富士通 PRIMEHPC FX10 チューニング連載講座 2. ファイルシステムとI/O性能

鴨志田良和
東京大学情報基盤センター

はじめに

富士通 PRIMEHPC FX10 (以下、FX10) では、全ての計算ノード、インタラクティブノードおよびログインノードからアクセスしてファイルを共有可能なストレージとして FEFS, Lustre による共有ファイルシステムを提供しています。「富士通 PRIMEHPC FX10 チューニング連載講座 第 2 回」では、FX10 で提供しているファイルシステムについて紹介し、そのI/O 性能について述べます。

FX10 で提供するストレージ環境

FX10 のストレージ環境は、共有ファイルシステム、ローカルファイルシステムと外部ファイルシステムから構成されます。共有ファイルシステムは、各ユーザーのホームディレクトリやグループコース用の共有ディレクトリを格納するファイルシステムであり、全計算ノード、インタラクティブノードおよびログインノードから参照可能です。利用可能容量は合計 2.1PByte です。ローカルファイルシステムは、計算ノードとインタラクティブノードでのみ参照可能なファイルシステムであり、利用可能容量は 1.1PByte です。ローカルファイルシステムを用いることで、ジョブ間の I/O 競合による影響を抑え、実行時間のブレを小さくすることが可能です。また、外部ファイルシステムは、全計算ノード、インタラクティブノードおよびログインノードから参照可能な Quota 制限のない共有ファイルシステムであり、一時的に生成される大容量のファイルを格納することが可能です。外部ファイルシステム保存されたファイルは、定期的に削除する予定です。共有ファイルシステムとローカルファイルシステムは FEFS、外部ファイルシステムは Lustre を使用して構築されています。

Lustre はオープンソースの並列ファイルシステムで、MDS (メタデータサーバ) と複数台の OSS (オブジェクト格納サーバ) に、並列にファイルを分散、管理することにより負荷分散を行い、高いレスポンスを実現します。複数のサーバにまたがる大規模ファイルを作成することも可能です。MPI などの並列アプリケーションからのデータ入出力など、多数のノードからの入出力を伴う処理に適しています。FEFS(Fujitsu Exabyte File System) は Lustre ファイルシステムをベースに富士通が開発したファイルシステムで、数万規模のクライアントによるファイル利用を想定した大規模分散ファイルシステムです。Lustre との互換性を維持しつつ、大規模システム向けに最大ファイルサイズ、最大ファイル数等の拡張を実施しています。

ファイルシステムの構成

共有ファイルシステム

共有ファイルシステムの主な構成は以下のとおりです。

- 4 組 (8 台) の MDS と 40 台の OSS
- 各 OSS は 2 本の InfiniBand 4xQDR でデータ系ネットワークと接続
- 計算ノード 96 台とインタラクティブノード 1 台あたり、2 台の I/O ノード (ローカルファイルシステムの I/O ノードと兼用) が接続 (合計 100 台)

- 各 I/O ノードからは 1 本の InfiniBand 4xQDR でデータ系ネットワークと接続
- ディスクアレイ装置として ETERNUS DX410 S2 を 80 台使用
- アレイ装置あたり、2 コントローラ、4G バイトのキャッシュを搭載しており、8Gbps FC ケーブル 4 本を接続
- ディスクには 600G バイト、10krpm の 2.5 インチ SAS ディスクを使用
- 480 個の RAID6 グループ (9D+2P)、スペアディスクは 160 個

全体の容量 2.1PB を 4 分割し、4 つの独立したファイルシステムとして運用しています。

ローカルファイルシステム

ローカルファイルシステムの主な構成は以下のとおりです。

- 1 組 (2 台) の MDS
- 計算ノード 96 台とインタラクティブノード 1 台あたり、3 台の I/O ノード (うち 2 台が共有ファイルシステムの I/O ノードと兼用) が接続 (合計 150 台)
- ディスクアレイ装置として ETERNUS DX80 S2 を 150 台使用
- アレイ装置あたり、2 コントローラ、4G バイトのキャッシュを搭載しており、8Gbps FC ケーブル 4 本を接続
- ディスクには 600G バイト、10krpm の 2.5 インチ SAS ディスクを使用
- 600 個の RAID5 グループ (4D+1P)、スペアディスクは 150 個

