Large-Scale File Service based on DCE/DFS in HEP

S. Yashiro [a], T. Sasaki [a], Y. Morita [a], T. Ishikawa [a], H. Mawatari [a], Y. Watase [a], N. Takashimizu [b], T. Itoh [c], Y. Kodama [d], Y. Munakata [d], S. Kohno [d], H. Itoh [d]

[a] *KEK, High Energy Accelerator Research Organization 1-1 Oho, Tsukuba, Ibaraki 305 Japan*

[b] *Niigata University, 8050 Ikarashi Ninocho, Niigata, Niigata 950 Japan*

[c] *Software Development Centre, Hitachi Ltd. TYG 11th Bldg., 3-16-1, Nakamachi, Atsugi, Kanagawa 243, Japan*

[d] *Government & Public Corporation Information Systems Division, Hitachi Ltd. Tsukuba Mitsui bldg.,1-6-1, Takezono, Tsukuba, Ibaraki 305, Japan*

The central computer system at KEK was designed and installed with the DCE to manage 10 distributed computer clusters for research groups. The files in the system are accessed through the DCE/DFS server-client mechanism, which provides a caching function for remote files. The HSM software was also implemented to facilitate seamless access to the files in the disk as well as in the tapes. For integrating HSM with DCE/DFS and improving the file-transfer speed, we changed the access sequence in the DCE/DFS and created a new policy of recycling the cache so as to avoid unwanted cache flushing for small-sized files by simultaneous caching of a large-sized one.

Key words: DCE;DFS;HSM

1 Introduction

The central computer system(KEKCC) at KEK was replaced from a main frame to a distributed computing system in January, 1996. This system is shared by many research groups; experimental groups, theorists, accelerator physicists and so on, which have a variety of computing requirements.

A new computing environment is built with the DCE using DCE/DFS from the view points of security and system management. For convenient handling of files, we introduced a Hitachi JP1/OmniStorage(OMNI), which is a hierarchical storage management (HSM) software integrated with DFS. Users can access these files as if they were on disks, while files are automatically migrated between disks and tapes.

In this paper we present the status of the first large-scale DFS service among major HEP laboratories in the world. Moreover, several enhancements for KEK requirements are described.

2 System Design

In 1995, we decided to use the DCE and DFS for KEKCC after long study of several technologies for software which provides a distributed computing system from the view points of security, scalability, user-friendliness and easiness of management, operation and maintenance. Many vendors are offering DCE products which were developed by themselves based on the source code provided by The Open Group; they can offer their own enhancement of DCE and DFS. Also, better support is expected compared with a third-party product(AFS etc.). Due to this reason we chose the DCE and DFS. The design of DCE and DFS promised higher scalability and availability, even for very large-scale distributed computing systems.

In the following, the design philosophy of KEKCC system is described:

(i) Multiple DCE cells
This system is shared by many research groups. To enable each research group to run their cluster system with their policy, the hardware is configured to be independent among the cluster systems. We made one DCE cell for a research group. KEKCC thus comprises 10 DCE cells. The user account and home directory are managed separately in each cluster system.

(ii) Inter-cell communication
Inter-cell communication is used to access the other cluster system. Currently, it is mainly used for global system monitoring.

(iii) Seamlessness
We integrated the UNIX login and DCE login, e.g., xdm and ftp so that users can seamlessly migrate from the legacy systems to the DCE. Actually, users meet the DCE user interface only in rare cases, such as when changing the password or shell, specifying a full file path, and so on.

(iv) Integrating HSM and DCE/DFS
All files, including home directories and data files, are accessed through DFS. File systems in HSM have to be accessed through DFS so that the transparent migration is available from any client.

