INFORMATION PROCESSING APPARATUS AND METHOD OF OPERATING THE SAME

Inventors: Takuya Oda, Yokohama (JP); Daisuke Iizuka, Yokohama (JP)

Assignee: Hitachi, Ltd., Tokyo (JP)

Appl. No.: 13/304,836

Filed: Nov. 28, 2011

Related U.S. Application Data
Continuation of application No. 12/393,142, filed on Feb. 26, 2009, now Pat. No. 8,073,938.

Foreign Application Priority Data
Nov. 25, 2008 (JP) ...................... 2008-299972

ABSTRACT
Using a service server list containing addresses of servers within a service system and an application list containing names of active applications within the service system, addresses of servers on which the active applications are running, and communication source addresses and communication destination addresses of the active applications, the following processes are carried out: specifying applications of a first layer in the service system from among applications lying within the service system; specifying applications of a second or subsequent layer in the service system from the application list and from addresses of servers on which the applications of the first layer are running; and judging applications having the same application name among applications belonging to the same layer as being of the same role for aggregation.
## SERVICE EXECUTION SERVER LIST

<table>
<thead>
<tr>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.2.1</td>
</tr>
<tr>
<td>192.168.2.2</td>
</tr>
<tr>
<td>192.168.2.3</td>
</tr>
<tr>
<td>192.168.2.4</td>
</tr>
<tr>
<td>192.168.2.5</td>
</tr>
<tr>
<td>192.168.2.6</td>
</tr>
<tr>
<td>APPLICATION LIST</td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>APPLICATION #1</td>
</tr>
<tr>
<td>APPLICATION #2</td>
</tr>
<tr>
<td>APPLICATION #3</td>
</tr>
<tr>
<td>APPLICATION #4</td>
</tr>
<tr>
<td>APPLICATION #5</td>
</tr>
<tr>
<td>APPLICATION #6</td>
</tr>
<tr>
<td>APPLICATION #7</td>
</tr>
</tbody>
</table>

FIG. 4
### APPLICATION ROLE CONFIGURATION LIST

<table>
<thead>
<tr>
<th>APPLICATION IDENTIFIER</th>
<th>LAYER IDENTIFIER</th>
<th>ROLE IDENTIFIER</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION #1</td>
<td>LAYER #1</td>
<td>ROLE #1</td>
</tr>
<tr>
<td>APPLICATION #2</td>
<td>LAYER #1</td>
<td>ROLE #1</td>
</tr>
<tr>
<td>APPLICATION #3</td>
<td>LAYER #2</td>
<td>ROLE #2</td>
</tr>
<tr>
<td>APPLICATION #4</td>
<td>LAYER #2</td>
<td>ROLE #3</td>
</tr>
<tr>
<td>APPLICATION #5</td>
<td>LAYER #2</td>
<td>ROLE #3</td>
</tr>
<tr>
<td>APPLICATION #6</td>
<td>LAYER #3</td>
<td>ROLE #4</td>
</tr>
</tbody>
</table>

**FIG. 5**

### AGGREGATED APPLICATION LIST

<table>
<thead>
<tr>
<th>ROLE IDENTIFIER</th>
<th>APPLICATION NAME</th>
<th>NUMBER OF APPLICATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLE #1</td>
<td>APPLICATION A</td>
<td>2</td>
</tr>
<tr>
<td>ROLE #2</td>
<td>APPLICATION B</td>
<td>1</td>
</tr>
<tr>
<td>ROLE #3</td>
<td>APPLICATION C</td>
<td>2</td>
</tr>
<tr>
<td>ROLE #4</td>
<td>APPLICATION D</td>
<td>1</td>
</tr>
</tbody>
</table>

**FIG. 6**
### Aggregated Application Communication List

<table>
<thead>
<tr>
<th>Communication Source Role Identifier</th>
<th>Communication Destination Role Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLE #1</td>
<td>ROLE #2</td>
</tr>
<tr>
<td>ROLE #1</td>
<td>ROLE #3</td>
</tr>
<tr>
<td>ROLE #2</td>
<td>ROLE #4</td>
</tr>
<tr>
<td>ROLE #3</td>
<td>ROLE #4</td>
</tr>
</tbody>
</table>

**FIG. 7**

### Service System List

<table>
<thead>
<tr>
<th>Application Identifier</th>
<th>Service System Name</th>
<th>Role Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION #1</td>
<td>ORDER MANAGEMENT SYSTEM</td>
<td>ROLE #1</td>
</tr>
<tr>
<td>APPLICATION #2</td>
<td>CUSTOMER MANAGEMENT SYSTEM</td>
<td>ROLE #2</td>
</tr>
<tr>
<td>APPLICATION #3</td>
<td>ORDER MANAGEMENT SYSTEM</td>
<td>ROLE #3</td>
</tr>
<tr>
<td>APPLICATION #3</td>
<td>CUSTOMER MANAGEMENT SYSTEM</td>
<td>ROLE #4</td>
</tr>
<tr>
<td>APPLICATION #4</td>
<td>ORDER MANAGEMENT SYSTEM</td>
<td>ROLE #5</td>
</tr>
<tr>
<td>APPLICATION #4</td>
<td>CUSTOMER MANAGEMENT SYSTEM</td>
<td>ROLE #6</td>
</tr>
<tr>
<td>APPLICATION #5</td>
<td>ORDER MANAGEMENT SYSTEM</td>
<td>ROLE #5</td>
</tr>
<tr>
<td>APPLICATION #5</td>
<td>CUSTOMER MANAGEMENT SYSTEM</td>
<td>ROLE #6</td>
</tr>
<tr>
<td>APPLICATION #6</td>
<td>ORDER MANAGEMENT SYSTEM</td>
<td>ROLE #7</td>
</tr>
<tr>
<td>APPLICATION #6</td>
<td>CUSTOMER MANAGEMENT SYSTEM</td>
<td>ROLE #8</td>
</tr>
</tbody>
</table>

**FIG. 8**
### APPLICATION CONFIGURATION BASELINE

<table>
<thead>
<tr>
<th>APPLICATION IDENTIFIER</th>
<th>SERVER ADDRESS</th>
<th>APPLICATION NAME</th>
<th>COMMUNICATION SOURCE ADDRESS</th>
<th>COMMUNICATION DESTINATION ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION #1</td>
<td>192.168.2.1</td>
<td>APPLICATION A</td>
<td>128.241.23.25</td>
<td>192.168.2.4</td>
</tr>
<tr>
<td>APPLICATION #2</td>
<td>192.168.2.2</td>
<td>APPLICATION A</td>
<td>128.241.23.25</td>
<td>192.168.2.4</td>
</tr>
<tr>
<td>APPLICATION #4</td>
<td>192.168.2.4</td>
<td>APPLICATION C</td>
<td>192.168.2.1</td>
<td>192.168.2.7</td>
</tr>
<tr>
<td>APPLICATION #4</td>
<td>192.168.2.4</td>
<td>APPLICATION C</td>
<td>192.168.2.2</td>
<td>192.168.2.7</td>
</tr>
<tr>
<td>APPLICATION #7</td>
<td>192.168.2.7</td>
<td>APPLICATION D</td>
<td>192.168.2.4</td>
<td></td>
</tr>
</tbody>
</table>

### APPLICATION CONFIGURATION CHANGE LIST

<table>
<thead>
<tr>
<th>APPLICATION IDENTIFIER</th>
<th>CONTENTS OF CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION #3</td>
<td>APPLICATION STARTUP</td>
</tr>
<tr>
<td>APPLICATION #5</td>
<td>APPLICATION STARTUP</td>
</tr>
<tr>
<td>APPLICATION #6</td>
<td>APPLICATION STARTUP</td>
</tr>
<tr>
<td>APPLICATION #7</td>
<td>APPLICATION STOP</td>
</tr>
</tbody>
</table>

### SETTING CHANGE APPLY LIST

<table>
<thead>
<tr>
<th>APPLICATION IDENTIFIER</th>
<th>CONTENTS OF SETTING</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION #1</td>
<td>CHANGE SETTING ITEM Y1 OF SETTING FILE X1 TO Z1</td>
</tr>
<tr>
<td>APPLICATION #2</td>
<td>DELETE SETTING ITEM Y2 OF SETTING FILE X2</td>
</tr>
</tbody>
</table>
### PHYSICAL/LOGICAL (VIRTUAL) SERVER LIST

<table>
<thead>
<tr>
<th>PHYSICAL SERVER ADDRESS</th>
<th>LOGICAL (VIRTUAL) SERVER ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.2.1</td>
<td></td>
</tr>
<tr>
<td>192.168.2.2</td>
<td></td>
</tr>
<tr>
<td>192.168.2.10</td>
<td>192.168.2.3</td>
</tr>
<tr>
<td>192.168.2.10</td>
<td>192.168.2.4</td>
</tr>
<tr>
<td>192.168.2.5</td>
<td></td>
</tr>
<tr>
<td>192.168.2.6</td>
<td></td>
</tr>
</tbody>
</table>

**FIG. 12**

### SHARED APPLICATION LIST

<table>
<thead>
<tr>
<th>ROLE IDENTIFIER</th>
<th>NUMBER OF SHARED APPLICATIONS</th>
<th>TYPE OF SHARING</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLE #2</td>
<td>2</td>
<td>PHYSICAL SERVER SHARING</td>
</tr>
<tr>
<td>ROLE #3</td>
<td>2</td>
<td>PHYSICAL SERVER SHARING</td>
</tr>
<tr>
<td>ROLE #4</td>
<td>1</td>
<td>PLURAL BUSINESS SYSTEMS SHARING</td>
</tr>
</tbody>
</table>

**FIG. 13**
START

APPLICATION ROLE SPECIFYING UNIT EXECUTES PROCEDURE DESCRIBED IN FIG. 16 TO SPECIFY APPLICATIONS BELONGING TO FIRST LAYER.

APPLICATION ROLE SPECIFYING UNIT PERFORMS PROCEDURE DESCRIBED IN FIG. 17 TO SPECIFY APPLICATIONS OF SECOND OR SUBSEQUENT LAYER.

APPLICATION ROLE SPECIFYING UNIT PERFORMS PROCEDURE DESCRIBED IN FIG. 18 TO CLASSIFY APPLICATIONS INTO A PLURALITY OF ROLES.

APPLICATION AGGREGATING UNIT REFERS TO APPLICATION ROLE CONFIGURATION LIST TO FIND THE NUMBER OF APPLICATIONS BELONGING TO EACH ROLE AND STORES THE NUMBERS TOGETHER WITH APPLICATION NAMES INTO AGGREGATED APPLICATION LIST.

APPLICATION AGGREGATING UNIT PERFORMS PROCEDURE DESCRIBED IN FIG. 19 TO SPECIFY COMMUNICATION SOURCE AND COMMUNICATION DESTINATION OF AGGREGATED APPLICATIONS.

APPLICATION CONFIGURATION DIAGRAM CREATING UNIT CREATES APPLICATION CONFIGURATION DIAGRAM FROM AGGREGATED APPLICATION LIST AND AGGREGATED APPLICATION COMMUNICATION LIST.

END

FIG. 15
APPLICATION ROLE SPECIFYING UNIT ACQUIRES LIST APLIST2 OF APPLICATION IDENTIFIERS DESCRIBED IN APPLICATION LIST.

APLIST2 IS NOT EMPTY.

APPLICATION ROLE SPECIFYING UNIT EXTRACTS CERTAIN APPLICATION IDENTIFIER APP2 IN APLIST2 (AND DELETS APP2 FROM APLIST2) AND REFERS TO APPLICATION LIST TO ACQUIRE COMMUNICATION SOURCE ADDRESS ADD3 OF APP2.

APPLICATION ROLE SPECIFYING UNIT READS APPLICATION ROLE CONFIGURATION LIST AND APPLICATION LIST AND ACQUIRES ADDRESS LIST ADLST2 OF SERVERS ON WHICH APPLICATIONS RUN THAT HAVE LAYER IDENTIFIER (LAYER #i-1) INDICATIVE OF LAYER ONE-PREVIOUS TO LAYER (I-TH LAYER) TO BE SPECIFIED.

ADD3 IS CONTAINED IN ADLST2.

APPLICATION ROLE SPECIFYING UNIT DETERMINES THAT APPLICATIONS CORRESPONDING TO APP2 TO BE APPLICATIONS OF I-TH LAYER AND STORES APPLICATION IDENTIFIER APP2 AND LAYER IDENTIFIER (LAYER #i) INTO APPLICATION ROLE CONFIGURATION LIST.

FIG. 17
APPLICATION AGGREGATING UNIT EXTRACTS CERTAIN APPLICATION IDENTIFIER APP5 FROM APPLIST5 (AND DELETES APP5 FROM APPLIST5) TO ACQUIRE COMMUNICATION DESTINATION ADDRESS ADDR5 FROM APPLICATION LIST.

APPLICATION AGGREGATING UNIT ACQUIRES FROM APPLICATION LIST APPLICATION IDENTIFIERS APP6 OF APPLICATIONS RUNNING ON SERVER HAVING ADDRESS INDICATED BY ADDR5.

APPLICATION AGGREGATING UNIT ACQUIRES ROLE IDENTIFIERS (ROLE #N) OF APPLICATIONS CORRESPONDING TO APP6 FROM APPLICATION ROLE CONFIGURATION LIST.

NO COMMUNICATION EXISTS WHOSE COMMUNICATION SOURCE ROLE IDENTIFIER IS ROLE #M AND WHOSE COMMUNICATION DESTINATION ROLE IDENTIFIER IS ROLE #N IN AGGREGATED APPLICATION COMMUNICATION LIST.

APPLICATION AGGREGATING UNIT STORES ROLE #M AND ROLE #N INTO COMMUNICATION SOURCE ROLE IDENTIFIER COLUMN AND COMMUNICATION DESTINATION ROLE IDENTIFIER COLUMN, RESPECTIVELY, OF AGGREGATED APPLICATION COMMUNICATION LIST.

FIG. 20
APPLICATION ROLE SPECIFYING UNIT EXECUTES PROCESSES OF S1401 AND S1402 DESCRIBED IN FIG. 15 TO SPECIFY LAYERS TO WHICH APPLICATIONS BELONG.

SERVICE SYSTEM SPECIFYING UNIT PERFORMS PROCEDURE DESCRIBED IN FIG. 24 TO SPECIFY SERVICE SYSTEMS TO WHICH APPLICATIONS OF FIRST LAYER BELONG.

