Dynamic Resource Controller Technology to Accelerate Processing and Utilization of IoT Data

● Makoto Kubota ● Shigeki Fukuta ● Yoshihide Nomura ● Kenichi Abiru

Recently, a concept called edge computing has been attracting attention for full-scale application of the Internet of Things (IoT) to business, and companies have started working on its use. Edge computing is intended to improve a computing system's response to terminals by distributing data processing between nodes provided near the site, rather than having all data processed in a cloud. It is hoped this will restrain the huge increase in the volume of communication traffic produced by the IoT. Fujitsu Laboratories has extended this concept and developed dynamic resource controller (DRC) technology, which utilizes computers (edge nodes) installed near the terminal on the site as if they are integrated into the cloud. This technology dynamically optimizes the locations of data processing and storage between the edge nodes and cloud according to changes in the site environment. Unlike the conventional edge computing, this not only reduces the usage of network resources but also allows for a positive increase in the usage of any surplus network bandwidth to gather raw data to the cloud that may lead to new discoveries. It can also be used to deal with addition of devices and edge node load variation by changing the location of some of the processing to edge nodes with surplus resources, which helps maintain the stability of the system. This paper describes the mechanism of the DRC technology, which allows a computing system to make a quick decision on the appropriate location of processing execution to deal with addition of devices and variation of the edge node load, and presents the results of its evaluation.

1. Introduction

Recently, information and communications technology (ICT) and cloud services have been becoming common and established as infrastructure to support human activities. In addition, the prices of sensor devices equipped with network connectivity are falling. In the future, the so-called Internet of Things (IoT), where a variety of things including vehicles, lighting appliances and all kinds of consumer electronics products in the real world (on sites) are connected with networks to grasp and analyze the site conditions for supporting human activities, is expected to become widespread. This in turn is expected to lead to a fusion between the real world and cyber world.

As the forms of utilization of ICT are changing in this way, this paper presents the dynamic resource controller (DRC) technology that we are working on for the purpose of maximizing and stabilizing systems that process and utilize data obtained through the IoT. It describes the significance and innovative nature of this DRC.

2. Changing forms of ICT utilization

In the past, ICT was mostly used for people to access data in the cloud from PCs and smartphones on an as-needed basis. However, with the IoT, which connects all kinds of things to networks, an enormous amount of things in the real world will constantly upload raw data representing the site conditions to a cloud via the networks. In the cloud, the gathered data are accumulated for processing, aggregation, and analysis. The true value of the IoT lies in utilizing the results of analysis of these data and delivering them to humans as information leading to new discoveries. Further, using the analysis results as the basis for controlling things on sites is expected to improve people's day-to-day activities and business.

The number of things connected to networks is estimated to reach 53 billion by 2020.[1] This may pose problems such as increase in management cost and network resource depletion and congestion caused by a tremendous volume and new forms of communication traffic starting from things. Other risks include lower network response due to the congestion, a degraded real-time nature of controlling devices on sites and leakage of private information because of the storage of all site condition information including personal data in the cloud.

3. Issues with edge computing

For solving the problems described in the previous section, wide-area distributed processing called edge computing has been devised. Edge computing involves data accumulation and application processing, which conventionally took place in a data center in a closed manner, performed on edge nodes, or computers installed near the terminal on the site. That is, data generated from things on the site are processed on the nearby edge nodes and the cloud is notified only of the results of processing. This reduces communication traffic and solves the problems of network resource depletion and accompanying lower response, which is the concept of edge computing, and various companies have started working on this technology.[2]

However, conventional approaches had the data processing locations fixed, and this posed the following issues.

1) Resource overflow on edge nodes due to increase in number of things

Let us assume here a factory monitoring solution in which edge nodes are installed on the factory floor and assigned with vital analysis processing, camera image analysis processing, and finished product aggregation processing, and the cloud is notified only of errors and the number of products that have undergone inspection. For example, if the number of cameras is increased in order to make the monitoring area larger or the number of workers is temporarily increased for a certain line, the volume of data gathered may increase enough to cause the analysis processing load to exceed the allowable load of the edge nodes. This could possibly result in an edge node crash that stops the system. This possibility of a system crash is not eliminated until the system administrator redesigns the system or strengthens the node resources.

2) Insufficient raw data gathering to the cloud

In the system described above, what can be grasped in the cloud only includes obvious abnormal vital signals of workers and their standing positions at the time, and the progress of inspection. That is, raw data are discarded on the edge nodes and, while the communication traffic is successfully reduced, the raw data are not gathered to the cloud. If raw data such as vital information including values smaller than the abnormality thresholds and work video information are available, detailed analysis may make it possible to identify the cause-and-effect relationship between the frequencies of yawning of workers and the probability of their inspection errors, for example. And this suggests the possibility of novel discoveries such as implementation of new yield improvement measures.

