Ultra-high-speed In-memory Data Management Software for High-speed Response and High Throughput

● Yasuhiko Hashizume ● Kikuo Takasaki ● Takeshi Yamazaki
● Shouji Yamamoto

The evolution of networks has created demand for a capability to process huge amounts of data at ultra-high-speed, far exceeding the existing levels and what has commonly been considered possible. This data processing capability is giving rise to completely new modes of service, which in turn are leading to new ways of using information and communications technology (ICT). For a system that must rapidly process huge amounts of data, both high-speed response and high throughput are of course important, but high reliability must also be achieved at the same time. Leveraging its extensive experience in mission critical systems and its strength in advanced technologies, Fujitsu is helping the arrowhead trading system of the Tokyo Stock Exchange to operate stably through its new ultra-high-speed data management software, developed within the latest project scope (hereafter "this software"). Based on the concept of diskless operation, this software achieves large-scale data processing together with superb extensibility, accessibility, and reliability. This paper describes Fujitsu's approach to achieving high-speed response and high throughput, explains the concept of this software, and introduces the new technologies used in this software.

1. Introduction

Through the innovation of business models, there has been a drastic increase in the number of system users and transaction numbers in various fields and industries such as financial trading, credit, logistics, travel and telecommunication. In our information society where a further increase in the amount of information is predicted, high-speed response and high throughput at a higher level are requested even in mission-critical systems.

In such circumstances, Fujitsu is offering ultra-high-speed data management middleware (hereafter "this software") for large-scale data management applications with high reliability, connectivity and extensibility.

In this paper, we will introduce our approaches to realize high-speed response and high throughput as well as the new technologies used to achieve these features in this software.

2. Approaches to realize high-speed response and high throughput

This section describes the relationship between information and communications technology (ICT) evolution and changes in the data processing model as well as the system architecture considered for achieving high-speed response and high throughput (**Figure 1**).

2.1 Abstraction model of data processing system

Fujitsu has an abstraction model called a "classical data processing system" based on a POS management system as its origin. In this model, business process applications are configured in the following order: 1) entry processing, 2) master

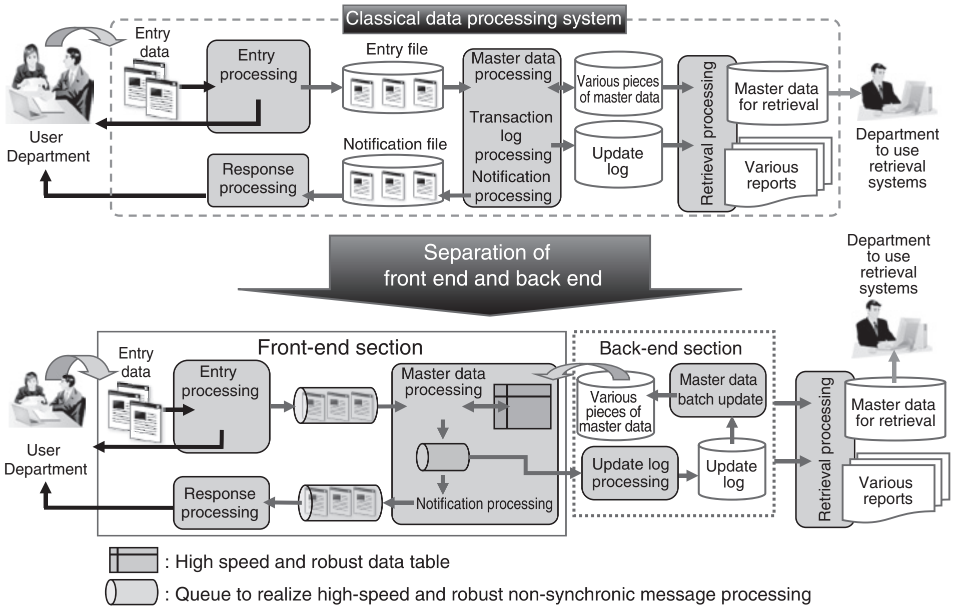


Figure 1
Implementation approach based on abstract model of data processing system.

data processing, 3) notification processing, and 4) response processing. Tasks that process data themselves are also integrated in this workflow. As for entry processing, the Internet has come to be used generally in step with the progress of real-time processing through the drastic advancement of networks.

