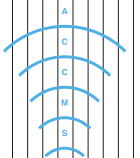


全国共同利用版

# 広報

## バーストバッファによる高速ファイルI/O

【巻頭言】「Vol.16, No.2号の発刊に当たって」深沢 圭一郎【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～IME編～」尾形 幸亮, 池田 健二, 疋田 淳一●「流体系シミュレーションでのプロセス分散書き出しにおけるIMEの効果」深沢 圭一郎【性能評価報告】「新スーパーコンピュータのベンチマーク評価報告」平石 拓



## 巻頭言

Vol. 16, 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

本号では、「バーストバッファによる高速ファイル I/O」というタイトルで、2017 年 10 月からサービスを開始したバーストバッファについて特集いたします。バーストバッファとは通常の HDD から成るストレージシステムでは読み書きの速度が不足する状況において性能改善が期待される SSD で構築された高速な一時領域です。2016 年 10 月からの新スーパーコンピュータシステム更新に伴い、DDN 社の IME と Cray 社の DataWarp がバーストバッファとしてセンターに導入されました。準備状況の違いにより現在のところユーザのみなさまにはサブシステム B と C から DDN 社の IME がご利用いただけます。その構成や理論性能、使い方など詳しくは特集内の「バーストバッファの利用について」に説明してあります。バーストバッファはその構成により、小さいサイズのファイルを読み書きする場合などに性能を発揮します。一方で、よくスーパーコンピュータシステムで行われている数値シミュレーションなどの大規模で連続したデータの読み書きではそれほど効果が期待できません。そこで特集の「流体系シミュレーションでのプロセス分散書き出しにおける IME の効果」では実際にスーパーコンピュータシステムで利用されているアプリケーションを使い、バーストバッファである IME の性能評価を行っています。この評価では、IME を利用すると書き出し性能は向上することが報告されています。HDD と SSD の I/O の違いほど劇的な性能向上はありませんが、使ってみる価値はあると言えますので、ユーザのみなさまには一度試していただければと思います。また、サブシステム A で利用される Cray 社の DataWarp につきましては、利用可能となり次第性能評価を含めお知らせいたします。

Vol.15 No.1 で特集した「新スーパーコンピュータサービス開始」では、新スーパーコンピュータシステム自体の紹介とサブシステム A のベンチマーク結果の報告でしたが、本号では「新スーパーコンピュータのベンチマーク評価報告」において、新サブシステム A, B, C の性能評価を行い、旧スーパーコンピュータシステムの性能とも比較しています。以前の報告では、サブシステム A において、MCDRAM をキャッシュモードでしか利用できませんでしたが、現在 Flat モードでの利用も可能となっており、その性能評価も報告されています。Xeon Phi (サブシステム A) と Xeon (サブシステム B) のどちらを利用した方が早く計算できるのかが、興味のあるところだと思いますが、各ベンチマーク結果を参考にご自分のアプリケーションではどちらが最適なのかご判断いただければと思います。

現在のスーパーコンピュータシステムには GPU が搭載されておらず、機械学習系の計算に利用されづらい環境となっていますが、センターとして Python や Tensorflow に代表される機械学習フレームワークの環境構築を行っており、できる限り高速に動作するような

最適化も行っているところです。センターとしては、すでにご利用されている計算機環境において、様々な計算やアプリケーションが動作することが大事だと考えておりますので、これからもユーザのみなさまにご利用いただけるような環境作りを行っていきたいと考えております。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。

## バーストバッファの利用について ～IME 編～

尾形 幸亮 池田 健二 疋田 淳一

京都大学 企画・情報部

## 1 はじめに

学術情報メディアセンター（以下、本センターという）では、2017年10月よりバーストバッファの提供を開始しました。バーストバッファは、通常のストレージシステムでは読み書きの速度が不足する状況における性能改善を目的とした高速な一時領域です。

現在のストレージシステムは Data Direct Networks 社の Exascaler を導入しており、16PB の容量と転送性能 100GB/sec を持っています。並列ファイルシステムであるため、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で高い性能を発揮できますが、小さい単位での I/O はやや苦手としています。一方、バーストバッファは多数の SSD で構成していることで高い転送性能を持ち、ランダムアクセスや小さいサイズの I/O についても高い性能を発揮しやすい構成となっています。ただし、大容量を確保することが難しいため、計算ノードとストレージの間にある一時領域として I/O 性能の改善に利用します。

本センターのバーストバッファとして、システム B、C 向けに Data Direct Networks 社の IME を導

入しています。IME は、容量 230TB、転送性能 240GB/sec であり、Exascaler の転送性能と比較し、約 2.4 倍の性能です。システム B、C、ストレージシステム、および IME の性能諸元値を記した構成図を図 1 に示します。以下、本稿では IME の利用方法について解説します。

## 2 利用申請

IME をお使いいただくためには、利用申請が必要です。バーストバッファは新しい技術ですので、利用される方と情報共有しながら性能をご確認いただくことを想定しております。

申請資格のある方は、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管理者の方です。エントリコースおよび専用クラスタの方はご利用いただけません。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。

<http://web.kudpc.kyoto-u.ac.jp/manual-new/ja/filesystem/ime#application>

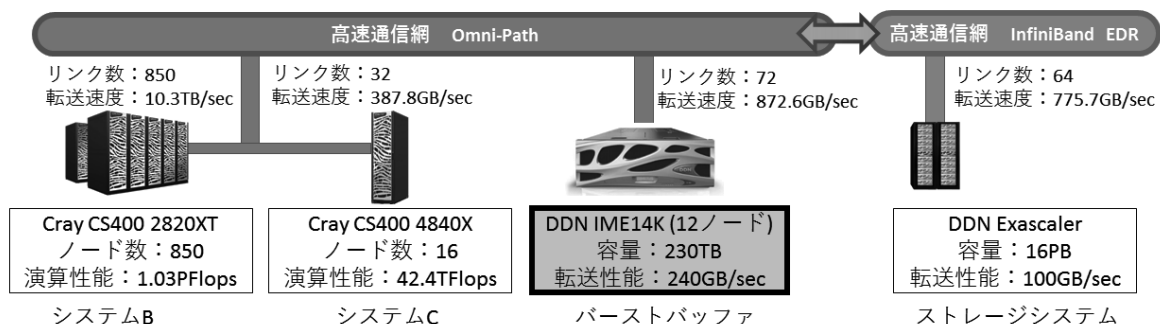


図 1 システム構成図

### 3 IME の利用イメージ

IME 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 IME 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。IME を利用する場合、プログラムの実行時間は、実行中に IME 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、IME 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。

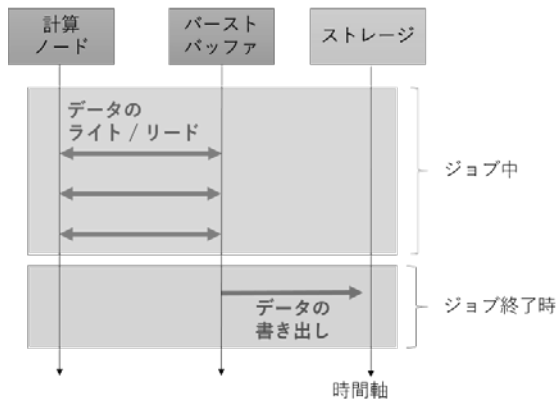


図 2 IME の利用イメージ

## 4 IME の基本的な使い方

### 4.1 概要

IME は、パーソナルコースやグループコース向けに提供しているストレージ領域の /LARGE0 のバーストバッファとして利用できます。現時点では /LARGE1,2,3 では利用できません。IME を利用するには、バッチジョブを実行する際に、-bb オプションを指定する必要があります。IME は POSIX インターフェイスに対応しているため、通常のファイルシステムと同じようにアクセスすることができます。cd や ls などの通常のコマンド操作も可能です。Exascaler と IME はメタデータ（ファイルの情報）を共有しているため、IME を使う場合でも /LARGE0 のファイルに透過的にアクセス可能です。次のように、ディレクトリの絶対 PATH は異なりますが、IME の PATH の先頭に /IME が付与される以外は、同じファイルツリーを参照できます。プログラムが

参照するファイルが相対 PATH で指定されている場合は、プログラムや入力ファイルを変更することなく利用することができます。ただし、計算ノードからのみ IME の領域を参照することが可能です。ログインノードからは IME 領域にアクセスすることはできません。

```
通常ストレージアクセス:
/LARGE0/group/file
IME 経由のアクセス:
/IME/LARGE0/group/file
```

### 4.2 ジョブスクリプトの記述

バーストバッファを使うには、ジョブスクリプトに -bb オプションを記述します。-bb オプションの記法は次の通りです。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
capacity	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) -bb capacity=2000G
pfs	バーストバッファを適用するディレクトリ。LARGE0 配下のディレクトリを指定。省略した場合はカレントディレクトリ。 例) -bb pfs=./data

ジョブスクリプトの例を以下に示します。この例ではバーストバッファの領域として、100GB を要求し、バーストバッファを利用するディレクトリを省略したことで、カレントディレクトリがバーストバッファの対象になります。

```
#!/bin/bash
#QSUB -q gr10001b
#QSUB -ug gr10001
#QSUB -A p=8:t=4:c=4:m=3413M
#QSUB -bb capacity=100G

mpirexec.hydra ./a.out
```

## 4.3 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 2 の上限があります。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。IME の利用申請をしていない場合もエラーになります。

表 2 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 グループコースの契約ノード数

## 4.4 IME を経由したファイル読み込み

IME は、事前にファイルを読み込まない限り、ストレージにあるファイルを直接読み込みます。予め IME に読み込んでおく場合は、専用のコマンドを実行する必要がありますが、同じファイルを何度も読み込む必要が無い限り、事前に読み込む利点はありません。後述するようにメタデータアクセスの低速さの問題もありますので、プログラムからのファイルの読み込みは IME を経由せず、ストレージから直接読むほうが高速なケースが多いと考えられます。IME の容量が小さいこともありますので、事前に IME に読み込むためのプリフェッチコマンドは、提供していません。利用したい場合はお問い合わせください。

## 4.5 IME を経由したファイル書き込み

IME 上のファイルパスに対して書き込み処理を行うと、IME 上にもみファイルが生成し、ストレージに書き込む前のキャッシュされている状態となります。バッチジョブの終了時には、ジョブ投入時に指定した領域(-bb の pfs)については自動で書き出しを行います。専用のコマンドを実行することで、自身で明示的に書き出しを行うことも可能です。また、ジョブスクリプトで指定していないディレクトリにファイルを書きしてしまった場合は自分で書き出しが必要です。

