

**SPECIFICATION****Title of the Invention:**

**DATA MANAGEMENT SYSTEM, SERVICE PROVISION SYSTEM, AND METHOD  
FOR EXPANDING FUNCTIONALITY THEREOF**

**Field of the Invention**

[0001]

The present invention relates to a data management system, a service provision system, and a method for expanding functionality thereof, in particular, relates to a technology for overlappingly managing data generated by business processing of an AP server for providing information processing services to a large number of client terminals with a plurality of DB servers and a technology for making co-existent in one casing, AP server functionality in charge of business processing and DB server functionality in charge of management of a database.

**Description of the Related Art**

[0002]

(1) For the sake of preparing against a physical damage of a database by large-scaled disasters such as an earthquake, tsunami, and explosion or human-made disasters such as a nuclear accident and terrorism, previously, business data has been overlappingly registered to a backup database installed at a remote location in a core system of a company.

For example, Patent Literature 1 discloses a technology

that data is registered to a disk device in a main control device, the same data is thereafter sent to a sub-control device installed at a remote location, and registration thereof is requested.

[0003]

As mentioned above, data is double-set by the two main control device and sub-control device. As a consequence, even in the case where an earthquake occurs in one area and the disk device is physically broken, data in another surviving disk device is used, thereby ensuring the continuousness of the business processing.

Further, the damage of a database is not limited to damage in a unit of data center due to an earthquake or the like, and can occur also by a machine trouble of a DB server. Therefore, it is desired to build up such a backup system even in the same campus.

[0004]

Incidentally, in the case where updating processing of a plurality of databases is executed, there might be a contradiction between data if a commitment to all the databases is not established. Therefore, an adjustment system called a two-phased commitment of an XA interface is usually used, and compliance between the databases is ensured.

[0005]

With the two-phased commitment first, a master server that performs the whole processing issues an inquiry about whether the commitment can be executed to each DB server as a

target. The DB server that is finished for updating preparation sends a response of "preparation completion", and it is checked that all DB servers finish the preparation. Thereafter, the master server issues a notification for starting the commitment, and the databases are rewritten all at once. In the case where abnormality has occurred in any of the databases during the rewriting, the DB server in which the abnormality happens issues a notification of failure, and the master server issues a notification of processing rescission to all servers and further issues an instruction to perform "roll back" processing to restore the data (refer to Non Patent Literature 2).

[0006]

With the two-phased commitment, the data compliance between the DB servers is ensured, thereby preventing the occurrence of illegal data without a failure.

However, it is necessary to repeat the exchange of data between the master server and the DB server installed at a remote location. Therefore, high speed of processing becomes victimized. Thus, it is effective to business processing such as ATM transaction in which a trouble does not occur by a waiting time of several seconds. However, there is a problem that it is not applicable to business processing in which the delay of a time less than one second is not permitted, for example, in an online trade.

[0007]

(2) In general, an AP server in charge of the business processing and a DB server for storing various data as a result

of the business processing into a database and managing the data include different server computers, respectively. The AP server is connected to the DB server via a communication network. Therefore, communication overhead due to Ethernet (registered trademark) occurs between the AP server and the DB server, and fast processing is not realized (refer to Non Patent Literature 3).

[0008]

Obviously, a program for the AP server and a program for the DP server are set up in a server computer. Thus, the AP server and the DB server can be conventionally formed in the same casing, thereby realizing elimination of the communication overhead.

However, in this case, there is a scramble of recourses of the computer between the AP server and the DB server. A merit of high speed as a result of solving the communication overhead is mostly offset. Further, it is not expected to dramatically improve the processing speed as a whole.

[0009]

Incidentally, the improvement of processing abilities of a computer previously depends on, mainly, the improvement of the number of clocks of a CPU. However, since problems of heat generation and consumption power occur, after 2003 to 2005, the number of clocks of the CPU peaks out. In place thereof, multi-core of the CPU recently advances even in the case where a CPU core with a relatively low number of clocks is used, four to six CPU cores are stored in one package, and the processing

is set in parallel. Thus, it is expected to handle an increasing amount of data processing.

[0010]

Obviously, in the case of scientific technological calculation, mainly pure operational processing such as DNA analysis or meteorological analysis, a large number of CPU cores are linked, thereby relatively easily improving the processing ability.

On the other hand, in the case of business-system processing that requires a reference on complicated combinations of a large amount of data stored in a database, it takes a time for extracting processing and operational processing of data by the DB server. Therefore, even when increasing the number of CPU cores of the server computer for implementing the AP server and the DB server, it is pointed out that this does not directly lead to increasing the processing speed.

[0011]

(3) In a large-scaled service provision system assuming linkage between a plurality of DB servers and the AP server, when the necessity arises to expand functionality halfway, correction is applied to add a column corresponding to new functionality to a table stored in each DB server and great change needs to be applied to an application program on the AP server side. Thus, there is a problem to easily generate a bug in the process thereof.

[0012]

That is, when the business data stored in the DB server is used, an SQL statement is generally issued to the DB server from the AP server and operational processing such as join-in, matching, or sorting is requested to the DB server. For the request, it is necessary to preliminarily write the SQL statement in the application program for business processing (refer to Non Patent Literature 4).

[0013]

The SQL statement generally has a complicated and long description and the third party thus does not easily understand contents thereof. Further, there is a problem that compile check does not work.

Furthermore, once a table stored in each DB server is corrected, large troublesomeness occurs in the case where a service needs to be returned to a conventional one due to a certain trouble.

Therefore, there is an opinion conflict between an operation section of services for fast correspondence to user's desire and a development section for emphasizing safety of the system, thereby disturbing frequent expansion of functionality.

[0014]

Patent Literature 1: Japanese Unexamined Patent Application Publication No. 11-85408

Non Patent Literature 1: Guide of Web system/First time Configuration of Web site and J2EE server

Internet URL:

<http://www.atmarkit.co.jp/fjava/rensai2/websys01/websys01.html>

Search date: 30th January, 2014

Non Patent Literature 2: Two-phased commitment

Internet URL:

<http://e-words.jp/w/2E79BB8E382B3E3839FE38383E38388.html>

Search date: 30th January, 2014

Non Patent Literature 3: "Wait!" You forget overhead?

Internet URL:

<http://itpro.nikkeibp.co.jp/free/TIS/denwa/20040517/1>

/ Search date: 3rd April, 2014

Non Patent Literature 4: Relationship between program and database

Internet URL:

[http://www.limy.org/prgram/coding/program\\_db.html](http://www.limy.org/prgram/coding/program_db.html)

Search date: 15th July, 2014

## **Summary of the Invention**

### **Problems to be solved by the Invention**

[0015]

The present invention is devised to solve the conventional problem (1). A first object is to realize a technology to certainly save important business data to a plurality of points that is geographically apart from each other without sacrificing high speed of processing.

[0016]

The present invention is devised to solve the

conventional problem (2). A second object is to provide a technology to greatly improve processing speeds of an AP server and a DB server by effectively using a multi-core of a CPU in the case of realizing the AP server and the DB server in the same casing.

[0017]

The present invention is devised to solve the conventional problem (3). A third object is to provide a technology to relatively easily expand functionality of a service provision system under environment in which a plurality of DB servers manages the redundancy of the same business data.

#### **Means for solving the problem**

[0018]

In order to accomplish the first object, a data management system according to claim 1 is a data management system having an AP server and a plurality of DB servers installed in each of a plurality of data centers provided apart from each other at a predetermined distance, wherein all the DB servers respectively include tables that store mutual common data and each table also has a limitation for prohibiting deletion and updating of a record, and wherein each of the AP servers is connected to the respective DB servers installed in the same data center, and is also connected to the respective DB servers installed in another data center, each of the AP servers includes at least one business processing unit, an artificial key management unit, a data control unit, and a plurality of

DB connection units, the business processing unit requests the artificial key management unit to issue an artificial key when generating business data which is a result of business processing, and generates business data with ID as a primary key having a unique artificial key issued by the artificial key management unit, the artificial key management unit refers to an artificial key management table arranged in the same data center and delivers a value of a latest artificial key at the moment to the business processing unit, and executes processing for updating a value of the table at the same time, the data control unit copies business data output from the business processing unit and outputs the data to each of the DB connection units, and also outputs a notification of registration completion to the business processing unit after checking that notifications of reception completion from all the DB servers are collected when the notifications of reception completion from the respective DB servers via the respective DB connection units are sent, each of the DB connection units has set therein specific information of a DB server of which the each DB connection unit is in charge, each DB connection unit issues an SQL to a DB server of which the each DB connection unit is in charge to request addition of a record when the data control unit outputs business data, and also, when a notification of reception completion is sent from the DB server, executes processing for outputting the notification to the data control unit, further, an identification code of an AP server is set to a predetermined position of each of the artificial keys, and

the business processing unit refrains from referring to the business data until the notification of registration completion is delivered from the data control unit.

[0019]

A data management system according to claim 2 is the system of claim 1, and further the business data includes a combination of a plurality of pieces of substantial data having a mutual relation generated as a result of business processing by the business processing unit and integrating data generated for bundling each data, the integrating data has only a data item for storing an integrating ID having an artificial key as a primary key, and each piece of the substantial data includes a data item for storing the substantial data, a data item for storing a unique ID having an artificial key as a primary key, and a data item for storing an ID of the integrating data as an foreign key.

[0020]

A data management system according to claim 3 is the system of claim 1 or 2, and further the respective AP servers and the respective DB servers are connected with each other via a plurality of connections, the DB connection unit is provided in each AP server per connection, and the business data is sent to each DB server in parallel via each DB connection unit and connection.

[0021]

A data management system according to claim 4 is the system of claims 1 to 3, and further the business processing unit newly

generates business data including a data item for storing an ID of the business data, a data item for storing a unique ID having an artificial key as a primary key, and a data item for storing substantial data after correction, when existing business data needs to be updated, and substantially updates existing business data by additionally registering the business data to an existing table provided in each DB server via the data control unit and each DB connection unit.

[0022]

A data management system according to claim 5 is the system of claims 1 to 4, and further the business processing unit generates business data for data cancel including a data item for storing an ID of the business data and a data item for storing a unique ID having an artificial key as a primary key, when existing business data needs to be deleted, and substantially deletes the existing business data by additionally registering the business data for data cancel to a table dedicated for cancel provided in each DB server via the data control unit and each DB connection unit.

[0023]

In order to accomplish the second object, a data management system according to claim 6 includes: a computer including a CPU package incorporating a plurality of CPU cores and an external storage device that stores a plurality tables; an AP server configured by running a program for AP server on the computer; and a plurality of DB servers configured by running a plurality of programs for DB server on the computer,

wherein the external storage device separately stores a plurality of tables that stores mutually common data per DB server, and a plurality of different CPU cores is dedicatedly assigned to the AP server and each DB server, respectively, wherein the AP server includes at least a function of specifying a kind of data necessary for executing business processing, a function of requesting extraction of necessary data to any of the DB servers, a function of performing necessary operational processing to data obtained from the DB server, and a function of simultaneously requesting all the DB servers to register data generated by the business processing, and wherein the AP server issues an SQL in a state of specifically designating a CPU core that is in charge of the processing when requesting the DB server to extract or register data, and the designated CPU core executes corresponding processing in the DB server having received the SQL.

[0024]

A data management system according to claim 7 is the system of claim 6, and further the computer includes a plurality of the CPU packages, a dedicated local memory is assigned to each CPU package, a CPU core belonging to a separate CPU package is assigned to each of the AP server and at least a part of the DB servers, and the AP server and each DB server have such a limitation that it is possible to access only the local memory and it is not possible to access the local memory assigned dedicatedly to another CPU package.

[0025]

A data management system according to claim 8 is the system of claim 6 or 7, and further the AP server has such a limitation that it is allowed only to request each of the DB servers to insert or extract data and it is prohibited to request each of the DB servers to update or delete data.