割り当てられるノード数に応じて 1 ノードあたり 240GiB のスクラッチ領域がジョブに割り当てられます。

外部ファイルシステム

外部ファイルシステムの主な構成は以下のとおりです。

- 1 組 (2 台) の MDS と 16 台の OSS
- ディスクアレイ装置として DDN SFA10000 を 4 台使用
- アレイ装置あたり、2 コントローラ、8G バイトのキャッシュを搭載しており、2 本の InfiniBand 4xQDR インタフェースに接続
- ディスクには 2T バイト、7200rpm の 3.5 インチ SATA ディスクを使用
- 236 個の RAID6 グループ (8D+2P)、スペアディスクは 40 個
- データ系ネットワークとの間は、40 本の InfiniBand 4xQDR で接続

全体を 3 分割し、3 つの独立したファイルシステムとして運用しています。

各ファイルシステムは表 1 で示されるマウントポイントにマウントされています。OST は Object Strage Target の略で、OSS から認識される物理的なストレージボリュームの単位で、FX10 のファイルシステムの構成では各 RAID グループが OST に対応しています。

表 1: 各ファイルシステムの容量

マウントポイント	種類	容量 *	OST 数	備考
/home	共有	541TiB	120	
/group1	共有	541TiB	120	
/group2	共有	541TiB	120	
/group3	共有	541TiB	120	
/mppxa	外部	1.5PiB	112	今後提供予定
/mppxb	外部	833TiB	62	
/mppxc	外部	833TiB	62	
/work	ローカル	1.1PiB	600	

* 容量は、フォーマット後のユーザが利用可能な容量

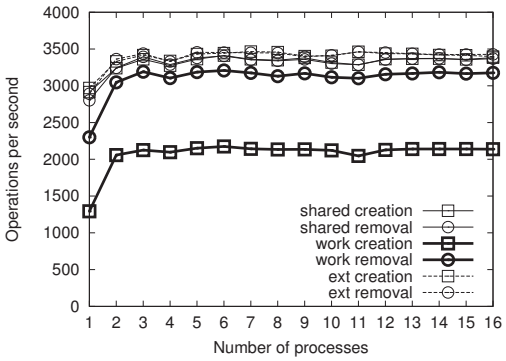

図 1: mdtest 1 ノード (ディレクトリ操作)

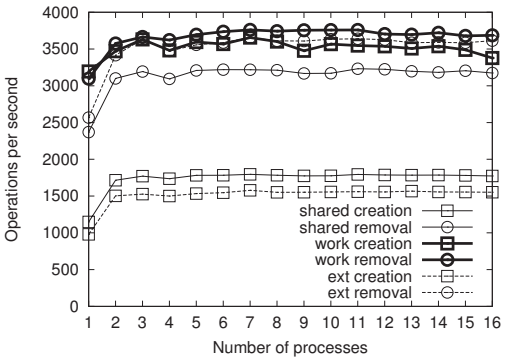

図 2: mdtest 1 ノード (ファイル操作)

性能評価

上に述べたそれぞれのファイルシステム上で、MDTEST ベンチマークと IOR ベンチマークを実行し、性能評価を行った結果を掲載します。これらのベンチマークは Lawrence Livermore National Laboratory の Livermore Computing Center が公開している I/O ベンチマーク[1]で、前者はメタデータアクセス性能、後者はブロック入出力のスループットを計測するものです。

クライアントのノードには、FX10 の計算ノードのうち、1 台から 16 台を使用しています。12 台までの計測では、各計算ノードはひとつの Tofu 単位に属し、データ系ネットワークに接続する I/O ノードを共有しています。13 台以上の計測では、ノードは 2 つの Tofu 単位に属しています。この 2 つの Tofu 単位は、I/O ノードを共有していません。なお、以下に示す各ベンチマークの結果は、運用中のシステム上で実行したものなので、同時に実行されていた他のジョブ等の影響を受けている可能性があります。共有ファイルシステムを使用した計測では`/group1` を、外部ファイルシステムを使用した計測では`/mppxb` を使用しています。