3) System redesign cost according to changes in site environment

Which edge nodes or cloud to use for analyzing and aggregating data gathered from on-site things is designed by systems engineers (SEs) in view of the load on the edge nodes and cloud in the system and the balance between the load on the networks and cost available for infrastructure strengthening. However, if changes in site conditions as described in 1) of this section occur frequently, redesigning the system each time becomes a heavy burden on the SEs. In particular, a huge system with a large number of things and edge nodes may require person-hours in units of a month for one case and it is not practical for SEs to accommodate minor changes such as a temporary increase of workers on the site.

4. DRC technology

This section describes DRC technology, which is being researched and developed by Fujitsu Laboratories.

DRC is an extension of the concept of edge computing and it not only transfers the locations of data processing and accumulation to edge nodes but also fuses edge nodes to a cloud. That is, it is an IoT-enabled cloud technology featuring dynamic optimization of the locations of data processing and accumulation between the edge nodes and the cloud according to changes in side conditions. Edge nodes provided in the vicinity of terminals and devices or on the edge of a wide area network are seen as an extended cloud environment across a wide area (**Figure 1**).

This allows the system to be kept stable by changing the execution locations of data processing and accumulation when loads on edge nodes or networks vary due to an increase or decrease or movement of things on the site, in addition to simply solving the problem of the huge volume of communication traffic starting from things [**1)** and **1)'** in **Figure 2**]. In addition, if any surplus bandwidth of the network resource is found, the execution locations of some of the on-site data processing can be transferred from edge nodes to the cloud to gather raw data leading to new discoveries to the cloud [**2)** and **2)'** in **Figure 2**]. In this way, DRC automatically optimizes the processing locations when edge node or network load variation or surplus resources are generated, and this releases SEs from redesign work [**3)** and **3)'** in **Figure 2**].

As an application of this technology, content data required by individuals can be deployed on edge nodes in advance by making use of the location information of the respective individuals or surplus network resources. This allows the quality of experience (response) in individuals' access to the content to be improved.[3)]

Figure 3 shows the basic architecture of DRC. It is composed of three layers: the application layer, the real infrastructure management layer, and the DRC control layer located between them. The following outlines the operation that realizes the behavior described in the previous section.

The DRC control layer first acquires from the application layer information about the application to be decentralized. Then, it acquires from the real infrastructure management layer the topology of the things and edge and cloud nodes, the states of the node and network resources, and information about the volume of communication that is actually flowing. By using all this information, the nodes to be used for executing the application are determined (the method of determination will be explained later). This determination is repeated every time any change is generated in the application or infrastructure.

In this way, the application information, infrastructure resource states and their changes are observed in the DRC control layer, which makes it possible to follow the changes, and change the processing locations.

The following subsections describe the roles and functions of these three layers and the input information required for them.

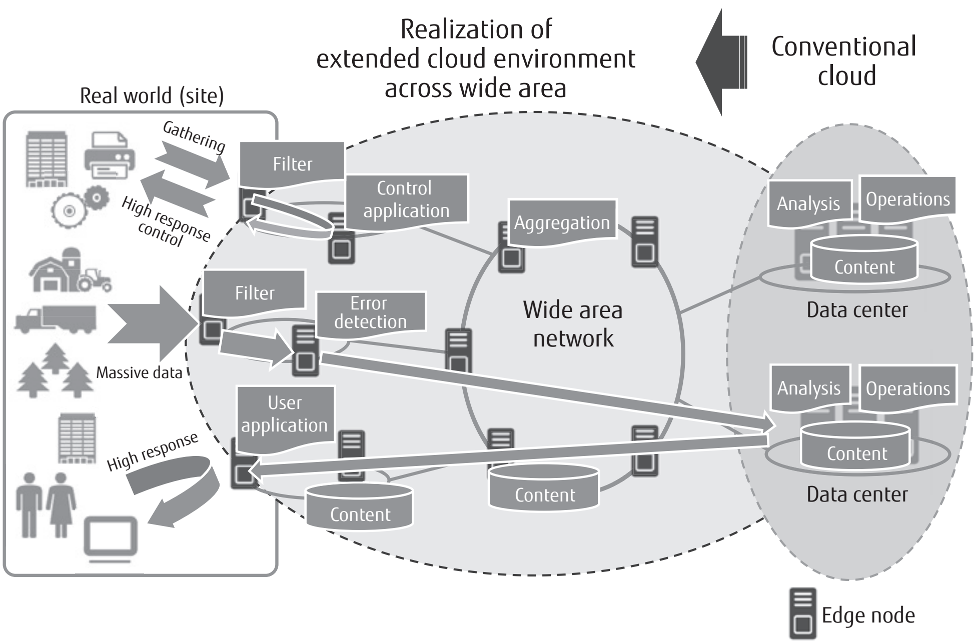


Figure 1
Concept of DRC.