[0026]

In order to accomplish the third object, a service provision system according to claim 9 includes an AP server and a plurality of DB servers connected to the AP server, and providing a predetermined information processing service to a client terminal, wherein the AP server includes a plurality of DB connection units provided per each of the DB servers, a business processing unit, and a DB control unit, each of the DB connection units has set therein specific information of a DB server of which the each DB connection unit is in charge, a table for storing mutually common data is provided in each of the DB servers, each table has such a limitation that prohibits deletion or updating of a record, the DB control unit, when having received a request for registering data from the business processing unit, copies and delivers the data to all the DB connection units, each DB connection unit having received the data issues an SQL statement to a DB server of which the each DB connection unit is in charge to request addition of a record, the DB control unit specifies one or a plurality of DB servers from among the DB servers and makes a request for extracting data to a DB connection unit that is in charge of the DB server, when a reference to specific data is requested

from the business processing unit, the DB connection unit having received the request issues an SQL statement to a DB server of which the DB connection unit is in charge to request extraction of a record, and also delivers the data received from the DB server to the DB control unit, and the DB control unit performs necessary operational processing on the received data and then delivers the resultant data to the business processing unit.  
[0027]

A service provision system according to claim 10 is the system of claim 9, and further the business processing unit newly generates data having a value after updating when existing data needs to be updated, and additionally registers the data to an existing table provided in each DB server via the DB control unit and each DB connection unit, thereby realizing substantive updating of the existing data.  
[0028]

A service provision system according to claim 11 is the system of claim 9 or 10, and further the business processing unit generates data for data cancel including a data item for storing an ID of the business data when existing business data needs to be deleted, and additionally registers the data for data cancel to a table dedicated for cancel provided in each DB server via the DB control unit and each DB connection unit, thereby realizing substantive deletion of the existing data.  
[0029]

A service provision system according to claim 12 is the system of claims 9 to 11, and further the AP server and at least

two or more DB servers of the DB servers are formed on a same computer, the computer includes a CPU package incorporating a plurality of CPU cores and a storage device that stores a plurality of tables, the AP server is configured by running a program for AP server on the computer, and the business processing unit, the DB control unit, and a plurality of DB connection units are configured by running a dedicated application program on the AP server, each of the DB servers is configured by running a plurality of programs for DB server on the computer, a plurality of different CPU cores is dedicatedly assigned to the AP server and each DB server, respectively, an SQL statement is issued in a form of specifically designating a CPU core that is in charge of the processing from the DB connection unit when extraction and registration of data is requested to the DB server, and the designated CPU core executes corresponding processing in the DB server having received the SQL statement.

[0030]

An method for expanding functionality of a service provision system according to claim 13 includes: when expanding functionality of a service provided to a client terminal in the system of claim 9, adding a table corresponding to functionality after expansion to each of the DB servers and setting a table related to functionality before expansion to an out-of-target of processing while leaving the table in each DB server, and the business processing unit, the DB control unit, and each DB connection unit of the AP server are replaced with an expansion

version corresponding to the added table.

### **Effects of the Invention**

[0031]

In the case of a data management system according to claim 1, an AP server and a plurality of DB servers are installed in a plurality of data centers that is apart in distance, respectively. Further, business data generated by the AP server is simultaneously sent to not only the plurality of DB servers installed in the same data center but also the plurality of DB servers installed in another data center. Thus, it is possible to improve the maintainability of the business data.

[0032]

Furthermore, deletion or updating of the business data requiring complicated processing for data recovery is prohibited in case of trouble. Only additional registration capable of restoring data simply by only copying shortage data from another DB server is permitted. Therefore, a strict adjustment system such as a two-phased commitment does not need to be used, and it is possible to admit the registration completion when a reception notification is simply returned from the DB server. Consequently, it is possible to drastically reduce the number of messages between the AP server and the DB server, and thus the high speed of data backup can be ensured.

[0033]

Not only a plurality of DB servers but also the AP server is installed in each data center, the business processing is

executed in each data center also at normal time, which contributes to load distribution of the business processing.

An artificial key including an identification code of each AP server is assigned as an ID to the business data generated in the each AP server. Therefore, even in the case of separately sending the business data from each AP server to all the DB servers installed in the data center, the IDs are not overlapped from each other and there is not a danger that the ID becomes an error on the DB server.

[0034]

In the case of a data management system according to claim 2, business data is divided into a plurality of pieces of substantial data. The business data is easily stored in a frame size of Ethernet (registered trademark), and the transfer efficiency between an AP server and a DB server is improved.

Each piece of substantial data is associated with each other with an ID of integrating data that is commonly held. Therefore, a business processing unit can handle the divided substantial data as a series of business data.

[0035]

In the case of a data management system according to claim 3, an AP server and each DB server are connected with a plurality of connections and business data is simultaneously sent in parallel via each connection. Therefore, the transfer efficiency is further improved between the AP server and the DB server.

[0036]

With the invention, a table of the DB server has a limitation to prohibit the deletion and updating of a record. By use of a structure according to claims 4 and 5, the substantive deletion and updating of the business data are realized.

[0037]

In the case of a data management system according to claim 6, separate CPU cores are dedicatedly assigned to an AP server and a plurality of DB servers activated on a computer, respectively. Therefore, there is not a competition of CPU resources if the AP server and the plurality of DB servers are made to coexist in the same casing.

Further, each DB server has a table independent of each other and common data is stored in each table. Therefore, the AP server simultaneously requests a plurality of DB servers to extract different data, thereby totally improving the processing speed.

In addition, when an SQL of data extraction is issued to the DB server from the AP server, the CPU core that is in charge of the processing is designated. Therefore, the potential of the multi-cored CPU can be pulled out maximally. That is, since the AP server knows the sequence of operation and a kind of data necessary for business processing, the AP server designates a different CPU core and requests it to extract different data on a scene in which concurrent processing is possible, thereby improving the operating rate of the CPU core.

The AP server, because of having a function of performing operation on the data extracted from the DB server, even in a

case of separately extracting data from a plurality of DB servers, can effectively execute processing for coupling, sorting, or matching each data.

[0038]

In the case of a data management system according to claim 7, a memory available for an AP server and each DB server is limited to a local memory and access to a remote memory is prohibited. Therefore, high speed of processing is realized.

[0039]

In the case of a data management system according to claim 8, as a consequence of simplifying processing contents on the DB server side, it is possible to further improve response performance from each DB server.

[0040]

In the case of a service provision system according to claim 9, a table stored in each DB server has a limitation to prohibit deletion and updating of a record, an SQL statement issued from the AP server is limited to that requesting extraction or registration of data, operational processing of the extracted data is executed on the AP server side, and the operational processing does not need to be instructed to the DB server. Therefore, the SQL statement does not need to be written in a program for realizing a business processing unit and a DB control unit. As a result, the development efficiency of the programs is improved and services can be released earlier and functionality can be frequently expanded.

Further, even if the SQL statement needs to be written

in the program for realizing a DB connection unit, the content thereof is limited to that of simply extracting or registering data to the DB server of which the DB connection unit is in charge. Thus, data does not need to be updated every time functionality of service is expanded.

[0041]

In the case of the invention, a table of the DB server has a limitation to prohibit deletion and updating of a record. However, by use of a structure according to claims 10 and 11, substantive deletion and updating of data can be realized.

[0042]

In the case of a service provision system according to claim 12, an AP server and a plurality of DB servers are activated on the same computer, and when an SQL statement for data extraction is issued to a DB server from each DB connection unit of the AP server, a CPU core that is in charge of the processing is designated. Therefore, the potential of a multi-cored CPU can be pulled out maximally and fast processing is possible.

[0043]

With a method for expanding functionality of a service provision system according to claim 13, in expansion of functionality, a table corresponding to functionality after expansion is newly provided in each DB server. A service is to shift to provision of a service based on the new table. Further, an existing table is maintained as-is without addition of a column. Thus, fast correspondence is possible when services need to be returned to previous ones because of some

circumstances.

### **Best mode for carrying out the Invention**

[0044]

Fig. 1 is a diagram showing the entire configuration of a data management system 110 according to the invention.

First, in a first data center 112 installed in Tokyo, a first AP server 114, a first DB server 116, and a second DB server 118 are arranged.

Further, in a second data center 120 installed in Osaka, a second AP server 122, a third DB server 124, and a fourth DB server 126 are arranged.

[0045]

The first AP server 114 and the second AP server 122 have the same application program, respectively, and further have a function of executing the same business processing in response to a request sent from a client terminal 128 via a communication network 127 such as the Internet.

[0046]

Note that the number of AP servers is not limited. A plurality of AP servers having the same functionality can be set in each data center, and load can be distributed via a load balancer (not shown).

As mentioned above, provision of a plurality of AP servers in the same data center, even at the time of planned stop or unplanned stop of any AP server, allows replacement with the remaining AP server in the same data center without switching

to an AP server in another data center, thereby minimizing the down time.

[0047]

The first DB server 116, the second DB server 118, the third DB server 124, and the fourth DB server 126 have a common table, respectively.

The number of DB servers is not limited, and three or more DB servers can be installed in the data center.

[0048]

The first AP server 114 is connected to the first DB server 116 and the second DB server 118 via LAN, and is further connected to the third DB server 124 and the fourth DB server 126 installed in the second data center 120 via a communication line 129.

The second AP server 122 is connected to the third DB server 124 and the fourth DB server 126 via LAN, and is further connected to the first DB server 116 and the second DB server 118 installed in the first data center 112 via the communication line 129.

[0049]

At normal time, a client terminal 128 installed in the eastern Japan is connected to the first AP server 114 installed in the first data center 112. A client terminal 128 installed in the western Japan is connected to the second AP server 122 installed in the second data center 120.

[0050]

On the other hand, when a large earthquake occurs in the capital region, for example, and the first data center 112 is

seriously damaged, the connection destination of the client terminal 128 installed in the eastern Japan is switched to the second AP server 122 installed in the second data center 120 with setting change of a DNS server (not shown). Therefore, the continuousness of services to users in the eastern Japan can be ensured.

Similarly, when the second data center 120 is damaged, the connection destination of the client terminal 128 installed in the western Japan is switched to the first AP server 114 installed in the first data center 112.

[0051]

As mentioned above, each data center provides services to a near client terminal 128 at normal time, while services are provided to the client terminals 128 throughout Japan at disaster time. Therefore, each DB server needs to mutually keep data with the same contents as usual.

[0052]

Fig. 2 shows internal configurations of the first AP server 114 and the second AP server 122, each including a business processing unit 130, an artificial key management unit 132, a data control unit 134, and a plurality of DB connection units 138.

[0053]

The business processing unit 130 executes registration processing, search processing and the like of data in response to a request from the client terminal 128, and is realized with an operation of a CPU of the AP server according to a dedicated

application program.

[0054]

The artificial key management unit 132 has a function of issuing an artificial key (surrogate key) in response to a request from the business processing unit 130 (the details will be described later). The artificial key management unit 132 is realized with an operation of the CPU of the AP server according to a dedicated middleware.

[0055]

The data control unit 134 copies data by the number of DB servers when having received an instruction for data registration from the business processing unit 130, and performs a function of delivering the copied data to the DB connection units 138, and is realized with an operation of the CPU of the AP server according to the dedicated middleware.

Further, the data control unit 134 has a function of issuing an instruction for updating artificial key management data to a DB connection unit 138 that is in charge of each DB server installed in the same data center in response to an instruction of the artificial key management unit 132 (the details will be described later).

[0056]

Each DB connection unit 138 has a function of issuing an SQL statement to the DB server preliminarily assigned to the each DB connection unit 138 and further issuing an instruction for registering or extracting data, and is realized with an operation of the CPU of the AP server according to the dedicated

middleware.

[0057]

In the system 110, data generated by the business processing unit 130 has the following features.

(1) Each data has an ID with a unique artificial key.

(2) The number of items of substantial data having a business meaning, generated as a result of business processing is limited to one. Therefore, when data on a plurality of mutually related items is generated as a result of the business processing, the data is divided into a plurality of pieces of independent data.

(3) The plurality of mutually related data has in common an ID (integrating ID) of integrating data (parent data) for bundling each data as an foreign key, thereby expressing a mutual relationship.

[0058]

A specific description will be given of features of the data based on Fig. 3.

For example, when the business processing unit 130 generates corresponding reservation data in response to a reservation request from the client terminal 128, with a conventional system, reservation data 140 is generated, including items of reservation ID, shop ID, customer ID, amount of money, and year, month, and day, as shown in Fig. 3(b). Values corresponding to one specific reservation request are filled to the items of the shop ID, customer ID, amount of money, and year, month, and day with a mutual relationship thereof.

