［Research Note］

Development of Server Function Recovery System for Peer-to-Peer Method Server Management System adopted for Virtual Server System

Mitsuyoshi KITAMURA†, Koki TANI‡, Toshikazu TAKESHITA† and Takumi YAMAGUCHI†

† Graduate School of Engineering, Tokyo Polytechnic University
‡ Quest Co., Ltd.

Abstract

A problem with peer-to-peer (P2P) server management systems is that they are highly dependent on file servers when performing server management functions. Herein, this problem is examined by applying a P2P method server management system to a virtual server system. To create a P2P method server management system that does not require a file server, we adopt a synchronous editing method to share management files and propose an automatic replication original server method as a backup server startup method to recover functions in the event of server failure. To validate our proposed method, we constructed an experimental virtual server system capable of operating both the proposed and conventional file server-based methods and then conducted experiments that confirm the superiority of our proposed method.

1. Introduction

The 5th generation mobile communication system (5G) [1] is now underway and has already achieved high-speed and large-capacity networks providing low-delay communications. In addition, telecommunications industries around the world are looking ahead to the development of the 6th generation mobile communication system (6G) that will make it possible for us to use even more convenient services via the Internet in the future. Within the context of both systems, the role of the servers that provide services is crucial.

Therefore, various important studies [2]-[5] on reliability improvement focusing on data centers have been conducted. Because the central unit for providing data center services is the server, numerous studies on high-availability server systems have been performed. These include the development of a multiple server backup system that can back up the server functions of multiple real servers only with one real server [6], high-availability distributed clusters that can dynamically change configurations while maintaining services based on scalability and traffic rates [7], and high-availability server clusters with greatly improved service reliability and availability, which were achieved by the adoption of fencing technology [8].

Generally speaking, because server systems are monitored by specific management servers, countermeasures for management server failure and management load increases based on the number of managed servers are matters of serious concern. To address this issue, a peer-to-peer (P2P) method server management system that does not require a specific management server has been reported recently [9]. However, a particular problem with the P2P method for server management systems is that the method is highly dependent on file servers when performing server management functions. To resolve this problem, a synchronous editing method (SEM) for file sharing and an original server method that serves as a backup server startup method for recovering failed server functions have been proposed [10]. Unfortunately, because the management target of the previously researched system [10] is a real server, it cannot be applied to the virtual server systems that are generally used in data centers.

In the present study, we propose a P2P method server management system that does not use a file server and can therefore be adopted in a virtual server system. That is, in this proposed system, the SEM can be adopted for sharing management files, whereas the original server method [10], which is the server function recovery method, cannot be applied due to a problem related to the number of original server arrangements.

[Research Note]
Received 1 April 2021, accepted 29 May 2021

To resolve this problem, we implement an automatic replication original server method and show its detailed configuration. Furthermore, we also report on the construction of an experimental virtual server system that can operate with both the proposed and conventional methods using a file server. The results of experiments conducted using this experimental system show that the SEM can be applied and that the proposed backup server startup method is superior to the conventional method.

The remainder of the present paper is organized as follows. Section 2 describes the configuration of a conventional P2P method server management system that uses a file server and its operation in the case of a server failure. In addition, our automatic replication original server method, which is capable of solving the problems that occur when adapting the P2P method server management system to the virtual server system, as well as its configuration and the detailed operations that take place when a real server failure occurs, are shown. Section 3 describes the construction of an experimental virtual server system that can be operated with both the proposed and conventional methods and shows its configuration and specifications. In addition, we show that the SEM can be applied to a virtual server system and that our proposed method improves the performance of the server function recovery operation. Finally, we provide our conclusions in Section 4.

2. Outline of the P2P method server management system and the proposed system

2.1. P2P method server management system that manages a virtual server system

Figure 1 shows the relationship between operations in the case of a server failure and the management file in a conventional P2P method server management system adopted for the virtual server system. In this figure, RS1 through RS3 are real servers and S1 through S6 are virtual servers that act as both management and target servers. S2b denotes a backup server used for recovering the server function if S2 breaks down. File servers with network file system (NFS) connections to the real and virtual servers are indicated by File (NFS).

