

GEO Gridの構築に向けたストレージシステムの予備評価

谷村 勇 輔[†] 山本 直 孝[†] 石橋 拓 也^{††}
中田 秀 基[†] 田中 良 夫[†] 関口 智 嗣[†]

衛星観測した数百TBのデータをオンラインのハードディスクに保管し、必要に応じていつでも参照できる環境の整備が求められている。そのバックエンドとなるストレージは、大規模データの保存とそれらの高速処理を実現するシステムでなくてはならない。そのようなシステムを検討するために、ASTER 衛星データとそのアプリケーションを例に、Gfarm と Lustre を用いて同等のシステムを構築した場合のデータアクセスの性能と運用方法を比較した。性能面では ASTER アプリケーションのデータ処理を解析した上で、各ストレージシステムでのアプリケーションの実行性能を比較した。運用面では故障に対する備え方、メンテナンスの作業内容や課題を比較した。これらの調査により、より大規模な衛星観測データの保存システム、あるいは類似アプリケーションのためのストレージを構築するための指針を得た。

Evaluation of The Storage System for Building The GEO Grid

YUSUKE TANIMURA,[†] NAOTAKA YAMAMOTO,[†] TAKUYA ISHIBASHI,^{††}
HIDEMOTO NAKADA,[†] YOSHIO TANAKA[†] and SATOSHI SEKIGUCHI[†]

An infrastructure to store hundreds terabytes of satellite observation data in the online hard disk drives, and to allow geosciences researchers to access the data from anywhere at anytime, is highly expected. A backend storage system on that infrastructure needs to meet the requirements of huge data storing capacity and high-performance data access. In order to design and build this kind of HPC storage system, Gfarm and Lustre were compared by the ASTER satellite image data application. In terms of the performance comparison, the data processing part was profiled, and the execution time of the major data conversion program was measured on both storage systems. In addition, configuration for fault tolerance, daily maintenance and issues for operation were compared between the two systems. This study provides the information to build a larger storage system for the satellite image data, and other applications which have similar data access patterns to the ASTER application.

1. はじめに

地球観測分野において、衛星観測したデータをオンラインのハードディスクに保管しておき、必要に応じていつでも参照できる環境を整備することは重要な課題の1つである。これに応えるため、産総研では既存のE-scienceの枠組を地球観測アプリケーションに適用したGEO Grid環境の構築を進めている。GEO Gridではデータを保管するためのストレージやデータを処理するための計算機、参照したいデータを見つけるための検索サービス、ユーザインタフェースとしてのWebポータルやアプリケーション開発ツールなどのコンポーネントがVO (Virtual Organization)の枠組に沿って相互に連携し、地球観測の研究を支援す

るIT環境を実現する。

GEO Gridにおいて、バックエンドのストレージは以下の要件を満たすことが必須である。保存に関する要件は数百TBの規模のデータを格納できること、データの損失がないこと、高い可用性があることである。データのアクセス性能については、複数のデータ処理プログラムを同時に実行する際に十分なスケールビリティが得られることである。これらの要件を満たしつつ、導入および運用時のコストが適度に低いことが望ましい。これに対して、産総研では無償のソフトウェアであるGfarm¹⁾とハードディスクを多数搭載できる汎用のサーバを組み合わせ、ASTER (Advanced Spaceborne Thermal Emission and Reflection Radiometer) 衛星の観測データを保管するストレージシステムを構築・運用してきた²⁾。しかし、地球観測衛星は年々大型化しており、それに搭載されるセンサから得られるデータは1PBを越えることが見込まれている。このため、ASTERの数十倍の規模のデータを

[†] 産業技術総合研究所 / National Institute of Advanced Industrial Science and Technology (AIST)

^{††} 株式会社 創夢 / SOUM Corporation

扱えるストレージの検討が課題となっている。

このような背景から ASTER 衛星データとそのアプリケーションを例題に、大規模な並列データ処理に向いていると考えられる Gfarm と Lustre に注目し、それぞれを用いて構築したストレージシステムの比較を行う。Gfarm は産総研を中心に開発してきた分散ファイルシステムであり、実際に ASTER 用のストレージシステムで利用している。Gfarm はアプリケーションがデータに局所的にアクセスし、かつデータインテンシブなアクセスを行う場合に性能を発揮する。一方、Lustre はスケラビリティの高さから HPC クラスタにおいて広く利用されている³⁾。

