［Research Note］

Development of a Power-saving, High-availability Server System by Compound Operation of a Multiple-server Backup System and Power-saving Server System

Mitsuyoshi KITAMURA[†], Youta SHIMIZU[‡], and Koki TANI[‡]

† Faculty of Engineering, Tokyo Polytechnic University
‡ Graduate School of Engineering, Tokyo Polytechnic University

Abstract

We propose the compound operation of a multiple-server backup system (MSBS) and a power-saving server system (PSS) that can operate independently configured as a power-saving, high-availability server system. The MSBS is based on multiple dynamic backup server systems (DBSSs), each of which manages only one target server. Usually, if two kinds of management programs are operated at the same time, both programs become complicated to prevent malfunctions. Here, we propose and adopt a system interface (SIF) to solve the complexity of the DBSS in compound operation, which is realized by allowing only the DBSS to request editing of the PSS configuration files. Rather than adding the file access and editing functions to the DBSS, we found that the structure was simpler when we added the SIF for this purpose.

1. Introduction

In our information-oriented society, a popular concept is the Internet of Things (IoT), in which many sensors and devices are connected to the Internet and information obtained from them generates new value [1]. The Internet that supports IoT hardware is becoming the communication infrastructure for socioeconomic activities and plays an important role in social infrastructure [2]. Server systems perform an important role for saving and analyzing data obtained from these IoT devices. Problems in server systems in data centers supporting cloud computing could cause large losses to the cloud provider and its users [3]. Therefore, high availability in the server system is very important. In addition, environmental and cost concerns make power-saving strategies very attractive.

A number of important studies related to failure countermeasures in data centers have been conducted [4]-[6]. The server system is the fundamental unit for providing service in a data center. In constructing the server system, it is necessary to sufficiently consider power saving in addition to high speed and security. Therefore, a number of important studies on power-saving network systems have been conducted [7], [8].

Although approximately half of all information technology devices are network devices and servers, little research has been conducted on power saving in server systems and few have looked at server systems that provide both power saving and high availability. Moreover, management programs that realize them tend to be complicated, and very little research has been done on methods to simplify these programs.

Therefore, in this paper, we propose a compound operation method for a multiple-server backup system (MSBS) [9] and a power-saving server system (PSS) [10] that can operate simultaneously and independently. In this way, a power-saving, high-availability server system can be achieved. In this study, we defined a server system that includes a target server offering services to clients and a management server that monitors and controls the target server. The MSBS is based on multiple dynamic backup server systems (DBSSs), each of which manages only one target server. To realize compound operation, only the DBSS is allowed to request editing of the PSS configuration files here. We propose and introduce a system interface (SIF) to avoid making the program configuring the DBSS more complicated. Actual file access and editing are performed by the SIF with instructions from the DBSS, which allows the DBSS to retain its simple structure.

[Research Note] Received 5 April 2018, revised 6 July 2018, accepted 17 September 2018.

2. Construction of proposed system and characteristics of MSBS and PSS

The MSBS described in [9] managed real servers and realized a high-availability server system that recovers functions of failed servers by starting up a virtual server if trouble in a target server is detected. Because the virtual server is used only until the recovery processing of the failed real server is completed, the MSBS did not manage the virtual server. The following changes were made to that MSBS for this study. Because the target server is a virtual server, the MSBS here creates and manages the virtual backup server for recovering the function of a failed server. Moreover, because simultaneous failure of multiple target servers is expected in the virtual server system, to reduce the load on the management server, the backup server created by the management server moves to the real server by live migration.

The PSS described in [10] realized power saving by measuring two values, CPU processing rate and total received and transmitted bytes on the network, as the load on a service group composed of virtual servers providing the same service. Then, it classifies the server configuration as one of three types: light, moderate, and heavy. For the light type, the PSS controls to operate only with a specific real server. At that time, the PSS starts a backup server (spare virtual server) on the real server to provide the missing service. Then, the PSS controls the load balancer to change the access from clients and sets the unnecessary real server to the suspended state. For the heavy type, the PSS starts the backup server, controls the load balancer, and distributes the access from clients. The following changes were made to the PSS for this study. The service group load is measured with a third value, memory utilization rate, in addition to CPU processing rate and received and transmitted bytes on the network. Also, the server configuration in the power-saving state is realized by establishing a virtual IP address and moving servers by live migration.

2.1. Configuration of proposed system

The proposed system is based on a load balancer, which is generally used to construct high-availability, distributed-load server systems. A schematic of the proposed system is shown in Fig. 1.

