InfoFrame Table Access Method for Real-Time Processing of Big Data

OOSAWA Hideki, MIYATA Tsuyoshi

Abstract

The era of big data is characterized by an increasing need to create new value and new business through the real-time processing of large amounts of data. This processing requires an increase in individual data processing speeds as well as high throughput. This paper introduces the InfoFrame Table Access Method, a memory DB product suitable for real-time big data processing thanks to its high-speed parallel data processing capability.

Keywords

memory DB, real-time processing, high-speed parallel data processing, big data

1. Introduction

The recent increase in sensor devices and the computerization of all kinds of information have brought about an explosive increase in the amount of data. Consequently, the need to process large amounts of data in real time and apply the results to business is also increasing.

This paper introduces the InfoFrame Table Access Method (TAM), a memory DB product capable of creating unprecedented new value in this era of big data.

2. Product Concept

The product concept of the memory DB product TAM is "to process big data at high speed and in real time."

The key to the implementation of this concept is to not only increase individual data processing performance but also prevent a drop in throughput performance even with parallel processing of big data.

The TAM is commercialized with the keyphrase "high-speed parallel data processing," which means achieving both high speed and high throughput.

3. Product Outline

3.1 Features and Product Configuration

The TAM implements high system availability and real-time processing under high-traffic conditions with the following features:

(1) High-speed search/update processing

High-speed data access is achieved by holding the information required for data access in memory.

(2) Excellent concurrent executability

In general, the main factor hindering the concurrent execution of parallel processing threads is resource access conflicts.

Resource access is controlled to improve throughput performance by increasing the multiplicity of parallel processing (the number of threads).

(3) Recovery combining fast recovery and multi-fault countermeasures

The risk of data loss in the case of multiple faults is dissolved by the replication function, which enables fast recovery from faults, and the journal function, which holds the information required for recovery in the storage device.

With the TAM, the tabulated data stored in memory is called the "memory tables." The TAM itself is composed of the core data access function called the "table access mechanism," the "utilities" and the "operation support tools," as shown in **Fig. 1** . Among these components, the table access mechanism includes the replication and journal functions described above as well as the API (Application Programming Interface) used for memory table access. The utilities consist of a group of tools used in system operations, including memory table maintenance such as generation and saving as well as operations related to the replication and journal functions. Finally, the operation support tools include a function that collects operating statistics information to identify trends in data quantity increases, etc. and to make future hardware enhancement decisions

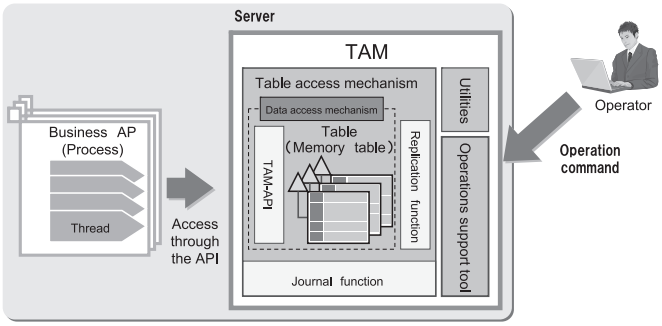

Fig. 1 Product configuration of the InfoFrame Table Access Method.

(extension of CPUs, memory, etc.) and a trace function that is used for performance analysis in the initial period after introduction.

In the following section, we will describe the outline of the table access mechanism, which is one of the biggest features and core functions of the TAM.

3.2 Data Access Functions

The following two points are important for the real-time processing of big data.

First, the time taken to process individual units of data should be extremely short (high-speed processing). Second, even when the amount of data processed by a server increases rapidly, the increase should be dealt with by an easy method, such as CPU extension, without large modification of the system architecture (maintenance of throughput performance with respect to data quantity increase).

The most generally used technique to improve the throughput performance of big data processing is to increase the multiplicity of processing (increasing the degree of parallelism). However, simply increasing multiplicity causes the limit of parallel processing to be reached. This occurs when the processing method used is of a kind that produces a large number of resource conflicts. With such a processing method, increasing multiplicity cannot improve throughput performance because of the overhead for the arbitration of resource conflicts (the processing cost of the exclusion control itself and the cost of context switching accompanying exclusive acquisition queuing and exclusive acquisition).

Therefore, to process big data at high speed and make it possible to deal with increases in data volume by increasing parallel processing, it is not sufficient to increase the processing speed required for the data access itself. It is also important to ensure that parallel data processing operations are not obstructed by resource conflicts.

In addition to providing a high-speed data access function, the TAM adopts a data structure and access processing method that can reduce the production of overhead due to resource conflicts.

(1) Increased data access speed

The TAM employs a simple data access method, which uses the specified part of the record as the key and accesses the records searched using this key (**Fig. 2**).

This method can be regarded as a data management method of the KVS (key-value store) type, with which keys and values are stored with correspondence relations assigned for them.