SERVICE SYSTEM SPECIFYING UNIT SPECIFIES SERVICE SYSTEMS ON WHICH APPLICATIONS OF SECOND OR SUBSEQUENT LAYER RUN, BASED ON COMMUNICATION DESTINATION ADDRESSES OF APPLICATIONS OF THE FIRST LAYER.

APPLICATION AGGREGATING UNIT ACQUIRES LIST APLIST10 OF APPLICATION IDENTIFIERS DESCRIBED IN BUSINESS SYSTEM LIST.

APLIST10 IS EMPTY.

Yes

APPLICATION AGGREGATING UNIT EXTRACTS CERTAIN APPLICATION IDENTIFIER APP10 DESCRIBED IN APLIST10 AND DELETES APP10 FROM APLIST10.

No

APPLICATION AGGREGATING UNIT ACQUIRES FROM SERVICE SYSTEM LIST, APPLICATION IDENTIFIER LIST APLIST11 (INCLUDING APP10) INDICATIVE OF SAME SERVICE SYSTEM NAME (GSNAME10) AS APP10 AND DELETES APLIST11 FROM APLIST10.

APPLICATION AGGREGATING UNIT APPLIES PROCESSES OF STEPS 1403 TO 1406 DESCRIBED IN FIG. 15 TO APPLICATIONS HAVING IDENTIFIERS DESCRIBED IN APLIST11, TO CREATE APPLICATION CONFIGURATION DIAGRAM OF SERVICE SYSTEM (GSNAME10). THE CURRENT ROLE NUMBER K AT STEP 1702 IS DEFINED AS CONSECUTIVE NUMBER (UNIQUE) FOLLOWING THE LAST NUMBER IN PRECEDING SERVICE SYSTEM AND IDENTIFIERS OF ROLES CLASSIFIED AT STEP 1403 ARE STORED ALSO IN SERVICE SYSTEM LIST.

FIG. 23
START

SERVICE SYSTEM SPECIFYING UNIT ACQUIRES APPLICATION IDENTIFIER LIST APLIST10 OF APPLICATIONS OF FIRST LAYER (HAVING ROLE IDENTIFIER OF LAYER #1) FROM APPLICATION ROLE CONFIGURATION LIST.

APLIST10 IS EMPTY.

Yes

SERVICE SYSTEM SPECIFYING UNIT STORES "APPL10 AND "SERVICE SYSTEM UNSPECIFIED" INTO SERVICE SYSTEM LIST.

No

SERVICE SYSTEM SPECIFYING UNIT EXTRACTS CERTAIN APPLICATION IDENTIFIER APP10 FROM APLIST10 (AND DELETES APP10 FROM APLIST10), TO ACQUIRE ADDRESS ADD10 OF SERVER ON WHICH APPLICATION INDICATED BY APP10 RUNS FROM APPLICATION LIST.

SERVICE SYSTEM SPECIFYING UNIT PERFORMS HTTP COMMUNICATION WITH SERVER INDICATED BY ADD10.

COMMUNICATION HAS SUCCEEDED.

Yes

SERVICE SYSTEM SPECIFYING UNIT APPLIES DOM ANALYSIS TO SOURCE CODE OF WEB PAGE DISPLAYED.

No

TITLE NODE EXISTS.

Yes

BUSINESS SYSTEM SPECIFYING UNIT STORES "APP10" AND "CONTENTS OF TITLE NODE" INTO BUSINESS SYSTEM LIST.

No

END
APPLICATION AGGREGATING UNIT ACQUIRES APLIST11 OF APPLICATION IDENTIFIERS DESCRIBED IN SERVICE SYSTEM LIST.

APLIST11 IS EMPTY.

APPLICATION AGGREGATING UNIT EXTRACTS CERTAIN APPLICATION IDENTIFIER APP11 FROM APLIST11 (AND DELETES APP11 FROM APLIST11), TO ACQUIRE ROLE IDENTIFIER LIST ROLELIST11 CORRESPONDING TO THE CERTAIN APPLICATION IDENTIFIER FROM SERVICE SYSTEM LIST.

ROLELIST11 IS NOT EMPTY.

APPLICATION AGGREGATING UNIT EXTRACTS CERTAIN ROLE IDENTIFIER ROLE11 WITHIN ROLELIST11 (AND DELETES ROLE11 FROM ROLELIST11).

APPLICATION AGGREGATING UNIT REFERS TO SHARED APPLICATION LIST TO ACQUIRE SHARED APPLICATION IDENTIFIER SHAREAPP11 CORRESPONDING TO APPLICATIONS HAVING ROLE IDENTIFIER OF ROLE11 AND TYPE-OF-SHARING INDICATIVE OF SHARING A PLURALITY OF SERVICE SYSTEMS.

SHAREAPP11 IS PRESENT.

APPLICATION AGGREGATING UNIT INCREMENTS THE NUMBER-OF-SHARED-APPLICATIONS OF APPLICATIONS HAVING SHARED APPLICATION IDENTIFIER OF SHAREAPP11 WITHIN SHARED APPLICATION LIST.

APPLICATION AGGREGATING UNIT STORES INTO SHARED APPLICATION LIST ROLE IDENTIFIER: ROLE11, NUMBER-OF-SHARED-APPLICATIONS: 1, TYPE-OF-SHARING: CONTENTS INDICATIVE OF SHARING A PLURALITY OF SERVICE SYSTEMS.

FIG. 25
START

APPLICATION AGGREGATING UNIT REFERS TO PHYSICAL/LOGICAL(VIRTUAL) SERVER LIST TO ACQUIRE LIST SRVADDLIST12 OF LOGICAL/VIRTUAL SERVER ADDRESSES INDICATIVE OF SAME PHYSICAL SERVER ADDRESS.

S2301

SRVADDLIST12 IS EMPTY.

No

APPLICATION AGGREGATING UNIT EXTRACTS CERTAIN ADDRESS ADD12 FROM SRVADDLIST12 (AND DELETES ADD12 FROM SRVADDLIST12), TO ACQUIRE FROM APPLICATION LIST AN APPLICATION IDENTIFIER LIST APPLIST12 OF APPLICATIONS RUNNING ON SERVER INDICATED BY THE CERTAIN ADDRESS.

S2302

Yes

END

APPLICATION AGGREGATING UNIT EXTRACTS CERTAIN APPLICATION IDENTIFIER APP12 FROM APPLIST12 (AND DELETES APP12 FROM APPLIST12) AND REFERS TO APPLICATION ROLE CONFIGURATION LIST TO ACQUIRE ROLE ROLE12 CORRESPONDING TO APP12.

S2303

S2304

APPLIST12 IS EMPTY.

No

APPLICATION AGGREGATING UNIT EXTRACTS CERTAIN APPLICATION IDENTIFIER SHAREAPP12 CORRESPONDING TO APPLICATIONS HAVING ROLE IDENTIFIER OF ROLE12 AND THE TYPE-OF-SHARING OF SHARING VIRTUAL SERVERS.

S2305

S2306

APPLICATION AGGREGATING UNIT REFERENCES TO SHARED APPLICATION LIST TO ACQUIRE SHARED APPLICATION IDENTIFIER SHAREAPP12 WITHIN APPLICATION LIST.

S2307

SHAREAPP12 IS PRESENT.

Yes

APPLICATION AGGREGATING UNIT INCREASES THE NUMBER-OF-SHARED-APPLICATIONS OF APPLICATIONS HAVING SHARED APPLICATION IDENTIFIER OF SHAREAPP12 WITHIN SHARED APPLICATION LIST.

S2308

No

APPLICATION AGGREGATING UNIT STORES INTO SHARED APPLICATION LIST ROLE IDENTIFIER ROLE12, NUMBER-OF-SHARED-APPLICATIONS, AND TYPE-OF-SHARING CONTENTS INDICATIVE OF SHARING OF VIRTUAL SERVERS.

S2209

FIG. 26
ORIGINAL PROPER (NORMAL) APPLICATION CONFIGURATION SHOULD BE AS FOLLOWS:

APPLICATION A

APPLICATION B x40

APPLICATION C x32

APPLICATION D x1

NEVERTHELESS, CURRENT APPLICATION CONFIGURATION IS AS FOLLOWS:

APPLICATION A

APPLICATION B x40

APPLICATION C x40

APPLICATION D x1

PROCEDURE OF THIRD EMBODIMENT ENABLES AGGREGATING DISPLAYS DEPENDING ON CONTENTS OF CHANGES FROM NORMAL CONFIGURATION:

APPLICATION B

APPLICATION Startup

APPLICATION STOP

APPLICATION A

APPLICATION C

APPLICATION D

NO CHANGE

x2

x20

x40

x32

FIG. 27
START

S2401

CONFIGURATION CHANGE MANAGING UNIT READS INTO LISTA LISTS OF APPLICATION IDENTIFIERS AND COMMUNICATION SOURCE/COMMUNICATION DESTINATION ADDRESSES OF APPLICATION CONFIGURATION BASELINE AND READS INTO LISTB LISTS OF APPLICATION IDENTIFIERS AND COMMUNICATION SOURCE/COMMUNICATION DESTINATION ADDRESSES OF CURRENT APPLICATION LIST.

S2402

LISTB IS NOT EMPTY.

Yes

S2403

CONFIGURATION CHANGE MANAGING UNIT EXECUTES PROCESS A AND PROCESS B OF FIG. 29.

No

S2404

CONFIGURATION CHANGE MANAGING UNIT READS LISTS OF APPLICATION IDENTIFIERS AND COMMUNICATION SOURCE/COMMUNICATION DESTINATION ADDRESSES OF CURRENT APPLICATION LIST INTO LISTA AND READS LISTS OF APPLICATION IDENTIFIERS AND COMMUNICATION SOURCE/COMMUNICATION DESTINATION ADDRESSES OF APPLICATION CONFIGURATION BASELINE INTO LISTB.

S2405

LISTB IS NOT EMPTY.

Yes

S2406

CONFIGURATION CHANGE MANAGING UNIT EXECUTES ONLY PROCESS A OF FIG. 29.

No

END

FIG. 28
START OF PROCESS A

S2501

CONFIGURATION CHANGE MANAGING UNIT EXTRACTS CERTAIN APPLICATION IDENTIFIER APP13 FROM LISTB (AND DELETES APP13 FROM LISTB).

S2502

APP13 IS NOT PRESENT IN LISTA.

NO

S2503

LISTA HAS BEEN ACQUIRED FROM APPLICATION CONFIGURATION Baseline.

NO

S2504

CONFIGURATION CHANGE MANAGING UNIT SAVES IDENTIFIER: APP13 AND CONTENTS-OF-CHANGE: "STARTUP OF APPLICATION" IN APPLICATION CONFIGURATION CHANGE LIST.

S2505

CONFIGURATION CHANGE MANAGING UNIT SAVES IDENTIFIER: APP13 AND CONTENTS-OF-CHANGE: "STOP OF APPLICATION" IN APPLICATION CONFIGURATION CHANGE LIST.

END OF PROCESS A

S2506

PROCESS B IS ALSO EXECUTED.

NO

S2507

START OF PROCESS B

END OF PROCESS A

S2508

ADDR13 AND ADDR15 DIFFER.

NO

S2509

CONFIGURATION CHANGE MANAGING UNIT SAVES IDENTIFIER: APP13 AND CONTENTS-OF-CHANGE: "CHANGE OF COMMUNICATION SOURCE" IN APPLICATION CONFIGURATION CHANGE LIST.

S2510

ADDR14 AND ADDR16 DIFFER.

NO

S2511

CONFIGURATION CHANGE MANAGING UNIT SAVES IDENTIFIER: APP13 AND CONTENTS-OF-CHANGE: "CHANGE OF COMMUNICATION DESTINATION" IN APPLICATION CONFIGURATION CHANGE LIST.

END OF PROCESS B

FIG. 29
-setting change managing unit accepts change request including role identifier role14 and setting contents change14 from system administrator.

-setting change managing unit refers to application role configuration list to acquire identifier list applist14 of applications having role identifier of role14 and stores application identifiers of applist14 and change14 into setting change apply list.

-setting change apply list is not empty.

-setting change managing unit extracts certain application identifier app17 and change contents change15 described in setting change apply list (and deletes app17 and change15 from setting change apply list).

-setting change managing unit acquires address addr17 of server on which application app17 runs from application list.

-setting change managing unit sends change15 to addr17.

-setting managing unit of business execution computer receives change15 to execute setting change process described in change15.

FIG. 30
START

SERVICE SYSTEM SPECIFYING UNIT ACQUIRES APPLICATION IDENTIFIER LIST APLIST15 FROM APPLICATION ROLE CONFIGURATION LIST.

S2701

APLIST15 IS EMPTY.

S2702

Yes

SERVICE SYSTEM SPECIFYING UNIT EXTRACTS CERTAIN APPLICATION IDENTIFIER APP18 FROM APLIST15 (AND DELETES APP18 FROM APLIST15) TO ACQUIRE ADDRESS ADD18 OF SERVER ON WHICH APP18 RUNS FROM APPLICATION LIST.

S2703

No

SERVICE SYSTEM SPECIFYING UNIT SENDS TO ADD18 REQUEST TO ACQUIRE LIST FILELIST18 OF WAR/EAR/JAR FILES STORED IN SERVER OF ADD18.

S2704

ACQUISITION OF FILE LIST HAS SUCCEEDED.

S2705

Yes

SERVICE SYSTEM SPECIFYING UNIT STORES APP18 AND FILELIST18 INTO SERVICE SYSTEM LIST.

S2706

Within Service System List There Exists Application Identifier APP19 Coincident With Contents of WAR/EAR/JAR File List.

S2707

No

SERVICE SYSTEM SPECIFYING UNIT STORES THE SAME ROLE IDENTIFIER AS THAT OF APP19 INTO ROLE IDENTIFIER CORRESPONDING TO APPLICATION IDENTIFIER OF APP18 WITHIN SERVICE SYSTEM LIST.

S2708

Yes

The Service System Specifying Unit Adds And Stores New Role Identifier Into Role Identifier Corresponding To Application Identifier Of APP18 Within Service System List.

S2709

FIG. 31
APPLICATION ROLE SPECIFYING UNIT EXECUTES PROCESS S1701 DESCRIBED IN FIG. 18.

APPLICATION ROLE SPECIFYING UNIT EXECUTES PROCESS S1702 DESCRIBED IN FIG. 18.

APLIST2 IS EMPTY.

APPLiST2 IS EMPTY.