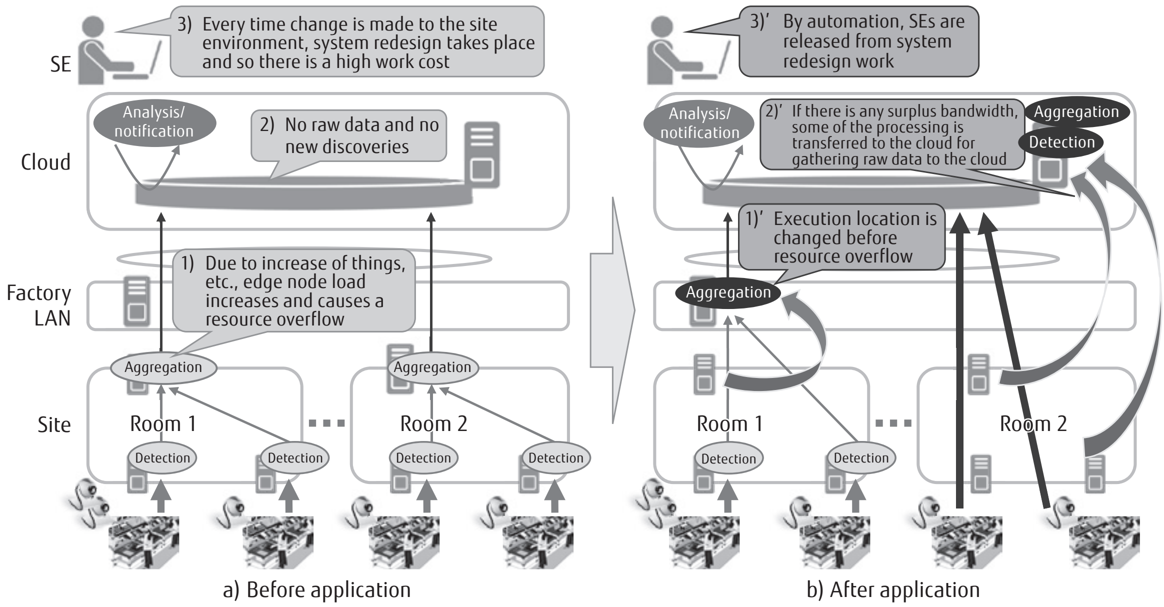


Figure 2
DRC application example.

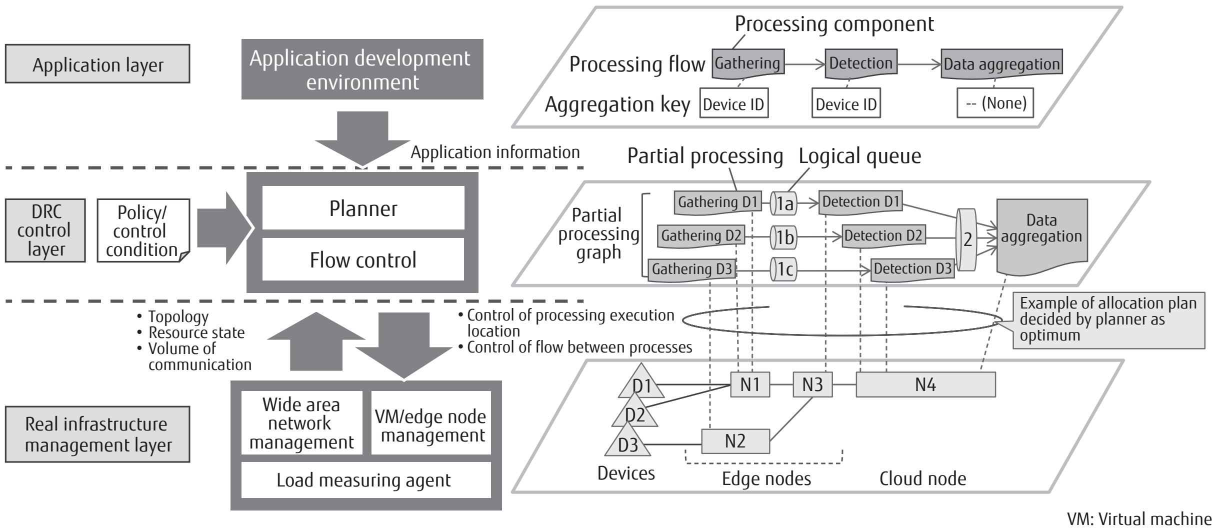


Figure 3
Basic architecture of DRC.

