［Original Paper］

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

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

† Graduate School of Engineering, Tokyo Polytechnic University
‡ Hitachi Information & Telecommunication Engineering, Ltd.

Abstract

This paper describes the configuration of a power-saving and high-availability server system achieved by the compound operation of a power-saving server system (PSS) and a multiple-server backup system (MSBS) that can operate independently. The PSS shifts the system configuration to either a normal condition or a power-saving condition according to server loads. In the power-saving condition, the proposed method has two server configurations. In the first, the servers providing services to clients are configured redundantly, and in the second they are not. Furthermore, a dynamic server recovery system that manages real servers is proposed in addition to a dynamic backup server system (DBSS) that manages a virtual server in the MSBS. Therefore, if real servers have trouble, recovery operations are automatic and prevent access disruption for the client. In the compound operation, the MSBS needs status information on configuration changes in the server system made by the PSS, and the PSS needs to execute the DBSS during the configuration change. Thus, a system interface, which has the functions of bi-directional control of the PSS and MSBS, is proposed to enhance the functions of the management program in the PSS and MSBS. In addition, an experimental server system incorporating the proposed method was constructed. In the power-saving condition, the proposed system operated at 70% of power consumption of the normal condition when virtual servers were configured redundantly, and operated at 40% when a non-redundant configuration was used. Experiments were conducted to reproduce several types of trouble in the virtual server or the real server in a server configuration under both normal and power-saving conditions. The proposed system is shown to have high availability in either condition.

1. Introduction

The Internet has become an important social infrastructure because an information-oriented society is rapidly developing [1]. The Internet of Things (IoT), in which many sensors and devices are connected to the Internet, generates new value by using information obtained from them [2], [3]. Server systems perform an important role in analyzing data obtained from these IoT devices. In data centers supporting cloud computing, problems in server systems could cause large losses to the cloud provider and its users [4]. Therefore, in constructing the server system, high availability is very important. In addition, it is necessary to sufficiently consider power saving in addition to high speed and security.

A number of important studies related to failure countermeasures in data centers have been conducted. Some studies have called for a precision time protocol for estimating network delay and packet loss in data centers [5], a high-availability virtual infrastructure management framework that considers the trouble rate of data center devices [6], and information collection and analysis of hardware failure in data centers [7]. Others have considered network coding to instantly start a hot spare node during virtual machine trouble [8], evaluated hardware trouble and service abnormality in relation to the security of a server and the network resources of a virtual data center [9], and developed an integrated method that combines the prediction of failure occurrence and the prediction of failure location, enabling automated or operator actions [10].

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 [11]. A report by the Ministry of Economy, Trade and Industry of Japan estimated that the power consumption by information technology devices will increase nine-fold by 2025, as compared with that in 2006 [12]. Therefore, a number of important studies on power-saving network systems have been conducted [13]-[16]. Although approximately half of all information technology devices are network devices and servers, little

[Original Paper] Received 2 July 2019, revised 22 September 2019, accepted 30 October 2019.

research has been conducted on power saving in server systems. One study considered power consumption of devices to determine server arrangement by a server relocation service [17]. Another demonstrated that a power-saving, high-availability server system could be constructed by alternately and independently operating a power-saving server system and a high-availability server system [18]. Finally, a virtual server can be used for the recovery of server functions on a real server system and can realize low cost and power savings by adopting a server management system that can back up the functions of several real servers by means of virtual servers [19].

Although many systems studies have focused on power saving, 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. For this reason, studies have been conducted on a construction method for a server system that provides both power saving and high availability, and avoids the need for an overly complicated management program. As a method for constructing the power-saving, high-availability server system, compound operation, which is realized by allowing only the multiple-server backup system (MSBS) [19] to request editing of configuration files for the power-saving server system (PSS) [18], is proposed. Further, to solve the complexity of each management program, a system interface (SIF) that implements its editing function has been introduced [20].

However, the system described in [20] has the following problems. In the server system configuration in the power-saving state, servers that provide services to clients have a non-redundant configuration. Therefore, if a server fails in the power-saving state, a service disconnection period occurs. Moreover, load measurement does not account for load fluctuations. In the control method for high availability, only virtual servers providing services to clients are managed, and the real servers creating virtual servers are not managed.

Therefore, in this paper, we propose a construction method for a power-saving, high-availability server system that solves the problems of [20] and adds new functions. As a construction method, a compound operation method in which the PSS and the MSBS operate independently is adopted. The PSS is improved by using two types of server system configurations in the power-saving state. When services are being provided to the client, servers providing services are configured redundantly, and when there is almost no access from the client, they are not configured redundantly. Moreover, for load measurement, the predicted load method [21] is adopted to consider load fluctuations. In the MSBS, in addition to multiple dynamic backup server systems (DBSSs), each of which manages only one target virtual server, multiple dynamic server recovery systems (DSRSs), each of which manages only one target real server, are added. As additional functions, the load calculation method according to server type, adoption of a spare server that supplements the service function provided to the client in the server system configuration of the power-saving state, and the SIF that enables bi-directional control of the PSS and MSBS are added. 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. Usually, if two kinds of management programs are operated at the same time, each program becomes complicated to coordinate actions and prevent malfunction. To realize compound operation, the MSBS is allowed to request editing of the PSS configuration files, and the PSS is allowed to start the DBSS that configures the MSBS. We propose and introduce the SIF that enables bi-directional control of the PSS and MSBS. Actual file access and editing are performed by the SIF with instructions from the MSBS to avoid making the program configuring the MSBS more complicated. In addition, the PSS uses a spare server to realize power saving. To avoid complicating the startup control of the DBSS in the PSS, the DBSS that manages the spare server is started by the SIF after it receives instructions from the PSS. An experimental system incorporating the proposed method was constructed, and its performance was evaluated by reproducing various troubles in the target server under both normal and power-saving conditions and observing server recovery.

The remainder of this paper is organized as follows. Section 2 outlines the construction of the proposed system and operation of the PSS and MSBS. Section 3 describes the method of solving problems in compound operation of the PSS and MSBS and describes the functions of the SIF. Section 4 describes the construction and specifications of an experimental system incorporating the proposed method and demonstrates that the proposed system can deal with various types of trouble in the target server under both the normal and power-saving conditions. The recovery times for various types of trouble were measured.

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

The proposed system is based on [20], and the power-saving, high-available server system is constructed by the compound operation method of the PSS [18] and MSBS [19]. In [18], if a server fails while the server system is configured in the power-saving state, the client experiences service disconnection. Therefore, the proposed system has two types of server configurations in the power-saving state. In the power-saving state, service disconnection does not occur when a server fails because servers are configured redundantly when they are providing services to clients. In [19], a server system in which real servers provide services is targeted for management. If this method is adopted for a server system in which virtual servers provide services, the real server creating the virtual server cannot be managed. Therefore, in the MSBS for the proposed system, the DSRS, which manages real servers, is added in addition to the DBSS, which manages virtual servers.

2.1. Configuration outline of proposed system

Figure 1 shows the outline of the proposed system. The proposed system is based on a load balancer, which is generally used to construct high-availability, distributed-load server systems. The real server group (RSG) creates virtual servers, and the virtual server group (VSG) offers services to clients. Here, the offered services are mail, web, and file transfer protocol (FTP) services. Virtual servers are distributed redundantly among each service group, and clients access the virtual servers via the round-robin method. The PSS and MSBS are installed on the management server, which monitors and controls the VSG and the RSG.

The PSS measures the load of the VSG and, when the load is judged as low, gathers virtual servers to a specific real server and suspends the unnecessary real servers to realize power saving. If the MSBS detects trouble, it creates a backup server on the management server and recovers functions of the trouble server. Live migration is used as a method of gathering virtual servers in [20]. However, a method using a spare server is adopted here to gather virtual servers because the startup time of the virtual server, which is set to enable live migration, tends to be delayed. Normally, redundant target servers can be managed by the load balancer. However, if a target server fails, there is a possibility that redundancy cannot be maintained. Therefore, the MSBS is used to maintain redundancy.

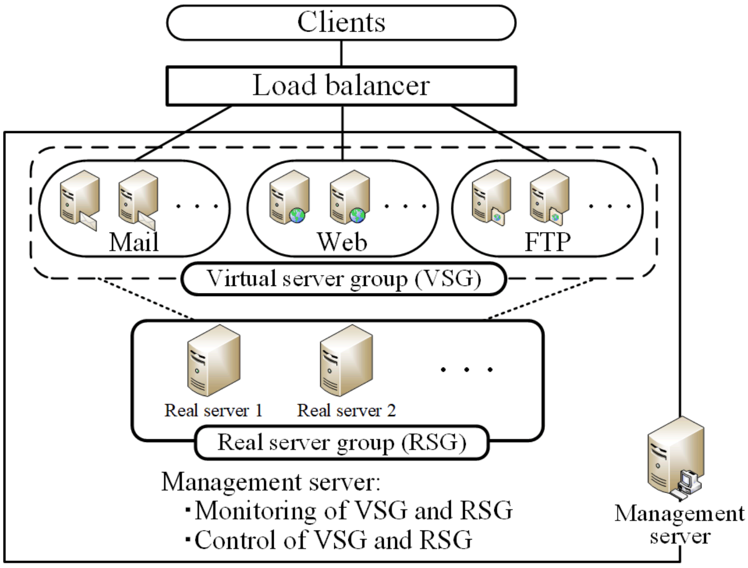


Fig. 1 Configuration of proposed system.

2.2. Operation flow and details of PSS

Figure 2 shows the operational flow and the configuration change of a server system by the PSS, where M1, M2, W1, W2, F1, and F2 are the virtual servers to be managed; Ms, Ws, and Fs are spare servers; RS1, RS2, and RS3 are the real servers that create them; and VIP denotes a virtual IP address. The virtual servers belonging to each service group are configured redundantly.

M1 and M2 are a service group for mail, W1 and W2 are a service group for web, and F1 and F2 are a service group for FTP. The PSS measures the loads (memory usage, CPU utilization, and transmission and reception of bytes in the network) of the virtual servers belonging to each service group. Then the PSS calculates the predicted loads and classifies them as a moderate load class or a light load class according to the load. The server configuration is changed to the normal (moderate) state or the power-saving (light) state according to the load class.

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 VIP addresses of redundant servers by the load balancer. When the PSS shifts the system to a light load condition, virtual servers are gathered to a specific real server called the base server. In the light load condition, two server conditions, power saving 1 (PS1) and power saving 2 (PS2), are set. Here, PS2 is designed in consideration of a server system in which the client frequently accesses the server during the day but does not access the server at all during the night. Therefore, a threshold for the number of accesses by clients is adopted as the criterion for shifting to PS2.