[0059]

On the other hand, with the system 110, as shown in Fig. 3(a), the business processing unit 130 generates five pieces of data of reservation integrating data 141, reservation shop data 142, reservation customer data 143, reservation money-amount data 144, and reservation data 145 on year, month, and day, in response to one reservation request.

[0060]

The "reservation integrating data 141" is upper-conceptual "parent data" for bundling the reservation shop data 142, the reservation customer data 143, the reservation money-amount data 144, and the reservation data 145 on year, month, and day, has only data item to which a unique artificial key "reservation integrating ID" is filled as a primary key (PK), and does not have a substantial data item that is meaningful in view of business processing.

[0061]

On the other hand, the "reservation shop data 142" also has a substantial data item "shop ID" to which specific information on a shop for reservation is stored in addition to a data item to which a unique artificial key "reservation shop ID" is filled as a primary key (PK). Further, the "reservation shop data 142" has a data item "reservation integrating ID" for storing a reservation integrating ID to be parent data of the "reservation shop data 142" as an foreign key (FK1).

[0062]

Similarly, the "reservation customer data 143" has a

substantial data item "customer ID" to which specific information on a reservation customer is stored in addition to a data item to which a unique artificial key "reservation customer ID" is filled as a primary key (PK). Further, the "reservation customer data 143" also has the data item "reservation integrating ID" for storing a reservation integrating ID to be parent data of the "reservation customer data 143" as an foreign key (FK1).

[0063]

The "reservation money-amount data 144" also has a substantial data item "amount of money" to which an amount of money of a reservation service is stored in addition to a data item to which a unique artificial key "reservation money-amount ID" is filled as a primary key (PK). Further, the data item "reservation integrating ID" for storing a reservation integrating ID to be parent data of the "reservation money-amount data 144" is set as an foreign key (FK1).

[0064]

The "reservation data 145 on year, month, and day" also has a substantial data item "year, month, and day" for storing reservation date in addition to a data item to which a unique artificial key "reservation ID on year, month, and day" is filled as a primary key (PK). Further, a data item "reservation integrating ID" for storing a reservation integrating ID to be parent data of the "reservation data 145 on year, month, and day" is set as an foreign key (FK1).

[0065]

As mentioned above, the reservation shop data 142, the reservation customer data 143, the reservation money-amount data 144, and the reservation data 145 on year, month, and day have the data item "reservation integrating ID", respectively, as an foreign key, to which the common reservation integrating ID is filled thereto. Thus, the business processing unit 130 refers to the reservation integrating ID of each data, thereby making it possible to integrate pieces of divided data and handle the integrating data as one piece of reservation data having mutually related pieces of data (refer to Fig. 3(b)).

[0066]

Note that the reservation integrating data 141, reservation shop data 142, reservation customer data 143, reservation money-amount data 144, and reservation data 145 on year, month, and day are respectively stored in a reservation integrating table 150, a reservation shop table 151, a reservation customer table 152, a reservation money-amount table 153, and a reservation table 154 on year, month, and day that are commonly provided in the first DB server 116, second DB server 118, third DB server 124, and fourth DB server 126.

[0067]

Herein, a description is given of a structure for assigning an artificial key in the system 110 based on a flowchart in Fig. 4.

First, the business processing unit 130 that needs to generate data as a result of the business processing designates a table and requests the artificial key management unit 132 to

issue an artificial key (S110). In the case of generating the reservation data, the business processing unit 130 requests to issue the artificial keys for the "reservation integrating ID", "reservation shop ID", "reservation customer ID", "reservation money-amount ID", and "reservation ID on year, month, and day".

[0068]

The artificial key management unit 132 having received the request refers to an artificial key management table 156 that is stored in the DB server installed in the same data center (S112), and issues a notification of the latest artificial keys thereof to the business processing unit 130 (S114).

[0069]

As shown in Fig. 5, the artificial key management table 156 has data items "table ID", "AP server ID", and "next key value", and manages the latest artificial key (next key value) per "table ID" + "AP server".

[0070]

Herein, the "artificial key" has a numerical value with a predetermined length (e.g., 32 bits or 64 bits).

An identification code (any numerical value of 0 to 9) for specifying the AP server that has issued an ID is set to the last digit (ending) of the numerical value.

Specifically, a relationship between an ID of each AP server and a numerical value of 0 to 9 is defined in a last-digit management table 157 of the artificial key, as shown in Fig. 5.

A relationship between each AP server and a data center

is defined in an AP server management table 158 and a data center management table 159.

[0071]

The artificial key management unit 132 that has completed the notification of the latest value of the artificial key to the business processing unit 130 updates a value of each record in the artificial key management table 156 (S116).

Specifically, a "value obtained by adding one to a numerical value having a part except for the identification code at the ending in a previous value (the latest value)" is set to the "next key value" in the reservation integrating table 150, reservation shop table 151, reservation customer table 152, reservation money-amount table 153, and reservation table 154 on year, month, and day, related to the AP server.

[0072]

However, an updating method of the latest value of the artificial key is not limited to such "ascending order". "Descending order" may be used to set a "value obtained by subtracting one from a numerical value of a part except for the identification code at the ending of a previous value (the latest value)". The identification code of the AP server also is not limited to a numerical value of "0 to 9" and another kind of character may also be used. Furthermore, the identification code may be inserted to the head of the artificial key.

Moreover, only a numerical value except for the identification code of the AP server is preliminarily stored in the "next key value" in the artificial key management table

156, and the artificial key management unit 132 can also add a corresponding identification code to the ending of the numerical value and issue the result to the business processing unit 130.

[0073]

The business processing unit 130 having received the latest artificial key from the artificial key management unit 132 generates data set to a primary key or an foreign key by using the artificial key as an ID (S118).

The data is sent to all the DB servers via the data control unit 134 and the DB connection units 138, and is inserted to a corresponding table provided in each of the DB servers.

[0074]

As mentioned above, for the artificial key, the uniqueness is ensured per "AP server × table", and each AP server is linked to a specific data center in the data center management table 159. Therefore, even if the artificial key is uniquely issued at each AP server, there is no danger of issuing an overlapped artificial key and the uniqueness of each data is certainly ensured because the uniqueness is ensured to a numerical value of the last digit.

[0075]

Note that a request to issue the artificial key issued from the business processing unit 130 in the AP server is once stored in a queue, and is processed under FIFO (first in, first out) rule. Therefore, there is no danger of erroneously issuing the artificial key of a small value to data generated later,

and the order is ensured.

[0076]

The data generated by the business processing unit 130 of each AP server is stored in the table in each DB server, as described above. Each table preliminarily has a limitation for prohibiting the deletion (delete) and updating (update).

In short, with the system 110, only insertion (insert) and selection (refer) are permitted to each table.

[0077]

Therefore, in a case where existing data needs to be set invalid, the same state is realized by inserting a new record into another table in place of deleting the record.

[0078]

For example, in the case of invalidating one reservation data (reservation integrating data + reservation shop data + reservation customer data + reservation money-amount data + reservation data on year, month, and day) shown in Fig. 3, as shown in Fig. 6(a), respective pieces of data for reservation cancel of cancel integrating data 160 + cancel target data 161 + cancel data 162 on year, month, and day are inserted into a cancel integrating table 163, a cancel target table 164, and a cancel table 165 on year, month, and day, respectively.

[0079]

The "cancel integrating data 160" is upper-conceptual "parent data" for bundling the cancel target data 161 and the cancel data 162 on year, month, and day, and includes only a data item to which a unique artificial key "cancel integrating

ID" is filled as a primary key (PK), and does not include a substantial data item that is meaningful in business processing.

[0080]

On the other hand, the "cancel target data 161" includes a substantial data item "cancel target" to which a reservation integrating ID of reservation data to be a cancel target is stored, in addition to a data item "cancel target ID" to which a unique artificial key is filled as a primary key (PK). Further, the "cancel target data 161" includes a data item "cancel integrating ID" for storing a cancel integrating ID to be parent data of the "cancel target data 161" as an foreign key (FK1).

[0081]

The "cancel data 162 on year, month, and day" also includes a substantial data item "year, month, and day" to which year, month, and day of cancel is stored, in addition a data item "cancel ID on year, month, and day" to which a unique artificial key is filled as a primary key (PK). Further, the "cancel data 162 on year, month, and day" includes a data item "cancel integrating ID" for storing the cancel integrating ID to be parent data of the "cancel data 162 on year, month, and day" as an foreign key (FK1).

[0082]

As mentioned above, the cancel target data and the cancel data on year, month, and day include the data item "cancel integrating ID", respectively, as an foreign key. A common cancel integrating ID is filled to the cancel target data and

the cancel data on year, month, and day, respectively, and the business processing unit 130 therefore can integrate the divided pieces of data and handle the resultant data as one piece of reservation cancel data 166 (refer to Fig. 6(b)) having mutually related data by referring to the cancel integrating ID of each data.

[0083]

After selecting the reservation data from the reservation integrating table 150, reservation shop table 151, reservation customer table 152, reservation money-amount table 153, and reservation table 154 on year, month, and day, the business processing unit 130 refers to the cancel integrating table 163, cancel target table 164, and cancel table 165 on year, month, and day, and when the "integration cancel ID" of the reservation data is stored in the "cancel target" of the cancel target data 161, the reservation data is set invalid and excluded.

[0084]

Note that the cancel integrating data 160, cancel target data 161, and cancel data 162 on year, month, and day are respectively stored in the cancel integrating table 163, cancel target table 164, and cancel table 165 on year, month, and day, which are commonly provided in the first DB server 116, second DB server 118, third DB server 124, and fourth DB server 126.

[0085]

In a case where correction of the registered data is required in the system 110, the case is handled by newly registering the record having data after the correction in place

of updating the existing record.

For example, as shown in Fig. 7, regarding a record 167 of "reservation shop ID: 12345678901" + "reservation integrating ID: 22345678901" + "shop ID: M012356" stored in the reservation shop table, in a case where the shop ID needs to be changed to "shop ID: M012357", the business processing unit 130 inserts a new record 168 including the shop ID after the correction in place of changing a value of the existing record 167.

[0086]

A value equal to the existing record 167 is set as a reservation integrating ID to the new record 168. On the other hand, a reservation shop ID (22345678901) with a value larger than that of the existing reservation shop ID is assigned to the new record 168. Therefore, the business processing unit 130 recognizes that the new record 168 indicates a value after the correction, and can realize the substantive updating of data by inserting the record.

Note that, for the sake of convenience in shown figures, referring to Fig. 7, a shorter ID is shown as an example than the artificial key actually used as "reservation shop ID: 12345678901", "reservation integrating ID: 22345678901", and "reservation shop ID 24567890141".

[0087]

Data generated by the business processing unit 130 is copied by the number of DB servers by the data control unit 134, and is thereafter sent all at once to all the DB servers via

the DB connection units 138 that are specifically in charge of the DB servers.

Fig. 8 shows a state of sending "(1) reservation integrating data 141", "(2) reservation shop data 142", "(3) reservation customer data 143", "(4) reservation money-amount data 144", and "(5) reservation data 145 on year, month, and day" to the first DB server 116 to fourth DB server 126.

[0088]

In this case, any arrival order to the DB server of each data may be used. If "(5) reservation data 145 on year, month, and day" as child data arrives at the DB server prior to "(1) reservation integrating data 141" as parent data, this does not cause an error.

This is because the "reservation integrating ID" of the "(1) reservation integrating data 141" is assigned as an foreign key to each child data and the business processing unit 130 can correctly bundle each data finally even if the arrival order is reversed.

[0089]

Each DB server sends a notification of reception completion to the DB connection unit 138 when having received data from the DB connection unit 138 and stored it in a memory of the own DB server. Thereafter, the DB server stores the record corresponding to the table provided in an external storage device such as a hard disk or SSD.

[0090]

The notification of reception completion sent from the

DB server is collected to the data control unit 134 via each DB connection unit 138.

The data control unit 134 outputs a notification of registration completion to the business processing unit 130 when a notification of reception completion is returned for all data of "(1) reservation integrating data 141", "(2) reservation shop data 142", "(3) reservation customer data 143", "(4) reservation money-amount data 144", and "(5) reservation data 145 on year, month, and day" from all the DB connection units 138.