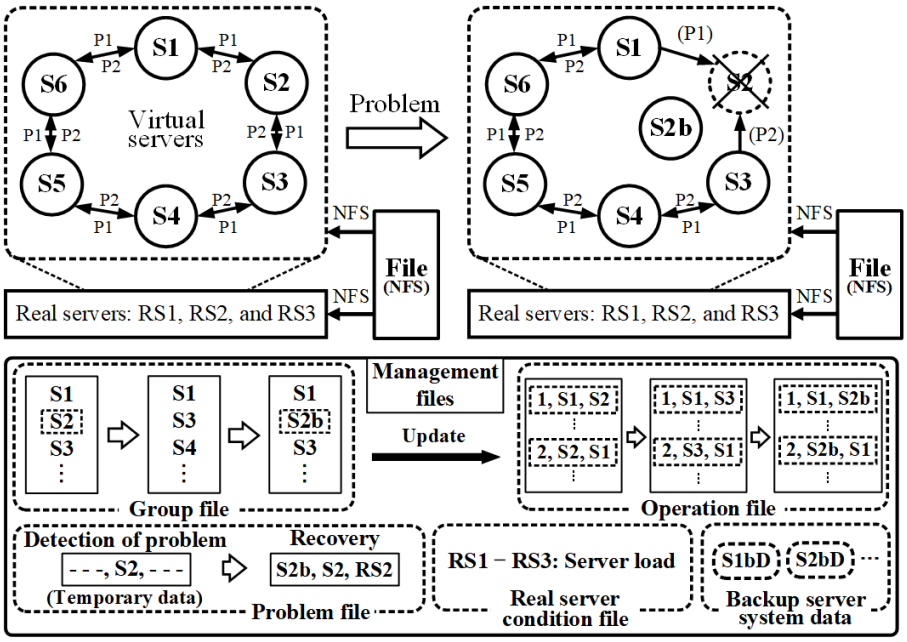


Fig. 1: P2P method server management system configuration and management file.

In the P2P method, because two management servers manage one target server, a server management priority is set for each server. In this figure, S2 is managed by S1 operating with management priority 1 (P1) and by S3 operating with management priority 2 (P2). In a conventional system, management files are centrally managed on the file server, and each management server accesses them through the NFS connection. Each management file includes the group, operation, problem, and real server condition files. Information on all management servers is recorded in the group file. The operation file, which contains the server management priority, the management server hostname, and the target server hostname, is constructed automatically based on the group file. If a problem is detected in the target server, status reports on the recovery operation are recorded in the problem file, and

the load value of each real server is recorded in the real server condition file.

In Fig. 1, we can see that S1 operating with P1 detects a problem in S2 and performs operations to create S2b on RS2, which has the lowest load among the real servers, based on the real server condition file. S1 edits the management file in the file server when a failure is detected and reflects the failure status on the system. Here, temporary data (the problem server name S2) are recorded in the problem file if S1 detects a problem in S2, and the S2 information is deleted from the group file. Then, the operation file is automatically updated based on the group file. S1 edits the temporary data of the problem file at the same time the S2b startup process is completed. After determining that S2 cannot be recovered, S2b data is added to the group file and the operation file is updated again. These operations show that the file server plays an important role in the P2P method server management system.

2.2. Sharing management files with SEM

To mirror the management file contents on all management servers without using a file server, the SEM [10] is adopted. In the SEM, file editing control data (Diff.dat) corresponding to the edited part of the management file is transmitted to each management server as the file-sharing method. As shown in Fig. 1, the management file is deleted, added, and modified. The Diff.dat contents are “filename, Del, deletion information” when deleting information; “filename, Add, addition information” when adding information; and “filename, Mod, information before change, information after change” when rewriting the contents.

The group file sharing method shown below is used if the management server detects that the target server has failed. Because information on the running management server is recorded in the group file, information for the server determined to be faulty is deleted from the list in that file. The management server transmits Diff.dat, which contains “Group, Del, failed server information”, to each management server. Each management server receives it and deletes the information of the failed server from the group file. The management files saved on each management server are shared by a similar procedure. Here, because the SEM is adopted for the virtual server system, it is installed on each virtual server.