The information required for access (index part, data part) is placed in memory in the form of a memory table. The data in memory is accessed directly through the API to increase data access speed.

(2) Control of exclusion control overhead

The TAM adopts a processing model that ensures that one of its features, excellent concurrent executability, can be manifested fully. With this processing model, every memory table is updated by only a single thread (**Fig. 3**). The TAM optimizes the API, lock section and lock granularity according to this model to eliminate access conflicts to the updating target resource and reduce exclusion control overhead.

While the number of threads updating a memory table is never more than one, the number of reference threads is not limited to one. Memory table data that is mainly used for referencing is provided with a function for referencing the table data from multiple threads (shared access function).

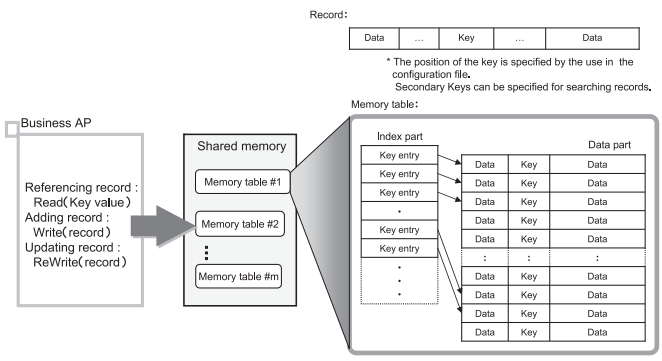

Fig. 2 Memory table structure.

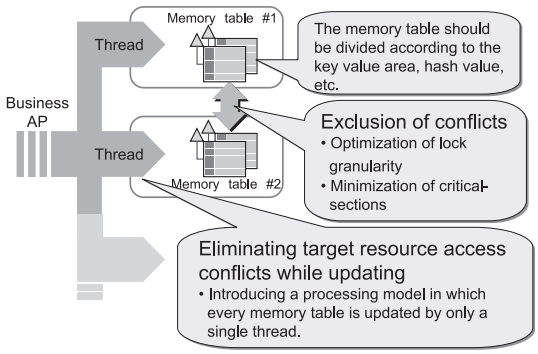

Fig. 3 Processing model.

The data structure and access processing method described above allow the TAM to improve processing parallelism and throughput performance by extending the CPUs and fragmenting table data according to the increase in data quantity to be processed.

3.3 Recovery Functions

The TAM has two data recovery functions: the replication function, which replicates table data on other servers throughout the network to cover the risk of data loss due to faults, and the journal function, which recovers a memory table from the update logs stored in the storage and the memory table images.

In the following, we will describe each of the recovery functions from the viewpoint of the availability and reliability of the system.

(1) Characteristics of the replication function

The replication function performs sequential synchronization of memory tables between multiple servers (**Fig. 4**).

The business AP updates data in the updating target memory table (master memory table) on the job server. The replication function transfers the contents of the data update (update log) to another server through the network to mirror the update in the replica memory table (replicated memory table) in the server at the transfer destination. The server keeping the master memory table at the transfer source is called the master server and the server keeping the replica memory table at the transfer destination is called the slave server. The contents of the memory table are synchronized with the replica table

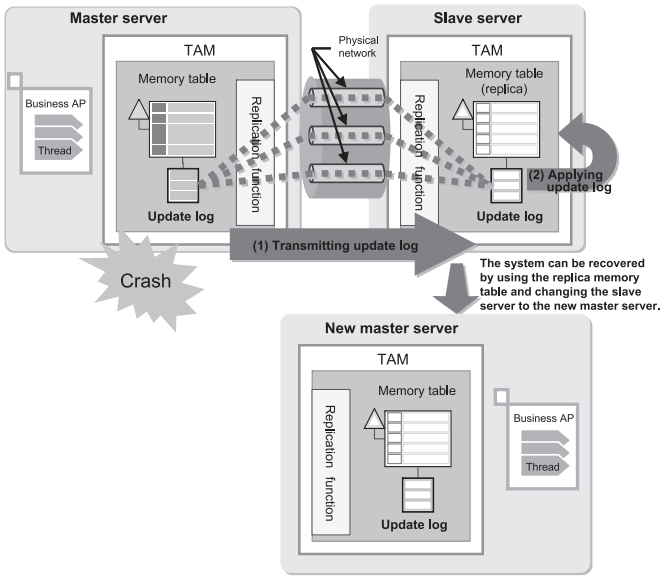

Fig. 4 Recovery by the replication function.

at the time of commitment by the business AP. This time lag of the synchronization between the master memory table and replica memory table prevents the loss of committed data in case of fault of the master server.

If a master server fault occurs, the system can be recovered by assigning the slave server as the new master server and using the replica memory table.

With recovery using the replication function, the memory tables are periodically synchronized at the time of commitment so that all that is needed for the data content of the memory table is to establish or discard non-established data. This makes recovery possible in less than a second [*1] , thereby considerably increasing system availability.