The server condition shifts to PS1 if the load condition shifts from moderate to light. In the PS1 state, because RS1 and RS2 are base servers, Fs is created on RS1, Ws is created on RS2, and then the VIP addresses of W2 and F2 are respectively changed to those for Ws and Fs. The PSS realizes power saving by suspending RS3. In condition PS1, the server condition shifts to PS2 when the number of accesses from clients is less than or equal to the threshold x. In condition PS2, because RS1 is the base server, the VIP addresses of F1, M2, and Ws are respectively changed to those for Fs, M1, and W1. The PSS realizes power saving by suspending RS2. Because the virtual servers are redundantly configured in condition PS1, the offered services are maintained if trouble is generated in the virtual server. The power consumption is approximately 2/3 that of the moderate load condition. In condition PS2, the virtual server is in a non-redundant configuration; however, the effect of server trouble in PS2 can be significantly reduced if the threshold value x is set to a low value. The power consumption is approximately 1/3 that of the moderate load condition.

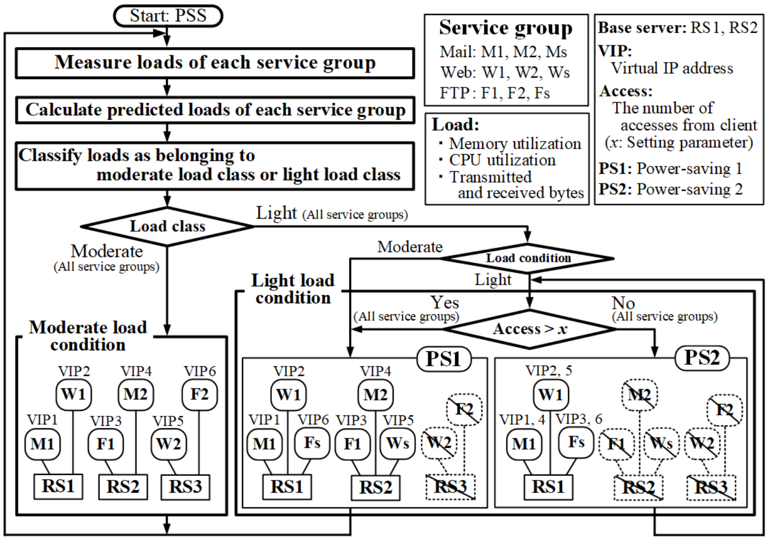
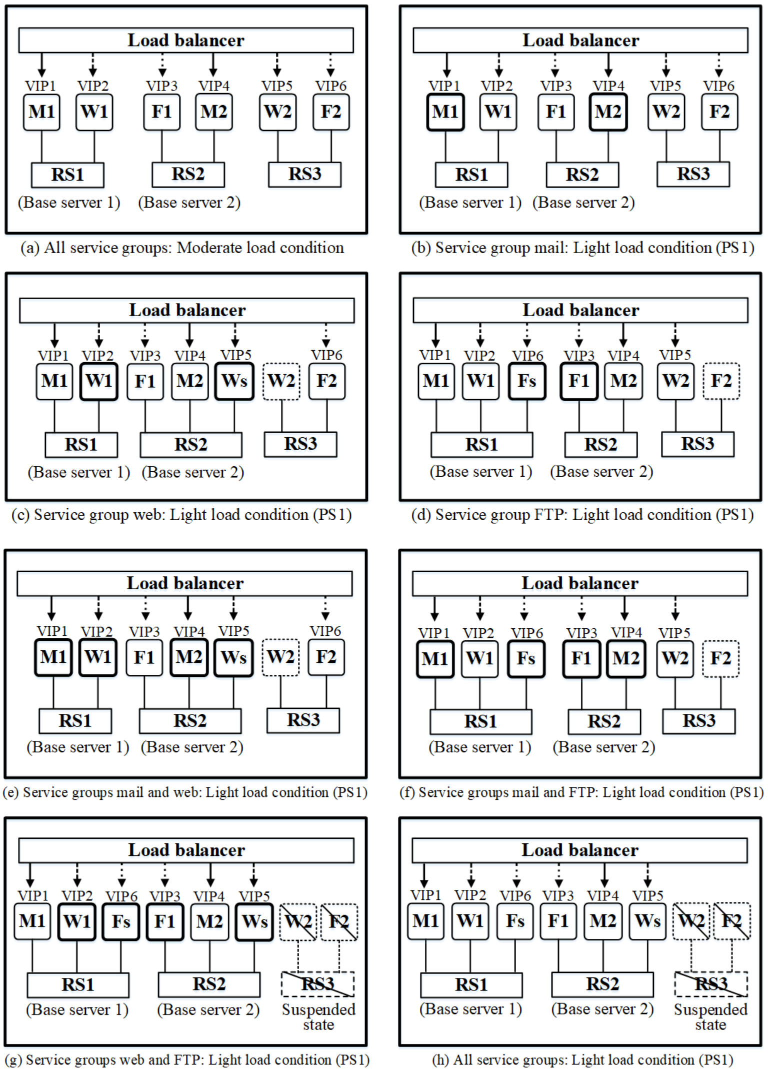


Fig. 2 Operational flow of PSS.

2.3. Shift to light load condition (PS1)

Figure 3 shows the state of light load condition PS1 for each service group. Figure 3(a) shows all service groups in the moderate load condition. Access to the servers belonging to the service group mail is indicated by a solid arrow, access to the servers belonging to the service group web is indicated by a dashed arrow, and access to the servers belonging to the service group FTP is indicated by a dotted arrow. The virtual servers belonging to each service group are configured redundantly. Clients access the VIP address set in the virtual servers by the round-robin method via the load balancer. Figure 3(b) shows that only the service group mail is in condition PS1. Because M1 and M2 are created from RS1 and RS2, which are base servers, the server system does not change from the condition of Fig. 3(a). Figure 3(c) shows that only the service group web is in condition PS1. Ws is created on RS2, which is the base server, and VIP5 is changed from W2 to Ws. Therefore, W2 becomes inaccessible to the load balancer. We define this state as an access stop state. Figure 3(d) shows that only the service group FTP is in condition PS1. Fs is created on RS1, which is the base server, and VIP6 is changed from F2 to Fs. Therefore, F2 enters the access stop state.

Figure 3(e) shows that the service groups mail and web are in condition PS1. Ws is created on RS2 and VIP5 is changed from W2 to Ws. Therefore, W2 enters the access stop state. Figure 3(f) shows that the service groups mail and FTP are in condition PS1. Fs is created on RS1 and VIP6 is changed from F2 to Fs. Therefore, F2 enters the access stop state. Figure 3(g) shows that the service groups web and FTP are in condition PS1. Because RS1 and RS2 are base servers, Fs is created on RS1 and Ws is created on RS2, and then VIP5 of W2 and VIP6 of F2 are respectively set to those for Ws and Fs. Here, all virtual servers created on RS3 enter the access stop state. Thus, the PSS suspends RS3 to realize power saving, and the power consumption of the server system is approximately 2/3 that of the moderate load condition. Figure 3(h) shows that all service groups are in condition PS1. Because the virtual servers created on RS3 are W2 and F2, the server system has the same configuration as in Fig. 3(g).



(a) All service groups: Moderate load condition

(b) Service group mail: Light load condition (PS1)

(c) Service group web: Light load condition (PS1)

(d) Service group FTP: Light load condition (PS1)

(e) Service groups mail and web: Light load condition (PS1)

(f) Service groups mail and FTP: Light load condition (PS1)

(g) Service groups web and FTP: Light load condition (PS1)

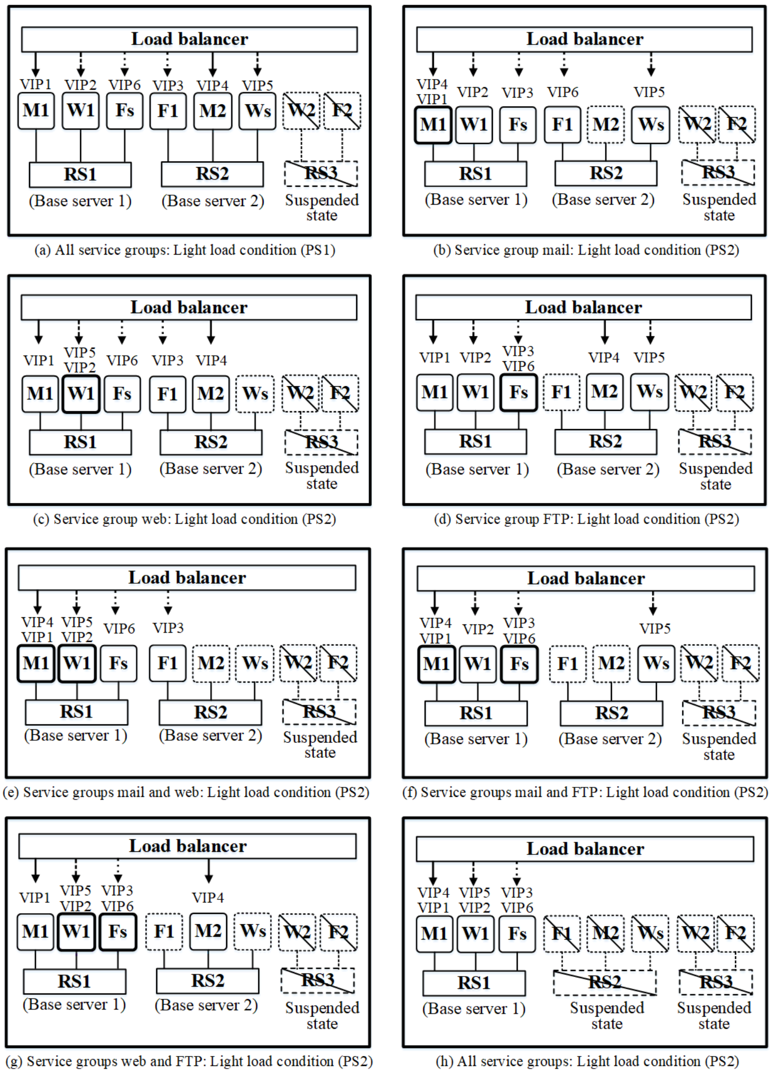
(h) All service groups: Light load condition (PS1)

Fig. 3 Shift to light load condition (PS1).

