Development and Runtime Platform and High-speed Processing Technology for Data Utilization

● Hidetoshi Kurihara ● Haruyasu Ueda ● Yoshinori Sakamoto
● Masazumi Matsubara

Dramatic increases in computing power and network speed, along with advances in sensing technology, have broadened the range of devices that can connect to the Internet. Large quantities of diverse, time-sequenced data are flowing in from the Web and countless sensors, and there is a strong demand to rapidly and efficiently extract any valuable information from that data and to utilize that data for various types of navigation systems. Fujitsu Laboratories envisions a Human-Centric Intelligent Society and is building a cloud platform to support it. This article introduces an integrated development and runtime platform technology for utilizing large quantities of data, a technology for extracting parallelism from complex event processing as a base technology for high-speed processing, and a distributed-parallel technology for complex event processing and describes the goals, features, and effects of each. It also gives an overview of directions for future technology initiatives.

1. Introduction

Fujitsu is focusing on the “value of data” arising from the activities of objects and of people. It is important to derive new value from the large amounts of diverse data that, till now, were not collected or were collected but never used. There is increasing expectation that, with advances in ICT, such data can be used to help meet the traffic congestion challenge facing urban areas, effectively using energy resources, easing traffic congestion, and reducing medical costs.

However, a combination of human intelligence, ICT infrastructure, and a rapid cycle of hypothesizing and testing is required to extract the “value of data.” The ICT infrastructure should have a simple interface for handling data and enable flexible system scaling, from small start-up to full service deployment.

Fujitsu Laboratories envisions a Human-Centric Intelligent Society and, with such a society in mind, is conducting research with the goal of “connecting and sharing information to expand ICT application fields and create new markets.”[1]

This article first gives an overview of development technologies, identifying inherent issues and how they can be addressed. It then describes three technologies based on these development technologies, including potential applications and their characteristics. These technologies are “an integrated development/runtime platform,” “parallelism extraction from complex event processing,” and “distributed-parallel complex event processing.” Finally, the effects of these technologies and technical directions needing to be addressed in the future are discussed.

2. Overview of development technologies

Figure 1 illustrates the circular nature of data utilization. It starts with batch processing analysis of the collected data. Features of the data are then detected in real time using knowledge and rules from the analysis. These two steps are then repeated. An example use case is a service that analyzes tendencies in point-of-sale (POS) data and uses the results to issue coupons to particular customer segments when they approach a certain location. Since all the details of the service cannot be established at service start-up, they must be refined in a cycle of hypothesizing and testing. Maintaining real-time performance as the service scale expands requires high-speed data processing using cloud or other computing resources that can be increased easily.

 FUJITSU Sci. Tech. J., Vol. 50, No. 1, pp. 16–23 (January 2014)