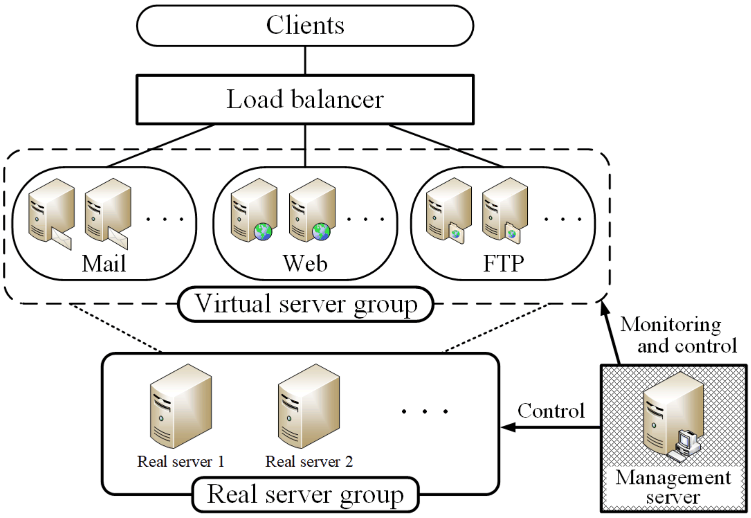


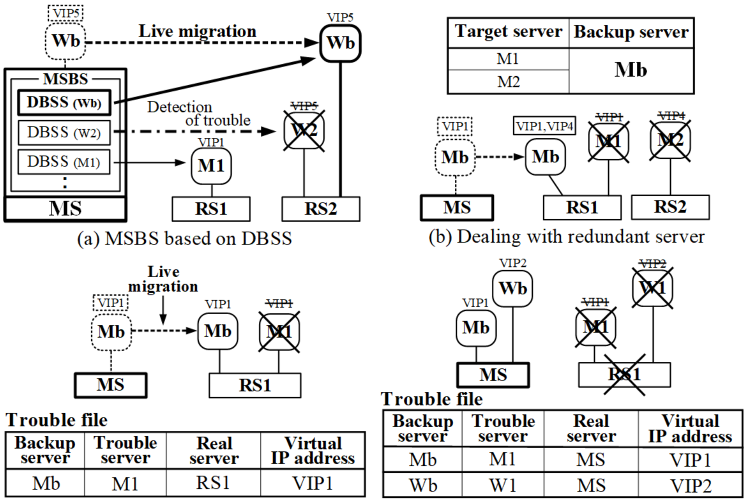
Fig. 1 Configuration of proposed system.

The real server group creates virtual servers, and the virtual server group offers services to clients. Here, virtual servers are distributed redundantly to each service group, and clients access the virtual servers via the round-robin method. The MSBS and PSS are installed on the management server, which monitors and controls the virtual server group and controls the real server group.

2.2. Operation of MSBS

A schematic of the MSBS is shown in Fig. 2. In the figure, MS is the management server; M1, M2, W1, and W2 are virtual target servers; Mb and Wb are backup servers to recover the function of a troubled server; RS1 and RS2 are real servers for establishing virtual servers; and VIP is the virtual IP address. The DBSS that manages target server W2 is denoted as DBSS (W2). As shown in the figure, the MSBS is realized by multiple DBSSs. Each DBSS monitors the network and the state of service on its target server, recovers the server function by setting the VIP address after starting the backup server, and recovers the target server by restarting the problem service program or target server.

In Fig. 2(a), DBSS (W2) detects trouble in target server W2, starts backup server Wb on the management server, and then sets address VIP5. As a result, the function of W2 is recovered. Thereafter, Wb is moved to RS2 by live migration to reduce the load on the management server. In addition, DBSS (W2) automatically starts DBSS (Wb) to manage the Wb and then is terminated. Basically, the DBSS has one backup server to manage one target server. Here, considering the management of the virtual server system, there is a concern that the number of backup servers becomes very large. Therefore, as shown in Fig. 2(b), in the management of redundant virtual servers, this problem is solved by using only one backup server for each service group. As shown in Fig. 2(c), if a virtual server fails, the DBSS moves the backup server started on the management server to the real server that created the troubled server. In the case where the real server that created the virtual server fails, the DBSS does not perform server movement by live migration, as shown in Fig. 2(d). The trouble file shown in Figs. 2(c) and 2(d) is configured to record trouble states in each DBSS.



Target server	Backup server
M1	Mb
M2	

(a) MSBS based on DBSS

(b) Dealing with redundant server

Trouble file

Backup server	Trouble server	Real server	Virtual IP address
Mb	M1	RS1	VIP1

(c) Virtual server trouble

Trouble file

Backup server	Trouble server	Real server	Virtual IP address
Mb	M1	MS	VIP1
Wb	W1	MS	VIP2

(d) Real server trouble

Fig. 2 Trouble management with MSBS.

The DBSS is executed using the management data as an argument. The data necessary for management are provided as eight lines for each target server, as shown in Fig. 3. These data are in a read-only file. The first line contains the server's VIP. Target server information is recorded in the second to fifth lines. Because one server offers multiple services to clients, the names of multiple service daemons, which are the programs that provide services, can be recorded in the fifth line using space separation, as in "dovecot postfix." Information for the backup server, which is used to recover the target server function if trouble occurs, is recorded in the sixth and seventh lines. The information for the real server that created the virtual server is recorded in the eighth line. Because the MSBS also manages the backup server, management data for the backup server are also recorded as Mb.dat. Fewer lines are needed because the VIP address (line 1) and the IP address of the real server (line 8) are set according to the situation. Because a backup server is not used when Mb fails, lines 6 and 7 are also empty.

