6-2 A Survey of Distributed Storage and Parallel I/O Technique for Security Incident Analysis

KAMISAKA Kikuko

Recently, unauthorized access and virus attack via open Internet are a big issue in many fields. With the spread of high-speed and large-scale network, a large amount of data is required for analysis of security incident information. Therefore, it is necessary to access and process these data with high-speed.

In this paper, we survey recent distributed storage and parallel I/O techniques for the high-speed incident analysis, and consider possible applications of these techniques in a research area of security.

Keywords
Parallel I/O, Distributed storage, Cluster system, Grid computing

1 Introduction

In recent years, unauthorized access and viral cyber attacks on business networks, Internet service providers, and other organizations have become a serious issue. One of the most widely employed countermeasures against such cyber attack attempts via the Internet is the installation of firewalls and/or intrusion detection systems ("IDS"), which monitor such intrusions or attacks and analyze each such incident. With each passing year, the manner of unauthorized access and viral attacks have become more sophisticated and diverse; accordingly, technology for high-speed and efficient data processing is required to carry out real-time analysis of data collected on traffic, viruses, incidents, and log information.

At the same time, the growth of broadband and high-speed network, such as Gigabit/10-Gigabit networks, has led to increasingly complex and extensive network infrastructures used by enterprises and other organizations. The volume of data passing through the Internet and intranets has also increased dramatically, and we can expect a corresponding substantial increase in the volume of data that must be collected for analysis of future unauthorized access and viral attacks. It is difficult to process such large volumes of data in systems of multiple analytical servers and storage devices. Accordingly, computer resources will need to feature sufficiently high computational performance and large-scale storage domains, offering high availability and scalability.

This paper investigates the potential of distributed storage and parallel I/O techniques as a data platform for efficient and high-speed processing of large volumes of data in security incident analysis.

2 Storage architecture

This section will examine the extent to which storage architectures currently in use can provide a mechanism for storing such traffic and log data.

Traditional RAID systems and DAS (Direct Attached Storage), which are directly connected to servers via a SCSI interface, had been commonly used as a storage system. However, in DAS, since the storage devices connected to multiple servers are scattered, the management and operation of storage system becomes complicated, and it cannot use the storage capacity of each device effectively. Later, in response to increasing architectural scales, storage networking techniques were developed to enable more efficient data management. In particular, systems came into popular use such as the SAN (Storage Area Network) and NAS (Network Attached Storage), overcoming the shortcomings of DAS through networking. The SAN is a dedicated high-speed network connected to a server and multiple storage devices, permitting effective utilization of disk resources through the integration of distributed storage. Access to storage is made via a network employing high-speed and high-priced Fibre-Channel-based FC protocols or the inexpensive IP-based iSCSI protocol. However, SAN does not support file-sharing, since access to the storage devices takes place at the block level. Thus, some form of file system must be provided in a higher layer to allow for the handling of files. In contrast, NAS was developed as a file-level data storage system for the purpose of sharing files among computers on a network. Since NFS (Network File System) and CIFS (Common Internet File System) are used as the file-transfer protocols, file sharing can take place between different platforms. However, data access performance of these systems is poor compared to the block-based SAN.

Recent studies in storage networking have focused on back-up, recovery, and remote-mirroring functions in the context of availability, as well as high-speed restoration, storage sharing, and storage virtualization, in the context of efficient management. Additionally, recent tactics to enhance storage access performance have included the installation of object-based storage architecture enabling high-speed data management. In recent years, distributed grid-based and cluster-based storage architectures with high scalability and I/O performance have been receiving greater attention, with the growing importance of data-intensive processing and applications. The next section will introduce file systems and parallel I/O techniques used in high-performance computing.

3 Parallel file system technique in HPC applications

In HPC (High-Performance Computing) applications, particularly those requiring scientific computations, the hierarchical structure enabling parallel I/O consists of an I/O library, I/O middleware, and a parallel file system[1]. Figure 1 shows the I/O hierarchical structure for scientific computations.