4.1 Application layer

In this layer, the application to be decentralized is developed and managed.

When the DRC control mechanism, which will be explained later, decentralizes an application, if the execution locations can only be changed for the entire application, the resource utilization efficiency may be reduced for reasons such as the tendency of the load to concentrate on specific nodes. To deal with this problem, DRC is provided with the capability of describing the processing flow of the application as a combination of processing components to determine the execution nodes for the respective processing components, and this is shown in Figure 3.

Furthermore, if processing, which may be one process logically, is physically assigned different processing execution edge nodes for the respective targets of processing such as the devices for which data are gathered, the resource of each edge node can be maximized, leading to improved resource efficiency of the entire system. For that purpose, DRC can specify keys (aggregation keys in Figure 3) to determine the division units of the respective processing components. This allows one processing component to be divided into "partial processes" in units of one of these aggregation keys so that execution nodes can be allocated to the respective partial processes. In Figure 3, for example, the aggregation key of the gathering processing is the device ID and one processing component is divided into three partial processes corresponding to devices D1, D2, and D3, to which execution nodes are allocated respectively.

The application development procedure is explained by using the prototype screen developed by Fujitsu Laboratories.[4] The application developer first drags and drops processing components to the processing flow description area shown in **Figure 4** and connects between the components with lines to describe the application logic in a flow format. The allocation of the execution nodes for the respective processing components is determined by the DRC control mechanism, which will be explained later, and does not need to be considered at the time of development. Then, in the property setting area, the developer specifies the aggregation keys for the processing components and the values of various parameters. Lastly, the developer uses the application registration button to register the processing flow information and aggregation key information for the developed application with the DRC control mechanism.

4.2 DRC control mechanism

The DRC control mechanism is at the core of DRC and composed of two technologies: planning technology, which optimizes the execution locations for partial processes of the application according to the load variation and surplus resources of edge nodes and networks, and flow control technology, which connects things in the real infrastructure with the partial processes accordingly. The following describes the respective technologies.

1) Location planning technology of processing execution

This technology optimizes allocation of execution nodes to the respective partial processes to maximize the effect, which will be explained later. The input information required for allocation includes:

- Policies and restrictive constraints indicating the

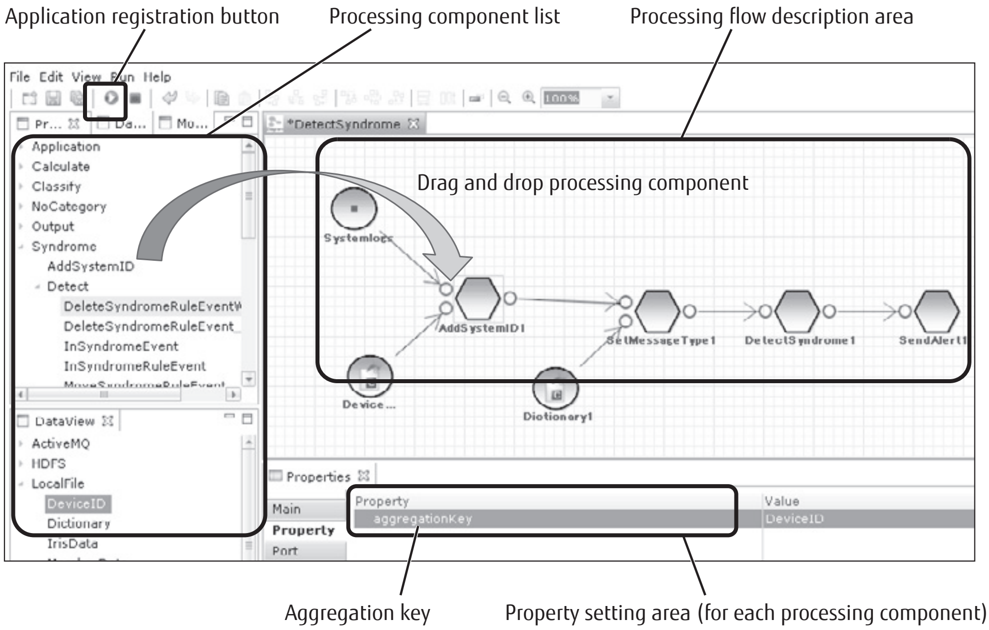


Figure 4
Screen for processing flow creation.

effect to be obtained by decentralization,
- Processing flow information for the application to be decentralized,
- Information about the real infrastructure to be used as the destination of decentralization (details will be explained later), and
- Operation logs for the respective applications after decentralization.