If the target server host name is M1, the DBSS on the management server is executed using M1.dat as an argument. If a problem is detected in M1, Mb, the backup server that duplicates the function of M1, is started on the management server. Then, the DBSS assigns VIP address 172.21.14.101 to Mb. As a result, the functionality of M1 is recovered because the backup server provides access and services to clients. The DBSS does not need to be modified to assume the identity of the target server, because it uses the target server management data as an argument.

Management data	M1.dat	Mb.dat
1st line: Virtual IP address (VIP)	172.21.14.101	—
2nd line: Host name	M1	Mb
3rd line: IP address (IPV)	172.21.14.1	172.21.14.51
4th line: Service group name	Mail	Mail
5th line: Service daemon name	dovecot postfix	dovecot postfix
6th line: Backup server host name	Mb	—
7th line: Backup server IP address (IPB)	172.21.14.51	—
8th line: Real server IP address	172.21.14.201	—

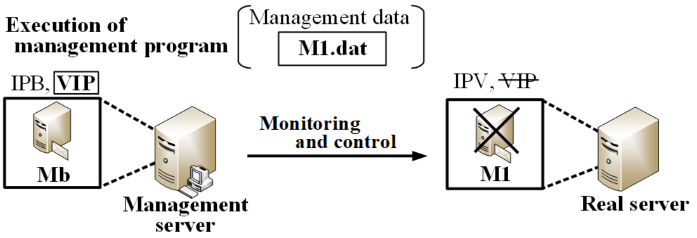


Fig. 3 Details of management data.

2.3. Operation flow and details of PSS

Figure 4 shows the operational flow of the PSS, where M1, M2, W1, W2, F1, and F2 are the virtual servers to be controlled; RS1, RS2, and RS3 are the real servers that create them; and LB denotes the load balancer. The virtual servers belonging to each service group are configured redundantly. M1 and M2 are a service group for mail, W1 and W2 belong to a service group for web, and F1 and F2 belong to a service group for FTP. The PSS measures the loads (memory usage rate, CPU processing rate, and transmission and reception rate in the network) of the virtual servers belonging to each service group and classifies them as a moderate load class or a light load class. The server configuration is changed to the normal (moderate) state or the power-saving (light) state according to the load class.

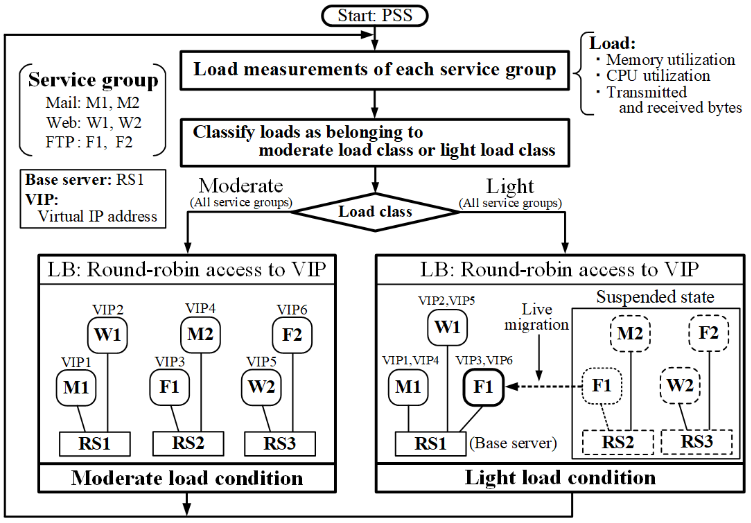


Fig. 4 Configuration of PSS.

In the example shown in the figure, all service groups have the same load class. The server system is started in a moderate load condition, and clients have round-robin access to VIPs of redundant servers by the load balancer. When the PSS determines that the load is light, because RS1 is the base server, the VIP addresses of M2 and W2 are respectively changed to those for M1 and W1, the VIP address of F2 is changed to that of F1, and F1 is moved from RS2 to RS1 by live migration. Then, RS2 and RS3 are placed in a suspended state (Suspend-to-RAM). Round-robin access to each VIP by the client through the load balancer continues, but only one real server is active. This server configuration is regarded as a light load condition. The power consumption under the light load condition is approximately one-third of the moderate load consumption.

