Big Data Processing in Cloud Environments

● Satoshi Tsuchiya ● Yoshinori Sakamoto ● Yuichi Tsuchimoto
● Vivian Lee

In recent years, accompanied by lower prices of information and communications technology (ICT) equipment and networks, various items of data gleaned from the real world have come to be accumulated in cloud data centers. There are increasing hopes that analysis of this massive amount of data will provide insight that is valuable to both businesses and society. Since tens of terabytes (TBs) or tens of petabytes (PBs) of data, big data, should be handled to make full use of it, there needs to be a new type of technology different from ordinary ICT. Furthermore, as important services such as social infrastructure services should keep running 24 hours a day, 7 days a week, technology to dynamically change system configurations is also required. Fujitsu and Fujitsu Laboratories are working on developing basic technologies and application-promoting technologies for processing big data in a cloud environment. In this paper, we introduce two fundamental technologies: distributed data store and complex event processing, and workflow description for distributed data processing. We hope this gives a perspective on the direction in which this new field should head.

1. Introduction

One essential quality of cloud computing is in aggregation of resources and data into data centers on the Internet. The present cloud services (IaaS, PaaS and SaaS) realize improved execution efficiency by aggregating application execution environments at various levels including server, OS and middleware levels for sharing them. Meanwhile, another approach of aggregating data into clouds has also been launched, and it is to analyze such data with the powerful computational capacity of clouds.

In this way, cloud is now in the phase of expanding from application aggregation and sharing to data aggregation and utilization. To make full use of data, tens of terabytes (TBs) or tens of petabytes (PBs) of data need to be handled and a new type of technology different from ordinary information and communications technology (ICT) is required.

This paper presents distributed data store and complex event processing, which are basic technologies for big data processing in cloud environments, and the research by Fujitsu Laboratories of Europe on workflow description for data processing. This should provide a perspective on the direction in which data processing technology will develop in the future.

2. Overall picture of big data processing in cloud environments

In recent years, along with the lowering of prices of ICT equipment and networks, various items of data from the real world have come to be accumulated in cloud data centers. For example, information from position sensors of the Global Positioning System (GPS) mounted on mobile phone handsets or automobiles and transaction records from store cash registers are

stored along with the location and time of their generation, and transferred via networks to data centers, where they are accumulated. These data can be analyzed in terms of time series and associated with factors such as purchase behavior of individuals so as to estimate what action such individuals are likely to take. In this way, it is beginning to become possible to derive valuable information such as estimates of the purchase behavior of individuals from data, which have so far been no more than records. These massive amounts of data are also called big data. According to a trial calculation, the amount of event data generated in the U.S. is estimated to be 7 million pieces per second, which adds up to a few tens to hundreds of PBs per month if accumulated as they are without compression.[1] This value is not equal to the amount of data actually transferred to data centers and processed. However, the future availability of such detailed data is raising hopes for the acquisition of valuable information for enterprises and individuals, such as estimates of what a person will purchase, and where, next Thursday.

For some time, data processing has been utilized in various ways. For example, POS data analysis is used as the basis for planning strategic rollouts in the convenience store industry. Items that have recently been attracting attention include processing event stream (flow) data, such as the position of a constantly moving person observed over a time series, in addition to accumulated numeric data such as inventory and purchase (stock), and targeting to produce individual results for individual persons (**Table 1**) rather than to optimize corporate activities.

The structure of a system that realizes big data processing on clouds should combine subsystems for judgment, accumulation and analysis with a large number of servers in order to handle massive event streams (**Figure 1**).

Subjects in the real world include mobile phone handsets and automobiles, and their numbers may be enormous possibly at 100 million units for mobile phones and tens of

Table 1
Subject and purpose of big data processing.