## 5 IME に関わるシステムの動作

### 5.1 ジョブスクリプトを起動するディレクトリ

通常のジョブの場合、qsub コマンドを実行したディレクトリに自動で移動したうえで、ジョブスクリプトを起動します。-bb によりバーストバッファの利用を宣言した場合、pfs で指定したディレクトリの IME 領域 (IME で始まるディレクトリ) に移動したうえでジョブスクリプトを起動します。pfs を省略した場合は、qsub を実行したカレントディレクトリに対応する IME 領域でジョブスクリプトを起動します。これにより、相対 PATH でアクセスするファイルは全て IME を経由したアクセスになります。

なお、ジョブ投入時のディレクトリは、環境変数 \$QSUB\_WORK\_DIR に保存されています。-bb オプションの pfs で指定したディレクトリは、\$QSUB\_BB\_DIR に保存されています。

### 5.2 メタデータアクセス

IME は/LARGE0 と同じメタデータサーバを参照するので、IME を経由しても同一のファイルの読み書きが可能です。しかし、メタデータアクセスの性能はリモートの/LARGE0 を参照することになるため、IME が劣ります。大量のデータを Open/Close するようなアクセスパターンには向かない点をご理解の上使用してください。

また、1つのディレクトリに置くファイル数が多いとメタデータへのアクセスも増えるため、ディレクトリに置くファイル数を最小限にすることをお勧めします。これは Exascaler においても同じことが言えますが、IME の方が強く影響があります。

### 5.3 ファイルの見え方

IME にファイルを書き込むと、/LARGE0 のメタデータサーバにもファイル情報が書かれるため、ファイルの存在をすぐに確認できます。しかし、IME にファイルをキャッシュしている状態では/LARGE0 にはまだファイルが無い状態の為、ls -l コマンドで確認しても容量が 0 の状態です。

ジョブの終了時に/IME から/LARGE0 にファイルの書き出し処理を行うまで、中身を参照することはできません。後述の ime-stat コマンドで、ファイル

の状況を確認することができます。

```
## IME のディレクトリにデータを作成
$ dd if=/dev/urandom ¥
  of=/IME/LARGE0/gr10001/testfile ¥
  bs=1M count=512

## /IME 領域に対して ls -l を実行
$ ls -lh /IME/LARGE0/gr10001/
-rw-r--r-- 1 b59999 gr10001 512M 10 Oct
13:53 /IME/LARGE0/gr10001/testfile

## /LARGE0 領域に対して ls -l を実行
$ ls -l /LARGE0/gr10001/
-rw-r--r-- 1 b59999 gr10001 0 10 Oct
13:53 /LARGE0/gr10001/testfile
```

## 5.4 ファイルのシンクとリリース

ジョブの終了時に `-bb` オプションで指定したディレクトリに対して、自動的にファイルのシンク処理 (`/IME` から `/LARGE0` にファイルを書き出す処理) とリリース処理 (`/IME` からの削除処理) を行います。IME 全体の容量が圧迫された場合にも、自動でシンクおよびリリースが行われることがあります。

ジョブスクリプトにコマンドを記述することでシンクやリリースを指示することも可能です。シンクにより `/IME` から `/LARGE0` にファイルの書き出し処理が行われますが、`/IME` 上にはファイルが残ったままの状態です。`/IME` 上のファイルを解放し、容量を空けるためにはリリースコマンドを使用する必要があります。コマンドは 6 章のコマンド一覧を参照ください。

## 5.5 IME 使用中の/LARGE0 のファイルの扱い

IME 使用中も、`/LARGE0` に既に存在するファイルを `/LARGE0` の `PATH` で読み込みは可能です。しかし、IME で書込みを行うファイルに対して、`/LARGE0` 側でも書込みを行うとファイルが壊れるため、避けてください。

## 5.6 IME 領域の容量監視について

IME は `LARGE` に比べて小さい容量しか持っていないため、`-bb` で宣言して頂いた容量以内に収まっ

ているか監視しています。容量を超過しているジョブは自動的に `kill` されますのでご注意ください。なお、IME を利用するディレクトリ配下の容量を合計した監視を行うため、複数のジョブで同じディレクトリを使う場合、ジョブ単独では、宣言した容量以下だとしても全体として超過している場合は `kill` されますのでご注意ください。

前述のメタデータ性能が高くない点も踏まえて、ジョブごとにディレクトリを分けることをお勧めします。

## 5.7 -bb で指定した領域以外へのアクセス

`/IME` で始まる領域は、普通にユーザおよびグループのパーミッションでアクセスできるため、`-bb` で指定していない領域もファイルの読み書きが可能です。しかし、ジョブ実行の最後に自動でシンクを行う対象は `-bb` オプションの `pfs` で指定した領域、あるいは省略したジョブ投入時のディレクトリのみが対象になるため、指定した領域外への書込みは避けてください。読み込みは特に問題ありません。

## 5.8 定期保守時の IME の残存ファイルの扱い

定期保守の際に、IME 領域に残っているファイルは全て削除します。保守に入る前に実行中のジョブは、一旦 `Hold` 状態にするため、`-bb` オプションで指定したディレクトリについては、自動的に書き出しが行われますが、それ以外で残存しているファイルは削除します。保守の前に忘れずにシンクを行ってください。

# 6 IME コマンドの紹介

## 6.1 ime-stat

IME に関するファイルの情報を出力します。`IM SIZE` が IME 上のファイルサイズで、`FS SIZE` がストレージ側のファイルサイズです。以下の使用例の場合、`FS SIZE` は 0 であり、IME 上のみファイルがある状態です。

```
Usage: ime-stat [option] <絶対パス>
Option:
  -r, --enable-recursive
```

使用例：

```
$ime-stat /IME/LARGE0/gr10000/512m.dat
File   : /IME/LARGE0/gr10000/512m.dat
IM Size  : 536870912
FS Size  : 0
Device  : 2c54f966h/743766374d
Inode   : 144117086384371784
Access  : (100644/-rw-r--r--)
Uid     : (41631/b00001)
Gid     : (20408/gr10000)
Access  : Wed Oct 26 13:48:03 2016
Modify  : Wed Oct 26 13:48:42 2016
Create  : Wed Oct 26 13:48:42 2016
```

## 6.2 ime-sync

IME にキャッシュされたデータをストレージにシンクします。-b オプションを指定すると、シンクが完了するまでコマンドが終了しないため、シンクの完了を明確に把握することができます。

Usage: ime-release [option] <絶対パス>

Option:

```
-b, --block
-r, --enable-recursive
-V, --verbose
```

## 6.3 ime-release

IME にキャッシュされたデータを削除するコマンドです。そのまま実行するとシンクが終わっていないファイルも削除しますが、-k オプションを付けることで、シンクが完了していない場合には削除しないという処理も可能です。

Usage: ime-release [option] <絶対パス>

Option:

```
-k, --keep-unsync
-r, --enable-recursive
-V, --verbose
```

## 6.4 ime-lsfiles

IME にキャッシュされているファイルをリストします。まだシンクしていない状態のファイルを探す際に利用します。

Usage: ime-lsfile <ディレクトリの絶対パス>

## 6.5 ime-cat

IME にキャッシュされたファイルを連結し、標準出力に出力します。cat コマンドと同様な機能です。

Usage: ime-cat <絶対パス>

## 6.6 IME のラッパーコマンド

ベンダー提供の ime で始まるコマンド群は、絶対 PATH の指定が必要であるなど不便なため、本センサーでは表 3 に示すラッパーコマンドを用意しています。以下に示す PATH の変換および表 3 に示すオプションの自動付与を行うことで、使いやすくしています。

コマンド共通の動作：

- 相対 PATH を絶対 PATH に変換したうえでオリジナルのコマンドを実行します。
- /IME を含む IME 上の PATH を /LARGE0 の PATH に変換してオリジナルのコマンドを実行します。
- -c -l -L オプションは無視します。

表 3 ラッパーコマンドの一覧

コマンド名	動作
imeutil-stat	ime-stat を実行します。
imeutil-sync	ime-sync を実行します。
imeutil-release, imeutil-force-release	imeutil-release は ime-release -k を実行します。imeutil-force-release は ime-release を実行します。
imeutil-cat	ime-cat を実行します。
imeutil-lsfiles	ime-lsfiles を実行します。
imeutil-sync-and-release	ime-sync -V -b を実行したあとに、ime-release -V -k を実行します。

## 7 おわりに

本稿では、IME の利用に必要な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-new/ja>



# 流体系シミュレーションでのプロセス分散書き出しにおける IME の効果

深沢 圭一郎

京都大学 学術情報メディアセンター

## 1 バーストバッファ, IME とは

バーストバッファについては、1 つ前の「バーストバッファの利用について」という紹介記事に詳しい説明がありますので、ここでは簡単な説明をします。京都大学のスーパーコンピュータ（スパコン）システム A, B, C には、ファイルアクセスの高速化を目的とした機構が接続されており、その機構をバーストバッファと呼んでいます。バーストバッファは、SSD（半導体素子から成るストレージデバイス）を利用して構築された、高速ですが小容量な一時保存用ストレージであり、計算ノードと通常のファイルシステムの間で接続されています。プログラムが計算ノードからデータを書き込む時に、ファイルシステムではなくバーストバッファにデータを書き込むことで、高速な書き込みを実現します。バーストバッファ上のデータは自動で（任意のタイミングも可）ファイルシステムに移動します。また、書き込みとは逆にファイルシステムからデータを読み込む場合にも、同様に高速な読み込みを実現するシステムです。

京都大学のスパコンシステムでは、システム A にクレイ社の DataWarp を、システム B と C には DDN 社の IME がバーストバッファとして導入されています。現在システム A のバーストバッファは準備中のため、本記事ではシステム B 上で IME を対象とした性能評価を行いました。

前述のようにバーストバッファは、SSD により構成されたストレージシステムのため、一般的にランダムアクセスや、多数のプロセスが共有ファイルにアクセスするような場合に効果を発揮します。しかしながら、スパコンを利用して計算を行っているアプリケーションの多くは、流体系の計算に代表されるように大規模なデータをシーケンシャルで分散

書き出しを行います。そこで、実際に研究に使用されている流体系のアプリケーションから IME を利用し、このような書き出しにも効果があるのか調査しました。

## 2 実アプリケーションによる性能評価

IME の性能評価には深沢が開発している宇宙プラズマのシミュレーションを行う MHD (Magnetohydrodynamic) コード[1]を利用しました。MHD コードはプラズマを扱うために電磁場を考慮した流体コードで、方程式が少し通常の流体より複雑になり、電磁場分だけ物理量が増えることで書き出し量も増加します。詳細は参考文献[1]をご覧ください。実際の出力部分を抜き出すと下記のようになります。

```
do k = 1, nz2
  write(output) f(1:nx2,1:ny2,k)
end do
```