2.4. Shift to light load condition (PS2)

Figure 4 shows the state of light load condition PS2 for each service group. Figure 4(a) shows that all service groups are in condition PS1. Figure 4(b) shows that only the service group mail is in condition PS2. VIP4 is changed from M2 to M1. Therefore, M2 enters the access stop state. Figure 4(c) shows that only the service group web is in condition PS2. VIP5 is changes from Ws to W1. Therefore, Ws enters the access stop state. Figure 4(d) shows that only the service group FTP is in condition PS2. VIP3 is changed from F1 to Fs. Therefore, F1 enters the access stop state.

Figure 4(e) shows that the service groups mail and web are in condition PS2. VIP4 of M2 and VIP5 of Ws are respectively changed to those for M1 and W1. Therefore, M2 and Ws enter the access stop state. Figure 4(f) shows that the service groups mail and FTP are in condition PS2. VIP4 of M2 and VIP3 of F1 are respectively changed to those for M1 and Fs. Therefore, M2 and F1 enter the access stop state. Figure 4(g) shows that the service groups web and FTP are in condition PS2. VIP5 of Ws and VIP3 of F1 are respectively changed to those for W1 and Fs. Therefore, Ws and F1 enter the access stop state. Figure 4(h) shows that all service groups are in condition PS2. VIPs of the virtual servers created on RS2 are changed to the virtual servers created on RS1. Here, the virtual servers created on RS2 enter the access stop state. Thus, the PSS suspends RS2 to realize power saving, and the power consumption of the server system is approximately 1/3 that of the moderate load condition.



(a) All service groups: Light load condition (PS1)

(b) Service group mail: Light load condition (PS2)

(c) Service group web: Light load condition (PS2)

(d) Service group FTP: Light load condition (PS2)

(e) Service groups mail and web: Light load condition (PS2)

(f) Service groups mail and FTP: Light load condition (PS2)

(g) Service groups web and FTP: Light load condition (PS2)

(h) All service groups: Light load condition (PS2)

Fig. 4 Shift to light load condition (PS2).

2.5. Load measurement and calculation of predicted load

The proposed method uses the predicted load method and the load calculation method according to server type which are given in [21]. Although the basic calculation method is the same, [21] applies the load calculation to each server, but the proposed method applies the load calculation to each service group consisting of multiple servers.

Figure 5 shows the load measurement and calculation of predicted load in the PSS. The PSS periodically performs measurements of the load in the virtual server by considering the memory, CPU, and network usage. Here, *i* is the number of measurements, *mi* is the memory usage of the server, *ci* is the CPU utilization, and *ni* is the total bytes transmitted and received by the network. Measurement of the memory load is performed in each virtual server using the command “free” in UNIX. Measurement of the CPU load and the network load is performed in each virtual server using the command “sar” in UNIX. Based on these measured values, average load values *A*m, *A*c, and *A*n are calculated. The difference in the average load value acquired at two different times is used to calculate the predicted load. The predicted memory load $M_{\text{use\_server}}$ (in kilobytes) is defined as follows:

$$M_{\text{use\_server}} = 2 \times A\text{m}_n - A\text{m}_{n\text{-}1}. \tag{1}$$

The predicted CPU load $C_{\text{use\_server}}$ (in percent) is given as follows:

$$C_{\text{use\_server}} = 2 \times Ac_n - Ac_{n\text{-}1}. \tag{2}$$

The predicted network load $N_{\text{use\_server}}$ (in kilobytes) is defined as follows:

$$N_{\text{use\_server}} = 2 \times A\text{n}_n - A\text{n}_{n\text{-}1}. \tag{3}$$

Because the predicted loads obtained from equations (1), (2), and (3) may have negative values and exceed upper limits due to fluctuations, the minimum values are set to zero and the maximum values are the set memory size of the virtual server for $M_{\text{use\_server}}$, 100% for $C_{\text{use\_server}}$, and the rated transfer speed × 2 for $N_{\text{use\_server}}$.

In “Calculation of service group load” in Fig. 5, the service group web is used as an example for calculating the load. The memory load *L*m (in percent) in the service group web is defined as follows:

$$L\text{m} = \frac{M_{\text{use\_W1}} + M_{\text{use\_W2}}}{M_{\text{max\_W1}} + \text{M}_{\text{max\_W2}}} \times 100, \tag{4}$$

where M_{use} is memory usage and M_{max} is maximum memory size. The CPU load *L*c (in percent) in the service group web is given by

$$Lc = \frac{C_{\text{use\_W1}} \times C_{\text{cor\_W1}} + C_{\text{use\_W2}} \times C_{\text{cor\_W2}}}{C_{\text{cor\_W1}} + C_{\text{cor\_W2}}}, \tag{5}$$

where C_{use} is the CPU utilization and C_{cor} is the number of CPU cores. The network load *L*n (in percent) in the service group web is calculated by

$$L\text{n} = \frac{N_{\text{use\_W1}} + N_{\text{use\_W2}}}{N_{\text{rat\_W1}} + N_{\text{rat\_W2}}} \times 100, \tag{6}$$

where N_{use} 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 rated at 1,000 Mbps is taken as an example, because 128 × 10^3 KB can be transferred in 1 s and *N*rat is 256 × 10^3 KB from the total value of transmission and reception.

The server type method [21] is introduced to calculate the load of each service group. This method defines a CPU-type server, which mainly uses the CPU; a NETWORK-type server, which mainly provides data; and a CPU_NET-type server, which both uses the CPU and provides data. When a general server is taken as an example, a login server is defined as the CPU type and FTP and mail servers are defined as the NETWORK type. A web server is defined as a CPU_NET-type server because it provides HTML and image data and uses computer-generated imagery. Here, *L*m, *L*c, and *L*n are weighted according to the server type, and the total value obtained from these is set to 100 at the maximum. Because the memory load *N*m is important for every server type, *N*m has a weight of 0.5. The load in the service group varies depending on its type. The load for the CPU-type server is given by

$$Ls = L\text{m} \times 0.5 + Lc \times 0.4 + L\text{n} \times 0.1. \tag{7}$$

The load for the NETWORK-type server is defined as follows:

$$Ls = L\text{m} \times 0.5 + Lc \times 0.1 + L\text{n} \times 0.4. \tag{8}$$

The load for the CPU_NET-type server is calculated by

$$Ls = L\text{m} \times 0.5 + Lc \times 0.25 + L\text{n} \times 0.25. \tag{9}$$

Because these weightings are used in [21], we adopt them here. However, it is desirable to determine these weightings according to the operating environment of a server system during actual operation.

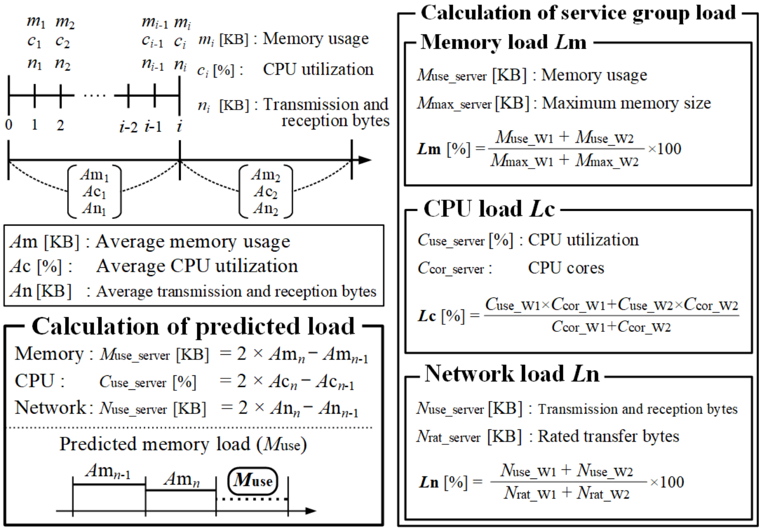

m_1 m_2
c_1 c_2
n_1 n_2
m_{i-1} m_i
c_{i-1} c_i
n_{i-1} n_i
0 1 2 ···· i-2 i-1 i
m_i [KB] : Memory usage
c_i [%] : CPU utilization
n_i [KB] : Transmission and reception bytes
Am_1 Ac_1 An_1
Am_2 Ac_2 An_2
Am [KB] : Average memory usage
Ac [%] : Average CPU utilization
An [KB] : Average transmission and reception bytes
Calculation of predicted load
Memory : M_{use_server} [KB] $= 2 \times Am_n - Am_{n-1}$
CPU : C_{use_server} [%] $= 2 \times Ac_n - Ac_{n-1}$
Network : N_{use_server} [KB] $= 2 \times An_n - An_{n-1}$
Predicted memory load (M_{use})
Am_{n-1} Am_n M_{use}
Calculation of service group load
Memory load Lm
M_{use_server} [KB] : Memory usage
M_{max_server} [KB] : Maximum memory size
$Lm\ [\%] = \frac{M_{use\_W1} + M_{use\_W2}}{M_{max\_W1} + M_{max\_W2}} \times 100$
CPU load Lc
C_{use_server} [%] : CPU utilization
C_{cor_server} : CPU cores
$Lc\ [\%] = \frac{C_{use\_W1} \times C_{cor\_W1} + C_{use\_W2} \times C_{cor\_W2}}{C_{cor\_W1} + C_{cor\_W2}}$
Network load Ln
N_{use_server} [KB] : Transmission and reception bytes
N_{rat_server} [KB] : Rated transfer bytes
$Ln\ [\%] = \frac{N_{use\_W1} + N_{use\_W2}}{N_{rat\_W1} + N_{rat\_W2}} \times 100$

Fig. 5 Load measurement and calculation method for predicting load.

2.6. Change of server condition

In Fig. 6(a), the PSS performs classification according to the value of load *Ls* for each service group and classifies the load as light or moderate according to two thresholds. As shown in the figure, loads that exceed the moderate load threshold are represented by area M. Loads that fall between the moderate and light load thresholds are represented by area ML. Finally, loads that are lower than the light load threshold are represented by area L. The vectors indicating loads (i) through (iv) show the changes in the loads. Black dots indicate the initial state, and the arrowhead indicates the final state. If the load state shifts to area ML, as shown in loads (i) and (iii), the load is classified as belonging to the initial state. This is done to reduce the influence of fluctuations near the threshold. Loads (i) and (iv) become the moderate load class, and loads (ii) and (iii) become the light load class.

Judgments for three service groups are performed. As shown in Fig. 6(b), 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 (PS1) only after a light load class was detected four times in a row. The setting of four times is an example, and in the proposed system this setting can be changed in consideration of the operating environment of the server system.