[0091]

When having received the notification of registration completion from the data control unit 134, the business processing unit 130 recognizes that the data is a reading target. Conversely, the business processing unit 130 sets the data as an out-of-target for reading until the notification of registration completion is returned, thereby preventing a danger of reading illegal data due to shift of registration timing at each DB server.

[0092]

However, in a case where the notification of reception completion is not returned from any of the DB servers even after elapse of a predetermined time, the data control unit 134 admits that a communication fault or a machine trouble has occurred, and shifts the DB server to an off-line mode.

Specifically, the DB server is temporarily disconnected from the system 110, and shifts to an operation mode based on

only the remaining DB servers.

That is, when the notification of reception completion has been received from all the DB servers other than the disconnected DB server, the notification of registration completion is output to the business processing unit 130 from the data control unit 134, and the registered data is a reading target.

[0093]

During the above, the disconnected DB server is inspected, and a necessary recovery countermeasure is performed. For example, when it is determined that the notification of reception completion has not been received due to a fault of a communication device, the DB server is set to a write mode (in a state of permitting only the writing of data and prohibiting the reference of the data) after replacement to a new communication device. Differential data is copied from another DB server installed in the same data center.

[0094]

When the stored data reached the latest state, the DB server is set to a read/write mode (in a state of permitting writing and reference of the data), and is re-connected to the system 110.

Subsequently, the data control unit 134 restarts sending and reception of the data via the DB connection unit 138 that is in charge of the DB server.

[0095]

In the case of the system 110, such a rule is applied that

only insertion of each business data is permitted and the deletion and updating are not permitted. Therefore, data recovery is extremely easy.

That is, when it is premised that the deletion and updating of data are permitted, in order to restore the data, it is necessary to reproduce the addition, deletion, and updating of data in order from one timing based on updating history information held by the DB server, and it takes a long time and large load for this. Further, it is required to provide a structure for certainly storing an updating log on the DB server side.

[0096]

On the other hand, in a case where it is premised that only the insertion of data is permitted and the deletion and updating are prohibited like the system 110, it is sufficient to compare the data registered in another normal DB server with the data registered in the DB server in which a fault has occurred, and simply to copy shortage data. The order (relationship of before/after) of data does not need to be considered.

[0097]

With the system 110, when the data sent from the DB connection unit 138 is stored to the memory in the DB server as mentioned above, the notification of reception completion is issued to the AP server from the DB server. Therefore, the business processing unit 130 can set the data as a reading target for an extremely short time.

[0098]

With the conventional common sense, the memory is a volatile storage unit so that data is lost when power supply stops, and a completion notification is to be returned when the data is stored to a non-volatile storage unit (hard disk, etc.).

On the other hand, in the case of the system 110, even when the data is damaged in one DB server, simply the copying operation of shortage data from an adjacent DB server catches up with the operation. Therefore, the operation does not need to wait until the data is stored to a hard disk.

[0099]

The data recovery is easy in this way. When the same data is stored to a plurality of DB servers, a conventional troublesome procedure such as the two-phased commitment is not required to be taken. Registration is simply requested to each DB server and reading is possible when the notification of reception completion is returned from the each DB server (so-called auto-commitment is realized).

As a result, the number of exchanges of messages between the AP server and the DB server is extremely reduced, and the general processing speed is improved.

[0100]

In the case of the system 110, the data generated by the business processing unit 130 is divided as described above and the capacity of one piece of transmission data is minimized. Therefore, a large amount of data can be easily stored in a frame size of the Ethernet (registered trademark), and the transfer efficiency between the AP server and the DB server can be

improved.

[0101]

As shown in Fig. 9, a plurality of connections 171 to 175 is formed between the AP server and the DB server, and data divided via the respective connections and the dedicated DB connection unit 138 can be sent concurrently. Further, the transfer efficiency between the AP server and the DB server can be improved.

[0102]

However, when the data generated by the business processing unit 130 is divided into units of data item and parallel processing using a large number of DB connection units 138 is executed, processing on the AP server side will be inevitably complexed. This can be handled by multi-core of the AP server side.

[0103]

That is, the computer mounting a large number of CPU cores is used as the AP server, a large number of threads are activated with functionality of an OS or middleware, and a multi-task system is employed that realizes the business processing unit 130, the artificial key management unit 132, and the large number of DB connection units 138 with each thread, thereby achieving improvement in efficiency of the processing on the AP server side.

[0104]

Fig. 10 shows a corresponding relationship between a CPU core 180 and a thread 181. A thread pool 182 is provided to

each thread 181, and it is imaged that a plurality of tasks 183 arranged there is sequentially processed by the thread 181.

The thread 181 in the diagram functions as the business processing unit 130, the artificial key management unit 132, the data control unit 134, and the DB connection units 138 of each AP server, and specific processing executed by each functional components corresponds to the task 183.

[0105]

Each thread 181 executes the tasks 183 stored in a thread pool 182 sequentially in chronological order. If the thread pool 182 of the thread 181 is empty, the tasks 183 stored in the thread pool 182 of another thread 181 are executed in new order.

[0106]

Note that, according to the invention, the dividing of the business data is not a requirement element. It may be configured such that a plurality of pieces of substantial data mutually-related is collected as one piece of data, and the business processing unit 130 generates business data (refer to Figs. 3(b) and 6(b)) obtained by adding an ID having the artificial key to the collected data, and sends the data to each DB server via the data control unit 134 and the plurality of DB connection units 138.

[0107]

The business processing unit 130 marks a time stamp with precision of millisecond to various kinds of the business data (reservation integrating data 141, reservation shop data 142,

reservation customer data 143, reservation money-amount data 144, reservation data 145 on year, month, and day, and the like). Therefore, even if a value of the artificial key reaches MAX and ID overlapped to the issued ID is re-issued (recycled), the business processing unit 130 compares the time stamps with each other, thereby determining whether the data is new or old.

[0108]

As described above, the description is developed on the premise that the AP server is physically independent of the DB server. Within the same data center, the AP server and the DB server can be formed by a server computer in the same casing.

[0109]

Fig. 11 shows a hardware structure for realizing another data management system 210 according to the invention. In a server computer 211, a mother board 220 is stored which includes a first CPU package 212, a second CPU package 214, a third CPU package 216, and a fourth CPU package 218.

[0110]

Fifteen CPU cores are formed in each CPU package. Therefore, the server computer 211 includes the total sixty CPU cores.

Each CPU package is attached to a CPU socket (not shown) provided on the mother board 220.

[0111]

A dedicated memory (local memory) is assigned to each CPU package.

That is, dedicatedly, a first memory 222 is provided to

the first CPU package 212, a second memory 224 is provided to the second CPU package 214, a third memory 226 is provided to the third CPU package 216, and a fourth memory 228 is provided to the fourth CPU package 218.

[0112]

An external storage device 230 having an SSD (Solid State Drive) is connected to the server computer 211. The external storage device 230 stores various tables (details will be described in later). In place of the SSD, a hard disk can constitute the external storage device 230.

[0113]

Peculiar functionality is preliminarily assigned to each CPU core.

That is, an AP server 232 is dedicatedly assigned to cores (01) to (15) formed in the first CPU package 212.

[0114]

A kernel thread (OS) 234 is dedicatedly assigned to a core (16) formed in the second CPU package 214.

A first DB server 236 is dedicatedly assigned to cores (17) to (23) formed in the second CPU package 214.

Similarly, a second DB server 238 is dedicatedly assigned to cores (24) to (30) formed in the second CPU package 214.

[0115]

A third DB server 240 is dedicatedly assigned to cores (31) to (38) formed in the third CPU package 216.

Similarly, a fourth DB server 242 is dedicatedly assigned to cores (39) to (45) formed in the third CPU package 216.

[0116]

A fifth DB server 244 is dedicatedly assigned to cores (46) to (53) formed in the fourth CPU package 218.

Similarly, a sixth DB server 246 is dedicatedly assigned to cores (54) to (60) formed in the fourth CPU package 218.

[0117]

As mentioned above, in one server computer 211, one program for AP server and six programs (instances) for DB server are activated. A plurality of CPU cores is dedicatedly assigned to each server.

[0118]

As mentioned above, as local memories, a first memory 222 is dedicatedly assigned to the AP server 232, a second memory 224 is dedicatedly assigned to the kernel thread (OS) 234, the first DB server 236, and the second DB server 238, a third memory 226 is dedicatedly assigned to the third DB server 240 and the fourth DB server 242, and a fourth memory 228 is dedicatedly assigned to the fifth DB server 244 and the sixth DB server 246.

In this case, each CPU core can access a local memory directly (at high speed). In order to access a remote memory assigned to another CPU package, it is necessary to access the remote memory via the CPU core belonging to the CPU package. Accordingly, a problem of reduction in processing speed is caused.

Therefore, in the system 210, each CPU core can access only the local memory, and the access to the remote memory under control of another CPU package is prohibited.

[0119]

The assignment and limitation are realized by input of the following command "NUMA control" when the OS is started.

[0120]

[1] Assignment on AP server

```
"#numactl
-physcpubind=1-15      ← Assign cores (01) to (15)
-membind=0            ← Assign first memory
nice -n -5            ← Designate process priority
java -jar aloha.jar"   ← Start AP server process (Java)
```

[0121]

[2] Assignment on first DB server

```
"#numactl
-physcpubind=17-23     ← Assign cores (17) to (23)
-membind=1             ← Assign second memory
nice -n -5            ← Designate process priority
su -postgres          ← Switch process execution user to
postgres (su)

-c"pg_ctl -D data1 start" ← Start first DB server process
(database for data area data1)"
```

[0122]

[3] Assignment on second DB server

```
"#numactl
-physcpubind=24-30     ← Assign cores (24) to (30)
-membind=1            ← Assign second memory
nice -n -5            ← Designate process priority
su -postgres          ← Switch process execution user to
```

```

postgres (su)
    -c"pg_ctl -D data2 start" ← Start second DB server
process
    (database for data area data2)"
[0123]
    [4] Assignment on third DB server
    "#numactl
    -physcpubind=31-38      ← Assign cores (31) to (38)
    -membind=2             ← Assign third memory
    nice -n -5             ← Designate process priority
    su -postgres          ← Switch process execution user to
postgres (su)
    -c"pg_ctl -D data3start" ← Start third DB server process
    (database for data area data3)"
[0124]
    [5] Assignment on fourth DB server
    "#numactl
    -physcpubind=39-45     ← Assign cores (39) to (45)
    -membind=2             ← Assign third memory
    nice -n -5             ← Designate process priority
    su -postgres          ← Switch process execution user to
postgres (su)
    -c"pg_ctl -D data4 start" ← Start fourth DB server
process
    (database for data area data4)"
[0125]
    [6] Assignment on fifth DB server

```

```

`#numactl
-physcpubind=46-53      ← Assign cores (46) to (53)
-membind=3             ← Assign fourth memory
nice -n -5             ← Designate process priority
su -postgres          ← Switch process execution user to
postgres (su)

-c"pg_ctl -D data5 start" ← Start fifth DB server process
(database for data area data5)"

```

[0126]

[7] Assignment on sixth DB server

```

`#numactl
-physcpubind=54-60      ← Assign cores (54) to (60)
-membind=3             ← Assign fourth memory
nice -n -5             ← Designate process priority
su -postgres          ← Switch process execution user to
postgres (su)

-c"pg_ctl -D data6 start" ← Start sixth DB server process
(database for data area data6)"

```

[0127]

Fig. 12 is a block diagram showing a functional configuration of the data management system 210. In an AP server 232, a screen processing unit 252, a processing content specifying unit 254, a DB processing assigning unit 256, and an operational processing unit 258 are formed.

A plurality of client terminals 262 is connected to the screen processing unit 252 via a communication network 260 such as the Internet.

[0128]

The screen processing unit 252, the processing content specifying unit 254, the DB processing assigning unit 256, and the operational processing unit 258 are realized by execution of predetermined processing with CPU cores (01) to (15) according to an application program for business processing. In this case, a plurality of threads is activated by an OS or middleware, and is assigned to the respective CPU cores, thereby realizing multi-task processing.

[0129]

Fig. 13 shows a corresponding relationship between each CPU core and a plurality of threads 264. A thread pool 266 is provided to each thread 264, and it is imaged that the thread 264 sequentially processes a plurality of tasks 268 arranged in the thread pool 266.

