Vol.2017-HPC-162 No.6
2017/12/18

# Oakforest-PACSにおけるIO-500の評価

建部 修見[1,a)]

**概要**：ストレージ性能はアクセスパターンにより大きく性能が変わり，多くの尺度が考えられる．IO-500ベンチマークはHPCにおける典型的なアクセスパターンのベンチマーク集合で，性能値を幾何平均により一つのスコアとする．今後の標準的なIOベンチマークとして期待されるものである．本研究では，IO-500ベンチマークにより，Oakforest-PACSのストレージシステムを評価する．ファイルキャッシュシステムの評価では，個別ファイルの書込で746 GiB/s，単一ファイルの書込で600 GiB/sであった．IO-500のスコアは101.48であり，2017年11月の第1回目のリストで第1位であった．

## 1. はじめに

ストレージは理論最大性能だけで語られることが多く，その性能は実際のアプリケーションのI/O性能とはかけ離れたものとなっている．また，ストレージシステムの性能はアプリケーションの性能向上のために重要であるにもかかわらず，とくにスパコン調達においてストレージシステムの優先度はCPUやネットワークに比べあまり高くないことが多い．ストレージの研究開発のためにも，理想的なアクセスパターンによる最大性能の追求だけではなく，より実際のアプリケーションのI/Oに近いアクセスパターンやより性能に厳しいアクセスパターンにおける性能向上は重要であるなど，ストレージの性能については，これら様々な問題がある．

これらの問題解決の一助としてIO-500ベンチマークの策定がすすめられている．IO-500は，これまで2016年11月にソルトレイクシティーで開催されたSC16と2017年6月にフランクフルトで開催されたISC 2017において二度のBirds-of-a-Featherセッションが開催され，ベンチマークの意義や内容について議論された．2017年11月にデンバーで開催されたSC17において，第一回目のIO-500リストが公表された．

本研究では，このIO-500ベンチマークを紹介するとともに，Oakforest-PACSのストレージシステムの性能評価を行う．ストレージ性能はアクセスパターンにより大きく性能が変わり，多くの尺度が考えられる．IO-500ベンチマークはHPCにおける典型的なアクセスパターンのベンチマーク集合で，性能値を幾何平均により一つのスコアとする．Oakforest-PACSのストレージシステムは，並列ファイルシステムとファイルキャッシュシステムで構成される．それぞれのシステムにおいて，IO-500ベンチマークで評価することにより性能モデルが分かる．

本研究の貢献は以下のものである．

- IO-500ベンチマークの紹介を行う．
- Oakforest-PACSのストレージシステムのIO-500ベンチマークによる性能評価を示す．
- アプリケーションユーザに，ストレージ性能向上のための指針を与える．

## 2. 関連研究

HPCにおいてストレージ性能を評価するベンチマークとしては，IORとmdtestがよく用いられている．IOR[2]はInterleaved-Or-Randomの頭文字をとったベンチマークで，MPI並列プロセスによる並列I/Oのバンド幅を計測する．APIはPOSIX，MPI-IO，HDF5などさまざまな利用可能である．並列プロセスのそれぞれが個別のファイルをアクセスするFPPモード，同一の単一ファイルだが別々の部分をアクセスするSSPモードを持つ．mdtest[3]は，主にメタデータ性能を計測するベンチマークである．ファイルの作成，書込，読込，stat，削除などの性能を計測する．並列プロセスのそれぞれが異なるディレクトリをアクセスする個別ディレクトリモード，同一ディレクトリをアクセスする共有ディレクトリモードを持つ．mdtestは2017年6月からIORと同じコードベースとなった．