APPLICATION ROLE SPECIFYING UNIT EXECUTES PROCESSES S1704 TO S1706 DESCRIBED IN FIG. 18.

APPLICATION ROLE SPECIFYING UNIT ACQUIRES COMMUNICATION DESTINATION LIST ADDRiLIST20 OF APLLiST4 FROM APPLICATION LIST.

APPLiST4 IS NOT EMPTY.

APPLICATION ROLE SPECIFYING UNIT EXTRACTS CERTAIN APPLICATION IDENTIFIER APP20 DESCRIBED IN APLLiST4, (DELETES APP20 FROM APLLiST4) AND ADDS IT TO APPLICATION IDENTIFIER LIST APLiST5.

APPLICATION ROLE SPECIFYING UNIT REFERs TO ADDRiLIST20 TO EXTRACT IDENTIFIER APP21 OF APPLICATION HAVING SAME ADDRESS AS APP20 FROM APLLiST4, (DELETES APP21 FROM APLLiST4) AND ADDS IT TO APLLiST5.

APPLICATION ROLE SPECIFYING UNIT STORES APPLICATION IDENTIFIER AND ROLE IDENTIFIER (ROLE#K) DESCRIBED IN APLLiST5 INTO APPLICATION ROLE CONFIGURATION LIST, TO INCREMENT K.
INFORMATION PROCESSING APPARATUS AND METHOD OF OPERATING THE SAME

CROSS-REFERENCE TO RELATED APPLICATIONS


BACKGROUND OF THE INVENTION

[0002] 1. Field of the Invention

[0003] The present invention relates generally to an information processing apparatus and a method of operating the same, and, more particularly, to an information processing apparatus and a method of operating the information processing apparatus, capable of easily and simply grasping the configuration of active applications in an environment where a plurality of applications are running on a plurality of servers.

[0004] 2. Description of the Related Art

[0005] To effectively manage operations of an IT-based service system, it is commonly necessary to correctly grasp the current system configuration. On the other hand, recent service systems are getting larger and larger due to introduction of system linkage technique, server virtualization technique, etc. When such a large-scale service system is to be managed, a difficulty may occur to the system administrator to correctly manage the current configuration of the service system. Because of such a difficulty, the system administrator may experience delay in recognizing or may fail to recognize any failure or configurational change of the service system, which may result in enormous damage to the service.


[0007] In accordance with the technique described in the above patent document, the system configuration diagram is automatically created by an output means that displays or prints out names of a plurality of groups obtained by classifying a service system. A specific procedure to automatically create the system configuration diagram is carried out by a means that, through the entry of the system administrator, stores a series of processes of the service system for featuring the groups classified when putting together the service system (e.g., the content of a process is whether the record is to be written when the definition element is a file), a means that analyzes processes of the service system to be classified and compares the results with a series of prestored processes to judge to which group the service system belongs, and the output means that displays or prints out the name of the classified group.

[0008] The method of the above patent document necessitates a processing unit that stores a series of processes of the service system for featuring the classified groups and requests the system administrator to enter the series of processes of the service system stored by the processing unit. The system administrator enters know-hows of the processes within the service system. Thus, to deal with a new process, maintenance of the know-hows is needed. If old know-hows are used intactly without any maintenance, the processes beyond the know-hows cannot be dealt with. Therefore, the service system executing such the processes cannot create correct service system configuration diagrams. This means that the system administrator cannot grasp the correct configuration of the current service system.

[0009] When displaying definition elements on a display screen of a management computer, if a multiplicity of service systems are to be administered, the definition elements cannot lie on a single screen because their display are not aggregated. This may render it difficult to grasp at a glance how many service systems exist.

SUMMARY OF THE INVENTION

[0010] It is therefore the object of the present invention to enable the configuration of active applications to easily and simply be grasped in an environment where a plurality of applications is running on a plurality of servers.

[0011] In order to achieve the above and other objects, one aspect of the present invention provides an information processing apparatus communicatively coupled via a network to servers on which a plurality of applications are running for operating a service system, the information processing apparatus comprising a storage unit that stores network addresses of the servers, storing, in a correlated manner, application identification information as information for identifying the applications, active server addresses as network addresses of the servers on which the applications are running, and communication source addresses of the applications; and an application aggregating unit acquiring from the storage unit and comparing the application identification information, the addresses of the servers corresponding thereto, and communication source addresses of the applications, so that when it is judged that the communication source addresses do not coincide with any of the addresses of the servers, the application aggregating unit judges the applications having the application identification information as being applications belonging to a first layer in the service system, acquiring from the storage unit and comparing application identification information of the applications judged as belonging to the first layer, the active server addresses of the applications, and the application identification information and communication source addresses of applications other than the applications judged as belonging to the first layer, so that the application aggregating unit judges applications having the application identification information correlated to the communication source addresses identical to the active server addresses as belonging to a second layer; and continuing acquiring from the storage unit communication source addresses of applications other than the applications judged as belonging to the first and the second layer, to repeat the judgment process to judge the layers of each application contained in the service system.

[0012] The other problems and solutions thereof disclosed herein will become apparent when reading the detailed description of the invention while referring to the accompanying drawings.

[0013] According to the present invention, the configuration of active applications can be easily and simply grasped in an environment where a plurality of applications is active in a plurality of servers.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. 1 is a block diagram of an exemplary configuration of the information processing system 1;
[0015] FIG. 2 depicts, by way of example, a configuration diagram of applications before aggregation and a configuration diagram of applications aggregated by a first embodiment of the present invention;

[0016] FIG. 3 depicts an example of a service execution server list;

[0017] FIG. 4 depicts an example of an application list;

[0018] FIG. 5 depicts an example of an application role configuration list;

[0019] FIG. 6 depicts an example of an aggregated application list;

[0020] FIG. 7 depicts an example of an aggregated application communication list;

[0021] FIG. 8 depicts an example of a service system list;

[0022] FIG. 9 depicts an example of an application configuration baseline;

[0023] FIG. 10 depicts an example of an application configuration change list;

[0024] FIG. 11 depicts an example of a setting change apply list;

[0025] FIG. 12 depicts an example of a physical/logical (virtual) server list;

[0026] FIG. 13 depicts an example of a shared application list;

[0027] FIG. 14 is an exemplary hardware configuration diagram of a server such as an operation management computer;

[0028] FIG. 15 is an exemplary flowchart of a general procedure effected by the information processing system according to the first embodiment;

[0029] FIG. 16 is an exemplary flowchart of a procedure for specifying applications of a first layer according to the first embodiment;

[0030] FIG. 17 is an exemplary flowchart of a procedure for specifying applications of a second or subsequent layer according to the first embodiment;

[0031] FIG. 18 is an exemplary flowchart of a procedure for specifying roles of applications according to the first embodiment;

[0032] FIG. 19 is an exemplary flowchart of a procedure for specifying communications between aggregated applications according to the first embodiment;

[0033] FIG. 20 is an exemplary detailed flowchart of a procedure for specifying communications between aggregated applications according to the first embodiment;

[0034] FIG. 21 is an explanatory diagram of a problem involved in an environment where a plurality of service systems mixedly coexist;

[0035] FIG. 22 is an explanatory diagram of a problem involved in sharing of applications by the plurality of service systems;

[0036] FIG. 23 is an exemplary flowchart of a procedure for specifying service systems according to a second embodiment of the present invention;

[0037] FIG. 24 is an exemplary flowchart of a procedure for specifying service systems to which applications of the first layer belong according to the second embodiment;

[0038] FIG. 25 is an exemplary flowchart of a procedure for specifying applications shared by a plurality of service systems according to the second embodiment;

[0039] FIG. 26 is an exemplary flowchart of a procedure for specifying applications running on virtual servers working on the same physical server according to the second embodiment;

[0040] FIG. 27 is an exemplary configuration diagram of applications aggregated by a procedure of a third embodiment of the present invention;

[0041] FIG. 28 is an exemplary flowchart of a procedure for specifying changes in the configuration according to the third embodiment;

[0042] FIG. 29 is an exemplary flowchart of a procedure for specifying changes in the communication status of applications according to the third embodiment;

[0043] FIG. 30 is an exemplary flowchart of a setting change procedure according to a fourth embodiment of the present invention;

[0044] FIG. 31 is an exemplary flowchart of a service system specifying procedure according to a fifth embodiment of the present invention; and

[0045] FIG. 32 is an exemplary flowchart of a procedure for specifying roles of applications according to a sixth embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0046] Exemplary embodiments of the present invention will now be described with reference to the accompanying drawings.

First Embodiment

[0047] FIG. 1 is a block diagram of the configuration of an information processing system 1 according to a first embodiment of the present invention. FIG. 2 depicts, by way of example, a configuration diagram applications before aggregation and a configuration diagram of applications aggregated and displayed by the first embodiment.

[0048] The information processing system 1 of the first embodiment includes an operation management server 100, an operation management terminal 11, and a service execution server 102, which are communicatively coupled to one another by way of a network 1109. Although the details will be described later, the operation management server 100, the operation management terminal 101, and the service execution server 102 are computers each having a typical hardware configuration, provided with a CPU, memories (RAM, ROM, etc.), a data I/O interface, a communication interface (NIC, etc.), an auxiliary storage device (disk device, etc.) storing software programs, entry devices such as a keyboard and a mouse, a display device, and an output device such as a printer.

[0049] Functional blocks (e.g., an application aggregating unit 106 of the operation management server 100) assigned to the operation management server 100, the operation management terminal 101, and the service execution server 102 are implemented by the CPU reading in a memory and executing a software program for that function which is stored in the auxiliary storage device.

[0050] The operation management server 100 will first be described. The operation management server 100 is a server computer that executes a procedure to aggregate and display applications running in the service execution server 102 which will be described later. The operation management server 100 includes a transmitter/receiver unit 103, a system configuration managing unit 104, an application role specifying unit 105, the application aggregating unit 106, an application configuration diagram creating unit 107, a service sys-
tem specifying unit 108, a configuration change managing unit 109, a settings change managing unit 110, and a storage unit 111.

[0051] The transmitter/receiver unit 103 executes a communication process for the operation management server 100. The transmitter/receiver unit 103 allocates information received via the network 1109 from the operation management terminal 101 or the service execution server 102 to the processing units of the operation management server 100 which are specified by the received information. The transmitter/receiver unit 103 sends out via the network 1109 information to be sent to the operation management terminal 101 and the service execution server 102 by the processing units of the operation management server 100.

[0052] The system configuration managing unit 104 stores, depending on the input data type, input data received from the operation management terminal 101 into a service execution server list 200, an application list 300, or an application configuration baseline 800 which are stored in the storage unit 111, which will be described later.

[0053] The application role specifying unit 105 specifies, for an application running in a computer to be managed by the operation management server 100, a role of the application in a service system. The object of management is a computer whose operation is to be managed by the operation management server 100. In this embodiment, it is the service execution server 102.

[0054] As used herein, the term “role of the application” represents each of groups classified by names of applications for each of application layers which will be described later. As used herein, the term “application layer” means “the communication order of applications in the service system communicated from the exterior of the service system”. That is, an application firstly communicated from the exterior of the service system is an application of a layer 1 (a first layer); an application communicated from the application of the layer 1 is an application of a layer 2; and an application communicated from the application of the layer 2 is an application of a layer 3.

[0055] The application role specifying unit 105 manages an application role configuration list 400 which will be described later. Upon receiving a message requiring to specify an application role, the application role specifying unit 105 receives from a server list 200 and the application list 300 from the storage unit 111 to specify an application role for storage in the application role configuration list 400 of the storage unit 111.

[0056] The application aggregating unit 106 specifies the role of each of applications to store information on groups which will be described later into the application role configuration list 400 and an aggregated application list 500 which will be described later. The applications are classified into a single group or a plurality of groups of each role based on information on those applications. In the first embodiment, the applications are grouped based on the role and the name of those applications.

[0057] The application aggregating unit 106 reads the application role configuration list 400 from the storage unit 111 to count the number of applications having the same role for storage in the aggregated application list 500 of the storage unit 111.

[0058] In a second and a third embodiments which will be described later, the application aggregating unit 106 reads a service system list 700 or an application configuration change list 900, in addition to the application list 300 and the application role configuration list 400, from the storage unit 111, to specify the roles of the applications.

[0059] The application aggregating unit 106 reads the application list 300, the application role configuration list 400, and the aggregated application list 500 from the storage unit 111, to specify a role as a communication source and a role as a communication destination for each of the roles, for storage in an aggregated application communication list 600, which will be described later, of the storage unit 111.

[0060] The application configuration diagram creating unit 107 reads the aggregated application list 500 and the aggregated application communication list 600 from the storage unit 111 to create an application configuration diagram. FIG. 2 depicts examples of application configuration diagrams before and after aggregation according to the first embodiment. As shown in the aggregated application configuration diagram, the application configuration diagram creating unit 107 creates data for grouping applications by roles for display on a single screen so as not to exceed the size of video data making up one screen of a monitor screen provided on the operation management server 100.

[0061] The service system specifying unit 108 reads the application role configuration list 400 and the application list 300 for connection of a role 1 application to Web to store a title displayed as a result thereof in the service system list 700.

[0062] The configuration change managing unit 109 reads the application list 300 and the application configuration baseline 800, which will be described later, from the storage unit 111 to specify contents of the change from the application configuration baseline 800 for storage in the application configuration change list 900, which will be described later.

[0063] Upon receiving a setting change request from the operation management terminal 101, the setting change managing unit 110 reads the application list 300 and the application role configuration list 400 from the storage unit 111 to specify an application whose setting is to be changed and a server in which that application is active, for transmission of the setting contents to that server.

[0064] The storage unit 111 is a storage that stores information required to implement an operation management procedure by the operation management server 100.

[0065] The storage unit 111 stores the service execution server list 200, the application list 300, the application role configuration list 400, the aggregated application list 500, the aggregated application communication list 600, the service system list 700, the application configuration baseline 800, the application configuration change list 900, a setting change apply list 1000, a physical/logical (virtual) server list 1200, and a shared application list 1300.

[0066] The service execution server list 200 is table-format data and stores addresses of servers such as the service execution server 102 to be managed by the operation management server 100. The “address” is an identifier allowing a server to be uniquely specified.

