An On-line Backup Function for a Clustered NAS System (X-NAS)

Yoshiko Yasuda, Shinichi Kawamoto, Atsushi Ebata, Jun Okitsu, and Tatsuo Higuchi
Hitachi, Ltd., Central Research Laboratory
1-280 Higashi-koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan
Tel: +81-423-23-1111, Fax: +81-423-27-7743
e-mail: {yoshikoy, skawamo, ebata, j-okitsu, higuchi}@crl.hitachi.co.ip

Abstract
An on-line backup function for X-NAS, a clustered NAS system designed for entry-level NAS, has been developed. The on-line backup function can replicate file objects on X-NAS to a remote NAS in real-time. It makes use of the virtualized global file system of X-NAS, and sends NFS write operations to both X-NAS and the remote backup NAS at the same time. The performance of the on-line backup function was evaluated and the evaluation results show that the on-line backup function of X-NAS improves the system reliability while maintaining 80% of the throughput of the X-NAS without this function.

1. Introduction

An entry-level NAS system is convenient in terms of the cost and the ease of management for offices with no IT experts. However, it is not scalable. To solve this problem, X-NAS, which is a simple, scalable clustered NAS architecture designed for entry-level NAS, has been proposed [6]. Like conventional NAS systems, it can be used for various clients, such as those using UNIX and Windows[1]. X-NAS aims at the following four goals.

- Cost reduction by using entry-level NAS as an element
- Ease of use by providing a single-file-system view for various kinds of clients
- Ease of management by providing a centralized management function
- Ease of scaling-up by providing several system-reconfiguration functions

To achieve these goals, X-NAS virtualizes multiple entry-level NAS systems as a unified system without changing clients' environments. In addition, X-NAS maintains the manageability and the performance of the entry-level NAS. It also can easily be reconfigured without stopping file services or changing setting information. However, when one of the X-NAS elements suffers a fault, file objects on the faulty NAS system may be lost if there are no backups. To improve the X-NAS reliability, a file-replication function must therefore be developed.

The goal of the present work is to introduce an on-line backup function of X-NAS that replicates original file objects on X-NAS to a remote NAS for each file access request in real-time without changing the clients' environments. The performance of the on-line backup function was evaluated and the evaluation results indicate that X-NAS with the on-line backup function improves the system reliability while maintaining 80% of the throughput of standard X-NAS.

[1] Windows and DFS are trademarks of Microsoft Corporation. Double Take is a trademark of Network Specialists, Inc. All other products are trademarks of their respective corporations.

2. On-line backup function for X-NAS

To improve the reliability of X-NAS, an on-line backup function for X-NAS has been developed. (Since the details of the X-NAS structure are discussed in another paper [6], they are not described here.) The on-line backup function consists of many sub-functions. Among these sub-functions, we focus on on-line replication, the heart of the on-line backup function, in this paper. The on-line replication replicates files of X-NAS to a remote NAS, which is called a backup NAS, in real-time for each file access request.

2.1. Requirements

The on-line backup function of X-NAS must meet the following requirements:

- Generate replicas of file objects in real-time in order to eliminate the time lag between the original data and the replicas.
- Use a standard file-access protocol such as NFS to communicate between X-NAS and the backup NAS in order to apply as many kinds of NAS as clients need.
- Do not change clients' environments in order to curb their management cost.

2.2. On-line replication

There are several methods for replicating file objects to remote systems via an IP network. One method is to use a block I/O [5]. Since using a block I/O is a fine-grain process, all file objects are completely consistent with copied objects. However, the system structure is limited because the logical disk blocks of the objects must be allocated to the same address between the original data and its replica. Another method is to change the client's system. DFS [1] is a simple method for replicating file objects to many NASs. It replicates file objects in constant intervals but not in real-time.

Xnfsd and the management partition in X-NAS enable the centralized management of many NAS elements and provide a unified file system view for clients (Fig. 1). Xnfsd is a wrapper daemon and receives an NFS operation in place of the NFS server and sends the operation to others. On-line replication of X-NAS makes use of Xnfsd in order to copy file objects to the backup NAS. By extending this function, Xnfsd sends the NFS operation not only to the NFS servers on the X-NAS but also to the NFS servers on the backup NAS. All file objects can thus be replicated in real-time for each NFS operation.

2.2.1. Operations

NFS operations handled in X-NAS can be divided into four categories. Category 1 is reading files; category 2 is writing files; category 3 is reading directories; and category 4 is writing directories. Xnfsd sends NFS operations belonging to categories 2 and 4 to both X-NAS and the backup NAS at the same time. On the other hand, NFS operations belonging to categories 1 and 3 are not sent to the backup NAS.