A high-level I/O library is provided for storage and parallel computation of massive amounts of data. The HDF5 (Hierarchical Data Format Version 5) is a file-format library developed by NCSA for storing scientific

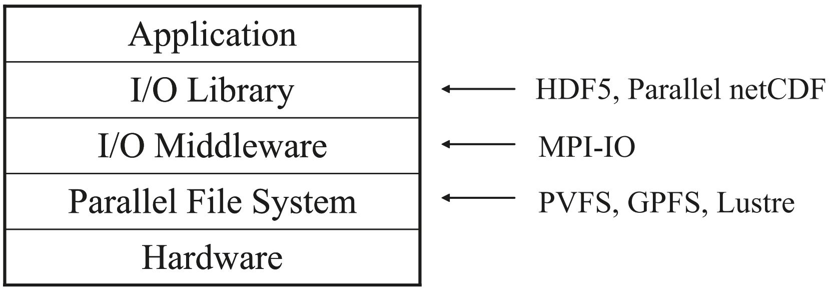

Fig.1 *I/O hierarchical structure for scientific computations*

data. The netCDF (Network Common Data Formant) library supports multi-dimensional data formats and is commonly used in computational applications in the field of astronomy and meteorology. The parallel netCDF uses MPI-IO to support parallel I/O.

On the I/O middleware level, coordinated operation between multiple processors is carried out to perform I/O processing. The MPI-IO, a representative example, is a high-speed I/O performance version of MPI (Message Passing Interface I/O) for distributed memory parallel computers.

Below this layer is a parallel file system enabling high-speed parallel access on the file level.

3.1 Cluster file system: GPFS, lustre, and PVFS

Generally, cluster system types consist mainly of the HA cluster, the load-balancing cluster, and the HPC cluster. The HA cluster system improves availability by deployment of multiple servers in parallel. In the event of failure in the system, system downtime is minimized by redundancy techniques, and high availability is secured by remote back-up and mirroring. The load-balancing cluster features architecture to distribute load in order to minimize access concentration on servers. The HPC cluster is designed for scientific computations, with large-scale high-speed calculations enabled through parallel processing, both of the data and the processing itself, via distribution to multiple calculation nodes.

In this section we will review the file system in existing HPC clusters in an examination of efficient data management.

Parallel file systems used in HPC clusters include the GPFS (General Parallel File System)[2], Lustre File System[3][4], and PVFS2[5].

The GPFS is a POSIX-compliant cluster file system for shared disks developed by IBM. Figure 2(a) shows an example of cluster architecture using GPFS. This system may be used for AIX or Linux clusters and, as storage architecture, may be applied to shared block devices such as SAN. The GPFS is a distributed shared file system offering high-speed I/O throughput via distributed storage of a single file on multiple server disks by file striping, allowing higher access speeds than available with NFS. Simultaneous access to a sin-