Purpose \ Subject	Stock (value, relation)	Flow (event, time series)
Group	Display shelf arrangement (use of POS)	Smart grid power allocation
Individual	EC site recommendation	Card fraud detection, behavioral targeting ads

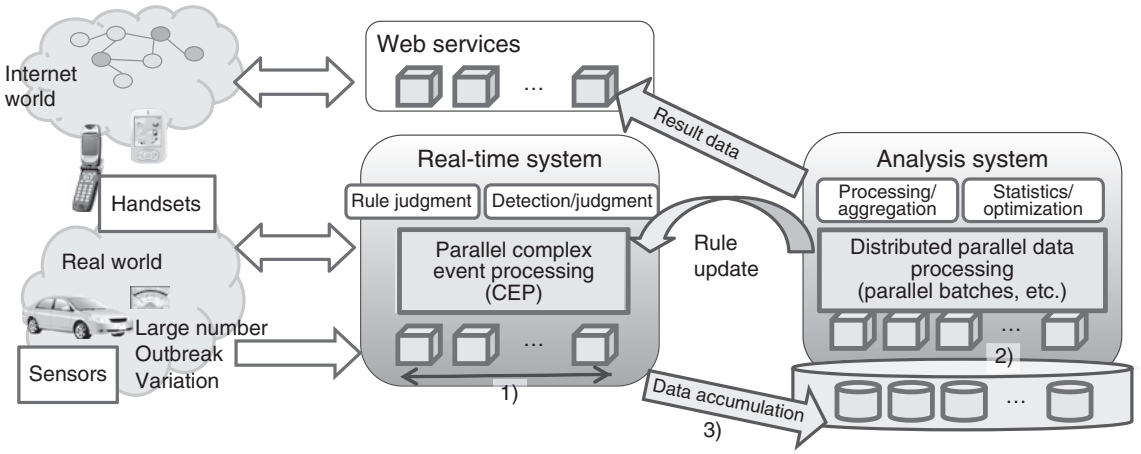

Figure 1
Overall picture of massive data processing on cloud.

millions of units for automobiles. One of the characteristics of event generation is that such events may occur unexpectedly, as in popular singers' concerts, and the amount of events may significantly vary depending on the season or time of day.

Accordingly, judgment processing must be capable of handling up to millions of events per second, which is a social system level, and making an immediate judgment without failing to respond to unexpected generation or rapid variation in the number of events. Accordingly, we need to have a number of servers running in parallel so as to flexibly change system configurations according to the demand without stopping services [1) in Figure 1].

For analysis processing, distributed parallel processing with hundreds to thousands of servers is required in order to perform statistical analysis of tens of TBs or PBs of data in a short time [2) in Figure 1].

Furthermore, the system must have high writing performance and it must be possible to flexibly expand it according to the required capacity so that the rapidly increasing amounts of data can be stored. The system must also be able to efficiently supply data for hundreds or thousands of analysis processes executed in parallel [3) in Figure 1].

The following sections describe the core of the basic technologies that meet these requirements.

3. Distributed data store

Big data processing on clouds may involve hundreds of entities such as application servers accessing data, and this leads to the generation of a massive amount of read and write requests. For that reason, it is said to be necessary to put tens to hundreds of servers for data storage in place to appropriately distribute the read and write load and to ensure that failure of any of the large number of servers does not stop the entire service.

One representative example of distributed data store is distributed key value store (KVS). With this technique, a data structure composed of keys and values is distributively stored in a number of servers and read and write take place in one server according to the request specified by a key to return a response (**Figure 2**). It offers high performance because the load can be prevented from concentrating on one server even if read and write requests from many users arrive at the same time. Another characteristic of this technique is its resistance to failure because data are copied and stored in other servers, and failure of one server does not lead to loss of the data. Provided with these features, distributed KVS is often used for storing session information of large-scale Websites such as online shopping sites, which need to process accesses of many users. For systems that handle sensor events, it is used where many read and write requests are generated such as portions for recording sensor data and for presenting the results of analysis to outside parties by using Web applications.

