情報爆発時代のグリッドファイルシステム上での大規模データ管理

佐藤 仁 [†1] 松岡 聡 [†1,†2] 遠藤 敏夫 [†1]

1. はじめに

近年、データインテンシブアプリケーションの実行環境としてグリッドの利用が実用的になりつつある。このような、グリッドでのデータ共有では、シングルシステムイメージを提供するためのファイルシステムを基盤に用いることが望ましい。これは、グリッド上の資源の存在を意識することなく、複数のアプリケーション間の連携がファイルベースで行える点などの利点があるためである。しかし、このようなファイルシステムをグリッドで運用しようとした場合、1) 遠方へのファイルアクセスの発生、2) 特定ノードへのファイルアクセス集中の発生、などの要因によるファイルアクセス性能低下が問題となる。このため、ファイルシステム側で必要に応じたファイルの移行や複製などのデータ管理を行うことで、ファイルアクセスの性能低下を抑える必要があるが、具体的にどのような戦略が有効であるかは明らかではない。

我々は InTrigger 上の 5 サイトの HPC クラスタを連携してファイルシステムを構成し、アクセスパターンに応じたシナリオを設定して、ファイルアクセスの性能を調査した。その結果、ネットワーク性能のみによる最適なファイル配置の実現は困難であり、実際のファイルアクセス性能のモニタリング情報の利用が有効であるという指針を得た。

2. データの自動管理機構を備えたグリッドファイルシステム

対象とするグリッドファイルシステムの構成を図 1 に示す。オープンソースで開発が進められている既存のグリッドファイルシステムである Gfarm File System (Gfarm)[1] にデータの自動管理機構を拡張して実現している。Gfarm は、ファイルシステムへのアクセスを提供するクライアント、ファイルシステム上の保存されるファイルのメタデータを扱うメタデータサーバ、また、実際にファイルを保存する I/O ノードの 3 つの主な構成要素からなる。これらの構成要素に加え、ファイルアクセスの挙動の解析や必要に応じてファイルの移行や複製などのデータ管理を行うために、クライアント上でファイルアクセス性能をモニタリングを行うデーモン、メタデータサーバでそれらのモニタリング情報の集約を行うデーモンが動作する。また、モニタリング情報を元に実際にファイルの移行や複製などの指令を行うデーモンがクライアント上で動作する。ファイルシステムとこれらの自動データ管理機構が協調動作することにより，グリッドでのファイルアクセスの性能低下の抑制を実現する。

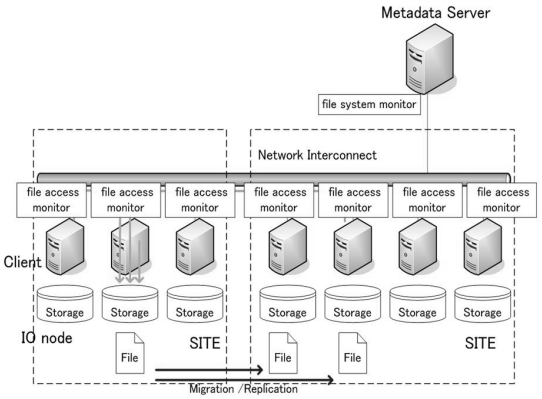

図 1 対象とするグリッドファイルシステムの構成

3. 実グリッド上でのデータの自動管理戦略の検討

グリッドでのファイルアクセスの性能を抑えるために核となるアイデアは、いかにアクセスが必要となるファイルをアクセス要求元の近くにおくか、という点である。このため、ファイルシステムを構成するノード間の空間的な関係を正確に推定し、ファイルのアクセス状況やファイルシステムを構成するノードの性能、また、接続されたノード同士のネットワークトポロジなどを考慮したファイル配置が必要である。近年，ネットワークトポロジの推定[2] やネットワーク上のノード

†1 東京工業大学
†2 国立情報学研究所

表 1 実験シナリオ

	rtt[ms]	bandwidth [MB/s]
1.LAN(best/worst)	0.162	117.3 / 41.3
2.WAN(低遅延, 高バンド幅)	1.66	104.5
3.WAN(高遅延, 低バンド幅)	6.03	97.63
4.WAN(低遅延, 高バンド幅)	2.59	10.74
5.WAN(高遅延, 低バンド幅)	12.5	10.53