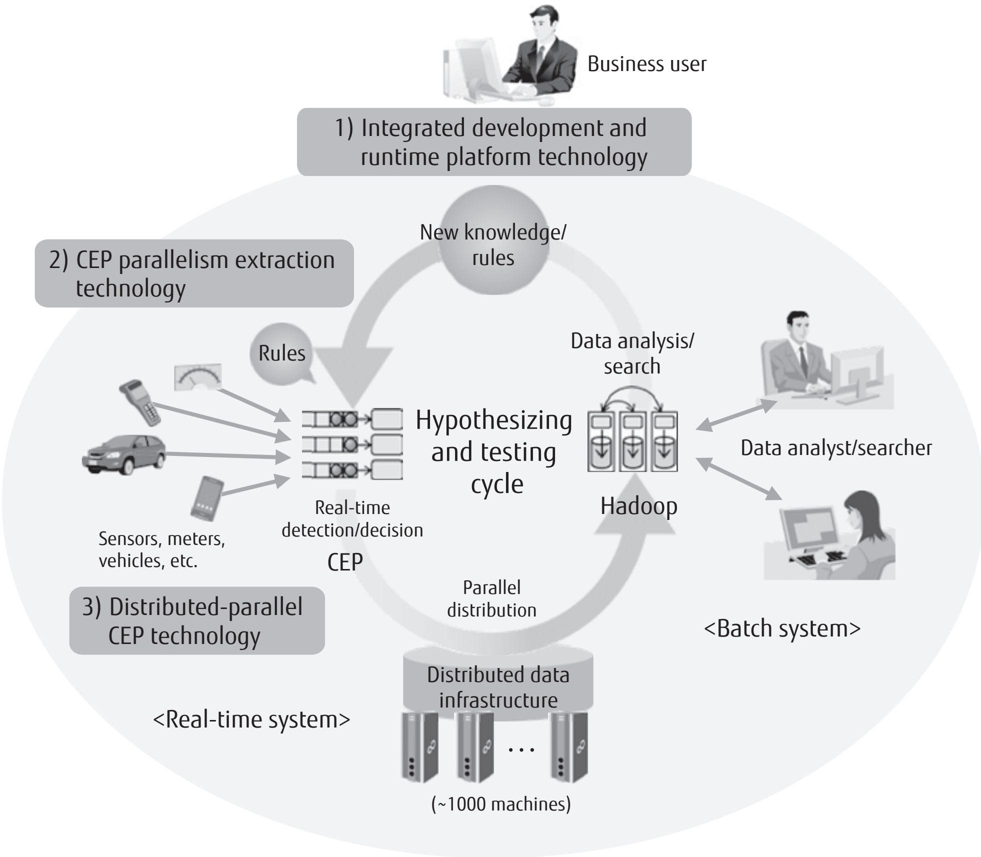

Figure 1
Overview of development and runtime technology.

Such a service would utilize large amounts of POS and customer location data, so an ICT infrastructure must be used, including a batch processing system such as Hadoop (an open-source parallel batch processing infrastructure), and a real-time processing system such as a complex event processing (CEP) system.

The issues inherent in meeting these requirements and our approaches to addressing them are discussed below.

2.1 Issues

1) Ease of service development

Although business users developing a new service may have a wealth of business knowledge, they are generally unfamiliar with individual ICT processing technologies. It is thus difficult for them to learn a processing language supported by Hadoop or CEP and to develop and/or customize their service.

Moreover, parallel processing must be used to maintain high performance as the scale of the business and the amount of data processed increases. Hadoop has advanced parallel processing technology and uses it automatically, but parallelization must be designed carefully in order to obtain adequate performance when processing complex events, and this is difficult for most business users.

2) Achieving both high performance and continuous operation

High-speed data processing is needed to maintain real-time performance of a service with CEP. It is also important to maintain continuous operation as the service expands from small start-up with a few servers to full service deployment with many servers and as the scale of the business expands and contracts. It is difficult to achieve both high performance and continuous operation, so services have normally been stopped, reconfigured, and restarted when the upper limit of the system was reached.

2.2 Approaches

1) Ease of service development

A development and runtime platform that generates programs automatically and independently of the processing language was developed [1) in Figure 1]. It provides a data flow diagram interface that defines the inputs and outputs for data processing, enabling business users to easily customize the analysis process and the service provision conditions, such as when coupons are issued. Another technology that was developed extracts parallelism from a description of complex event processing and automatically recommends combinations of parallel operations [2) in Figure 1]. By simply describing a data flow diagram, even business users can implement high-speed real-time services that use parallel processing.

2) Achieving both high performance and continuous operation

A function was developed that adjusts computing resources and rapidly distributes the load dynamically, without stopping the system, in accordance with the volume of input events [3) in Figure 1]. It maximizes performance by using the parallelization extraction technology. Business users can thus provide continuous, stable, real-time services, independent of the volume of input events, and computing resources can be used more efficiently.

3. Integrated development and runtime platform technology

Systems are conventionally built in accordance with the type of processing to be done, such as using Hadoop for batch processing and CEP for real-time processing, and development platforms are not unified. To enable even business users to develop and customize services easily, technology was developed to integrate the development and runtime platforms, enabling the cycle of hypothesizing and testing to be speeded up.

As shown in **Figure 2**, the business user defines the process details using a data flow diagram on the basis of the application properties (parameters are defined for each process) and determines the process type (batch or real-time). An automatic program generation pattern is then used to automatically generate either a batch program (Hadoop) or a real-time program (CEP). Next, data-type conversion and other processes needed in accordance with the process details are added. The program and data are then distributed to the appropriate batch or real-time processing environment for execution.