2.3. Proposed automatic replication original server method

Figure 2 shows the structures of the conventional system, the original server method [10], and the proposed system that provides the backup server startup method. Figure 2(a) shows the structure of the conventional backup server startup method when it is used in a virtual server system. In the conventional method, the system data of the backup server for recovering the failed virtual server function is saved in the file server. The conventional method also creates the backup server (S1b - S6b) on the real server (RS1- RS3) through the NFS connection with the file server.

Figure 2(b) shows an operational outline of the original server method when recovering the failed S2 function. The original server method consists of the original server that installs all the service programs and the installation procedure file that contains the installation procedure of the configuration file for the service program provided by the failed server. In this figure, the original server is created using the system data for the original server. Next, to use the original server as S2b, “Installation procedure file (S2b)” is transferred and installed. This procedure allows the original server to operate as a backup server to recover the S2 server function. Additionally, because installation procedure files are small, the installation procedure files for all target servers can be stored on all real servers.

The structure of the proposed system applying the original server method (and its operational outline) is shown in Fig. 2(c). In the P2P method that manages the real server system, only one backup server is created on one real server, even if half of the system servers fail [9]. Therefore, two ODs are set on each real server [10]. However, if a virtual server system is targeted, the number of ODs saved on the real server cannot be specified. Therefore, we propose an automatic method for replicating the original server that enables the creation of the required number of ODs from the ODS. In this process, the IP address, hostname, and MAC address are changed after cloning, and the related configuration file is edited as well. The number of ODs to be saved on each real server is the average number of virtual servers that can be created on each real server in the server system. Here, because the average number of virtual servers is two, the number of ODs set for each real server is also two.

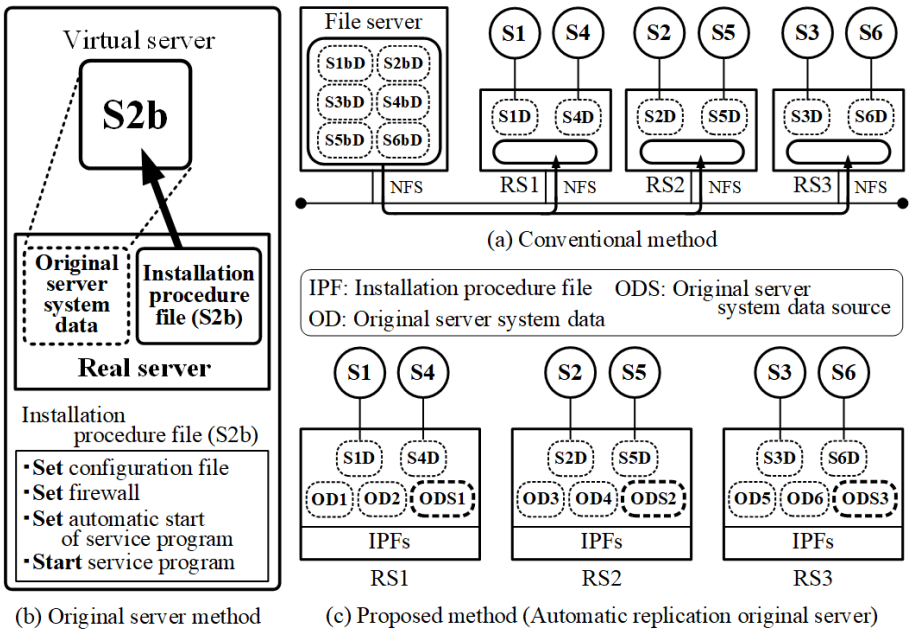
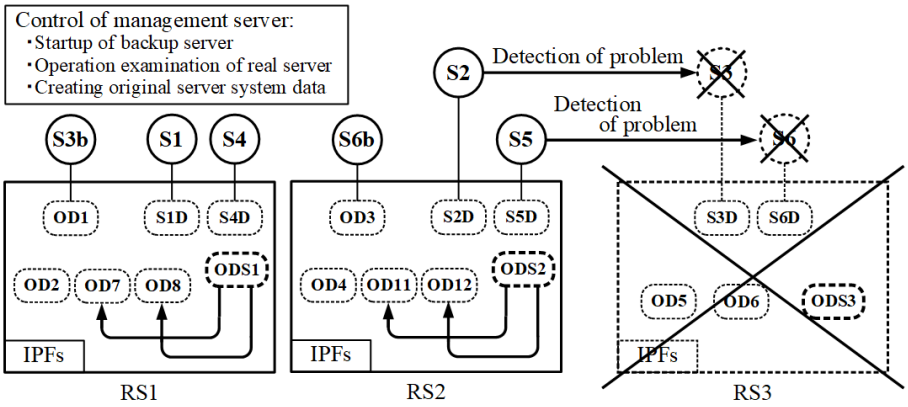


Fig. 2: Structures of conventional, original server, and proposed methods.