Distributed KVS achieves this high performance and availability by distributing data among multiple servers, but distributing data makes some specific types of processing difficult at the same time. A typical example of this type of processing is aggregation processing, in which multiple data are aggregated to produce results. With distributed KVS, where data are distributed among servers, a great many pieces of communication between servers are generated for aggregation, which takes time [Figure 2 (b)].

To address this problem, Fujitsu Laboratories has worked on developing a technique to realize high-speed aggregation with distributed KVS, and achieved a research prototype offering about eight times higher speed than that of the existing method.[2)] With the new technique, several basic operations (such as rekey, map, filter and reduce) that efficiently run in distributed KVS have been used and combined to realize aggregation processing. The individual

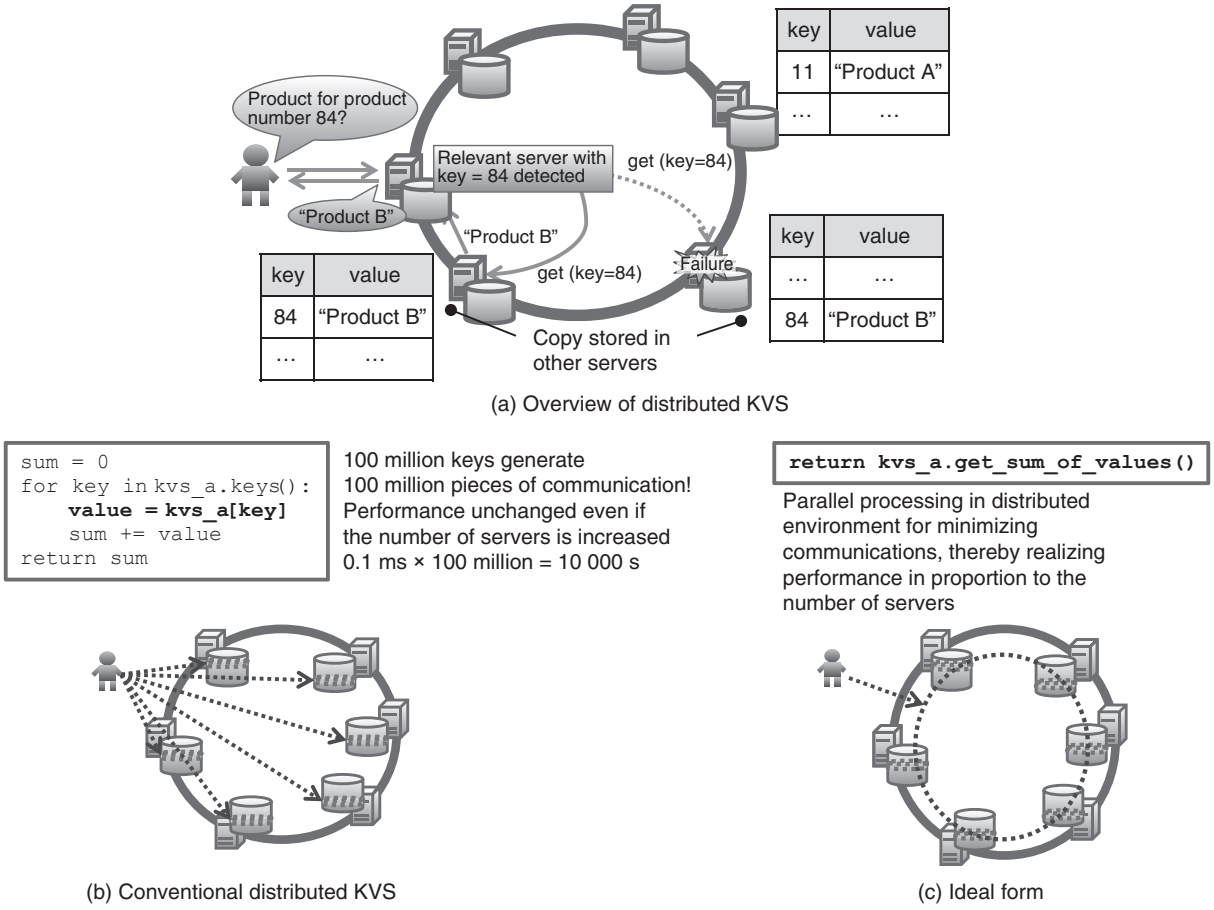

Figure 2
Overview of distributed KVS, and ideal form of aggregation.

basic operations are designed to efficiently run in parallel in distributed KVS and the entire aggregation processing is efficiently run in distributed KVS.

