EVALUATION OF LARGE-SCALE STORAGE SYSTEMS FOR SATELLITE DATA IN GEO GRID

Y. Tanimura, N. Yamamoto, Y. Tanaka, K. Iwao, I. Kojima, R. Nakamura, S. Tsuchida, and S. Sekiguchi

National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba city, Ibaraki Prefecture., Japan - (yusuke.tanimura, iwao.koki, yoshio.tanaka, r.nakamura, s.tsuchida, s.sekiguchi)@aist.go.jp, - (naotaka, kojima)@ni.aist.go.jp

KEYWORDS: Data, Archiving, Analysis, Spatial Information Sciences, Global-Environmental-Databases

ABSTRACT:

The Global Earth Observation (GEO) Grid infrastructure is an E-Science infrastructure which enables global research activity and drives geosciences to get a significant discovery or achievement in their fields. The main design principal of the GEO Grid system is the open and standard protocol-based architecture. It supports creation of a virtual organization (VO) by integrating data and computing services according to the requirements by the VO. VO-level access control realizes flexible and scalable security infrastructure, against the increasing number of VOs and users. For extending the GEO Grid infrastructure, the number of resource providers should be more. Though the GEO Grid framework provides a facilitating tool of managing a GEO Grid site, building and operating a large storage system is one of the difficult tasks. For example, the ASTER storage which joins the GEO Grid system is now operated by AIST and it is faced on the capacity issue for new sensors. In order to build a larger, petabytes-scale storage system, a Gfarm-based storage system and a Lustre-based storage system, were compared. Real data sets of the satellite images were imported into both storage systems and the performance was measured by a practical application. Functions for fault tolerance and daily maintenance work issues are investigated to reveal operation cost. The comparison results indicate that a factor of choosing storage system is not performance but installation and operation cost. This paper provides an overview of the GEO Grid system, and summary information of what the GEO Grid resource providers should consider about for their internal storage systems.

1. INTRODUCTION

As the Earth's ecosystem is a spatially and temporally complex system by nature, it is not sufficient to observe such events and phenomena locally; problems must be solved on a global scale. Therefore, the accumulation of knowledge about the earth in various forms, and a scientifically correct understanding of the earth are necessary. The authors have been leading the "GEO (Global Earth Observation) Grid" project since 2005 which is primarily aiming at providing an E-Science infrastructure for worldwide Earth Sciences community. In the community, there are wide varieties of existing data sets including satellite imagery, geological data, and ground sensed data that each data owner insists own licensing policy. Also, there are so many of related projects that will be configured as virtual organization (VO) enabled by Grid technology. The GEO Grid is designed to integrate all the relevant data virtually, again enabled by Grid technology, and is accessible as a set of services.

As a site of the GEO Grid infrastructure, AIST preserves global data sets of ASTER which is a sensor onboard a satellite, and provides online data service to geoscientists. More than 150 TB of data are stored on hard disk drives of numerous computers. The integration of those computers and disks is handled by a cluster file system. We adopt an open-source cluster file system due to reasonable cost performance although there are many commercial storage products such as SAN-based storage systems. This ASTER storage system is now stable and there is no performance problem. The issue we have is to satisfy growing capacity demand. A next sensor of ASTER is going to produce more than one petabytes data in its operation period. Therefore, we analyze our ASTER storage system and then evaluate two storage systems with a different cluster file system, towards the petabytes-scale capacity. Evaluation is performed in both performance and operation aspects. In performance evaluation, real data set and a practical application program is used. In operational evaluation, functions for fault tolerance, daily maintenance work issues and so on are compared. Knowledge learned from the evaluation is reported in this paper.

In this paper, we present an overview of the GEO Grid and briefly discuss its architecture and implementation in Section 2. Section 3 describes current status and issues of the ASTER storage system. Section 4 explains two storage systems which are compared in this study, and shows evaluation results. Section 5 discusses about the results in Section 4 and the paper is concluded in Section 6.