## 3. IO-500ベンチマーク

IO-500ベンチマーク[1]は，**表1**に示されるように，バンド幅を計測する4種類のベンチマークと，メタデータ性

[1] 筑波大学計算科学研究センター
[a)] tatebe@cs.tsukuba.ac.jp

**表 1** IO-500 ベンチマーク

| | | |
|---|---|---|
| バンド幅 | 個別ファイル | 書込 ($B_1$)，読込 ($B_3$) |
| | 単一ファイル | 書込 ($B_2$)，読込 ($B_4$) |
| メタデータ性能 | 個別ディレクトリ | 0 Byte ファイル作成 ($M_1$)，stat($M_4$)，削除 ($M_6$) |
| | 単一ディレクトリ | 3,901 Byte ファイル作成 ($M_2$)，stat($M_5$)，読込 ($M_7$)，削除 ($M_8$) |
| | find | 10 分以内のファイル名に 01 を含むファイル検索 ($M_3$) |

能を計測する 8 種類のベンチマークで構成される．

バンド幅を計測するためのプログラムとしては IOR を用いる．バンド幅計測では大きく二種類あり，個別ファイルに対するものと単一ファイルに対するものがある．個別ファイルに対するものは，並列プロセスの各プロセスがそれぞれ個別のファイルをアクセスするパターンである．読込は，書込プロセスとは違うランクのプロセスが読込む．書込サイズは定められないが，5 分以上書込むことが条件となる．こればバッファキャッシュなどの影響を排除するためである．このパターンは，プロセス数に応じてファイル数が増えてしまうため，プロセス数が多くなったときに問題が起こるが，HPC ではよく用いられるパターンである．ストレージシステムにとり，比較的性能をだしやすいパターンであるため，IOR Easy とよばれる．

単一ファイルに対するものは，並列プロセスが単一ファイルの別々の場所をアクセスするパターンである．しかも，47,008 Byte のブロックサイズでのアクセスであり，ストレージのブロックサイズとは整合しない．読込は，書込プロセスとは違うランクのプロセスが読込む．こちらも，書込サイズは定められないが 5 分以上書込むことが条件となる．このパターンは，プロセス数が多くなっても 1 ファイルしか生成しないが，ストレージシステムにとり性能をだすのが難しいパターンである．そのため，IOR Hard とよばれる．

メタデータ性能を計測するためのプログラムとしては主に mdtest を用いる．こちらは大きく三種類あり，個別ディレクトリに対するもの，単一ディレクトリに対するもの，そして find である．個別ディレクトリに対するものでは，各プロセスがそれぞれ個別のディレクトリに対し，0 Byte ファイルの作成，stat，削除を行う．作成ファイル数は定められないが，ファイル作成を 5 分以上行うことが条件となる．このパターンはディレクトリ単位に並列にアクセスするため比較的性能をだしやすいパターンである．そのため，MDT Easy とよばれる．

単一ディレクトリに対するものは，並列プロセスが同一ディレクトリに対し，3,901 Byte ファイルの作成，stat，読込，削除を行う．こちらも作成ファイル数は定められず，ファイル作成を 5 分以上行うことが条件となる．このパターンは，同一ディレクトリに対して，並列プロセスが並列に異なるファイルを作成するものであり，ストレージシステムにとり性能をだすのが難しいパターンである．そのため，MDT Hard とよばれる．

find は，IOR Easy，IOR Hard，MDT Easy，MDT Hard で作成されたファイルを対象に，10 分以内に作成され，ファイル名に 01 を含むファイルを見つけるベンチマークである．プログラムは自由に作成してもいいが，MPI で並列化された並列 find のプログラムが提供されている．

IO-500 ベンチマークは計 12 種類のベンチマークの計測からなる．IOR Easy，MDT Easy，IOR Hard，MDT Hard の書込，ファイル生成のベンチマークが順にそれぞれ 5 分以上実行され，引き続き，find，IOR Easy の読込，MDT Easy の stat，IOR Hard の読込，MDT Hard の stat，MDT Easy の削除，MDT Hard の読込，削除が順に実行される．