Determining the optimum allocation by using these types of information is referred to as planning. The functional entity responsible for this planning is the planner in Figure 3. The following describes the method of planning using the input information mentioned above.

First, the planner reads the processing flow information defined in the application layer as information about the target of decentralization and the aggregation key information for the respective processes. Then, it determines the indicator of what is to be maximized (effect) in decentralization. This effect depends on the operation policy of the application and cannot be determined by the DRC. Accordingly, the planner allows the decentralization policies indicating the desired effect to be input externally. The following lists examples of policies that can be set.
- Make the most of the network bandwidth up to the allowable value.
- Minimize the network bandwidth.
- Transfer processing to the cloud if the edge node load is equal to or higher than a certain level.
- Maximize the response from things and terminals.

In some cases, the operator may want to determine the execution nodes for certain partial processes (for example, e-mail notification processing should always be allocated to the cloud for the purpose of leaving an audit trail). Accordingly, the system is made to also allow external input of the constraints relating to the execution nodes for the respective partial processes.

In addition, the processing flow information for the target of decentralization, aggregation key information for the processing and the device list information for the real infrastructure are used for dividing the unit of node allocation into partial processes of the respective devices (partial processing graph in Figure 3).

Then, the information gathered from the real infrastructure (specifically, the device list information for the real infrastructure, the topology of the nodes, node states such as CPU load and information about the state of link between nodes such as the bandwidth upper limit and surplus bandwidth) is used as the input. This input is utilized to create combination patterns that are possible as execution nodes for partial processes, and the pattern producing the highest effect of decentralization is identified as the optimum plan. For example, if the policy is "Make the most of the network bandwidth up to the allowable value," the combination pattern providing the maximum volume of traffic that is equal to or lower than the allowable value is identified as the optimum plan. Then, control such as deployment of the respective partial processes is provided and the partial processes are executed in the respective nodes.[5)]

The planning procedure described above is not finished in one go but is repeated every time any change is made to the input. For example, if the policy is "Make the most of the network bandwidth up to the allowable value" and the surplus network bandwidth has decreased, replanning takes place and, of the partial processes in Figure 3, only the execution location for detection D2 is changed from the cloud node N4 to the edge node N3. This allows the optimum processing allocation to be maintained that satisfies the policy even if the real infrastructure conditions change.

Usage of the CPU resource and bandwidth resource for the respective partial processes depends on the application or site environment and cannot be figured out until actual operation. Accordingly, the planner also uses as input the resource usage information obtained by referring to the operation log for the respective partial processes after decentralization, thereby making it possible to improve the accuracy of the plan made by the second and subsequent runs of the procedure.

2) Flow control technology

The flow control technology is responsible for routing control for transferring and delivering data with the aggregation key between partial processes according to the node allocation plan for the respective partial processes determined by the planner. The following describes the method of flow control.

The flow control technology first identifies from the partial processing graph in Figure 3 a logical queue that connects between partial processes. Then, it creates this logical queue in the nodes to which the corresponding partial processes are allocated. To take the logical queue 1a in Figure 3 as an example, it sets routing table rules that transfer the data from "Gathering D1, Detecting D1" in the nodes N1 and N3, to which corresponding partial processes "Gathering D1, Detecting D1" are allocated.

If the planner changes any execution node for a partial process, only the difference in the routing table is given out to the nodes to follow the determination made by the planner, and this improves the efficiency of control at the time of node change.

4.3 Mechanism for managing real infrastructure

This mechanism manages the configuration and capacity of the nodes to be used as decentralized execution locations for partial processes and the input information for calculating the allocation of partial processes. For example, it manages the following types of information.

- Topology (network configuration of devices, edge nodes and cloud node)
- Node capacity and load (CPU, memory, etc.)
- Link capacity and load (bandwidth, delay, etc.)
- List of devices for which data are gathered
- Operational statuses of the respective partial processes after decentralization (volume of communication, etc.)

The means of information management are separated from the DRC technology and information is acquired via application programming interfaces (APIs). Examples of the means assumed include:

1) As the management entity for the capacity and load of the topology and link mentioned above, the management system for the wide area network including edge nodes and
2) As the management entity for the capacity and load of nodes, the cloud virtual machine (VM) management mechanism and edge node management mechanism that manage information gathered from load-measuring agents in the edge nodes and cloud node.

5. Planning acceleration technology

For the planner described above, we have developed a technology that can reallocate processing execution nodes in a short time to follow any change detected in the site conditions even if the scale of the infrastructure is increased. The following describes the features of this technology.