When a UNIX client sends a WRITE operation for file *f* to X-NAS, Xnfsd on P-NAS (parent NAS) receives the operation in place of the NFS daemon. Figure 1 shows the flow of this operation, and Figure 2 shows the timing chart with or without the on-line backup function. Firstly, Xnfsd specifies a data partition that stores the file entity by using the inode number of the dummy file *f* on the management partition (#1). Secondly, Xnfsd invokes a sub thread and then sends the WRITE operation to the backup NAS by

using the thread (#2). Thirdly, Xnfsd sends the WRITE operation to the NFS daemon on the specified C-NAS (child NAS), and then the C-NAS processes the operation (#3). Finally, Xnfsd waits for the responses of the operations from the NFS server on the C-NAS and from the backup NAS (#4), and then it makes one response from all the responses and sends it back to the client. We call this procedure a synchronized backup.

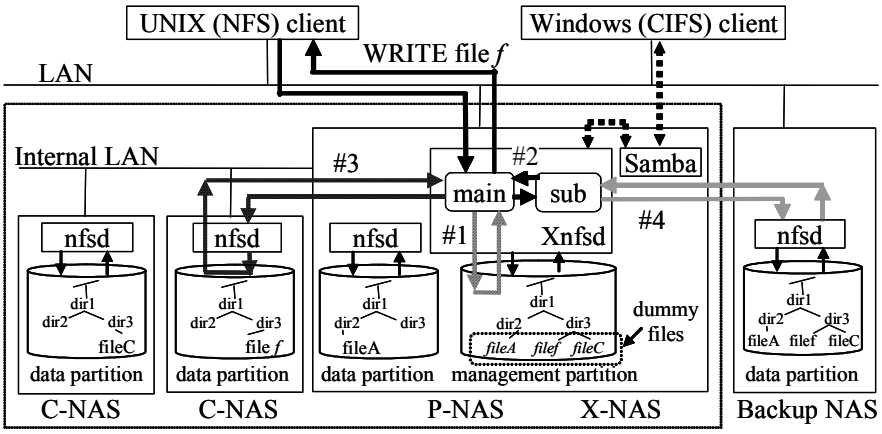

Figure 1: Flow of WRITE operation with online backup function.

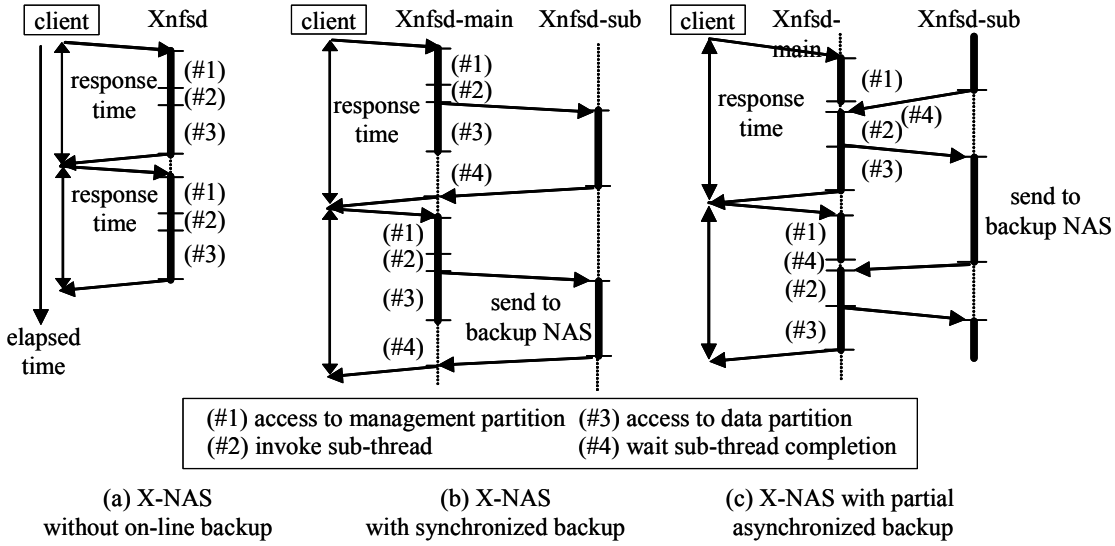

Figure 2. Timing charts of WRITE operation with or without on-line backup function.

2.2.2. Key features

An on-line backup function must guarantee the consistency of data between X-NAS and the backup NAS. To achieve this, Xnfsd waits for all responses from both one of the NFS servers on the X-NAS and the backup NAS for each NFS operation. However, waiting for the responses degrades total performance. To solve this problem, the performance of the on-line replication function must be improved through three key features as follows.

