレクトリの親ディレクトリは、DataWarp による透過的なアクセスの範囲外となります。環境変数 \$QSUB_BB_PFS_DIR や絶対パスを使用してアクセスしてください。

絶対パスにより /home や /LARGE{0,1,2,3} のファイルにアクセスする場合、DataWarp のキャッシュ機能は働きません。その場合は相対パスにより DataWarp 経由でファイルアクセスするようにプログラム等の改修をご検討ください。

ジョブが終了すると、自動的に DataWarp 上のキャッシュデータはすべて指定された領域に書き戻され、DataWarp のキャッシュデータは削除されます。ストレージシステム側の容量制限等でエラーとなった場合、データが破壊または消失する場合がありますので、残容量にご注意下さい。

DataWarp のメタデータは、/home や /LARGE{0,1,2,3} と共有されています。このため、新規ファ

イルを書き込む際は、メタデータをストレージ側にも作成します。ただし、ジョブ終了まで作成されたファイルにデータは書き込まれないため、ジョブが終了し書き戻しが行われるまでの間の容量はゼロとなります。

DataWarp 領域の容量が不足した場合、ジョブ実行中であっても自動的にファイルの書き戻しが行われます。確保した容量が小さすぎると、性能が出にくい場合がありますのでご注意ください。

2.3.3 利用可能な容量の上限

-bb オプションの capacity で設定できる容量は、サービスコースごとに異なり、表 3 に記載した上限値が設定されています。

表 3 設定できる capacity の上限

サービスコース	capacity の上限
パーソナル	800GB
グループ	200GB x 申請ノード数

表 3 に記載された上限以上の値をジョブスクリプトに設定しても、ジョブ投入時にエラーとなりますので、ご注意ください。また、DataWarp の利用申請をしていない場合もエラーとなります。

3 利用申請

DataWarp をお使いいただくためには、利用申請が必要です。申請資格のある方は、システム A のパーソナルコースおよびグループコースの申請者、またはグループキューの管理者の方となっています。

利用申請される場合は、下記の URL に記載の方法で申請をお願いします。

<https://web.kudpc.kyoto-u.ac.jp/manual/ja/filesystem/datawarp#application>

4 おわりに

本稿では、DataWarp の利用に必要なとなる情報を解説しました。最新の情報は以下 Web ページをご確認ください。

- スーパーコンピュータの使い方

<http://web.kudpc.kyoto-u.ac.jp/manual/ja>

システム A 運転状況 (2018 年 4 月 ~ 2018 年 9 月)

1) 保守作業に伴うサービス休止およびシステムダウン障害発生状況

保守作業に伴うサービス休止

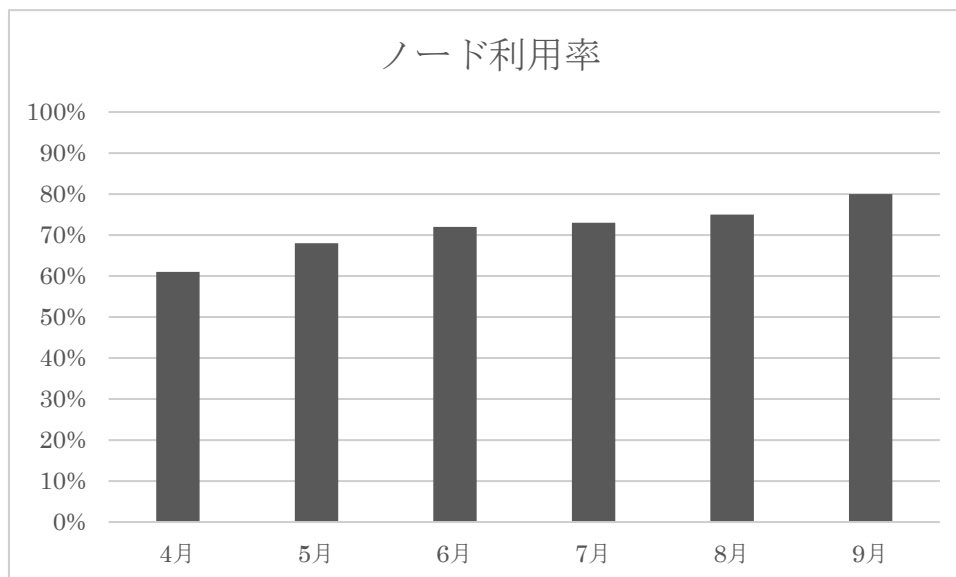
保守開始日時	サービス再開日時	保守時間[h]
2018/04/01 0:00	2018/04/05 13:00	109.00
2018/06/05 9:00	2018/06/06 9:30	24.50
2018/08/07 9:00	2018/08/07 17:00	8.00