The thread 264 in the diagram functions as the screen processing unit 252, the processing content specifying unit 254, the DB processing assigning unit 256, and the operational processing unit 258 of the AP server 232. Specific processes executed by the respective functional components correspond to the tasks 268.

[0130]

Each thread 264 sequentially executes the tasks 268 stored in the thread pool 266 in chronological order. If the thread pool 266 of the thread 264 is empty, the thread 264 executes the tasks 268 stored in the thread pool 266 of another thread 264 in reverse chronological order.

[0131]

The DB processing assigning unit 256 issues an SQL to the first DB server 236 to sixth DB server 246, and performs a function of instructing extraction (selecting) or addition (inserting) of data necessary for the business processing.

[0132]

The DB processing assigning unit 256 and each DB server are connected via a number of connection pools 270 corresponding to the number of CPU cores of the each DB server.

For example, since seven CPU cores are assigned to the first DB server 236, seven connection pools 270 are provided and each connection pool 270 is preliminarily associated with a specific CPU core.

Since eight CPU cores are assigned to the fifth DB server 244, eight connection pools 270 are provided and each connection pool 270 is preliminarily associated with a specific CPU core.

[0133]

The external storage device 230 stores a plurality of sets of table groups each including, for example, a purchase table 272, an order table 273, an order specification table 274, an order cancel table 275, a product item table 276, a completed product table 277, a raw material table 278, a raw material index table 279, a cost table 280, and a supplier table 281 per each DB server.

Herein, since the first DB server 236 to sixth DB server 246 are provided, six sets of table groups A to F, each dedicated for each DB server, are stored in separate areas in the external

storage device 230, respectively.

As will be described later, the same records are simultaneously overlappingly stored in the table with the same name belonging to each table group.

[0134]

Fig. 14 shows, as an example, a structure of records stored in each table.

That is, the purchase table 272 stores a record having data items "purchase ID", "cost ID", and "quantity".

The order table 273 stores a record having data items "order ID" and "customer ID".

The order specification table 274 stores a record having data items "order specification ID", "order ID", "product item ID", and "quantity".

The order cancel table 275 stores a record having the data item "order specification ID".

[0135]

The product item table 276 stores a record having the data item "product item ID".

The completed product table 277 stores a record having the data item "product item ID".

The raw material table 278 stores a record having the data item "product item ID".

The raw material index table 279 stores a record having the data items "raw material index ID", "parent product item ID", "child product item ID", and "quantity".

The cost table 280 stores a record having the data items

"cost ID", "supplier product item ID", and "cost".

The supplier table 281 stores a record having the data items "supplier product item ID", "supplier ID", and "product item ID".

[0136]

Each of the tables 272 to 281 has the following limitations

(1) to (3).

(1) The number of key items is limited to one.

Therefore, it is prohibited to structure the key item (complex key) with a combination of a plurality of items.

[0137]

(2) When the AP server 232 accesses each table, only the reference and addition of the record are permitted, and the deletion and updating of the existing record are prohibited.

[0138]

Therefore, when it is necessary to make the existing data invalid, a state equal to the deletion is realized by adding a new record to another table in place of deleting the record. , For example, when the order is canceled, the state is generally managed by describing a value representing "cancel" to the order data to be canceled. However, in the system 210, a fact of "cancel" is expressed by adding the record "order specification ID" as a target to the order cancel table 275.

[0139]

When a value of the existing record is changed, the updating of the value is substantially realized by adding the record of a different primary key having a corrected value to

the same table. For example, when the quantity of a record stored in the order specification table 274 is changed from "3" to "5", a new record with quantity "5" is added to the order specification table 274. Since a time stamp is marked to each record with precision of milliseconds, the new record or the old record (updating fact) is clarified by comparing the time stamps.

[0140]

(3) A column of the table has limitations of NULL value prohibition, flag value prohibition, and segmented value prohibition.

First, the "NULL value prohibition limitation" means the prohibition of filling NULL (no value) as a value of the data item, and the setting of the data item assuming that the NULL value is filled is not permitted.

Next, the "flag value prohibition limitation" means the prohibition of filling a flag value (binary data such as "1/0", "ON/OFF", or "TRUE/FALSE") as a value of the data item, and the setting of the data item assuming that such a flag value is filled is not permitted.

The "segmented value prohibition limitation" means the prohibition of filling a segmented value ("1: full-time employee", "2: part-time employee", or "3: casual employee") as a value of the data item, and the setting of the data item assuming that such a segmented value is filled is not permitted.

[0141]

As mentioned above, the deletion and updating of the

record by the AP server 232 are prohibited. Therefore, the data at any past time is kept in each table. Advantageously, it is possible to handle also the case where a change history will need to be reproduced in the future.

However, when time sufficiently passes and it is completely unnecessary to keep the data, it is possible to delete the record by directly logging in to each DB server and issuing an SQL of deletion.

[0142]

In the case of the system 210, since individual records stored in each table are maximally normalized as described above, the operational processing between the data is performed by the operational processing unit 258 of the AP server 232, and redundant intermediate data is not generated with the operational processing on the DB server side, though it is handled by insertion of the record in place of the deletion or updating of the record, the disc capacity is not unnecessarily compressed as compared with the conventional manner. Further, the hit rate of a cache memory is improved, thereby contributing to high speed of processing by the DB server.

[0143]

The database management system of each DB server checks whether the limitation rules are adequate or not when designing a new table or issuing an SQL statement. The execution of processing that infringes the limitation rules is rejected, thereby ensuring its effectiveness.

[0144]

Under a situation in which the business data is managed by such a table group, a user who desires the registration of new data inputs data via a data input screen (not shown) sent to the client terminal 262 from the screen processing unit 252. [0145]

The processing content specifying unit 254 having received the input data from the screen processing unit 252 requests the DB processing assigning unit 256 to register the data.

The DB processing assigning unit 256 issues an SQL that requests the registration of the input data to the first DB server 236 to sixth DB server 246. In this case, the DB processing assigning unit 256 issues the SQL to each DB server via a connection pool associated with a specific CPU core, thereby designating a CPU core in charge in each DB server. [0146]

In each DB server having received the SQL, the CPU core that is designated by the DB processing assigning unit 256 executes the insertion of records to the corresponding table all at once.

Consequently, the same record is overlappingly stored to the table with the same name that is independently managed by each DB server.

[0147]

Next, a description is given of a processing procedure when sending a check request of the number of stocks of a completed product (e.g., product item ID "NRI-0001") from the

client terminal with flowcharts in Figs. 15 and 16.

[0148]

First, on a screen for designating a search condition sent from the screen processing unit 252, a user designates a stock search of the product item ID "NRI-0001" as a completed product. The processing content specifying unit 254 specifies processing required for realizing the search request (S210 in Fig. 15).

[0149]

Specifically, the following processing contents are specified and are thereafter sequentially executed.

(01) All records with the product item ID "NRI-0001" are extracted from the order specification table (S212).

(02) The accumulated order quantity of the product item ID "NRI-0001" is calculated (S214).

(03) All records with the product item ID "NRI-0001" are extracted from the order cancel table (S216).

(04) The accumulated order cancel quantity of the product item ID "NRI-0001" is calculated (S218).

(05) The accumulated order cancel quantity is subtracted from the accumulated order quantity of the product item ID "NRI-0001", thereby calculating the accumulated actual order quantity at the moment (S220).

(06) With reference to the raw material index table, a child product item ID with the product item ID "NRI-0001" as the parent product item ID and the quantity thereof are obtained (S222). The child product item ID indicates a raw material of the product item ID "NRI-0001" as a completed product.

(07) The supplier product item ID is obtained from the supplier table with each child product item ID as a key (S224 in Fig. 16).

(08) The cost ID is obtained from the cost table with the supplier product item ID as a key (S226).

(09) All records on the raw material are extracted from the purchase table with the cost ID as a key (S228).

(10) The accumulated purchase quantity is calculated for each raw material (S230).

(11) A producible quantity of the product item ID "NRI-0001" is calculated based on the accumulated purchase quantity of each raw material (S232).

(12) The current stock quantity of the completed product is calculated by subtracting the accumulated actual order quantity from the producible quantity (S234 in Fig. 15).

(13) Finally, a processing result screen with a description of the stock quantity of "NRI-0001" is sent to the client terminal 262 (S236).

[0150]

In the processing, the data extraction processing is executed by issuing the SQL to each DB server from the DB processing assigning unit 256. In this case, the DB processing assigning unit 256 sends the SQL to each DB server via the connection pool associated with a specific CPU core. As a result, the designated CPU core executes record extracting processing in the DB server.

The operational processing unit 258 performs necessary

operational processing such as sorting, merging, matching, and control break on the data sent from each DB server.

[0151]

Incidentally, the above-mentioned each processing includes processing in which succeeding processing is not executed when the previous processing is not completed and processing that can be concurrently performed.

For example, as shown in Fig. 15, it is possible to concurrently perform the processing in S212 for extracting the record of the product item ID from the order specification table 274, the processing in S216 for extracting the record of the product item ID from the order cancel table 275, and the processing in S222 for obtaining the child product item ID and the quantity with the product item ID as the parent product item ID by referring to the raw material index table 279, when the product item ID is specified.

Therefore, the DB processing assigning unit 256 designates separate CPU cores to the above-mentioned processing, respectively, and requests extraction processing to any DB server.

[0152]

At timing when a kind of raw material necessary for producing the completed product is specified in the processing in S222, as shown in Fig. 16, extraction processing of data necessary for calculating a purchase quantity of each raw material is started. However, S224 to S228 also are also concurrently processed for each raw material.

For example, if the number of kinds of raw materials is 250, the DB processing assigning unit 256 designates 50 mutually different CPU cores, and requests each DB server to extract a record.

As mentioned above, it is possible to efficiently refer to data by directly assigning the extraction of a specific record to a specific CPU core.

[0153]

Note that, if the number of kinds of raw materials is several hundred or more, processing per raw material cannot be assigned to different CPU cores at a time.

In this case, the DB processing assigning unit 256 requests each CPU core to perform the processing for each raw material in order, and repeats assignment of the processing for each remaining raw material in order to the CPU core when the processing ends.

[0154]

The application program of the AP server 232 recognizes the kind of data and an extracting procedure necessary for realizing specific business processing. Therefore, the extraction of a specific record is requested to each DB server with designation of the CPU core, thereby most efficiently pulling out the potential of the multi-core CPU.

[0155]

Note that, since the OS is provided with a function of daringly reducing the number of clocks of the CPU in the middle of the processing in order to realize the efficiency and power

saving of the data processing based on static information. However, such kind of unnecessary service is desirably disabled in advance, because it is impeditive when processing contents are finely controlled per CPU core like the present system 210.

[0156]

Fig. 17 is a whole configuration diagram of a service provision system 310 according to the invention. First, in a data center 312, an AP server 314, a first DB server 316, a second DB server 317, a third DB server 318, a fourth DB server 319, a fifth DB server 320, a sixth DB server 321, and a seventh DB server 322 are installed.

[0157]

Although the details will be described later, the same server computer 324 includes the AP server 314 and the first DB server 316 to sixth DB server 321 that are connected to each other via signal lines on a mother board.

On the other hand, the seventh DB server 322 is provided on a server computer in another casing, and is connected to the AP server 314 via LAN.

[0158]

The AP server 314 is connected to an eighth DB server 326, a ninth DB server 328, and a tenth DB server 330 that are installed at a remote area, via a communication line 332.

Further, the AP server 314 is connected also to a plurality of client terminals 336 via an Internet 334.

[0159]

A common table is provided to the first DB server 316 to

tenth DB server 330, respectively (the details will be described later).

[0160]

Fig. 18 shows an internal structure of the AP server 314 including a business processing unit 340, a DB control unit 342, a first DB connection unit 343, a second DB connection unit 344, a third DB connection unit 345, a fourth DB connection unit 346, a fifth DB connection unit 347, a sixth DB connection unit 348, a seventh DB connection unit 349, an eighth DB connection unit 350, a ninth DB connection unit 351, and a tenth DB connection unit 352.

[0161]

The business processing unit 340 executes the business processing such as registration or search of data in response to a request from the client terminal 336. A CPU of the server computer 324 operates according to a dedicated application program, thereby realizing the processing.