The automatic program generation patterns, which are a core component of the development and runtime platform, are explained below using the example of analyzing trends in accumulated POS or other data. The business user defines processing parameters as properties for trend analysis, such as the input data types and the initial number of clusters (upper-right in Figure 2). This information is combined with the set of automatic generation patterns pre-configured in the development and runtime platform (center-right) to generate the program automatically. For example, a pattern to generate a trend analysis program automatically could add pre-processing to select columns from the database of accumulated data and convert them into vector form and add post-processing to summarize the analysis results in a table. In this way, only the minimum necessary processing parameters are specified; the pre-, post-, and main processing programs are generated automatically by the pattern. The resulting program is distributed to the runtime platform and executed.[2)–4)]

The data-flow diagram editor and other components of the development and runtime platform are implemented using the Eclipse Remote Application Platform (RAP), so business users can use them as a Web service in the cloud and do not need to install them. Moreover, program transformations for the development and runtime platform use tools such as Eclipse Acceleo, which is a model transformation engine conforming to OMG MTL standard specifications, so it is easy to develop automatic generation patterns that conform to accepted standards. Functions that are essential for the hypothesizing and testing cycle, such as repository management and version control for the data-flow diagram and data schema definitions, are also implemented.

Use of this technology enables the time from development of the analysis program to development of the complex event processing program to be reduced to approximately one-fifth (from eight weeks to 1.5 weeks). It also enables any of the parameters to be easily changed without reprogramming, making it easier to iterate the hypothesizing and testing cycle in the development platform. For example, knowledge

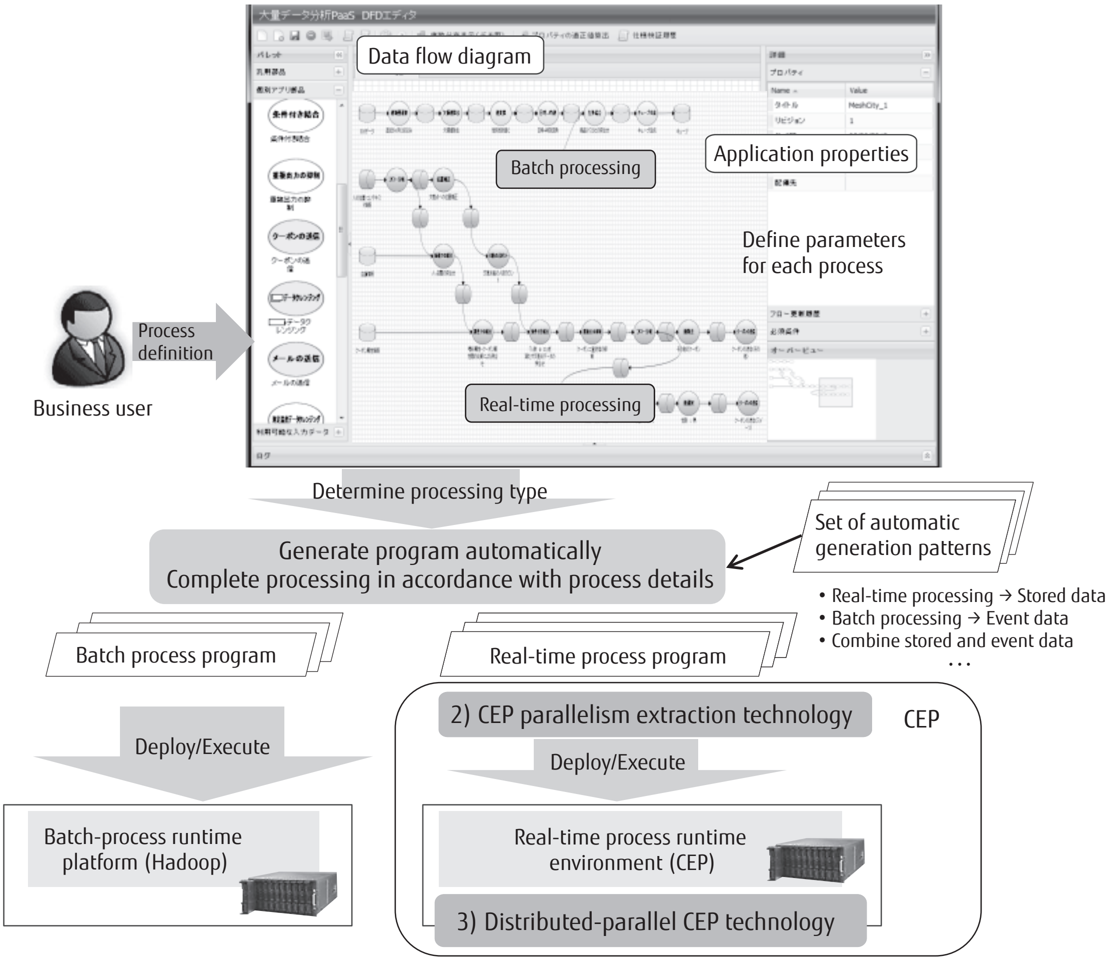

Figure 2
Integrated development and runtime platform operation.

gained from the analysis results can be readily applied to the event detection conditions.