We conducted an experiment in which recommendations (of relevant pages) in a social networking services (SNS) were assumed and a log of 1 million user accesses was summarized. By doubling the number of servers, the processing time was halved, which confirmed that adding servers linearly improved performance.

As compared with other distributed batch methods, we confirmed a performance improvement of about eight times (**Figure 3**).

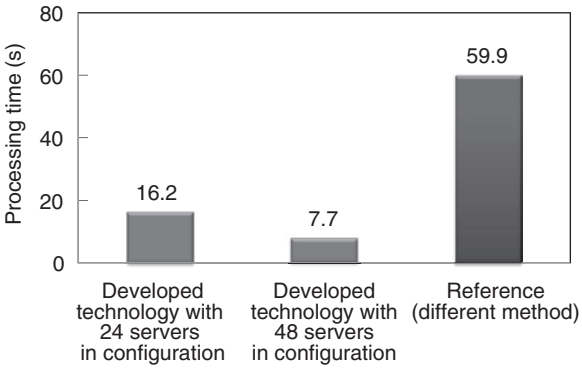

Figure 3
Aggregate results of measuring 1 million pieces of data.

4. Parallelization of complex event processing (CEP)

Complex event processing (CEP) refers to technology that processes and analyzes in real time complicated and massive event series that are constantly generated in real-world activities and operations. Possible applications of this event processing technology include, in the field of traffic systems, for example, a large amount of ever-changing location information of individual automobiles gathered to analyze the state of traffic congestion in respective roads, based on which a system can present routes to avoid congestion.

In the future, social services that make use of cloud will need to be capable of handling events, which will further grow in quantity, even if the load varies significantly or events are unexpectedly generated. In addition, critical social systems such as those for detecting signs of disasters or disaster prevention will have to run 24 hours a day, every day. Accordingly, Fujitsu and Fujitsu Laboratories have acknowledged the importance of technology for dynamically distributing event processing load in cloud environments to ensure an ability to operate in real time without stopping services. We have positioned it as technology that differentiates us from our competitors and are conducting research and development.

In a processing model as shown in **Figure 4**, this technology allows the load of event stream processing [set of data manipulation rules (queries)] to be dynamically distributed according to the event stream characteristics and load, thereby optimizing the processing.

CEP has a basic structure as shown in **Figure 5**.

Dynamic load balancing of CEP has the following technical challenges.

- Dividing the processing state (or simply "state") of event processing engine and its movement (order of event arrival and operation results, for example)
- Ensuring the order of event arrival and synchronization of time information

In FY 2010, we prototyped the basic operation of dynamic load balancing of CEP. To summarize the important points, we used a method in which the current and the extra systems run in parallel. The rules and data to be used in the CEP engine are copied from the current to the extra system. Then, events that arrived during parallel operation are delivered to