The combinatorial optimization problem of allocating execution nodes to partial processes described in the previous section is an integer programming problem. It is generally characterized by a computational complexity that increases exponentially as the scale of the problem increases. For this reason, the calculation time of the execution node allocation may become enormous along with the increase in the numbers of partial processes and nodes. As an example, let us assume decentralization of an application composed of 3,000 partial processes into 1,000 nodes for execution by using a conventional combinatorial optimization calculation method. A planning calculation for adding one device based on this assumption took more than 35 hours on a server with an eight-core CPU. This is not practical as the time required for following a change that occurs on site.

Accordingly, we focused on the characteristics of applications for the IoT that gather data from the site for processing: there are many processes in which multiple pieces of site data are gathered and the volume of output data is lower than that of the input data. That is, we incorporated schemes according to these characteristics into a search algorithm to search for combinations of nodes that provide a high effect of decentralization and it will be described later. In this way, we have made it possible to search for a quasi-optimal solution (solution that is not optimum but close to it) by narrowing the candidates of execution nodes down to those with a high effect.

The nature of the schemes available depends on the effect required of decentralization (speed-up, cost reduction, etc.). The following uses **Figure 5** to describe a case in which the required effect is minimization of the network bandwidth.[6),7)]

1) Refining search according to network path

Using the shortest path from the site to the cloud for processing eliminates detouring of data and provides high utilization efficiency of the network resources. In particular, for partial processes that aggregate data from multiple sites as input, the node in the topology where the shortest paths from the respective sites meet can be used for processing execution.

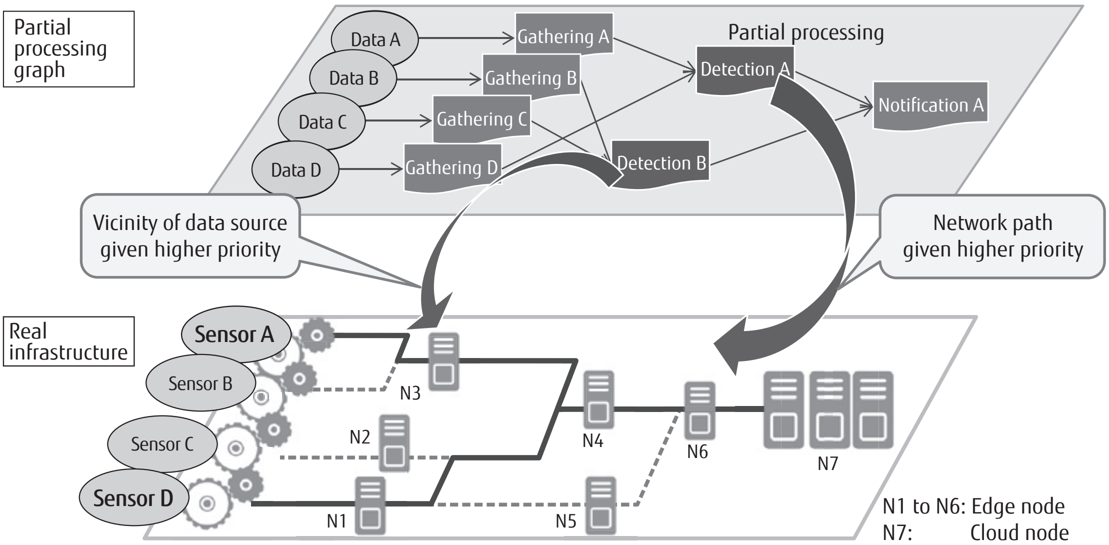


Figure 5
Refining search for partial process execution locations.

If this is possible, detouring for aggregation is eliminated, and this also improves the utilization efficiency of the network resources.

For that purpose, the starting locations for node searches for the respective partial processes are narrowed down to points on the shortest paths from the sites to the cloud (broken lines in the figure) and searches are performed in order starting from the meeting points (N6 and N4 in the figure) of the paths with the cloud as the destination. This allows searches to start from nodes with a higher effect, leading to shorter search times.

2) Refining search according to data input-output volume ratio

To partial processes with a small ratio of the output data volume to the input data volume, nodes as close as possible to things or processes with large output data volumes can be allocated. In this way, even if data of smaller volumes are detoured, the overall traffic can be reduced as a result.

Accordingly, for partial processes with the ratio of the output data volume equal to or smaller than a certain level, searching starting from nodes closer to the data source (N1, N2, and N3 in the figure) in order allows searching for nodes that offer a higher effect, which reduces the search time.

Either of these two search logics can be used depending on the input-output volume ratio for the respective partial processes.