[1]Scalable I/O Benchmark Downloads, Lawrence Livermore National Laboratory:
`https://computing.llnl.gov/?set=code&page=sio_downloads`

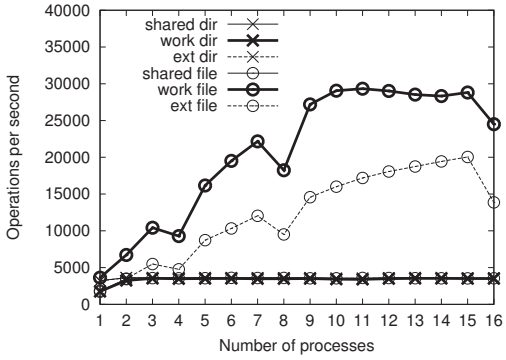

図 3: mdtest 1 ノード (stat)

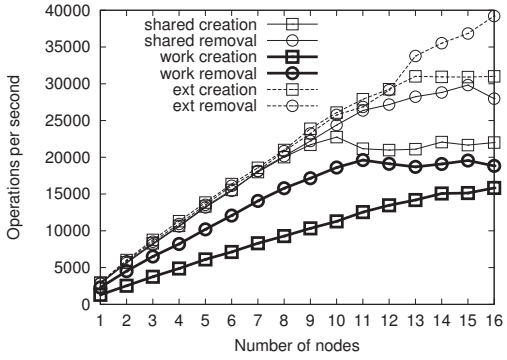

図 4: mdtest 複数ノード (ディレクトリ操作)

MDTEST ベンチマーク

MDTEST は多数のプロセスが一斉に共有ファイルシステムにアクセスし、一定の処理を行う時間から共有ファイルシステムのメタデータアクセス性能を測定するツールです。今回の性能評価では以下の条件で計測を行いました。

- 共有ファイルシステム (shared)・ローカルファイルシステム (work)・外部ファイルシステム (ext) のそれぞれについて測定
- ファイル・ディレクトリの作成 (creation)・削除 (removal)・存在チェック (stat) の速度を測定
- プロセスごとに上記の操作を 5,000 回ずつ実行
- プロセスごとに個別の作業ディレクトリを作成して処理を実行
- 10 回の測定を行い、平均値からアクセス速度を計算

図 1 から図 3 は 1 台の計算ノードで MDTEST を実行した結果です。横軸にはノードあたりのプロセス数を、縦軸にはアクセス速度 (Operations per second) を示しています。アクセス速度は、ファイル作成等の各操作の実行回数を全プロセスで合計した数を 1 秒あたりの値に正規化したものです。図 1 はディレクトリの作成・削除について、図 2 はファイルの作成・削除について、図 3 はディレクトリ・ファイルの存在チェックについて、それぞれ示しています。計算ノード 1 台の場合、ディレクトリ・ファイルの作成・削除の処理については、いずれのファイルシステムでも、2 プロセス以上でほぼ横ばいの性能になる事がわかります。存在チェックについても同様の結果ですが、ローカルファイルシステム (work) と外部ファイルシステム (ext) のファイル存在チェックだけはプロセス数が増加に従って性能も向上する傾向にあることがわかります。

一方、図 4 から図 6 はノードあたりのプロセス数を 1 に固定し、ノード数を変えて MDTEST を実行した結果です。図 4 はディレクトリ操作について、図 5 はファイル操作について、図 6 はディレクトリ・ファイルの存在チェックについて、それぞれ示しています。こちらは、どの操作についてもノード数増加に従って性能が向上する傾向にあり、ディレクトリ、ファイル操作については、10 台程度のところで性能向上が鈍化していることがわかります。