システムダウン障害発生状況

障害発生日時	サービス再開日時	ダウン時間[h]
2018/09/15 10:49	2018/09/15 16:08	5.32

2) サービス状況

	サービス時間[h]	ジョブ					
		処理件数	経過時間[h]	占有時間[h]	CPU時間[h]	平均稼動ノード数	ノード利用率
4 月	611.00	21,273	105,080	14,082,616	52,027,239	1781.8	61 %
5 月	744.00	21,398	104,093	15,476,626	55,651,343	1799.7	68 %
6 月	695.50	32,819	110,266	14,867,306	60,965,186	1799.5	72 %
7 月	744.00	29,797	110,062	14,675,751	58,786,401	1755.5	73 %
8 月	736.00	36,835	180,502	15,155,928	63,273,894	1699.4	75 %
9 月	720.00	21,406	136,714	66,926,400	46,066,324	1756.6	80 %
計	4250.50	163,528	746,717	141,184,627	336,770,387	1765.4	72 %



- 占有時間 = 合計(経過時間×占有コア数)
- 平均稼動ノード数 = 電源 ON 状態のノード数の月平均 (10 分間隔のサンプリングデータより算出)
- ノード利用率 = 稼動ノードに対するジョブが実行されているノードの割合

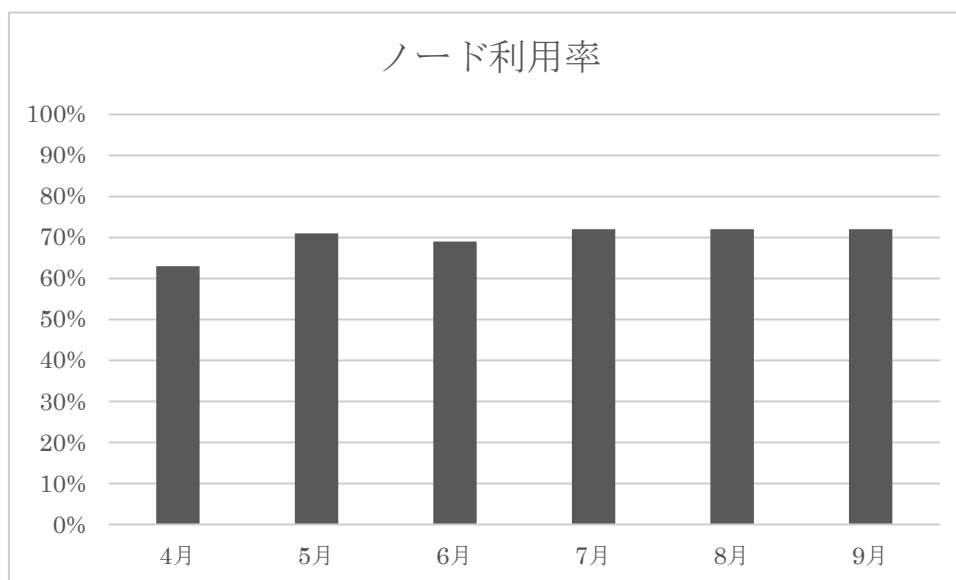
システム B 運転状況 (2018 年 4 月 ~ 2018 年 9 月)

1) 保守作業に伴うサービス休止およびシステムダウン障害発生状況

保守作業に伴うサービス休止			システムダウン障害発生状況		
保守開始日時	サービス再開日時	保守時間[h]	障害発生日時	サービス再開日時	ダウン時間[h]
2018/04/01 0:00	2018/04/03 9:30	57.50	2018/06/08 22:14	2018/06/08 23:19	1.08
2018/06/05 9:00	2018/06/06 9:30	24.50	2018/07/20 19:06	2018/07/21 01:44	6.63
2018/08/07 9:00	2018/08/07 17:00	8.00	2018/07/30 7:00	2018/07/30 9:25	2.42
			2018/09/20 7:00	2018/09/20 7:08	0.13

2) サービス状況

	サービス時間[h]	ジョブ					
		処理件数	経過時間[h]	占有時間[h]	CPU時間[h]	平均稼働ノード数	ノード利用率
4 月	662.50	76,687	545,830	12,396,959	9,097,320	833.5	63 %
5 月	744.00	92,325	601,336	15,083,286	11,726,252	840.0	71 %
6 月	694.42	67,449	630,402	13,581,340	11,134,697	829.2	69 %
7 月	734.95	94,033	663,871	15,058,021	11,896,726	837.7	72 %
8 月	736.00	72,252	666,835	15,169,631	11,237,561	839.8	72 %
9 月	720.00	126,281	620,865	14,735,169	12,042,714	839.0	72 %
計	4286.32	529,027	3,729,139	86,024,406	67,135,270	837.0	70 %