両者を用いたストレージシステムの比較はデータアクセスの性能だけでなく、構築の手間や運用時の管理コストの面でも行う。性能比較では、ASTER のデータ処理におけるファイルアクセスパターンを調査した上で、各処理のファイルアクセスの性質に合ったストレージシステムを明らかにする。さらに、一般によく利用される NFS を用いて構築したストレージシステムの性能も提示する。運用面についてはディスクやノードの障害に対する備え方、システムのメンテナンスの作業内容や課題について調査する。こうして、より大規模な衛星観測データを扱うためのストレージシステム、および類似のアプリケーションに適したストレージシステムを構築するための指針を得る。

2. ASTER 衛星観測データの処理

ASTER は広帯域のスペクトラムと高い空間分解能を備えた地球観測センサである。経済産業省によって開発され、NASA の EOS (Earth Observing System) プログラムの最初の衛星として 1999 年 12 月に打ち上げられた。観測および、データの保管は ERSDAC (Earth Remote Sensing Data Analysis Center) によって行われている。ERSDAC では L0 と呼ばれる ASTER の生データをテープ装置に保管しているため、データを取り出すのに時間がかかる。ユーザがデータを入手するのにかかる時間を短縮するため、産総研によって常時オンラインでデータを提供できるストレージシステムが Gfarm を用いて構築されている。

図 1 に ASTER 用のストレージシステムの構成を示す。2006 年 11 月～2007 年 5 月にかけて ERSDAC に保存された全ての過去データをこのストレージシステムにインポートする作業が行われた。現在は新しく観測された 70～100GB のデータを毎日受信している。ASTER データの受信からデータ処理、公開までの流れは次の通りである。

(1) FTP

複数の FTP サーバを用いて ERSDAC から 1 日分 (8 時間 × 3 セット) の生データ (L0) を受信し、ストレージシステムに保存する。1 日

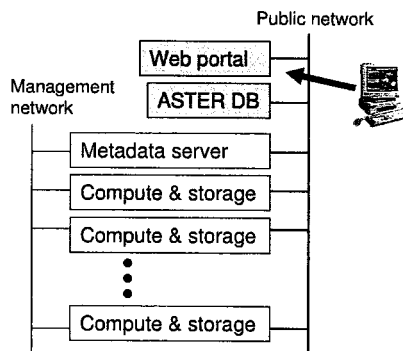


図 1 ASTER 用ストレージシステム

分のデータサイズは合計で 70～100GB である。転送はセット毎に行われ、1 回の接続で受信するファイル数は 30 程度である。

(2) MV

L0 データを観測単位で DB に登録し、適切なディレクトリに移動する。

(3) BZIP2

L0 データの各ファイルを bzip2 を用いて圧縮する。

(4) REPLICATE

Gfarm の機能を利用して、L0 データの複製を 1 ファイルにつき 1 つ作成する。

(5) BUNZIP2

次の L1PGE によるデータ処理のため、L0 データを bunzip2 を用いて展開する。

(6) L1PGE

L0 データを入力ファイルとして、L1PGE と呼ばれるプログラムを実行する。L1PGE は複数の L0 ファイルと 1 組の観測計画ファイル (OSF) を用いて、複数のシーン (2 次元) 画像データ (L1A) を得る。1 日分の観測データから約 500 ファイルが生成され、それらを全てストレージシステムに保存する。

(7) DTMSOFT

L1A データを入力ファイルとして、DTMSOFT と呼ばれるプログラムを実行する。DTMSOFT は視差がある 2 つの可視光バンドの画像からステレオマッチングを行い、数値標高モデル (DEM) を作成する。作成した DEM を用いて L1A データの全てのバンドに対して地形補正を行い、斜めに見た画像を真上から見た画像に変換する。結果として、1 つの L1A ファイルから 10 個程度の地形補正面像 (L3A) ファイルを得る。L3A ファイルのみをストレージシステムに保存し、DTMSOFT の実行中に生成された中間ファイルは全て消去する。