[0067] FIG. 3 depicts an example of a service execution server list 200. The service execution server list 200 stores in rows addresses of servers to be managed by the operation management server 100. The service execution server list 200 has a service execution server address column 201. The service execution server address column 201 stores addresses allowing unique specification of servers to be managed by the operation management server 100.
Through the operation management terminal 101, the system administrator enters information shown in the service execution server list 200 so that the system configuration managing unit 104 stores the entered information into the service execution server list 200 of the storage unit 111. The system configuration managing unit 104 may store in the service execution server list 200 results of address lists of the service servers which are collected using separate tools or utilities and received from the transmitter/receiver unit 103.

The application list 300 is table-format data and it stores information of applications running on servers such as the service execution server 102 to be managed by the operation management server 100. FIG. 4 depicts an example of the application list 300. The application list 300 stores information of applications. The application list 300 has an application identifier column 301, a server address column 302, an application name column 303, a communication source address column 304, and a communication destination address column 305. The application identifier column 301 stores application identifiers as identification information uniquely specifying applications. The server address column 302 stores addresses specifying servers having active applications, by which addresses correlations are made with the service execution servers 102 specified by the service execution server address column 201. The application name column 303 stores names (e.g., product names) of applications. The communication source address column 304 stores addresses of servers on which active applications are sending communications to respective applications. The communication destination address column 305 stores addresses of servers on which active applications are receiving communications from applications.

Through the operation management terminal 101, the system administrator enters information shown in the application list 300 so that the system configuration managing unit 104 stores the entered information in the application list 300 of the storage unit 111. The system configuration managing unit 104 may store in the application list 300 information of applications running on the service system which are collected using separate tools or utilities and received from the transmitter/receiver unit 103.

The application role configuration list 400 is table-format data and it has information on roles of applications in the service system stored in the application list 300. FIG. 5 depicts an example of the application role configuration list 400. The application role configuration list 400 stores information specifying the roles of applications in respective rows. The application role configuration list 400 has an application identifier column 401, a layer identifier column 402, and a role identifier column 403. The application identifier column 401 stores application identifiers as identification information specifying applications, by which identification information correlations are made with the application information specified by the application identifier column 301 of the application list 300. The layer identifier column 402 stores identifier information specifying the layers of applications specified by the application role specifying unit 105. The role identifier column 403 stores identifier information specifying the roles of applications specified by the application role specifying unit 105.

The application role specifying unit 105 stores in the application role configuration list 400 the layers and the roles of applications specified based on the contents of the service execution server list 200 and the application list 300. The application aggregating unit 106 stores the identifiers specifying the layers and the roles of applications in the application role configuration list 400, based on the contents of the application list 300 and the application role configuration list 400.

The aggregated application list 500 is table-format data and it stores information on roles of applications classified by the application aggregating unit 106. FIG. 6 depicts an example of the aggregated application list 500. The aggregated application list 500 stores information on the roles of applications in respective columns. The aggregated application list 500 has a role identifier column 501, an application name column 502, and a within-role number-of-applications column 503.

The role identifier column 501 stores identification information specifying roles of applications, by which identification information correlations are made with role information of applications specified by the role identifier column 403. The application name column 502 stores the names of applications within roles. The within-role number-of-applications column 503 stores the number of applications falling within each of the roles.

The application aggregating unit 106 stores the role information of each of applications into the aggregated application list 500, based on the contents of the application list 300 and the application role configuration list 400. The application aggregating unit 106 stores the role information on each of applications into the aggregated application list 500, based on the contents of the application list 300 and the application role configuration list 400 as well as the contents of the service system list 700 and/or the application configuration change list 900, which will both be described later. The number of applications falling within each of roles counted by the application aggregating unit 106 is stored in the aggregated application list 500.

The aggregated application communication list 600 is table-format data and it stores role-to-role communication information. FIG. 7 depicts an example of the aggregated application communication list 600. The aggregated application communication list 600 stores information on communications between roles in respective rows. The aggregated application communication list 600 has a communication source role identifier column 601 and a communication destination role identifier column 602. The communication role identifier column 601 stores identification information specifying the role of an information sending party in a communication. By this identifier information, correlations are made with the role information of applications specified by the role identifier column 501 of the aggregated application list 500. The communication destination role identifier column 602 stores identification information specifying the role of an information receiving party in the communication. By this identifier information, correlations are made with role information of applications specified by the role identifier column 501 of the aggregated application list 500.

The application aggregating unit 106 stores the role-to-role communication information specified based on the application list 300 and the application role configuration list 400 into the aggregated application communication list 600.

The service system list 700 is table-format data and it stores information on service systems and correlations between the service systems and applications used therein. The service systems are systems executing service processes required to operate a company such as sales, production, accounting, and physical distribution, and exemplified typi-
cally by a financial management system, a pay management system, an online purchase system, and a service management system. FIG. 8 depicts an example of the service system list 700.

[0079] The service system list 700 stores information on service systems and correlations between the service systems and applications used therein in respective rows. The service system list 700 has an application identifier column 701, a service system name column 702, and a role identifier column 703. The application identifier column 701 stores application identifiers as identification information specifying applications. This identifier information allows correlations with information of applications specified by the application identifier column 301 of the application list 300. The service system name column 702 stores names of service systems (e.g., the financial management system). The role identifier column 703 stores identification information specifying roles of applications. This identifier information allows correlations with role information of applications specified by the role identifier column 403 of the application role configuration list 400 and by the role identifier column 501 of the aggregated application list 500.

[0080] The service system specifying unit 108 stores in the service system list 700 the information on service systems to which belong applications specified based on the contents described in the application list 300 and the application role configuration list 400.

[0081] The application configuration baseline 800 is table-format data and it stores information of applications which are to be active on servers such as the service execution servers 102 to be managed by the operation management server 100. The application configuration baseline 800 is used when compared with the current application configuration. FIG. 9 depicts an example of the application configuration baseline 800.

[0082] The application configuration baseline 800 stores information of applications in respective rows. The application configuration baseline 800 has an application identifier column 801, a server address column 802, an application name column 803, a communication source address column 804, and a communication destination address column 805. The application identifier column 801 stores information of applications as identification information specifying applications. This identifier information allows correlations with information of applications specified by the application identifier column 301 of the application list 300. The server address column 802 stores addresses specifying servers having active applications thereon, by which addresses correlations are made with service servers specified by the service execution server address column 201 of the service execution server list 200. The application name column 803 stores names of applications. The communication source address column 804 stores addresses of servers on which active applications are sending communications to applications. The communication destination address column 805 stores addresses of servers on which active applications are receiving communications from the applications.

[0083] By way of the operation management terminal 101, the system administrator enters information indicated in the application configuration baseline 800 so that the system configuration managing unit 104 stores the entered information into the application configuration baseline 800 of the storage unit 111. The system configuration managing unit 104 may store in the application configuration baseline 800 information of applications running on the service systems which are collected using separate tools or utilities and received from the transmitter/receiver unit 103. The system configuration managing unit 104 may read the contents of the application list 300 and store the read contents into the application configuration baseline 800.

[0084] The application configuration change list 900 is table-format data and it stores information on change of configurations of applications from the application configuration baseline 800. FIG. 10 depicts an example of the application configuration change list 900. The application configuration change list 900 stores configuration change information of applications in respective rows. The application configuration change list 900 has an application identifier column 901 and a contents-of-change column 902.

[0085] The application identifier column 901 stores therein application identifiers as identification information specifying applications. By this identifier information, correlations are made with information of applications specified by the application identifier column 301 of the application list 300 and by the application identifier column 801 of the application configuration baseline 800. The contents-of-change column 902 stores the contents of change in configuration of applications. As used herein, the content of change in configuration of applications refers to information of changes applied to applications making up service systems, and includes events such as startup of an application, stop of an application, finding of a new communication destination, finding of a new communication source, interruption of communication with communication source, and interruption of communication with communication destination.

[0086] The configuration change managing unit 109 stores change information of applications into the application configuration change list 900, based on the contents of the application list 300 and the application configuration baseline 800. The transmitter/receiver unit 103 may receive, e.g., changes of contents of setting files of applications described in the application list 300 or the application configuration baseline 800, which are detected using separate tools or utilities. The configuration change managing unit 109 may then store the received contents into the application configuration change list 900.

[0087] The setting change apply list 1000 is table-format data and it stores information of changes for setting information of applications running on servers such as the service execution servers 102. FIG. 11 depicts an example of the setting change apply list 1000. The setting change apply list 1000 stores information that changes setting information of applications in respective rows. The setting change apply list 1000 has an application identifier column 1001 and a contents-of-setting column 1002.

[0088] The application identifier column 1001 stores application identifiers as identification information specifying applications. This identifier information allows correlations with information of applications specified by the application identifier column 301 of the application list 300. The contents-of-setting column 1002 stores the contents of change of the setting information of applications.

[0089] As used herein, the content of change of the setting information of applications refers to the content of change applied to the setting information of applications running on servers such as service execution servers 102, and includes change of parameters described in a specific setting file or
memory, deletion of parameters described in the specific setting file or memory, addition of parameters to a setting file or memory, and the like.

[0090] Through the operation management terminal 101, the system administrator enters information indicated in the setting change apply list 1000 so that the setting change managing unit 110 stores the entered information into the setting change apply list 1000 of the storage unit 111. Through the operation management terminal 101, the system administrator may also enter identification information specifying a specific single or a plurality of roles and the contents of setting the roles so that from the entered role specifying identifier information the setting change managing unit 110 may specify application identifiers belonging to the role while referring to the application role configuration list 400, and store the specified application identifiers together with the contents of setting of the role into the setting change apply list 1000 of the storage unit 111. By entering information indicated in the setting change apply list 1000 through separate utilities or tools, the setting change managing unit 110 may store the entered information into the setting change apply list 1000 of the storage unit 111.

[0091] The physical/logical (virtual) server list 1200 is table-format data and it stores information on types of servers to be managed such as the service execution servers 102 that the operation management server 100 manages. FIG. 12 depicts an example of the physical/logical (virtual) server list 1200. The physical/logical (virtual) server list 1200 stores information on types of servers in respective rows.

[0092] The physical/logical (virtual) server list 1200 has a physical server address column 1201 and a logical (virtual) server address column 1202. The physical server address column 1201 stores addresses of physical servers to be managed. The physical server address column 1201 stores addresses of physical servers on which logical (virtual) servers described in the logical (virtual) server address column 1202 are running. By this address information, correlations are made with server address information of applications specified in the server address column 302 of the application list 300.

[0093] The logical (virtual) server address column 1202 stores addresses of logical (virtual) servers to be managed. The logical (virtual) server address column 1202 stores addresses of logical (virtual) servers running on the physical servers described in the physical server address column 1201. By this address information, correlations are made with server address information of the service execution server list 200 and the application list 300.

[0094] Through the operation management terminal 101, the system administrator enters information indicated in the physical/logical (virtual) server list 1200 so that the system configuration managing unit 104 stores the entered information into the physical/logical (virtual) server list 1200. By entering information indicated in the physical/logical (virtual) server list 1200 through separate utilities or tools, the system configuration managing unit 104 may store the entered information into the physical/logical (virtual) server list 1200 of the storage unit 111.

[0095] The shared application list 1300 is table-format data and it stores information on whether applications running on servers such as the service execution servers 102 to be managed by the operation management server 100 are shared by a plurality of service systems and whether physical servers having the running virtual servers thereon are shared in case of applications running on the virtual servers. FIG. 13 depicts an example of the shared application list 1300. The shared application list 1300 stores, in respective rows, information on whether those applications are shared by a plurality of service systems and whether physical servers having the running virtual servers thereon are shared in case of applications running on the virtual servers.

[0096] The shared application list 1300 has a role identifier column 1301, a number-of-shared-applications column 1302, and a type-of-sharing column 1303. The role identifier column 1301 stores identification information for specifying roles of applications. This identification information allows correlations with role information of applications specified in the role identifier column 403 of the application role configuration list 400, with role information of applications specified in the role identifier column 501 of the aggregated application list 500, and with information on service systems and applications specified in the role identifier column 703 of the service system list 700.

[0097] The number-of-shared-applications column 1302 stores the number of applications shared by a plurality of service systems or the number of applications running on virtual servers that work on the same physical server. The type-of-sharing column 1303 stores the type of sharing of applications, as well as the information specifying whether the applications are ones shared by a plurality of service systems or whether the applications are ones running on virtual servers that work on the same physical server.

[0098] The application aggregating unit 106 counts the number of applications belonging also to other roles for each role, based on the contents described in the service system list 700, and stores the number together with the type of sharing into the shared application list 1300 of the storage unit 111. The application aggregating unit 106 counts the number of applications running on virtual servers that work on the same physical server for each role, based on the contents described in the physical/logical (virtual) server list 1200, the application list 300, and the service system list 700, and stores the number together with the type of sharing into the shared application list 1300 of the storage unit 111. Through the operation management terminal 101, the system administrator may also enter the information indicated in the shared application list 1300 so that the system configuration managing unit 104 may store the entered information into the shared application list 1300 of the storage unit 111. By entering information indicated in the shared application list 1300 through separate utilities or tools, the system configuration managing unit 104 may store the entered information into the shared application list 1300 of the storage unit 111.

[0099] The operation management server 100 runs on a server machine 1101 having a hardware configuration shown in FIG. 14. The server machine 1101 includes a CPU 1102, a main storage 1103 (RAM, ROM, etc.), an external storage interface 1104, a communication interface 1105 (NIC, etc.), an external storage 1106 (disk device, etc.) storing programs, etc., an input device 1107 such as a keyboard and a mouse, and an output device 1108 such as a display device and a printer.

[0100] The transmitter/receiver unit 103, the system configuration managing unit 104, the application role specifying unit 105, the application aggregating unit 106, the application configuration diagram creating unit 107, the service system specifying unit 108, the configuration change managing unit 109, and the setting change managing unit 110 are functional blocks implemented by the CPU 1102 executing programs.
stored in the external storage 1106 and read into the main storage 1103 via the external storage interface 1104. The transmitter/receiver unit 103 may also be implemented by the communication interface 1105 and a communication control program controlling the communication interface 1105. The storage unit 111 is implemented by the main storage 1103 and/or the external storage 1106.

[0101] Hardware configurations of the operation management terminal 101 and of the service execution server 102 are similar to the configuration of the server machine 1101 shown in FIG. 14 despite possible differences in their scales and performances.

[0102] Though the operation management server 100, the operation management terminal 101, and the service execution server 102 are represented as separate servers in FIG. 1, those servers may each be composed of a plurality of servers or may be integrated into a single server. The operation management server 100 is not limited to a physical server. Namely, it may be a virtual server or may be a logical server defined by cluster configuration, etc.