- 占有時間 = 合計(経過時間×占有コア数)
- 平均稼働ノード数 = 電源 ON 状態のノード数の月平均 (10 分間隔のサンプリングデータより算出)
- ノード利用率 = 稼働ノードに対するジョブが実行されているノードの割合

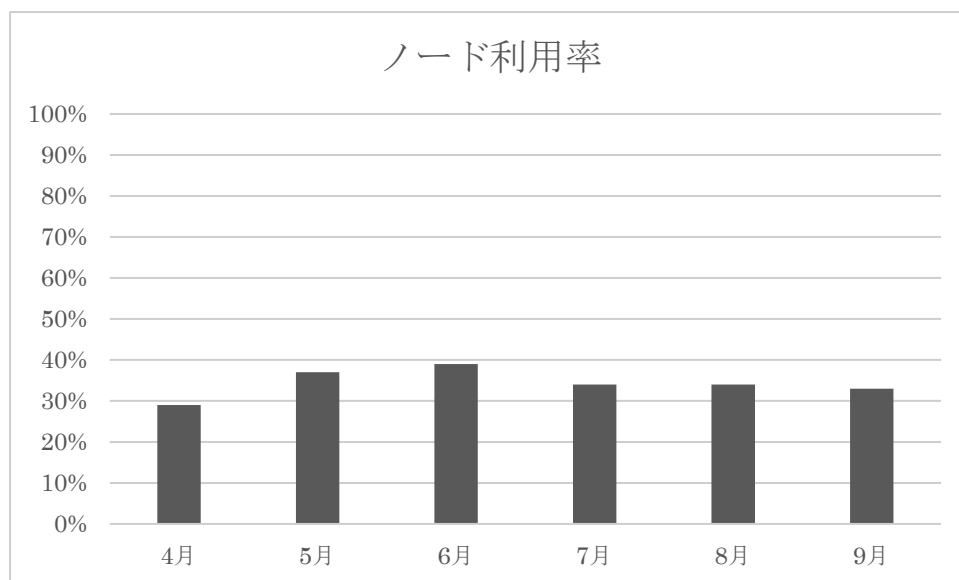
システム C 運転状況 (2018 年 4 月 ~ 2018 年 9 月)

1) 保守作業に伴うサービス休止およびシステムダウン障害発生状況

保守作業に伴うサービス休止			システムダウン障害発生状況		
保守開始日時	サービス再開日時	保守時間[h]	障害発生日時	サービス再開日時	ダウン時間[h]
2018/04/01 0:00	2018/04/03 9:30	57.50	2018/07/30 7:00	2018/07/30 9:25	2.42
2018/06/05 9:00	2018/06/06 9:30	24.50			
2018/08/07 9:00	2018/08/07 17:00	8.00			

2) サービス状況

	サービス時間[h]	バッチ					
		処理件数	経過時間[h]	占有時間[h]	CPU 時間[h]	平均稼動ノード数	ノード利用率
4 月	662.50	8,251	12,439	162,910	120,276	15.5	29 %
5 月	744.00	28,834	30,472	211,115	178,028	16.0	37 %
6 月	695.50	34,954	35,752	213,615	202,932	15.8	39 %
7 月	741.58	6,340	17,875	196,105	176,028	16.0	34 %
8 月	736.00	2,744	18,780	196,828	145,472	15.9	34 %
9 月	720.00	4,117	19,516	189,405	177,190	15.9	33 %
計	4,299.58	85,240	134,834	1,169,978	999,926	15.9	34 %



- 占有時間 = 合計(経過時間×占有コア数)
- 平均稼動ノード数 = 電源 ON 状態のノード数の月平均 (10 分間隔のサンプリングデータより算出)
- ノード利用率 = 稼動ノードに対するジョブが実行されているノードの割合

センター利用による研究成果

—平成 29 年度計算機利用結果報告書から—

【1501 生活科学一般】

1. A. Sharma, W. M. C. Sameera, M. Jin, Masayoshi, L. Adak, C. Chiemi, T. Iwamoto, M. Kato, M. Nakamura, K. Morokuma : DFT and AFIR Study on the Mechanism and the Origin of Enantioselectivity in Iron-Catalyzed Cross-Coupling Reactions : J. Am. Chem. Soc, Vol.139, No.45, pp.16117 - 16125, 2017
2. M. Isegawa, W. M. C. Sameera, A. K. Sharma, T. Kitanoosono, M. Kato, S. Kobayashi, K. Morokuma : Copper-catalyzed Enantioselective Boron Conjugate Addition: DFT and AFIR Study on Different Selectivities of Cu(I) and Cu(II) Catalysts : ACS Catal., Vol.7, No.8, pp.5370 - 5380, 2017