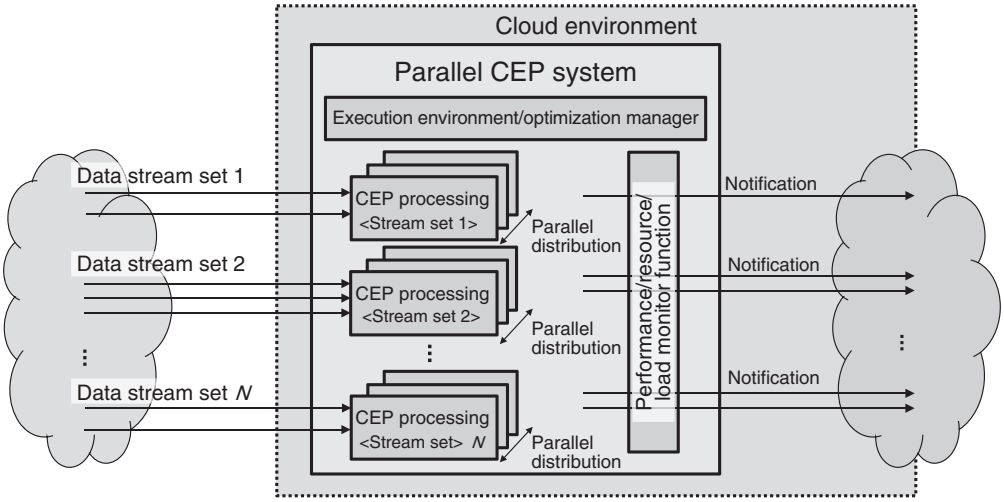

Figure 4
Configuration of parallel CEP system.

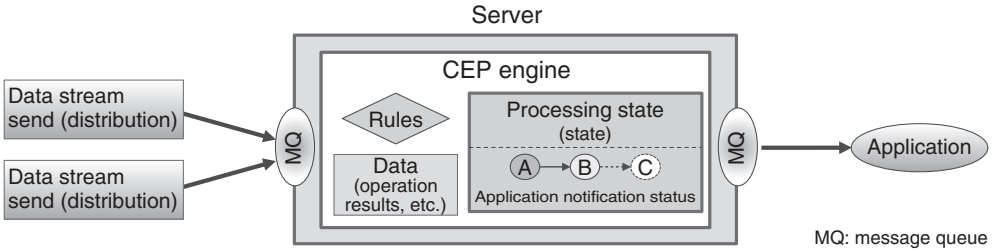

Figure 5
Basic structure of CEP.