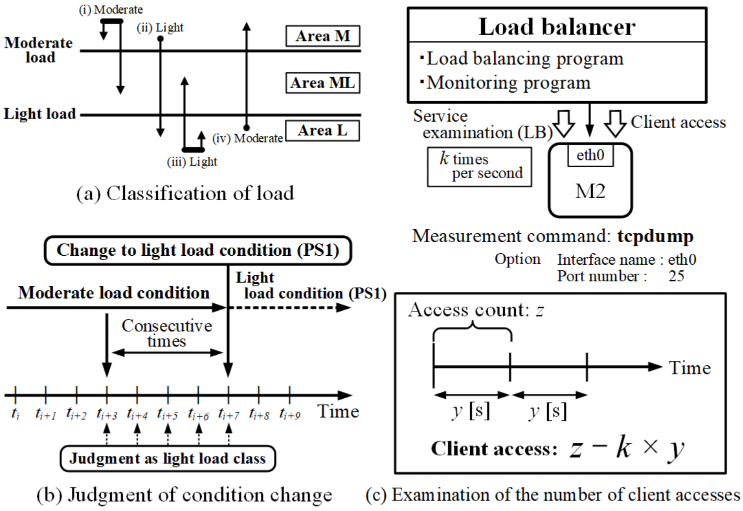

(i) Moderate
(ii) Light
Moderate load
Light load
Area M
Area ML
Area L
(iv) Moderate
(iii) Light
(a) Classification of load
Load balancer
·Load balancing program
·Monitoring program
Service examination (LB)
Client access
k times per second
eth0
M2
Measurement command: tcpdump
Option Interface name : eth0
Port number : 25
Change to light load condition (PS1)
Moderate load condition
Light load condition (PS1)
Consecutive times
t_i t_{i+1} t_{i+2} t_{i+3} t_{i+4} t_{i+5} t_{i+6} t_{i+7} t_{i+8} t_{i+9} Time
Judgment as light load class
(b) Judgment of condition change
Access count: z
Time
y [s] y [s]
Client access: $z - k \times y$
(c) Examination of the number of client accesses

Fig. 6 Shift of load condition and examination of the number of accesses.

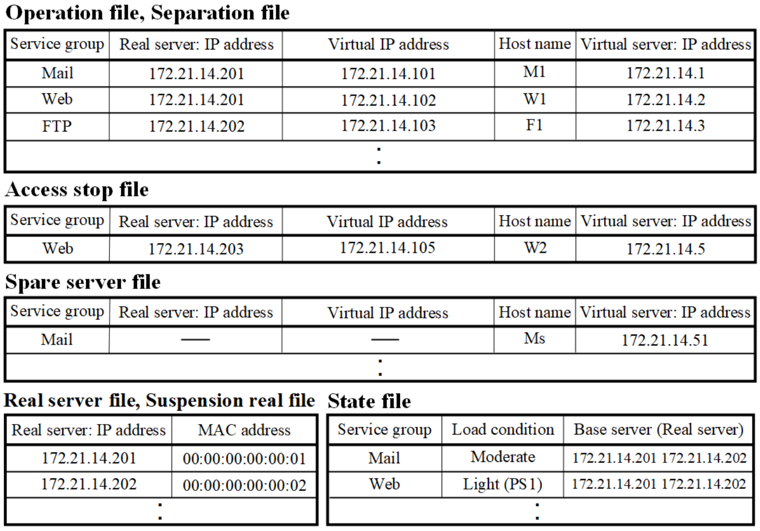
We adopted the number of accesses from clients as the criterion for shifting to condition PS2. Measurement of server access is performed by using the command “tcpdump” in UNIX. Fig. 6(c) shows the method for measuring the number of accesses from clients. A load balancing program and a monitoring program are run in the load balancer. Thus, M2 is accessed by both clients and the monitoring program. The monitoring program performs *k* accesses to M2 per second. The access from clients for *y* seconds can be calculated by subtracting the number of accesses by the monitoring program ($k \times y$) from the total number of accesses (z).

2.7. PSS configuration files

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 competitive editing, these configuration files adopt a locking function that refuses access from other programs during editing.

Detailed information for the virtual servers is recorded in the operation file, which contains the service group name, the IP address of the real server that created the virtual server, the VIP address, the host name of the virtual server, and the IP address of the virtual server. Here, as in W2 shown in Figure 3(c), a virtual server that enters a state inaccessible from the load balancer is defined as access stopped. The PSS moves the information of the virtual server in the access stopped state from the operation file to the access stop file. If the virtual server returns to the moderate load condition, the virtual server information is moved from the access stop file back to the operation file. Information on the virtual server generating trouble is moved from the operation file to the separation file. Moreover, the spare server file contains virtual server information for complementing the missing services on the base server during the shift to the light load condition. Because the real server and the VIP address for the spare server are determined during the shift, its information is not recorded in the spare server file. Information on the real server creating the virtual server is recorded in the real server file, which contains the IP address of the real server creating the virtual server and the MAC address. Here, the MAC address is used for re-activating a real server in the suspended state. The PSS moves the information of the real server in the suspended state from the real server file to the suspension real file. If the real server returns to the active state, the real server information is moved from the suspension real file back to the real server file. The state of each service group, moderate, light (PS1), or light (PS2), is recorded in the state file, which contains the service group name, the state of the group, and the IP address of the base server that gathered virtual servers in each service group during power saving. Because multiple base servers are required in condition PS1, multiple IP addresses are recorded for them.

Table 1 PSS configuration files.



Operation file, Separation file

Service group	Real server: IP address	Virtual IP address	Host name	Virtual server: IP address
Mail	172.21.14.201	172.21.14.101	M1	172.21.14.1
Web	172.21.14.201	172.21.14.102	W1	172.21.14.2
FTP	172.21.14.202	172.21.14.103	F1	172.21.14.3
:				

Access stop file

Service group	Real server: IP address	Virtual IP address	Host name	Virtual server: IP address
Web	172.21.14.203	172.21.14.105	W2	172.21.14.5

Spare server file

Service group	Real server: IP address	Virtual IP address	Host name	Virtual server: IP address
Mail	—	—	Ms	172.21.14.51
:				

Real server file, Suspension real 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	172.21.14.201 172.21.14.202
Web	Light (PS1)	172.21.14.201 172.21.14.202
:		

2.8. Operation of MSBS

A schematic of the MSBS is shown in Fig. 7. In the figure, M1 and W1 are virtual servers to be monitored, 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 denotes the virtual IP address. The DBSS that manages target server M1 is denoted as DBSS (M1) and the DSRS that manages target server RS1 is denoted as DSRS (RS1).

As shown in Fig. 7(a), the MSBS is realized by multiple DBSSs and DSRSs. The DSRS is proposed and introduced in this research. Each DBSS monitors the network and the state of service on its target virtual server in the VSG, and each DSRS monitors the network and the state of virtualization software on its target real server in the RSG. Basically, the DBSS has one backup server to manage one target server. In Fig. 7(b), DBSS (M1) detects trouble in target server M1, starts backup server Mb on the management server, and then sets address VIP1. As a result, the function of M1 is recovered. Thereafter, Mb is moved to RS1 by live migration to reduce the load on the management server. In addition, DBSS (M1) automatically starts DBSS (Mb) to manage Mb and is then terminated.

If a real server fails, the recovery processing in the MSBS is as shown in Fig. 7(c). Virtual servers created on the real server also enter a trouble state if the real server fails. If DBSS (M1) and DBSS (W1) detect trouble in M1 and W1, Mb and Wb are created on the management server, and a VIP address is set for each. Thereafter, DBSS (M1) and DBSS (W1) automatically start DBSS (Mb) and DBSS (Wb) and then are terminated. If DSRS (RS1) detects trouble in RS1, it examines whether RS1 is in a restart state. If RS1 is determined to be in trouble, it attempts to restart RS1 by using wake on LAN. If RS1 starts up, DSRS (RS1) creates M1 and W1 on RS1, and then VIP addresses of Mb and Wb are respectively set to those for M1 and W1. Thereafter, DSRS (RS1) executes DBSS (M1) and DBSS (W1) and then terminates DBSS (Mb) and DBSS (Wb) and suspends Mb and Wb.

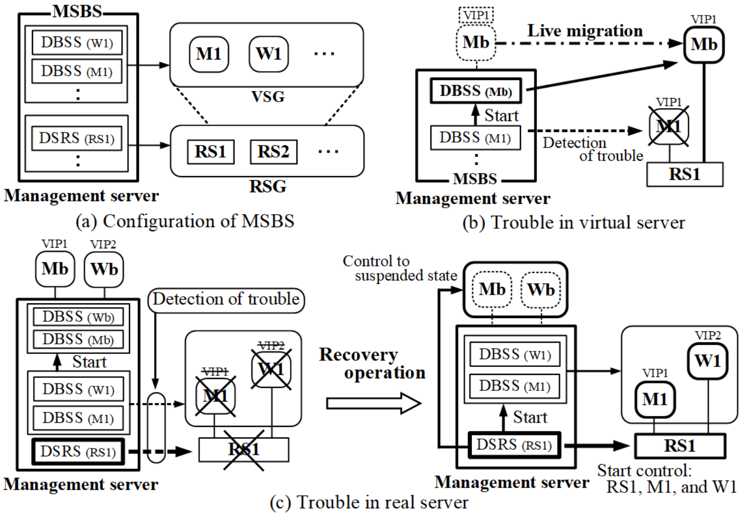


Fig. 7 Configuration and operation outline of MSBS.

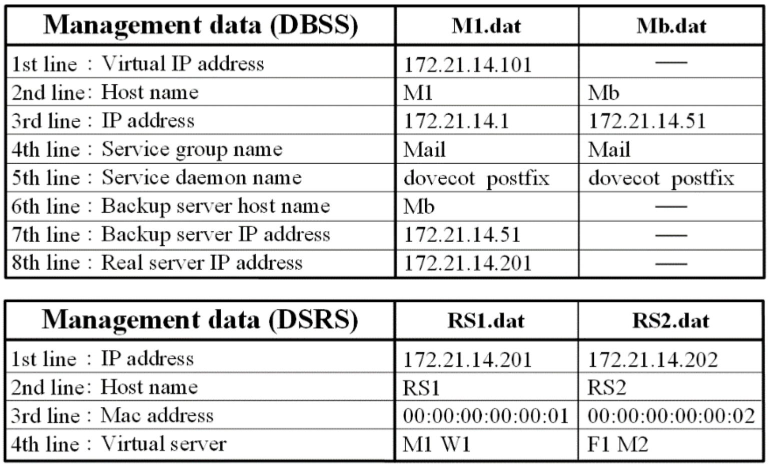
2.9. Management data used by DBSS and DSRS

Table 2 shows the management data used by the DBSS and DSRS. These data are in a read-only file. The DBSS and DSRS are executed using the management data as an argument. M1.dat used by the DBSS represents data used to manage virtual server M1, Mb.dat represents data used to manage backup server Mb, and RS1.dat and RS2.dat used by the DSRS represent data used to manage these respective real servers. The management data for the DBSS are provided as eight lines for each target virtual server. The first line contains the server’s VIP address. 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. The VIP address and the real server are determined for the first time when the backup server is started. Therefore, the first and eighth lines are blank. In addition, because the backup server is not set, the sixth and seventh lines are also blank. The management data for the

DSRS are provided as four lines for each target real server. Target real server information is recorded in the first to third lines, and the third line contains the MAC address used for wake on LAN. Moreover, the fourth line contains virtual server names creating on the real server. Because one real server creates multiple virtual servers, the names of multiple virtual servers can be recorded in the fourth line using space separation, as in “M1 W1.”