【2202 自然災害科学】

3. Asano, K., T. Iwata, H. Sekiguchi, K. Somei., K. Miyakoshi, S. Aoi, and T. Kunugi : Surface wave group velocity in the Osaka sedimentary basin, Japan, estimated using ambient noise cross-correlation functions : Earth, Planets and Space, Vol.69, No.108, pp.1 - 20, 2017

【2301 基礎ゲノム科学】

4. Bowman J.L, Kohchi T., Yamato K.T, et al. : Insights into Land Plant Evolution Garnered from the Marchantia polymorpha Genome : Cell, Vol.171, No.2, pp.287 - 304, 2017

【4104 基礎解析学】

5. 岸本 展 , 齊木 吉隆, 中井 拳吾 , 米田 剛: 粘性項を変形した流体方程式に対する大域解の有界性と Energy Cascade の考察: 第 14 回数学総合若手研究集会 : 数学の交叉点, Vol.173, pp.403 - 406, 2018

【4304 数理物理・物性基礎】

6. Mitsuru Tanaka : Effect of gravity on the development of homogeneous shear turbulence

laden with finite-size particles : Journal of Turbulence, Vol.18, No.12, pp.1144 - 1179, 2017

【4403 超高層物理学】

7. Kazem Ardaneh, Dongsheng Cai, and Ken-Ichi Nishikawa : COLLISIONLESS ELECTRON-ION SHOCKS IN RELATIVISTIC UNMAGNETIZED JET-AMBIENT INTERACTIONS: NON-THERMAL ELECTRON INJECTION BY DOUBLE LAYER : The Astrophysical Journal, Vol.827, pp.124 - 15pp, 2016

【4801 機能材料・デバイス】

8. Tomohiro Higashino, Shimpei Nimura, Kenichi Sugiura, Yuma Kurumisawa, Yukihiro Tsuji, Hiroshi Imahori : Photovoltaic Properties and Long-Term Durability of Porphyrin-Sensitized Solar Cells with Silicon-Based Anchoring Groups : ACS Omega, Vol.2, pp.6958 - 6967, 2017

【5001 機械材料・材料力学】

9. Masaaki Nishikawa, Keisuke Nagata, Yuta Naito, Masaki Hojo, Naoki Matsuda : Development of dynamic deployment simulation of thin composite layer for shape morphing structure : Proc. of 21st International Conference on Composite Materials (ICCM-21), 2017
10. 西川雅章, 高橋奈緒子, 西正人, 王俊翔, 平島禎, 北條正樹, 松田直樹: 炭素繊維織物の賦形時の変形に関する実験と有限要素モデリング: 2017 年度 JCOM 若手シンポジウム 講演論文集, 2017
11. 西川 雅章: 局所繊維配向を有する複合材料の損傷発生に関する Peridynamics 解析法の検討: 日本機械学会 第30回計算力学講演会 講演論文集, 2017