While ICT and tools such as software have made significant progress in recent years, the classical model has continued to be used, at least in several data hand-over styles, such as the one between the entry processing and the master data processing or the ones among the retrieval processing tasks. Further, many technologies related to data linkage use disks.

2.2 System architecture for high-speed response and high throughput

We considered that high-speed response and high throughput could be achieved by replacing all of the aforementioned disk-based data processing by in-memory data processing. The key point is to separate the model into two sections, one for the front end, where high-speed response and high throughput are regarded as important, and the other for the back end, where data accumulation and utilization are the key roles. Based on this separation, we can identify the data in the front end that requires a high-speed response and it shall be deployed in the in-memory area. We identify the updated portion of “in-memory data” to be saved and write it into adequate master data file(s) in the back end. For each type of master data, persistence should be ensured by general-purpose DBMS.

Using such a universal abstraction model as the basis, we concluded that the most effective approach to achieve high-speed response and high throughput was to offer an in-memory non-synchronic message queue. This queue should be able to hand over the transactional data and in-memory two-dimensional table equivalent to master data file between processes.

Our approach based on the abstraction model of a data processing system is indicated in Figure 1.

3. Concept of this software

In the development of this software, we tried to achieve the following five merits at the same time besides achieving high-speed response and high throughput. Some of these requirements came from the software's use in mission critical systems.

1) Origin of the concept: Completely diskless processing

To achieve high-speed response and high throughput, completely diskless data processing is adopted. The data processing speed is drastically enhanced by processing all the data on a memory without generating I/O processing, instead of using a hard disk drive.

Generally speaking, the middleware products that achieved remarkable processing speed by placing all the data on a memory are normally called an "in-memory database." The software explained here can be considered as an in-memory database as well.

Typical in-memory databases write only the log into the disk on a periodical basis. This kind of log is used to maintain consistency of processing after restoration in the event of any system trouble. However, in this software, such a log is also implemented as in-memory so as to achieve high processing speed. In this sense, this product is "diskless" in every aspect.

2) Large-scale data management: Management from several hundred GBs to several TBs

To process all the data on an in-memory basis, this software needs to deploy data of a size ranging from several hundred GBs to several TBs on the memory. If the amount of memory needed exceeds the limit of a single physical server, it is necessary to deploy the memory across multiple servers. Nevertheless, we wanted to handle the data as if they were on a single server, so we need to use some kind of distributed processing that allows us to do so.

3) Reliability: Securing reliability by exploiting redundancy of hardware generally available

For this software, while a high level of availability, robustness, reliability, consistent operation and performance extensibility are sought after, only generally available hardware is used to run it, without using any special product for their components including semiconductor memory. We tried to fully exploit the performance of such hardware. By using generally available components, it is possible to improve economic efficiency and ensure flexibility for potential applications in future.

Such data-processing middleware to realize ultra-high-speed processing on an in-memory basis is going to be used in the financial industry and advanced systems in the technology field where extremely high performance is requested. If any system trouble should occur, it could affect not only a single organization but also possibly the whole of society, the economy and industries. A characteristic of in-memory data management middleware is that it carries a social responsibility associated with system reliability, because the range of impact in the real world is difficult to predict in the case of a system shutdown.

Therefore, technology to ensure swift resumption of business processes should also be more sophisticated to safeguard against any system disturbance.

4) Accessibility: High-speed and simultaneous access on a large scale

Applications requiring a high-speed response and high throughput are those that need to handle accesses from an enormous number of clients in a very short time. In the case of a Web application used by general consumers, for instance, accesses from many areas around the world may concentrate in a very short time. Or, an enormous number of computers as clients instead of human beings may access the system automatically in intervals in the order of milliseconds.

Therefore, it is imperative to establish a technology to process an enormous number of accesses on a parallel basis with an equal opportunity, efficiently, and with moderate response time distribution.

5) Extensibility: Swift handling of transaction

increases

In the financial industry and advanced systems in the technology field where in-memory data processing is needed, an unpredictable and drastic increase in the number of transactions may occur very often. Therefore, a speedy extension of performance within a short time is required that could not be considered in conventional systems such as performance extension with load distribution within 30 minutes.

4. New technologies for high-speed response and high throughput

In this software, we tackled the following five issues to realize large-scale data management with high reliability, high accessibility and high extensibility based on a completely diskless system.