Figure 5 shows the method for calculating and classifying the load and judging the need for changing the server system configuration according to the load. In "Calculation of load" in Fig. 5, the web service group is used as an example for calculating the load. Measurement of the memory load is performed in each target server using the command "free" in UNIX. The memory load in the service group is defined as follows:

$$Lm\ [\%] = \frac{M_{\mathrm{use\_w1}} + M_{\mathrm{use\_w2}}}{M_{\mathrm{max\_w1}} + M_{\mathrm{max\_w2}}} \times 100, \tag{1}$$

where M_{use} is memory utilization and M_{max} is maximum memory size. Measurement of the CPU load is performed in each target server using the command "sar" in UNIX. The CPU load in the service group is given by

$$Lc\ [\%] = \frac{C_{\mathrm{use\_w1}} \times C_{\mathrm{cor\_w1}} + C_{\mathrm{use\_w2}} \times C_{\mathrm{cor\_w2}}}{C_{\mathrm{cor\_w1}} + C_{\mathrm{cor\_w2}}}, \tag{2}$$

where C_{use} is the CPU utilization and C_{cor} is the number of CPU cores. Measurement of the network load is performed using the command "sar" in UNIX. The network load in the service group is calculated by

$$Ln\ [\%] = \frac{N_{\mathrm{trb\_w1}} + N_{\mathrm{trb\_w2}}}{N_{\mathrm{rat\_w1}} + N_{\mathrm{rat\_w2}}} \times 100, \tag{3}$$

where N_{trb} is the total bytes transferred (transmitted and received) on the network and N_{rat} is the total number of bytes transferred per second for the transmission and reception rating of the network. Here, if a router with 1,000 Mbps is taken as an example, because 128×10^3 KB can be transferred in 1 s, N_{rat} is 256×10^3 KB from the total value of transmission and reception.

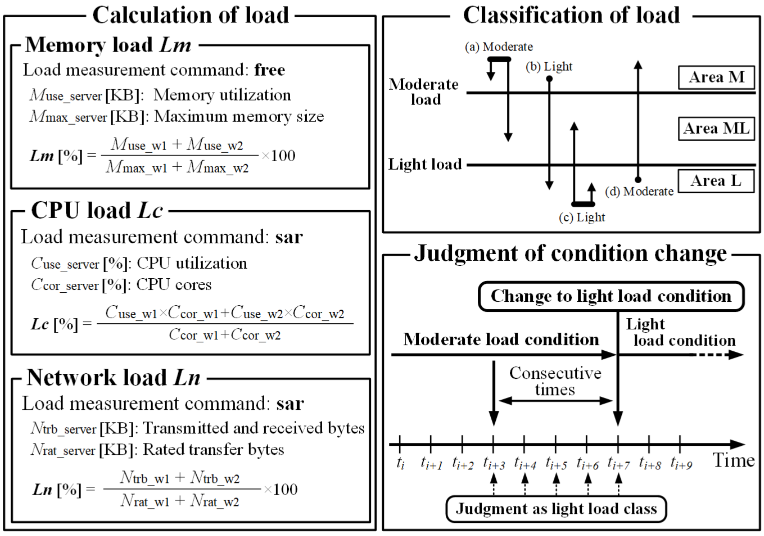
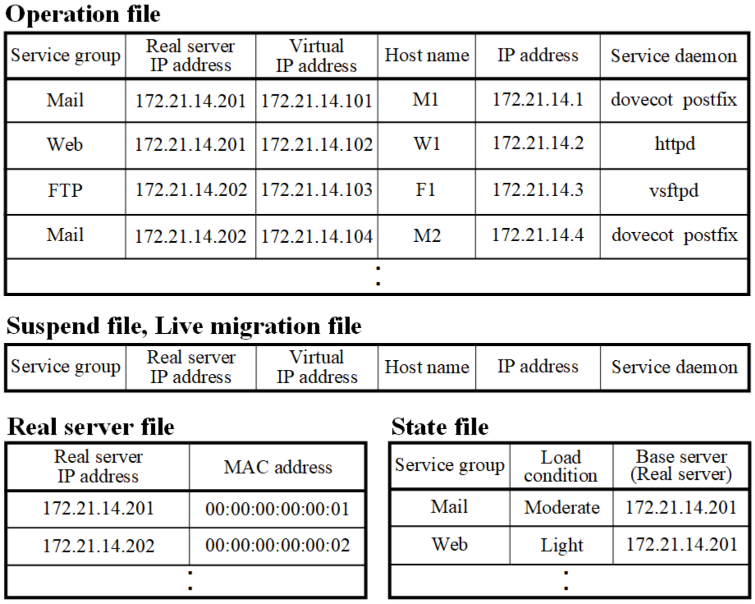

Calculation of load
Memory load Lm
Load measurement command: free
Muse_server [KB]: Memory utilization
Mmax_server [KB]: Maximum memory size
Lm [%] = (Muse_w1 + Muse_w2) / (Mmax_w1 + Mmax_w2) ×100
CPU load Lc
Load measurement command: sar
Cuse_server [%]: CPU utilization
Ccor_server [%]: CPU cores
Lc [%] = (Cuse_w1×Ccor_w1+Cuse_w2×Ccor_w2) / (Ccor_w1+Ccor_w2)
Network load Ln
Load measurement command: sar
Ntrb_server [KB]: Transmitted and received bytes
Nrat_server [KB]: Rated transfer bytes
Ln [%] = (Ntrb_w1 + Ntrb_w2) / (Nrat_w1 + Nrat_w2) ×100
Classification of load
(a) Moderate
(b) Light
Moderate load
Area M
Area ML
Light load
(d) Moderate
Area L
(c) Light
Judgment of condition change
Change to light load condition
Moderate load condition
Light load condition
Consecutive times
ti ti+1 ti+2 ti+3 ti+4 ti+5 ti+6 ti+7 ti+8 ti+9
Time
Judgment as light load class

Fig. 5 Judgment of need for configuration change of server system according to load.