6. Evaluation of change-following performance

We evaluated the technique described in the previous section by using simulation. The following presents the results of evaluating the initial calculation time of the planner after application registration and the recalculation time of the planner as the site conditions change after decentralization is completed.

1) Evaluation 1: time required for the initial plan calculation

We evaluated the calculation time required for initial planning relating to the respective partial processes after the registration of the application with the DRC control mechanism.

The system configuration assumed is as shown in Figure 5. The number of partial processes of the application is between 30 and 3,000 and the number of edge nodes is 1,000. The decentralization policy is “Minimize the network bandwidth” and restrictive constraints are given that keep the load on each processing node equal to or lower than a certain level.

With the conditions above, two methods of optimum node searching for the respective partial processes were used for comparative evaluation in terms of the calculation time and the rate of reduction of the network bandwidth used. The two methods are A) the conventional method, or randomly determining the search starting node and searching for the optimum node, and B) searching by means of the schemes described in the previous subsection. The results are shown in **Figure 6**. The rate of reduction of the bandwidth refers to the reduction rate with reference to the network bandwidth used for executing all partial processes in the cloud.

As shown in Figure 6 a), it has been confirmed that the calculation time for 3,000 partial processes is over 35 hours when using the conventional method A) but only 65 seconds with the proposed method B), or approximately 1/2,000. Meanwhile, the traffic reduction effect is almost the same between A) and B) as shown in Figure 6 b), which indicates that the proposed method speeds up the calculation without affecting the rate of reduction of the network bandwidth.

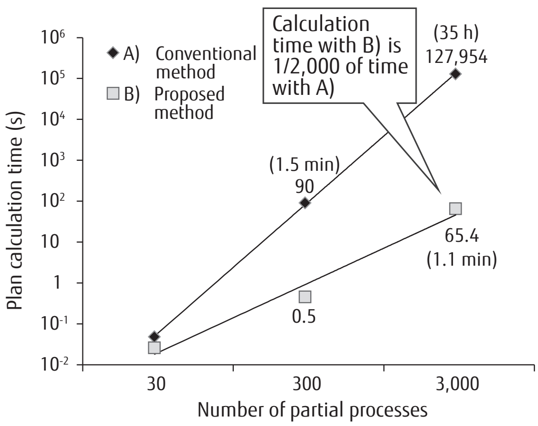
2) Evaluation 2: recalculation time required for accommodating additional sensors

When sensors are added to a system with partial processes already decentralized for execution, the partial processes for the additional sensors are required in addition to the partial processes already being executed. We evaluated the calculation time required for this planning.

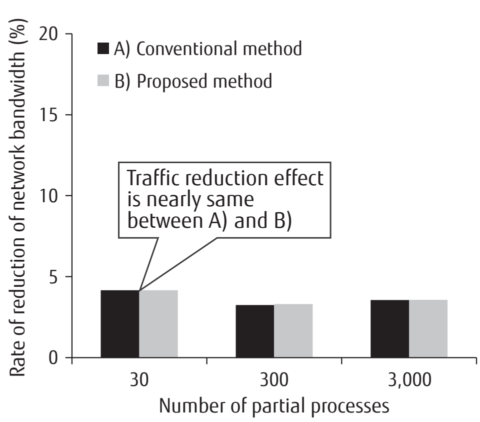
The system configuration assumed is that shown in Figure 5, as with 1). As the conditions, 100,000 to 1 million sensors in operation are assumed in the entire system, to which the number of sensors corresponding to 0.2% of the original number are added at some point.

With these conditions, the calculation time required by the proposed method was evaluated, and the results are shown in **Figure 7**. When the proposed method B) was used, in a system containing 1 million sensors, recalculation to accommodate the 0.2% addition of sensors took 41 seconds to complete. The number of sensors and the calculation time for replanning are in a linear relationship and it has been found that changes can be accommodated in a short time even with large-scale systems. The calculation time required with the conventional method A) results in an exponential graph with the calculation taking more than one day even with a system containing 100,000 sensors and the values on a totally different order of magnitude, and this is not shown in the figure.

As a result of evaluation by using simulation as described above, the proposed method has proven to be capable of significantly reducing the calculation time as compared with the conventional simple search method. It has also been confirmed that, with a system of about 3,000 partial processes and 1,000 edge nodes, the initial calculation can be completed in a time in the order of a few minutes. Furthermore, in a running system that has 1 million sensors, for example, the calculation for reoptimization has been confirmed to only take one minute to complete, which means that, with a system that undergoes a gradual change, the change can be followed with a practical calculation time.



a) Initial plan calculation time



b) Rate of reduction of network bandwidth used

Figure 6
Results of evaluating initial plan's calculation time.