4.1 Exceeding the limit of physical memory capacity

The technology to exceed the limit of memory capacity on a single physical server is indicated in **Figure 2**. Because this software needs to deploy data on its memory, it fails to construct a system if the amount of data exceeds the amount of memories available on a single physical server. For instance, even if the main memory with a maximum memory capacity of 256 GB is used, the data available for deployment is 128 GB (half the actual memory level) because redundancy should be ensured to enhance reliability. Further, among this 128 GB, some space should be spared for accommodating programs and such like instead of devoting all the space to data accommodation. The capacity of the main memory for each server has its limits. Therefore, to be able to deploy data in the order of several TBs, it is essential to have distributed deployment.

In this software, the following three technologies are used to distribute the data.

1) Table partitioning

This technology is used to split data of a single table into multiple parts to allocate them on multiple servers. With this technology, it is possible to deploy a table that exceeds the physical memory capacity of a single server.

2) Virtualization

This technology frees applications from the need to consider the relationship between physical servers and the place where actual data

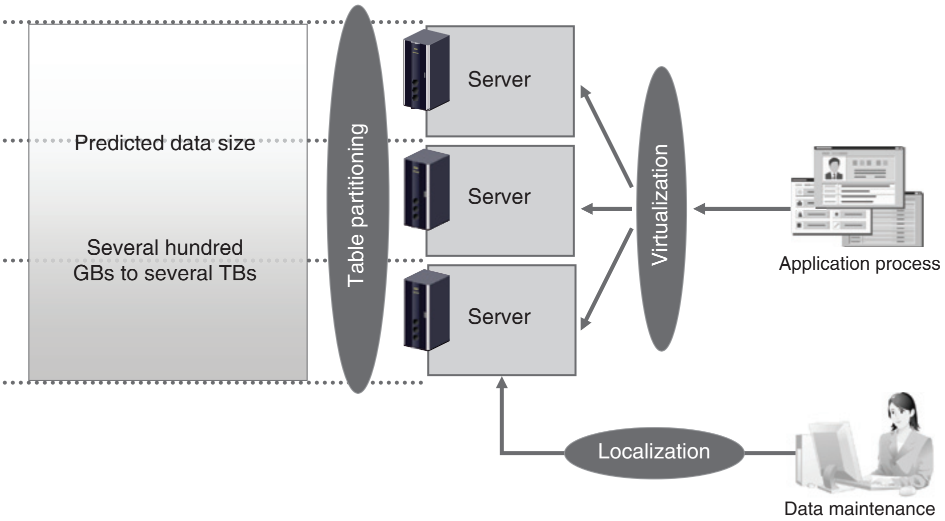


Figure 2
Technology to exceed limit of physical memory.

is accommodated. Because applications are not aware of physical partitions, no special handling of data is necessary for applications. This results in a lower workload when developing applications.

3) Localization

This technology enables independent processing of operation management in each partitioned unit. For instance, daily maintenance and operation management work such as partial correction of master data can be carried out without stopping the whole system.

4.2 Swift resumption of business process after failure

There are various failure including hardware failure, network failure, OS failure, middleware failure and business application failure. It is impossible to totally eliminate these failures. Therefore, it is essential to ensure abnormalities are detected and promptly switch systems or resources.

This software has introduced new technologies for "detecting and switching in a second" in addition to conventional PRIMECLUSTER as its base. PRIMECLUSTER is Fujitsu's cluster solution with a proven performance in a wide range of applications. First, to enable abnormalities to be detected within few seconds at a 100% probability, comprehensive HeartBeat[note 1)] diagnosis technology was developed. This technology can diagnose the comprehensive communication status including the diagnosis of Business LAN[note 2)] and synchronic LAN[note 3)] in addition to the HeartBeat diagnosis via conventional management LAN[note 4)].

Furthermore, data mirroring technology was used to resume business services within a few seconds in the event of a failure. This technology ensures the latest data is preserved on a stand-by system at all times by mirroring all the generated data. In addition, system availability is enhanced by preparing multiple stand-by systems for a single active system.

The following technologies support the comprehensive HeartBeat diagnosis technology and the data mirroring technology:

1) Cluster system

A technology to realize a high level of availability by using multiple computers

2) Failover

A technology to resume memory table manipulation at high speed (in the order of seconds)

3) Mirroring of in-memory data

Following the high-speed updating of memory table contents in an active system, this technology synchronizes the contents of memory tables on stand-by systems so that they can be kept equivalent to the original.

