HITACHI SR16000/M1 チューニング連載講座 3. 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

HITACHI SR16000 モデル M1(以下、SR16000) では、全ての計算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 General Parallel File System for AIX(以下、GPFS) による共有ファイルシステムを提供しています。「HITACHI SR16000/M1 チューニング講座 第3回」では、SR16000 で提供している GPFS について紹介し、その I/O 性能について述べます。

1　GPFS の概要

GPFS は複数のノードに接続されるディスクを一つの仮想ボリュームに見せることで、大容量のファイルシステムを実現することが可能です。このファイルシステムは外部のネットワークを経由せずに、各ノードからノード間を接続する高速なネットワークと FC による SAN を経由して直接参照可能です。GPFS は以下の特徴を持っています。

1. ファイル入出力の際は、データを一定のブロックに分割して複数の I/O ノードから並列にディスクアクセスすることで高い IO スループット性能を実現します。
2. 全ての I/O ノードが全てのディスクにアクセスできるため、単一障害点がありません。一台の I/O ノードがダウンした場合も、他の I/O ノードによりサービスを引き継ぐことが可能です。また、独立したメタ・データサーバが不要です。
3. ジャーナリングファイルシステムであり、ディスク障害時のリカバリ時間を短縮することが可能です。
4. ファイルシステムサイズは、理論的には 2^{99} バイトまで拡張可能です。
5. quota 機能をサポートしています。複数の物理ボリュームを統合し、一つのファイルシステムを作成することが可能です。また異なるノードからの競合アクセスを制御することが可能です。

2　共有ファイルシステムの構成

SR16000 の共有ファイルシステムの主な構成は以下のとおりとなっています。

- 4 台の I/O サーバ
- ディスクアレイ装置として Hitachi AMS2500 16 台 (32 コントローラ) を使用
- 各コントローラは 2G 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、15krpm の 3.5 インチ SAS ディスクを使用 (ホットスワップ可能)
- 139 個の RAID6 パリティグループ (7D+2P × 138, 2D+2P × 1)、スペアディスクは 67 台

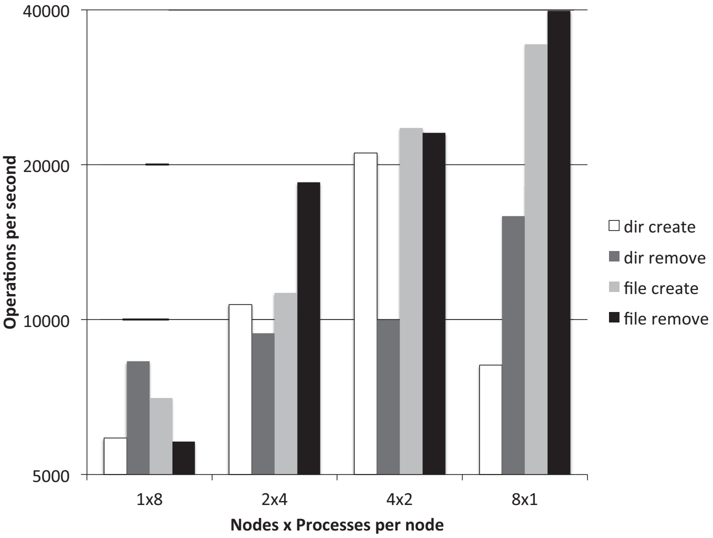

図 1: mdtest の実行結果 (8 プロセス)

ファイルシステムの容量は、フォーマット後のユーザーが利用可能な容量として、500T バイトを超える容量を提供しています。また、共有ファイルシステムを構成する I/O ノード、ストレージ間の部分で一点障害が発生した場合においても、共有ファイルシステム全体に影響を与えることなく、パス切替、I/O ノード切替を行い、運用を継続することが可能です。

以降、この共有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。

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。

3 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。

- ファイル・ディレクトリの作成・削除の速度を測定
- プロセスごとに上記の操作を 3000 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
- 5 回の測定を行い、平均値からアクセス速度を計算