Table 2 Management data used by DBSS and DSRS.



Management data (DBSS)	M1.dat	Mb.dat
1st line : Virtual IP address	172.21.14.101	—
2nd line: Host name	M1	Mb
3rd line : IP address	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	172.21.14.51	—
8th line : Real server IP address	172.21.14.201	—

Management data (DSRS)	RS1.dat	RS2.dat
1st line : IP address	172.21.14.201	172.21.14.202
2nd line: Host name	RS1	RS2
3rd line : Mac address	00:00:00:00:00:01	00:00:00:00:00:02
4th line : Virtual server	M1 W1	F1 M2

3. Compound operation of PSS and MSBS

In general, if two kinds of management programs are operated at the same time, both programs become complex to prevent malfunction. 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. It is assumed that multiple DBSSs access the configuration files for the PSS. Thus, a file lock function in the PSS is adopted to prevent malfunction in the file access.

3.1. Problems of compound operation and their solution

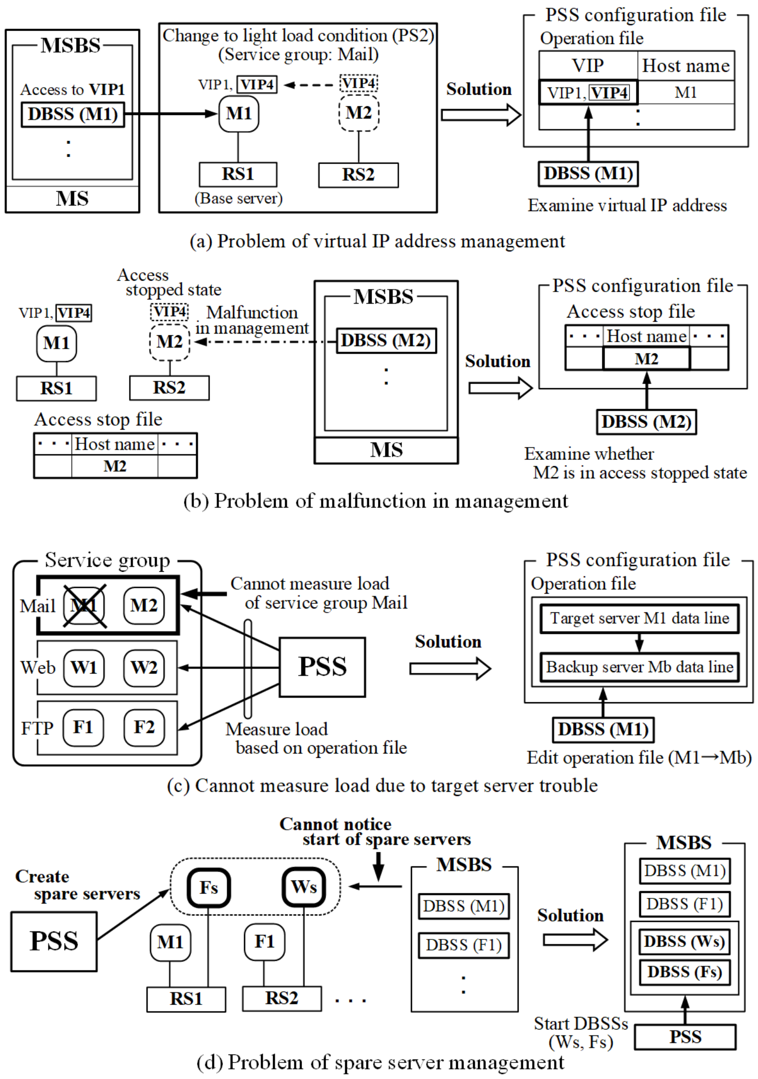
Figure 8 shows the typical problems and solutions of compound operation of the PSS and MSBS. The notation is the same as that used in previous figures.

Figure 8(a) shows the problem and the solution related to management of the VIP address when the PSS configures the system for the light load condition (PS2). Here, the DBSS monitors M1 and examines access to VIP1. When the service group mail is changed to the light load condition (PS2) by the PSS, the 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 8(b) shows the problem and the solution related to malfunction in the management by the DBSS when the PSS configures the system for light load operation (PS2). The DBSS (M2) is monitoring M2. When the service group mail is changed to the light load condition (PS2) by the PSS, M2 enters a state in which the VIP address is not set, as shown in Fig. 8(a). However, DBSS (M2) cannot notice that. Therefore, the DBSS can cause a management malfunction. As a countermeasure, it is necessary for the DBSS to know the data for M2 recorded in the PSS access stop file.

Figure 8(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 service group mail 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 service group mail. 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.

Figure 8 (d) shows the problem and solution when the PSS creates a spare server. The PSS creates spare servers Fs and Ws when it is shifting the server condition to the light condition (PS1). Each spare server is a server that provides services to clients. The MSBS must manage spare servers, but the MSBS cannot notice the start of spare servers. Therefore, as a countermeasure, when a spare server is started by the PSS, it is necessary for the PSS to control the start of the DBSS to manage the spare server.



(a) Problem of virtual IP address management

(b) Problem of malfunction in management

(c) Cannot measure load due to target server trouble

(d) Problem of spare server management

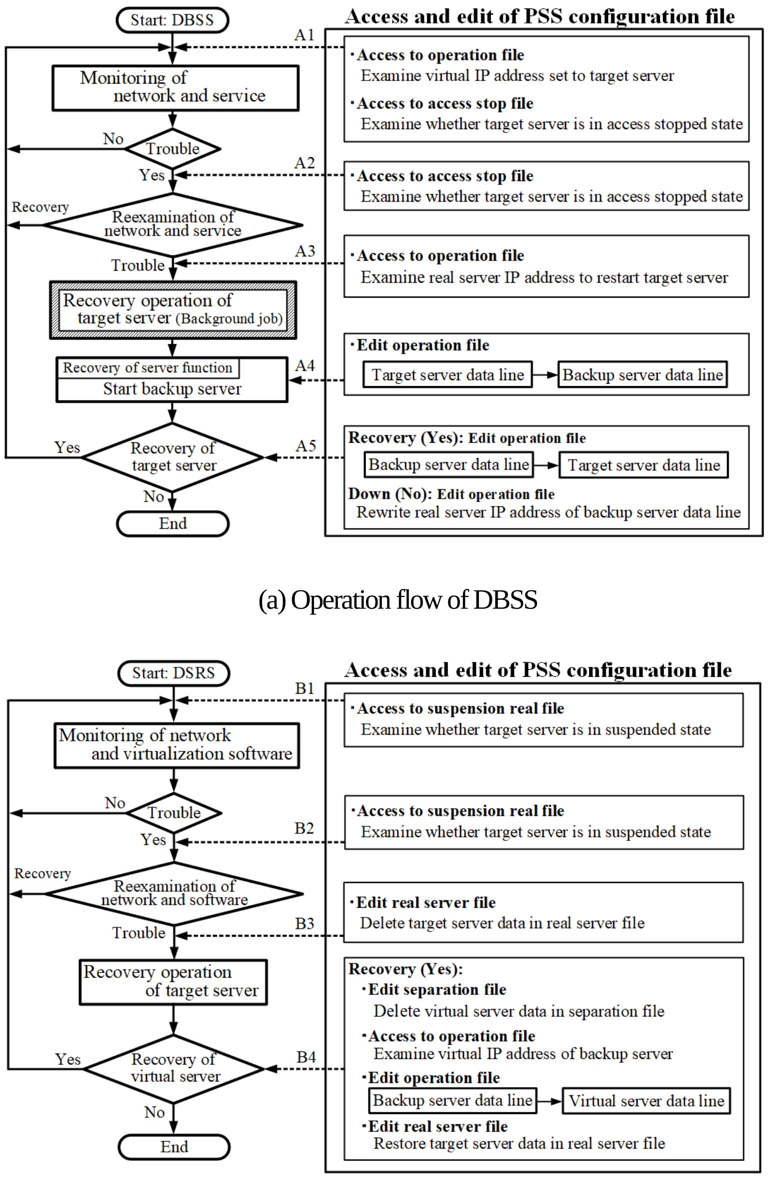
Fig. 8 Problems and solutions in compound operation of PSS and MSBS.

3.2. Additional processing by DBSS and DSRS in compound operation

To realize compound operation with the PSS and MSBS, processing for controlling the PSS is coordinated with operation of the DBSS and the DSRS. Figure 9 provides operation flowcharts of the DBSS and the DSRS that show the timing of the needed PSS control actions. The DBSS or the DSRS manages only one target server and has a very simple structure.

In Fig. 9(a), 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. The processing of line A1 is performed just before the DBSS monitors the target server. In A1, whether the target server is in the access stopped 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 access stopped state is determined. The processing of line A3 is performed after the DBSS reexamines the trouble of the failed server. In

A3, 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. When the DBSS recovers the server function, the processing of line A4 is performed. 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; therefore, 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 the backup server created on the management server is moved by live migration.