Technologies to support swift resumption of business process are indicated in **Figure 3**.

4.3 Localization of failure depending on location of disturbance

To enhance reliability, it is also important to localize the range of systems that may be affected by any disturbance. If this range can be narrowed down from the whole system depending on each disturbance, countermeasures for continuing business processes will be necessary only for that limited range and excessive investment can be avoided. The technologies to localize failure are indicated in **Figure 4**.

In this software, the following two technologies are used to carry out failover to address each section of failure or disturbance.

1) Three-layer cluster

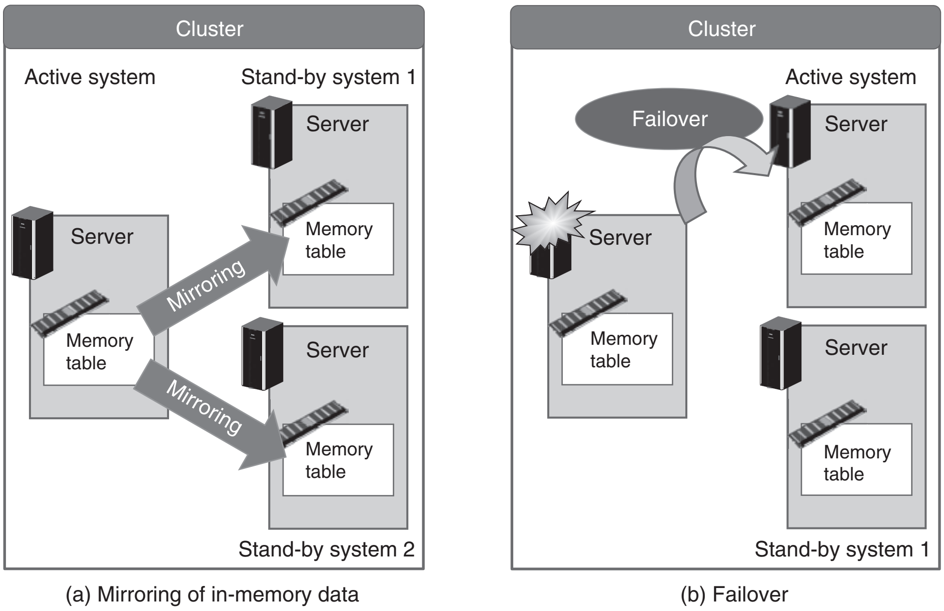
This technology performs a failover suitable for each failed section based on each layer (application layer, middleware layer and physical

note 1) Network signal emitted periodically to notify operators that computers and network components are operating normally.

note 2) LAN for use of sending and receiving business data.

note 3) LAN for use of mirroring data.

note 4) Control LAN among serves that constitute a cluster.



(a) Mirroring of in-memory data

(b) Failover

Figure 3
Technology to support rapid reopening of business.

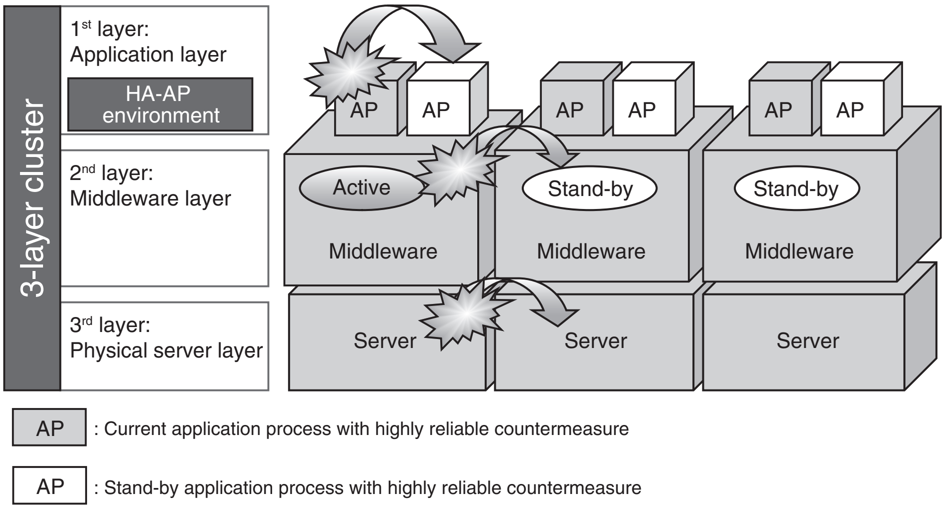

3-layer cluster
1st layer: Application layer
HA-AP environment
2nd layer: Middleware layer
3rd layer: Physical server layer
AP
AP
AP
AP
AP
AP
Active
Stand-by
Stand-by
Middleware
Middleware
Middleware
Server
Server
Server
AP
: Current application process with highly reliable countermeasure
AP
: Stand-by application process with highly reliable countermeasure