Figure 3 shows the operation of the proposed method if a real server fails. In our proposed method, an available original server file and a reservation file, which are saved on each real server, are added as the management file. The available original server file contains the hostname of the original server and its IP address. This file is referred to when the original server is used to recover the function of the failed server. The information of the used original server is deleted from this file. The reservation file contains the hostname of the original server to be duplicated and its IP address. The information used to duplicate the original server is deleted from this file. The number of virtual servers can be obtained from the group file and the number of real servers can be obtained from the real server condition file. From those results, the number of original servers to be automatically duplicated is calculated. The group file and the real server condition file are edited if the management server detects a failure.



Management files in RS1 before RS3 breaks down

Available original server file	Reservation file	Group file	Real server condition file
OD1, OD2	OD7, OD8, OD9, OD10	S1, S2, S3, S4, S5, S6	RS1, RS2, RS3

Management files in RS1 after RS3 breaks down

Available original server file	Reservation file	Group file	Real server condition file
OD2, OD7, OD8	OD9, OD10	S1, S2, S3b, S4, S5, S6b	RS1, RS2

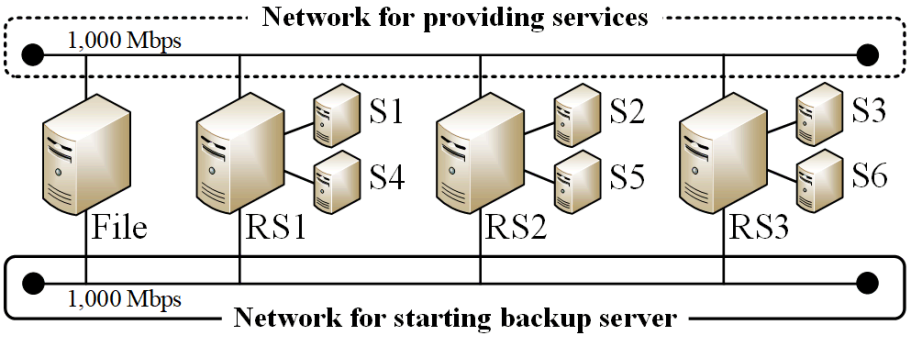
Fig. 3: Operation at the time of real server failure in the proposed method.

In the P2P method server management system, the real server to create the backup server is determined by judging both the load status and the number of created virtual servers in each real server. Here, as shown in the figure, if RS3 fails (one-third of the virtual server in the server system fails), S3b is created on RS1 and S6b is created on RS2. In this state, the average number of virtual servers per real server is three. Therefore, in the proposed method, two available ODs are automatically created from the ODS, and three ODs are prepared. In RS1, the system data of the useable original server is OD2, OD7, and OD8.

3. Measurement of file synchronization time and backup server startup time

We constructed the experimental system shown in Fig. 4 to examine the management file synchronization time required by the SEM as well as the backup server startup time for the conventional and proposed systems. The experimental system consists of three real servers, six virtual servers created on the real servers, and a file server used in the conventional method. It also has a network for providing services to clients and a network for creating a backup server via the file server. In the P2P method server management system, the virtual servers have functions for both the management and target servers. Because the three real servers must create the virtual server, the memory is set at 8,192 MB and a solid-state drive (SSD) is used as the storage medium. One CPU and 2,048 MB of memory are assigned to each virtual server. All of the servers use the CentOS operating system, which is often used for servers, and operate with a character user interface (CUI) to facilitate efficient memory utilization.

In the proposed system, as shown in Fig. 2(c), both the original server system data and the system data for cloning are stored in each real server. In the conventional system, as shown in Fig. 2(a), the backup server system data is saved in the file server. In the virtual server, the startup time in the suspended condition is much shorter than that in the normal condition. For this reason, the original server and the backup server are in a suspended condition in both the proposed and conventional methods.