(1) Multi-threaded wrapper daemon

Xnfsd waits for all responses from both NAS systems. This incurs an overhead because of frequent accesses to the network and the disk drives. To reduce this cost, the main

thread of Xnfsd invokes a sub thread to send the file I/Os to the backup NAS. This feature enables X-NAS to process the disk accesses of both X-NAS and the backup NAS in parallel.

(2) File-handle cache
The cost of specifying the full path name and the file handle on the backup NAS is high because of frequent accesses to the network and the disk drives. To reduce this cost, X-NAS makes use of a file-handle cache, which records the correspondence between the file handle of the dummy file, i.e., the global file handle, and the file handle of the backup NAS.

(3) Partial asynchronized backup
Although the synchronized backup is a simple method, the execution cost is high because this method waits for all the responses from the NFS servers on the X-NAS and the backup NAS. A method that does not wait for the response from the backup NAS achieves the same performance as X-NAS without the on-line backup function. However, when X-NAS or the backup NAS becomes faulty, it is difficult to guarantee the consistency of data between X-NAS and the backup NAS. Using a log is one solution to guarantee the consistency. However, since the log size is limited, it is not a perfect solution for entry-level NAS, which usually has a small-sized memory. Furthermore, according to the X-NAS concept, the architecture must be simplified as much as possible. Xnfsd thus supports a partial asynchronized backup method in addition to the synchronized backup. Figure 2(c) shows the timing chart of the WRITE operation with partial asynchronized backup. In the method, after processing disk accesses to the data partition on the X-NAS element, Xnfsd sends back a response to a client without waiting for the response from the backup NAS. As a result, the client can send the next operation. The main thread of Xnfsd can perform the disk accesses to the management partition for the next operation during the waiting time for the response from the backup NAS.

3. Performance evaluation

To evaluate the on-line backup function of X-NAS, an X-NAS prototype based on the NFSv3 implementation was developed. We ran NetBench [3] and SPECsfs97 [4] on the X-NAS prototype with or without on-line backup function. In this evaluation, by taking account of permissible range for the entry-level NAS's users, we set the performance objective for X-NAS with the on-line backup function at 80% of the performance of X-NAS without the function. Throughput and average response time are used as the performance metrics. In this evaluation, we implemented the partial asynchronized backup function in the WRITE operation. This is because the ratio of the WRITE operations to all operations is higher than other operations in the workload mix of the benchmarks. Furthermore, since the file sizes used by the benchmark programs are from 100 to 300 KB, many WRITE operations are issued continuously and then each process in a WRITE operation could be overlapped.

3.1. Experimental environment

In the experimental environment, the maximum number of X-NAS elements is fixed to four. Each X-NAS element and the backup NAS configured with one 1-GHz Pentium III

processor, 1 GB of RAM and a 35-GB Ultra 160 SCSI disk drive running Red Hat Linux 7.2. For the NetBench test, one to eight clients running Windows 2000 Professional were used. The clients, P-NAS, C-NASs, and the backup NAS were connected by 100-Megabit Ethernet because most offices still use this type of LAN.

3.2. Results

Figures 3 and 4 show the results of our performance evaluation in terms of throughput and average response time. The throughputs of X-NAS with the synchronized backup function are about 80% of those without the function. Under an experimental environment with NetBench, the average response time for X-NAS with the function is about 1.2 times higher than that for X-NAS without the function. On the other hand, under an experimental environment with SPECsfs, the average response time for X-NAS with the function is about 1.4 times higher than the time for X-NAS without it. Although the partial asynchronized backup can improve both throughput and average response time by several percentage, the performance objective for the response time in the case of SPECsfs cannot be achieved yet.

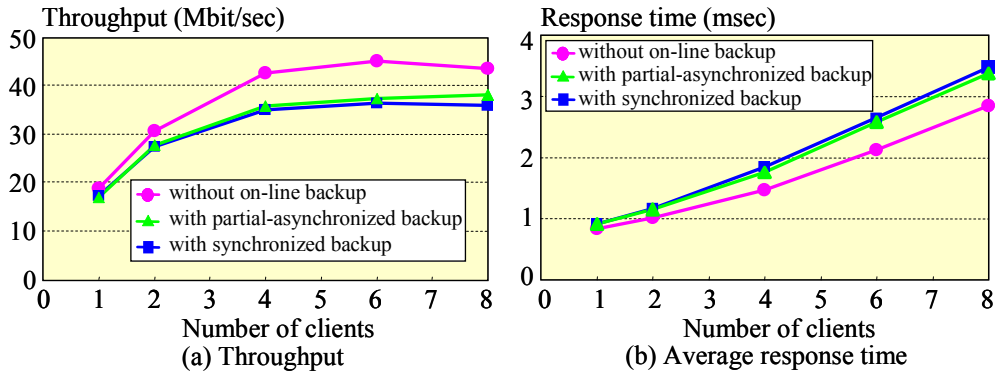

Figure 3. Throughput and average response time of X-NAS with or without on-line backup function in the case of NetBench.