12. M. Nishikawa, N. Matsuda, M. Hojo : Modeling of Multiple-Type, Multiple-Site Damage in Composite Laminates Using Peridynamics Theory: Proc. American Society for Composites (ASC)?32nd Annual Technical Conference, 2017
13. 西川雅章, 高橋奈緒子, 松田直樹, 北條正樹: 炭素繊維織物の賦形プロセス後の繊維配向と欠陥発生に関する検討: 第55回飛行機シンポジウム 講演論文集, 2017
- 【5004 流体工学】
14. Yong Hu, Ryoichi Kurose : Nonpremixed and premixed flamelets LES of partially premixed spray flames using a two-phase transport equation of progress variable : Combustion and Flame, Vol.188, pp.227 - 242, 2018
15. Rudra N. Roy, Masaya Muto, Ryoichi Kurose : Direct numerical simulation of ignition of syngas (H₂/CO) mixtures with temperature and composition stratifications relevant to HCCI conditions : International Journal of Hydrogen Energy, Vol.42, pp.26152 - 26161, 2017
- 【5202 構造工学・地震工学・維持管理工学】
16. 山下将一, 富山 潤, 阿部和久, 佐伯竜彦, 紅露一寛: 数値シミュレーションと逆解析によるコンクリート橋の付着塩分量推定: 土木学会第72回年次学術講演会, No.V-434, pp.867 - 868, 2017
17. 山下将一, 富山 潤, 阿部和久, 佐伯竜彦, 紅露一寛: 風向を考慮した3次元数値解析によるコンクリート橋の付着塩分量推定: コンクリート構造物の補修, 補強, アップグレード論文報告集, Vol.17, pp.609 - 614, 2017
18. 野口恭平, 金城佑紀, 秦聡一郎, 白土博通, 八木知己, 中西克佳: 任意地点橋梁における部位別の飛来塩分量予測: 土木学会論文集 A1 (構造・地震工学), Vol.73, No.2, pp.364 - 375, 2017
19. Kyohei Noguchi, Hiromichi Shirato, Tomomi Yagi : Numerical evaluation of sea salt amounts deposited on bridge girders : Journal of Bridge Engineering, Vol.22, No.7, 2017
- 【5204 水工水理学】
20. 鳥生大祐, 牛島省: 異なる物性値を有する圧縮性流体と固体の熱連成問題に対する多相場解析手法: 土木学会第20回応用力学シンポジウム 講演概要集, Vol., No.C000053, pp., 2017
21. 榊原幸星, 鳥生大祐, 牛島省: 流体と温度依存性のある超弾性体との連成計算: 日本流体力学会 年会 2017, 2017
22. S. Ushijima, D. Toriu, K. Aoki, H. Itada and D. Yagyu : Numerical prediction for many floating debris transported in city model due to tsunami-induced flows : VII International Conference on Computational Methods in Marine Engineering (MARINE2017), 2017
23. H. Yanagi, S. Ikeda, D. Toriu and S. Ushijima : Computations of Density Currents by Numerical Models with and without Boussinesq Approximations as Well as by Compressible-Fluid Model : Compsafe2017, No.S16110, 2017
24. S. Ushijima and D. Toriu : Parallel computation method for momentum and heat interactions between fluids and complicated-shaped solids : ICHMT International Symposium on Advances in Computational Heat Transfer (CHT-17), No.CHT-17-254, 2017
25. 鳥生大祐, 牛島省: 混合体モデルに基づく圧縮性流体と移動する固体の熱連成計算手法: 土木学会論文集 A2 (応用力学), Vol.73, No.2, pp.I_143 - I_152, 2017
- 【5401 金属物性】
26. L. A. Burton, Y. Kumagai, A. Walsh, and F. Oba : DFT investigation into the underperformance of sulfide materials in photovoltaic applications : Journal of Materials Chemistry A, Vol.5, No.19, pp.9132 - 9140, 2017
27. Y. Hinuma, H. Hayashi, Y. Kumagai, I. Tanaka, and F. Oba : Comparison of approximations in density functional theory calculations: Energetics and structure of binary oxides : Physical Review B, Vol.96, No.9, pp.094102-1 - 094102-24, 2017
28. Y. Kumagai, K. Harada, H. Akamatsu, K. Matsuzaki, and F. Oba : Carrier-induced band-gap variation and point defects in Zn₃N₂ from first principles: Physical Review Applied,