The redundancy of the replication function can be improved by specifying more than one replication destination or by using more than one line for replication.

(2) Characteristics of the journal function

The journal function holds in storage the information required for data recovery. The information used by the journal function is the update log and the image file of the memory table to be used as the basis for journal recovery (the TAM file).

Update log information is written cyclically to a raw vol-

[*1] Recovery time varies depending on the environment, configuration and usage. The specified recovery time is not guaranteed under arbitrary conditions.

ume, aiming to reduce overhead throughout the file system. When the write destination volume becomes full and is switched to another volume, the written information is automatically outputted to a log archive file by the TAM's archive control process (**Fig. 5**).
The TAM file is generated by a TAM operation command. The memory table image saved as a TAM file contains the contents of the memory table at the time the command is issued.
Should the memory tables of all the servers be lost due to multiple server faults, etc., the lost data can be recovered by loading the generated TAM file into master server memory and applying the update log information in the archive file to the memory table (roll forwarding) (**Fig. 6**).
With the TAM's journal function, dual output destinations can be specified for the TAM file, log volume and archive file. This can improve reliability by reducing the risk of data loss due to multiple faults compared to the case in which only the replication function is used.
With the TAM, either or both of these recovery functions can be selected according to the requirements of the system (availability, reliability).

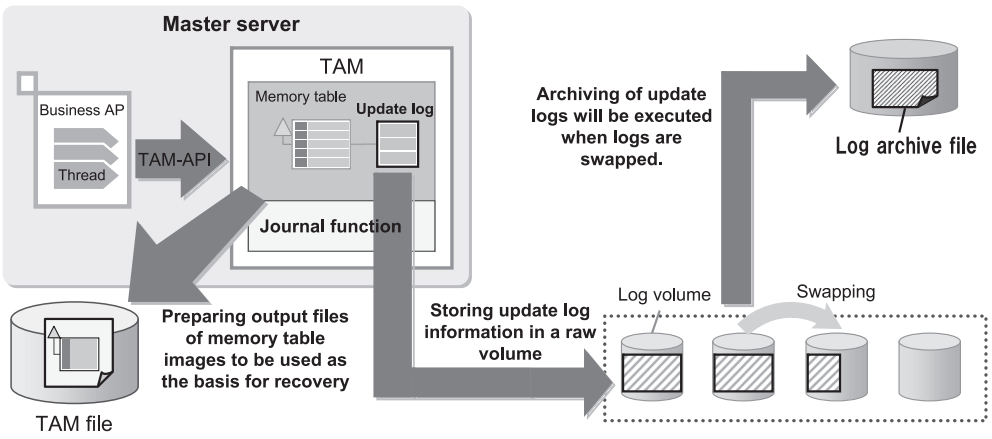

Fig. 5 Storage of Update Log Information by the Journal Function.

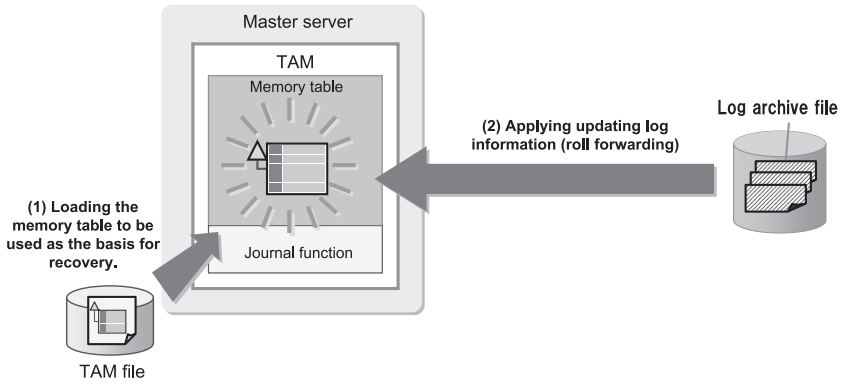

Fig. 6 Recovery by the Journal Function.

4. Conclusion

The memory DB product TAM boasts the achievement of being the platform-related product of a career-type system processing a large number of events or a financial-type system performing real-time distribution. With high reliability and high availability, this product is capable of real-time processing of big data.

We expect that the need to create new value and new business by utilizing in real time the growing amount of data in the big data era, as well as the data that has not yet been utilized, will rise every day in the future.

We are determined to continue feedback of our customers' opinions and enhance these technologies further so that our product can satisfy customers even more than it does today.

Authors' Profiles

OOSAWA Hideki
Manager
3rd IT Software Division
IT Software Operations Unit

MIYATA Tsuyoshi
Assistant Manager
3rd IT Software Division
IT Software Operations Unit

Information about the NEC Technical Journal

Thank you for reading the paper.
If you are interested in the NEC Technical Journal, you can also read other papers on our website.

Link to NEC Technical Journal website

Japanese

English

Vol.7 No.2 Big Data

Vol.7 No.2
September, 2012

Special Issue TOP