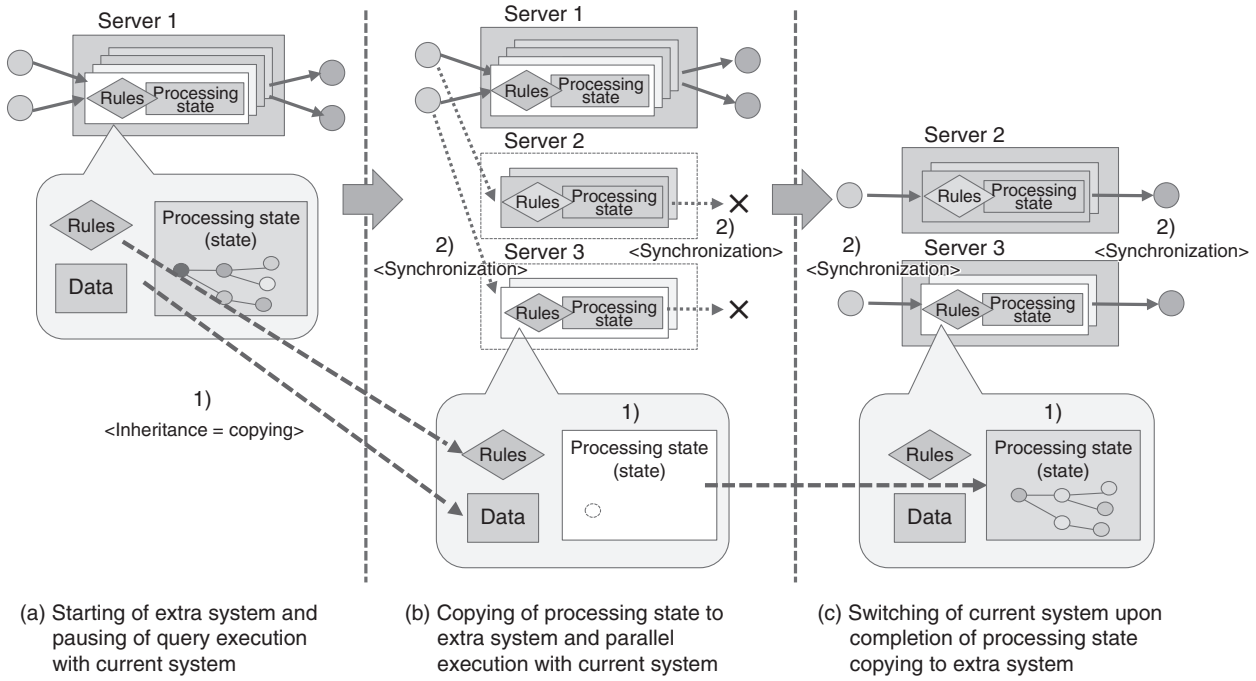

Figure 6
CEP dynamic load balancing method (transition from current system to extra system).

both the current and extra systems. In this way, the method synchronizes the processing state.

To ensure the order of event arrival, the manager that manages the execution environment sends instructions to the individual agents to synchronize the entire processing work at the start and end of configuration changes. This prevents the order of event arrival from being disturbed (**Figure 6**).

We have used this basic operation as a prototype to implement a use case in which people's location and preference information is matched with product information of stores and to issue appropriate coupons (**Figure 7**), and evaluated it. The model assumes that the number of people is up to 400 000 and the location information of each person is provided once every five seconds. In this model, as shown in **Figure 8**, the location and preference information of the individuals is extracted (Query 1). Next, information about stores within a certain distance from the individual is extracted (Query 2). The preference information of the individual and the product information of the stores extracted are matched (Query 3). And then a coupon is issued when they match.

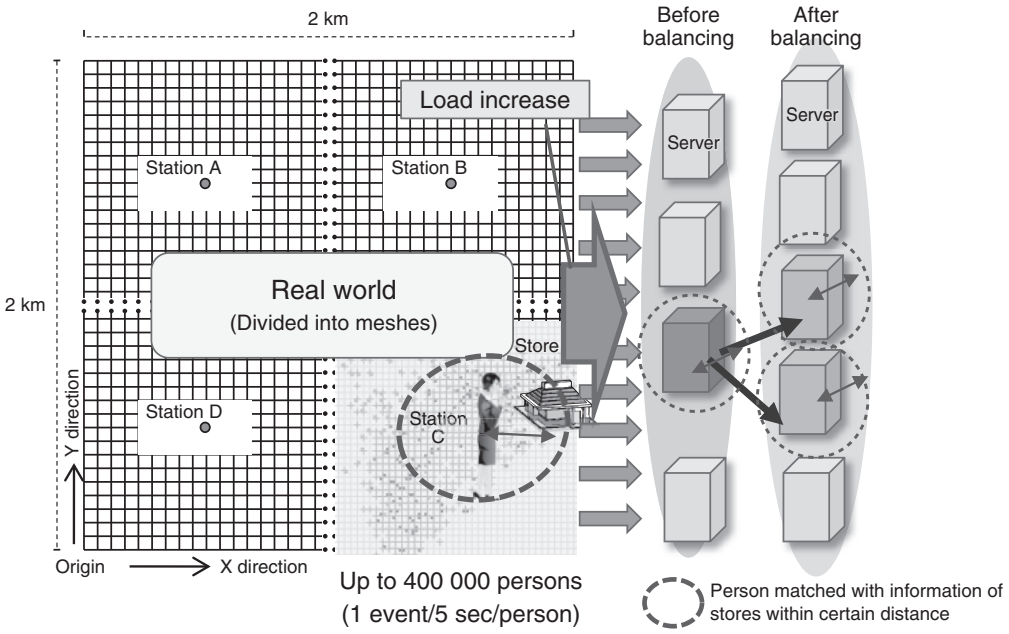

Figure 7
Evaluation system of CEP dynamic load balancing.