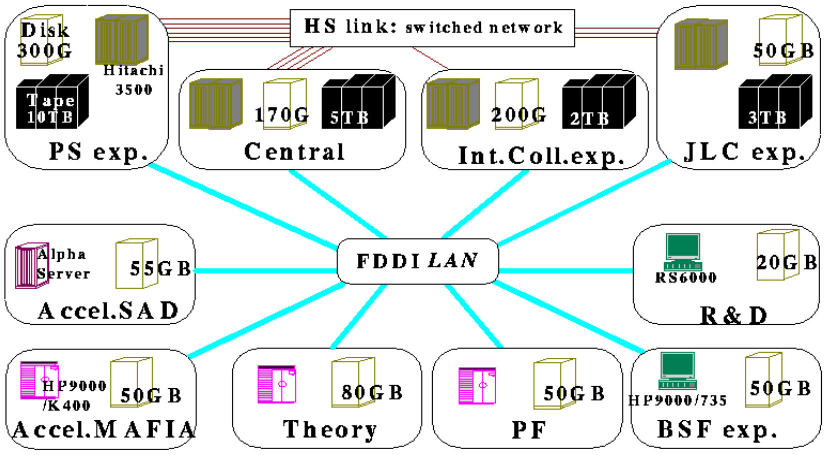

Fig. 1. KEK Distributed Computer System Configuration

The configuration of the CPU server and data server of each research group is shown in Figure 1 and Table 1. The Hitachi 3500 is a SMP server machine equipped with a 120-MHz PA-7200. The operating system, HI-UX, was developed by Hitachi Ltd. All Hitachi 3500s are connected to each other with a Hitachi crossbar switch network:HS-Link, which accomplishes a 20MB/sec transmission rate at the physical level.

Experimental groups need a huge storage capacity to store their data on it. To answer this, the system is equipped with large-scale mass-storage systems, a SONY PetaSite tape system [1], which has 20TB capacity, and a 1TB RAID system.

Cluster name	Server Machine		Disk	Tape Library	
Central	4 $\times$ Hitachi 3500	16 CPUs	170 GB	5 TB	3 drives
PS exp.	6 $\times$ Hitachi 3500	22 CPUs	300 GB	10 TB	12 drives
JLC exp.	3 $\times$ Hitachi 3500	10 CPUs	200 GB	2 TB	2 drives
Int. Coll. exp.	1 $\times$ Hitachi 3500	4 CPUs	50 GB	3 TB	3 drives
Total	25 hosts	75 CPUs	1 TB	20 TB	20 drives

Table 1. Configuration of major clusters

Figure 2 shows the hardware configuration of the central service. Simultaneously it makes a DCE cell.

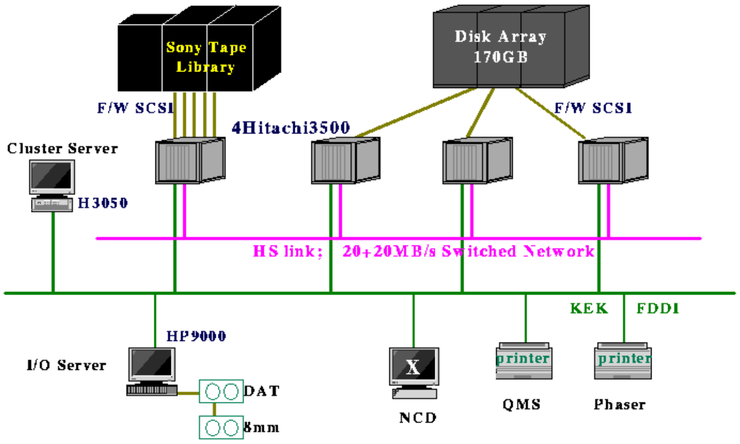

Fig. 2. Central System Configuration

3 Implementation of DCE and HSM

In this section we present the implementation of HSM integrated in DFS and several enhancements to obtain a higher performance.

In October, 1996, Hitachi's DCE was upgraded to DCE 1.2, where the DFS performance is remarkably improved.

3.1 Implementation of HSM under DFS

Using the provided DFS form The Open Group, the RPC time out occurs because the handling, mounting and loading of a tape takes several minutes in the worst case. In the specification of the DFS, the client disconnects the RPC connection to the server unless the client receives a response within a finite time (4.5 minutes).

To avoid this RPC time-out, we added the following mechanism on both Hitachi's DCE and the HSM, OMNI. For example, we will explain the control flow for the 'READ' action (See Fig. 3). The 'WRITE' action is similar to it.