スコアは，以下の式のようにバンド幅の幾何平均とメタデータ性能の幾何平均の幾何平均で計算される．

$$\sqrt{\sqrt[4]{B_1 B_2 B_3 B_4}\sqrt{M_1 M_2 \cdots M_8}}$$

## 4. Oakforest-PACS のストレージシステム

ストレージシステムの性能，容量は CPU 性能にバランスさせる必要がある．Oakforest-PACS のピーク演算性能は 25 PFLOPS であるため，経験則的には 25 PB ほどの容量が必要と考えられる．また，ストレージ性能は，チェックポインティングや各計算ステップでデータ書込性能が目安とされる．Oakforest-PACS の総メモリ容量は 897 TiB であり，ストレージへの書込を 10 分で行うためには，1.5 TiB/s の性能が必要となる．また，計算ノード数は 8,208 ノードであり，各ノード 4 プロセスで 32,832 プロセスのファイル生成を 1 秒で完了させるためには，ファイル作成性能は 32,832 io/s が必要となる．

1.5 TiB/s のストレージバンド幅を達成するのはそんなに簡単なことではない．ハードディスクドライブ (HDD) のデータ転送レートは 200 MiB/s ほどであり，1.5 TiB/s を達成するためには 7,500 台の HDD が必要となる．8 TB の HDD で構成すると，総容量は 60 PB となり，必要以上の容量となってしまう．しかも，この性能はデータ転送レートのピーク性能であり，アクセスパターンによってはこの性能を出すことは難しい．各プロセスが個別ファイルをアクセスする N-N アクセスパターンでは比較的容易であるが，各プロセスが単一ファイルの異なる場所をアクセスする N-1 アクセスパターンでは，ピーク性能に近い性能を出すことは難しい [4], [7]．そのため，Oakforest-PACS では，並列ファイルシステムとアプリケーションの間に，

ファイルキャッシュシステムを導入した．ファイルキャッシュシステムはバーストバッファともよばれ，他システムでも導入が進んでいる [6].

Oakforest-PACS のストレージ構成を**表 2** に示す．ファイルシステムキャッシュシステムの容量は，ほぼ総メモリ容量程となっている．この容量は十分な容量とはいえないが，必要最小限ではあるといえる．また，必要なバンド幅は，ファイルキャッシュシステムに対しては確保している．並列ファイルシステムの容量はほぼ必要とされる容量である．

ファイルキャッシュシステムは，DataDirect Networks (DDN) の Infinite Memory Engine (IME) を用い，25 台の IME14KX を用い構成している．各 IME14KX は 48 台の 800GB NVMe SSD を持ち，100 Gbps の Omni-Path ネットワーク 8 本で接続されている．各 NVMe SSD のデータ転送性能を 1,300 MB/s として，ファイルキャッシュシステム全体での物理ピークバンド幅は 1,560 GB/s となる．全体の物理容量は 960 TB であるが，10D+1P の erasure coding を行うため，利用可能な容量は 864 TB となる．

並列ファイルシステムは Lustre ファイルシステム [5] を用い，3 セットのメタデータサーバ (MDS) と 40 台のオブジェクトストレージサーバ (OSS) で構成される．各 MDS は 4 サーバと 26 台の 480GB SAS SSD で構成される．4D+2P の RAID6 構成で 2 台のホットスペアディスクも含まれるため，利用可能な容量は 23 TB である．Lustre ファイルシステムの i ノード当りの必要量を 2 KB と見積もると，約 11.5 億ファイルが格納可能である．OSS は 4,200 台の 8TB NL-SAS HDD で構成される．8D+2P の RAID6 構成で 100 台のホットスペアディスクが含まれるため，利用可能な容量は 26.24 PB である．各 OSS は 100 Gbps の Omni-Path で接続されるため，物理的なピークバンド幅は 500 GB/s となる．

ファイルキャッシュシステムで用いている IME は，並列ログファイルシステムベース [4] の並列ファイルシステムとアプリケーションの間にある中間層である．計算ノードからは，アプリケーションは POSIX あるいは MPI-IO でアクセス可能である．ファイルキャッシュシステムと並列ファイルシステムの間のデータ転送は，バッチキューイングシステムによるステージイン，ステージアウトの他，クライアントコマンドによる操作も可能である．アプリケーション実行中に一時的に生成される一時ファイルは，並列ファイルシステムにステージアウトしないで，バッチキューイングシステムがファイルキャッシュシステム上のファイルをリリース（削除）する．