In "Classification of load," the PSS performs classification according to the values of Lm, Lc, and Ln for each service group and classifies the load as light or moderate according to two thresholds. Black dots indicate the initial state, and the arrowhead indicates the final state. If the load state shifts to area ML, as is the case for loads (a) and (c), the load is classified as belonging to the initial state. This is done to reduce the influence of fluctuations near the threshold. Loads (a) and (d) are moderate, and loads (b) and (c) are light. If any load class of the three loads is moderate, the load class of the service group becomes moderate; thus, the load class of the service group becomes light when all service groups are classified as light. In "Judgment of condition change" in Fig. 5, judgments for three service groups are performed. As shown in Fig. 4, the PSS changes the server system configuration according to the load class, which varies. Because the load state in the server system varies, the configuration change considering the state is desirable. As shown in the figure, the configuration of the server system is changed from the moderate load condition to the light load condition only after a light load class was detected four times in a row.

The formats of the configuration files for the PSS are given in Table 1. These configuration files are automatically edited by the PSS. To avoid malfunction due to the competitive editing, these configuration files adopt a locking function that refuses access from other programs at the time of editing. Detailed information for the virtual servers is recorded in the operation file. Here, we define the virtual server that is not providing services to clients as being in a suspended state. When shifting from the moderate load condition to the light load condition in the service group, to provide services to clients by only one virtual server in the service group, the PSS sets all virtual IP addresses for the service group to the virtual server. At the same time, the PSS moves the information of the virtual server in the suspended state from the operation file to the suspend file. If the virtual server returns to the moderate load condition, the virtual server information is moved from the suspend file back to the operation file. Moreover, if live migration is used to change to the light load condition, the information for the target virtual server is copied to the live migration file. Thereafter, the IP address of the real server in the operation file is changed to that of the real server that creates the virtual server moved by the live migration. If the PSS changes the server configuration from the light load condition to the moderate load condition and returns the virtual server by live migration, the contents of the live migration file are returned to the operation file. Information on the real server creating the virtual server is recorded in the real server file. Here, the MAC address in the real server file is used for re-activating a real server in the suspended state. The state of each service group, moderate or light, is recorded in the state file.

Table 1 Configuration files of PSS.



Operation file

Service group	Real server IP address	Virtual IP address	Host name	IP address	Service daemon
Mail	172.21.14.201	172.21.14.101	M1	172.21.14.1	dovecot postfix
Web	172.21.14.201	172.21.14.102	W1	172.21.14.2	httpd
FTP	172.21.14.202	172.21.14.103	F1	172.21.14.3	vsftpd
Mail	172.21.14.202	172.21.14.104	M2	172.21.14.4	dovecot postfix
⋮					

Suspend file, Live migration file

Service group	Real server IP address	Virtual IP address	Host name	IP address	Service daemon

Real server file

Real server IP address	MAC address
172.21.14.201	00:00:00:00:00:01
172.21.14.202	00:00:00:00:00:02
⋮	

State file

Service group	Load condition	Base server (Real server)
Mail	Moderate	172.21.14.201
Web	Light	172.21.14.201
⋮		