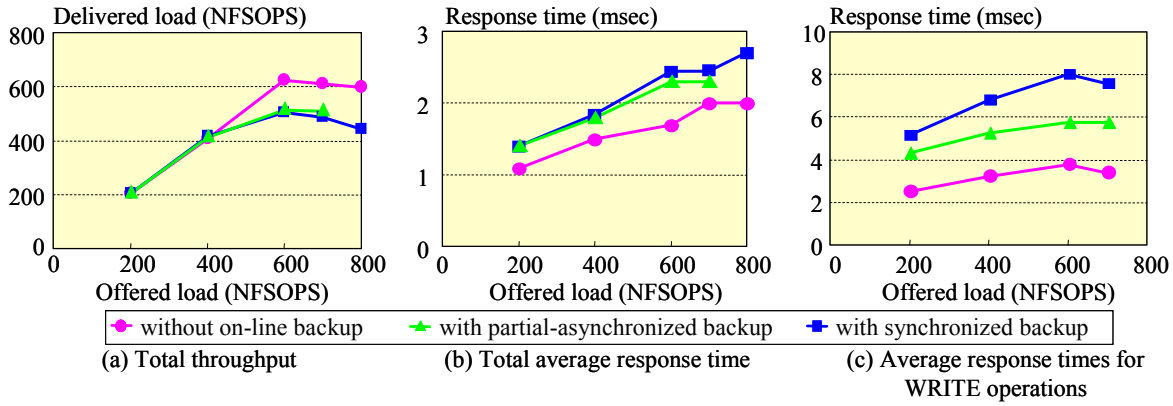

Figure 4. Throughput and average response time of X-NAS with or without on-line backup function in the case of SPECsfs.

3.3. Discussion

To specify the reason for the longer response time in the case of SPECsfs, the average

response time of each NFS operation in the case of the synchronized backup was analyzed. The average response times for some write requests such as WRITE, SETATTR and CREATE are longer than those for X-NAS without that function. In particular, the average response time for WRITE operations is 2.5 times higher than that for the other operations. Profiling results of the WRITE operations shows that waiting time for the sub-thread completion is about 24% of the total processing time and access to the data partition via an IP network is about 48% of that time. By applying the partial asynchronized backup to X-NAS, this waiting time can be reduced to almost zero. Figure 4(c) shows the effects of the partial asynchronized backup in the case of WRITE operations. The average response time for WRITE operations with the partial asynchronized backup can be reduced to from 2.5 times to 1.8 times the time for X-NAS without the function. As a result, the total average response time for SPECsfs with the function can be reduced to 1.3 times that without it. However since the ratio of data transmission time to the total processing time is still higher in the case of the 100-Megabit Ethernet, using a Gigabit network is effective because it can reduce the data transmission time for 100-Megabit Ethernet to at least one-fifth. Furthermore, by optimizing other operations such as CREATE and COMMIT, the performance objective of 1.2 times can be achieved.

4. Related work

There are several methods for replicating file objects between several NAS systems via the network. DFS [1] is a simple and easy file-replication function on Windows systems. DRBD [5] is a kernel module for building a two-node HA cluster under Linux. Double Take [2] is a third-vendor software to replicate file objects on the master NAS to the slave NAS.

5. Conclusions

An on-line backup function for X-NAS, a clustered NAS system, has been developed. On-line replication, the core of the on-line backup function, replicates file objects on X-NAS to a remote backup NAS in real-time for each NFS operation. A multi-threaded wrapper daemon with a low overhead, the developed file-handle cache and the partial asynchronized backup method can reduce the overhead for accessing the backup NAS. An X-NAS prototype with the on-line backup function, based on NFSv3 running the NetBench and SPECsfs97 programs attains 80% of the performance of X-NAS without the function. This function improves the dependability of entry-level NAS while maintaining its manageability.

References

[1] Deploying Windows Powered NAS Using Dfs with or without Active Directory. http://www.microsoft.com, 2001.

[2] Double-Take Theory of Operations. http://www.nsisoftware.com, 2001.

[3] NetBench 7.0.3. http://www.etestomglabs.com/benchmarks/netbench, 2002.

[4] SFS3.0 Documentation Version 1.0. http://www.spec.org, 2002.

[5] P. Reisner. DRBD. In Proceedings of the 7th International Linux Kongress, 2000.

[6] Y. Yasuda et al. Concept and Evaluation of X-NAS: a highly scalable NAS system. In Proceedings of the 20th IEEE/11th NASA MSST2003.