この MHD コードは Fortran で書かれており、MPI でプロセス並列されています。そのため、上記の書き出しが各プロセスで分散的に行われるということになります。1 つのプロセスにデータを集めて (MPI\_gather など) で、その 1 プロセスのみで書き出しを行うテストも行いましたが、性能の傾向は後述の結果と変わりませんでしたので、ここでは省きました。ただし、(当たり前ですが) 書き出し速度は 1 プロセスが圧倒的に遅くなります。また、計測は出力時間が計測の度に大きくばらつくため、10 回計測し、著しく遅い結果は除外し、平均を取りました。

まず、計算サイズ (出力サイズ) を 12 GB に固定し、1 ノード内で利用できる範囲でプロセスを変化

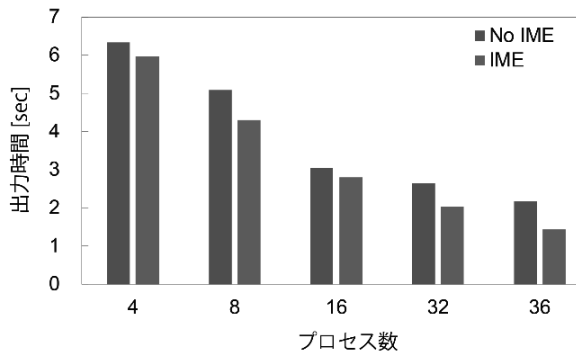


図1 計算サイズを12GBに固定し、プロセス数を変化させた場合の出力時間変化

させ(36プロセスまで)、IMEを利用する場合としない場合のデータ出力時間を測定しました。測定結果が図1になります。横軸が利用プロセス数、縦軸が書き出しにかかった時間です。システムBでは1ノードに18コアのXeonが2つ搭載されていますので、32プロセスからは2CPUを利用した結果になります。IMEを利用した場合には、IMEを利用しない場合よりも書き出しが全般的に早く行われており、IMEの効果が見えます(6~50%高速)。書き出し性能は16プロセスまでは性能向上がよく見えますが、IMEを利用しない場合は2CPUを利用するところから性能向上が頭打ちになっています。IME利用時では、同じ2CPU利用時でも書き出し速度が向上していることが分かります。これは、出力サイズは変わらず、プロセス数が増加することで、多プロセスが小規模ファイルの書き出しを行うこととなることから、IMEの効果が出やすい状況のためと考えられます。

次に、プロセス数を36(=1ノード)に固定し、計算サイズを変化させた場合の性能評価を行った結果、図2のようになりました。横軸が出力データサイズ、縦軸が出力時間になります。MHDコードの特性上、システムBではノード当たり計算サイズ19GBが最大のため、このサイズまで測定しています。この評価でもIMEを利用した場合に出力時間が短くなっていることがわかります。計算サイズが増えると、出力時間も増えますが、IMEを利用しない場合には12GBと19GBの出力であまり時間に変化が見られないことから、出力サイズが小さい場合には、通常ファイルシステムへの書き出し性能を使い切れていないことが考えられます。

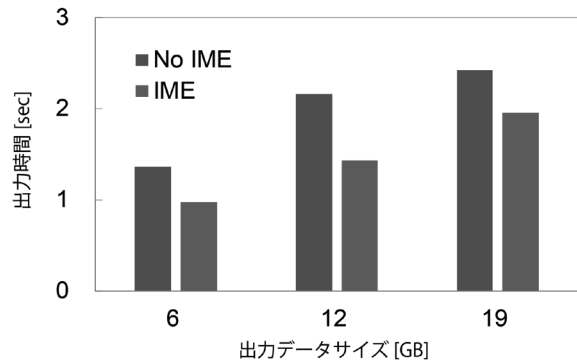


図2 36プロセス(1ノード)において、出力サイズを変化させた場合の出力時間変化

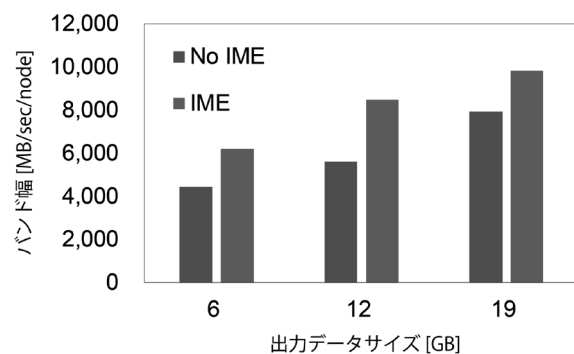


図3 36プロセス(1ノード)において、出力サイズを変化させた場合の出力バンド幅変化

図2から出力バンド幅を計算したものを図3に示します。これはノード当たりの出力バンド幅になります。横軸は出力データサイズ、縦軸がバンド幅です。IMEを使う場合には12GBと19GBでバンド幅があまり変わらないことから、出力限界が約10GB/sec/nodeと想像されます。IMEを使わない場合には、19GB書き出し時の7.9GB/sec/nodeとなります。IMEを使うことで、2割以上出力バンド幅の向上が見られ、12GB以上の出力では、計算サイズに依らず高い出力バンド幅を達成できていると言えます。また、システムB1ノード当たりの理論転送速度が12GB/secということを考えると、これらは妥当な値と考えられます。

最後に複数ノードを利用した性能測定を行いました。1ノード当たりのプロセス数を36に、計算サイズを19GBと設定し、2, 4, 8, 16ノードを利用し、IME利用時と利用しない場合の出力時間を計測しました。図4がその計測結果となります。横軸が利用ノード数、縦軸が出力時間となります。この

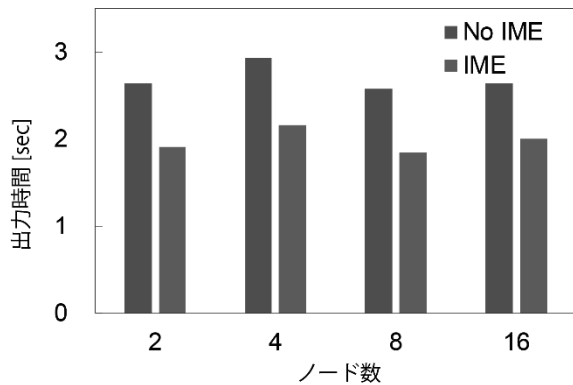


図4 ノード当たりのプロセス数, 計算サイズを固定し, ノード数を変化させた場合の出力時間変化

評価ではノード当たりの出力サイズは変わらないため, 出力時間が一定となるとスケーラビリティが良いと言えます. 測定結果では性能が少しばらついていますが, IME を利用する場合, しない場合でもスケーラビリティは悪くない結果と言えます. 性能自体は1ノード利用時と同様で, IME を利用した場合の性能が良いことが分かりますが, スケーラビリティは IME の有無による違いはないことが分かります.

### 3 まとめ

本記事では, 新しく京都大学のスパコンに導入された IME について, 流体系のシミュレーションコードである MHD コードを用いて, 性能評価を行いました. パーストバッファと呼ばれる高速 I/O 一時記憶装置の1つである IME では, その構造から, 細かなファイルの書き出しや, 共有ファイルへの同時書き出しなどに効果があるとされています. 一方で, スパコンを利用する多くのアプリケーションは, シーケンシャルに大規模データを分散書き出すことが多いため, 実際の利用を想定し, そのような書き出しにも IME の効果があるのかを調査しました.

今回計測した結果では, すべてにおいて IME を利用した方が書き出しは早くなるという結果が得られました. また, IME を利用した場合には, プロセス数の増加や書き出しサイズの増加といった変化に対しても, リニアに書き出し性能が変化している結果となり, 多数のプロセスで, 大きなデータを書き出すような計算では IME を利用する価値があると言えそうです. 実際のプロダクトランでは, 時

間発展毎に書き出すなどチェックポイント的な出力が多くあり, この部分に IME を利用することで, IME から通常のファイルシステムへのシンクの時間にも計算を進めることができるため, 実利用でも総計算時間の短縮に繋がることが期待されます.

現在, IME は主にデータの書き出し利用に向けた運用となっていますが, 読み込み時にも利することで, チェックポイントリスタート時に, ジョブ開始前に IME にデータを読み込んでおくなどの利用法が考えられます. IME の機能自体もバージョンアップしていくと開発元の DDN 社から報告を受けていますので, 今後も実際の計算に役に立つ使い方などを随時スパコンシステム利用ユーザのみなさまへお知らせできるようにいたしますので, ご協力よろしくお願いたします.

### 参考文献

- [1] Fukazawa, K., T. Ogino, and R.J. Walker, The Configuration and Dynamics of the Jovian Magnetosphere, *J. Geophys. Res.*, 111, A10207, doi:10.1029/2006JA011874, 2006.