3. Problems and solutions of compound operation and characteristics of SIF

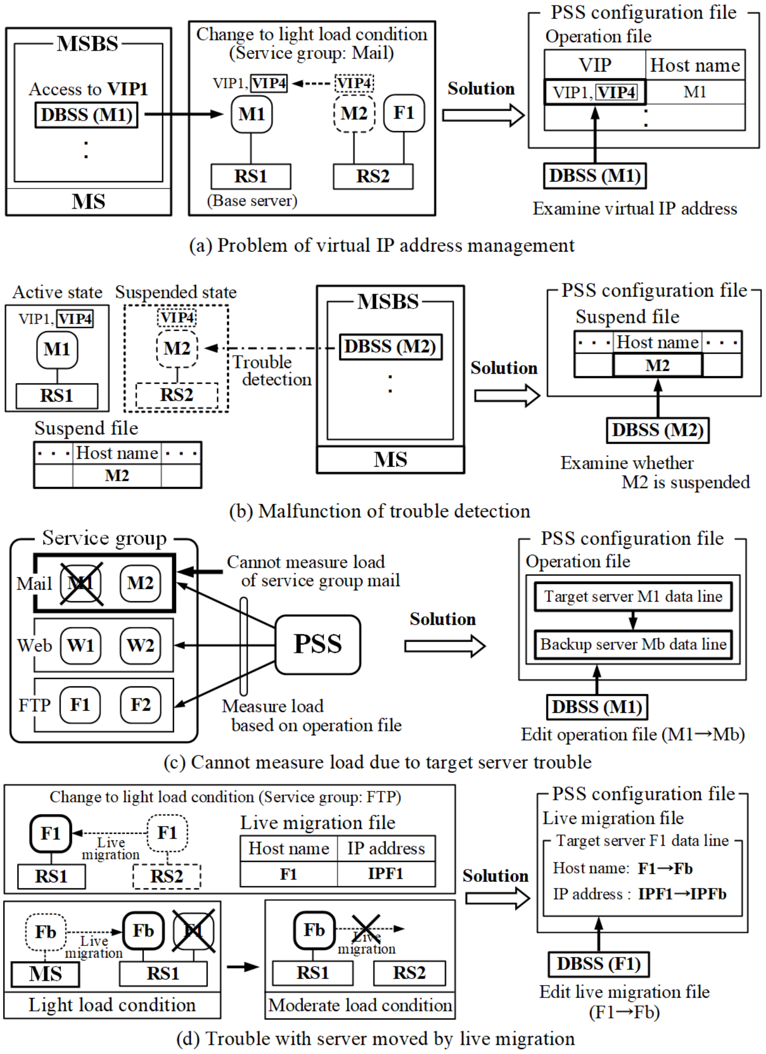
In general, if two kinds of management programs are operated at the same time, both programs become complex to prevent malfunctions. Here, compound operation is realized by allowing only the DBSSs constituting the MSBS to initiate editing of the configuration files for the PSS. In compound operation, the timing of monitoring of the target server by the DBSS and the timing of configuration change by the PSS are processed asynchronously.

3.1. Problems of compound operation and their solution

Figure 6 shows the typical problems and solutions of compound operation of the MSBS and PSS. The notation is the same as that used in previous figures. Figure 6(a) shows the problem and the solution related to management of the VIP address when the PSS configures the system for light load operation. Here, the DBSS monitors M1 and examines access to VIP1. When the mail service group is changed to the light load condition by the PSS, VIP4 set for M2 is changed to the VIP address for M1. The DBSS managing M1 cannot confirm the resetting of VIP4. As a countermeasure, it is necessary for the DBSS to know the data for M1 recorded in the PSS operation file.

Figure 6(b) shows the problem and the solution related to malfunction of trouble detection by the DBSS when the PSS configures the system for light load operation. The DBSS is monitoring M2. At light load, VIP4 set for M2 is changed to the VIP address for M1 because M2 is set to the suspended state. Then, RS2 is suspended because RS2 is creating only M2. As a result, the DBSS may detect network trouble in M2. As a countermeasure, it is necessary for the DBSS to know the data for M2 recorded in the PSS suspend file.

Figure 6(c) shows the problem and the solution of not being able to measure the load of a service group to which a troubled server belongs if the virtual server providing services to clients fails. The PSS measures the load of the virtual server recorded in the operation file and totals the loads of the virtual servers belonging to each service group. Based on the result, the PSS implements the appropriate server configuration. Here, if M1 belonging to the mail service group fails, the PSS cannot measure the load of that group. As a countermeasure, after the DBSS detects the failure of M1, the DBSS creates backup server Mb and recovers the server function of M1. Thereafter, it is necessary for the data line for M1 to be deleted from the operation file and a data line for Mb to be added. Then the PSS can measure the load of the mail service group. In addition, after M1 recovers, the data line for Mb needs to be deleted from the operation file and a data line for M1 added.



(a) Problem of virtual IP address management

(b) Malfunction of trouble detection

Service group
Mail
M1
M2
Web
W1
W2
FTP
F1
F2
Cannot measure load of service group mail
PSS
Measure load based on operation file
Solution
PSS configuration file
Operation file
Target server M1 data line
Backup server Mb data line
DBSS (M1)
Edit operation file (M1→Mb)

(c) Cannot measure load due to target server trouble

(d) Trouble with server moved by live migration

Fig. 6 Problems and solutions in compound operation of MSBS and PSS.

Figure 6(d) shows the problem and the solution related to a server moving from another real server by live migration. If the FTP service group is changed to the light load condition by the PSS, F1 is migrated from RS2 to RS1, and information for F1 is recorded in the PSS live migration file. Moreover, in the operation file, the IP address of the real server in the data line for F1 is changed from RS2 to RS1. If F1 fails in this state, Fb is created on the management server and moved to RS1 by live migration. Here, if the PSS changes the server configuration from the light load condition to the moderate load condition, Fb cannot move onto RS2 by live migration because information for Fb is not recorded in the live migration file. As a countermeasure, when the server recovery function is implemented by starting a backup server, it is necessary for the host name and IP address of the failed server to change to those of the backup server in the live migration file.