(8) CONVERT

L3A データを入力ファイルとして画像処理を施し、Web での閲覧用の JPEG や PNG 画像を生成する。

このようにして作成した L3A 画像は、地球観測に携わる研究者によって図 1 の Web ポータルから検索され、必要なデータがダウンロードされる。JPEG や PNG 画像は検索した画像を Web ブラウザで閲覧し、必要なデータかどうかを確認するために用いられる。なお、DTMSOFT は多様なパラメータをもっているため、全パラメータに対する処理結果を保管するのではなく、Web ポータルを通してユーザからのリクエストに応じてデータ処理を実行できる機能も提供している。

3. 比較するストレージシステム

本研究では Gfarm, Lustre, NFS の各ネットワークファイルシステムを用いて構築したストレージシステムを比較する。ただし、それぞれのファイルシステムは設計思想が異なるため、同じハードウェア構成で比較するのではなく、それぞれの特徴を活かしたシステムを構成し、ASTER アプリケーションを実行する性能と運用に必要な作業をもって比較する。以下に各ストレージシステムの特徴を述べる。

3.1 Gfarm

Gfarm¹⁾ はネットワーク上に分散するストレージサーバを統合し、仮想的なディレクトリツリーを用いてファイルにアクセスできる分散ファイルシステムを実現する。大規模なデータを並列に処理できるよう、ストレージサーバ上の CPU とディスクを 1 つのセットと考えて、アクセスする予定のファイルが保存されたノードでプログラムを実行することで性能のスケラビリティを得る。データアクセスにおいて I/O スループットの占める割合が大きい場合やデータアクセスと同時に多数の CPU 処理を要求する場合に、アプリケーションは Gfarm 上で高い性能を発揮できるように設計されている。本研究においても、観測データの処理とデータの保存をできる限り同じノードで行えるようにシステムを構築する。

3.2 Lustre

Lustre は TCP/IP のネットワークだけでなく、InfiniBand や Myrinet など高速なネットワーク機器をサポートするクラスターファイルシステムである³⁾。ストレージサーバはオブジェクト化されており、Lustre のクライアントは 1 つのファイルを分割して、任意のストレージサーバに対してストライピングで書き込むことで、単一ディスクの性能以上の I/O スループットを達成する。Gfarm と異なり、通常はストレージノードとデータ処理のノードが分離されるため、ノード間のネットワーク性能が重要になる。本研究では Force 10 E600 を用いて各ノードを 1Gbps のネットワークで接続する。

3.3 NFS

NFS⁴⁾ はノード間でのデータ共有の目的のために広く利用されているファイルシステムである。NFS を用いた場合、ストレージサーバを 1 台用意して、データ処理のノードからネットワーク越しにデータにアクセスする形になる。しかし、1 ノードに搭載できるディスク量は限定され、現在は単一のマシンで数百 TB の衛星観測データを保管することはできない。例えば、大容量のストレージサーバである Sun Fire X4500⁵⁾ でも 500GB のディスクを 48 台搭載して 24TB の容量である。数百 TB を用意するには複数台の NFS サーバを用意する必要がある。一方、複数の NFS サーバからなるシステムを構築したとしても、マウントポイントの管理の手間を除けば、性能は単一の NFS と同じである。そこで本研究では X4500 を 1 台だけ用い、それを NFS サーバとしてシステムを構築する。