# 新スーパーコンピュータのベンチマーク評価報告

平石 拓\*

\*学術情報メディアセンター

## 1 はじめに

京都大学学術情報メディアセンターは、2016年10月に新しいスーパーコンピュータ Camphor 2 のサービスを開始した。また、同年12月からはさらに Laurel 2 と Cinnamon 2 のサービスを開始した。

本稿では、これらのうち Camphor 2 および Laurel 2 と、2014年度にサービスを開始し既に運用を終了した旧スーパーコンピュータシステムにおける、いくつかのマイクロベンチマークおよび実アプリケーションに基づくベンチマークプログラムによる性能評価の結果を示し、両者の性能を比較する。

なお、本原稿は昨年度の本誌記事 [8] の補完版という位置付けである。具体的には、前記事執筆以後に利用可能になった以下の評価を追加している。

- Laurel 2 の性能評価。
- Camphor 2 の MCDRAM のメモリモードを Flat モードに設定した際の性能評価。

## 2 システム構成

2016年度、2014年度、2012年度に稼働を開始したスーパーコンピュータの性能諸元をそれぞれ表1、表2、表3に示す。

Camphor 2 は 2012 年度に導入された Camphor と同じく高い電力性能比を指向したシステムであり、Camellia (2014–2018 年度稼働) に搭載されている Knights Corner (KNC) の後継である最新の Xeon Phi メニーコアプロセッサ Knights Landing (KNL) を採用している。世界のスパコンの Linpack 性能を競う Top500 では 33 位 (日本では 4 位)、電力比性能を競う Green500 では 9 位 (同 3 位) にラ

ンキングされた (いずれも初出時である 2016 年 11 月のランキング) [7]。

Camellia の KNC は、通常の Xeon プロセッサノードに追加するコプロセッサとして搭載されていたが、Camphor 2 の KNL はホスト CPU であり、Linux 等の OS も KNL 上で直接動作する。そのため、Camellia のようにホストコプロセッサ間のデータ転送を意識するプログラミングは不要となる。KNL は SIMD 演算性能の強化、アウトオブオーダー実行ができるようになっているなどコアあたりの性能も KNC より大幅に強化されている。

また、Camphor 2 のノードは通常の DDR4 メモリ (96GB) に加えて MCDRAM と呼ばれる高速メモリ (16GB) を有している。この MCDRAM は、DDR4 メモリのキャッシュとして使えるモード (Cache モード) のほか、MCDRAM と DDR4 のアドレスを分離してどちらのメモリにどのデータを置くかをユーザレベルで使い分けができるモード (Flat モード)、この両者のモードを併用するモード (Hybrid モード) のいずれかで利用することができるようになっている。ただし、本稿執筆時点ではユーザは Cache モードまたは Flat モードのみを利用できる運用となっている。

Laurel 2 は旧システムの Laurel や Magnolia と同様、一般的な Xeon プロセッサを搭載したシステムであるが、プロセッサのモデルは最新のものに更新され、ノードあたりのメモリ量も 128GB に増加している。Cinnamon 2 は Cinnamon と同様にノードあたりメモリを大容量にしたシステムである。こちらも新しいモデルのプロセッサに更新され、1.5TB であったノードあたりのメモリ量は 3TB に増加している。

さらに、DataWarp (Camphor 2) および Infinite Memory Engine (IME, Laurel 2 および Cinnamon 2)

と呼ばれるバーストバッファを備えていることも新システムの大きな特徴である。これは、計算中に発生する大量のファイルI/Oを高速・低容量なストレージを経由して行うことにより、計算時のI/Oによる遅延を最小限に抑えるための仕組みである。なお、DataWarpの利用は2018年度にユーザに解放される予定である。

### 3 測定条件

#### 3.1 計算機環境

今回の性能評価では新旧世代の Xeon Phi および Xeon プロセッサの性能を比較するため、Camphor 2 (新サブシステム A), Laurel 2 (新サブシステム B), Magnolia (サブシステム D) および Camellia (サブシステム E) の 4 種類のサブシステムでそれぞれ測定を行った。

Camphor 2 の MCDRAM のメモリモードは、Cache モードと Flat モードの二種類の設定で測定した。本システムにおいては、Flat モードはジョブスクリプトに

```
#QSUB -mm flat
```

のオプションを追加すると利用できる。また、

```
aprun -n $QSUB_PROCS -d $QSUB_THREADS \  
-N $QSUB_PPN numactl --preferred=1 \  
./a.out
```

のように `--preferred=1` オプションを付加した `numactl` コマンドを用いることにより、MCDRAM のサイズの範囲内までは MCDRAM 内でメモリを確保し、それを超えるデータについては通常のメモリ (DDR4) を確保する設定でプログラムを実行した。

Camellia では KNC をネイティブ実行モード (ホストプロセッサを使わず、KNC 上でのみプログラムを実行するモード) でのみ用いた。そのため、メモリが 8 GB しか使用できず、一部のベンチマークでは他のサブシステムに比べて小さい問題サイズで測定を行っている。

#### 3.2 ベンチマークプログラム

本稿で示す評価に用いたベンチマークプログラムは以下の通りである。

- (a) HPLinpack: Innovative Computing Laboratory, University of Tennessee (ICL-UT) が配布する HPCC ベンチマーク [1] に含まれる、密行列の連立一次方程式を解くベンチマークプログラムであり、Top500 の評価にも用いられる。本評価では、複数ノード (各サブシステムの一部) を使用したときの演算性能を測定した。測定に用いた HPCC のバージョンは 1.4.2 である。
- (b) 高速フーリエ変換プログラム (FFTE) : 上記 HPCC ベンチマークに含まれる。複数ノードによる性能測定を行った。
- (c) FDTD 法による電磁場解析プログラム (FDTD) : 電場と磁場の時間発展をステンシル計算 (各格子点の次の時刻の状態を、近傍の格子点の値を使って求める) によって導く。参照局所性を高めるため、時空間タイリングという手法が用いられている [6]。MPI と OpenMP によるハイブリッド並列化が行われており、単一ノードおよび複数ノードによる性能測定を行った。
- (d) 並列ポアソンソルバ (Poisson) : 反復法によるポアソン方程式のソルバであり、マルチグリッド法による収束性向上やブロック化等による性能改善が施されている [3, 4]。本測定では、OpenMP 版による単一ノードでの評価と、Flat MPI 版による複数ノードでの評価を行った。
- (e) Intel MPI Benchmarks (IMB) : Intel 社が配布 [2] する、一対一通信、集団通信等の MPI の基本性能を測定するベンチマークプログラムである。今回測定に用いたバージョンは 4.1.10 である。8 ノード × 1 プロセス実行の評価を行った。
- (f) メモリアクセス性能評価ベンチマーク (STREAM) : John D. McCalpin 氏が作成・配布 [5] する、メモリ性能を実測により測定するベンチマークである。(性能が最大になるような) 任意の数のスレッドを生成し、それぞれのスレッドが大きなサイズの配列に対するロード・ストアを行うことでメモリの性能評価を行う。単一ノードによる性能評価を行った。

表 1: スーパーコンピュータの性能諸元 (2016 年度導入システム)

名称	サブシステム名 愛称 機種名	サブシステム A Camphor 2 Cray XC40	サブシステム B Laurel 2 CS400 2820XT	サブシステム C Cinnamon 2 CS400 4840X
稼動期間		2016 年 10 月～		2016 年 12 月～
ノード数		1800	850	16
理論演算性能		5.4 PFlops	1.03 PFlops	42.4 TFlops
Linpack 性能		3.07 PFlops	822 TFlops	—
Top500	初出	33rd (Nov. 2016)	193rd (Jun. 2017)	—
	最新	41st (Nov. 2017)	279th (Nov. 2017)	—
CPU	製品名, 周波数	Intel Xeon Phi Knights Landing 1.4 GHz	Intel Xeon Broadwell 2.1 GHz	Intel Xeon Haswell 2.3 GHz
	ソケット数 / node	1	2	4
	コア数 / node (/ socket)	68 (68)	36 (18)	72 (18)
メモリ	規格	MCDRAM + DDR4-2133	DDR4-2400	DDR3-1600
	総バンド幅	921 GB/s (MCDRAM) + 102 GB/s (DDR4)	154 GB/s	205 GB/s
	容量	16 GB (MCDRAM) + 96 GB (DDR4)	128 GB	3 TB
ネット ワーク	リンク	Cray Aries (15.75 GB/s)	Intel Omni-Path (12 GB/s)	Intel Omni-Path × 2 (12 GB/s × 2)
	トポロジ	Dragonfly	Fat tree	Fat tree
	バイセクション バンド幅	13.5 TB/s	half-bisection	full-bisection
	ストレージ	DDN ExaScaler (16 PB, 100 GB/s) (2018 年度に 8 PB, 50 GB/s を増強予定)		
バーストバッファ		Cray DataWarp (230 TB, 206 GB/s)	DDN IME (230 TB, 250 GB/s)	

表 2: スーパーコンピュータの性能諸元 (2014 年度導入システム)

名称	サブシステム名 愛称 機種名	サブシステム D Magnolia Cray XC30	サブシステム E Camellia Cray XC30
稼動期間		2014 年 7 月～2016 年 12 月	2014 年 4 月～
ノード数		416	482
理論演算性能		428.6 TFlops	583.6 TFlops
Linpack 性能		307.2 TFlops	380.5 TFlops
Top500	初出	174th (Nov. 2014)	101st (Jun. 2014)
	最終	456th (Jun. 2016)	455th (Nov. 2016)
CPU	製品名, 周波数	Intel Xeon Haswell 2.6 GHz	Intel Xeon Ivy Bridge 2.6 GHz + Intel Xeon Phi Knights Corner 1.053 GHz
	ソケット数 / node	2	1 (Xeon) + 1 (Xeon Phi)
	コア数 / node (/ socket)	28 (14)	10 (Xeon) + 60 (Xeon Phi)
メモリ	規格	DDR4-2133	DDR3-1600 (Xeon) + GDDR5 (Xeon Phi)
	総バンド幅	136 GB/s	51.1 GB/s (DDR4) + 352 GB/s (GDDR5)
	容量	64 GB	32 GB (DDR4) + 8 GB (GDDR5)
ネット ワーク	リンク	Cray Aries (15.75 GB/s)	Cray Aries (15.75 GB/s)
	トポロジ	Dragonfly	Dragonfly
	バイセクション バンド幅		4.6 TB/s
	ストレージ	DDN SFA12K (3 PB, 24 GB/s)	

### 3.3 コンパイラ・ライブラリ

ほとんどプログラムのコンパイルには, Intel Compiler を用いた, バージョンは Camphor 2 および Laurel 2 では 17, Magnolia および Camellia では 16 (一部のベンチマークでは 14) である. ただし, Camphor 2 の HPLinpack および IMB では Cray Compiler バージョン 8.5.2 を用いた.

MPI ライブラリには, Camphor 2, Magnolia, Camellia では Cray MPI を用いた. バージョンは Camphor 2 では 8.5.8, Magnolia および Camellia では 7.3.2 である. Laurel 2 では Intel MPI バージョン 2017 を用いた.