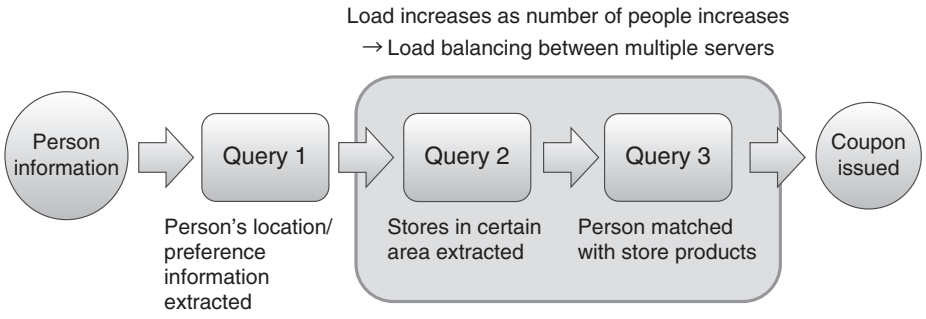

Figure 8
Actual rules (query) in evaluation system.

Queries 2 and 3 involve a significantly increased processing load as the number of people (input events) increases. However, we have confirmed that smoothly adding servers without interruption (four processing servers increased to eight, for example) reduces CPU usage and ensures high-speed processing.

This implementation has also revealed some challenges such as variations in processing due to garbage collection[note)] and event delivery synchronization. In the future, we intend to work on improving the performance and resource efficiency of the CEP dynamic load balancing function and optimization by linking with the operation function while solving these challenges.

note) One of the memory management functions, it automatically releases areas of dynamically allocated memory that are no longer needed when programs are executed.

5. Data processing workflow description language

To derive valuable information, we need a simple and productive environment as repetitive processes on the source data, such as retrieval and summary, need to be carried out.

Data-Intensive Systems Process Engineering Language, DISPEL, is a scripting language for the high-level description of data processing workflows. The language was developed in the ADMIRE project, an EU Framework 7 project that Fujitsu Laboratories of Europe joined. This project aims to significantly improve the exploitation of data by delivering an architecture and platform for data-intensive computing and demonstrate significant knowledge discovery using relevant use cases.

5.1 Language overview

Designed for data-intensive distributed systems, DISPEL uses a notation similar to Java and describes streaming data workflows.

The following example is processing on a seismic database. It transforms the source data that has been accumulated in the last 24 hours and stores them in another database. DISPEL can describe this simply by using abstract components.

```
use dispel.db.SQLQuery;
use dispel.lang.Results;
use eu.seismo.Transform;

SQLQuery sq  = new SQLQuery;
Transform tr = new Transform;
Results res  = new Results;

sq.data   => tr.input;
tr.output => res.input;
|-"uk.ac.bgs.earthquakes"-| => sq.source;
|-"SELECT FROM WHERE"-| => sq.expression;
|-"Last 24 hours"-| => res.name;

submit res;
```

The key features of the language are:

1) It is a three-level system that helps to separate concerns, e.g. developer's and data engineer's concerns, and end user's such as seismologist's application concerns.
2) It has powerful, higher-order functions to construct a complex workflow of connected processing elements.
3) It can annotate data-flow connections to support information flow, process termination, and error management.

These features accelerate the collaboration between seismologists and developers in a smooth and more accurate way.

5.2 DISPEL use cases in Europe

1) Platform for Analytical Customer Relationship Management (ACRM)

This use case applies DISPEL to analyze telecommunication data to predict cancellation. The purpose is to understand why a customer might move to another company and hence prepare an appropriate counter-offer. In this specific ACRM development, the challenge is to prepare a representative dataset and to evaluate the model. The data need to be extracted from four different databases:

- Customer DB – age, sex;
- Contract DB – number of contracts, tariff plan transitions;
- Contact DB – number of complaints or inquires registered at the call center;
- Call DB – number of outbound calls made in the evening.

In most cases, a significant part of the workflow is to prepare and clean the data, prior to applying the complex data mining algorithms.