[0162]

The first DB connection unit 343 to tenth DB connection unit 352, to which the first DB server 316 to tenth DB server 330 are dedicatedly assigned in advance, respectively, perform a function of issuing an SQL statement to a corresponding DB server and instructing the registration and extraction of data.

Each DB connection unit is realized by operating a CPU of the server computer 324 according to a dedicated middleware.

[0163]

The DB control unit 342 performs a function of copying

data by the number of DB servers when having received an instruction of data registration from the business processing unit 340 and delivering the data to each DB connection unit and a function of extracting the data from a specific DB server via any DB connection unit and performing necessary operational processing when having received an instruction of data reference from the business processing unit 340.

The DB control unit 342 is realized by operating the CPU of the server computer 324 according to a dedicated application program.

[0164]

Fig. 19 shows a hardware configuration example inside the server computer 24, enclosing in a casing, a mother board 368 including a first CPU package 360, a second CPU package 362, a third CPU package 364, and a fourth CPU package 366.

[0165]

Fifteen CPU cores are formed in each CPU package, and thus the server computer 324 includes total 60 CPU cores.

Each CPU package is attached to a CPU socket (not shown) provided on the mother board 368.

[0166]

A dedicated memory (local memory) is assigned to each CPU package.

That is, a first memory 370, a second memory 371, a third memory 372, and a fourth memory 373 are dedicatedly provided to the first CPU package 360, the second CPU package 362, the third CPU package 364, and the fourth CPU package 366,

respectively.

[0167]

An auxiliary storage device 374 having an SSD (Solid State Drive) is connected to the server computer 324, and the auxiliary storage device 374 stores various tables (the details will be described later). In place of the SSD, a hard disk can constitute the auxiliary storage device 374.

[0168]

Peculiar functionality is preliminarily assigned to each CPU core.

That is, the AP server 314 is assigned dedicatedly to cores (01) to (15) formed in the first CPU package 360.

[0169]

A kernel thread (OS) 376 is dedicatedly assigned to a core (16) formed in the second CPU package 362.

The first DB server 316 is dedicatedly assigned to cores (17) to (23) formed in the second CPU package 362.

Similarly, the second DB server 317 is dedicatedly assigned to cores (24) to (30) formed in the second CPU package 362.

[0170]

The third DB server 318 is dedicatedly assigned to cores (31) to (38) formed in the third CPU package 364.

Similarly, the fourth DB server 319 is dedicatedly assigned to cores (39) to (45) formed in the third CPU package 364.

[0171]

The fifth DB server 320 is dedicatedly assigned to cores (46) to (53) formed in the fourth CPU package 366.

Similarly, the sixth DB server 321 is dedicatedly assigned to cores (54) to (60) formed in the fourth CPU package 366.

[0172]

As mentioned above, one program for AP server and six programs (instances) for DB server are activated in one server computer 324, and a plurality of CPU cores is dedicatedly assigned to each server.

[0173]

As mentioned above, a first memory 370 is dedicatedly assigned to the AP server 314, a second memory 371 is dedicatedly assigned to a kernel thread (OS) 376, the first DB server 316 and second DB server 317, a third memory 372 is dedicatedly assigned to the third DB server 318 and fourth DB server 319, and a fourth memory 373 is dedicatedly assigned to the fifth DB server 320 and sixth DB server 321, as local memories.

[0174]

In this case, each CPU core can directly (at high speed) access the local memory. In order to access a remote memory assigned to another CPU package, the access via the CPU core belonging to the corresponding CPU package is required. Accordingly, a problem of reduction in the processing speed is caused.

Therefore, in the system 310, each CPU core can access only the local memory, and the access to the remote memory under

control of another CPU package is prohibited.

[0175]

The assignment and limitations as described above are realized by inputting the following command "NUMA control" when the OS is activated.

[0176]

[1] Assignment on AP server 314

```
"#numactl
```

```
-physcpubind=1-15      ← Assign cores (01) to (15)
```

```
-membind=0             ← Assign first memory
```

```
nice -n -5             ← Designate process priority
```

```
java -jar aloha.jar"   ← Activate AP server process
```

(Java/registered trademark)

[0177]

[2] Assignment on first DB server 316

```
"#numactl
```

```
-physcpubind=17-23     ← Assign cores (17) to (23)
```

```
-membind=1             ← Assign second memory
```

```
nice -n -5             ← Designate process priority
```

```
su -postgres           ← Switch process execution user to
```

postgres (su)

```
-c"pg_ctl -D data1 start" ← Start first DB server process  
(database for data area data1)"
```

[0178]

[3] Assignment on second DB server 317

```
"#numactl
```

```
-physcpubind=24-30     ← Assign cores (24) to (30)
```

```

-membind=1                ← Assign second memory
nice -n -5                ← Designate process priority
su -postgres              ← Switch process execution user to
postgres (su)

-c"pg_ctl -D data2 start" ← Activate second DB server
process
(database for data area data2)"

```

[0179]

[4] Assignment on third DB server 318

```

"#numactl

-physcpubind=31-38        ← Assign cores (31) to (38)
-membind=2                ← Assign third memory
nice -n -5                ← Designate process priority
su -postgres              ← Switch process execution user to
postgres (su)

-c"pg_ctl -D data3 start" ← Activate third DB server
process
(database for data area data3)"

```

[0180]

[5] Assignment on fourth DB server 319

```

"#numactl

-physcpubind=39-45        ← Assign cores (39) to (45)
-membind=2                ← Assign third memory
nice -n -5                ← Designate process priority
su -postgres              ← Switch process execution user to
postgres (su)

-c"pg_ctl -D data4 start" ← Activate fourth DB server

```

process

(database for data area data4)"

[0181]

[6] Assignment on fifth DB server 320

"#numactl

-physcpubind=46-53      ← Assign cores (46) to (53)

-membind=3              ← Assign fourth memory

nice -n -5              ← Designate process priority

su -postgres          ← Switch process execution user to

postgres (su)

-c"pg\_ctl -D data5 start" ← Activate fifth DB server

process

(database for data area data5)"

[0182]

[7] Assignment on sixth DB server 321

"#numactl

-physcpubind=54-60      ← Assign cores (54) to (60)

-membind=3              ← Assign fourth memory

nice -n -5              ← Designate process priority

su -postgres          ← Switch process execution user to

postgres (su)

-c"pg\_ctl -D data6 start" ← Activate sixth DB server

process

(database for data area data6)"

[0183]

The business processing unit 340, the DB control unit 342, and the first DB connection unit 343 to tenth DB connection unit

352 in the AP server 314 are realized by execution of predetermined processing by the CPU cores (01) to (15) according to the dedicated program as mentioned above. In this case, a plurality of threads is activated by the OS or middleware and is assigned to each CPU core, thereby realizing the multi-task processing.

[0184]

Fig. 20 shows a corresponding relationship between each CPU core and a plurality of threads 378. A thread pool 380 is provided to each thread 378, and it is imaged that the thread 378 sequentially processes a plurality of tasks 382 arranged thereto.

The thread 378 in the diagram functions as the business processing unit 340, the DB control unit 342, and the first DB connection unit 343 to tenth DB connection unit 352 in the AP server 314. Specific processing executed by each of the functional components corresponds to the task 382.

[0185]

Each thread 378 sequentially executes the tasks 382 stored in the thread pool 380 in chronological order. If the thread pool 380 of the thread 378 is empty, executes the tasks 382 stored in the thread pool 380 in another thread 378 in reverse chronological order.

[0186]

As shown in Fig. 21, the first DB connection unit 343 to sixth DB connection unit 348 are connected to the first DB server 316 to sixth DB server 321 via a number of connection pools 384

corresponding to the number of CPU cores assigned to each DB server.

For example, since seven CPU cores are assigned to the first DB server 316, seven connection pools 384 are provided and each connection pool 384 is preliminarily associated with a specific CPU core of the first DB server 316.

Since eight CPU cores are assigned to the fifth DB server 320, eight connection pools 384 are provided and each connection pool 384 is preliminarily associated with a specific CPU core of the fifth DB server 320.

[0187]

The auxiliary storage device 374 stores, per each DB server, a plurality of sets of table groups, each including, for example, a purchase table 386, an order table 387, an order specification table 388, an order cancel table 389, a product item table 390, a completed product table 391, a raw material table 392, a raw material index table 393, a cost table 394, a supplier table 395, and a customer table 396.

[0188]

Herein, the first DB server 316 to sixth DB server 321 are provided, thereby storing tables of 6 sets of table groups A to F in separate areas dedicated for each DB server in the auxiliary storage device 374.

As will be mentioned later, the same records are simultaneously overlappingly stored in tables with the same name belonging to each table group.

[0189]

Although not shown, the auxiliary storage device of the seventh DB server 322 to tenth DB server 330 also has a plurality of sets of table groups each including the purchase table 386, order table 387, order specification table 388, order cancel table 389, product item table 390, completed product table 391, raw material table 392, raw material index table 393, cost table 394, supplier table 395, customer table 396, and the like, and stores data with the same contents in the first DB server 316 to sixth DB server 321.

[0190]

Fig. 22 shows a structure of records stored in each table.

That is, the purchase table 386 stores records having the data items "purchase ID", "cost ID", and "quantity".

Further, the order table 387 stores records having the data items "order ID", "customer ID", and "order date and time".

The order specification table 388 stores records having the data items "order specification ID", "order ID", "product item ID", and "quantity".

In addition, the order cancel table 389 stores records having the data items "order specification ID" and "cancel date and time".

[0191]

The product item table 390 stores a record having the data item "product item ID".

The completed product table 391 stores a record having the data item "product item ID".

The raw material table 392 stores a record having the data

item "product item ID".

The raw material index table 393 stores records having the data items "raw material index ID", "parent product item ID", "child product item ID", and "quantity".

The cost table 394 stores records having the data items "cost ID", "supplier product item ID", and "cost".

The supplier table 395 stores records having the data items "supplier product item ID", "supplier ID", and "product item ID".

The customer table 396 stores a record having the data item "customer ID".

[0192]

Each of the tables 386 to 396 preliminarily has the following limitations (1) to (4).

(1) The number of key items is limited to one.

Therefore, it is prohibited to form a key item (complex key) with a combination of a plurality of items.

[0193]

(2) When the AP server 314 accesses each table, only reference and addition of the record are permitted and the deletion and updating of the existing record are prohibited.

[0194]

Therefore, when the existing data needs to be made invalid, a new record is added to another table in place of deleting the record, thereby realizing a state similar to the deletion. For example, in the case of canceling the order, in general, the state is managed by describing a value indicating "cancel" to

the order data to be canceled. In the system 310, a fact "cancel" is expressed by adding the record "order specification ID" as a target to the order cancel table 389.

[0195]

When a value of the existing record is changed, the value is substantially updated by adding the record of a different primary key having a corrected value to the same table. For example, when the cost of the record stored in the cost table 394 is changed from "300 yen" to "500 yen", a new record with the cost "500 yen" is added to the cost table 394. Since a time stamp is marked to each record with precision of milliseconds, with comparison between the time stamps, it (fact of updating) is clarified which record is new or old.

[0196]

As mentioned above, the deletion and updating of the record by the AP server 314 are prohibited. Therefore, data at any past timing is ensured to each table. It is advantageous to handle a case in which the reproduction of a change history will be required in the future.

However, when time sufficiently passes and it is completely not necessary to hold the data, it is possible to delete the record by directly logging in to each DB server and issuing an SQL for deletion.

[0197]

(3) A column of the table has limitations of NULL value prohibition, flag value prohibition, and segmented value prohibition.

First, the "NULL value prohibition limitation" means the prohibition of filling NULL (no value) as a value of the data item, and the setting of the data item assuming that the NULL value is filled is not permitted.

Next, the "flag value prohibition limitation" means the prohibition of filling a flag value (binary data such as "1/0", "ON/OFF", or "TRUE/FALSE") as a value of the data item, and the setting of the data item assuming that such a flag value is filled is not permitted.

The "segmented value prohibition limitation" means the prohibition of filling a segmented value ("1: full-time employee", "2: part-time employee", or "3: casual employee") as a value of the data item, and the setting of the data item assuming that such a segmented value is filled is not permitted.  
[0198]

(4) In expansion of functionality, it is handled by newly providing a table in place of adding a column to the existing table,.