HPLinpack では, 数値ライブラリとして Camphor 2 では Cray Libsci 16.11.1, Laurel 2 では Intel MKL バージョン 2017, Magnolia では Intel MKL バージ

表 3: スーパーコンピュータの性能諸元 (2012 年度導入システム)

名称	サブシステム名 愛称 機種名	サブシステム A Camphor Cray XE6	サブシステム B Laurel Appro GreenBlade 8000	サブシステム C Cinnamon Appro 2548X
稼動期間		2012 年 4 月~2016 年 8 月	2012 年 4 月~2016 年 12 月	
ノード数		940	601	16
理論演算性能		300.8 TFlops	242.6 TFlops (含 GPU) 193.0 TFlops (除 GPU)	10.6T Flops
Linpack 性能		251.7 TFlops	135.4 TFlops (除 GPU)	—
Top500	初出	73rd (Jun. 2012)	126th (Jun. 2012)	—
	最終	405th (Nov. 2015)	494th (Jun. 2014)	—
CPU	製品名, 周波数	AMD Opteron Abu Dhabi 2.6 GHz	Intel Xeon Sandy Bridge 2.6 GHz	Intel Xeon Sandy Bridge 2.6 GHz
	ソケット数 / node コア数 / node (/ socket)	2  32 (16)	2  16 (8)	4  32 (8)
GPU		—	NVIDIA Tesla M2090 (64 ノードに搭載)	—
メモリ	規格	DDR3-1600	DDR3-1600	DDR3-1066
	総バンド幅 容量	102 GB/s 64 GB	102 GB/s 64 GB	136 GB/s 1.5 TB
ネット ワーク	リンク	Cray Gemini (9.3 GB/s or 4.6 GB/s per link)	Infiniband FDR × 2 (6.8 GB/s × 2)	Infiniband FDR × 2 (6.8 GB/s × 2)
	トポロジ	3D-torus	Fat tree	Fat tree
	バイセクション バンド幅	1.7 TB/s	3.1 TB/s	217 GB/s
ストレージ		DDN SFA10000 (5 PB, 54 GB/s)		

ョン 11.3.2, Camellia では Intel MKL バージョン 11.2.4 をそれぞれ用いた。

## 4 測定結果

各ベンチマークの性能評価結果を表 4–表 9 に示す。なお、これらの表中の「システム」は、「新 A(H)」「新 A(F)」がそれぞれ Camphor 2 の Hybrid モードおよび Flat モード、「新 B」が Laurel 2, 「D」が Magnolia, 「E」が Camellia を意味する。

### (a) HPLinpack

測定結果を表 4 に示す。プロセス分割数などのチューニングや問題サイズが十分ではなく、Top500 用の測定値ほどの対理論性能は得られていない。特に Laurel 2 ではもっと性能改善の余地があると考えられる。HPL ではノード数の比例に近い性能を得られることが期待できるため、ノードあたりでは Camphor 2 は Magnolia や Camellia より高い性能が得られると考えられる。また、入力サイズは Camphor 2 の MCDRAM の容量よりも大きくしており、Cache モードが有効に機能していることが読み取れる。

### (b) FFTE

測定結果を表 5 に示す。Camphor 2 の性能は Magnolia よりやや低くなっているが、Camellia よりは大幅に改善されていることが確認できる。また、参照の局所性があまり高くないプログラムであるため、Flat モードと Cache モードの差は HPLinpack より小さくなっている。

### (c) FDTD

測定結果を表 6 に示す。このプログラムはベクトル化率が比較的高いため、単一ノード、複数ノードともに Camphor 2 は Magnolia より高い性能が得られている。また、参照の局所性を高めるプログラムの工夫を行っているため、Cache モードと flat モードの差も大きくなっている。Laurel 2 も Magnolia に対して良好な性能が得られている。

### (d) Poisson

測定結果を表 7 に示す。このプログラムもベクトル化率は高く、Camphor 2 の性能は Magnolia を上回っている。Laurel 2 でも Magnolia に対してやや良好な性能が得られている。なお、KNL と KNC 上での OpenMP 版の実行では、ハイパースレッディングを利用して、1 物理コアに 2 スレッド割り当てることでコアあたり 1 スレッド

表 4: HPLinpack の性能測定結果

システム	新 A(H)	新 A(F)	新 B	D	E
使用ノード数	8	8	8	32	4
プロセス数 × スレッド数	544×1	544×1	288×1	896×1	240×1
N (問題サイズ)	200000	200000	200000	400000	25000
測定値 (HPL TFLOPS)	6.64	3.74	3.43	19.7	0.465

表 5: FFTE の性能測定結果

システム	新 A(H)	新 A(F)	新 B	D	E
使用ノード数	8	8	8	19	3
プロセス数	512	512	256	512	128
N (問題サイズ)	200000	200000	200000	400000	25000
測定値 (MPIFFT GFLOPS)	56.6	49.4	83.8	153	6.70

表 6: FDTD の性能測定結果

システム	新 A(H)	新 A(F)	新 B	D	E
スレッド数	64	64	36	28	60
測定値 (GFLOPS)	87.5	35.3	70.3	65.8	27.3

(a) 単一ノード性能

システム	新 A(H)	新 B	D	E
使用ノード数	8	8	8	4
プロセス数 × スレッド数	32 × 17	16 × 18	16 × 14	24 × 10
測定値 (GFLOPS)	410	437	379	42.0

(b) 複数ノード性能

表 7: Poisson の性能測定結果

システム	新 A(H)	新 A(F)	新 B	D	E
スレッド数	128	128	36	28	120
測定値 (Mpoints/s)	537	627	352	304	81.7

(a) 単一ノード性能

システム	新 A(H)	新 B	D	E
使用ノード数	8	8	16	8
プロセス数	512	288	448	480
測定値 (Gpoints/s)	2.87	2.07	3.91	0.228

(b) 複数ノード性能

の割り当てより 20%程度高い性能が得られた。

## (e) IMB

測定結果を表 8 に示す。表 1, 表 2 が示す通り, Camphor 2, Magnolia, Camellia のスペック上の通信性能に大きな差はないが, 実測性能は Magnolia > Camphor 2 > Camellia の順となっている。これは, 通信性能が CPU コアの性能に律速されているためだと考えられる。通信性能をフルに使い切るためには, 1 ノード内で複数プロセスが同時に通信するようにするなど

の方策が必要となると考えられる。Omni Path を採用している Laurel 2 では一対一の通信性能は Magnolia より高いが, 集団通信では Magnolia をやや下回る性能となっている。このベンチマークは MCDRAM のメモリ容量内で動作する。Camphor 2 の Flat モード実行では, Alltoall を除いて Cache モード実行よりおおむね良好な性能が得られている。

## (f) STREAM

測定結果を表 9 に示す。Camphor 2 では, メモ



表 8: IMB の性能測定結果

システム	新 A(H)	新 A(F)	新 B	D	E
ノード数 × プロセス数	8 × 1				
PingPong 4MB (MB/s)	5055	5637	11416	9165	3604
PingPong 8B (MB/s)	1.97	1.90	4.41	3.94	0.5
Sendrecv 4MB (MB/s)	8519	9595	14628	11494	3673
Sendrecv 8B (MB/s)	1.57	1.90	7.87	5.62	0.63
Allreduce 4MB ( $\mu$ s)	11378	11668	2509	2504	14050
Reduce 4MB ( $\mu$ s)	6340	5641	2593	4355	47976
Reduce-Scatter 4MB ( $\mu$ s)	4412	8467	1264	1310	7138
Allgather 4MB ( $\mu$ s)	10175	9933	4134	5148	17034
Alltoall 4MB ( $\mu$ s)	25310	9351	5837	6636	17587
Bcast 4MB ( $\mu$ s)	1687	5650	991	1445	3288

表 9: STREAM の性能測定結果

システム	新 A(H)	新 A(F)	新 B	D	
スレッド数	64	64	36	28	
性能値 (copy, GB/s)	$S = 0.3$ GB	275	436	114	108
	$S = 3$ GB	202	442	115	—
	$S = 14.9$ GB	62.5	120	116	—
	$S = 22.4$ GB	59.0	106	116	—

(S : 配列サイズ)

リアクセスが MCDRAM のみで完結するようにした場合、DDR4 メモリもアクセスされる場合との性能を比較するため、いくつかの配列サイズを試した。予想通り、小さな配列サイズでは MCDRAM による高いアクセス性能が得られるが、サイズを大きくすると性能が落ち込んでしまうことが確認できる。特に Flat モード実行は、小さい配列に対しては MCDRAM のみのアクセスで完結するため、非常に高い性能が得られている。このベンチマークプログラムには参照局所性がほとんどないため、Cache モードはほとんど有効に機能せず、大きいサイズの配列に対しても Flat モードを下回る性能となっている。Laurel 2 のメモリアクセス性能は配列サイズによらずほぼ一定であり、かつ Magnolia よりやや高い性能が得られることが確認できる。

## 5 おわりに

本稿では、メディアセンターの新しいスパコンと旧スパコンにおけるいくつかのベンチマークの測定結果を示し、システム間の実性能を比較した。Camphor 2 (KNL) のコアあたりの性能は Laurel 2 (Broadwell)

や Magnolia (Haswell) を下回るが、ベクトル化率を高めたプログラムを多数のプロセス・スレッドを使って並列実行すれば、これらのシステムより高いノードあたり性能を得ることができる。ベクトル化率や並列度が十分でなかったり、MPI プロセス数を増やしすぎた場合には性能が落ち込んでしまう。ただし、KNL におけるそのような性能の落ち込みは KNC ほど極端ではなく、より使いやすいメニーコアプロセッサになっていると評価できる。また、ネットワークの通信性能を使い切れるようにすることや、MCDRAM のメモリモードの適切選択を含めた有効活用も Camphor 2 の性能を生かし切るための重要なポイントである。

Laurel 2 は、旧システムの Magnolia とほぼ同様の性能の傾向を示しつつ、実アプリケーションではおおむね Magnolia より高い性能が得られることが確認できた。

本稿がシステムの利用検討およびアプリケーション開発の一助になれば幸いである。

## 参考文献

- [1] Innovative Computing Laboratory, University of Tennessee: HPC Challenge. <http://icl.cs.utk>.

edu/hpcc/.

- [2] Intel Corporation: Getting Started with Intel MPI Benchmarks. <https://software.intel.com/en-us/articles/intel-mpi-benchmarks>.
- [3] Kawai, M., Iwashita, T. and Nakashima, H.: *SIMD Implementation of a Multiplicative Schwarz Smoother for a Multigrid Poisson Solver on an Intel Xeon Phi Coprocessor*, Springer International Publishing, pp. 57–65 (2015).
- [4] Kawai, M., Iwashita, T., Nakashima, H. and Marques, O.: *Parallel Smoother Based on Block Red-Black Ordering for Multigrid Poisson Solver*, Springer Berlin Heidelberg, pp. 292–299 (2013).
- [5] McCalpin, J. D.: STREAM: Sustainable Memory Bandwidth in High Performance Computers. <http://www.cs.virginia.edu/stream/>.
- [6] Minami, T., Hibino, M., Hiraishi, T., Iwashita, T. and Nakashima, H.: *Automatic Parameter Tuning of Three-Dimensional Tiled FDTD Kernel*, Springer International Publishing, pp. 284–297 (2015).
- [7] TOP500.org: Top500 Supercomputer Sites. <https://www.top500.org>.
- [8] 平石拓: 新スーパーコンピュータ Camphor 2 のベンチマーク評価報告, 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. 15, No. 1, pp. 15–18 (2017).