[0103] Referring back to FIG. 1, the operation management terminal 101 will be described. The operation management terminal 101 is a server that controls processes of data input to and data output from the operation management server 100. Through the operation management terminal 101, the system administrator sends to the operation management server 100 a request message to store the service execution server list 200, the application list 300, the application configuration baseline 800, and the setting change apply list 1000. Through the operation management terminal 101, the system administrator also sends a request message to create an application configuration diagram to the operation management server 100.

[0104] Further, through the operation management terminal 101, the system administrator sends to the operation management server 100 a request message to specify a service system for an application described in the application list 300. Through the operation management terminal 101, the system administrator also sends to the operation management server 100 a request message to compare an application described in the application list 300 with another application described in the application configuration baseline 800 to specify the contents of change of the application configuration.

[0105] Further yet, through the operation management terminal 101, the system administrator sends to the operation management server 100 a request message to change setting information on an application running on a server such as the service execution server 102. Through the operation management terminal 101, the system administrator also sends to the operation management server 100 a request message to create an application configuration diagram from information on an aggregated application described in the aggregated application list 500 and the aggregated application communication list 600.

[0106] The operation management terminal 101 has an input unit 112, an output unit 113, a transmitter/receiver unit 114, and a communication processing unit 115. The input unit 112 accepts input of information on the various request messages entered by the system administrator, whereas the output unit 113 outputs the information to the system administrator.

[0107] The transmitter/receiver unit 114 is a processing unit that executes a transmission/reception process including transmission of information received from the processing units such as the communication processing unit 115 of the operation management terminal 101 to the network 1109 and transmission of information received from the network 1109 to the processing units.

[0108] The communication processing unit 115 executes a communication process with the operation management server 100.

[0109] The operation management terminal 101 works on the server machine 1101 having the hardware configuration of FIG. 14.

[0110] The communication processing unit 115 can be implemented by the CPU 1102 executing a program on the external storage 1106 read into the main storage 1103.

[0111] The transmitter/receiver unit 114 can be implemented by the communication interface 1105.

[0112] The input unit 112 can be implemented by the input device 1107. The input device 1107 is for example a pointing device such as a mouse and/or a keyboard.

[0113] The output unit 113 can be implemented by the output device 1108. The output device is for example a monitor and/or a printer.

[0114] The service execution server 102 is a server on which are working service systems to be managed by the operation management server 100. The service execution server 102 and the service system may not necessarily have a one-to-one correspondence. That is, a single service system may work on the service execution server 102, or alternatively the service system may work for each of virtual servers running on the service execution server 102. Instead, the service system may also be made up of a plurality of service execution servers 102. The service execution server 102 is not limited to a physical server. That is, it may be a virtual server or may be a logical server defined by a cluster configuration etc.

[0115] The service execution server 102 includes a transmitter/receiver unit 116 and a setting managing unit 117. The transmitter/receiver unit 116 executes a process of communication between the setting managing unit 117 and the operation management server 100 or the operation management terminal 101. The setting managing unit 117 executes processes of setting or changing of setting information on the service execution server 102.

[0116] The operation management terminal 101 works on the server machine 1101 having the hardware configuration of FIG. 14.

[0117] The transmitter/receiver unit 116 and the setting managing unit 117 can be implemented by the CPU 1102 executing programs on the external storage 1106 read into the main storage 1103.

[0118] The transmitter/receiver unit 116 may be implemented by the communication interface 1105.

[0119] Description will hereinafter be made of an application aggregated display process implemented by this embodiment of the present invention. FIG. 15 is an exemplary flow-chart of a general procedure effected by the information processing system 1 of this embodiment.

[0120] Process steps S1401, S1402, S1403, and S1405 included in the flow of FIG. 15 will be detailed referring to FIGS. 16 to 19, respectively.

[0121] First, the application role specifying unit 105 specifies applications belonging to a first layer and stores information of the applications into the application role configuration list 400 (step L401).

[0122] Specifically, the procedure of FIG. 16 is carried out. The application role specifying unit 105 acquires all of appli-
The application role specifying unit 105 judges whether the list ApList1 is not empty (step 1502). If negative, i.e., the list ApList1 is empty (No at step 1502), then the procedure comes to an end. If affirmative, i.e., the list ApList1 is not empty (Yes at step 1502), then the application role specifying unit 105 executes processes from step 1503 onward.

The application role specifying unit 105 extracts an arbitrary application identifier App1 described in the list ApList1 and deletes the application identifier App1 from the list ApList1 so as to acquire a communication source address Addr1 and a communication destination address Addr2 of an application corresponding to the application identifier App1 from the application list 300 (step 1503).

The application role specifying unit 105 acquires an address list AdList1 of service servers described in the service execution server list 200 (step 1504).

The application role specifying unit 105 judges whether the communication source address Addr1 is not contained in the service execution server address list AdList1 (step 1505). If negative, i.e., the address Addr1 is contained in the list AdList1 (No at step 1505), then it judges that the application specified by the application identifier App1 is not an application belonging to the first layer, returns to the process of step 1502. If affirmative, i.e., if the address Addr1 is not contained in the list AdList1 (Yes at step 1505), then it judges that the application specified by the application identifier App1 is an application candidate belonging to the first layer, and executes processes from step 1506 onward.

The application role specifying unit 105 judges whether the communication destination address Addr2 is contained in the service execution server address list AdList1 (step 1506). If affirmative, i.e., if the address Addr2 is contained in the list AdList1 (Yes at step 1506), then it judges that the application specified by the application identifier App1 is an application belonging to the first layer, and executes a process of step 1507. If negative, i.e., if the address Addr2 is not contained in the list AdList1 (No at step 1506), then it judges that the application specified by the application identifier App1 is not an application belonging to the first layer, and returns to the process of step 1502.

The application role specifying unit 105 defines the application specified by the application identifier App1 to be an application belonging to the first layer, stores in the application role configuration list 400 the identifier (App1) of that application and information (layer #1) identifying a layer to which that application belongs (step 1507), and returns to the process of step 1502.

Referring to FIGS. 3 to 5, by way of example, application#1 and application#2 of all applications described in FIG. 4 are applications having communication addresses 304 different from addresses of the service execution server 102 described in FIG. 3 (192.168.2.1 to 192.168.2.6). The communication destination addresses 305 of the application#1 and application#2 are indicating the addresses of the service execution servers 102 described in FIG. 3. Hence, the application role specifying unit 105 judges that the application#1 and application#2 are the applications of the first layer which are the first applications to accept data from the exterior of the service execution server 102, and stores the application identifiers (application#1 and application#2) and the layer identifier (layer #1) into the application role configuration list 400, as shown in FIG. 5.

Referring back to FIG. 15, the application role specifying unit 105, then, specifies applications belonging to a second or subsequent layer and stores information on those applications into the application role configuration list 400 (step 1402).

Specifically, the procedure of FIG. 17 is carried out. The application role specifying unit 105 acquires all of application identifiers described in the application list 300 to create an application identifier list ApList2 of all the application identifiers acquired (step 1601).

The application role specifying unit 105 judges whether the list ApList2 is not empty (step 1602). If negative, i.e., the list ApList2 is empty (No at step 1602), then the procedure comes to an end. If affirmative, i.e., the list ApList2 is not empty (Yes at step 1602), then the application role specifying unit 105 executes processes from step 1603 onward.

The application role specifying unit 105 extracts an arbitrary application identifier App2 described in the list ApList2 and deletes the application identifier App2 from the list ApList2 to acquire a communication source address Addr3 of an application corresponding to the application identifier App2 from the application list 300 (step 1603).

The application role specifying unit 105 reads the application role configuration list 400 and acquires all of application identifiers each having a layer identifier (layer #i−1) indicative of a layer one-previous to a layer to be specified (layer i), to thereby create an address list AdList2 of servers on which applications with the layer identifier run (step 1604).

The application role specifying unit 105 judges whether the communication source address Addr3 is contained in the server address list AdList2 (step 1605). If affirmative, i.e., if the address Addr3 is contained in the list AdList2 (Yes at step 1605), then it judges that the application specified by the identifier App2 is an application of the i-th layer, and executes a process of step 1606. If negative, i.e., the address Addr3 is not contained in the list AdList2 (No at step 1605), then it judges that the application specified by the application identifier App2 is not an application of the i-th layer, and returns to the process of step 1601.

The application role specifying unit 105 defines applications specified by the application identifier App2 as applications of the i-th layer and stores the identifier (App2) of those applications and information (layer #i) identifying a layer to which those applications belong into the application role configuration list 400 (step S1606), and returns to the process of the step 1601.

In FIGS. 4 and 5, by way of example, the application role specifying unit 105 first acquires application identifiers (application#1 and application#2) of applications having the layer identifiers of layer #1 (−2−1) described in FIG. 5. The application role specifying unit 105 then refers to the application list 300 of FIG. 4 to acquire addresses (192.168.2.1, 192.168.2.2) of servers on which applications corresponding to the application identifiers (application#1 and application#2) run.

The application role specifying unit 105 then defines as applications of role 2 applications having addresses 192.168.2.1 and 192.168.2.2 in the communication source address column 304, of applications described in FIG. 4, and
stores application identifiers (application#3, application#4, application#5) and a layer identifier (layer #2) into the application role configuration list 400, as shown in FIG. 5.

[0139] Returning to FIG. 15 for description, the application role specifying unit 105 classifies applications into a plurality of roles to store role information on those applications into the application role configuration list 400 (step 1703). The application role specifying unit 105 acquires an application identifier list ApList3 of applications having layer #1 as the layer identifier described in the application role configuration list 400. The application role specifying unit 105 further defines the current role number k as 1 (step 1702).

[0141] The application role specifying unit 105 judges whether the application identifier list ApList3 is empty (step 1703). If affirmative, i.e., if ApList3 is empty (Yes at step 1703), then the procedure comes to an end. If negative, i.e., ApList3 is not empty (No at step 1703), then it executes processes from step 1704 onward. The application role specifying unit 105 extracts an arbitrary application identifier App3 from the application list ApList3 and deletes the identifier App3 from the list ApList3 so as to acquire an application name ApName3 corresponding to the identifier App3 from the application list 300 (step 1704).

[0142] The application role specifying unit 105 acquires application identifiers of applications having the same application names as ApName3 among those described in the application list 300 to create an application server list ApList4 (inclusive of App3) of the application identifiers acquired, and deletes the application identifiers described in the application identifier list ApList4 from the application identifiers described in the application identifier list ApList3 (step 1705).

[0143] The application role specifying unit 105 stores the application identifiers and role identifiers (role #k) described in the application identifier list ApList4 into the application role configuration list 400 to thereafter increment k (step 1706).

[0144] The application role specifying unit 105 judges whether the application identifier list ApList3 is empty (step 1707). If negative, i.e., if ApList3 is not empty (No at step 1707), then the procedure returns back to the process of step 1704. If affirmative, i.e., if ApList3 is empty, then a process of step 1708 is carried out. The application role specifying unit 105 increments the current layer number j (step 1708), and returns to the process of step 1702.

[0145] By way of example, description will be made of a procedure for classifying applications of a second layer into a plurality of roles. In FIGS. 4 and 5, the application role specifying unit 105 first acquires application identifiers 401 (application#3, application#4, application#5) having the layer identifier 402 of layer #2, of applications described in FIG. 5.

[0146] The application role specifying unit 105 then refers to the application list 300 of FIG. 4 to acquire application names 303 (Application B, Application C, and Application C) corresponding respectively to the application identifiers 401 (application#3, application#4, and application#5).

[0147] The application role specifying unit 105 then judges the applications having the same application name 303 of those applications to be of the same role. As shown in FIG. 5, the application role specifying unit 105 classifies the application #3 as one having a different role (the role identifier 403 of role 2) from the role of the application #4 and application #5, and classifies the application #4 and application #5 as ones having the same role as each other but having a different role (the role identifier 403 of role 3) from the role of the application #2. Afterward, the application role specifying unit 105 stores the specified role identifiers 403 into the application role configuration list 400.

[0148] Description will return to the procedure of FIG. 15. The application aggregating unit 106 refers to the application role configuration list 400 to count the numbers of applications belonging to each of the roles and stores the numbers together with the application names 303 into the aggregated application list 500 (step 1404).

[0149] In case of a role “#1” of FIG. 5 by way of example, the application aggregating unit 106 acquires the application identifiers 401 (application #1 and application #2) and the number (2) of the applications having the role identifier 403 of “role #1” among those described in FIG. 5. The application aggregating unit 106 acquires the name 303 (Application A) of applications having the application identifiers 301 of “application #1” or “application #2” described in FIG. 4. The application aggregating unit 106 then stores the numbers of applications and the application name 303 thus acquired into the aggregated application list 500, as shown in FIG. 6.

[0150] The application aggregating unit 106 then specifies the communication source and communication destination of the applications by the role (step 1405).

[0151] Specifically, the procedure of FIG. 19 is carried out. The application aggregating unit 106 defines the current role number m as 1 (step 1801). The application aggregating unit 106 acquires the application identifiers 401 of applications having the role identifiers 403 of role #m described in the application role configuration list 400, to create an application identifier list ApList5 of the application identifiers (step 1802).

[0152] The application aggregating unit 106 judges whether the application identifier list ApList5 is empty (step 1803). If affirmative, i.e., if ApList5 is empty (Yes at step 1803), then the procedure comes to an end. If negative, i.e., if ApList5 is not empty (No at step 1803), then the application aggregating unit 106 executes processes from step 1804 onward.

[0153] The application aggregating unit 106 then executes processes from step 1901 to 1905 described in FIG. 20 to specify communications between aggregated applications (step 1804). Specifically, the application aggregating unit 106 first extracts an arbitrary application identifier App5 from the application identifier list ApList5 and deletes the application identifier App5 from the list ApList5, to acquire a communication destination address Add5 corresponding to the application identifier App5 from the application list 300 (step 1901).

[0154] The application aggregating unit 106 acquires from the application list 300 application identifiers App6 of applications running on a server having an address indicated by Add5 (step 1902).

[0155] The application aggregating unit 106 acquires role identifiers (role #n) of applications corresponding to the application identifiers App6 from the application role configuration list 400 (step 1903).

[0156] The application aggregating unit 106 refers to the aggregated application communication list 600 to judge
whether no communication exists whose communication source role identifier 601 is role #n and whose communication destination role identifier 602 is role #n (step 1904). If negative, i.e., if the communication exists (No at step 1904), then the application aggregating unit 106 executes the processes from step 1805 onward without execution of a process of step 1905. If affirmative, i.e., if no communication exists (Yes at step 1904), then it executes the process of step 1905 to thereafter execute the processes from step 1805 onward.