- Vol.8, No.1, pp.014015-1 - 014015-12, 2017
29. Y. Kumagai, N. Tsunoda, and F. Oba : Point defects and p-type doping in ScN from first principles : *Physical Review Applied*, Vol.9, No.3, pp.034019-1 - 034019-10, 2018
 - 【5403 複合材料・物性】
 30. S. Iwata : Molecular dynamics simulation of effect of glycerol monostearate on amorphous polyethylene in the presence of water : *Journal of Molecular Modeling*, Vol.23, No., pp.115(1) - 115(7), 2017
 31. S. Iwata : Interaction between 1-phenylethanone, 2-phenyl-2-propanol, and isopropenylbenzene with water molecules: A computational study : *Computational and Theoretical Chemistry*, Vol.1117, pp.188 - 195, 2017
 32. S. Iwata, H. Uehara, T. Takada : Computational study on acetophenone in amorphous polyethylene : *Journal of Molecular Modeling*, Vol.23, pp.274(1) - 274(9), 2017
 - 【5501 化工物性・移動操作・単位操作】
 33. (阪大院基工) 矢野雅貴・山本卓也・岡野泰則・(藤森工業) 都倉知浩・(産総研) 金森敏幸・(阪大院工) 紀ノ岡正博 : 懸濁培養最適化のための数値流体解析・離散要素法連成シミュレーション : 化学工学会 第 82 年会, 2017
 34. M. Yano, T. Yamamoto, Y. Okano, T. Kanamori, and M. Kino-oka : Numerical study of fluid dynamics in suspension culture of iPS cell : *Asean Journal of Chemical Engineering*, Vol.17, No.1, pp.73 - 81, 2017
 35. 堀内鷹之・山本卓也・岡野泰則・宇治原徹 : 誘導加熱 TSSG 法による SiC 結晶成長時の面内成長速度分布に及ぼす各種因子の影響に関する数値解析 : 化学工学会第 4 9 回秋季大会要旨集, No.DA118, 2017
 36. 堀内鷹之・WANG Lei・山本卓也・宇治原徹・岡野泰則 : 数値解析を用いた誘導加熱 TSSG 法による SiC 結晶成長製造装置の最適化条件の探索 : JCCG-46 予稿集, No.28p-P32, P33 - P33, 2017
 37. 堀内鷹之・WANG Lei・山本卓也・宇治原徹・岡野泰則 : 誘導加熱 TSSG 法による SiC 結晶作製時の各種成長条件の影響に関する数値解析 : 化学工学会金沢大会 2 0 1 7 予稿集, No.D106, 2017
 38. 堀内鷹之・WANG Lei・関本敦・岡野泰則・宇治原徹 : SiC 溶液成長におけるるつぼサイズ及びるつぼ回転の影響に関する数値解析 : 化学工学会第 8 3 回年会要旨集, No.PD305, 2018
 - 【6801 化学系薬学】
 39. Taryn, M.; Murata, A.; Kobayashi, Y.; Takemoto, Y. : Enantioselective Synthesis of anti- β -Hydroxy- α -amino Esters via an Organocatalyzed Decarboxylative Aldol Reaction : *Synlett*, Vol.28, No.11, pp.1295 - 1299, 2017
 40. Saito, M.; Kobayashi, Y.; Tsuzuki, S.; Takemoto, Y. : Electrophilic Activation of Iodonium Ylides by Halogen-Bond-Donor Catalysis for Cross-Enolate Coupling : *Angew. Chem. Int. Ed.*, Vol.56, No., pp.7653 - 7657, 2017
 41. Kobayashi, Y.; Masakado, S.; Takemoto, Y. : Photoactivated N-Acyliminoiodinanes Applied to Amination: an ortho-Methoxymethyl Group Stabilizes Reactive Precursors : *Angew. Chem. Int. Ed*, Vol.57, pp.693 - 697, 2017

別表1 スーパーコンピュータシステム

			利用負担額	提供サービス						
コース	タイプ	セット		システム	パッチ	システム資源	経過時間 (時間)	ストレージ (TB)	無料 利用者数	
エントリ	-	基本	12,600 円/年	B	共有	最大1ノード相当((36コア、128GBメモリ)×1)	1	0.2	-	
パーソナル	タイプA	基本	100,000 円/年	A	共有	最大4ノード相当((68コア、16+96GBメモリ)×4)	168	3.0	-	
	タイプB	基本	100,000 円/年	B	共有	最大4ノード相当((36コア、128GBメモリ)×4)		3.0		
	タイプC	基本	100,000 円/年	C	共有	最大1ノード相当((72コア、3072GBメモリ)×1)		3.0		
グループ	タイプA1	最小	200,000 円/年	A	優先	4ノード((68コア、16+96GBメモリ)×4)	336	24.0	8	
		追加単位	100,000 円/年			2ノード((68コア、16+96GBメモリ)×2)		12.0	4	
	タイプA2	最小	240,000 円/年		準優先	8ノード((68コア、16+96GBメモリ)×8)		28.8	16	
		追加単位	60,000 円/年			2ノード((68コア、16+96GBメモリ)×2)		7.2	4	
	タイプA3	最小	600,000 円/年		占有	8ノード((68コア、16+96GBメモリ)×8)		48.0	16	
		追加単位	300,000 円/年			4ノード((68コア、16+96GBメモリ)×4)		24.0	8	
	タイプB1	最小	210,000 円/年	B	優先	4ノード((36コア、128GBメモリ)×4)		24.0	8	
		追加単位	105,000 円/年			2ノード((36コア、128GBメモリ)×2)		12.0	4	
	タイプB2	最小	252,000 円/年		準優先	8ノード((36コア、128GBメモリ)×8)		28.8	16	
		追加単位	63,000 円/年			2ノード((36コア、128GBメモリ)×2)		7.2	4	
	タイプB3	最小	630,000 円/年		占有	8ノード((36コア、128GBメモリ)×8)		48.0	16	
		追加単位	315,000 円/年			4ノード((36コア、128GBメモリ)×4)		24.0	8	
	タイプC1	最小	130,000 円/年	C	優先	1ノード((72コア、3072GBメモリ)×1)		24.0	8	
		追加単位	130,000 円/年			1ノード((72コア、3072GBメモリ)×1)		24.0	8	
	タイプC2	最小	156,000 円/年		準優先	2ノード((72コア、3072GBメモリ)×2)		28.8	16	
		追加単位	78,000 円/年			1ノード((72コア、3072GBメモリ)×1)		14.4	8	
大規模ジョブ	タイプA	最小	20,000 円/週(7日)		A	占有	8ノード((68コア、16+96GBメモリ)×8)	168	-	-
		追加単位	10,000 円/週(7日)				4ノード((68コア、16+96GBメモリ)×4)			
	タイプB	最小	21,000 円/週(7日)	B	占有	8ノード((36コア、128GBメモリ)×8)				
		追加単位	10,500 円/週(7日)			4ノード((36コア、128GBメモリ)×4)				
	タイプC	最小	13,000 円/週(7日)	C	占有	2ノード((72コア、3072GBメモリ)×2)				
		追加単位	6,500 円/週(7日)			1ノード((72コア、3072GBメモリ)×1)				
専用クラスタ	-	最小	630,000 円/年	B	-	8ノード((36コア、128GBメモリ)×8)	-	48.0	16	
		追加単位	315,000 円/年			4ノード((36コア、128GBメモリ)×4)		24.0	8	
ストレージ容量追加			10,000 円/年	ストレージ容量10TBの追加につき						
ライセンスサービス			20,000 円/年	可視化ソフト(AVS.ENV/IDL)およびプリポソフトウェアの1ライセンスにつき						