---

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

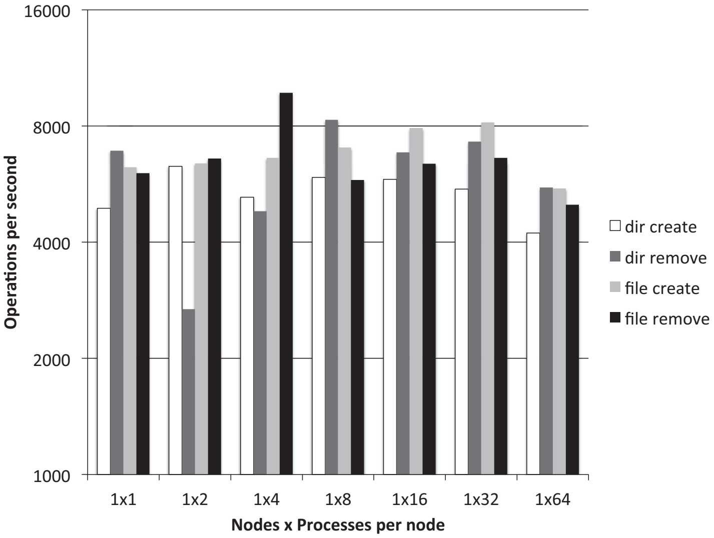

図 2: mdtest の実行結果 (1 ノード)

図 1 はプロセス数を 8 に固定して MDTEST を実行した結果です。横軸には使用したノード数とノードあたりプロセス数の組み合わせを、縦軸にはアクセス速度 (Operations per second) を対数目盛で示しています。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。8 プロセスの実行の場合、すべての操作について 5,000 回/秒以上の速度であり、操作によって結果のばらつきはありますが、ディレクトリ作成以外の操作についてはノード数が多いほど高い性能となることが分かります。ファイル作成については、8 ノードの場合で秒間約 35,000 回実行できています。一方、図 2 はノード数を 1 に固定してノード内で起動するプロセス数を変えて MDTEST を実行した結果です。こちらは、どのプロセス数でも各操作で一秒あたり数千回となり、ノード内のプロセス数は性能にあまり影響していないようです。ただ、64 プロセスの場合は他の場合の平均と比較して 10%から 30%低い性能となっています。これはノードの物理コア数 32 より多いプロセスを起動していることが影響していると推測されます。

4 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、上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
- ファイルの書き込みの性能を測定
- プロセスごとに 16GB のファイルを出力