2. GEO GRID INFRASTRUCTURE

2.1 Systems Enabling Secured Integration of Data and Applications

2.1.1 Motivational Background and the Scenario: We are living on the planet Earth. Hence, earth observations are indispensable to all of our activities, especially disaster mitigation, weather prediction, natural resource exploration, environment monitoring, and so on. In particular, satellite sensors are crucial to global observations and the data rates are currently growing dramatically. For example, the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) (Yamamoto and et al., 1998), and the Moderate Resolution Imaging Spectroradiometer (MODIS) (Justice and et al., 1998) on the Terra earth observation mission satellite has produced more than several hundred terabytes of raw data since its launch in 1999. PALSAR (Phased Array L-band Synthetic Aperture Radar) onboard that DAICHI satellite, which started operation in 2006, will generate more than 1 petabytes of raw data in its five-year nominal mission period. In addition to such vast amounts of satellite data, scientists may be interested in accessing other databases, such as the data from land-use map, climate data, and field sensors to know the real world from ground. Computing that includes simulation and analysis of the data may also be of interest to the users. However, it is a complex and difficult task for an individual user to retrieve the desired data from distributed databases and process that data on remote computing resources. In order to help scientists advance research on earth observations using distributed data and computation, easy and effective access methods for the databases and computing resources must be provided. On the other hand, the original data from some earth observation satellite sensors, such as ASTER and PALSAR, are provided as commercial products. The providers of the data require the system to authenticate and authorize users, and control access to their data according to the policy of the data providers.

2.1.2 Functional Requirements for the IT infrastructure: The following requirements are drawn from typical scenarios which are based on our motivational background.

Size scalability in near-real-time data handing and distribution: Remote sensing data obtained from a satellite nowadays requires a capacity of more than several hundred terabytes during its nominal operation period. Such huge datasets, not limited to satellite imagery data, will be made available with minimum time delay and at minimum cost.

Handling wide diversification of data types, associated metadata, products and services: Data about earth observation is diverse. For example, the modeling of the carbon cycle needs various input parameters, such as land-use map, climate data, and vegetation indices. Research communities wish to integrate some of these data according to their interests. Metadata and derivatives associated with the original data should be taken into account. Hence the IT infrastructure must support the creation of user groups which represent various types of virtual research or business communities, and the federation of distributed and heterogeneous data resources which is shared in such communities.

Respecting data owner's publication policies: Some data sets such as ASTER imagery cannot be freely accessible. Due to restrictions concerning the protection of national security, intellectual property, privacy, confidentiality, and relevant ethical issues, the data owner is generally willing to permit only a range of data access, certain choices of data format. They wish to require the users to accept, certain limits on the transfer of the rights, etc., and wish to reserve the authority to set and modify licensing rights and conditions. Therefore, the IT infrastructure must provide a security infrastructure which supports flexible publication policies for both data and computing services providers.

Smooth interaction and loose coupling between data services and computing services: Typical usage by applications includes simple and easy data transformation or marshalling to feed into the next service. In order to obtain reliable results from the carbon cycle model, for example, scientists need to change one source and/or the combination of the inputs for sensitivity analysis. A desirable IT architectural style would achieve loose coupling among interacting software agents to allow users both to create services independently, and to produce new application from them. Additionally, IT infrastructure must support sharing, coordination, and configuration of environments for application programs and resources, depending on the user's requirements.

Ease of use: Even if the IT infrastructure were an innovative one that satisfies the above requirements, it must be easy to use for both users and service providers. End users should be able to access data and computing resources without the burden of installing special software and taking care of security issues such as certificate management. Data and service providers should be able to easily make their resources available as services. Administrators and leaders of communities should be able to create virtual communities easily by configuring appropriate access control for services for each user. Therefore, we must provide an ease-of-use framework for publishing services and user interfaces (e.g. portal) which can be customized and extended easily.

2.2 Overall Design and Implementation

Comprised of observation and information systems for earth observations, the GEO Grid system implements an infrastructure for flexible, secure, and coordinated sharing of resources such as satellite data, field sensor data, and computing for simulations and data analysis.

2.2.1 Design Policy: First, in order to satisfy the requirements described in previous section, the GEO Grid system introduces the concept of a virtual organization (VO) (Foster and Kesselman, 2004), for its design, in which various data and computing resources are provided as services represented by standard protocols. A VO is a dynamic collection of individuals, institutions, and resources, in which sharing of data, computers, software, and other resources are highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. Figure 1 illustrates an overview of the GEO Grid system in which data services, processing services, and users each form VOs for their own purposes, such as disaster mitigation, weather prediction, or natural resource exploration. A VO categorizes users into groups according to their tasks or research activity, and the authorization to utilize services on the GEO Grid system is appropriately performed.