3.2. SIF functions and its features

To realize compound operation with the DBSS and PSS, processing for controlling the PSS is coordinated with operation of the DBSS. Figure 7 is an operation flowchart of the DBSS that shows the timing of the needed PSS control actions. The DBSS manages only one target server and has a very simple structure. The operation flow of the recovery process in the DBSS is as follows. If the DBSS detects a failure in the target server, the DBSS performs recovery processing for the target server as a background job after reexamining the failure, starts the backup server, and recovers the function of the target server. In the background job, two types of recovery processing are performed. In the case of service trouble, the target server is restarted. In the case of a network trouble, the DBSS determines whether the target server is in the restart state, and if it is not, attempts a forced restart. Thereafter, in accordance with the result of the recovery process of the target server by the background job, the operation of the DBSS returns to the monitoring state for the target server or terminates.

In Fig. 7, dotted arrows (A1 to A5) indicate the timing of accessing or editing the PSS configuration file by the DBSS. Because the timing of monitoring of the target server by the DBSS and the timing of configuration change by the PSS are processed asynchronously, the DBSS recognizes the configuration change at the timing of A1 or A2.

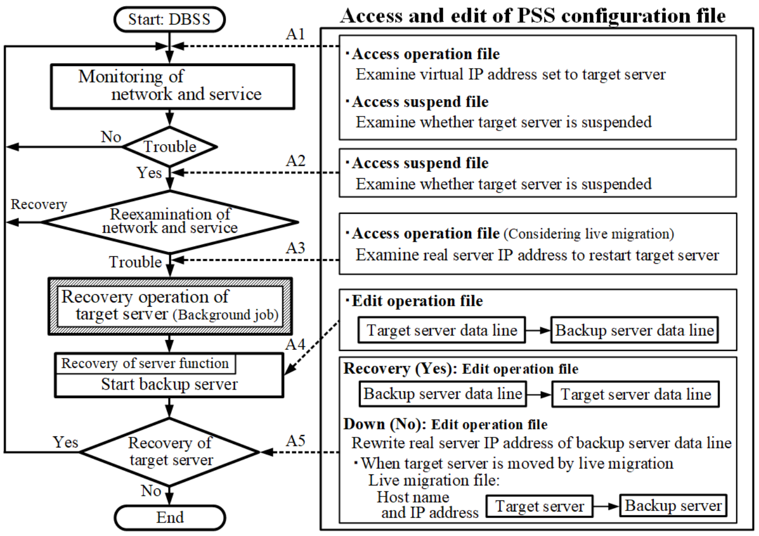

Start: DBSS
A1
Monitoring of network and service
No
Trouble
Yes
A2
Recovery
Reexamination of network and service
Trouble
A3
Recovery operation of target server (Background job)
A4
Recovery of server function
Start backup server
Yes
Recovery of target server
A5
No
End
Access and edit of PSS configuration file
・Access operation file
Examine virtual IP address set to target server
・Access suspend file
Examine whether target server is suspended
・Access suspend file
Examine whether target server is suspended
・Access operation file (Considering live migration)
Examine real server IP address to restart target server
・Edit operation file
Target server data line
Backup server data line
Recovery (Yes): Edit operation file
Backup server data line
Target server data line
Down (No): Edit operation file
Rewrite real server IP address of backup server data line
・When target server is moved by live migration
Live migration file:
Host name and IP address
Target server
Backup server

Fig. 7 Coordination of DBSS processing and PSS information updates in compound operation.

In A1, whether the target server is in the suspended state and the VIP address for the target server are examined. The processing of line A2 is performed after the DBSS detects trouble in the target server because there is a possibility that the state changes between A1 and A2. In A2, whether the target server is in the suspended state is determined. In A3, because there is a possibility that the target server has been moved from another real server by live migration, the IP address of the real server creating the target server is examined. This examination is necessary for restarting the troubled server during its recovery in the background job. In A4, because the function of the target server is recovered by starting the backup server, the data line of the target server is deleted and that of the backup server is added in the operation file so the PSS has current information. The processing of line A5 is performed according to the result of the recovery operation of the target server by the background job. Here, if the target server recovers, the data line for the backup server is deleted and that of the target server is restored in the PSS operation file. If the target server does not recover, the real server IP address in the data line for the backup server recorded in the operation file is rewritten to that of the destination real server, because a backup server created on the management server is moved by live migration. Here, when of the target server has been moved from another real server by live migration, the host name and its IP address in the data line for the target server recorded in the live migration file are rewritten to that of the backup server.

If the additional steps for updating information for the PSS were added to the DBSS, its program would become complicated. To retain the simple structure of the DBSS, we propose the SIF for updating PSS configuration files during compound operation. Figure 8 provides an outline of SIF operation. The SIF is automatically started with the DBSS, and it accesses and controls PSS configuration files according to the signals sent from the DBSS by inter-process communication. The SIF also provides any replies needed by the DBSS. Because the SIF receives the signal from the DBSS by interrupt processing, high-speed response is realized.