Figure 4
Technology to localize failure.

server layer).

2) High Availability - Application (HA-AP) environment

This technology ensures high availability by making application processes redundant. If an application process that executes business logics is abnormally terminated, in conventional systems, an additional application process will be rebooted to file up the lack of processing ability. However, when high-speed response in the order of microseconds and high throughput are requested, application process reboot would increase the load on each system resource, which would in turn result in degraded processing performance of the whole system.

To avoid this situation, a process for stand-by application is started up and pooled in advance. If any unexpected abnormal termination of an application process occurs in the current application, the supplementary process in the stand-by application assumes the operation. This technology ensures high reliability of applications through instantaneously resuming the business services in this way.

4.4 Balanced and speedy accessibility even during simultaneous access on a large scale

In a Web application, to realize high-level accessibility it is essential to ensure an enormous number of clients can be handled in microseconds. To achieve high-speed access performance when there is simultaneous access on a large scale, we extended and tuned our technologies for the database management system (DBMS) towards in-memory data management. Fujitsu has sophisticated DBMS technologies nurtured over more than half a century.

This software adopts a distributed server configuration based on networking of multiple servers. As such, response or throughput will ultimately be influenced significantly by the performance of control communications and data linkage performance among servers. This is the case even if in-memory access performance is reinforced. Thus, we developed a novel high reliability communication technology using User Datagram Protocol (UDP).

UDP represents a protocol in a transport layer of the TCP/IP family. In the conventional system, UDP communication is less reliable than the TCP communication, while it offers higher communication speed. To address this issue, various delivery confirmation patterns are implemented in this software for multiple recipient servers (active system, stand-by system etc.). By using an appropriate pattern for each situation and minimizing the overhead for delivery confirmation, highly reliable communication has been realized in UDP communication.

Further, to address the event of "packet lost" which tends to occur in UDP communications, this software adopts a unique delivery confirmation technology. It transfers multiple reproductions of the same packet to multiple channels at a time (i.e. redundancy) instead of starting to re-transmit the packet after the detection of "packet lost." In this way, the system ensures communication without performance degradation even in the event of an abnormality in communication channels. This results in both high-speed data communication and high reliability (**Figure 5**).

High-reliability UDP communication technology is also suitable for mirroring in-memory data.

4.5 Scalability for extension of processing performance

The following two technologies are used mainly to realize system capacity build-up within the extremely short time allowed to address any unexpected increase in transactions.

1) Dynamic scale out

This technology transfers an environment to execute a business application program to another cluster on an on-line basis. For instance, if memory capacity is expected to be insufficient because of an increase in the number of transactions, you may be able to give Cluster 1 a margin by migrating part of the business application from Cluster 1 to Cluster 2, which has more room to accommodate the load.

An example of the dynamic scale out is indicated in **Figure 6**.

2) Virtualization of table allocation

The unit of splitting memory tables (i.e. partition) is determined by the users who give consideration to the effective data configuration for their system from the standpoint of load distribution and risk distribution. However, application programs are still free from the need to be aware of where data is located physically.