Second, in order to reduce the cost of software development and realize better interoperability with existing systems, the GEO Grid system uses standard technologies and protocols, such as Web services and the Grids.

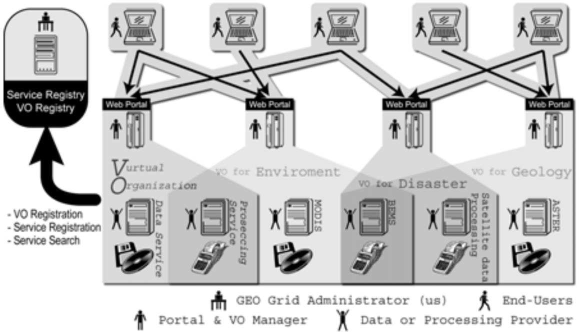


Figure 1. Overview of the GEO Grid System

2.2.2 Security: The GEO Grid system uses the GSI and VO-level authorization mechanisms. GSI is the standard technology for security in the Grid and a public-key-based, X.509 compliant system that relies on trusted third parties for signing user, host, and service certificates. For VO-level authorization, VOMS (VO Membership Service) (Alfieri and et al., 2005) is used. In the GEO Grid system, the most prominent feature is that the security framework is scalable in terms of the number of users, organizations, and resources. Most procedures required for security are automated and only critical operations are left to each entity's hand. In addition, this framework does not eliminate existing services which provide geoscientists with free content. Various security tools related to GSI and VOMS are setup on the resource-provider side, VO-administration side and Web portal of each VO.

2.2.3 Services: In the GEO Grid system, resources are provided to users as services. Typically services are implemented as Web services that can be deployed as Grid services with Globus Toolkit version 4 (GT4) (Foster, 2006). The GT4 makes heavy use of Web services mechanisms, which deals with issues of message handling, resource management, and security to support the development of distributed system components. GT4 comprises both a set of service implementations, and associated client libraries. The application services implemented with GT4 libraries are Web Service compliant and use GSI security mechanism. The GEO Grid SDK, which is described later, provides tools and APIs for facilitating implementation of such services. In addition, service providers are able to utilize predefined GT4 services such as job management (GRAM), Reliable File Transfer (RFT), Monitoring and Discovery Service (MDS) and Data Access and Integrations (OGSA-DAI).

2.2.4 Data Integration: In order to achieve database federation that can integrate various kinds of distributed data, the GEO Grid uses OGSA-DAI (OGSA-DAI project, n.d.), which is service-based database access software based on a Web service infrastructure, such as WSRF or SOAP. By using OGSA-DAI, a database query processing service with distributed joins over multiple, heterogeneous database resources can be implemented. For complicated application workflow, OGSA-DAI provides an activity framework which connects outputs of the previous task and inputs of the next task. For example, if a query result is too large to send back to a client, it will be send to a storage service. OGSA-DAI also supports VO-level authentication.

2.2.5 GEO Grid Toolkit: The GEO Grid toolkit is aiming at providing a set of software for the geosciences users to create and maintain the GEO Grid infrastructure easily and comfortably, by hiding complicated IT issues (e.g. security, data integration, computing resource management, and so on). The GEO Grid toolkit consist of three major components: One is GEO Grid Service Development Kit (SDK) for data and program owners; the others are GEO Grid Virtual Organization Tool (VOT), and GEO Grid Portal Development Kit (PDK) for scientists.

3. BACKEND STORAGE SYSTEM IN A GEO GRID SITE

3.1 Requirements of the Backend Storage System

The GEO Grid infrastructure consists of multiple GEO Grid sites which provide site-specific services over various resources. As described before, a storage system is one of the most important components because most geosciences applications correlate to the data. There are surely the following requirements for storage in each GEO Grid site: 1) from hundreds TB to petabytes scale disk capacity, 2) no data lost, 3) highly available service, 4) high-scalability in performance, in particular at concurrent access from multiple clients to the storage system, and 5) reasonable cost for system installation and operation.