Start: DBSS
Access and edit of PSS configuration file
A1
Monitoring of network and service
·Access to operation file
Examine virtual IP address set to target server
·Access to access stop file
Examine whether target server is in access stopped state
No
Trouble
Yes
A2
·Access to access stop file
Examine whether target server is in access stopped state
Recovery
Reexamination of network and service
Trouble
A3
·Access to operation file
Examine real server IP address to restart target server
Recovery operation of target server (Background job)
·Edit operation file
Target server data line
Backup server data line
Recovery of server function
Start backup server
A4
Yes
Recovery of target server
A5
Recovery (Yes): Edit operation file
Backup server data line
Target server data line
No
Down (No): Edit operation file
Rewrite real server IP address of backup server data line
End

(a) Operation flow of DBSS

Start: DSRS
Access and edit of PSS configuration file
B1
Monitoring of network and virtualization software
·Access to suspension real file
Examine whether target server is in suspended state
No
Trouble
Yes
B2
·Access to suspension real file
Examine whether target server is in suspended state
Recovery
Reexamination of network and software
Trouble
B3
·Edit real server file
Delete target server data in real server file
Recovery operation of target server
Recovery (Yes):
·Edit separation file
Delete virtual server data in separation file
·Access to operation file
Examine virtual IP address of backup server
Yes
Recovery of virtual server
B4
·Edit operation file
Backup server data line
Virtual server data line
No
·Edit real server file
Restore target server data in real server file
End

(b) Operation flow of DSRS

Fig. 9 Additional processing by DBSS and DSRS.

In Fig. 9(b), dotted arrows (B1 to B4) indicate the timing of the DSRS accessing or editing the PSS configuration file. Because the monitoring of the target real server by the DSRS and configuration changes by the PSS are processed asynchronously, the DSRS recognizes the configuration change at the timing B1 or B2. The processing of line B1 is performed just before the

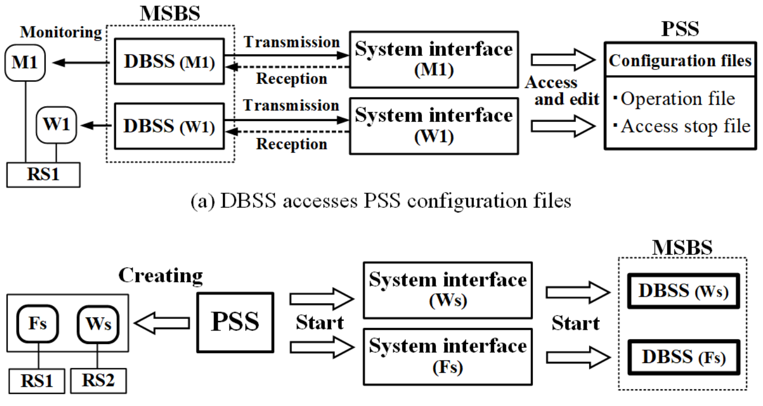
DSRS monitors the target real server. In B1, whether or not the target server is in the suspended state is examined. The processing of line B2 is performed after the DSRS detects trouble in the target server because there is a possibility of the state changing between B1 and B2. In B2, whether or not the target server is in the suspended state is determined. The processing of line B3 is performed after the DSRS reexamines the trouble of the failed server. In B3, the data line for the trouble server is deleted in the PSS real server file. The processing of line B4 is performed if the virtual servers are recovered after the recovery of the target real server. The data of the virtual servers created on the target real server are deleted in the PSS separation file. Then, the data lines for the backup servers are deleted and those of the virtual servers are restored in the PSS operation file. In addition, the data line for the target real server is restored in the real server file.

3.3. SIF capable of bi-directional control

If the additional steps for updating information for the PSS were added to the DBSS and the DSRS, their programs would become complicated. To retain the simple structure of the DBSS and the DSRS and enhance their functionality, we propose the SIF with functions to update PSS configuration files and start management of the spare server during compound operation. Figure 10 provides an outline of SIF operation.

In Fig. 10(a), the SIF is automatically started with the DBSS, and it accesses and controls the 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. Introducing the SIF method allows us to realize compound operation with the PSS by adding only signal transmission and reception instructions to the DBSS. As a result, the SIF avoids excessive complexity in the management program in the DBSS.

Fig. 10(b) shows the operation when the PSS creates a spare server. The PSS creates spare servers Ws and Fs when shifting the server condition to the light load condition (PS1). The MSBS must start the DBSS to manage the spare server; however, the MSBS cannot notice the startup of spare servers. Therefore, the PSS starts the SIF for the spare server and then the SIF starts the DBSS to manage the spare server. Thereafter, the PSS sends necessary data to manage the spare server via the SIF to the DBSS. If the spare server is unnecessary, the DBSS managing it and the SIF for the DBSS are shut down automatically.



(a) DBSS accesses PSS configuration files

(b) PSS creates spare server

Fig. 10 Functions of SIF.

4. Operation experiment for power-saving, high-availability server system

4.1. Experimental system

The construction of the experimental server system is shown in Fig. 11. The experimental system consisted of a management server for managing the VSG and the RSG, three real servers creating virtual servers, a load balancer, a client, a 1000BASE-T switching hub, and a 100BASE-TX switching hub. The PSS and MSBS were installed on the management server to manage and control the server system. In addition, the management server functions as a file server. In the network environment of the experimental system, the access speed from clients was set to one-tenth the transfer speed between servers, as in a general network system.

The virtual server offered mail, web, and FTP services to the client. Two virtual mail servers (M1 and M2) belonged to the service group mail, two virtual web servers (W1 and W2) belonged to the service group web, and two virtual FTP servers (F1 and F2) belonged to the service group FTP. Each service group was made by virtual servers configured redundantly. In the experimental system, a postfix server program that handles SMTP was installed on the virtual mail server, an httpd server program that handles HTTP was installed on the virtual web server, and a vsftpd server program that handles FTP was installed on the virtual FTP server.

Management data used by the MSBS, the configuration files used by the PSS, and the system program of the SIF were stored on the management server. The operating states of the PSS and MSBS were recorded in a system log file. System data for the spare servers and the backup servers were stored on the management server. Each real server uses the system data saved on the management server to create the spare servers or the backup servers. The virtual servers and the real servers were connected to the management server through a Network File System connection.

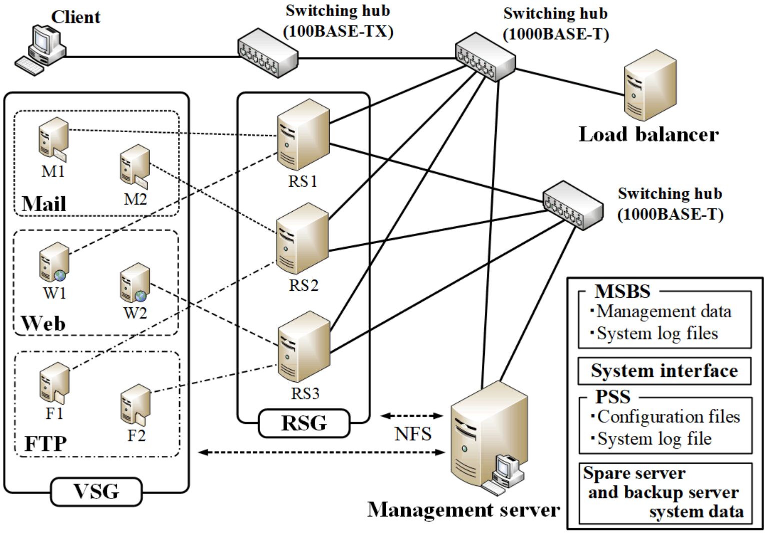


Fig. 11 Configuration of experimental system.

The specifications of the management server, the real server, the virtual server, the backup server, the spare server, and the load balancer in the experimental system are listed in Table 3. CentOS (64-bit), which is often adopted as an operating system (OS) for servers, was installed on the management server, real servers, virtual servers, backup servers, spare servers, and the load balancer. Virtualization was accomplished with Kemel-based Virtual Machine (KVM) software running on the management server and the real servers. Each real server operated with a character-based user interface to use memory effectively.

Table 3 Specifications of experimental system.

Specifications	Management server (PSS, MSBS, and SIF)	Real server (RS1, RS2, and RS3)
CPU	Intel Core i7-3770 3.40 GHz (TB: 3.90 GHz)	
Memory	8,192 MB	
Hard disk	SATA 2 (7,200 rpm)	
System software	KVM 1.5.3 nfsd	KVM 1.5.3
OS	CentOS 7.5 (64-bit)	

Specifications	Virtual, backup, and spare servers
CPU	1
Memory	2,048 MB
System software	postfix, httpd, vsftpd
OS	CentOS 7.5 (64-bit)

Specifications	Load balancer
CPU	Intel Core i3-530 2.93 GHz
Memory	4,096 MB
System software	UltraMonkey-L7
OS	CentOS 6.9 (64-bit)

4.2. Characteristics of experimental system

The characteristics of the experimental system are listed in Table 4. The times for the real, virtual, backup, and spare servers are the average times given by the command “time” in UNIX. “Virtual server” represents the characteristics of the virtual server created by the real server. Note that the startup time for the first virtual server is 25.14 s, and the startup time for the second is 11.34 s. This time lag is assumed to be caused by disk cache processing in the real computer system, because the virtual server is software. “Backup server” represents the characteristics of the backup server created by the management server. Backup servers are created when the MSBS starts up and are suspended until needed. This method shortens the startup time for the backup servers. “Spare server” represents the characteristics of the spare server created by the real server connected to the management server, which is stored the spare server system data, through the Network File System connection. Spare servers are created when the PSS starts up and are suspended until needed.

Table 4 Characteristics of experimental system.