ファイルキャッシュシステムと並列ファイルシステムは，マウントポイントは異なるが，同じ名前空間を持つ．ステージインしていないファイルをアクセスした場合は，エラーとはならず，並列ファイルシステムから読込まれる．

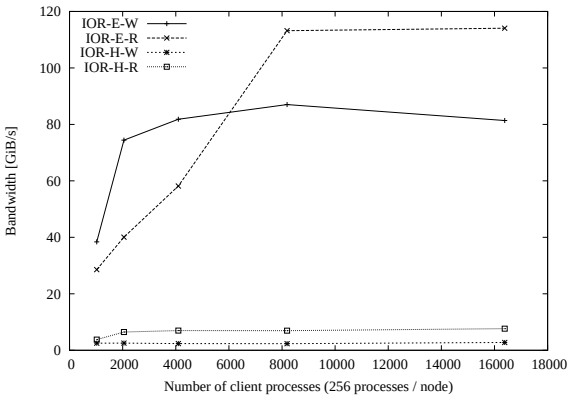


**図 1** 並列ファイルシステムのバンド幅.

## 5. IO-500 ベンチマークによる性能評価

Oakforest-PACS のストレージシステムは並列ファイルシステムとファイルキャッシュシステムで構成される．それぞれのシステムについて IO-500 ベンチマークによる性能評価を行う．

### 5.1 並列ファイルシステム

**図 1** に IOR Easy，IOR Hard の結果を示す．プロセス数が 4,096 までは IOR Easy の書込性能が読込性能に比べて高いが，8,192 プロセスでは逆転している．書込は 8,192 プロセスの時に 87 GiB/s，読込は 16,384 プロセスの時に 114 GiB/s であった．IOR Easy に比べ，IOR Hard の性能は低い．IOR Hard の書込性能は 2.7 GiB/s，読込性能は 7.6 GiB/s であり，一桁以上の性能差がある．この性能差があるためファイルキャッシュシステムが必要となる．

**図 2** にメタデータ性能を示す．find の性能は 8,192 プロセスまでは高いが，16,384 プロセスで急激に下がってしまっている．それ以外のメタデータ性能は 2,048 プロセス以上ではほぼ変わらない．MDT Hard では，単一ディレクトリに 3,901 Byte のファイルを作成するが，Lustre ファイルシステムの制限で，単一ディレクトリには約 850 万ファイルまでしか作成できない．IO-500 の規程で 5 分以上作成する必要があるが，28.3 Kio/s 以上の性能ではその制限を越えてしまう．Oakforest-PACS の並列ファイルシステムはこの性能を越えるため，図 2 に示した MDT Hard のファイル作成は 5 分に満たないものである．そのため IO-500 の条件を満たすことができない．

### 5.2 ファイルキャッシュシステム

ファイルキャッシュシステムに対するアクセスは POSIX と MPI-IO が可能である．MPI-IO でのアクセスは，その下位層で，ファイル名に ime:をつけてアクセスする ROMIO ドライバと，POSIX を用いる方法がある．しかしながら，IO-500 の変更不可のスクリプト中に POSIX によるアクセ

**表 2** Oakforest-PACS のストレージ構成

| | 容量 (TB) | 物理バンド幅 (GB/s) |
|---|---|---|
| ファイルキャッシュシステム | 864 | 1,560 |
| 並列ファイルシステム | 26,240 | 500 |