IOR ベンチマーク

IOR は多数のプロセスが一斉に共有ファイルシステム上のファイルを読み書きし、データ転送性能を測定するツールです。使用するファイルのプロセスへの割り当てについては、プロセスごとに別のファイルを

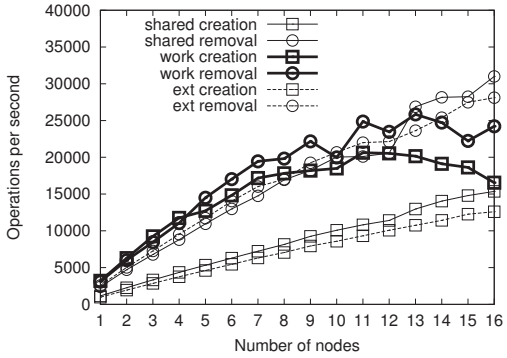

図 5: mdtest 複数ノード (ファイル操作)

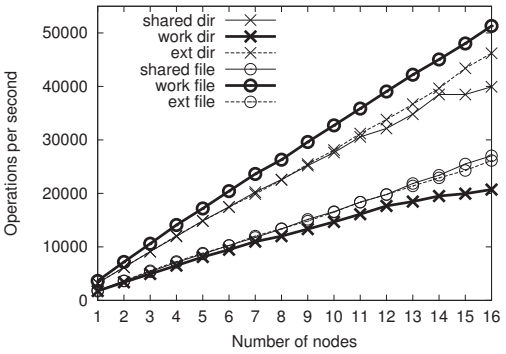

図 6: mdtest 複数ノード (stat)

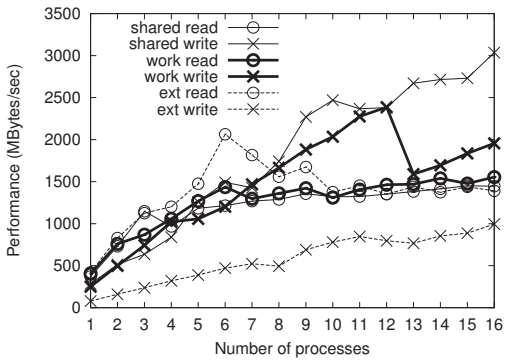

図 7: ior-multi 1 ノード

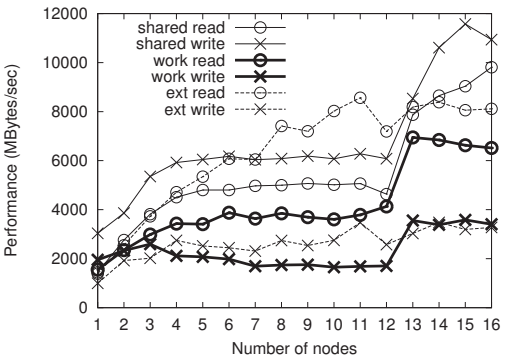

図 8: ior-multi 複数ノード

割り当てるか、単一ファイル内で、プロセスごとに別々の領域に割り当てるかを選択することができ、以下では前者を ior-multi, 後者を ior-single と呼ぶことにします。今回の性能評価では、この両者について以下の条件で計測を行いました。

- POSIX I/O を使用
- ファイルの読み込み、書き込みの性能を測定
- 読み書きのブロックサイズは 1M バイト
- プロセスごとに 5GB のファイルを出力
- 共有ファイルシステム (shared)・ローカルファイルシステム (work)・外部ファイルシステム (ext) のそれぞれについて測定。ただし、ローカルファイルシステムについては ior-single を計測しない
- ior-single の計測時は、ファイルに 4095MBytes のストライプサイズを設定、使用 OST 数は最大値 (共有ファイルシステムは 120、外部ファイルシステムは 62) を指定

実行結果を図 7 から図 10 までのグラフに示します。各グラフの縦軸にはデータ転送性能 (MB/sec) として、全プロセスが生成するファイルサイズの合計を、全プロセスがファイル読み込み・書き込みを完了するまでにかかる時間で割ったものを示しています。図 7 と図 9 は、ior-multi、ior-single それぞれについて、計算ノード 1 台でノードあたりのプロセス数を変化させて計測した結果です。ファイルシステムの種類