3.2 Current Status

The GEO Grid has been developed and deployed the ASTER storage system as a component of the GEO Grid. ASTER (Yamaguchi and et al., 1998) is a high spatial resolution multi-spectral imaging radiometer on the Terra satellite, which was the first satellite of NASA's EOS (Earth Observing System) Program. The ASTER was developed by the Ministry of International Trade and Industry (MITI) of Japan, and the Earth Remote Sensing Data Analysis Center (ERSDAC) has been responsible for operation, data processing, and data distribution. ASTER Ground Data System at ERSDAC has accumulated the data since the successful launch in December, 1999, at the rate of approximately 70-100 gigabytes per day. Now all of the data is incrementally copied to the ASTER storage system located in National Institute of AIST, Japan, and the data is available through the web portal, to only researchers who have the data-use contract with ERSDAC (Yamamoto and et al., 2006). A use diagram of the ASTER storage system is illustrated in Figure 2.

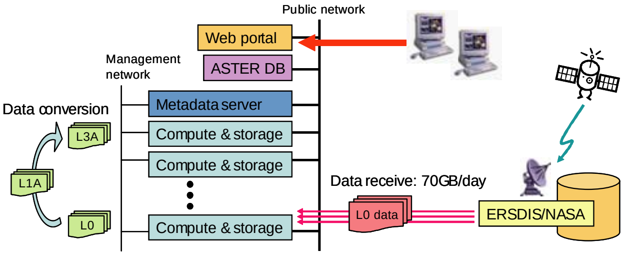


Figure 2: ASTER Storage System in AIST

3.2.1 ASTER Storage System: In the ASTER storage system, all data is stored in the hard disk drives of the storage nodes. Each node has dual Intel Xeon 3.8 GHz processors, 4 GB memory, and 7 TB disk space by 16-drives with RAID-6. At the end of July in 2007, 24 storage nodes and a single metadata server, which are connected to Gigabit Ethernet, provided totally about 170 TB disk space and 153 TB was already used for the data archives. Gfarm (Tatebe and et al., 2002), an open-source, distributed file system, is used to integrate disk space of the nodes. PostGIS, GIS enhancement of PostgreSQL, is used as a data management database.

Transmitted ASTER data from NASA is compressed and stored in a Gfarm file system. Data files are well balanced over storage nodes and replicated for fault tolerance. Normally raw (Level 0) data is converted to higher level. This conversion is also performed on the storage nodes. If a lot of data conversion is required at the same time, another cluster, which has 256 dual Intel Xeon nodes connected to Gigabit Ethernet, will be used.

Here are typical application programs on the ASTER storage system, in a flow from data transmission to publishing.

FTP: 3 sets of Level 0 data are received from ERSDAC by parallel FTP transmission. Each set contains 8-hours observation data which is separated to about 30 files. Average data size per day is about 70 ~ 100 GB.

REGISTER: The received data is registered to the database per observation-period basis and moved to the appropriate directory.

BZIP2: This compresses a received data from NASA, before importing it into the Gfarm file system.

REPLICATE: This replicates a file which is stored in the Gfarm file system, by the Gfarm command (gfrep).

BUNZIP2: This decompresses Level 0 data stored in the Gfarm file system and extract files on a local working directory, for L1PGE program.

L1PGE: This produces two-dimensional scene images (Level 1A data) using a set of the observation data files. About 500 files are generated from the observation data of one day. All Level 1A data is archived in the ASTER storage system.

DTMSOFT: This generates ortho-rectified images and Digital Terrain Model (DTM) images using Level 1A data. About 10 files of Level 3A data is generated from 1 file of Level 1A data and kept in the ASTER storage system. Intermediate data is deleted after the data conversion. This program is executed 500 times per day in average, in a batch at night.

CONVERT: This converts Level 3A data to JPEG or PNG images for the Web browser.

3.2.2 Next Storage System: Our implementation choice of the ASTER storage system is to use free software and commodity hardware, instead of proprietary software and high-end servers and networks. At this level of scale, our implementation is quite successful to meet the requirements stated previously. The satellite of the earth observation becomes larger, however, and it is estimated that a new sensor on the satellite produces more than petabytes in its operation period. The storage system should have much more capacity than the one the ASTER storage system has. Currently diverse combinations of the storage architecture and tools are available but few knowledge and experiences for building such a storage system are not shared in common.