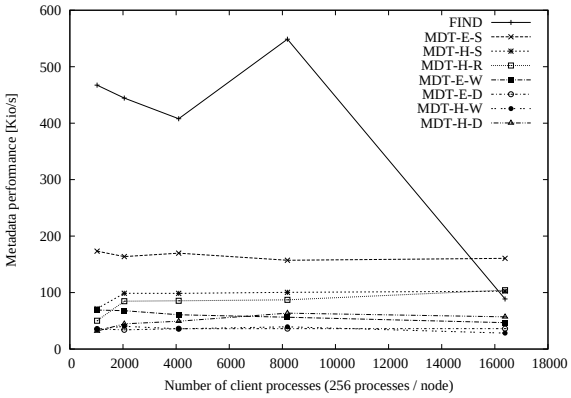


**図 2** 並列ファイルシステムのメタデータ性能.

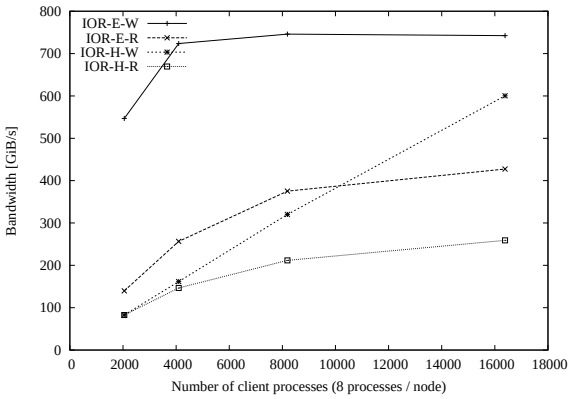


**図 3** ファイルキャッシュシステムのバンド幅.

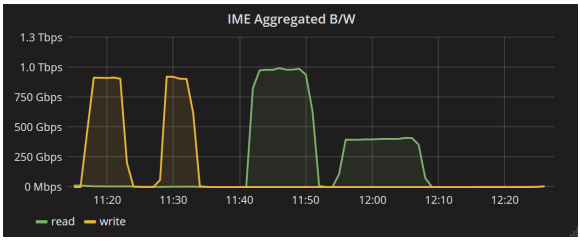


**図 4** IME パフォーマンスモニタによるファイルキャッシュシステムのバンド幅.

スがあるため，ime:をつけたディレクトリ名の指定はできない．従って，ROMIO ドライバは利用できず，POSIX でアクセスしている．

**図 3** にファイルキャッシュシステムのバンド幅を示す．IOR Easy の書込は 4,096 プロセスでほぼさちっており，8,192 プロセスの時は 746 GiB/s であった．一方で，IOR Easy の読込は半分程度の性能となっている．この読込の性能低下についての原因を探るため，IME のパフォーマンスモニタを**図 4** に示す．

11:15 頃から IOR Easy の書込，11:28 頃から IOR Hard の書込が行われ，11:40 すぎから IOR Easy の読込が行われている．グラフを見る限り，読込の方が高いバンド幅を示している．一方で，書込に対し，読込の方が面積が広く，書込データより多く読込んでいることが分かる．読込では readahead による先読みが行われるが，この先読みデータがクライアント側のバッファ不足などの原因で捨てられていると考えられる．

IOR Hard では，書込は 16,384 プロセスまでスケールし，600 GiB/s を達成している．読込はブロックサイズが小さいため IOPS がネックとなり 259 GiB/s となっている．

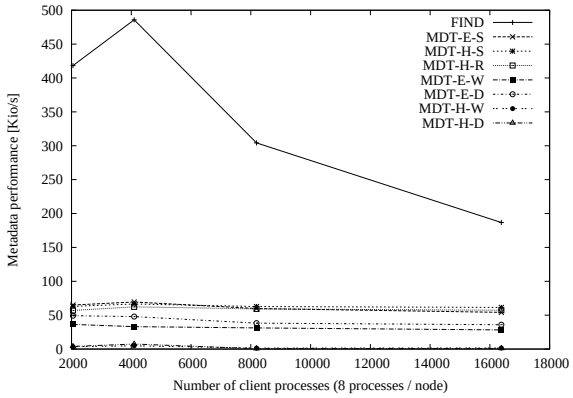


**図 5** ファイルキャッシュシステムのメタデータ性能.

**図 5** にファイルキャッシュシステムのメタデータ性能を示す．ファイルキャッシュシステムの名前空間は並列ファイルシステムと同じであり，並列ファイルシステムのメタデータサーバを用いていることから，メタデータ性能は並列ファイルシステムよりは高くならない．並列ファイルシステムの性能と比べると，find の性能はほぼ変わらず，MDT Easy の作成は半分程，stat は 1/3 程，削除はほぼ変わらない性能である．一方で，MDT Hard の性能差は大きく，作成は 1/26 程，stat と読込は 2/3 程，削除は 1/70 程の性能となっている．MDT Hard については改善の余地が大きい．