4. 性能比較

性能比較は次の手順で進める。まず、産総研の ASTER 用ストレージシステムを用いて、ASTER の主なデータ処理プログラムのファイルアクセスパターンを調査する。次に、メタデータ操作の性能を測定するベンチマーク (fsysbench) と I/O スループットの性能を測定するベンチマーク (thput-fsys) を用いて Gfarm, Lustre, NFS の性能を明らかにする。その後、ASTER アプリケーションとして利用頻度の高い DTMSOFT を並列実行した時の実行時間を比較する。実験に用いる計算機のスペックは表 1 の通りである。Gfarm の実験では Gfarm v1.4.1 と GfarmFS-FUSE v2.0, Lustre の実験では Lustre v1.4.5 を用いる。

fsysbench, thput-fsys はいずれも Gfarm ソフトウェアに含まれるベンチマークである。前者は open, chmod, rename, unlink など基本的なファイルのメタデータ操作のコストを計測する。後者は逐次的なブロック書き込み、ブロック読み込みの性能を測定する。

4.1 GEO アプリケーションの特徴

2 節に記した各アプリケーションのファイルアクセスパターンを調べるため、入出力データをローカルファイルシステム上に置いた時と Gfarm ファイルシステム上に置いた時の実行時間を比較した。さらに、Gfarm のライブラリ内の処理時間を計測し、READ, WRITE に要した時間の内訳を調査した。表 2 に得られた結果を示す。

表 2 より、Gfarm ファイルシステムを使うことで実行時間の増加が大きいアプリケーションは FTP, L1PGE, DTMSOFT であることが分かる。他のアプリケーションは CPU インテンシブであるため、ファイルシステムによる処理速度の違いは小さい。FTP のファイル操作は 8 割が書き込みであり、Gfarm において処理速度が倍になっているのはローカルファイ

表 1 ASTER ストレージシステムと実験環境

ASTER storage servers	Xeon 3.8GHz × 2, 4GB memory, 16-drive RAID 6 disk array (7TB), CentOS 4.4.
ASTER metadata server	Dual-core Opteron 2.6GHz × 2, 16GB memory, CentOS 4.4, PostgreSQL 8.2.1.
F32 cluster nodes	Xeon 3.06GHz × 2, 4GB memory, U-ATA disk (WDC WD2500SB), RedHat 8.0.
Sun Fire X4500	Dual-core Opteron 2.6GHz × 2, 16GB memory, 6-drive RAID-Z disk array (2.2TB), Solaris10.
GFM cluster nodes	Xeon 2.8GHz × 2, 1GB memory, U-ATA disk (MK4018GAS), CentOS 4.4, PostgreSQL 8.1.6.

表 2 アプリケーションの実行時間と Libgfarm 内の処理

Application	On Local-FS	On Gfarm-FS	Libgfarm	READ	WRITE	Note
FTP	7.31 [sec]	13.8 [sec]	5.98 [sec]	0 [%]	80.8 [%]	Received data size was 486 MB.
MV	-	1.43	1.43	0	0	The directory contained 78 files.
BZIP2	4559	4668	541	60.2	36.1	74 files whose total size were 16 GB, were compressed.
BUNZIP2	2704	2755	601	29.1	68.4	74 files whose total size were 7.3 GB, were decompressed.
LIPGE	639	1898	1322	16.0	4.5	L0 data taken at 1:45:55 UTC, Feb. 27, 2007 was used as input.
DTMSOFT	749	1158	317	8.4	6.4	L1A data generated from L0 data taken at 1:46:28 UTC, Mar 31, 2007 was used as input.
CONVERT	66.4	67.7	3.0	21.3	13.6	The 6222×4828-sized image was processed.

表 3 NFS のパラメータ

Server side	NFSD.SERVERS=260
Client side	rsize=32K, wsize=32K, UDP, noatime

ルシステムと FUSE 経由での Gfarm ファイルシステムの書き込み速度の差といえる。

LIPGE や DTMSOFT は Libgfarm 内の処理時間が大きい、READ、WRITE が占める割合は小さい。これは Gfarm の I/O 操作よりもメタデータアクセスに時間がかかっているためである。このような場合、入出力データを Gfarm ファイルシステム上に置くのは望ましくない。