表 2 各サイトの HPC クラスタのノード性能

	c2d (64bit)	pm (32bit)
CPU	Core2Duo 2.33GHz	Pentium M 1.8GHz
#cores	2	1
Memory	4GB	1GB
OS	Linux 2.6.18 (64bit)	Linux 2.6.18 (32bit)
Network	GBEther	

のグルーピング手法[3] が数多く提案されているが，実際のグリッドでのファイルシステムの運用を想定した場合、具体的にどのようなメトリックがファイルアクセス性能に影響を明らかではない。

我々は、InTrigger 上の東京 (hongo, okubo)、神奈川 (suzuk)、千葉 (chiba)、京都 (kyoto) の計 5 サイトに存在する HPC クラスタを 2 節のグリッドファイルシステムで連携し、表 3 に示すネットワーク性能に応じたシナリオを設定してファイルアクセスの性能を調査した。具体的には、hongo クラスタの 1 ノードから他ノード (hongo, suzuk, chiba, okubo, imade) 上に存在する 1 つのファイル (128MB) に対して内容を全て読み取るような read アクセスを 1 回行った際のファイルアクセスの I/O バンド幅を測定した。構築したグリッドファイルシステムにおいて、メタデータサーバは suzuk クラスタの 1 ノードに割り当て、他のクラスタノードをクライアントと I/O ノードとして割り当てた。 実験で使用した各サイトの HPC クラスタのノードの性能を表 2 に示す。各サイトのマシンの性能は同一で c2d で示されるノードを用いた。ただし、hongo クラスタのみ c2d ノードの他に pm で示されるノードも用いた。そのため、hongo クラスタ内ではマシン構成によりノード間のバンド幅が異なるので表 3 の LAN の項目ではバンド幅の最良値と最悪値を併記している。

実験結果を表 3 に示す。ファイルアクセス性能はノード間のバンド幅の性能に強い影響を受けることを確認した。シナリオ 2 とシナリオ 4 の場合、同等の RTT を示したが、ネットワークバンド幅の性能の違いに影響を受け、大幅なファイルアクセス性能の違いを示した。一方で、いくつかのシナリオでは例外が見

表 3 実験結果

	bandwidth[MB/s]
1. LAN (best / worst)	54.3 / 7.55
2.WAN(低遅延、高バンド幅)	15.2
3.WAN(高遅延、低バンド幅)	4.92
4.WAN(低遅延、高バンド幅)	4.78
5.WAN(高遅延、低バンド幅)	1.20

られた。例えば、シナリオ 1 の最悪値とシナリオ 2 の場合を比較した場合、LAN 内のノード間でファイルアクセスを行うよりも WAN をまたいで別クラスタのノードへアクセスしたほうが良好な性能を示した。また、シナリオ 2 とシナリオ 3 の比較では、同等のネットワークバンド幅の性能を示すのにも関わらず遅延の違いにより大幅な性能の違いを示した。これらは、ファイルアクセスの性能が、ノード間のネットワーク性能だけでなく、マシン構成やそれらに設定されているパラメタの違いなどの複合的な要因により決まるため、これらのうちの限定されたバラメタのみに依存してファイルシステムを構成するノード間の空間的な関係を正確に推定することが困難であることがわかった。このため、実際のファイルアクセス性能をモニタリングし、ファイルシステムを構成するノードの関係を推定する手法などが必要である。

4. おわりに

本稿では、実際のグリッドでの最適なファイル配置を実現するための戦略を明らかにするために、InTrigger 上で行ったアクセスパターンのシナリオに応じたファイルアクセス性能に関する調査について述べた。今後は、ファイルシステム上でファイルアクセス性能のモニタリングを行い、実際のアプリケーションのワークロードを想定したデータ管理手法を検討していく予定である。

参考文献

1) Gfarm, `http://datafarm.apgrid.org`.
2) Shirai, T., Saito, H. and Taura, K.: A Fast Topology Inference — A building block for network-aware parallel computing, in *The 16th IEEE International Symposium on High Performance Distributed Computing (HPDC 2007)*, pp. 11–21 (2007).
3) Xu, Q. and Subhlok, J.: Automatic Clustering of Grid Nodes, in *GRID '05: Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, pp. 227–233, Washington, DC, USA (2005), IEEE Computer Society.