Performances	Real server	Virtual server		Backup server		Spare server	
		First	Second	First	Second	First	Second
Start time (s)	47.39	25.14	11.34	23.49	11.95	47.38	29.08
Restart time (s)	48.01	13.57		14.07		26.57	
Time to suspend active server (s)	2.47			2.45		2.32	
Time to activate suspended server (s)	8.53			0.53		1.09	

Power-saving system		
Load condition (All service groups)	Active real server (Number of active virtual servers)	Power consumption (W) (Total of 3 real servers)
Moderate	RS1, 2, 3 (6)	84
Light (PS1)	RS1, 2 (6)	59
Light (PS2)	RS1 (3)	34

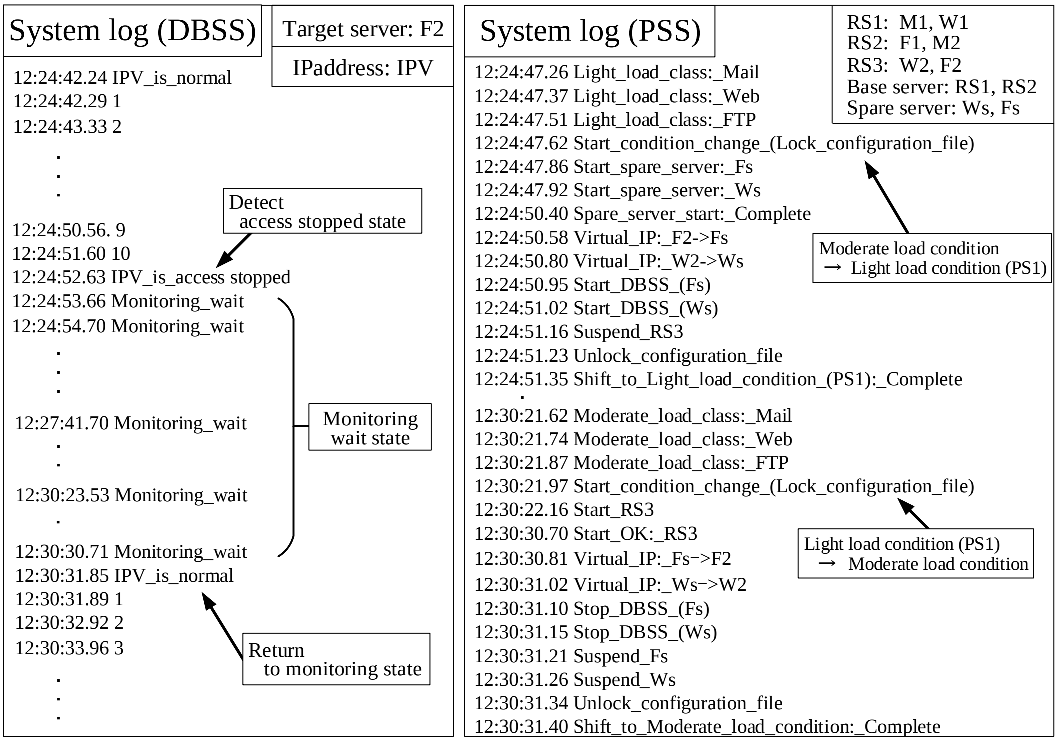
Load condition	Operation for condition change	Required time (s)
Moderate → Light (PS1) (All service groups)	Start two spare servers → Set two virtual IP addresses → Suspend one active real server	3.73
Light (PS1) → Moderate (All service groups)	Activate one suspended real server → Set two virtual IP addresses → Suspend two spare servers	9.43

In the PSS, maximum electric power consumption is 84 W when all three real servers are running with the service groups at moderate load. Because RS1 and RS2 are running (with RS3 suspended) when all of the service groups are in the light load condition (PS1), the power consumption is 59 W (approximately 70% of moderate load consumption). Because only RS1 is running (with RS2 and RS3 suspended) when all of the service groups are in the light load condition (PS2), the minimum power consumption is 34 W (approximately 40% of moderate load consumption). The time required to change the system configuration from the moderate load condition to the light load condition (PS1) is 3.73 s. The time needed to start up the spare server greatly affects this time. The time required to change the system configuration from the light load condition (PS1) to the moderate load condition is 9.43 s. The time needed to activate the suspended real server greatly affects this time.

4.3. Log file indicating operation status

The DBSS and PSS recorded operating status and time in each of their respective system log files. In the DBSS system log, log data were recorded by both the foreground and background jobs. The background job added the string "_ _" to the beginning of each record that it recorded to distinguish background job status in the log data. Figure 12 shows the operating status when the PSS shifted the server configuration between moderate and light load conditions, and Fig. 13 shows the operation status when trouble occurred in the target server. Both figures also show the contents of the log file with comments to highlight important entries.

Figure 12 shows the operating status of the DBSS when the server configuration was changed by the PSS. The target server managed by the DBSS was F2, which has the IP address IPV. The DBSS detected that F2 was in a normal state at 12:24:42.24. While in the moderate load condition, the PSS determined that the loads of all three service groups were light and then initiated the configuration change of the server system at 12:24:47.62. Because the base servers were RS1 and RS2, spare server Fs was created on RS1 and spare server Ws was created on RS2. Then, the VIP addresses for W2 and F2 were reconfigured to Ws and Fs, respectively. Thereafter, RS3 became unnecessary and was suspended. The DBSS detected that F2 was in the access stopped state at 12:24:52.63. Then the PSS determined that the loads of all three service groups were moderate and initiated a configuration change at 12:30:21.97. RS3 was started and the VIP addresses for Ws and Fs were reconfigured to W2 and F2, respectively. The DBSS detected that F2 was in the normal state at 12:30:31.85 and resumed monitoring.



System log (DBSS)

Target server: F2

IPaddress: IPV

```
12:24:42.24 IPV_is_normal
12:24:42.29 1
12:24:43.33 2
      .
      .
      .
12:24:50.56. 9
12:24:51.60 10
12:24:52.63 IPV_is_access stopped
12:24:53.66 Monitoring_wait
12:24:54.70 Monitoring_wait
      .
      .
      .
12:27:41.70 Monitoring_wait
      .
      .
12:30:23.53 Monitoring_wait
      .
12:30:30.71 Monitoring_wait
12:30:31.85 IPV_is_normal
12:30:31.89 1
12:30:32.92 2
12:30:33.96 3
      .
      .
      .
```

System log (PSS)

RS1: M1, W1
RS2: F1, M2
RS3: W2, F2
Base server: RS1, RS2
Spare server: Ws, Fs

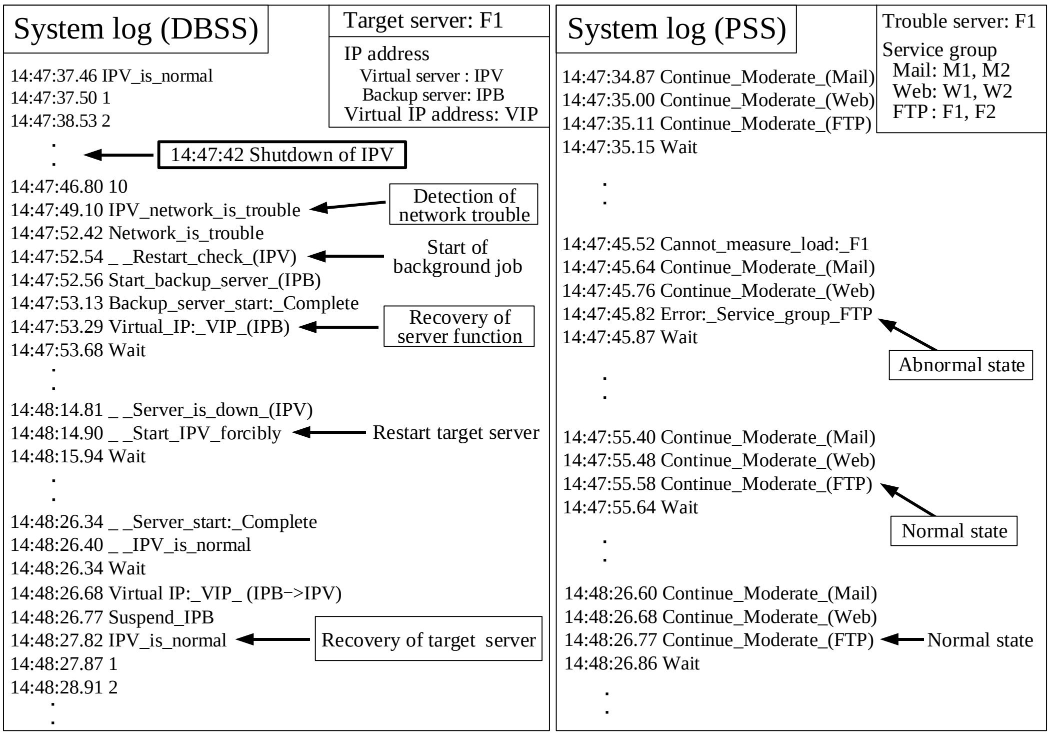
```
12:24:47.26 Light_load_class:_Mail
12:24:47.37 Light_load_class:_Web
12:24:47.51 Light_load_class:_FTP
12:24:47.62 Start_condition_change_(Lock_configuration_file)
12:24:47.86 Start_spare_server:_Fs
12:24:47.92 Start_spare_server:_Ws
12:24:50.40 Spare_server_start:_Complete
12:24:50.58 Virtual_IP:_F2->Fs
12:24:50.80 Virtual_IP:_W2->Ws
12:24:50.95 Start_DBSS_(Fs)
12:24:51.02 Start_DBSS_(Ws)
12:24:51.16 Suspend_RS3
12:24:51.23 Unlock_configuration_file
12:24:51.35 Shift_to_Light_load_condition_(PS1):_Complete
      .
12:30:21.62 Moderate_load_class:_Mail
12:30:21.74 Moderate_load_class:_Web
12:30:21.87 Moderate_load_class:_FTP
12:30:21.97 Start_condition_change_(Lock_configuration_file)
12:30:22.16 Start_RS3
12:30:30.70 Start_OK:_RS3
12:30:30.81 Virtual_IP:_Fs−>F2
12:30:31.02 Virtual_IP:_Ws−>W2
12:30:31.10 Stop_DBSS_(Fs)
12:30:31.15 Stop_DBSS_(Ws)
12:30:31.21 Suspend_Fs
12:30:31.26 Suspend_Ws
12:30:31.34 Unlock_configuration_file
12:30:31.40 Shift_to_Moderate_load_condition:_Complete
```

Moderate load condition
→ Light load condition (PS1)
Light load condition (PS1)
→ Moderate load condition

Fig. 12 Log files of operation status during power saving.

Figure 13 shows that network trouble was reproduced by shutting down F1 at the midpoint (5 s after start) of the monitoring interval. The DBSS system log shows that trouble occurred at 14:47:42 in the monitoring interval time and network trouble was detected at 14:47:49.10. The backup server was started at 14:47:52.56 after reexamination of the network trouble at 14:47:52.42, and startup was completed at 14:47:53.13. The server function was recovered at 14:47:53.29 after assignment of the VIP address. The server function recovery time is the difference between the time at which the server function was recovered and the time at which the network trouble was detected.

In the DBSS system log, the background job started recovery of the target server at 14:47:52.54. The target server was judged to be in a stop condition at 14:48:14.81, and forced restart was attempted. The restart of the target server was completed at 14:48:26.34, and the background job was terminated after the target server was determined to be normal at 14:48:26.40. Thereafter, the target server was judged to be started normally, and the server was recovered at 14:48:26.68 after setting the VIP address. The server recovery time is the difference between the time at which the target server was recovered and the time at which the network trouble was detected. The PSS system log shows that the PSS did not perform the load examination of the service group FTP, because the load of F1 was not measured at 14:47:45.52. Because the server function of F1 was recovered by the DBSS, the service group FTP was in normal state at 14:47:55.58. Although the recovery job for the target server was performed at 14:48:26.68, no abnormality was found in the load examination of the service group FTP.