4.2 Gfarm と NFS の比較

Gfarm と NFS の比較において、Gfarm を用いたストレージは F32 クラスタの 64 ノードをデータ処理ノード、GFM クラスタの 1 ノードをメタデータサーバのノードとして構成した。Gfarm へのアクセスは GfarmFS-FUSE を利用し、FUSE の `direct_io` オプションを有効にした。NFS を利用したストレージは X4500 を NFS サーバとし、Gfarm の実験で用いた F32 クラスタの 64 ノードからマウントする構成とした。NFS のパラメータは表 3 の通りである。X4500 には 48 台のディスクが搭載されているが、本実験では 6 台のディスクからなる ZFS の 1 ブールを利用した。それぞれのストレージシステムに対して、複数のクライアントから同時にアクセスした時の性能のスケラビリティを測定した。図 2 はファイル数を 1000 と設定した `fsysbench` の実行時間を比較し、図 3 は各

* ただし、本調査により DTMSOFT はファイルの OPEN が異常に多いことが分かった。その後、ファイルの OPEN が少なくなるようにプログラムを修正したところ、Libgfarm 内の処理が全実行時間の約 6.1% になった。すなわち、DTMSOFT も比較的 CPU インテンシブなアプリケーションといえる。

ノードが 1GB のファイルに同時アクセスした時のノードあたりの I/O スループットを比較している。同時実行するクライアントは異なるディレクトリ、およびファイルにアクセスし、Gfarm の実験ではアクセスは全てローカル I/O になるようにしている。結果より、NFS はクライアント数の増加に対して、メタデータ操作にかかる時間のコストはほとんど増加しないが、1 クライアントあたりの I/O スループットはサーバ側のネットワークインタフェースの性能に制限されていることが分かった。一方、Gfarm ではメタデータ操作にかかるコストは同時アクセスするクライアント数に比例して増加する。これはメタデータサーバである PostgreSQL の処理性能が限界に達しているためである。ノードあたりの I/O スループットは、ストレージノード上のファイルに対する I/O である限りネットワーク性能に依存しないため、ほとんど劣化しない。

図 4 は DTMSOFT を同時実行した時の 1 クライアントの平均実行時間とその内訳を比較している。DTMSOFT は表 2 よりメタデータアクセスが多いことが分かっており、図 2 の結果に近い結果が予想されたが、実際にその通りになった。また、DTMSOFT の処理のうち Parallax と Ortho と呼ばれるサブプログラムの処理時間が増加しており、これらのメタデータアクセスがボトルネックであることが分かった。

4.3 Lustre の性能

Lustre を利用したストレージシステムは GFM クラスタの 5 ノードをストレージ (OSS)、1 ノードをメタデータサーバ (MDS)、6 ノードをデータ処理 (Client) として構成した。そして、Lustre 用に構築したマシンの一部を用いてメタデータアクセスの性能を NFS、I/O スループットの性能を Gfarm と比較した。NFS の構成は GFM クラスタの 1 ノードをサーバとし、6 台のデータ処理ノードをクライアントとした。Gfarm の構成は、5 台のデータ処理ノードにおい