## システム A 運転状況 (2017 年 4 月 ~ 2017 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

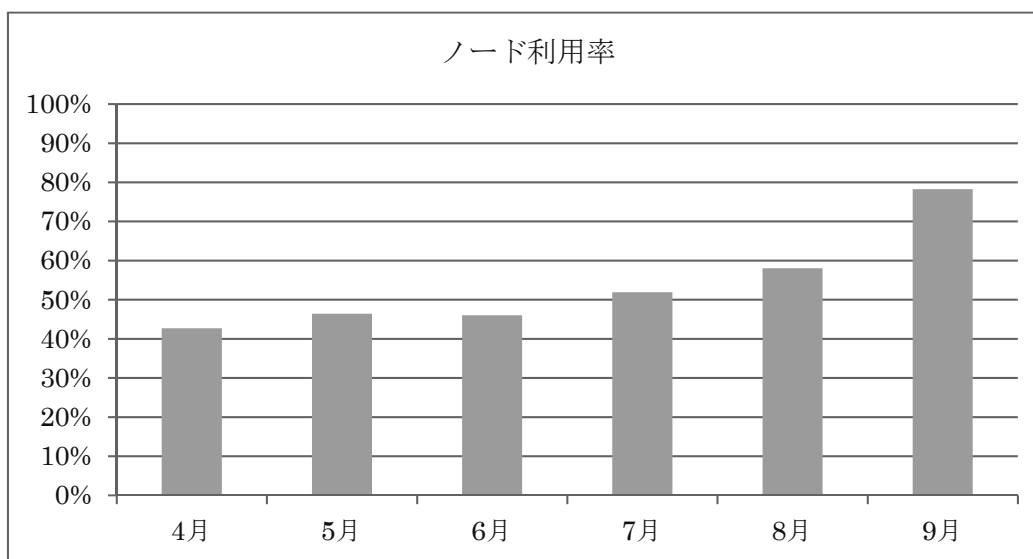
保守開始日時	サービス再開日時	保守時間[h]
2017/04/01 0:00	2017/04/05 9:10	105.17
2017/04/14 12:30	2017/04/14 17:30	5.00
2017/06/06 9:00	2017/06/07 14:00	29.00
2017/08/02 9:00	2017/08/09 9:30	168.50
2017/08/11 0:00	2017/08/11 7:00	7.00

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]
2017/04/10 21:20	2017/04/11 10:00	12.67
2017/04/13 17:50	2017/04/14 0:20	6.50
2017/05/01 2:20	2017/05/01 6:15	3.92
2017/05/07 22:08	2017/05/08 16:44	18.60
2017/05/09 23:22	2017/05/09 23:38	0.27
2017/08/01 16:55	2017/08/01 18:01	1.10

### 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	590.67	13,520	75,718	27,264,100	18,764,300	1795.8	43 %
5 月	721.22	38,025	123,574	40,041,600	30,422,000	1798.2	46 %
6 月	691.00	14,630	84,279	37,801,700	30,499,500	1790.2	46 %
7 月	744.00	13,989	117,107	46,980,900	38,713,400	1798.4	52 %
8 月	567.40	12,522	95,899	41,236,900	40,592,300	1734.8	58 %
9 月	720.00	24,905	133,060	67,713,100	56,856,600	1799.6	78 %
計	4034.29	117,591	629,637	261,038,300	215,848,100	1,786.1	54 %



- 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 (2017 年 4 月 ~ 2017 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

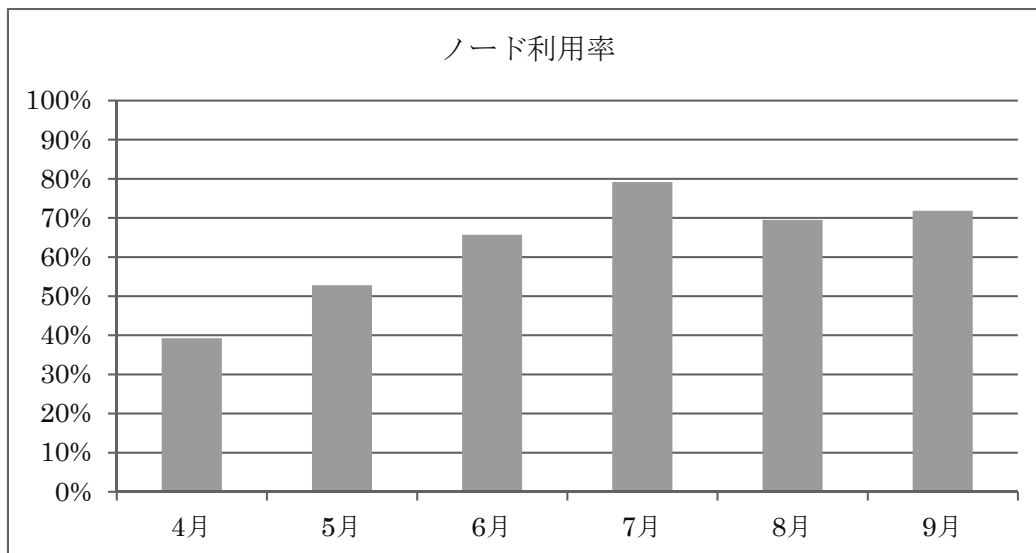
保守開始日時	サービス再開日時	保守時間[h]
2017/04/01 0:00	2017/04/05 9:10	105.17
2017/04/14 12:30	2017/04/14 17:30	5.00
2017/06/06 9:00	2017/06/07 14:00	29.00
2017/08/02 9:00	2017/08/03 9:30	24.50
2017/08/11 0:00	2017/08/11 7:00	7.00

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]
2017/05/01 2:20	2017/05/01 6:15	3.92
2017/08/01 16:55	2017/08/01 18:01	1.10

### 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]	平均稼働ノード数	ノード利用率
4月	609.83	97,424	493,282	6,093,890	5,102,350	796.0	39 %
5月	740.08	66,707	835,251	12,186,200	9,433,630	829.2	53 %
6月	691.00	62,721	921,452	10,562,900	9,225,890	791.6	66 %
7月	744.00	128,201	983,099	14,398,800	12,501,000	831.9	79 %
8月	711.40	97,968	872,477	11,021,700	9,353,030	823.6	69 %
9月	720.00	70,862	1,057,060	12,787,600	11,024,500	831.9	72 %
計	4216.31	523,883	5,162,621	67,051,090	56,640,400	817.4	63 %



- 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 (2017年4月～2017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

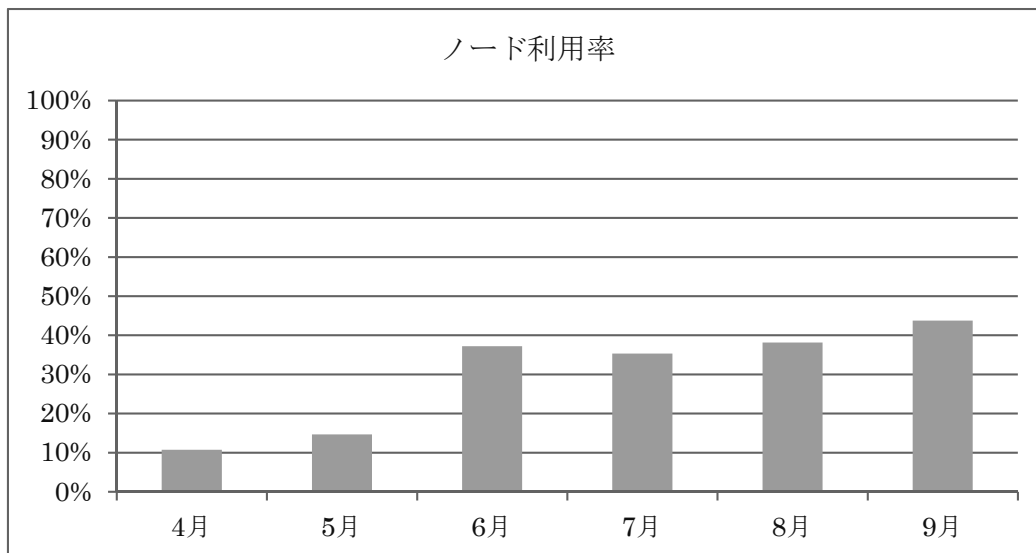
保守開始日時	サービス再開日時	保守時間[h]
2017/04/01 0:00	2017/04/05 9:10	105.17
2017/04/14 12:30	2017/04/14 17:30	5.00
2017/06/06 9:00	2017/06/07 14:00	29.00
2017/08/02 9:00	2017/08/03 9:30	24.50
2017/08/11 0:00	2017/08/11 7:00	7.00

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]
2017/05/01 2.20	2017/05/01 6:15	3.92
2017/08/01 16:55	2017/08/01 18:01	1.10

### 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	609.83	971	5,571	98,747	81,974	15.7	11 %
5月	740.08	1,213	13,282	126,511	109,568	15.7	15 %
6月	691.00	1,403	23,236	147,288	134,074	15.6	37 %
7月	744.00	1,614	21,164	101,793	77,640	15.8	35 %
8月	711.40	1,622	8,709	159,266	92,895	15.9	38 %
9月	720.00	1,079	12,635	200,865	72,977	16.0	44 %
計	4216.31	7,902	84,598	834,470	569,128	15.8	30 %



- 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 E 運転状況 (2017 年 4 月 ~ 2017 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

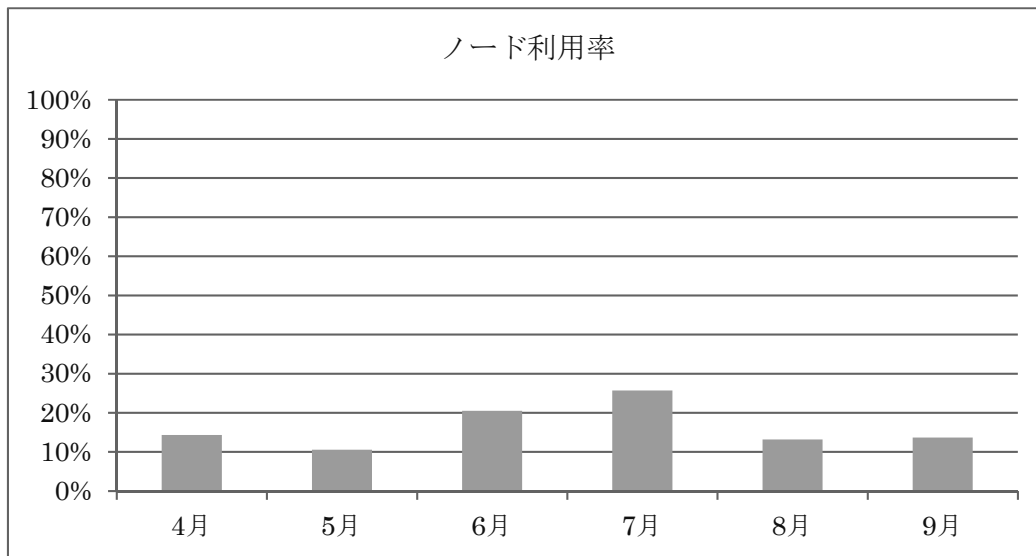
保守開始日時	サービス再開日時	保守時間[h]
2017/04/01 0:00	2017/04/05 9:10	105.17
2017/06/06 9:00	2017/06/07 14:00	29.00
2017/08/02 9:00	2017/08/03 9:30	24.50
2017/08/11 0:00	2017/08/11 7:00	7.00