備考

- 利用負担額は、年度単位で算定している。また、総額表示である。パーソナルコース、グループコース又は専用クラスタコースを、年度途中から利用を開始する場合及び年度途中で利用を終了する場合の利用負担額は、上記表中の利用負担額を12で除した後、利用月数を乗じて算出するものとし、100円未満に端数が出た場合は、10円単位を四捨五入するものとする。
なお、月途中から利用を開始する場合及び月途中で利用を終了する場合は、それぞれ1月の利用とする。
- 大型計算機システムの全ての利用者は、上記表のサービスの他、次のサービスを受けることができる。
 - 大判プリンタサービス
 - その他、大型計算機システムが提供するサービス、機器の利用
- 上記表の大規模ジョブコース、ストレージ容量追加、ライセンスサービスの申請には、スーパーコンピュータシステムの利用者であることが必要である。
- 「共有」：当該カテゴリのユーザ間で一定の計算資源を共有するベストエフォートのスケジューリングを行う。
「準優先」：定常稼働状況において記載値(以上)の計算資源が確保されるように優先スケジューリングを行う。
また、稼働状況によらず記載値の1/4の計算資源が確保されることを保証する。
「優先」：定常稼働状況において記載値(以上)の計算資源が確保されるように優先スケジューリングを行う。
また、稼働状況によらず記載値の1/2の計算資源が確保されることを保証する。
「占有」：稼働状況によらず記載値の計算資源が確保されることを保証する。
- ストレージ容量はバックアップ領域(最大で総容量の1/2)を含む。
- グループコース及び専用クラスタコースの利用者番号は利用者あたり年額5,000円を負担することで追加できる。
- 機関・部局定額制度
他機関又は学内における部局(『国立大学法人京都大学の組織に関する規程』第3章第2節から第11節で定める組織をいう。)の組織が、その組織単位でグループコースサービスを利用申請する場合の利用負担額は、別表1に規定する1.5倍の額とする。なお、利用負担額が年額150万円未満の場合は100人、年額150万円を超える場合は、150万円毎に100人までの利用者を認める。ストレージは、1.5倍の容量とする。
- スパコン連携サービス
学術情報メディアセンターのスーパーコンピュータシステムと密な連携により、学内における部局の組織が計算サーバ等を設置する場合、下記の負担額を支払うものとする。

冷却方式	利用負担額	利用負担額算定単位
水冷	9,800 円/月	水冷冷却方式の計算サーバ等の定格電力 1kWにつき
空冷	11,500 円/月	空冷冷却方式の計算サーバ等の定格電力 1kWにつき

別表2 汎用コンピュータシステム

区 分	利用負担額	単 位
仮想サーバホスティングサービス	36,000円／年	1仮想サーバにつき

備考

1. 利用負担額は、総額表示である。
2. 上記表の仮想サーバホスティングサービスを利用するには、スーパーコンピュータシステムの利用者であること。
3. 1仮想サーバに割当てシステム資源は、CPU:2コア、メモリ:4GB、ディスク:100GBである。
4. 仮想サーバホスティングサービスにおいて、下記の負担額を支払うことによりCPU、メモリ、ディスクを増量することができる。

区 分	利用負担額	単 位
CPU増量	3,000円／年	2コアにつき(最大8コアまで)
メモリ増量	3,000円／年	4GBにつき(最大64GBまで)
ディスク増量	6,000円／年	100GBにつき(最大1,000GBまで)