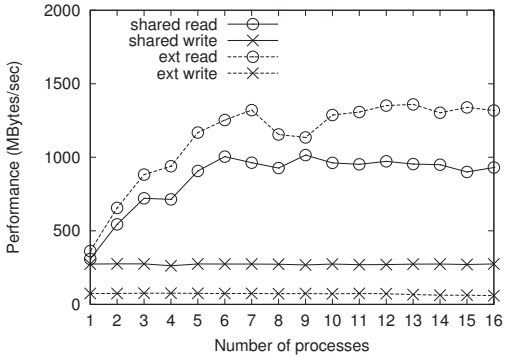

図 9: ior-single 1 ノード

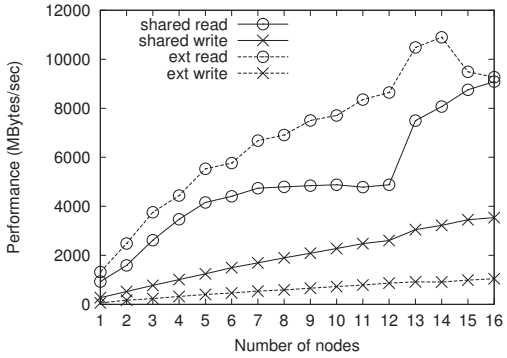

図 10: ior-single 複数ノード

にかかわらず、Read は似たような性能で、8 プロセス程度で性能が頭打ちになっていることがわかります。Write については、外部ファイルシステムよりも共有ファイルシステム・ローカルファイルシステムの性能が高いことがわかります。ローカルファイルシステムでは、13 プロセスを超えると性能が落ちています。1 台の計算ノードと接続している I/O ノードは 3 台ありますが、この I/O ノード群に接続している OST の数は 12 です。このため、13 プロセス以上の同時アクセスでは、OST へのアクセス競合が発生するため、性能が低下します。ior-multi と ior-single を比較すると、メタデータアクセスの競合などがあるため、ior-single の性能は ior-multi の性能と比較してかなり低くなる傾向にあるようです。ior-single の性能は、ストライプサイズを変える、使用する OST の数を減らすなどして性能が改善する可能性がありますが、今回は基本的な性能を紹介することにとどめます。

図 8 と図 10 は、ior-multi、ior-single それぞれについて、複数台の計算ノードを使用して実験した結果です。ノードあたりのプロセス数は 16 測定しました。ior-multi において、ローカルファイルシステムの性能が低いのは、計算ノード 1 台での実験のときと同様、使用可能な OST の数が 12(13 台以上の場合は 24) に制限されていることが原因であると考えられます。計算ノードが 12 台以下の場合と 13 台以上の場合で性能が大きく異なるのは、13 台以上の場合に使用される I/O ノードの数が、12 台までの場合と比較すると 2 倍になっているためです。ior-single は ior-multi と比較すると、Write の性能が低くなるようです。

ior-single の共有ファイルシステムと外部ファイルの比較では、Read については外部ファイルシステムのほうが、Write については共有ファイルシステムのほうが、それぞれ高い性能であることがわかりました。

おわりに

Oakleaf-FX の I/O 性能はプロセス数、ノード数、I/O ノード数、OST 数など、様々な要素がボトルネックとなる可能性があります。今回は 16 ノードまでの計測でしたが、共有ファイルシステムにおいてメタデータ操作では 3 万回/秒程度、ブロック書き込み性能では 10GByte/秒程度の性能を達成できることがわかりました。更に大規模な場合は、性能の傾向も異なってくる可能性があります。また、インタラクティブノードの場合も、性能の傾向が異なります。使用される I/O ノードの数や OST の数は、ジョブごとに異なる可能性があるので、高い I/O 性能を実現するにはこれらを意識して処理を行う必要があります。