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]
なし		

### 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	614.83	363	597	408,750	982	482.0	14 %
5月	744.00	97	1,215	350,092	1,976	482.0	11 %
6月	691.00	451	1,915	710,425	3,149	482.0	21 %
7月	744.00	141	2,527	1,025,580	4,637	482.0	26 %
8月	712.50	337	994	336,861	3,226	482.0	13 %
9月	720.00	49	1,233	545,959	1,886	482.0	14 %
計	4226.33	1,438	8,480	3,377,667	15,856	482.0	16 %



- 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 28 年度計算機利用結果報告書から—

- 【1002 ソフトウェア】
1. Tatsuya Abe, Tomoharu Ugawa, Toshiyuki Maeda, and Kousuke Matsumoto : Reducing State Explosion for Software Model Checking with Relaxed Memory Consistency Models : Proceedings of the 2nd International Symposium on Dependable Software Engineering: Theories, Tools and Applications, Vol.LNCS9984, pp.118-135, 2016
  6. Hirotaka Kitoh-Nishioka, Daisuke Yokogawa, and Stephan Irle: Forster Resonance Energy Transfer between Fluorescent Proteins: Efficient Transition Charge-Based Study : The Journal of Physical Chemistry C, Vol.121, No.8, pp.4220-4238, 2017
- 【4304 数理物理・物性基礎】
2. Yosuke Harashima, Kiyoyuki Terakura, Hiori Kino, Shoji Ishibashi and Takashi Miyake : First-principles study on stability and magnetism of NdFe<sub>11</sub>M(N) for M=Ti, V, Cr, Mn, Fe, Co, Ni, Cu, Zn : Journal of Applied Physics, Vol.120, pp.203964-1 - 203964-6, 2016
  7. Hirotaka Kitoh-Nishioka, Kai Welke, Yoshio Nishimoto, Dmitri G. Fedorov, and Stephan Irle : Multiscale Simulations on Charge Transport in Covalent Organic Frameworks Including Dynamics of Transfer Integrals from the FMO-DFTB/LCMO Approach : The Journal of Physical Chemistry C, Vol.121, No.33, pp.17712-17726, 2017
  3. Yosuke Harashima, Kiyoyuki Terakura, Hiori Kino, Shoji Ishibashi, and Takashi Miyake : Nitrogen as the best interstitial dopant among X = B, C, N, O, and F for strong permanent magnet NdFe<sub>11</sub>TiX: First-principles study : Physical Review B, Vol.92, pp.184426-1 - 184426-13, 2015
  8. Miyake, Y., H. Usui : Particle-in-cell modeling of spacecraft-plasma interaction effects on double-probe electric field measurements : Radio Science, Vol.51, pp.1905 - 1922, 2016
- 【4306 生物物理・化学物理】
4. Shun Sakuraba, Hidetoshi Kono : Spotting the difference in molecular dynamics simulations of biomolecules : The Journal of the Chemical Physics, Vol.145, pp.074116 - , 2016
  9. Yoshio Nishimoto and Dmitri G. Fedorov : Three-Body Expansion of the Fragment Molecular Orbital Method Combined with Density-Functional Tight-Binding : Journal of Computational Chemistry, Vol.38, No.7, pp.406 - 418, 2017
  5. Hirotaka Kitoh-Nishioka, Koji Ando : FMO3-LCMO study of electron transfer coupling matrix element and pathway: Application to hole transfer between two tryptophans through cis- and trans-polyproline-linker systems : The Journal of Chemical Physics, Vol.145, pp.114103 - , 2016
  10. M. Hatanaka, Y. Hirai, Y. Kitagawa, T. Nakanishi, Y. Hasegawa, K. Morokuma : Organic linkers control the thermosensitivity of the emission intensities from Tb(III) and Eu(III) in a chameleon polymer : Chemical Science, Vol.8, pp.423-429, 2016
  11. Yoshio Nishimoto : Analytic Hyperpolarizability and polarizability

- derivative with fractional occupation numbers for large extended systems : The Journal of Chemical Physics, Vol.146, No.8, pp.084101-, 2017
- 【4704 機能物質化学】
12. D. V. Konarev, Y. Nakano, S. S. Khasanov, A. V. Kuzmin, M. Ishikawa, A. Otsuka, H. Yamochi, G. Saito, R. N. Lyubovskaya : Magnetic and optical properties of layered (Me4P+)[M(IV)O(Pc•3-)]•- (TPC)0.5 C6H4Cl2 salts (M = Ti and V) composed of  $\pi$ -stacking dimers of titanyl and vanadyl phthalocyanine radical anions : Cryst. Growth Des., Vol.17, No.2, pp.753-762, 2016
  13. D. V. Konarev, S. I. Troyanov, A. V. Kuzmin, Y. Nakano, M. Ishikawa, M. A. Faraonov, S. S. Khasanov, A. L. Litvinov, A. Otsuka, H. Yamochi, G. Saito, R. N. Lyubovskaya : The Salts of Copper Octafluoro- and Hexadecafluorophthalocyanines Containing [CuII(F8Pc)4-]2- Dianions and [CuF16Pc]- Monoanions : Inorg. Chem., Vol.56, No.4, pp.1804-1813, 2017
  14. D. V. Konarev, S. S. Khasanov, M. Ishikawa, Y. Nakano, A. Otsuka, H. Yamochi, G. Saito, R. N. Lyubovskaya : Tetrabutylammonium salts of aluminum(III) and gallium(III) phthalocyanine radical anions bonded with fluoren-9-olato- anions and indium(III) bromide phthalocyanine radical anions : Chem. Asian J., Vol.12, 2017
  15. 中野義明, 大江佳毅, 石川学, 矢持秀起, 売市幹大 : 3 回対称性ドナー分子を用いた電荷移動錯体の合成と物性評価 (ポスター発表) : 第 10 回分子科学討論会 2016 神戸, 2016
  16. 中野義明, 大江佳毅, 石川学, 矢持秀起, 売市幹大 : 3 回対称性分子と TCNQ からなる電荷移動錯体の作製と物性評価 (ポスター発表) : 京都大学物性科学センター第 15 回講演会・研究交流会, 2017
  17. Y. Nakano, Y. Oe, M. Ishikawa, H. Yamochi, M. Uruichi : Development of Organic Electronics Material Based on C3-Symmetric Molecule (ポスター発表) : 12th International Conference on Nano-Molecular Electronics (ICNME2016), 2016
- 【4801 機能材料・デバイス】
18. Genta Kawaguchi, Mitsuhiko Maesato, Tokutaro Komatsu, Tatsuro Imakubo, Andhika Kiswandhi, D. Graf, Hiroshi Kitagawa : Use of Halogen Bonding in a Molecular Solid Solution to Simultaneously Control Spin and Charge : Chemistry of Materials, Vol.28, No.20, pp.7276 - 7286, 2016
  19. Mikihiro Hayashi, Kazuya Otsubo, Mitsuhiko Maesato, Tokutaro Komatsu, Kuniyoshi Sugimoto, Akihiro Fujiwara, Hiroshi Kitagawa : An Electrically Conductive Single-Component Donor-Acceptor-Donor Aggregate with Hydrogen-Bonding Lattice : Inorganic Chemistry, Vol.55, No.24, pp.13027 - 13034, 2016
  20. Tokutaro Komatsu, Hirokazu Kobayashi, Kohei Kusada, Yoshiki Kubota, Masaki Takata, Tomokazu Yamamoto, Syo Matsumura, Katsutoshi Sato, Katsutoshi Nagaoka, Hiroshi Kitagawa : First-Principles Calculation, Synthesis and Catalytic Properties of Rh-Cu Alloy Nanoparticles : Chemistry - A European Journal, Vol.23, No.1, pp.57 - 60, 2016
  21. Natalia Palina, Osami Sakata, L. S. R. Kumara, Chulho Song, Katsutoshi Sato, Katsutoshi Nagaoka, Tokutaro Komatsu, Hirokazu Kobayashi, Kohei Kusada & Hiroshi Kitagawa : Electronic Structure Evolution with Composition Alteration of RhxCuy Alloy Nanoparticles : Scientific Reports, Vol.7, pp.41264-, 2017
- 【4904 応用物理学一般】
22. Tien-Hsiu Tsai and Ikuo Kanno : A simulation study on the influence of scattered X-rays in energy-resolved computed tomography : Journal of Nuclear Science and Technology, Vol.54, No.2, pp.205-212, 2016
- 【4905 工学基礎】