Specifications	File server	Real server: RS1 – RS3	Virtual server: S1 – S6
CPU	Intel Core i7-2600 3.40 GHz (TB: 3.80 GHz)	Intel Core i7-3770 3.40 GHz (TB: 3.90 GHz)	1
Memory	8,192 MB		2,048 MB
Storage	SSD (SATA 3)		
System software	nfsd	KVM 1.5.3	
OS	CentOS 7.7 (64-bit)		

Fig. 4: Measurement system for file synchronization and backup server startup times and its specifications.

Table 1 shows the management file synchronization time in a case where the SEM is adopted in the virtual server system along with comparison results for the backup server startup time of the proposed and conventional methods. Each measurement time is the average value of ten experimental results. In this experiment, S1, which detected the failure of S2, edits the management file and sends Diff.dat from S3 to S6 and the time until the edited content is reflected and measured in the management file on another management server. The measurement time is extracted from the log file recorded by the SEM. From the S3 to S6 results, we can see that the synchronization time for the system is 3.27×10^{-1} seconds. Here, because S4 is created on

the real server, which is also creating S1, the synchronization time is short. In addition, it can be seen that there are almost no fluctuations in the synchronization time of each management server from each variation coefficient.

In the P2P server management system, two management servers monitor one target server as shown in Fig. 1. Thus, the file synchronization time shown below is critically important in this system. After detecting a server failure, it is necessary to complete the problem file synchronization during the failure reexamination process. This reexamination takes two seconds in the case of a network problem and three seconds in the case of a service fault [9]. Therefore, considering the synchronization time shown in Table 1 we can see that the SEM can be adopted without any problems.

Table 1: File synchronization time in SEM and backup server startup time.

Synchronous editing method (Management server: S1 (RS1), target server: S2 (RS2))

	S3 (RS3)	S4 (RS1)	S5 (RS2)	S6 (RS3)
Synchronization time (s)	3.27×10^{-1}	2.78×10^{-1}	3.23×10^{-1}	3.22×10^{-1}
Variation coefficient	5.31×10^{-2}	3.62×10^{-2}	4.03×10^{-2}	5.63×10^{-2}

Original server method

		Start time of one server (s)	Time to activate one suspended server (s)	Time to activate two suspended servers (s)
Proposed method (Original server method)	Average (Variation coefficient)	1.40×10^{1} (5.24×10^{-3})	2.17×10^{0} (4.12×10^{-2})	2.27×10^{0} (2.61×10^{-2})
Conventional method (Use file server)	Average (Variation coefficient)	2.01×10^{1} (1.76×10^{-2})	4.41×10^{0} (1.52×10^{-3})	7.76×10^{0} (1.01×10^{-2})

Create time of clone server system data (s)	3.56×10^{1} **(Variation coefficient:** 8.42×10^{-3}**)**
Setting time (s): Clone server → Original server	3.08×10^{1} **(Variation coefficient:** 2.44×10^{-2}**)**
Total create time of original server (s)	6.64×10^{1} **(Variation coefficient:** 1.34×10^{-2}**)**

In the P2P method server management system, the backup server startup time is very important for recovering the function of the failed server. The startup time of one backup server, which is the normal state, and the startup time of one backup server along with the simultaneous startup time of two backup servers, which comprise the suspended state, are shown for both the proposed and conventional methods. The coefficient of variation in each case is also shown. The average times in this figure are given by the command "time" in UNIX.

For all the results obtained, it is clear that the startup times for the proposed method are shorter than those for the conventional method. In the simultaneous startup time for two servers, the startup time for the proposed method is almost the same as the startup time of one server, whereas the startup time for the conventional method is significantly increased. This increase is caused by delays resulting from simultaneous NFS access to the file server.

In addition, the processing ((i)-(vi)) shown in the list below, which takes 6.64×10^{1} seconds, is executed to create the original server from the original server source. In (i), the KVM command virt-clone, which takes 3.56×10^{1} seconds, is used to create the clone server. In (ii) to (vi), the setting time to change from the clone server to the original server is 3.08×10^{1} seconds. This time includes the server startup and the restart times needed to reflect the set contents.

(i) Create a clone server from the original server source.
(ii) Startup of the clone server.
(iii) Change the MAC address, IP address, and hostname of the clone server.
(iv) Change the settings for software due to the hostname change.
(v) Restart the clone server (which will be the original server after rebooting).
(vi) Set the original server to the suspend state.