4. COMPARISON OF GFARM- AND LUSTRE-BASED STORAGE SYSTEMS

In this study, the Gfarm-based storage system and the Lustre-based storage system were built using 20 nodes of X4500 (Sun Fire X4500 Server, n.d.), which is possibly enhanced to the petabytes-scale. X4500 has 24 TB disk space and even if only 16 TB is available for application data area, 64 nodes provide 1 PB space. Both Gfarm and Lustre are open-source, distributed file system and they can be used in production situation. Because there is a big difference in design concept between them, however, we configured each system so that it achieves the best performance with given resources. The comparison was not only performed in the performance aspect but also operation cost. This section describes features and our configuration of both systems, and reports performance evaluation and operational evaluation.

4.1 Gfarm-based Storage System

Gfarm (Tatebe and et al., 2002) integrates multiple storage servers on a network and provides a distributed file system interface that users can access a file with a virtual directory tree. Gfarm has a unique feature that each file system node acts as a client of the Gfarm file system and an application program is launched on the node where input file is physically located. Due to this locality optimization, aggregated read and write access from multiple applications realizes super-scalable I/O performance. When data access is I/O throughput intensive and CPU processing is also required, Gfarm can provide outstanding performance.

In our configuration, 1 node is allocated to the metadata server and the rest 19 nodes are allocated to the file system server, as shown in Figure 3. The nodes are connected to Gigabit Ethernet. Each node runs Solaris 10 and Gfarm version 1.4.1 which is customized to support Solaris 10/Opteron. PostgreSQL is started on the metadata server node and the Gfarm metadata is stored on the UFS. The metadata cache server (gfarm_agent) is also setup with master-mode on the metadata server node. In the file system node, 7 sets of 5-drives RAID-Z form 1 ZFS pool for storage area of Gfarm. Because data will never be corrupted in ZFS when write caching is enabled, the caching is enabled. The other ZFS parameters are set to default. As a result, each node provides 13.5 TB (12.5 TB after format). In 13 drives which are not used for Gfarm, two of them are used for system area and others are set as spare.

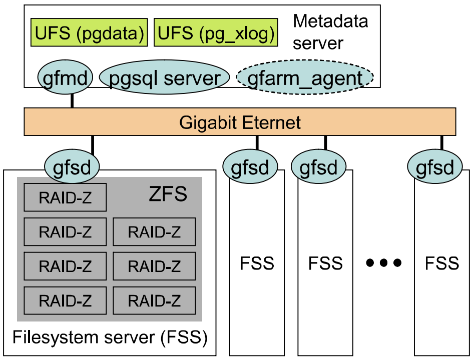


Figure 3: Configuration of the Gfarm-based Storage System

4.2 Lustre-based Storage System

Lustre (Lustre, n.d.) is designed for a shared file system in a cluster. Lustre provides POSIX compliant interface, and because of high scalability, it is widely used in the HPC clusters. Normally, clients and storage servers are separated in a Lustre file system. The network among nodes is TCP/IP or high-speed network such as Infiniband or Myrinet. The storage server (OSS: Object Storage Server) serves multiple OST (Object Storage Target) and one file is fragmented and stored over several OSTs. This enables aggregation of disk capacity at each server and provides higher throughput than what a single device can theoretically achieve. By using striping, it is possible to create a file larger than petabytes.