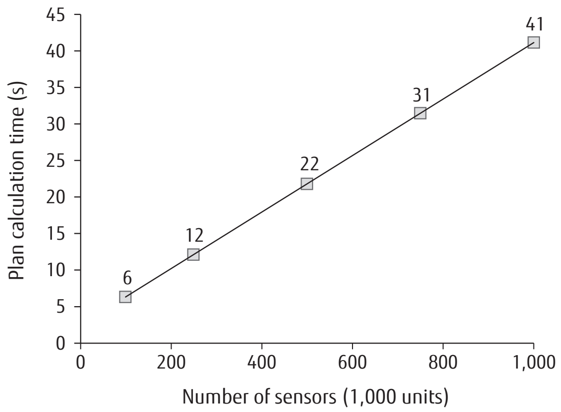


Figure 7
Results of evaluating calculation time of proposed method.

7. Conclusion

For full-scale application of the IoT to business in the future, this paper has presented dynamic resource controller technology featuring dynamic optimization of data processing and storage locations between the edge nodes and the cloud according to changes in the site conditions. This technology allows the stability of a system to be maintained in the event of load variation in edge nodes and networks due to an increase or decrease or movement of things by means of changing the execution locations for data processing and storage. It has also been shown that raw data leading to new discoveries can be gathered to the cloud by transferring the execution locations for some of the on-site data processing from edge nodes to the cloud in order to make use of the surplus bandwidth of the network resources.

In addition, this paper has presented the mechanism of planning, or calculation of the optimum processing locations, which characterizes this technology, and shown based on simulation evaluation that it is possible to follow changes in a short time even with large-scale systems. Part of this technology has already been put to practical use in FUJITSU Cloud Service IoT Platform.[8)]

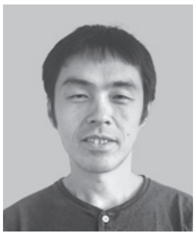
From now on, for expanding its uses, we intend to strengthen the technology so that planner and flow control can be used to accommodate a mobile environment that involves frequent changes of the connection relationships between things and edge nodes and the topology between nodes so that smart devices can also be used as edge nodes.

References

1) Ministry of Internal Affairs and Communications: White paper 2015 Information and Communications in Japan. *http://www.soumu.go.jp/johotsusintokei/whitepaper/eng/WP2015/2015-index.html*
2) The Overwhelming Power of Edge Computing, Nikkei Electronics, November 2015 Issue (in Japanese).
3) K. Abiru et al.: Improving Quality of Experience for Users through Distributed Service Platform Technology. FUJITSU Sci. Tech. J., Vol. 52, No. 2, pp. 56–63 (2016). *http://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol52-2/paper06.pdf*
4) Y. Nomura et al.: Massive Event Data Analysis and Processing Service Development Environment Using DFD. In Services (SERVICES), 2012 IEEE Eighth World Congress on Services, pp. 80–87, June 2012.
5) S. Fukuta et al.: Distribution of Event Processing Function to the Sensor Network. BS-4. Optimum, Autonomous and Power-saving Control Technologies Supporting Networks, Symposium Session. Proceedings of the Society Conference of IEICE, 2011 (2).
6) Fujitsu Laboratories: Fujitsu Develops Distributed Service Platform Technology to Optimize Cloud Environments over a Wide-Area Network. *http://www.fujitsu.com/global/about/resources/news/press-releases/2014/0314-01.html*
7) Proposal of the Scalable Method of Following to Environmental Changes for Automatic Wide Area Process Distribution. Multimedia, Distributed, Cooperative and Mobile (DICOMO2014) Symposium, July 2014. IPSJ Transactions on Consumer Device & System, Vol. 5, No. 4, pp. 31–41, October 2015 (in Japanese).
8) Fujitsu: Fujitsu Releases FUJITSU Cloud IoT Platform, a Platform Service for IoT Data. *http://www.fujitsu.com/global/about/resources/news/press-releases/2015/0610-01.html*

Makoto Kubota
Fujitsu Laboratories Ltd.
Mr. Kubota is currently engaged in research and development of dynamic resource controller technology.

Shigeki Fukuta
Fujitsu Laboratories Ltd.
Mr. Fukuta is currently engaged in research and development of dynamic resource controller technology.



Yoshihide Nomura
Fujitsu Laboratories Ltd.
Mr. Nomura is currently engaged in research on integrated development and execution environment for big data and IoT.

Kenichi Abiru
Fujitsu Laboratories Ltd.
Mr. Abiru is currently engaged in research and development of dynamic resource controller technology.