That is, when the order stored in the "order specification table 388" as mentioned above is canceled, it is handled by registering the order specification ID of the order desired to be canceled to the "order cancel table 389" at the moment. Therefore, as shown in Fig. 23(a), even in a case of reducing only the quantity while keeping the order, all the quantities are canceled once. Thus, the user needs to perform the order of reduced quantity again.

[0199]

Herein, when expanding functionality to reduce only the quantity while keeping the order, as shown in Fig. 23(b), it can be considered to add a column "cancel quantity" to the order cancel table 389. As a result, certainly, an object to reduce only the quantity is accomplished while keeping the order. However, it is not easy to change all at once the tables being used currently in a large number of DB servers. When it is necessary to return the functionality to the ordinal one due to the reason that an application program supporting new functionality has a trouble, a problem that it takes time to do so is caused.

[0200]

Therefore, with the system 310, as shown in Fig. 24, while the conventional order cancel table 389 remains, such a method is used to newly provide a partial cancel table 389' having the data items "order specification ID", "cancel quantity", and "cancel date and time" to each DB server.

That is, the order cancel table 389 is separated from the system 310 at switching timing to a new service for accepting partial cancel per quantity. The order cancel is recorded by adding a record having the order specification ID and cancel quantity as targets to the partial cancel table 389' (in case of canceling the order, all the quantities of order is set as "cancel quantity".)

[0201]

As mentioned above, as compared with correcting all together the tables that are working currently in a large number

of DB servers, it is easier to realize the addition of a new table to each DB server. A table (order cancel table 389) before the expansion of functionality is ensured without change. Thus, it is advantageous to fast handle even a case where the service needs to be returned to the original one.

[0202]

The database management system of each DB server checks whether the limitation rules (1) to (4) are adequate or not at design time of a new table or issuing time of an SQL statement. The effectiveness is ensured by rejecting the execution of processing that infringes the limitation rules.

[0203]

In the case of the system 310, since individual records stored in each table are maximally normalized as described above, the operational processing between the data is performed by the DB control unit 342 of the AP server 314, and redundant intermediate data is not generated with the operational processing on the DB server side, though it is handled by insertion of the new record in place of the deletion or updating of the record, the disc capacity is not unnecessarily compressed as compared with the conventional manner. Further, the hit rate of a cache memory is improved, thereby contributing to high speed of processing by the DB server.

[0204]

Under the situation in which the business data is managed with such a table group, a user who desires the registration of new data inputs data via a data input screen (not shown) sent

to the client terminal 336 from the business processing unit 340.

[0205]

The DB control unit 342 having received the input data from the business processing unit 340 requests the first DB connection unit 343 to tenth DB connection unit 352 to register the data.

Each DB connection unit issues an SQL statement for requesting the registration of the input data to the DB server of which the DB connection unit is in charge.

[0206]

In each DB server having received the SQL statement, records are inserted all at once to corresponding tables.

As a result, the same record is overlappingly stored to the table with the same name that is independently managed by each DB server.

[0207]

Note that the first DB connection unit 343 to sixth DB connection unit 348 issue an SQL via a connection pool 384 associated with a specific CPU core to each corresponding DB server (first DB server 316 to sixth DB server 321). Thus, a CPU core in charge is designated in each DB server.

In each DB server having received the SQL, the designated CPU core executes all at once the insertion of the record to a corresponding table.

[0208]

Each DB server sends a notification of reception

completion to the DB connection unit when the DB server receives the data from the DB connection unit and stores the data to a memory thereof. Thereafter, the DB server stores a corresponding record to the table provided in the auxiliary storage device such as a hard disk or SSD.

[0209]

The notification of reception completion sent from the DB server is collected to the DB control unit 342 via each DB connection unit.

The DB control unit 342 outputs a notification of registration completion to the business processing unit 340 when the notification of reception completion is returned from all the DB connection units.

[0210]

When having received the notification of registration completion from the DB control unit 342, the business processing unit 340 recognizes the data as a reading target. Conversely, the business processing unit 340 sets the data as an out-of-target for reading until the notification of registration completion is returned, thereby preventing a danger of reading illegal data due to shift of registration timing in each DB server.

[0211]

In the case of the system 310, only the insertion of each business data is permitted and a rule that does not accept the deletion and updating is applied, as mentioned above. Therefore, it is extremely easy to restore the data.

That is, when it is premised that the deletion and updating of the data are permitted, it is necessary to reproduce the addition, deletion, and updating of the data in order from certain timing based on updating history information held by the DB server. This takes a long time and large load. Further, it is necessary to provide a structure for certainly storing an updating log on the DB server side.

[0212]

On the other hand, as the system 310, when it is premised that only the insertion of data is permitted and the deletion and updating are prohibited, the data registered to another normal DB server is compared with data registered to the DB server in which a default is generated. Lacking data may be simply copied and the order of data (relationship of before/after) may not be considered.

[0213]

For example, when the server computer 324 is troubled, data in the seventh DB server 322 installed in the same campus is taken into a new server computer, thereby extremely easily restoring data.

Alternatively, when a large disaster such as an earthquake occurs and the data center 312 is seriously damaged, the data can be restored relatively earlier based on the data in the eighth DB server 326 to tenth DB server 330 installed at a remote area.

[0214]

With the system 310, when data sent from the DB connection

unit as mentioned above is stored to a memory of the DB server, a notification of reception completion is issued from the DB server to the AP server 314. Therefore, the business processing unit 340 can set the data as a reading target for an extremely short time.

[0215]

With the conventional common sense, the memory is a volatile storage unit and loses data when power supply stops. Therefore, the notification of completion is to be returned when data is stored to a non-volatile storage unit (hard disk, etc.).

On the other hand, in the case of the system 310, even if the data is damaged in one DB server, it is possible to catch up with the operation simply by copying shortage data from another DB server. Therefore, it is not necessary to wait until the data is stored in a hard disk or the like.

[0216]

Next, a description is given of a processing procedure when a reference request for the number of stocks of a completed product (e.g., product item ID "NRI-0001") is sent from a client terminal with reference to flowcharts in Figs. 25 and 26.

[0217]

First, on a search condition designating screen (not shown) sent from the business processing unit 340, a user instructs a stock search of the product item ID "NRI-0001" as a completed product, and the business processing unit 340 specifies processing required for realizing the search request (S310 in Fig. 25).

[0218]

Specifically, the following processing contents are specified and are thereafter sequentially executed.

(01) All records having the product item ID "NRI-0001" are extracted from the order specification table 388 (S312).

(02) The accumulated order quantity of the product item ID "NRI-0001" is calculated (S314).

(03) All records having the product item ID "NRI-0001" are extracted from the order cancel table 389 (S316).

(04) The accumulated order cancel quantity of the product item ID "NRI-0001" is calculated (S318).

(05) The accumulated order cancel quantity is subtracted from the accumulated order quantity of the product item ID "NRI-0001" and the accumulated actual order quantity is calculated at the moment (S320).

(06) The child product item ID having the product item ID "NRI-0001" as the parent product item ID and the quantity thereof are obtained by referring to the raw material index table 393 (S322). The child product item ID indicates the raw material of the product item ID "NRI-0001" of the completed product.

(07) The supplier product item ID is obtained from the supplier table 395 with each child product item ID as a key (S324 in Fig. 26).

(08) The cost ID is obtained from the cost table 394 with the supplier product item ID as a key (S326).

(09) All records related to the raw material are extracted

from the purchase table 386 with the cost ID as a key (S328).

(10) The accumulated purchase quantity is calculated for each raw material (S330).

(11) A producible quantity of the product item ID "NRI-0001" is calculated based on the accumulated purchase quantity of each raw material (S332).

(12) The current stock quantity of the completed product is calculated by subtracting the accumulated actual order quantity from the producible quantity (S334 in Fig. 25).

(13) Finally, a processing result screen (not shown) having a description of the stock quantity of "NRI-0001" is sent to the client terminal 336 (S336).

[0219]

In the processing, any of the first DB connection unit 343 to sixth DB connection unit 348 issues an SQL statement to any of the first DB server 316 to sixth DB server 321, thereby executing the data extraction processing.

In this case, each DB connection unit sends the SQL statement to each DB server via the connection pool 384 associated with a specific CPU core. As a result, the extraction processing of the record in the DB server is executed by the designated CPU core.

The DB control unit 342 performs necessary operational processing such as sorting, merging, matching, and control break on the data sent from each DB server.

[0220]

Incidentally, the processing includes processing with

which subsequent processing cannot be executed when the previous processing is not completed and processing with which the processing can be performed concurrently.

For example, as shown in Fig. 25, when the product item ID is specified, it is possible to concurrently perform, the processing in S312 for extracting the record of the corresponding product item ID from the order specification table 388, the processing in S316 for extracting the record of the corresponding product item ID from the order cancel table 389, and the processing in S322 for obtaining the child product item ID having the corresponding product item ID as the parent product item ID and the quantity thereof by referring to the raw material index table 393.

Therefore, the DB control unit 342 designates the separate CPU cores to the processing, respectively, and requests each DB server to perform the extraction processing via the corresponding DB connection unit.

[0221]

When a necessary kind of raw materials for producing the completed product is specified with the processing in S322, as shown in Fig. 26, extraction processing of data necessary for calculating the purchase quantity of each raw material is started. However, S324 to S328 is also concurrently performed for each raw material.

For example, if the number of kinds of raw materials is 50, the DB control unit 342 designates 50 mutually different CPU cores, and requests each DB server to extract the record

via the corresponding DB connection unit.

As mentioned above, it is possible to efficiently refer to the data by directly assigning the extraction of a specific record to a specific CPU core.

[0222]

Note that, when the number of kinds of raw materials is several hundred or more, processing per raw material cannot be assigned to different CPU cores at a time.

In this case, the DB control unit 342 repeats to request each CPU core to perform the processing per raw material in order, and assign the processing per remaining raw material to the CPU core when the processing ends.

[0223]

The application program of the AP server 314 recognizes the kind and extracting procedure of data necessary for realizing specific business processing. Therefore, it is possible to most effectively pull out the potential of the multi-core CPU by requesting each DB server to extract a specific record with designation of the CPU core.

[0224]

Note that, in order to realize the efficiency and power saving of data processing based on statistic information, the OS is provided with a function of daringly reducing the number of clocks of the CPU in the middle of the processing. However, such kind of unnecessary service is desirably disabled in advance, because it is impeditive when processing contents are finely controlled per CPU core like the present system 310.

[0225]

As mentioned above, in the case of the system 310, the same data is overlappingly stored in each DB server. Therefore, the DB control unit 342 can freely obtain necessary data from the DB server that can process the data at the highest speed, and can dispersively obtain the data from a plurality of DB servers and connect the data.

[0226]

Since an SQL statement necessary for reference and registration of data is issued by each DB connection unit, the SQL statement does not need to be written in the application program for realizing the business processing unit 340 and the DB control unit 342.

Moreover, the table stored in each DB server is normalized maximally and each DB connection unit is in charge of only the registration and extraction of the data designated by the DB control unit 342. Therefore, the SQL statement written in the program for DB connection unit may be extremely simple, and does not need to be changed point by point when the service functionality is expanded.

Obviously, the application program for realizing the business processing unit 340 and the DB control unit 342 needs to be corrected when the functionality of services is expanded. Since the description of the SQL statement is not included, the prospect is preferable and a bug can be easily found.

## **Brief Description of Drawings**

[0227]

[Fig. 1] Fig. 1 is a diagram of the whole configuration of a data management system according to the invention.

[Fig. 2] Fig. 2 is a block diagram showing an internal configuration of a first AP server and a second AP server.

[Fig. 3] Fig. 3 is a diagram showing a structure of reservation data.

[Fig. 4] Fig. 4 is a flowchart showing an issuing procedure of an artificial key.

[Fig. 5] Fig. 5 is a diagram showing a structure of an artificial key management table.

[Fig. 6] Fig. 6 is a diagram showing a structure of reservation cancel data.

[Fig. 7] Fig. 7 is a diagram showing a state of substantially updating a record.

[Fig. 8] Fig. 8 is a diagram showing a state of sending a series of related data to each DB server.

[Fig. 9] Fig. 9 is a diagram showing a state of sending in parallel a series of related data from an AP server to a DB server via a plurality of connections.

[Fig. 10] Fig. 10 is a diagram showing a corresponding relationship between a CPU core and a thread.