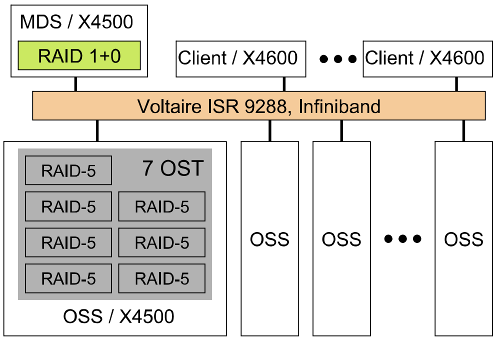


Figure 4: Configuration of the Lustre-based Storage System

Figure 4 shows our configuration of the Lustre-based storage system. 10 nodes of X4500 are allocated to OSS and 1 node is to MDS (Metadata Sever). Disk layout of OSS is almost same as Gfarm's. 7 sets of 5-drives RAID-5 are configured by software and each set is setup as OST of Lustre. In MDS, 4-drives RAID-1+0 is allocated to MDT which stores metadata of Lustre. In this system, 16 client nodes are additionally prepared for data processing. The client node is X4600 which has 8 dual-core AMD Opteron. Each node runs Linux and its kernel is patched for the Lustre version 1.6.2. All nodes are connected to both Gigabit Ethernet and Infiniband 4X. Data access through Lustre uses Infiniband.

4.3 Performance Evaluation

The performance evaluation was performed under practical-scale storage system, using real dataset. DTMSOFT was used for a sample application because DTMSOFT is the most frequent execution program in the ASTER storage system. Before the experiment, read/write ratio of DTMSOFT was analyzed when the file was stored on a Gfarm file system. The result shows 7.3% of execution time is for data access. The rest of the execution time is supposed to be consumed by data processing with CPU. Therefore the I/O throughput does not seem to be a bottleneck but there is another concern that excessive, concurrent metadata access by many clients may slow down the execution.

In preparation of the experiment, it was very difficult to import all data stored in the ASTER storage system in AIST, into the storage systems used for the experiment. 3137 data was sampled and imported to the systems. Then many copies were created with a different file name, until about 75% of the capacity was filled. Many files with 0 bytes were also created so that the number of the metadata entries was similar to the production system's. Eventually the Gfarm-based storage system stored 173 TB data (73% of disk capacity) and 18 millions' files over 19 file system nodes. The Lustre-based storage system stored 93 TB (78% of disk capacity) and 9.5 millions' files over 10 OSS.

The execution time of DTMSOFT which processes a set of 3137 sample data was measured in the experiment. DTMSOFT was invoked to an idle CPU slot accordingly and executed in parallel. The measurement was from the first start of DTMSOFT execution to the end of the last DTMSOFT execution. In the Gfarm-based storage system, a working directory was created at each execution because Gfarm's metadata access should be carefully minimized and POSIX-like interface was not available in this environment. The input file on the Gfarm file system was copied to the working directory by the Gfarm command (gfexport) and DTMSOFT accessed the copied file. The output file was imported to the Gfarm file system by another Gfarm command (gfreg) and unnecessary intermediate data was just deleted. In the Lustre-based storage system, there is less concern about metadata access than Gfarm's and POSIX interface is available. DTMSOFT directly accessed the input file on the Lustre file system. All outputs including intermediate data was once written to the Lustre file system, and unnecessary data was deleted at the end of the execution.

Table 1 shows the execution time with parameters. In the Gfarm-based storage system, there are two parameters. Local-io is the case that input data is always stored on the node where the DTMSOFT is launched. Remote-io is the case that input data is always not stored on the node where the DTMSOFT is launched. In the latter case, a ring topology is created and the clients read the data from next node through the network. In the Lustre-based system, all input data is striped (stripe count is 3 and stripe size is 2MB) but two cases whether output is striped or not, are attempted.

For the comparison, 16 nodes were used for data processing in both storage systems. Each node invokes at most 4 DTMSOFT at the same time. The result of Table 1 shows the Lustre-based storage system achieves slightly faster execution time than the Gfarm-based. Ratio in the table indicates parallel efficiency, which is calculated from sequential execution of processing 30 sample data. For the Gfarm-based, the data for the sequential execution was directly stored on ZFS/RAID-Z. For the Lustre-based, the data was stored on Ext3/RAID-1. Each execution time was 27911 [sec] and 24797 [sec]. The ratio is calculated with assumption that the 3137/30 times of those sequential execution time with 30 samples is almost equal to the sequential execution with 3137 samples by a single process. From this result, both systems achieved expected speed-up in 64 parallel executions. Furthermore, 256 parallel executions achieved super-linear efficiency in the Lustre storage system. The reason is because AMD PowerNow raises CPU clock frequency during execution of 16 processes per node.