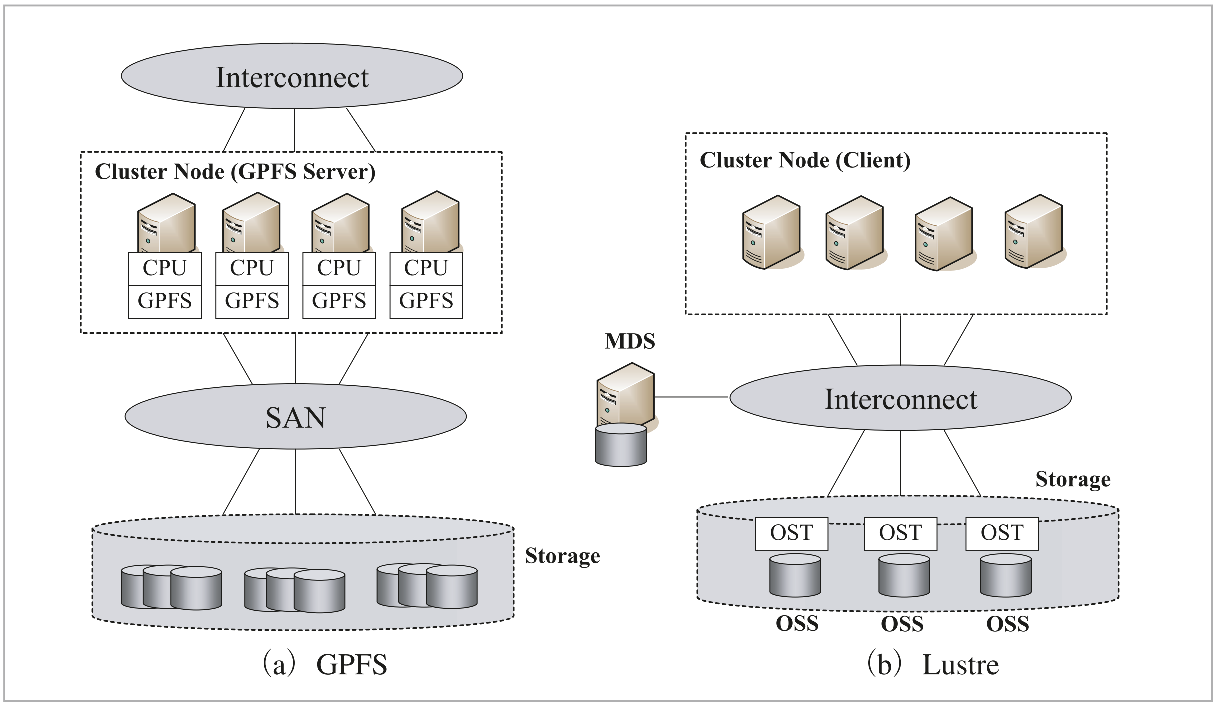

Fig.2 System configuration example of GPFS and lustre

gle file is also possible through the use of optimized MPI-IO. Furthermore, even when one of the GPFS servers is down, the remaining GPFS servers can take over processing, facilitating recovery from failure.

The Lustre File System, on the other hand, is an open-source distributed shared file system developed by Cluster File Systems, Inc. for Linux and Solaris. Figure 2(b) shows an example of cluster architecture using the Lustre File System. Lustre may be used on a variety of networks, and consists of an MDS (MetaData Server), which manages metadata; an OST (Object Storage Target), which manages storage of file data; and clients, or nodes. The metadata is separated from the I/O data and the file data are stored on each OST by striping. This structure allows simultaneous access from multiple clients, while parallel I/O for large files stored over multiple OSTs is enabled through MPI-IO. However, failure of the MDS will prevent access to OST, and so multiple MDS units should be provided to create a redundant architecture. Connection may fail in the event of concentrated access from several hundred clients in an I/O node, and so this parallel file system may be regarded as placing priority on performance over stability.

In reference[6], Yu et al. suggested that the file striping process reduces performance and have succeeded in improving performance through split writing and hierarchical striping.

The PVFS2 (Parallel Virtual File System 2), developed by a team based mainly at Clemson University, is a file-based storage model suited for open-source clusters. The PVFS stores file data in local Linux disks by striping. Communication between nodes is based on TCP/IP; as a result, the network will become bottlenecked when handling large volumes of data, preventing scalable performance in cases of simultaneous access from multiple applications.