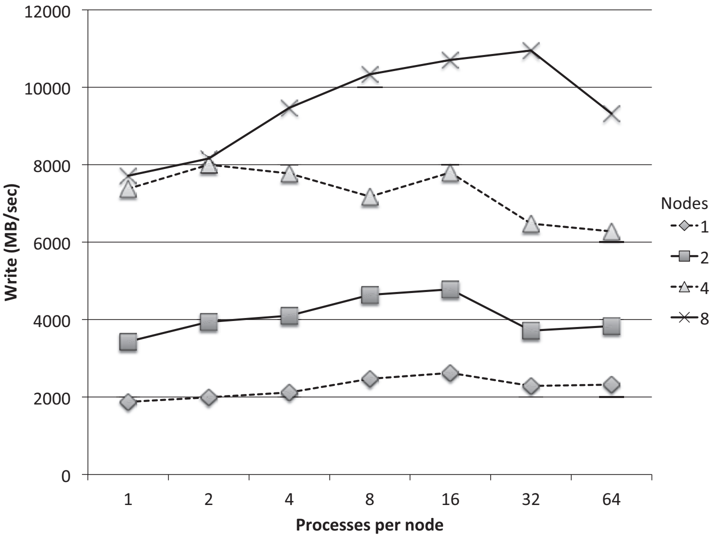

図 3: ior-multi の実行結果

実行結果を図3から図5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 (Write (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3と図4は、ior-multi、ior-single それぞれについて、書き込みのブロックサイズを1Mバイトとして、ノード数と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。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 の性能と比較して若干低くなる傾向にあります。また、ノード数を4ノードから8ノードに増加させた場合の性能向上率は低いものの、ノード数を増加させると性能は向上する傾向にあることが分かります。一方、同じノード数の場合の性能を比較すると、1ノード、2ノード、ior-multi の4ノードの実験では、ノードあたりのプロセス数を変化させても性能に大きな差が出ませんでした。これは、GPFS がノードごとにI/O帯域幅の制御を行っているためであると考えられます。ノード数およびプロセス数を増加させた場合に性能にばらつきがあるのは、他のジョブ等の影響があると思われます。図5は1ノードでの ior-multi 実行を、書き込みのブロックサイズを変化させて計測した結果です。まずノード内のプロセス数に着目すると、プロセス数が8または16の場合に最も高い性能となり、それ以上増やしても性能はよくならないことが分かります。また、ブロックサイズを4Kバイトから64Kバイトに変化させるとデータ転送性能の向上が見られますが、64Kバイト以上に増加させても性能はほぼ横ばいとなります。

読み込みの性能については、1ノード、16プロセスの場合、約1600Mバイト/秒です。これは書き込み時の約2500Mバイト/秒と比較すると3∼4割低い値です。

おわりに

メタデータのアクセス速度、データ転送性能のいずれについても、使用ノード数を増加させることで性能向上が見込めます。同じノード数で比較した場合は、ノード内のプロセス数を増減させても、性能はあまり変わりません。また、データ転送性能を向上させるには、入出力のブロックサイズを、例えば64Kバイトのように、ある程度大きな値にすることも重要です。

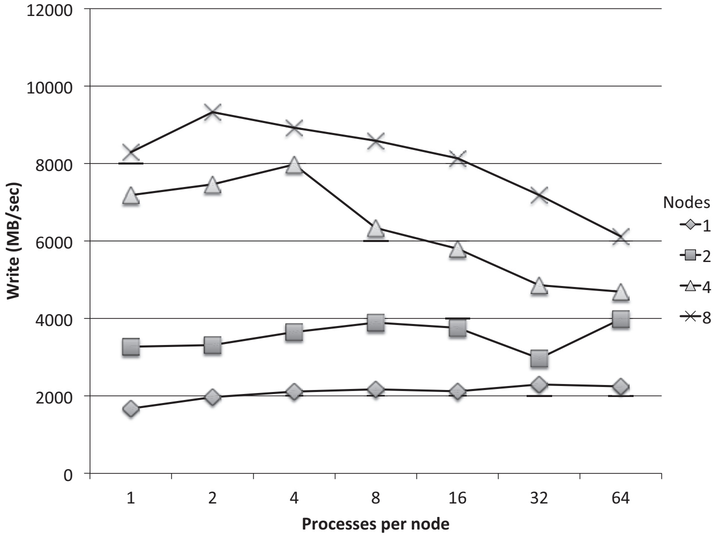

図 4: ior-single の実行結果

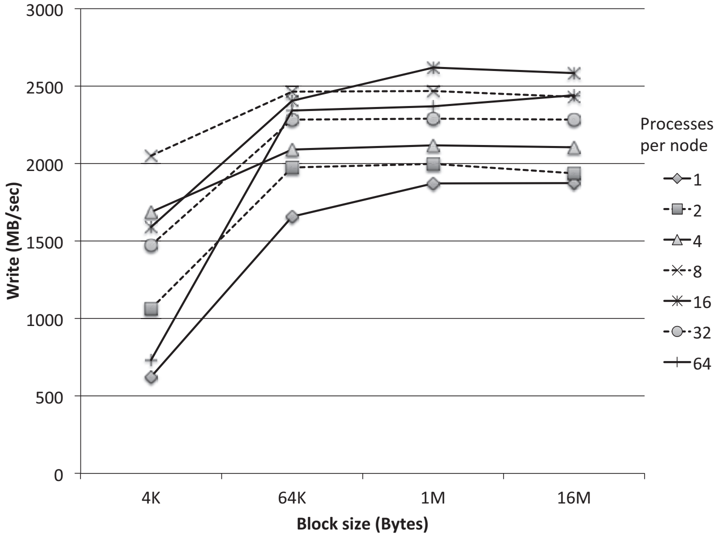

図 5: ior-multi の実行結果 (1 ノード)