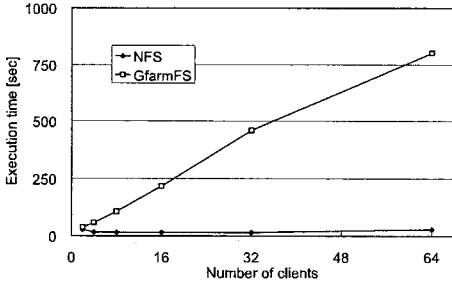


図 2 同時アクセス時のメタデータ操作の性能

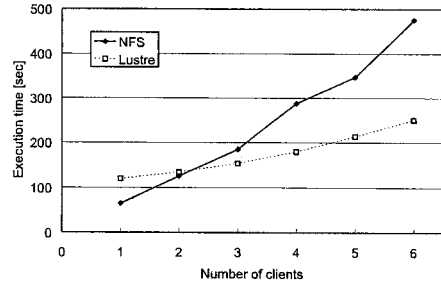


図 5 同時アクセス時のメタデータ操作の性能

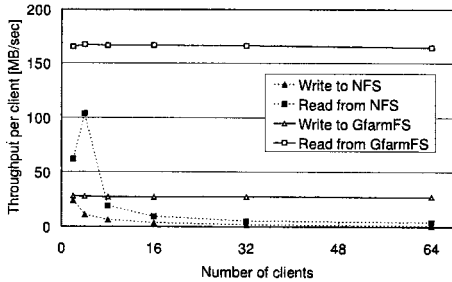


図 3 同時アクセス時の I/O スループットの性能

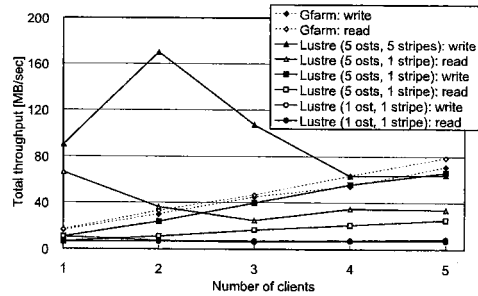


図 6 同時アクセス時の I/O スループットの性能

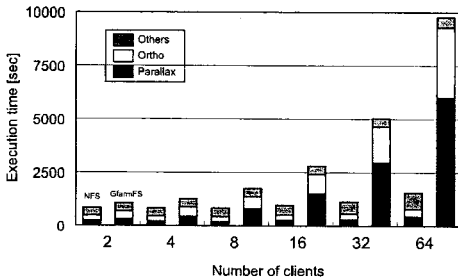


図 4 DTMSOFT を同時に起動した時の実行時間

で Gfarm のストレージサーバを動かし、別のノードでメタデータサーバを設定した。Gfarm へのアクセスには gfs_hook を利用した。

メタデータアクセスの性能比較では、各システムに 100 万個以上のファイルを格納し、複数のクライアントからファイル数を 10,000 に設定した fsysbench を同時実行した。図 5 より、クライアント数の増加に対する fsysbench の実行時間は NFS よりも Lustre の方が小さいといえる。

同様に、複数のクライアントからファイルサイズを 2GB として thput-fsys を同時実行した結果を図 6 に示す。クライアント数が少ない時の性能は Lustre が

高く、クライアント数が増えるにつれて Gfarm が良い性能を示した。これは Lustre がストライピングを利用して性能を得ているが、ストライプファイルが格納されるストレージが重なった場合、そのノードの HDD がボトルネックになるためと考えられる。ストライプ数を 1 としてその重なりをなくした場合、Lustre の Write 性能は Gfarm のそれに近い値となった。また、Lustre では単一 OSS を使用した時の Read 性能がディスク性能に比べてかなり低いことが分かった。

5. 運用面の比較

ASTER アプリケーションの要件に基づきストレージシステムを運用する場合の可用性の確保、メンテナンスの作業内容や問題点について Gfarm v1.4.1 と Lustre v1.6 を比較した。以下に主な違いを紹介する。

5.1 高可用性の実現

ASTER 用のストレージシステムは ERSDAC から L0 データを毎日受信し、研究者がいつでも必要な L3A データにアクセスできる環境を提供するために高い可用性が求められる。可用性を高めるためには想定される障害に対する備えを講じておくことが重要であるが、Gfarm と Lustre は障害に対して表 4 のような違いがある。Gfarm を利用したストレージシステムは、ファイル毎に複製を作成し、障害が発生してもアプリケー

表 4 故障に対する備え

Fault item	Gfarm	Lustre
Storage node	File replication	Failover
Disk on storage node	File replication or RAID	RAID
Metadata server node	PgPool for PostgreSQL	Failover

ションが必要なデータにアクセスできるようにしておくことが大切である。ファイル複製は頻繁にデータを更新する用途に適さないが、データの重要度によって複製の有無をファイル毎に決められる利点がある。

一方、Lustre はノードの障害に対しては Failover を設定しておくことで対応する。これは障害検知を行う Heartbeat⁶⁾ などと組み合わせて利用し、導入時の設定が若干面倒であるものの、運用開始後の手間はかからない。ディスクの障害は RAID を利用することを前提としている。