IO-500 のスコアを**表 3**，**表 4** に示す．*のついた性能，スコアは 5 分以上という規程を満たしていないものである．並列ファイルシステムでは，規程を満たすことができず，どのスコアも公式なものではない．また，8,192 プロセスでほぼバンド幅はさちっている．

ファイルキャッシュシステムでは，16,384 プロセスまでバンド幅は向上している．この向上は，IOR Hard の書込性能がこのプロセス数まで向上していることによる．メタデータ性能は*のついた 5 分以内の計測では比較的性能が高いが，5 分を越えると下がる傾向がある．こちらも性能

**表 3** 並列ファイルシステムの IO-500 ベンチマークスコア

| プロセス数 | バンド幅 [GiB/s] | メタデータ性能 [Kio/s] | スコア |
|---|---|---|---|
| 1,024 | 10.10* | 73.05* | 27.17* |
| 2,048 | 14.88* | 84.23* | 35.40* |
| 4,096 | 16.76 | 83.05* | 37.30* |
| 8,192 | 20.04 | 88.78* | 42.18* |
| 16,384 | 21.04* | 67.19 | 37.60* |

**表 4** ファイルキャッシュシステムの IO-500 ベンチマークスコア

| プロセス数 | バンド幅 [GiB/s] | メタデータ性能 [Kio/s] | スコア |
|---|---|---|---|
| 2,048 | 151.31 | 35.22* | 73.00* |
| 4,096 | 257.52 | 41.01* | 102.76* |
| 8,192 | 371.12 | 22.80 | 92.00 |
| 16,384 | 471.25 | 21.85 | 101.48 |

改善の余地が大きいところである．

## 6. まとめ

IO-500 ベンチマークの紹介と，Oakforest-PACS による性能評価を行った．IO-500 ベンチマークは 2016 年から策定が進められ，2017 年 11 月に初めての IO-500 リストが公開された．ストレージ性能はアクセスパターンにより大きく性能が変わる．そのため，多くの尺度が考えられるが，IO-500 ベンチマークでは典型的なアクセスパターンの幾何平均により一つのスコアとして算出している．これにより，バンド幅性能，メタデータ性能のバランスがとれたストレージシステムの研究開発も進むと考えられる．2017 年 11 月のリストでは Oakforest-PACS のファイルキャッシュシステムが最も高速なシステムとなった．今後は，今回判明したファイルキャッシュシステムの性能問題の解決に努めるとともに，ストレージの性能モデルが明かとなったため，アプリケーションのストレージ性能向上に取り組んでいきたい．

**謝辞** 本研究の一部は JST-CREST JPMJCR1303「EBD：次世代の年ヨッタバイト処理に向けたエクストリームビッグデータの基盤技術」，JST-CREST JPMJCR1413「広域撮像探査観測のビッグデータ分析による統計計算宇宙物理学」，JSPS 科研費 JP17H01748 による．

### 参考文献

[1] *IO-500*. http://www.io500.org/.
[2] *IOR*. https://github.com/IOR-LANL/ior.
[3] *mdtest*. https://github.com/IOR-LANL/ior.
[4] Bent, J., Gibson, G., Grider, G., McClelland, B., Nowoczynski, P., Nunez, J., Polte, M. and Wingate, M.: PLFS: a checkpoint filesystem for parallel applications, *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, pp. 1–12 (online), DOI: 10.1145/1654059.1654081 (2009).
[5] Braam, P. J.: *Lustre File System*. http://www.lustre.org/.
[6] NERSC: *Burst Buffer Architecture and Software Roadmap*. http://www.nersc.gov/users/computational-systems/cori/burst-buffer/burst-buffer/.
[7] Nisar, A., k. Liao, W. and Choudhary, A.: Delegation-Based I/O Mechanism for High Performance Computing Systems, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 23, No. 2, pp. 271–279 (online), DOI: 10.1109/TPDS.2011.166 (2012).