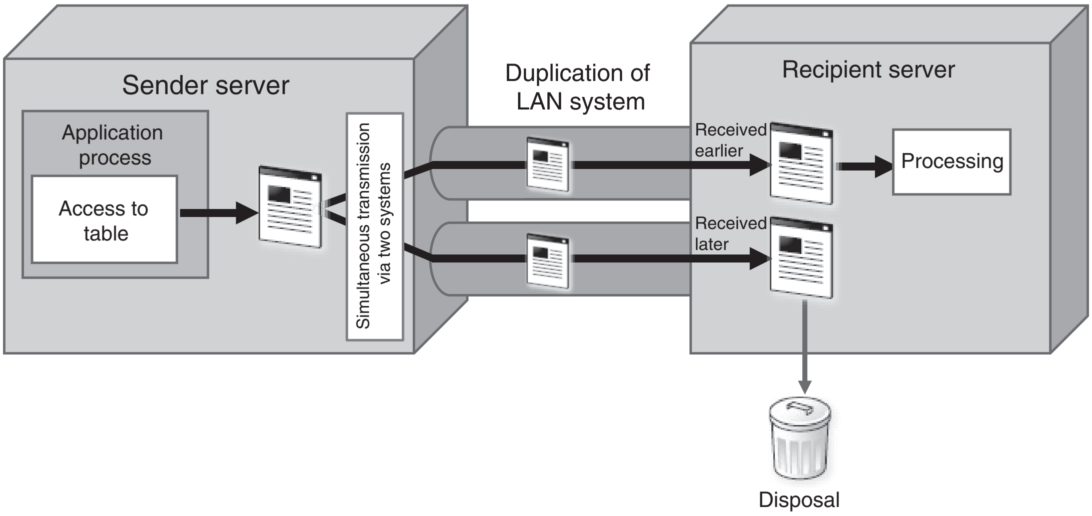


Figure 5
Highly reliable UDP communication.

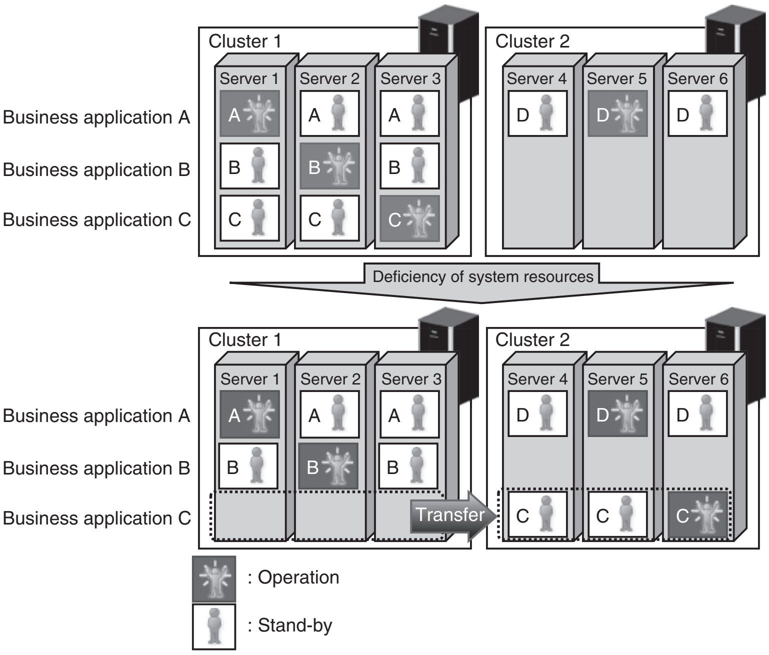


Figure 6
Dynamic scale out.

Through these features, the software achieves an unlimited extension of performance and easy scale out, which helps drastically reduce the risk of a total shutdown of business applications.

5. Conclusion

This software is in-memory data management middleware that aims to ensure high speed and reliability. Its response speed is between 10 and 100 times as high as the response speed of general purpose DBMS.

To achieve this speed, various types of data management technologies, network technologies and cluster technologies are used in this software in a sophisticated way. On the other hand, steady approaches of the conventional kind are important and also applied to this software, including programming technologies to reduce erroneous hits of cache that affect CPU average command execution time as well as the optimization of size and configuration of control tables.

Fujitsu has developed this software as a solution for markets that require ultra-high-speed processing of huge amounts of data that far exceed the existing levels and what has commonly been considered possible. Fujitsu plans to support customers' businesses through widely deploying the technologies developed in this software across other middleware in future.

Yasuhiko Hashizume
Fujitsu Ltd.
Mr. Hashizume is currently engaged in the development of middleware.

Takeshi Yamazaki
Fujitsu Ltd.
Mr. Yamazaki is currently engaged in the development of middleware.

Kikuo Takasaki
Fujitsu Ltd.
Mr. Takasaki is currently engaged in the development of middleware.

Shouji Yamamoto
Fujitsu Ltd.
Mr. Yamamoto is currently engaged in the development of middleware.