23. Shugo YASUDA: Monte Carlo simulation for kinetic chemotaxis model: an application to the traveling population wave : *Journal of Computational Physics*, Vol.330, pp.1022-1042, 2016
- 【5001 機械材料・材料力学】
24. 西川雅章, 内藤悠太, 金崎真人, 北條正樹: 炭素繊維熱可塑性複合材料の加熱成形プロセスに関する熱伝導特性と粘弾性変形の連成解析: *JCOM 若手シンポジウム 講演論文集*, 2016
25. Masaaki Nishikawa, Akira Fukuzo, Naoki Matsuda, Masaki Hojo : Load-Transfer-Based Micromechanical Simulation for Evaluating Elastic-Plastic Response of Discontinuous Carbon Fiber Reinforced Thermoplastics : *Proc. of 31st Technical Conference of American Society for Composites*, 2016
26. 古賀貢史, 新玉重貴, 西川雅章, 北條正樹, 松田直樹: ボイドを含む CFRP の層間せん断試験の樹脂の降伏を考慮したモデル化: 第 41 回複合材料シンポジウム 講演論文集, 2016
- 【5004 流体工学】
27. Abhishek L. Pillai, Masaya Muto, Ryoichi Kurose : Numerical investigation of effect of Reynolds number on noise from turbulent non-premixed hydrogen jet flames : In *Proc. of the Asian Congress on Gas Turbines (ACGT2016)*, pp.USB (9 pages) -, 2016
- 【5202 構造工学・地震工学・維持管理工学】
28. Kyohei Noguchi, Soichiro Hata, Hiromichi Shirato, Tomomi Yagi : Effect of de-icing salts scattered by vehicle running : *Proceedings of the Twenty-ninth KKHTCNN Symposium on Civil Engineering*, pp.525-528, 2016
29. 野口恭平, 秦聡一郎, 白土博通, 八木知己: 車両走行による凍結防止剤の飛散がもたらす周辺塩分環境の変化: *土木学会第 71 回年次学術講演会講演概要集*, pp.91-92, 2016
- 【5401 金属物性】
30. S. Toyoda, K. Fukuda, K. Horiba, M. Oshima, K. Kumagai, Y. Kumagai, F. Oba, Y. Uchimoto, and E. Matsubara : Ligancy-driven controlling of covalency and metallicity in a ruthenium two-dimensional system : *Chem. Mater.*, Vol.28, pp.5784 - 5790, 2016
31. Y. Kumagai, L. A. Burton, A. Walsh, and F. Oba : Electronic structure and defect physics of tin sulfides: SnS, Sn<sub>2</sub>S<sub>3</sub>, and SnS<sub>2</sub> : *Phys. Rev. Applied*, Vol.6, pp.014009-1 - 014009-14, 2016
32. Y. Hinuma, T. Hatakeyama, Y. Kumagai, L. A. Burton, H. Sato, Y. Muraba, S. Iimura, H. Hiramatsu, I. Tanaka, H. Hosono, and F. Oba : Discovery of earth-abundant nitride semiconductors by computational screening and high-pressure synthesis : *Nature Commun.*, Vol.7, pp.11962-1 - 11962-10, 2016
33. Y. Hinuma, G. Pizzi, Y. Kumagai, F. Oba, and I. Tanaka : Band structure diagram paths based on crystallography : *Comput. Mater. Sci.*, Vol.128, No., pp.140 - 184, 2017
34. H. Hayashi, S. Katayama, T. Komura, Y. Hinuma, T. Yokoyama, K. Mibu, F. Oba, and I. Tanaka : Discovery of a novel Sn(II)-based oxide  $\beta$ -SnMoO<sub>4</sub> for daylight-driven photocatalysis: *Adv. Sci.*, Vol.4, pp.1600246-1 - 1600246-8, 2017
35. Y. Hinuma, Y. Kumagai, I. Tanaka, and F. Oba : Band alignment of semiconductors and insulators using dielectric-dependent hybrid functionals: Toward high-throughput evaluation : *Phys. Rev. B*, Vol.95, pp.075302-1 - 075302-10, 2017
36. Y. Kumagai, K. T. Butler, A. Walsh, and F. Oba : Theory of ionization potentials of nonmetallic solids : *Phys. Rev. B*, Vol.95, pp.125309-1 - 125309-10, 2017
- 【5403 複合材料・物性】
37. Shinya Iwata : Influence of Humidity Treatment on Electrical Tree Propagation in Epoxy Resin : *IEEE Transactions on Dielectrics and Electrical Insulation*, Vol.23, No.5, pp.2556-2561, 2016



別表1 スーパーコンピュータシステム

コース	タイプ	セット	利用負担額	提供サービス									
				システム	バッチ	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	2.0	-				
	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)		2.0					
	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)		2.0					
	タイプE	基本	100,000 円/年	E	共有	最大2ノード相当((10コア、32GBメモリ+ 1MIC)×2)		2.0					
グループ	タイプA1	最小	230,000 円/年	A	優先	4ノード((68コア、16+96GBメモリ)×4)	336	16.0	8				
		追加単位	115,000 円/年			2ノード((68コア、16+96GBメモリ)×2)		8.0	4				
	タイプA2	最小	276,000 円/年		準優先	8ノード((68コア、16+96GBメモリ)×8)		19.2	16				
		追加単位	69,000 円/年			2ノード((68コア、16+96GBメモリ)×2)		4.8	4				
	タイプA3	最小	690,000 円/年		占有	8ノード((68コア、16+96GBメモリ)×8)		32.0	16				
		追加単位	345,000 円/年			4ノード((68コア、16+96GBメモリ)×4)		16.0	8				
	タイプB1	最小	240,000 円/年	B	優先	4ノード((36コア、128GBメモリ)×4)		336	16.0	8			
		追加単位	120,000 円/年			2ノード((36コア、128GBメモリ)×2)			8.0	4			
	タイプB2	最小	288,000 円/年		準優先	8ノード((36コア、128GBメモリ)×8)			19.2	16			
		追加単位	72,000 円/年			2ノード((36コア、128GBメモリ)×2)			4.8	4			
	タイプB3	最小	720,000 円/年		占有	8ノード((36コア、128GBメモリ)×8)			32.0	16			
		追加単位	360,000 円/年			4ノード((36コア、128GBメモリ)×4)			16.0	8			
	タイプC1	最小	150,000 円/年	C	優先	1ノード((72コア、3072GBメモリ)×1)			336	16.0	8		
		追加単位	150,000 円/年			1ノード((72コア、3072GBメモリ)×1)				16.0	8		
	タイプC2	最小	180,000 円/年		準優先	2ノード((72コア、3072GBメモリ)×2)				19.2	16		
		追加単位	90,000 円/年			1ノード((72コア、3072GBメモリ)×1)				9.6	8		
	タイプE1	最小	280,000 円/年		E	優先				4ノード((10コア、32GBメモリ+ 1MIC)×4)	336	16.0	8
		追加単位	140,000 円/年							2ノード((10コア、32GBメモリ+ 1MIC)×2)		8.0	4
	タイプE2	最小	336,000 円/年	準優先		8ノード((10コア、32GBメモリ+ 1MIC)×8)				19.2		16	
		追加単位	84,000 円/年			2ノード((10コア、32GBメモリ+ 1MIC)×2)				4.8		4	
タイプE3	最小	840,000 円/年	占有	8ノード((10コア、32GBメモリ+ 1MIC)×8)		32.0	16						
	追加単位	420,000 円/年		4ノード((10コア、32GBメモリ+ 1MIC)×4)		16.0	8						
大規模ジョブ	タイプA	最小	23,000 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	11,500 円/週(7日)			4ノード((68コア、16+96GBメモリ)×4)							
	タイプB	最小	24,000 円/週(7日)	B	占有	8ノード((36コア、128GBメモリ)×8)							
		追加単位	12,000 円/週(7日)			4ノード((36コア、128GBメモリ)×4)							
	タイプC	最小	15,000 円/週(7日)	C	占有	2ノード((72コア、3072GBメモリ)×2)							
		追加単位	7,500 円/週(7日)			1ノード((72コア、3072GBメモリ)×1)							
	タイプE	最小	28,000 円/週(7日)	E	占有	8ノード((10コア、32GBメモリ+ 1MIC)×8)							
		追加単位	14,000 円/週(7日)			4ノード((10コア、32GBメモリ+ 1MIC)×4)							
専用クラスタ	-	最小	720,000 円/年	B	-	8ノード((36コア、128GBメモリ)×8)	-	32.0		16			
		追加単位	360,000 円/年			4ノード((36コア、128GBメモリ)×4)		16.0		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.ENVI/IDL)およびプリポストウェアの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に規定する額の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	11,300 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	13,200 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ためには、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るシステム資源は、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	19.2	16	1,104,000 円/年
	12ノード(68コア、16+96GBメモリ)×12)	336	28.8	24	1,656,000 円/年
	16ノード(68コア、16+96GBメモリ)×16)	336	38.4	32	2,208,000 円/年
B	8ノード(36コア、128GBメモリ)×8)	336	19.2	16	1,152,000 円/年
	12ノード(36コア、128GBメモリ)×12)	336	28.8	24	1,728,000 円/年
	16ノード(36コア、128GBメモリ)×16)	336	38.4	32	2,304,000 円/年
C	2ノード(72コア、3072GBメモリ)×2)	336	19.2	16	720,000 円/年
	3ノード(72コア、3072GBメモリ)×3)	336	28.8	24	1,080,000 円/年
	4ノード(72コア、3072GBメモリ)×4)	336	38.4	32	1,440,000 円/年
E	8ノード(10コア、32GBメモリ+ 1MIC)×8)	336	19.2	16	1,344,000 円/年
	12ノード(10コア、32GBメモリ+ 1MIC)×12)	336	28.8	24	2,016,000 円/年
	16ノード(10コア、32GBメモリ+ 1MIC)×16)	336	38.4	32	2,688,000 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ーコースを、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。
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
- システム E : camellia.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>

## 編 集 後 記

本号では、昨年度 No.1 の記事執筆以降にサービス開始となったスーパーコンピュータサブシステム Laurel2, Cinnamon2 に加え、新たにユーザに機能が開放されたバーストバッファ IME と Camphor2 のメモリモード指定についての特集を行いました。2018 年度にはさらに Camphor2 のバーストバッファシステム DataWarp の機能開放も予定しており、本誌でも関連記事を公開予定です。これら最新の機能を使って性能を引き出すのは難しいところもありますが、本誌記事が少しでも参考になれば幸いです。

(副部長)

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. 16, No. 2

2018年 3月 30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. 075-753-7400  
<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 (副部長)

水谷 幸弘

南雲 円

熊谷 真由美

尾形 幸亮

高見 好男

表紙デザイン : 谷 卓司

(ティアンドティ・デザインラボ)

# 目次

## 【巻頭言】

- ・ Vol.16, No.2号の発刊に当たって ..... 深沢圭一郎 ..... 1

## 【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～IME編～ ..... 尾形幸亮, 池田健二, 疋田淳一 ..... 3
- ・ 流体系シミュレーションでのプロセス分散書き出しにおけるIMEの効果 ..... 深沢圭一郎 ..... 8

## 【性能評価報告】

- ・ 新スーパーコンピュータのベンチマーク評価報告 ..... 平石拓 ..... 11

## 【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 ..... 18
- ・ センター利用による研究成果(平成28年度) ..... 22

## 【資料】

- ・ 大型計算機システム利用負担金 別表 ..... 26
- ・ サービス利用のための資料一覧 ..... 29

## 【編集後記】

- ・ 編集後記、奥付 ..... 30