4. Conclusions

It has long been known that the conventional P2P method server management system is highly dependent on the file server when performing server management functions. Therefore, if the file server fails, server management cannot be continued. In this paper, as part of efforts to realize a P2P method server management system that does not require a file server, we adopted an SEM for sharing management files and proposed an automatic replication original server method that can apply an original server method as the backup server startup method. An experimental virtual server system was constructed to test and compare the conventional and proposed methods, and the following results were obtained.

We determined that the SEM could be applied to the virtual server system and that in the backup server startup method, the startup time for the proposed method is shorter than that for the conventional method. We also found that this difference is further widened in the case of a simultaneous startup. From these results, we conclude that the high availability of the server systems that provide Internet services will be improved because the reliability of the server management that adopts the P2P method will be enhanced. Thus, further expansion of Internet services can be expected, thereby presenting greater convenience in our daily lives.

References

[1] The Fifth Generation Mobile Communications Promotion Forum https://5gmf.jp/en/, 2019.7.

[2] C. Kari, S. Chen, A.-M. Sepehr, and V. Pallipuram, "Data Migration in Large Scale Heterogeneous Storage Systems with Nodes to Spare," Proc. IEEE Conf., Vol. 2019, No. ICNC, 2019, pp. 854-858.

[3] T. Roua, A. Safwan, D. Abdulhalim, A. Nadjib, and V. Cyril, "SDN-Based Batch Flow Routing in CamCube Server-Only Data Center Networks," Proc. IEEE Conf., Vol. 2020, No. ICC, 2020, pp. 1-6.

[4] Q. Shan, Z. Qin, Z. Jinbei, S. Yuan, and W. Xinbing, "Performance Analysis on Distributed Storage Systems in Ring Networks," IEEE Trans. on Vehicular Technology, Vol. 69, No. 7, 2020, pp. 7762-7777.

[5] B. Artsiom, Y. Alan, and R. Elyse, "An Interpretable Predictive Model for Early Detection of Hardware Failure," Proc. IEEE Conf., Vol. 2020, No. IRPS, 2020, pp. 1-5.

[6] M. Kitamura, "Configuring a Low-cost, Power-saving Multiple Server Backup System: Experimental Results," IEICE Trans. Commun., Vol. E95-B, No. 1, 2012, pp. 189-197.

[7] T. Komatsu and K. Ueda, "Data Redundancy Dynamic Control Method for High Availability Distributed Clusters," IEICE Technical Report, Vol. 117, No. 459, 2018, pp. 91-96.

[8] M. Ljubojevic, A. Bajic, and D. Mijic, "Implementation of High-Availability Server Cluster by Using Fencing Concept," Proc. IEEE Conf., Vol. 2019, No. INFOTEH, 2019, pp. 1-5.

[9] M. Kitamura, Y. Udagawa, H. Nakagome, and Y. Shimizu, "Development of a Server Management System Incorporating a Peer-to-Peer Method for Constructing a High-availability Server System," JISSJ, Vol. 13, No. 2, 2018, pp. 14-40.

[10] M. Kitamura and K. Tani, "Development of File Management System for a Peer-to-Peer Method Server Management System," JISSJ, Vol. 16, No. 1, 2020, pp. 1-16.

Authors Biography

Mitsuyoshi KITAMURA

He received his B.E. degree from Tokyo Polytechnic University in 1984. In 1990, he became a research assistant at Tokyo Polytechnic University, where he became an assistant professor in 2003. His current interests are the optimum design and construction of server-client systems and the analysis of various characteristics of server systems.

Koki TANI

He received his M.E. degree in 2020 from Tokyo Polytechnic University, where he is currently in system development and management in cloud computing at Quest Co., Ltd.

Toshikazu TAKESHITA

He received his B.E. degree in 2020 from Tokyo Polytechnic University, where he is currently working toward the M.S. degree. He is in charge of implementation of experimental hardware systems and analyses of power-saving data.

Takumi YAMAGUCHI

He received his B.E. degree in 2020 from Tokyo Polytechnic University, where he is currently working toward the M.S. degree. He is engaged in developing software for experimental systems and analyzing performance data.