3.2 Grid file system: Gfarm

One of the parallel processing architectures in HPC is grid computing. Sophisticated grid computing is executed by connecting multiple computers present on a wide-area network (WAN). The Gfarm (Grid Datafarm) developed by Advanced Industrial Science and Technology (AIST)[7][8] is a parallel file system for grids, designed to execute data-intensive applications using locally distributed computing and storage resources. Figure 3 shows a Gfarm architecture. Gfarm applications extend from small- to large-scale PC clusters, and the system is also designed for

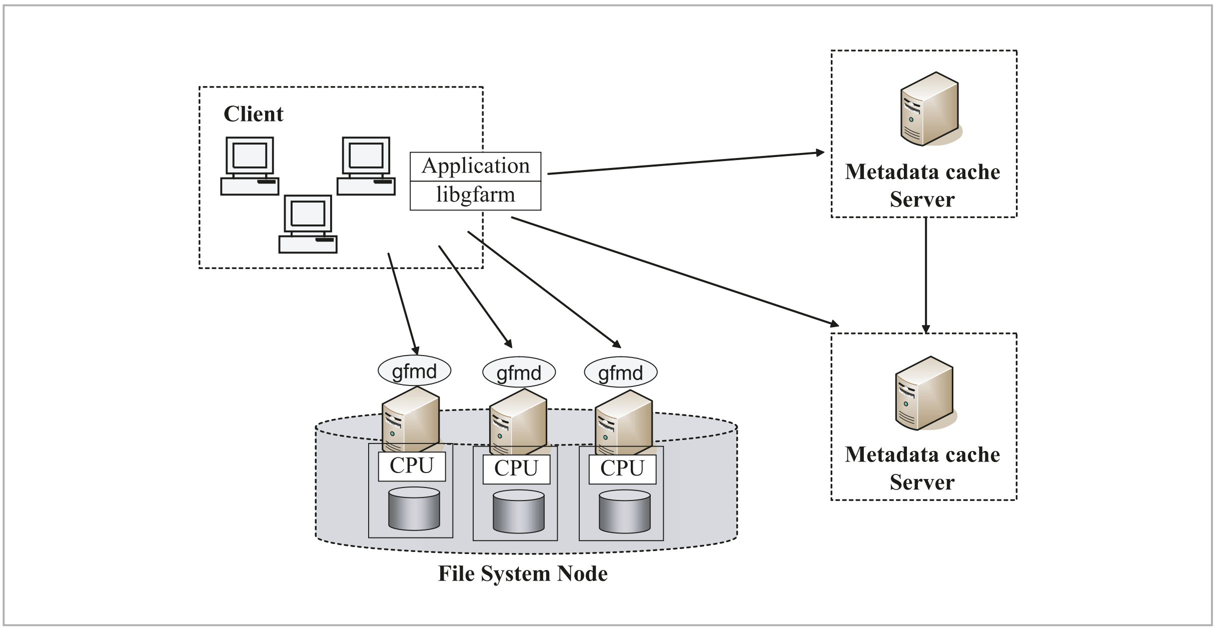

Fig.3 Gfarm architecture

use in high-speed processing of large-scale data analysis in a wide-area distributed environment. The Gfarm architecture consists of clients with libgfarm libraries, a metadata cache server, servers to manage metadata, and file system nodes that integrate computation nodes and I/O nodes. High-performance parallel I/O is achieved by striping using the local I/O of the computation nodes when executing data-intensive applications. Furthermore, storage of multiple copies of files within the WAN enables load balancing and provides fault tolerance, with clients unaffected by the actual location of a given file when using the system. As indicated in reference[7], experimental results have revealed NFS-equivalent performance in local environments.