[0157] When the communication does not exist, the application aggregating unit 106 stores role #n and role #n into the communication source role identifier column and the communication destination role identifier column, respectively, of the aggregated application communication list 600 (step 1905), and returns the procedure to step 1805 of FIG. 19.

[0158] The application aggregating unit 106 then judges whether the application identifier list Apf.List is empty (step 1805). If negative, i.e., if not empty (No at step 1805), then the procedure returns to the process of step 1804, whereas if affirmative, i.e., if empty (Yes at step 1805), then the application aggregating unit 106 executes a process of step 1806. The application aggregating unit 106 increments the current role member number m (step 1806), and returns to the process of step 1802.

[0159] Description will be given of a process of specifying a role of communicating with role #1 by way of example. The application aggregating unit 106 acquires the application identifiers 401 (application #1 and application #2) of applications having the role identifiers 403 of “role #1” stored in the application role configuration list 400 of FIG. 5.

[0160] The application aggregating unit 106 then refers to the application list 300 of FIG. 4 to acquire the communication destination addresses 305 (192.168.2.3, 192.168.2.4, and 192.168.2.5 for both) of applications specified by the application identifiers 301 of application #1 and application #2.

[0161] The application aggregating unit 106 refers to the application list 300 of FIG. 4 to acquire the application identifiers 301 (application #3, application #4, and application #5) of applications running on servers having the addresses acquired.

[0162] The application aggregating unit 106 then refers to the application role configuration list 400 of FIG. 5 to acquire the role identifiers 403 (role #2, role #3, and role #3) corresponding to the application identifiers 401 acquired. The application aggregating unit 106 then stores role #1 and role #2 into the communication source role identifier column and the communication destination role identifier column, respectively, of the aggregated application communication list 600, as shown in FIG. 7. The application aggregating unit 106 further stores role #1 and role #3 into the communication source role identifier column and the communication destination role identifier column, respectively, of the aggregated application communication list 600.

[0163] The application configuration diagram creating unit 107 reads the aggregated application list 500 and the aggregated application communication list 600 to create an application configuration diagram (step 1406).

[0164] To summarize, the first embodiment makes use of: the service execution server list 200 storing addresses of servers such as the service execution servers #2 to be managed by the operation management server 100; the application list 300 storing information of applications running on servers to be managed; the application role configuration list 400 storing information roles in a service system of applications stored in the application list 300; the aggregated application list 500 storing information on roles of applications classified by the application aggregating unit; and the aggregated application communication list 600 storing information on role-to-role communications. The first embodiment through use of the above lists includes the following steps of: specifying application roles in the service system from the service execution server list 200 and the application list 300 for storage in the application role configuration list 400; aggregating applications by roles from the application list 300 and the application role configuration list 400 for storage of the aggregation result into the aggregated application list 500 and the aggregated application communication list 600; and creating an application configuration diagram from the aggregated application list 500 and the aggregated application communication list 600.

[0165] According to the first embodiment, without any know-how, application configuration diagrams can be created that allow the system administrator to grasp the current configuration at a single glance, by providing aggregated displays of applications for each of roles in a service system.

[0166] It is to be noted that the first embodiment may or may not use: the service system specifying unit 108, the configuration change managing unit 109, the setting change managing unit 110, the service system list 700, the application configuration baseline 800, the application configuration change list 900, the setting change apply list 1000, the physical/logical (virtual) server list 1200, and the shared application list 1300.

[0167] The operation management server 100 of the first embodiment may or may not be provided with: the service system specifying unit 108, the configuration change managing unit 109, the setting change managing unit 110, the service system list 700, the application configuration baseline 800, the application configuration change list 900, the setting change apply list 1000, the physical/logical (virtual) server list 1200, and the shared application list 1300.

Second Embodiment

[0168] In the first embodiment, even when a plurality of service systems mixedly coexist, they are aggregated according to roles, but the aggregation and display may also be performed according to service systems as shown in FIG. 21. To this end, HTTP communications are attempted to addresses of servers on which applications of the first layer run, and titles of displayed pages are acquired to add a process of specifying, by the titles, service systems to which the applications belong.

[0169] A second embodiment of the present invention provides a method of automatically specifying service systems to which applications running on the servers belong and aggregating applications according to service systems to create an application configuration diagram when the servers such as the service execution servers #2 to be managed consist of a plurality of the service systems. The second embodiment further provides a method of aggregating individually applications shared by a plurality of service systems or applications running on virtual servers working on the same physical server as shown in FIG. 22 when the applications are aggregated and displayed according to service systems, so that those events can be grasped, to create an application configuration diagram.

[0170] In the second embodiment, the following procedure is executed of specifying service systems on which applica-
The service system specifying unit 108 judges whether the application identifier list ApList10 is empty (step 2102) and, if ApList10 is empty (Yes at step 2102), executes processes from step 2003 onward, whereas if not empty (No at step 2102), it executes processes from step 2103 onward.

[0176] The service system specifying unit 108 extracts an arbitrary application identifier App10 from the application identifier list ApList10 and deletes the application identifier App10 from the list ApList10, to acquire an address Add10 of a server on which an application specified by App10 runs from the application list 300 (step 2103).

[0177] The service system specifying unit 108 adds “https://” anterior to Add10 and adds “/” posterior to Add10 to send an HTTP connection request by way of the transmitter/receiver unit 103 (step 2104).

[0178] The service system specifying unit 108 makes judgment of whether the communication has succeeded (step 2105) and, if affirmative (Yes at step 2105), executes processes from step 2106 onward, whereas if negative (No at step 2105), then it performs a process of step 2109.

[0179] The service system specifying unit 108 applies DOM analysis to a source code of a Web page acquired and displayed from the address via the communication (step S2106).

[0180] Based on the result of the DOM analysis applied to the source code of the Web page displayed, the service system specifying unit 108 judges whether a Title node exists (step 2107) and, if affirmative (Yes at step 2107), executes a process of step 2108, whereas if negative, then it executes the process of step 2109.

[0181] If the Title node is present, the service system specifying unit 108 stores App10 and the contents of the Title node into the application identifier column 701 and the service system name column 702, respectively, of the service system list 700 (step 2108).

[0182] If the Title node is absent, the service system specifying unit 108 stores App10 and information indicative of service system being unspecified into the application identifier column 701 and the service system name column 702, respectively, of the service system list 700 (step 2109).

[0183] The service system specifying unit 108 specifies service systems on which applications of the second or subsequent layer run, based on communication destination addresses of applications of the first layer (step 2003).

[0184] Specifically, the service system specifying unit 108 acquires application identifiers 401 of applications having the layer identifier 402 of layer #1 described in the application role configuration list 400, to create a list ApList100 of application identifiers 401.

[0185] The service system specifying unit 108 acquires an arbitrary application identifier App100 from the application identifier list ApList100. The service system specifying unit 108 reads the application list 300 to acquire all of communication destination addresses Add100 of applications indicated by the application identifier App100.

[0186] The service system specifying unit 108 reads the application list 300 to acquire applications App101 running on servers indicated by the communication destination address Add100 so as to store App101 and names of service systems on which applications specified by the application identifier App100 run into the application identifier column 701 and the service system name column 702, respectively, of the service system list 700. The above processes are repeated to specify service systems on which applications of the third or subsequent layer run.

[0187] The application aggregating unit 106 acquires application identifiers 701 described in the service system list 700 to create a list ApList10 of the application identifiers 701 (step 2004).

[0188] The application aggregating unit 106 judges whether the application identifier list ApList10 is empty (step 2005) and, if affirmative (Yes at step 2005), terminates the procedure, whereas if negative (No at step 2005), it executes processes from step 2006 onward.

[0189] The application aggregating unit 106 extracts an arbitrary application identifier App10 stored in the application identifier list ApList10 and deletes the application identifier App10 from the list ApList10 (step 2006).

[0190] The application aggregating unit 106 acquires from the service system list 700 an application identifier list ApList11 (including App10) indicative of the same service system name (GSName10) as one of the application identifier App10 and deletes the application identifier list ApList11 from the list ApList10 (step 2007).

[0191] The application aggregating unit 106 applies the processes from the steps 1403 to 1406 described in FIG. 15 to applications having identifiers described in ApList11, to create an application configuration diagram of the service system (GSName10).

[0192] The current role number k at step 1702 is defined as a consecutive number (unique) following the last number in the preceding service system, and identifiers of roles classified at step 1403 are stored in the service system list 700 as well (step 2008), returning the procedure to the process of step 2005.

[0193] Description will next be given of a procedure for specifying applications shared by a plurality of service systems. FIG. 25 is a flowchart of a procedure for specifying applications shared by the plurality of service systems in the second embodiment. The flow is executed in addition to the flow specifying the service systems shown in FIG. 23.

[0194] The application aggregating unit 106 acquires the list ApList11 of application identifiers described in the service system list 700 (step 2201).
The application aggregating unit 106 judges whether the application identifier list ApList11 is empty (step 2202) and, if affirmative (Yes at step 2202), brings the procedure to an end, whereas if negative (No at step 2202), it executes processes from step 2203 onward.

If the application identifier list ApList11 is not empty, then the application aggregating unit 106 extracts an arbitrary application identifier App11 from the list ApList11 and deletes the application identifier App11 from the list ApList11, to acquire a role identifier Role11 corresponding to the application identifier App11 from the service system list 700 (step 2203).

The application aggregating unit 106 judges whether the role identifier Role11 is not empty (step 2204) and, if negative (No at step 2204), returns to the process of step 2202, whereas if affirmative (Yes at step 2204), it executes processes from step 2205 onward.

If the role identifier list Role11 is not empty, the application aggregating unit 106 extracts an arbitrary role identifier Role11 within the list Role11 and deletes the role identifier Role11 from the role identifier list Role11 (step 2205).

The application aggregating unit 106 refers to the shared application list 1300 to acquire a shared application identifier ShareApp11 corresponding to applications having the role identifier Role11 and the type-of-sharing 1303 indicative of sharing a plurality of service systems (step 2206).

The application aggregating unit 106 judges whether the shared application identifier ShareApp11 exists (step 2207) and, if affirmative (Yes at step 2207), executes a process of step 2208, whereas if negative (No at step 2207), it executes a process of step 2209.

If the shared application identifier ShareApp11 is present, then the application aggregating unit 106 increments the number-of-shared-applications of applications having the shared application identifier of ShareApp11 within the shared application list 1300 (step 2208).

The application aggregating unit 106 stores Role11 into the role identifier column 1301 of the shared application list 1300, stores 1 into the number-of-shared-applications column 1302, and stores the contents indicative of sharing a plurality of service systems into the type-of-sharing column 1303 (step 2209).

Description will next be given of a procedure for specifying applications running on virtual servers that work on the same physical server. FIG. 26 is a flowchart of a procedure for specifying applications running on virtual servers that work on the same physical server in the second embodiment. The flow is executed in addition to the flow specifying the service systems shown in FIG. 23.

The application aggregating unit 106 refers to the physical/logical (virtual) server list 1200 to acquire a list SrvAdd12 of logical/virtual server addresses indicative of the same physical server address 1201 (step 2301).

The application aggregating unit 106 judges whether the list SrvAdd12 is empty (step 2302) and, if affirmative (Yes at step 2302), brings the procedure to an end, whereas if negative (No at step 2302), it executes processes from step 2303 onward.

If the physical/logical server address list SrvAdd12 is not empty, then the application aggregating unit 106 extracts an arbitrary address Add12 from the list SrvAdd12 (and deletes the address Add12 from the list SrvAdd12), to acquire from the application list 300 an application identifier list ApList12 of applications running on a server indicated by the address Add12 (step 2303).

The application aggregating unit 106 judges whether the application identifier list ApList12 is empty (step 2304) and, if affirmative (Yes at step 2304), returns to the process of step 2302, whereas if negative (No at step 2304), it executes processes from step 2305 onward.

If the application identifier list ApList12 is not empty, then the application aggregating unit 106 extracts an arbitrary application identifier App12 from the list ApList12, deletes the application identifier App12 from the list ApList12, and refers to the application role configuration list 400 to acquire a role Role12 corresponding to the application identifier App12 (step 2305).

The application aggregating unit 106 refers to the shared application list 1300 to acquire a shared application identifier ShareApp12 corresponding to applications having the role identifier Role12 and the type-of-sharing 1303 of sharing virtual servers (step 2306).

The application aggregating unit 106 judges whether the shared application identifier ShareApp12 exist (step 2307) and, if affirmative (Yes at step 2307), executes a process of step 2308, whereas if negative (No at step 2307), it executes a process of step 2309.

If the shared application identifier ShareApp12 is present, the application aggregating unit 106 increments the number-of-shared-applications of applications having the shared application identifier of ShareApp12 within the shared application list 1300 (step 2308).

If the shared application identifier ShareApp12 is absent, the application aggregating unit 106 stores Role12 into the role identifier column 1301 of the shared application list 1300, stores 1 into the number-of-shared-applications column 1302, and the contents of sharing virtual servers into the type-of-sharing column 1303 (step 2309).

To summarize, the second embodiment makes use of: the service execution server list 200 storing therein addresses of servers such as the service execution servers 102 to be managed by the operation management server 100; the application list 300 storing information of applications running on servers to be managed; the application role configuration list 400 storing information roles in a service system of applications stored in the application list 300; the aggregated application list 500 storing therein information on roles of applications classified by the application aggregating unit; the aggregated application communication list 600 storing information on role-to-role communications; the service system list 700 storing therein service systems to which applications belong; the physical/logical (virtual) server list 1200 storing types of servers on which the applications run; and the shared application list 1300 storing the contents that an application is shared by a plurality of service systems or that a physical server having active applications thereon is shared by other service systems. The second embodiment through use of the above lists includes the following steps of: specifying application roles in the service systems from the service execution server list 200 and the application list 300 for storage in the application role configuration list 400; acquiring addresses of servers on which applications of role 1 run from the application role configuration list 400 and the application list 300; attempting HTTP communications with the acquired addresses to specify pages displayed upon the execution of HTTP communications as service systems to
which the applications belong; aggregating the applications on a role-by-role basis from the application list 300 and the application role configuration list 400 in a prejudged service system for storage of the aggregation results into the aggregated application list 500 and the aggregated application communication list 600; storing from the service system list 700 the number of applications shared by a plurality of service systems into the shared application list 1300; referring to the application list 300, the service system list 700, and the physical/logical (virtual) server list 1200 to store into the shared application list 1300 the number of applications running on virtual servers working on the same physical server; and creating an application configuration diagram from the aggregated application list 500, the aggregated application communication list 600, and the shared application list 1300.