5.2 メンテナンス作業の手間と課題

計画的に一部のストレージノードを停止する作業は、Gfarm も Lustre もメタデータサーバに停止ノードを通知することでシステム全体を止めずに行うことができる。停止されるノードが保存するファイルへのアクセスを妨げないためには、表 4 に示した障害時の対応と同様の設定が必要である。

ストレージノードの追加や交換においてはノード間のデータ移行が必要である。これに関して Gfarm も Lustre も専用の移行ツールは提供していない。いずれにおいても、移行すべきデータを調べて別のノードにコピーする作業をいくつかのコマンドを組み合わせる必要がある。

Gfarm の運用においてメタデータサーバ (PostgreSQL) とメタデータキャッシュサーバ (gfarm_agent) の設定は、メタデータインテンシブなアプリケーションの性能に大きな影響があり、非常に重要である。メタデータサーバに用いるマシンには十分なメモリを搭載し、PostgreSQL のバキューム処理を定期的に行って不要レコードを削除する運用が必要である。なお、ファイル名の長さにも依存するが、Gfarm では約 800 万ファイルに対して 4GB 程度のメモリが必要である。

Lustre は専用のカーネルが必要となるため、導入に際して利用できるカーネルのバージョンや Linux のディストリビューションに厳しい制限がある。加えて、Lustre の設定に問題があったり障害が発生したりした場合にカーネルの挙動が不安定になる可能性があるため、運用前に安定性の検証を十分に行う必要がある。

6. 考察とまとめ

大規模なデータの保存と処理を要求するアプリケーションとして ASTER を例に、Gfarm と Lustre の性

能を比較した。ASTER アプリケーションの各処理のプロファイルをとることで、ほとんどの処理がデータアクセスが少ない計算主体の処理、あるいはメタデータインテンシブなファイルアクセスを行う処理であることを確認した。

計算が主体の処理ではバックエンドのストレージの性能はアプリケーションの実行時間にあまり影響しない。ただし、運用面では Gfarm はデータの保存と処理を一体化したノードでシステムを構成できるため、システムの構築にかかる費用を抑えられる。

メタデータインテンシブな処理では Lustre や NFS が高い性能を達成した。これらに比べて、Gfarm はメタデータアクセスの性能が低く、少しでも高い性能を得るために運用上の工夫が必要である。Gfarm v2⁷⁾ では複数のメタデータサーバを用いて、それらのメモリ上にメタデータを持つように実装することで性能改善を図ろうとしている。

ASTER のデータ処理では重要でないが、データインテンシブな処理では同時アクセスに対して Gfarm が高い性能を得た。Lustre において同様のスケラビリティを得るためにはストレージノードやターゲットの追加、高速な HDD の利用を考える必要がある。

本研究では各ストレージシステムの構築や運用において留意すべき点についても調査した。これらの比較調査の結果は ASTER に類似したデータアクセスを行う科学技術アプリケーションや、ASTER 以外の衛星観測データのためのストレージシステムを構築する際に有用であると考えられる。

参考文献

- 1) O. Tatebe, Y. Morita, S. Matsuoka, N. Soda, and S. Sekiguchi. Grid Datafarm Architecture for Petascale Data Intensive Computing. In *Proceedings of the 2nd IEEE/ACM Symposium on Cluster Computing and the Grid (CCGrid)*, pp. 102-110, 2002.
- 2) N. Yamamoto and et al. GEO Grid: Grid Infrastructure for Integration of Huge Satellite Imagery and Geoscience Data sets. p. 75, 2006.
- 3) Lustre. <http://www.lustre.org/>.
- 4) B. Callaghan, B. Pawlowski, and P. Staubach. NFS Version 3 Protocol Specification: RFC 1813. Internet Engineering Task Force, Network Working Group, 1995.
- 5) Sun Fire X4500 Server. <http://www.sun.com/servers/x64/x4500/>.
- 6) Linux HA, Heartbeat. <http://www.linux-ha.org/Heartbeat/>.
- 7) 建部修見, 曾田哲之, 関口智嗣. 広域分散ファイルシステム Gfarm v2 の設計と実装. 情報処理学会研究報告 HPC-99, pp. 145-150, 2004.