4 Distributed file system technology

Distributed file systems designed for web applications that are currently receiving attention include the GFS (Google File System)[9], BigTable[10], and MapReduce[11] developed by Google. GFS is a Linux-based distributed file system, BigTable is a distributed storage system, and MapReduce is a programming model that carries out distributed data processing. The GFS consists of three types of elements: a master that manages the chunk server location, file name, and lock; multiple chunk servers for data storage; and clients. Instead of making use of sophisticated machines, the system employs large numbers of inexpensive hardware components for distributed data storage, resulting in highly redundant data management to counter failures. This structure offers improved performance at the cost of POSIX compliance, and performance remains limited in R/W operations with small files and random access. Expectations are high for MapReduce[10], which excels in parallel data processing for large clusters in applications involving multi-core environments. MapReduce separates large-volume data processing into a Map task and a Reduce task, distributes the processing tasks, and executes these tasks independently. The system has thus facilitated the description of parallel processing programs, which may be written as a combination of tasks. The Map task decomposes and extracts data, and the Reduce task aggregates and computes the data and outputs the results. This structure permits high-speed processing in multi-core environments, web searching, and batch processing. Reference[12] reports on the implementation of the MapReduce library for SMP clusters. The open source versions of GPFS, MapReduce, and BigTable consist of HDFS (Hadoop Distributed File System), Hadoop MapReduce, and hBase, respectively.

5 Conclusions

The expansion of network scale seen in recent years has necessitated the development of techniques allowing efficient storage and high-speed access to large volumes of data in order to monitor cyber attacks and to perform incident analysis.

Based on the concept that large-scale data platforms offering high redundancy and high scalability are essential for successful security incident analysis, this paper presented an investigation of storage architecture, distributed storage techniques, and parallel I/O technology from the viewpoint of HPC applications. In the past, however, the development of such storage architectures and file systems were architecture-specific or aimed at applications in different fields; the effectiveness of the combined use of these architectures and systems remains to be seen. A variety of schemes may be envisioned depending on the types, volumes, and intended purposes of the data, and an optimized system must be designed based on the relevant system requirements for security incident analysis. In future studies, we intend to investigate the shortcomings of existing technologies such as distributed storage and parallel I/O in terms of security applications, and will propose new techniques to complement these technologies.

References

1 Robert Ross, Rajeev Thakur, and Alok Choudhary, “Achievements and Challenges for I/O in Computational Science”, Journal of Physics: Conference Series (SciDAC 2005), pp.501-509, 2005.

2 Frank Schmuck and Roger Haskin, “GPFS: A Shared-Disk File System for Large Computing Clusters”, In Proc. the Conference on File and Storage Technologies (FAST’02), 28-30, pp.231-244, Jan. 2002.

3 “Lustre File System”, http://wiki.lustre.org/index.php?title=Main_Page.

4 BRAAM, P. J., AND SCHWAN, P. Lustre: The Intergalactic File System. In Proc. the Ottawa Linux Symposium (2002), pp.50-54.

5 “PVFS2”, http://www. pvfs.org/

6 Weikuan Yu, Jeffrey Vetter, R. Shane Canon, and Song Jiang, “Exploiting Lustre File Joining for Effective Collective I/O”, In Proc. The 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid ‘07), May 2007.

7 “Gfarm”, http://datafarm.apgrid.org/index.ja.html

8 Yusuke Tanimura, Yoshio Tanaka, Satoshi Sekiguchi, and Osamu Tatebe, “Performance Evaluation of Gfarm Version 1.4 as a Cluster Filesystem”, In Proc. the 3rd International Workshop on Grid Computing and Applications (GCA 2007), pp.38-52, 2007.

9 Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, “The Google system”, In Proc. of the 19th ACM SOSP (Dec. 2003), pp.29-43.

10 Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable: A distributed storage system for structured data. In 7th USENIX Symposium on OSDI, pp.205-218, Nov. 2006.

11 Jeffrey Dean and San jay Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters”, In Proc. OSDI’04: Sixth Symposium on Operating System Design and Implementation, pp.137-150.

12 Colby Ranger, Ramanan Raghuraman, Arun Penmetsa, Gary Bradski, and Christos Kozyrakis, “Evaluating MapReduce for Multi-core and Multiprocessor Systems”, In Proc. IEEE 13th International Symposium on High Performance Computer Architecture (HPCA 07), pp.13-24.

***KAMISAKA Kikuko**, Ph.D.*

Expert Researcher, Traceable Secure Network Group, Information Security Research Center

Networks Security