The lower half of Fig. 8 shows partial SIF operation as an example for DBSS (M1). If the DBSS sends signal 1 and host name M1 to the SIF, the SIF accesses the PSS operation and suspend files, examines the VIP address of M1 and whether M1 is in the suspend state, and replies to the DBSS with the results. Moreover, if the DBSS sends signal 2 and host names of M1 and Mb to the SIF, the SIF deletes the data line for failed M1 and adds that of Mb in the operation file. The DBSS can immediately execute the next processing step after sending signal 2 because it does not need to receive a reply. Introducing the SIF method allows us to realize compound operation with the PSS by adding only signal transmission and reception instructions to the DBSS.

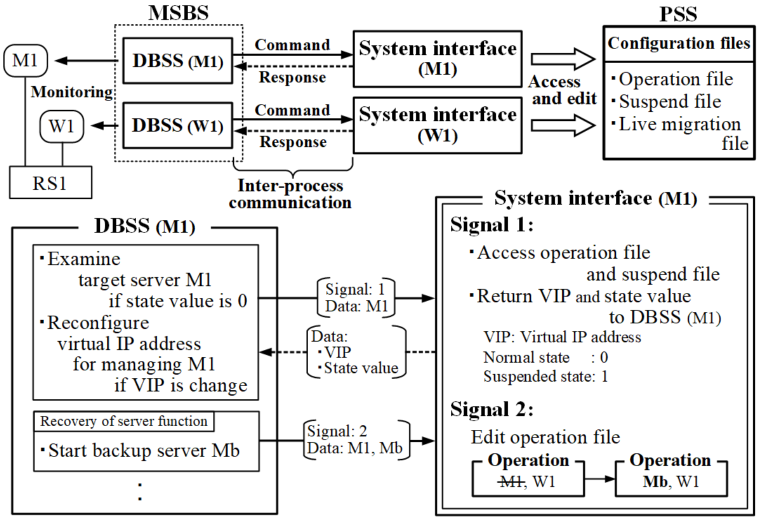


Fig. 8 Operation outline of system interface.

4. Conclusions

In this study, we adopted a MSBS and PSS that are operated simultaneously and independently. We examined the problems encountered during compound operation and developed solutions to them. The key solution was to control the accessing or editing of the PSS configuration files by the DBSS. Further, to retain the simple structure of the DBSS, we proposed a SIF to perform configuration file access and editing with instructions from the DBSS. The results of this study indicate that both power saving and high availability of the server system can be realized effectively. Therefore, expansion of Internet services affording greater convenience in our daily lives can be expected.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Commun. Surveys & Tutorials, Vol. 17, No. 4, 2015, pp. 2347-2376.

[2] Z. Lin and L. Dong, "Clarifying Trust in Social Internet of Things," IEEE Trans. Knowledge and Data Eng., Vol. 30, No. 2, 2018, pp. 234-248.

[3] N. A. Davis, A. Rezgui, H. Soliman, S. Manzanares, and M. Coates, "FailureSim: A System for Predicting Hardware Failures in Cloud Data Centers Using Neural Networks," IEEE Conference Proc., Vol. 2017, No. Cloud, 2017, pp. 544-551.

[4] D. A. Popescu and A. W. Moore, "PTPmesh: Data Center Network Latency Measurements Using PTP," IEEE Conference Proc., Vol. 2017, No. MASCOTS, 2017, pp. 73-79.

[5] M.-G. Rabbani, M.-F. Zhani, and R. Boutaba, "On Achieving High Survivability in Virtualized Data Centers," IEICE Trans. Commun., Vol. E97-B, No. 1, 2014, pp. 10-18.

[6] H. Otsuka, K. Joshi, M. Hiltunen, S. Daniels, and Y. Matsumoto, "Online Failure Prediction with Accurate Failure Localization in Cloud Infrastructures," IEICE Technical Report, Vol. 113, No. 496, 2014, pp. 7-12.

[7] M. Hirono, T. Sato, J. Matsumoto, S. Okamoto, and N. Yamanaka, "HOLST: Architecture Design of Energy-efficient Data Center Network based on Ultra High-speed Optical Switch," IEEE Conference Proc., Vol. 2017, No. LANMAN, 2017, pp. 1-6.

[8] J. K. Perin, A. Shastri, and J. M. Kahn, "Design of Low-Power DSP-Free Coherent Receivers for Data Center Links," Journal of Lightwave Tec., Vol. 35, No. 21, 2017, pp. 4650-4662.

[9] 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.

[10] M. Kitamura, "Configuration of a Power-saving High-availability Server System Incorporating a Hybrid Operation Method," JISSJ, Vol. 10, No. 2, 2015, pp. 1-17.

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.

Youta SHIMIZU

He received his B.E. degree in 2017 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.

Koki TANI

He received his B.E. degree in 2018 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.