(i) Application requests the 'READ' action to the DFS server through the DFS client.
(ii) If the file exists on disks, data transmission starts immediately, as usual.
(iii) If not, the DFS server sends a request for migration to the OMNI client.

Then, the OMNI client sends a asynchronous request of migration-to-disk to the OMNI server, and sends return code with `E_NODISC` to the DFS client through the DFS server.

(iv) The DFS client repeatedly trys to send a request until it receives the normal return code.

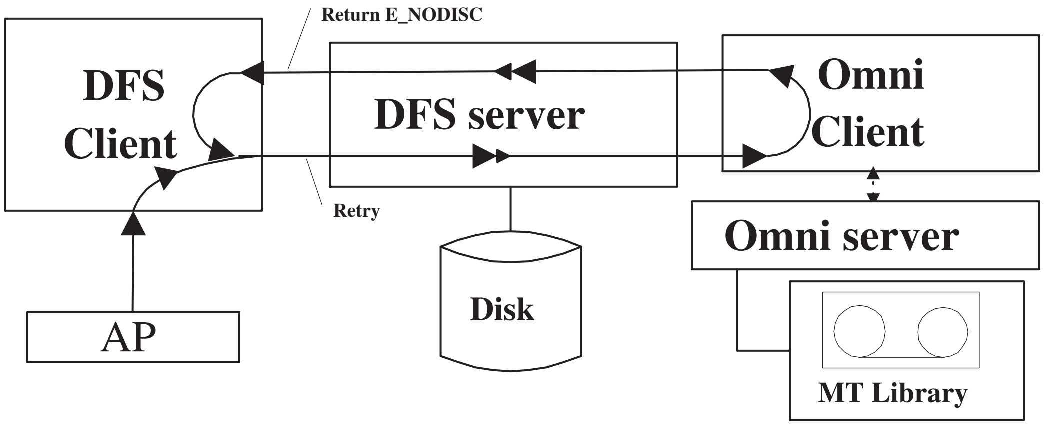

Fig. 3. Mechanism of DFS with OMNI

3.2 Improvement on the DFS cache algorithm

We also made enhancements on the DFS client's cache algorithm in order to avoid slashing on the cache.

In the original cache algorithm of DCE, small-sized files tend to be flushed when a large-sized file(data file) is accessed. In general the small files, i.e., user programs and so on, must take advantage of the cache. To improve the performance of a system with a cache mechanism, we introduced a new function that the DFS client limits the available cache size for a large-sized file. Then, the DFS client keeps the small files from being flushed out.

3.3 DFS performance of DCE 1.2

We measured the performance of DFS data transmission under several conditions. Table 3. shows the performances of a 90MB file transfer using HS-Link with memory and disk cache, changing the size of DFS cache. In this case, we set 256 KB as the chunk size for best performance.

We found that the memory cache is more effective, as expected. There is no advantage for having a large-sized cache. In general, in the case of a large-sized cache, the DFS server controls too many files in the cache, so that the

Action	WRITE		READ	
DFS cache	20 MB	100 MB	20 MB	100 MB
Memory	3065	2924	2398	2604
Disk	1650	1041	1501	1141

Table 3. The measured performance of DFS data transmission (KB/sec).

load of the system becomes heavy. After elaborate tunings of various DFS parameters, we chose a 20MB memory cache empirically.

4 Summary

The distributed computing system of KEK has been operational since January, 1996, with the HSM under the DCE/DFS. After installing the latest version of Hitachi's DCE based on DCE 1.2, including the enhancements, the robustness of the system was highly improved. The whole system has been successful in operation in spite of the heavy load.

For the batch function, we are planning to introduce the LSF(Load Sharing Facility) under DCE.

In KEK, DCE is also introduced in the large computer system for the KEK B factory, which comprises a Fujitsu AP3000, a 4TB RAID and a 150TB tape library.

We will continue to improve the performance of the large-scale file service based on the DCE/DFS and extend the service to PC clients.

References

[1] H.Fujii, et al, *Proc. on CHEP'95*, 36-39, 1995.
Y.Morita .et.al, *Proc. on CHEP'97*, 321-323, 1997.