Parameters	PE×Node	Exec. time	Ratio
Gfarm: local io	4×16	40310 [sec]	1.01
Gfarm: remote io	4×16	40460	1.00
Lustre: No striping	4×16	40142	1.14
Lustre: Striping	4×16	39579	1.15
Lustre: No striping	16×16	9441	1.21
Lustre: Striping	16×16	9330	1.22

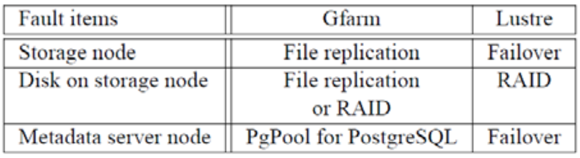
Table 1: Execution time to process 3137 samples by DTMSOFT

4.4 Operational Cost

Based on requirements for operating the ASTER storage system, the methodologies of high availability, maintenance work issues are compared between Gfarm version 1.4.1 and Lustre version 1.6.2. This section describes major differences among them.

4.4.1 High Availability: The ASTER storage system receives Level 0 data from ERSDAC everyday and serves geoscientists to access the necessary Level 3A data anytime. High availability is vital in this use case. In order to realize high availability, some mechanism should be prepared for faults.

Table 2 shows differences in configuration between Gfarm and Lustre for fault tolerance. The Gfarm-based storage system relies on file replication. A file is replicated in advance so that applications can access the data if a part of the originals/replicas is lost. File replication is not suitable for frequently updated data, but users can specify the replication level per file. The Lustre-based storage system supports a failover function with Heartbeat (Linux HA, Heartbeat, n.d.). However, a backup node must share a storage device with an in-service node and it requires some installation cost. Lustre version 1.6.2 assumes that RAID is used under OST, though Lustre may support RAID-1 functionality in the future.



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

Table 2: Possible configuration for fault tolerance

When any fault occurs at the storage node during file access, the access will be an error in the Gfarm-based storage system. However, if an application retries the access and there is a replica on another node, the access will be switched to the new server. If failover is setup in the Lustre-based storage system, fault tolerance is transparent. If failout is configured, an application receives an error in file access and the file will not be accessed until the concerned node recovers.

4.4.2 Maintenance Work Issues: Scheduled maintenance can be performed without stopping the entire system in both Gfarm-based and Lustre-based. System administrators only need to notify the node detachment to the metadata server. In order to continue the service to access the file stored on the detached node, methods shown in Table 2 for fault tolerance is necessary.

In the addition or exchange of the storage node, data migration is essential. Neither Gfarm nor Lustre provides a special tool for this. System administrators need to check which data should be migrated to the other, and to copy and rebalance of disk usage manually, with a combination of several commands.

In the operation of Gfarm, configuration of the metadata server and the metadata cache server significantly affects on the performance. The metadata server should have sufficient memory and system administrators periodically vacuums PostgreSQL database to delete unnecessary records. 4 GB memory is required to store 8 millions' files on the metadata cache server though this depends on the file name's length.

Gfarm does not depend on underlying file system and can make use of a powerful file system such as ZFS. Lustre depends on the Linux kernel and it sometimes restricts users to choose their favorite Linux kernel version or Linux distribution.

When system administrators use Lustre, they can allocate compute resource and storage resource separately. This flexibility is not available in Gfarm. If CPU load dynamically changes or a new application arrives, compute resource might become surplus or lack in the Gfarm-based storage system. Gfarm would be useful when CPU load is constantly moderate and administrators want to save money for additional data processing nodes.

5. DISCUSSION

Both Gfarm-based and Lustre-based storage systems showed scalable performance with 64 parallel executions of DTMSOFT. We concerned collision of metadata access might slow down the execution, in particular about 10 millions' files are stored in the storage systems. However, by minimizing metadata access in the Gfarm-based storage system, it did not become a bottleneck. Metadata access performance of Lustre is excellent and DTMSOFT could directly access the files stored on the Lustre-based storage system without performance degradation. However, when a new data conversion or analysis program is executed on the storage systems, the file access pattern of the the program must be analyzed. The way of analyzing the data access patterns, and estimating performance on several types of the storage system would be useful for the site administrators. As a comparison result, there are more differences in operation cost between the two systems, rather than performance.