[Fig. 11] Fig. 11 is a schematic diagram showing a hardware configuration for realizing another data management system according to the invention.

[Fig. 12] Fig. 12 is a block diagram showing a functionality configuration of the data management system

according to the invention.

[Fig. 13] Fig. 13 is a schematic diagram showing a corresponding relationship between each CPU core of the AP server and a plurality of threads.

[Fig. 14] Fig. 14 is a diagram showing a structure example of records stored in each table.

[Fig. 15] Fig. 15 is a flowchart showing a processing procedure in the system.

[Fig. 16] Fig. 16 is a flowchart showing a processing procedure in the system.

[Fig. 17] Fig. 17 is a diagram of the whole configuration of a service provision system according to the invention.

[Fig. 18] Fig. 18 is a block diagram showing a functionality configuration of the AP server.

[Fig. 19] Fig. 19 is a schematic diagram showing a hardware configuration of the AP server.

[Fig. 20] Fig. 20 is a schematic diagram showing a corresponding relationship between each CPU core in the AP server and a plurality of threads.

[Fig. 21] Fig. 21 is a schematic diagram showing a corresponding relationship between a part of DB connection units in the AP server and each DB server provided in the same casing as that of the AP server.

[Fig. 22] Fig. 22 is a diagram showing a structure example of the records stored in each table.

[Fig. 23] Fig. 23 is a diagram showing a change example of a table structure at time of functionality expansion of

services.

[Fig. 24] Fig. 24 is a diagram showing a change example of a table structure at time of the functionality expansion of services.

[Fig. 25] Fig. 25 is a flowchart showing an example of a processing procedure when data is extracted in the system.

[Fig. 26] Fig. 26 is a flowchart showing an example of the processing procedure when data is extracted in the system.

#### **Explanation of references**

[0228]

110: data management system

112: first data center

114: first AP server

116: first DB server

118: second DB server

120: second data center

122: second AP server

124: third DB server

126: fourth DB server

127: communication network

128: client terminal

129: communication line

130: business processing unit

132: artificial key management unit

134: data control unit

138: DB connection unit

140: reservation data  
141: reservation integrating data  
142: reservation shop data  
143: reservation customer data  
144: reservation money-amount data  
145: reservation data on year, month, and day  
150: reservation integrating table  
151: reservation shop table  
152: reservation customer table  
153: reservation money-amount table  
154: reservation table on year, month, and day  
156: artificial key management table  
157: last-digit management table  
158: server management table  
159: data center management table  
160: cancel integrating data  
161: cancel target data  
162: cancel data on year, month, and day  
163: cancel integrating table  
164: cancel target table  
165: cancel table on year, month, and day  
166: reservation cancel data  
170 to 174: connection  
180: CPU core  
181: thread  
182: thread pool  
183: task

210: data management system  
211: server computer  
212: first package  
214: second package  
216: third package  
218: fourth package  
220: mother board  
222: first memory  
224: second memory  
226: third memory  
228: fourth memory  
230: external storage device  
232: AP server  
236: first DB server  
238: second DB server  
240: third DB server  
242: fourth DB server  
244: fifth DB server  
246: sixth DB server  
252: screen processing unit  
254: processing content specifying unit  
256: DB processing assigning unit  
258: operational processing unit  
260: communication network  
262: client terminal  
264: thread  
266: thread pool

268: task  
270: connection pool  
272: purchase table  
273: order table  
274: order specification table  
275: order cancel table  
276: product item table  
277: completed product table  
278: raw material table  
279: raw material index table  
280: cost table  
281: supplier table  
310: service provision system  
312: data center  
314: AP server  
316: first DB server  
317: second DB server  
318: third DB server  
319: fourth DB server  
320: fifth DB server  
321: sixth DB server  
322: seventh DB server  
324: server computer  
326: eighth DB server  
328: ninth DB server  
330: tenth DB server  
332: communication line

334: Internet  
336: client terminal  
340: business processing unit  
342: DB control unit  
343: first DB connection unit  
344: second DB connection unit  
345: third DB connection unit  
346: fourth DB connection unit  
347: fifth DB connection unit  
348: sixth DB connection unit  
349: seventh DB connection unit  
350: eighth DB connection unit  
351: ninth DB connection unit  
352: tenth DB connection unit  
360: first CPU package  
362: second CPU package  
364: third CPU package  
366: fourth CPU package  
368: mother board  
370: first memory  
371: second memory  
372: third memory  
373: fourth memory  
374: auxiliary storage device  
378: thread  
380: thread pool  
382: task

384: connection pool  
386: purchase table  
387: order table  
388: order specification table  
389: order cancel table  
390: product item table  
391: completed product table  
392: raw material table  
393: raw material index table  
394: cost table  
395: supplier table  
396: customer table

**CLAIMS**

[Claim 1] A data management system having an AP server and a plurality of DB servers installed in each of a plurality of data centers provided apart from each other at a predetermined distance, wherein

all the DB servers respectively include tables that store mutual common data and each table also has a limitation for prohibiting deletion and updating of a record, and wherein

each of the AP servers is connected to the respective DB servers installed in the same data center, and is also connected to the respective DB servers installed in another data center,

each of the AP servers includes at least one business processing unit, an artificial key management unit, a data control unit, and a plurality of DB connection units,

the business processing unit requests the artificial key management unit to issue an artificial key when generating business data which is a result of business processing, and generates business data with ID as a primary key having a unique artificial key issued by the artificial key management unit,

the artificial key management unit refers to an artificial key management table arranged in the same data center and delivers a value of a latest artificial key at the moment to the business processing unit, and executes processing for updating a value of the table at the same time,

the data control unit copies business data output from the business processing unit and outputs the data to each of

the DB connection units, and also outputs a notification of registration completion to the business processing unit after checking that notifications of reception completion from all the DB servers are collected when the notifications of reception completion from the respective DB servers via the respective DB connection units are sent,

each of the DB connection units has set therein specific information of a DB server of which the each DB connection unit is in charge,

each DB connection unit issues an SQL to a DB server of which the each DB connection unit is in charge to request addition of a record when the data control unit outputs business data, and also, when a notification of reception completion is sent from the DB server, executes processing for outputting the notification to the data control unit,

further, an identification code of an AP server is set to a predetermined position of each of the artificial keys, and

the business processing unit refrains from referring to the business data until the notification of registration completion is delivered from the data control unit.

[Claim 2] The data management system according to claim 1, wherein

the business data includes a combination of a plurality of pieces of substantial data having a mutual relation generated as a result of business processing by the business processing unit and integrating data generated for bundling each data,

the integrating data has only a data item for storing an integrating ID having an artificial key as a primary key, and each piece of the substantial data includes a data item for storing the substantial data, a data item for storing a unique ID having an artificial key as a primary key, and a data item for storing an ID of the integrating data as an foreign key.

[Claim 3] The data management system according to claim 1 or 2, wherein

the respective AP servers and the respective DB servers are connected with each other via a plurality of connections,

the DB connection unit is provided in each AP server per connection, and

the business data is sent to each DB server in parallel via each DB connection unit and connection.

[Claim 4] The data management system according to any of claims 1 to 3, wherein

the business processing unit

newly generates business data including a data item for storing an ID of the business data, a data item for storing a unique ID having an artificial key as a primary key, and a data item for storing substantial data after correction, when existing business data needs to be updated, and

substantially updates existing business data by additionally registering the business data to an existing table

provided in each DB server via the data control unit and each DB connection unit.

[Claim 5] The data management system according to any of claims 1 to 4, wherein

the business processing unit

generates business data for data cancel including a data item for storing an ID of the business data and a data item for storing a unique ID having an artificial key as a primary key, when existing business data needs to be deleted, and

substantially deletes the existing business data by additionally registering the business data for data cancel to a table dedicated for cancel provided in each DB server via the data control unit and each DB connection unit.

[Claim 6] A data management system comprising:

a computer including a CPU package incorporating a plurality of CPU cores and an external storage device that stores a plurality tables;

an AP server configured by running a program for AP server on the computer; and

a plurality of DB servers configured by running a plurality of programs for DB server on the computer,

wherein the external storage device separately stores a plurality of tables that stores mutually common data per DB server, and

a plurality of different CPU cores is dedicatedly

assigned to the AP server and each DB server, respectively,  
wherein the AP server includes at least  
a function of specifying a kind of data necessary for  
executing business processing,  
a function of requesting extraction of necessary data to  
any of the DB servers,  
a function of performing necessary operational  
processing to data obtained from the DB server, and  
a function of simultaneously requesting all the DB  
servers to register data generated by the business processing,  
and  
wherein the AP server issues an SQL in a state of  
specifically designating a CPU core that is in charge of the  
processing when requesting the DB server to extract or register  
data, and  
the designated CPU core executes corresponding  
processing in the DB server having received the SQL.

[Claim 7] The data management system according to claim 6,  
wherein

the computer includes a plurality of the CPU packages,  
a dedicated local memory is assigned to each CPU package,  
a CPU core belonging to a separate CPU package is assigned  
to each of the AP server and at least a part of the DB servers,  
and

the AP server and each DB server have such a limitation  
that it is possible to access only the local memory and it is

not possible to access the local memory assigned dedicatedly to another CPU package.

[Claim 8] The database management system according to claim 6 or 7, wherein

the AP server has such a limitation that it is allowed only to request each of the DB servers to insert or extract data and it is prohibited to request each of the DB servers to update or delete data.

[Claim 9] A service provision system comprising an AP server and a plurality of DB servers connected to the AP server, and providing a predetermined information processing service to a client terminal, wherein

the AP server includes a plurality of DB connection units provided per each of the DB servers, a business processing unit, and a DB control unit,

each of the DB connection units has set therein specific information of a DB server of which the each DB connection unit is in charge,

a table for storing mutually common data is provided in each of the DB servers,

each table has such a limitation that prohibits deletion or updating of a record,

the DB control unit, when having received a request for registering data from the business processing unit, copies and delivers the data to all the DB connection units,

each DB connection unit having received the data issues an SQL statement to a DB server of which the each DB connection unit is in charge to request addition of a record,

the DB control unit specifies one or a plurality of DB servers from among the DB servers and makes a request for extracting data to a DB connection unit that is in charge of the DB server, when a reference to specific data is requested from the business processing unit,

the DB connection unit having received the request issues an SQL statement to a DB server of which the DB connection unit is in charge to request extraction of a record, and also delivers the data received from the DB server to the DB control unit, and

the DB control unit performs necessary operational processing on the received data and then delivers the resultant data to the business processing unit.

[Claim 10] The service provision system according to claim 9, wherein

the business processing unit

newly generates data having a value after updating when existing data needs to be updated, and

additionally registers the data to an existing table provided in each DB server via the DB control unit and each DB connection unit, thereby realizing substantive updating of the existing data.

[Claim 11] The service provision system according to claim 9 or 10, wherein

the business processing unit

generates data for data cancel including a data item for storing an ID of the business data when existing business data needs to be deleted, and

additionally registers the data for data cancel to a table dedicated for cancel provided in each DB server via the DB control unit and each DB connection unit, thereby realizing substantive deletion of the existing data.

[Claim 12] The service provision system according to any of claims 9 to 11, wherein

the AP server and at least two or more DB servers of the DB servers are formed on a same computer,

the computer includes a CPU package incorporating a plurality of CPU cores and a storage device that stores a plurality of tables,

the AP server is configured by running a program for AP server on the computer, and the business processing unit, the DB control unit, and a plurality of DB connection units are configured by running a dedicated application program on the AP server,

each of the DB servers is configured by running a plurality of programs for DB server on the computer,

a plurality of different CPU cores is dedicatedly assigned to the AP server and each DB server, respectively,

an SQL statement is issued in a form of specifically designating a CPU core that is in charge of the processing from the DB connection unit when extraction and registration of data is requested to the DB server, and

the designated CPU core executes corresponding processing in the DB server having received the SQL statement.

[Claim 13] A method for expanding functionality of a service provision system, comprising:

when expanding functionality of a service provided to a client terminal in a service provision system according to claim 9,

adding a table corresponding to functionality after expansion to each of the DB servers and setting a table related to functionality before expansion to an out-of-target of processing while leaving the table in each DB server, and

the business processing unit, the DB control unit, and each DB connection unit of the AP server are replaced with an expansion version corresponding to the added table.