5. 利用負担額は、当該年度(4月から翌年3月まで)の利用に対して年額として算定するが、年度途中から利用を開始する場合には月数に応じて減額する。

別表3 スーパーコンピュータシステム

システム	システム資源	経過時間 (時間)	ストレージ (TB)	無料 利用者数	利用負担額
A	8ノード(68コア、16+96GBメモリ)×8)	336	28.8	16	960,000 円/年
	12ノード(68コア、16+96GBメモリ)×12)	336	43.2	24	1,440,000 円/年
	16ノード(68コア、16+96GBメモリ)×16)	336	57.6	32	1,920,000 円/年
B	8ノード(36コア、128GBメモリ)×8)	336	28.8	16	1,008,000 円/年
	12ノード(36コア、128GBメモリ)×12)	336	43.2	24	1,512,000 円/年
	16ノード(36コア、128GBメモリ)×16)	336	57.6	32	2,016,000 円/年
C	2ノード(72コア、3072GBメモリ)×2)	336	28.8	16	624,000 円/年
	3ノード(72コア、3072GBメモリ)×3)	336	43.2	24	936,000 円/年
	4ノード(72コア、3072GBメモリ)×4)	336	57.6	32	1,248,000 円/年

備考

1. 利用負担額は、年度単位で算定している。また、総額表示である。パーソナルコース、グループコース又は専用クラスタコース
年度途中から利用を開始する場合及び年度途中で利用を終了する場合の利用負担額は、上記表中の利用負担額を12で除し
利用月数を乗じて算出するものとし、100円未満に端数が出た場合は、10円単位を四捨五入するものとする。
なお、月途中から利用を開始する場合及び月途中で利用を終了する場合は、それぞれ1月の利用とする。
2. ストレージ容量はバックアップ領域(最大で総容量の1/2)を含む。

— サービス利用のための資料一覧 —

1. スーパーコンピュータシステム・ホスト一覧

- システム A : camphor.kudpc.kyoto-u.ac.jp
- システム B・C : laurel.kudpc.kyoto-u.ac.jp
 - システム B (SAS 利用時) : sas.kudpc.kyoto-u.ac.jp

※ ホストへの接続は SSH(Secure SHell) 鍵認証のみ、パスワード認証は不可

2. 問い合わせ先 & リンク集

- 情報環境機構のホームページ
<http://www.iimc.kyoto-u.ac.jp/>
- 学術情報メディアセンターのホームページ
<http://www.media.kyoto-u.ac.jp/>
- スーパーコンピュータシステムに関する問い合わせ先
 - 利用申請などに関する問い合わせ先
【情報環境支援センター】
E-mail : zenkoku-kyo@media.kyoto-u.ac.jp / Tel : 075-753-7424
URL: <http://www.iimc.kyoto-u.ac.jp/ja/services/comp/>
 - システムの利用など技術的な問い合わせ先
【スーパーコンピューティング掛】
E-mail : consult@kudpc.kyoto-u.ac.jp / Tel : 075-753-7426
URL: <http://www.iimc.kyoto-u.ac.jp/ja/services/comp/contact.html>

京都大学学術情報メディアセンター全国共同利用版広報 Vol. 17, No. 2

2019年 3月 11日 発行

編集者 京都大学学術情報メディアセンター
全国共同利用版広報編集部会
発行者 〒606-8501 京都市左京区吉田本町
京都大学学術情報メディアセンター
Academic Center for Computing and Media Studies
Kyoto University
Tel. 075-753-7407
<http://www.media.kyoto-u.ac.jp/>
印刷所 〒616-8102 京都市右京区太秦森ヶ東町 21-10
株式会社エヌジーピー

広報編集部会

深沢 圭一郎 (部会長)

平石 拓 (副部会長)

水谷 幸弘

熊谷 真由美

小林 正信

尾形 幸亮

表紙デザイン：中山 豊
(中山商店)

目次

【巻頭言】

- ・ Vol.17, No.2号の発刊に当たって 深沢 圭一郎 1

【講習会参加報告】

- ・ Cray XC40 プログラミング講習会－Scaling DL Training Workloads with the Cray PE Plugin 平石 拓 2

【バーストバッファによる高速ファイルI/O】

- ・ バーストバッファの利用について～ DataWarp 編～ 當山 達也, 尾形 幸亮, 疋田 淳一 6

【サービスの記録・報告】

- ・ スーパーコンピュータシステムの稼働状況 9
- ・ センター利用による研究成果(平成29年度) 12

【資料】

- ・ 大型計算機システム利用負担金 別表 16
- ・ サービス利用のための資料一覧 19

【奥付】

- ・ 奥付 20