Due to the difference of the design concept, we configured the system so that each can achieve the best performance. Hence, the following are not considered in the comparison: file allocation to gain locality optimization is sometimes complicated in Gfarm. We assume that our application allows locality optimization. We also assume that Infiniband is available, though it is not commodity hardware. If Gigabit Ethernet is used instead of Infiniband, I/O throughput of each Lustre client node would be limited.

6. CONCLUSION

The GEO Grid infrastructure provides secured access to whole data sets related to earth observations, and integrates the data and applications. It introduces VO-based security framework used in the Grid, and standard, Web service based protocols for integration of various services. The GEO Grid system architecture is designed to meet functional requirements to IT technology from geosciences. It is implemented by integration and customization of the existing Grid tools.

Since most GEO Grid applications are data-centric, there is a strong demand for a site-internal large-scale storage system. Moreover the capacity requirement is constantly increasing. Even so, it was not clear how the site should build and operate it, and each site is spending a lot of time to learn the latest technology. In this paper, two ways of building the next storage system for geosciences are introduced and compared, based on our experiences of the ASTER storage system. One is a Gfarm-based storage system and the other is a Lustre-based storage system. The performance evaluation was performed by real data sets and data conversion programs, and both systems achieved fairly scalable performance. On the other hand, there are several differences in operational issues between the two systems. The differences come from design concept and software maturity. These experiment and investigation results indicate that a factor of choosing either storage system is operation cost. The investigation result in Section 4.4 would be useful for the GEO Grid site administrators and other people who are planning to build the data archives for another satellite sensor. Evaluation of other storage systems such as Bigtable/Google file system (Chang and et al, 2006, Ghemawat and et al., 2003), scalable cluster databases, and so on is our future work.

ACKNOWLEDGEMENT

The performance evaluation part of this work was performed in collaboration between National Institute of AIST and Tokyo Institute of Technology. We would gratefully thank Takeshi Nishikawa, Satoshi Matsuoka, and Takeshi Yamanashi from Tokyo Institute of Technology, Shuuichi Ihara and Junichi Koike from Sun Microsystems K. K.

The research for the ASTER data described in this paper was partially supported by the Ministry of Economy, Trade and Industry in Japan.

REFERENCES

Alfieri, R. and et al., 2005. From gridmap-file to VOMS: managing authorization in a Grid environment. Future Generation Computer Systems 21(4), pp. 549–558.

Chang, F. and et al., 2006. Bigtable: A Distributed Storage System for Structured Data. In: 7th Symposium on Operating System Design and Implementation (OSDI).

Foster, I., 2006. Globus Toolkit Version 4: Software for Service Oriented Ssystems. In: IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp. 2–13.

Foster, I. and Kesselman, C. (eds), 2004. The Grid: Blueprint for a New Computing Infrastructure. 2nd edn, Morgan Kaufmann Publishers.

Ghemawat, S. and et al., 2003. The Google File System. In: 19th ACM Symposium on Operating System Principles.

Justice, C. and et al., 1998. The Moderate Resolution Imaging Spectroradiometer (MODIS): Land remote sensing for global change research. IEEE Trans. on Geosci. Remote Sensing 36(4), pp.1228–1249.

Linux HA, Heartbeat, n.d. http://www.linux-ha.org/Heartbeat/.

Lustre, n.d. http://www.lustre.org/.

OGSA-DAI project, n.d. http://www.ogsadai.org.uk/.

Sun Fire X4500 Server, n.d. http://www.sun.com/servers/x64/x4500/.

Tatebe, O. and et al, 2002. Grid Datafarm Architecture for Petascale Data Intensive Computing. In Proceedings of the 2nd IEEE/ACM Symposium on Cluster Computing and the Grid, pp. 102-110.

Yamaguchi, Y. and et al., 1998. Overview of Advanced Spaceborne Tthermal Emission and Reflection Radiometer (ASTER). IEEE Trans. Geosci. Remote Sensing 36(4), pp. 1062–1071.

Yamamoto, N., 2006. GEO Grid: Grid Infrastructure for Integration of Huge Satellite Imagery and Geoscience Data Setsets. IEEE International Conference on Computer and Information Technology (CIT), pp. 75.