Work is currently underway to enhance the set of automatic generation patterns to be more general and extensible so that they can be used as a cloud platform (i.e., platform-as-a-service [PaaS]) for various types of data utilization.

4. Extracting parallelism from complex event processing

To maintain high performance even as the amount of data being processed increases, parallel processing has to be implemented in the real-time processing programs generated automatically by the integrated development and runtime platform. A technology was thus developed to extract parallelism from the programs and to recommend effective ways of using this parallelism (**Figure 3**).

In a distributed parallel CEP system[5] developed earlier at Fujitsu Laboratories, a single process (query) is shared among several servers, and events are processed in parallel by distributing them among several servers, thereby enabling events arriving in real time to be processed rapidly. For example, in the coupon-issuing example, the ID of a pedestrian (phone number, etc.) or of a regional area can be used as the key to partition processing by person or region, respectively.

Automatic partitioning technology from the parallel database field can be extended and used to extract this sort of parallelism automatically for event processing. An identifier (key) suitable for parallel processing is extracted from the description of a process, such as grouping (group by) or joining, and the value of the key in the data is used to determine which server

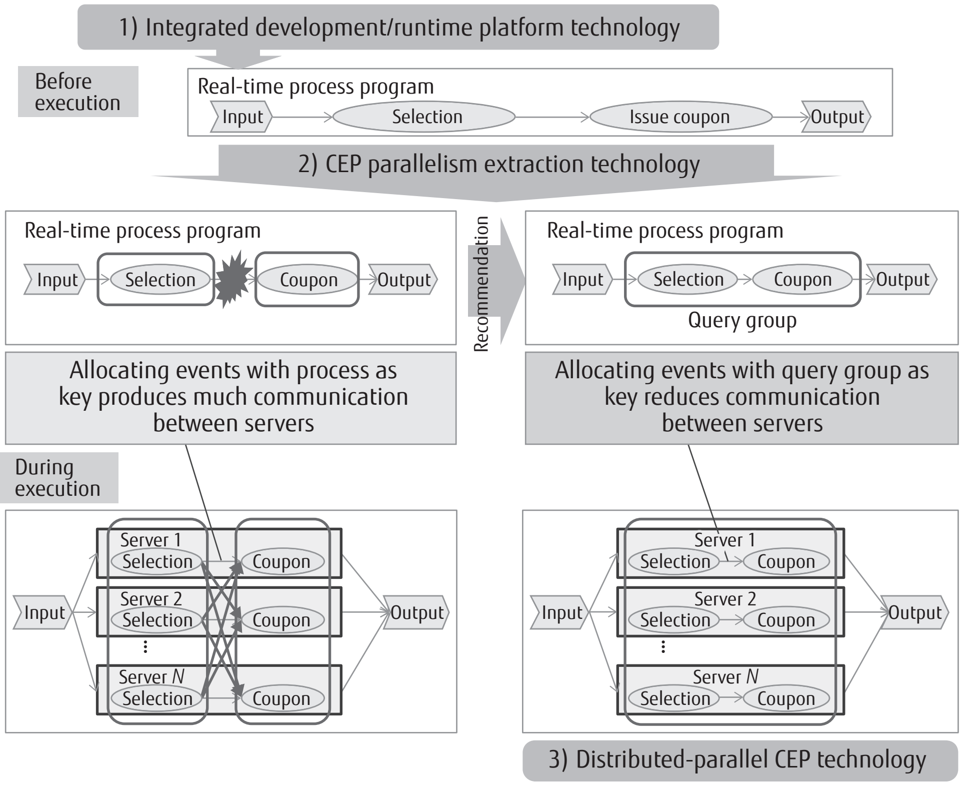

Figure 3
Technology to extract parallelism from CEP.

will be used to process the data. This technology was extended to event processing functions such as event-pattern detection and filtering, making it possible to extract keys suitable for parallel processing and to process events in parallel by using the key values.