System log (DBSS)

Target server: F1
IP address
Virtual server : IPV
Backup server: IPB
Virtual IP address: VIP

```
14:47:37.46 IPV_is_normal
14:47:37.50 1
14:47:38.53 2
      .
      .
14:47:46.80 10
14:47:49.10 IPV_network_is_trouble
14:47:52.42 Network_is_trouble
14:47:52.54 _ _Restart_check_(IPV)
14:47:52.56 Start_backup_server_(IPB)
14:47:53.13 Backup_server_start:_Complete
14:47:53.29 Virtual_IP:_VIP_(IPB)
14:47:53.68 Wait
      .
      .
14:48:14.81 _ _Server_is_down_(IPV)
14:48:14.90 _ _Start_IPV_forcibly
14:48:15.94 Wait
      .
      .
14:48:26.34 _ _Server_start:_Complete
14:48:26.40 _ _IPV_is_normal
14:48:26.34 Wait
14:48:26.68 Virtual IP:_VIP_ (IPB−>IPV)
14:48:26.77 Suspend_IPB
14:48:27.82 IPV_is_normal
14:48:27.87 1
14:48:28.91 2
      .
      .
```

System log (PSS)

Trouble server: F1
Service group
Mail: M1, M2
Web: W1, W2
FTP : F1, F2

```
14:47:34.87 Continue_Moderate_(Mail)
14:47:35.00 Continue_Moderate_(Web)
14:47:35.11 Continue_Moderate_(FTP)
14:47:35.15 Wait
      .
      .
14:47:45.52 Cannot_measure_load:_F1
14:47:45.64 Continue_Moderate_(Mail)
14:47:45.76 Continue_Moderate_(Web)
14:47:45.82 Error:_Service_group_FTP
14:47:45.87 Wait
      .
      .
14:47:55.40 Continue_Moderate_(Mail)
14:47:55.48 Continue_Moderate_(Web)
14:47:55.58 Continue_Moderate_(FTP)
14:47:55.64 Wait
      .
      .
14:48:26.60 Continue_Moderate_(Mail)
14:48:26.68 Continue_Moderate_(Web)
14:48:26.77 Continue_Moderate_(FTP)
14:48:26.86 Wait
      .
      .
```

Fig. 13 Log files of operation status during server trouble recovery.

4.4. Experimental results for the case of target server trouble

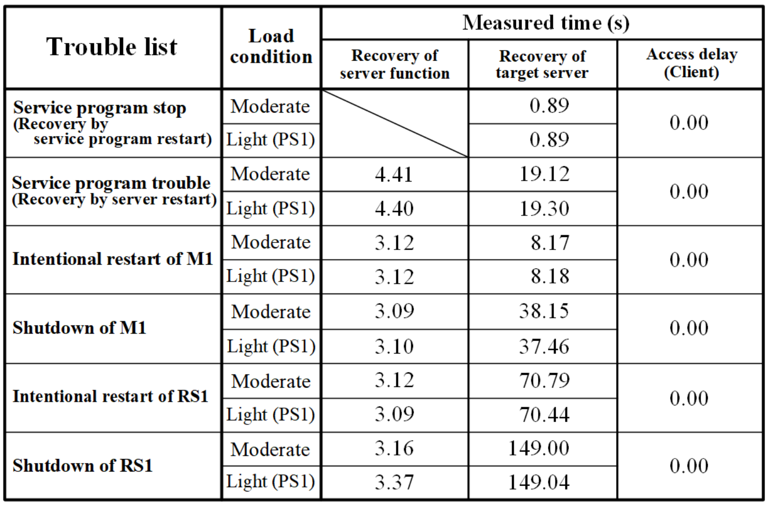
The recovery times in the proposed system and access delays for the client were measured in an experiment in which trouble was intentionally generated in the network or service program on M1 and the network on RS1. The client accesses mail servers through the load balancer, and the results accessed at 1-s intervals by the client are recorded in the client’s access log file together with the time. The recovery times were calculated from log data recorded by the DBSS or the DSRS, and the access delay was measured based on the client access log. Each reported time is the average of multiple experiments. The recovery times and access delays for the case in which trouble occurs on M1 or RS1 under the moderate load condition and the light load condition (PS1) are listed in Table 5. The time required to recover the server function, time required to return the target server to the normal state, and the client access delay are indicated as “Measured time.” “Recovery of server function” indicates the time until the backup server assumed the server function after trouble occurred, and “Recovery of target server” indicates the time until the DBSS or the DSRS returned the target server to the normal state from the trouble state, as shown in Fig. 9. The monitoring interval time was 10 s in the experimental system, and trouble was generated at the midpoint (5 s after startup) of the monitoring interval.

We used two methods to generate trouble in the service program: service program stop and service program freeze requiring server restart for recovery. In service program stop, the time until the target server returned to the normal state was the same under both moderate and light loads and server recovery took less than 1 s. The service program freeze trouble had the longest server function recovery times among all the tests reported in Table 5. Target server recovery required less than 20 s and was similar for both load conditions.

We used two methods to generate trouble in the virtual server network: intentional restart and shutdown of M1. The server function recovery time was approximately 3 s for both tests under both load conditions. The time until the target server returned to the normal state was approximately 8 s for the intentional restart test under both load conditions. For the server shutdown test, the server recovery times were less than 39 s.

We used two methods to generate trouble in the real server network: intentional restart and shutdown of RS1. Again, the server function recovery time for both test and load conditions was approximately 3 s. In the intentional restart of RS1, the time until the server system returned to the normal state was the second-highest among all the tests and similar for both load conditions. In the shutdown of RS1, the time until the server system returned to the normal state was the longest among all test recovery times. For all the tests, no delay of access by the client occurred under both moderate and light load conditions because servers providing services to the client were configured redundantly.

Table 5 Recovery time of proposed system for different types of trouble.



Trouble list	Load condition	Measured time (s)		
		Recovery of server function	Recovery of target server	Access delay (Client)
Service program stop (Recovery by service program restart)	Moderate		0.89	0.00
	Light (PS1)		0.89	
Service program trouble (Recovery by server restart)	Moderate	4.41	19.12	0.00
	Light (PS1)	4.40	19.30	
Intentional restart of M1	Moderate	3.12	8.17	0.00
	Light (PS1)	3.12	8.18	
Shutdown of M1	Moderate	3.09	38.15	0.00
	Light (PS1)	3.10	37.46	
Intentional restart of RS1	Moderate	3.12	70.79	0.00
	Light (PS1)	3.09	70.44	
Shutdown of RS1	Moderate	3.16	149.00	0.00
	Light (PS1)	3.37	149.04	

5. Conclusions

In this study, we adopted the compound operation of a MSBS and PSS that are operated independently. We examined the problems encountered during compound operation and developed solutions to them. The key solutions were to control the accessing or editing of the PSS configuration files by the MSBS and to control the startup of the DBSS, which manages the spare server, by the PSS. Therefore, we introduced the SIF that has functions for bi-directional control of the PSS and MSBS. As a result, their control programs, which were enhanced functions, did not need to be made more complicated to achieve the combined operation of the PSS and MSBS. An experimental power-saving, high-availability server system incorporating the proposed method was constructed and the times to recover the target server function and target server were measured for various troubles under both moderate and light load conditions (power saving is implemented during light load condition). The proposed system was able to rapidly recover the server function and target server under all the scenarios tested. For all the tests, the client experienced no access delay under both moderate and light load conditions. The results of this study indicate that both power saving and high server availability can be realized effectively. Therefore, expansion of Internet services affording greater convenience in our daily lives can be expected.

References

[1] Ministry of Internal Affairs and Communications
http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/html/nc121100.html, 2017.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] G. Wang, L. Zhang, and W. Xu, "What Can We Learn from Four Years of Data Center Hardware Failures?," IEEE Conference Proc., Vol. 2017, No. DAN, 2017, pp. 25-36.

[8] W.-L. YEOW, C. WESTPHAL, and U.-C. KOZAT, "Highly Available Virtual Machines with Network Coding," Proc. IEEE INFCOM, Vol. 1, 2011, pp. 386-390.

[9] Q. Zhang, M.-F. Zhani, M. Jabri, and R. Boutaba, "Venice: Reliable Virtual Data Center Embedding in Clouds," Proc. IEEE INFOCOM 2014, Vol. 1, 2014, pp. 289-297.

[10] 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.

[11] W. Shengquan, L. Jun, C. Jian-Jia, and L. Xue, "Power Sleep: A smart Power-Saving Scheme with Sleep for Servers under Response Time Constraint", IEEE J. Emerg. Sel. Top Circuit Syst., Vol. 1 No. 3, 2011, pp. 289-298.

[12] Ministry of Economy, Trade and Industry,
http://www.meti.go.jp/committee/materials/downloadfiles/g80520c03j.pdf, 2008.

[13] 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.

[14] L. Saulo O. D., P. Angelo, C. Bruna M. J., N. Breno H. M., and A. Gabriela M. da S., "Optimization of Timeout-based Power Management Policies for Network Interfaces," IEEE Trans. Consum. Electron. Vol. 59, No. 1, 2013, pp. 101-106.

[15] 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

[16] T. Pan, T. Zhang, J. Shi, Y.Li, L. Jin, F. Li, J. Yang, B. Zhang, and B. Liu, "Towards Zero-Time Wakeup of Line Cards in Power-Aware Routers," Proc. IEEE INFOCOM 2014, Vol. 1, 2014, pp. 190-198.

[17] Y. Fukushima, T. Murase, and T. Yokohira, "Energy-Aware Optimal Server Location Decision in Server Migration Service," IEICE Technical Report, Vol. 116, No. 428, 2017, pp. 53-58.

[18] 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.

[19] 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.

[20] M. Kitamura, Y. Shimizu, and K. Tani, "Development of a Power-saving, High-availability Server System by Compound Operation of a Multiple-server Backup System and Power-saving Server System," JISSJ, Vol. 14, No. 2, 2019, pp. 79-88.

[21] M. Kitamura, Y. Udagawa, H. Nakagome, 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.

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 M.E. degree in 2019 from Tokyo Polytechnic University, where he is currently in charge of implementation and analysis of experimental hardware systems at Hitachi Information & Telecommunication Engineering, Ltd.

Koki TANI

He received his B.E. degree in 2018 from Tokyo Polytechnic University, where he is currently working toward the M.E. degree. He is engaged in developing software for experimental systems and analyzing performance data.