According to the second embodiment, even when a plurality of service systems mixedly coexist as objects to be managed by the operation management server 100, application configuration diagrams can be created that allow the system administrator to grasp the current configuration at a single glance, by automatically specifying service systems to which applications belong and providing aggregated displays of the applications for each of the service systems.

Further, even when one application is shared by a plurality of service systems or even when there exist applications of different service systems working on the same physical server, application configuration diagrams can be created that allow the system administrator to grasp the current configuration at a single glance, by aggregating the applications by shared applications.

The second embodiment may or may not use the configuration change managing unit 109, the setting change managing unit 110, the application configuration baseline 800, the application configuration change list 900, and the setting change apply list 1000. The operation management server 100 may or may not be provided with the configuration change managing unit 109, the setting change managing unit 100, the application configuration baseline 800, the application configuration change list 900, and the setting change apply list 1000.

Third Embodiment

In the first embodiment, even when a plurality of service systems mixedly coexist, the plurality of service systems are brought together so that applications are aggregated according to roles of the applications. As shown in FIG. 27, aggregation may also be made based on given additional information (e.g., setting information on each application, change information or fault information on each application) so that displays can be provided so as to allow the system administrator to intuitively grasp applications needing any measures or modifications such as a singular application having different settings from others (which may probably be attributable to setting error) or applications having changes or faults. To this end, an additional process is provided to classify applications based on the setting information, change information, fault information, etc.

A third embodiment of the present invention provides a method of, when additional information is given to applications running on objects to be managed such as the service execution server 102, aggregating the applications depending on the additional information to create an application configuration diagram. As used herein, additional information means various pieces of information of applications to be aggregated, such as configuration change information of applications, host type information of hosts (such as a virtual server, a physical server, or a logical server) on which the applications run, and setting information difference information of applications.

By way of example of the third embodiment, description will be made of a method of, when the application configuration baseline 800 indicating a proper application configuration is given as shown in FIG. 27, comparing the current application configuration with the baseline 800 to specify changes in the configuration and aggregating applications by the contents of the changes in the configuration so as to allow the system administrator to grasp information on changes of applications to be managed at a single glance.

Following the steps 1401 to 1407 described in the first embodiment, to implement the method, additional processes are provided of: storing into the application configuration change list 900 the result of comparison of the configuration baseline 800 of a given application with the current configuration of each of applications; specifying a group of the application based on the contents stored in the application configuration change list 900; and storing identification information specifying the role of the application into the application role configuration list 400.

FIG. 28 depicts a flow of a procedure for: storing into the application configuration change list 900 the result of comparison of the configuration baseline 800 of a given application with the current configuration of each of applications; specifying a group of the application based on the contents stored in the application configuration change list 900; and storing identification information specifying the role of the application into the application role configuration list 400.

The configuration change managing unit 109 reads lists of application identifiers 801 and communication source/communication destination addresses 804 and 805 of the application configuration baseline 800 into ListA generated by the configuration change managing unit 109 and reads lists of the application identifiers 301 and communication source/communication destination addresses 304 and 305 of the current application list 300 into ListB generated by the configuration change managing unit 109 (step 2401). ListA, and ListB may be stored in an appropriate memory area of the storage unit 111 for example.

As used herein, the application baseline refers to table-format data describing therein an originally proper application configuration. In the third embodiment, for simplicity of the comparison process, the contents of the baseline are the same as those of the application list 300, but the former may be different from the latter as long as changes in the application configuration can be specified.

The configuration change managing unit 109 judges whether ListB is not empty (step 2402). If ListB is not empty (Yes at step 2402), then the configuration change managing unit 109 executes a process of step 2403, whereas if ListB is empty (No at step 2402), then it executes processes from step 2404 onward.

The configuration change managing unit 109 then executes process A and process B described in FIG. 29 (step 2403).

Referring to FIG. 29, specifically, the configuration change managing unit 109 first extracts an arbitrary application identifier App13 from ListB and deletes App13 from ListB (step 2501).
The configuration change managing unit 109 judges whether App13 is not present in ListA (step 2502). If App13 is not present in ListA (Yes at step 2502), then it executes processes from step 2503 onward, whereas if App13 is present in ListA (No at step 2502), then it executes processes from step 2506 onward.

The configuration change managing unit 109 judges whether information of ListA has been acquired from the application configuration baseline 800 (step 2503). If the information of ListA has been acquired from the application configuration baseline 800 (Yes at step 2503), then it executes a process of step 2504, whereas if the information of ListA has not been acquired from the application baseline 800 (No at step 2503), then it executes a process of step 2505.

If the information of ListA has not been acquired, then the configuration change managing unit 109 saves App13 in the application identifier column 901 of the application configuration change list 900, saves the contents indicating startup of the application in the contents-of-change column 902 (step 2504), and returns to the process of step 2402.

If the information of ListA has not been acquired, then the configuration change managing unit 109 saves App13 in the application identifier column 901 of the application configuration change list 900, saves information indicating stop of the application in the contents-of-change column 902 (step 2505), and returns to the process of step 2402.

The configuration change managing unit 109 judges whether the process B is also to be executed and, since the process B is also to be executed in this process step, executes processes from step 2507 onward (step 2506).

The configuration change managing unit 109 acquires a communication source address Addr13 and a communication destination address Addr14 of App13 from ListA, acquires a communication source address Addr15 and a communication destination address Addr16 of App13 from ListB (step 2507), and returns to the process of step 2402.

The configuration change managing unit 109 judges whether Addr13 differs from Addr14 (step 2508). If different (Yes at step 2508), then it executes processes from step 2509 onward, whereas if not different (No at step 2508), then it executes processes from step 2510 onward.

If Addr13 and Addr15 differ, the configuration change managing unit 109 saves App13 in the application identifier column 901 of the application configuration change list 900, saves information indicative of change of the communication source in the contents-of-change column 902 (step 2509), and proceeds to step 2510.

If Addr13 and Addr15 do not differ, then the configuration change managing unit 109 judges whether Addr14 and Addr16 differ (step 2510). If different (Yes at step 2510), then it executes a process of step 2511, whereas if not different (No at step 2510), then it returns to the process of step 2402.

The configuration change managing unit 109 saves App13 in the application identifier column 901 of the application configuration change list 900, saves information indicative of change of the communication destination in the contents-of-change column 902 (step 2511), and returns to the process of step 2402.

The configuration change managing unit 109 reads lists of the application identifiers 301 and communication source/communication destination addresses 304 and 305 of the current application list 300 into ListA, and reads lists of application identifiers 801 and communication source/communication destination addresses 804 and 805 of the application configuration baseline 800 into ListB (step 2404).

The configuration change managing unit 109 judges whether ListB is not empty (step 2405). If ListB is not empty (Yes at step 2405), then it executes a process of step 2406, whereas if ListB is empty (No at step 2405), then it terminates the procedure.

The configuration change managing unit 109 then executes only the process A described in FIG. 29 (step 2406).

Specifically, the procedure of FIG. 29 is carried out. The configuration change managing unit 109 extracts an arbitrary application identifier App13 from ListB and deletes App13 from ListB (step 2501).

The configuration change managing unit 109 judges whether App13 is not present in ListA (step 2502). If App13 is not present in ListA (Yes at step 2502), then it executes the processes from step 2503 onward, whereas if App13 is present in ListA (No at step 2502), then it executes the processes from step 2506 onward.

The configuration change managing unit 109 judges whether information of ListA has been acquired from the application configuration baseline 800 (step 2503). If the information of ListA has been acquired from the application configuration baseline 800 (Yes at step 2503), then it executes the process of step 2504, whereas if the information of ListA has not been acquired from the application baseline 800 (No at step 2503), then it executes the process of step 2505.

If the information of ListA has been acquired from the application configuration baseline 800 (Yes at step 2503), then it executes the process of step 2504, whereas if the information of ListA has not been acquired from the application baseline 800 (No at step 2503), then it executes the process of step 2505.

If the information of ListA has been acquired from the application configuration baseline 800 (Yes at step 2503), then it executes the process of step 2504, whereas if the information of ListA has not been acquired from the application baseline 800 (No at step 2503), then it executes the process of step 2505.

The configuration change managing unit 109 saves App13 in the application identifier column 901 of the application configuration change list 900, saves the contents indicating startup of the application in the contents-of-change column 902 (step 2504), and returns to the process of step 2405.

If the information of ListA has not been acquired from the application configuration baseline 800, then the configuration change managing unit 109 saves App13 in the application identifier column 901 of the application configuration change list 900, saves information indicating stop of the application in the contents-of-change column 902, and returns to the process of step 2402.

If the information of ListA has been acquired from the application configuration baseline 800, then the configuration change managing unit 109 saves App13 in the application identifier column 901 of the application configuration change list 900, saves information indicating stop of the application in the contents-of-change column 902, and returns to the process of step 2402.

The configuration change managing unit 109 judges whether the process B is also to be executed and, since the process B is not executed in this process step, returns to the process of step 2402 (step 2506).

To summarize, the third embodiment makes use of: the service execution server list 200 storing addresses of servers such as the service execution servers 102 to be managed by the operation management server 100; the application list 300 storing information of applications running on servers to be managed; the application role configuration list 400 storing information roles in a service system of applications stored in the application list 300; the aggregated application list 500 storing information on roles of applications classified by the application aggregating unit; and the aggregated application communication list 600 storing information on role-to-role communications. The third embodiment through use of the above lists includes the following steps of: specifying application roles in the service system from the service execution server list 200 and the application list 300 for storage in the application role configuration list 400; and aggregating applications by roles from the application list 300 and the application role configuration list 400. The third embodiment further includes: comparing the application list...
300 with the application configuration baseline 800 describing therein an originally proper application configuration; storing change information derived from the application configuration baseline 800 into the application configuration change list 900; aggregating applications in detail depending on the contents of change from the application configuration change list 900, to update the application role configuration list 400; storing the results of aggregation into the aggregated application list 500 and the aggregated application communication list 600; and creating an application configuration diagram from the aggregated application list 500 and the aggregated application communication list 600.

[0247] According to the third embodiment, even when changes occur in the configuration of applications to be managed by the operation management server 100, application configuration diagrams can be created that allow the system administrator to grasp the contents of changes in the configuration at a single glance, by providing aggregated displays of the applications according to the contents of changes.

[0248] It is to be noted that the third embodiment may or may not use the service system specifying unit 108, the setting change managing unit 110, the service system list 700, the setting change apply list 1000, the physical/logical (virtual) server list 1200, and the shared application list 1300.

[0249] The operation management server 100 of the third embodiment may or may not be provided with the service system specifying unit 108, the setting change managing unit 110, the service system list 700, the setting change apply list 1000, the physical/logical (virtual) server list 1200, and the shared application list 1300.

Fourth Embodiment

[0250] A fourth embodiment of the present invention provides a method of relieving the workload on the system administrator arising from changes in setting, by collectively changing the setting information of applications having the same roles or the same names that run on objects such as the service execution servers 102 to be managed by the operation management server 100.

[0251] A specific procedure to implement the setting changes includes, in addition to the application aggregating procedure from steps 1401 to 1406 exemplarily depicted in FIG. 15, executing a process of changing setting information of applications which follows. Based on the contents of setting and identifier information specifying roles sent by the system administrator from the input unit 112 of the operation management server 101 through the transmitter/receiver unit 114 to the setting change managing unit 110 of the operation management server 100, setting information of applications belonging to the roles is changed.

[0252] FIG. 30 depicts a flow of a procedure for changing setting information according to the fourth embodiment.

[0253] The setting change managing unit 110 accepts a change request including a role identifier Role14 and setting contents Change14 from the system administrator (step 2601).

[0254] It is noted that the role identifier Role14 and setting contents Change14 may be the singular form or the plural form. The setting contents Change14 mean data describing therein a process of changing the setting information of applications. The setting contents Change14 designate processes such as "change setting item Y1 of setting file X1 to Z1", "delete setting item Y2 from setting file X2", and "add new setting item Y3 to setting file X3".

[0255] The setting change managing unit 110 refers to the application role configuration list 400 to acquire an identifier list Applist14 of applications having the role identifier of Role14, and stores the application identifiers of Applist14 and the setting contents Change14 into the setting change apply list 1000 (step 2602).

[0256] The setting change managing unit 110 judges whether the setting change apply list 1000 is not empty (step 2603). If empty (No at step 2603), then it brings the procedure to an end, whereas if not empty (Yes at step 2603), then it executes processes from step 2604 onward.

[0257] The setting change managing unit 110 extracts an arbitrary application identifier Appl17 and change contents Change15 described in the setting change apply list 1000 and deletes Appl17 and Change15 from the setting change apply list 1000 (step 2604).

[0258] The setting change managing unit 110 acquires an address Addr17 of a server on which an application Appl17 runs from the application list 300 (step 2605).

[0259] The setting change managing unit 110 sends Change15 to Addr17 (step 2606). Specifically, the change contents Change15 are sent via the transmitter/receiver unit 114 of the operation management terminal 101 to the transmitter/receiver unit 116 of the service execution server 102 having the address of Addr17.

[0260] The transmitter/receiver unit 116 of the service execution server 102 receives Change15 to send the reception contents to the setting managing unit 117, which in turn executes a setting change process described in Change15 (step 2607).

[0261] To summarize, the fourth embodiment makes use of: the service execution server list 200 storing therein addresses of servers such as the service execution servers 102 to be managed by the operation management server 100; the application list 300 storing information of applications running on servers to be managed; the application role configuration list 400 storing information on roles in a service system of applications stored in the application list 300; the aggregated application list 500 storing information on roles of applications classified by the application aggregating unit; and the aggregated application communication list 600 storing information on role-to-role communications. The fourth embodiment through use of the above lists includes the following steps: specifying application roles in the service systems from the service execution server list 200 and the application list 300 for storage in the application role configuration list 400; aggregating the applications on a role-by-role basis from the application list 300 and the application role configuration list 400 for storage of the aggregation results into the aggregated application list 500 and the aggregated application communication list 600; creating an application configuration diagram from the aggregated application list 500 and the aggregated application communication list 600; and thereafter acquiring applications whose settings are to be changed and setting contents from the setting change apply list 1000 to apply the changes to the applications.