This technology can also be used to obtain recommended optimal combinations of keys for multiple processes (a distribution strategy). It is generally easier to balance the load and achieve higher parallel performance by dividing a process into smaller units and selecting keys by sub-process, but with event processing, better performance can be achieved by reducing communication between processes. Thus, selecting key combinations that enable execution of as many processes as possible on the same server reduces the amount of communication required and increases the overall processing efficiency. Processes gathered together in this way are referred to as a *query group*.

The CEP parallelism extraction technology was used to extract parallelism automatically from all queries in the coupon-issuing program and to recommend query groups with the pedestrian ID or the regional area ID as the key. This is normally done by a parallel processing specialist, so this technology eliminates the need for the high-level knowledge and tuning normally required for parallelization design, reducing development time.

The performance of sample programs, each allocating events differently between processes, was measured to evaluate the increase in performance gained by optimizing the combination of keys used to distribute a program across multiple processes. Allocating using a key by query group resulted in a 60% reduction in the amount of communication compared to using a per-process key, increasing processing efficiency by a factor of 3.5. This corresponds to an increase in processing efficiency of 16 times compared to using no parallel processing.

A future direction is to develop technology to extract even more parallelism and achieve higher processing efficiency by using trends in the data being processed and not simply the description of the real-time processing program.

5. Distributed parallel complex event processing technology

In the distributed parallel CEP system, query processing that cannot be distributed is processed on a single server, and query processing that can be distributed is distributed over multiple servers using the distribution strategy described in the previous section (**Figure 4**).

As described above, the server to be used to process the data is determined on the basis of a query-group key when distributing processing. Thus, by moving the keys being handled by each server in suitably small units, accurate load balancing among multiple servers and continuous operation can be achieved. The number of keys is large compared to the number of servers, so the load can be moved in units in accordance with the key and distributed with very fine granularity by determining which server will process which key values. However, if the load is moved in key units with granularity that is too fine, communication and other overhead can increase and result in data processing delays. Because of this, the data is divided into approximately 1000 partitions (key partitions) using a specified distribution method such as a key hash function (a computation that generates a fixed-length random number from arbitrary data), and the load is moved between servers in units of those partitions. If an event related to a key partition arrives while the key partition is being moved, the event is forwarded to the new server after the migration has completed.

High performance is maintained by determining the number of servers needed to operate the system without interruption for the current operating conditions and distributing the load equally among the servers. Thus, in this parallel CEP system, the execution manager periodically collects usage state data from each CEP engine for resources such as CPU, memory, and network. It then determines whether load redistribution is needed, taking into consideration this resource-state data as well as the input event load state and query characteristics. If it determines that redistribution is needed, it calculates a suitable range for the number of servers, adjusts the number if the current number is not in this range, and then balances

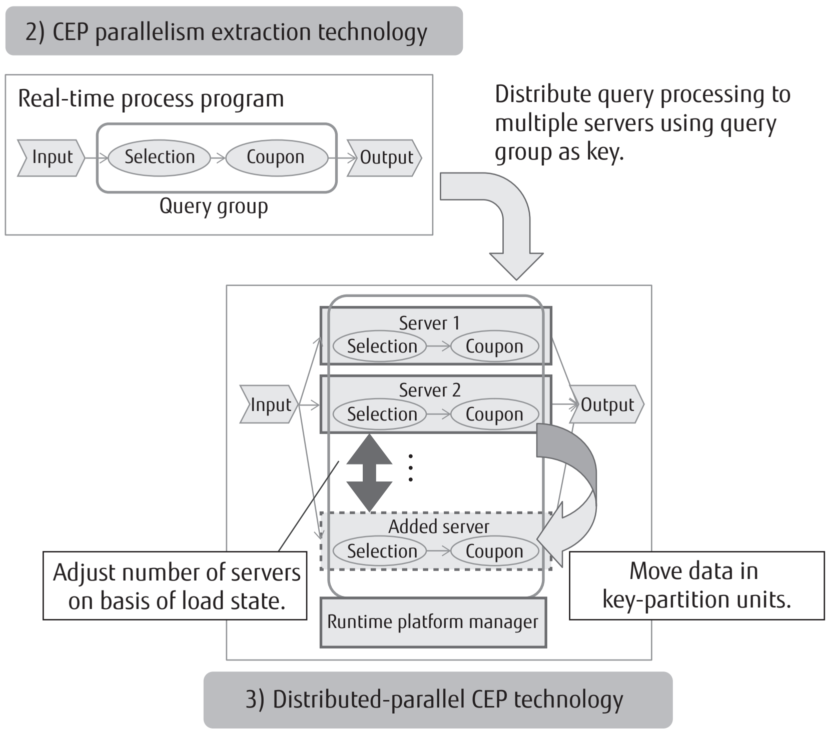

Figure 4
Parallel CEP system structure.

the overall load across the servers (by migrating key partitions).