In normal business use, the detailed data-intensive engineering process, described in DISPEL code, is hidden behind the user interface of a business portal. As an example, the data-intensive engineer can offer the business domain expert the option to select multiple models and have them vote to see what strategy is best at a

business level. The knowledge discovery process with DISPEL also expedites reuse, as multiple domains can share many engineering patterns.

The ADMIRE project has demonstrated that the approach can simplify the management of complex workflows. A single DISPEL program can capture every detail of the workflow in terms of data retrieval, data transformation, data cleaning, data federation, data mining and results delivery.

2) Data-Intensive Seismology

The increasing wealth of global seismic waveform data being captured and made available has created a great opportunity to conduct data-driven research rather than process-oriented engineering.

An archive of the Data Management Centre (DMC) of the Incorporated Research Institutions (IRIS) collates the data from several international seismological networks and is growing at approximately 21 TBs per year. This means the traditional approaches to studying waveforms recorded by the observatories is becoming increasingly challenging. The solution is to apply DISPEL to retrieve, pre-process, cross-correlate and stack the many TBs of data from two seismic data archives—the British Geological Survey and the ORFEUS Data Centre of the European Seismological Commission.

At a technical level, ambient noise data analysis of surface wave dispersion measurements have been implemented, containing four steps: 1) single-station data preparation, 2) cross-correlation and temporal stacking, 3) measurement of dispersion curves, and 4) quality control.

The DISPEL workflow document controls the entire process. The scientific results significantly increase our understanding of the earth's crust, and hence earthquakes, by analyzing the information from multiple data sources simultaneously.

6. Conclusion

Cloud is beginning to expand from application of ICT to business processes to innovation, which is intended to increase sales and optimize systems by identifying valuable information via analysis of data aggregated into clouds.

One point that makes innovation significantly different from the conventional ICT application is that innovation starts where users do not know what to do, and this becomes gradually clear while the big data collected are analyzed in various ways. For that reason, data analysis must be repeated many times from different perspectives and high-speed and low-cost processing is required in all phases of development and operation (from failure response to accumulation costs). The benefits offered by cloud such as temporary availability of massive computational resources and cost reduction by resource sharing have the potential to meet this need. If Fujitsu's cloud services already on offer are enhanced by high-speed and low-cost processing for big data processing and a simple and high-productivity development environment for data utilization, a new market for innovation should be opened up.

Fujitsu and Fujitsu Laboratories are working on distributed data store and parallelization of complex event processing, which are basic technologies of big data processing. They have developed technology for high-speed aggregation and addition of servers without interrupting services in a distributed environment. Moreover, Fujitsu Laboratories of Europe is conducting research on a description language for distributed processing of big data, which has allowed smoother data analysis and contributed to the advancement of scientific research. In the future, we plan to closely link various functions together that run in cloud environments. At the same time, we will enhance and develop these elemental technologies and practices. We aim to provide technology to allow

prompt development and execution of complex processing.

Regarding parallelization of complex event processing (CEP), we have made use of the results of commissioned operations of Industrial Technology Research and Development Commission Expense "Program to develop and demonstrate basic technology for next-generation high-reliability, energy-saving IT equipment" by Ministry of Economy, Trade and Industry, Japan.

References

1) Big data to drive a surveillance society—Computerworld.
http://www.computerworld.com/s/article/9215033/Big_data_to_drive_a_surveillance_society
2) Fujitsu succeeds in accelerating distributed data store technology for cloud services. (in Japanese), June 17th, 2010.
http://pr.fujitsu.com/jp/news/2010/06/17.html

Satoshi Tsuchiya
Fujitsu Laboratories Ltd.
Mr. Tsuchiya is currently engaged in research and development of parallel distributed processing technology in general.

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

Yuichi Tsuchimoto
Fujitsu Laboratories Ltd.
Mr. Tsuchimoto is currently engaged in research and development of parallel distributed processing technology and distributed data store.

Vivian Lee
Fujitsu Laboratories of Europe Ltd.
Ms. Lee is currently engaged in research and development of development environment for big data processing.