[0262] According to the fourth embodiment, setting information of the applications can collectively be changed according to each of the aggregated roles.

[0263] It is to be noted that the fourth embodiment may or may not use the service system specifying unit 108, the configuration change managing unit 109, the service system list 700, the application configuration baseline 800, the applica-
tion configuration change list 900, the physical/logical (virtual) server list 1200, and the shared application list 1300.

[0264] The operation management server 100 of the fourth embodiment may or may not be provided with the service system specifying unit 108, the configuration change managing unit 109, the service system list 700, the application configuration baseline 800, the application configuration change list 900, the physical/logical (virtual) server list 1200, and the shared application list 1300.

Fifth Embodiment

[0265] Another method of providing aggregated display in groups each consisting of a plurality of service systems includes additional processes of acquiring a list of files that applications use and specifying service systems to which the applications belong by the list of files. For example, files such as WAR/EAR/JAR files are acquired in accordance with the JSR 77 model that is a standard model on management of the J2EE platform. The files may be acquired by other approaches such as MBean (Managed Bean), SNMP (Simple Network Management Protocol) and CIM (Common Information Model).

[0266] A fifth embodiment of the present invention provides a method of, when an object such as the service execution server 102 to be managed consists of a plurality of service systems, automatically specifying service systems to which applications running on the server belong, and aggregating the applications on the basis of those service systems to create an application configuration diagram.

[0267] The fifth embodiment includes executing a process of specifying service systems on which applications run from information of the list of files that the applications use, in addition to the steps 1401 to 1406 (FIG. 15) described in the first embodiment.

[0268] FIG. 31 is a flowchart of a procedure for specifying service systems on which applications run according to the fifth embodiment.

[0269] The service system specifying unit 108 acquires an application identifier list ApList15 from the application role configuration list 400 (step 2701).

[0270] The service system specifying unit 108 judges whether ApList15 is empty (step 2702) and, if ApList15 is empty (Yes at step 2702), terminates the procedure, whereas if ApList15 is not empty (No at step 2702), it executes processes from step 2703 onward.

[0271] The service system specifying unit 108 extracts an arbitrary application identifier App18 from ApList15 and deletes App18 from ApList15, to acquire an address18 of a server on which App18 runs from the application list 300 (step 2703). The service system specifying unit 108 sends to Addr18 a request to acquire a list FileList18 of WAR/EAR/JAR files stored in the server of Addr18 (step 2704).

[0272] The service system specifying unit 108 judges whether the acquisition of the file list has succeeded (step 2705) and performs processes from step 2706 onward only when the acquisition of the file list has succeeded (Yes at step 2705).

[0273] The service system specifying unit 108 stores App18 and FileList18 into the service system list (step 2706).

[0274] The service system specifying unit 108 judges whether within the service system list there exists an application identifier App19 coincident with the contents of WAR/EAR/JAR file list (step 2707). If App19 is present (Yes at step 2707), it executes a process of step 2708, whereas if App19 is not present (No at step 2707), it executes a process of step 2709.

[0275] If App19 is present, then the service system specifying unit 108 stores the same role identifier as that of App19 into the role identifier 403 corresponding to the application identifier 401 of App18 within the service system list 700 (step 2708), returning to the process of step 2702.

[0276] If App19 is not present, then the service system specifying unit 108 adds and stores a new role identifier into the role identifier 403 corresponding to the application identifier 401 of App18 within the service system list 700 (step 2709), returning to the process of step 2702.

[0277] To summarize, the fifth embodiment makes use of: the service execution server list 200 storing therein addresses of servers such as the service execution servers 102 to be managed by the operation management server 100; the application list 300 storing therein information of applications running on servers to be managed; the application role configuration list 400 storing therein information roles in a service system of applications stored in the application list 300; the aggregated application list 500 storing therein information on roles of applications classified by the application aggregating unit; the aggregated application communication list 600 storing therein information on role-to-role communications; and the service system list 700 storing therein service systems to which applications belong. The fifth embodiment through use of the above lists includes the following steps of: specifying application roles in the service systems from the service execution server list 200 and the application list 300 for storage in the application role configuration list 400; acquiring addresses of servers on which applications described in the application list 300 run; attempting communications with the acquired addresses to acquire a list of files that the applications use; specifying applications coincident with the contents of the file list as belonging to the same service system; aggregating the applications on a role-by-role basis from the application list 300 and the application role configuration list 400 in a prejudged service system for storage of the aggregation results into the aggregated application list 500 and the aggregated application communication list 600; and creating an application configuration diagram from the aggregated application list 500 and the aggregated application communication list 600.

[0278] According to the fifth embodiment, even when a plurality of service systems mixedly coexist as objects to be managed by the operation management server 100, application configuration diagrams can be created that allow the system administrator to grasp the current configuration at a single glance, by automatically specifying service systems to which applications belong and providing aggregated displays of the applications on the basis of those service systems.

[0279] It is to be noted that the fifth embodiment may or may not use the configuration change managing unit 109, the setting change managing unit 110, the application configuration baseline 800, the application configuration change list 900, the setting change apply list 1000, the physical/logical (virtual) server list 1200, and the shared application list 1300.

[0280] The operation management server 100 of the fifth embodiment may or may not be provided with the configuration change managing unit 109, the setting change manag-
ing unit 110, the application configuration baseline 800, the application configuration change list 900, and the setting change apply list 1000.

Sixth Embodiment

[0281] While in the first embodiment the roles are specified using only the communication source addresses 304 and the names 303 of applications, more detailed role classification would be feasible by specifying the roles while taking into consideration the communication destination addresses 305 of the applications as well.

[0282] A sixth embodiment of the present invention provides a method of, when there exist a plurality of applications running on the service execution server 102 and having different roles and just the same name 303 and communication source address 304, acquiring communication destination addresses 305 of the applications running on the server, specifying the roles of the applications by the communication destination addresses 305, and aggregating the applications on a role-by-role basis, to thereby create an application configuration diagram.

[0283] The sixth embodiment includes executing a process of specifying roles of the applications from the communication destination addresses 305 of the applications, in addition to the steps 1401 to 1406 (FIG. 15) described in the first embodiment.

[0284] FIG. 32 is a flowchart of a procedure for specifying the roles of applications according to the sixth embodiment.

[0285] The application role specifying unit 105 executes the process (the current layer number j=1) of step 1701 described in FIG. 18 (step 2801).

[0286] The application role specifying unit 105 executes the process of step 1702 (acquiring an application identifier list Applist3 of applications having layer i as layer identifiers described in the application role configuration list, with the current layer number j=1) described in FIG. 18 (step 2802).

[0287] The application role specifying unit 105 judges whether Applist2 is empty (step 2803) and, if Applist2 is empty (Yes at step 2803), terminates the procedure, whereas if Applist2 is not empty (No at step 2803), it executes a process of step 2804.

[0288] The application role specifying unit 105 executes the processes from steps 1704 to 1706 described in FIG. 18 (step 2804).

[0289] The application role specifying unit 105 acquires a communication destination list Addrlist20 of Applist4 from the application list 300 (step 2805).

[0290] The application role specifying unit 105 judges whether Applist4 is not empty (step 2806) and, if Applist4 is empty (No at step 2806), executes a process of step 2810, whereas if Applist4 is not empty (Yes at step 2806), it executes processes from step 2807 onward.

[0291] The application role specifying unit 105 extracts an arbitrary application identifier App20 described in Applist4, deletes App20 from Applist4, and adds it to the application identifier list Applist5 (step 2807).

[0292] The application role specifying unit 105 refers to Addrlist20 to extract an identifier App21 of an application having the same address as that of App20 from Applist4, deletes App21 from Applist4, and adds it to Applist5 (step 2808).

[0293] The application role specifying unit 105 stores the application identifier 401 and the role identifier 403 (role #k) described in Applist5 into the application role configuration list 400, to increment k (step 2809).

[0294] The application role specifying unit 105 increments j (step 2810), returning to the step 2802.

[0295] To summarize, the sixth embodiments makes use of: the service execution server list 200 storing addresses of servers such as the service execution servers 102 to be managed by the operation management server 100; the application list 300 storing information of applications running on servers to be managed; the application role configuration list 400 storing information roles in a service system of applications stored in the application list 300; the aggregated application list 500 storing information on roles of applications classified by the application aggregating unit; and the aggregated application communication list 600 storing information on role-to-role communications. The sixth embodiment includes the following steps of: specifying application roles in the service systems from the service execution server list 200 and the application list 300 for storage in the application role configuration list 400; acquiring communication destinations of applications described in the application list 300; specifying applications having the same communication destination and belonging to the same layer to be of the same role; aggregating the applications on a role-by-role basis from the application list 300 and the application role configuration list 400; storing the results of aggregation into the aggregated application list and the aggregated application communication list 600; and creating an application configuration diagram from the aggregated application list 500 and the aggregated application communication list 600.

[0296] According to the sixth embodiment, even when there exist applications having the same communication source address and the same name but having different roles, more detailed application configuration diagrams can be created by automatically specifying the above applications as being of different roles as long as they have different communication destination addresses, and providing aggregated displays of the applications on a role-by-role basis.

[0297] It is to be noted that the sixth embodiment may or may not use the service system specifying unit 108, the configuration change managing unit 109, the setting change managing unit 110, the service system list 700, the application configuration baseline 800, the application configuration change list 900, the setting change apply list 1000, the physical/logical (virtual) server list 1200, and the shared application list 1300.

[0298] The operation management server 100 of the sixth embodiment may or may not be provided with the service system specifying unit 108, the configuration change managing unit 109, the setting change managing unit 110, the service system list 700, the application configuration baseline 800, the application configuration change list 900, the setting change apply list 1000, the physical/logical (virtual) server list 1200, and the shared application list 1300.

Although the present invention has been described in detail hereinabove based on the embodiments thereof referring to the accompanying drawings, the present invention is not to be construed as being limited to the above embodiments. It is also to be understood that any variants and equivalents fall under the scope of the present invention without departing from the spirit of the present invention.

What is claimed is:

1. An information processing apparatus communicatively coupled via a network to servers on which a plurality of
applications are running for operating a service system, the information processing apparatus comprising:

- a storage unit that stores network addresses of the servers, storing, in a correlated manner, application identification information as information for identifying the applications, active server addresses as network addresses of the servers on which the applications are running, and communication source addresses of the applications; and

- an application aggregating unit

acquiring from the storage unit and comparing the application identification information, the addresses of the servers corresponding thereto, and communication source addresses of the applications, so that when it is judged that the communication source addresses do not coincide with any of the addresses of the servers, the application aggregating unit judges the applications having the application identification information as being applications belonging to a first layer in the service system,

acquiring from the storage unit and comparing application identification information of the applications judged as belonging to the first layer, the active server addresses of the applications, and the application identification information and communication source addresses of applications other than the applications judged as belonging to the first layer, so that the application aggregating unit judges applications having the application identification information correlated to the communication source addresses identical to the active server addresses as belonging to a second layer; and

continuing acquiring from the storage unit communication source addresses of applications other than the applications judged as belonging to the first and the second layer, to repeat the judgment process to judge the layers of each applications contained in the service system.

2. The information processing apparatus of claim 1, wherein the information processing apparatus generates and displays a configuration diagram representing a correlation status among the applications based on the judged layers.

3. The information processing apparatus of claim 2, wherein the configuration diagram of the applications represents the number of applications contained in the layer for each of the judged layers.

4. The information processing apparatus of claim 1, wherein the storage unit stores the name of each of the applications, and the application aggregating unit compares the layer and the name of each of the applications and judges the applications belonging to the same layer and having the same name as being of the same role.

5. The information processing apparatus of claim 4, wherein the storage unit stores communication destination addresses of the applications, and the application aggregating unit acquires from the storage unit the communication destination addresses of the applications having been judged as being of the same role, to judge the applications judged as having different communication addresses to be of different roles.

6. The information processing apparatus of claim 5, wherein when generating and displaying a configuration diagram representing a correlation status among the applications based on the judged layers, the applications judged as having different roles from each other are displayed as different applications from each other even if they belong to the same layer.

7. The information processing apparatus of claim 4, wherein the name of the application is a title contained in a Web page acquired by communicating with the active server of the application.

8. The information processing apparatus of claim 1, wherein the storage unit stores communication destination addresses of the applications, and the application aggregating unit acquires the addresses of the servers, as well as the active server addresses, communication source addresses, and communication destination addresses of the applications from the storage unit, and when it is judged that the communication source addresses do not coincide with any of the addresses of the servers and that the communication destination addresses coincide with one of the addresses of the servers other than the active server addresses of the applications, the application aggregating unit judges the applications as applications belonging to the first layer.

9. The information processing apparatus of claim 1, wherein the storage unit stores, in a correlated manner, the applications and the service system identification information as information for identifying the plurality of service systems, and the application aggregating unit acquires the service system identification information correlated with the applications judged as belonging to the first layer from the storage unit, and judges the layers of each of the applications for the service system specified by the service system identification information.

10. The information processing apparatus of claim 1, wherein the storage unit stores, in a correlated manner, combinations of file names of files stored in the server and identification information of the service systems operated by the server, and the application aggregating unit communicates with the active servers of the applications to acquire file names of all the files stored in the active servers from the active servers, acquires the combinations of the file names from the storage unit to compare the file names acquired from the active servers with the file names acquired from the storage unit, and if coincidence of the two is judged, then judges that the applications are being used in the service system correlated with the combinations of the file names acquired from the storage unit.

11. The information processing apparatus of claim 4, wherein the storage unit stores application identification information as information for identifying applications to be
active on the service system and identification information of currently active applications in the service system, and
the application aggregating unit acquires and compares, between the applications judged as being of the same role, identification information of applications to be active and the identification information of currently active applications, and if the two have a difference, judges the activity of the applications having the role as being defective.

14. The information processing apparatus of claim 2, wherein

display data to display a configuration diagram representing the correlation status among the applications is generated so as to fall within the data size making up one screen of a display device coupled to the information processing apparatus.

15. A method of operating an information processing apparatus communicatively coupled to servers on which a plurality of applications are running for operating a service system, the apparatus having a storage unit that stores network addresses of the servers, wherein

a setting change managing unit that accepts setting change contents information indicative of setting change contents of the application and the role identifiers correlated with the applications executing the setting changes, acquiring from the storage unit addresses of servers on which the applications correlated with the role identifiers run from the storage unit; and
sending setting change contents of the application to the address of the server.

* * * * *