We evaluated the performance of a parallel CEP system with these functions by using it to implement the coupon-issuing application described above. It was used to match pedestrian location and preference data with product information from surrounding shops (up to 10 000 shops) and issuing coupons appropriately. Up to eight servers were used, and pedestrian data events were input, gradually increasing in number from 50 000 to 350 000. The results confirmed that, as the load increased, the number of servers increased as needed, and the load was dynamically distributed across them. The system operated without interruption even while the load was being distributed. It processed queries with a latency of between 100 and 200 μs, maintaining stable, high-speed processing.

This technology enables dynamic configuration changes in accordance with the load state of service utilization, while maintaining real-time performance. A prototype system[6] built in 2011 required a long time and approximately double the resources when dynamically changing the configuration, but these problems are eliminated with the current method.

Practical testing based on several examples is planned, and functions will be developed that enable easier detection of higher-level patterns.

6. Conclusion

These technologies simplify the development of services through use of the hypothesizing and testing cycle, reducing development time by approximately 80% in a scenario where coupons are issued on the basis of POS data analysis. They also enable real-time service with continuous operation to be maintained even as the processing load increases.

We are currently integrating these technologies and targeting several projects to test them, such as services utilizing location data. The market for utilizing big data is maturing rapidly, and great value can be produced depending on how big data is used. Means for producing even greater value must be nurtured by combining these technologies with practical knowledge and expertise. As the amount of sensor data increases and virtualization technology advances, it will become more important to optimize overall systems, including cloud infrastructure, networks, and terminals. We will continue to apply and extend the technologies described here and to work on resolving the technical issues standing in the way of such overall optimization.

The part of this research related to distributed parallel complex event processing technology was conducted under contract as part of a Japanese Ministry of Economy, Trade and Industry project, "Development and Verification of Next-Generation Reliable, Energy-saving IT Infrastructure Technology," in FY2010 and FY2011.

References

1) Y. Sakashita et al.: Cloud Fusion Concept. *Fujitsu Sci. Tech. J.*, Vol. 48, No. 2, pp. 143–150 (2012).
2) Y. Nomura et al.: Massive Event Data Analysis and Processing Service Development Environment Using DFD, Services (SERVICES). 2012 IEEE Eighth World Congress on Services, pp. 80–87, 2012.
3) K. Kimura: A Method of DFD to Software Components Transformation using Multi-query Integration. IPSJ Software Engineering Symposium 2012 Proceedings, 2012, pp. 1–8, 2012 (in Japanese).
4) K. Kimura et al.: Multi-Query Unification for Generating Efficient Big Data Processing Components from a DFD. Cloud Computing (CLOUD), 2013 IEEE International Conference on Cloud Computing, pp. 260–268, 2013.
5) Ministry of Economy, Trade and Industry: FY2011 Next Generation High-reliability, Energy-saving IT Infrastructure Technology Development Report: R&D on Large-Scale Data Stream Processing Infrastructure (in Japanese).
http://www.meti.go.jp/policy/mono_info_service/joho/cloud/2011/02.pdf
6) S. Tsuchiya et al.: Big Data Processing in Cloud Environments. *Fujitsu Sci. Tech. J.*, Vol. 48, No. 2, pp. 159–168 (2012).

Hidetoshi Kurihara
Fujitsu Laboratories Ltd.
Mr. Kurihara is engaged in research and development of application development and runtime platforms.

Yoshinori Sakamoto
Fujitsu Laboratories Ltd.
Mr. Sakamoto is engaged in research and development of parallel complex event processing technology.

Haruyasu Ueda
Fujitsu Laboratories Ltd.
Mr. Ueda is engaged in research and development of distributed processing for data utilization.

Masazumi Matsubara
Fujitsu Laboratories Ltd.
Mr. Matsubara is engaged in research and development of parallel complex event processing technology.