Large-Scale Real-Time Processing Technology for M2M Service Platform

ISOYAMA Kazuhiko, SATO Tadashi, KIDA Koji, YOSHIDA Makiko

Abstract

Recent progress in networks and devices has promoted a great diversity of M2M services and the number of devices connected to the M2M service platforms is expected to continue to grow. This paper introduces a complex event processing (CEP) technology that detects significant data changes in the M2M services in real time via data collected from a large number of devices.

Keywords

M2M, complex event processing, CEP, scalability, load distribution

1. Introduction

Sensors for collecting diverse information such as power consumption data, traffic information and remotely controllable devices (actuators) such as network cameras and smart home appliances are recently being connected to the networks in growing numbers. Because of this trend and due to its capability of controlling devices based on sensor information not requiring human intervention, the M2M (Machine-to-Machine) technology is attracting attention.

In the future, it is estimated that devices connected to networks will increase significantly and information generated from them will reach enormous amounts, thereby causing the M2M services to grow in scale. To deal with such growth, a platform that can deal with it effectively by increasing the scale of services is regarded as being essential. NEC has developed CONNEXTIVE M2M platform that enables large-scale M2M services by handling a huge amount of information from devices. This paper introduces the large-scale, real-time technology that executes the information processing as a part of this platform.

2. Context and Event

In this paper, we use word "context" in a narrow sense to refer to "status of people and things." Such a context can be estimated from information generated from devices including sensors. For example, information from GPS (Global Positioning System) sensors can be used for estimating the locations of people. A service which provides optimum service based on context is called a "context-aware service."

Information generated from such sensors is called "events," and the process of estimating contexts or procedures based on events is called "event processing." In particular, the technology, which estimates a context that cannot be estimated from individual simple events and thereby executes appropriate processing based on combination of events, is referred to as "complex event processing (CEP)."

3. Required Processing Capabilities

Table shows the estimation of the amount of events which is required for the M2M platform to process in order to realize each service. It also shows that, in order to provide a nation wide M2M service utilizing mobile terminals and vehicles, the platform must have a processing capability of more than a million events/sec.

For example, for a real-time location information service accommodating mobile phone users all over Japan and assuming that 50 million mobile phone users send location information every minute, the total event traffic would be 830,000 events/sec.

In order to enable a "small start" that will permit a future scale expansion of services, the platform is required to adopt a scalable architecture.

The next section introduces SCTXPF (Scalable Context Platform), which is a platform architecture developed by NEC and capable of executing event processing at more than one million events/sec.

Table Processing capabilities required for M2M services.

Services	Necessary events		Response time required by users
	Number of events, explanation	Calculation logic	
Mobile phone location information	• 830,000 events/sec.: A large capacity is required for the 830,000 events	• 50 million people (half the mobile phone users)/60 sec. (one transmission per min.) = 830,000 events/sec.	• Approx. 1 sec.: A high response is required for services utilizing location information.
Probe car	• 1,330,000 events/sec.: A large capacity is required for more than one million events	• 80 million vehicles (Automobile Inspection & Registration Information Association data of 2008)/60 sec. (one collection per min.) = 1,330,000 events/sec.	• Approx. 10 sec.: Based on estimation that the response required for providing traffic jam info may be about 10 sec.
Home appliance eco-monitoring	• 130,000 events/sec.: The capacity may become large according to applications because the frequency of information co llection by the system is determined according to the control timings.	• 400,000 high-end houses (estimated as equal to the number of households subscribed in security services) × Number of home appliances (estimated as 20 units/house) × Collection frequency (every minute) = 130,000 events/sec.	• Approx. 1 sec.: High response time is required if home appliances are to be controlled.
Agriculture	• 200,000 events/sec.: A sufficiently large capacity may be required because farming field data (temperature, humidity, wind flow, wind shift, sunshine, etc.) has to be collected from a broad area.	• 1.2 million farmhouses (assuming that 40% of the 3 million Japanese farmhouses introduce field servers) × Number of sensors (estimated as 10 units/house) × Collection frequency (every minute) = 200,000 events/sec.	• Approx. 1 min.: It is estimated that the response time should be about 1 minute if advanced IT agriculture is practiced.

4. SCTXPF

4.1 Publish/Subscribe Architecture

The SCTXPF adopts Publish/Subscribe (Pub/Sub) architecture. With this architecture, an application requesting event processing to the platform subscribes details of the event processing to the platform as event processing rules. When a device generates (publishes) an event matching the event-processing rules, the platform in which the event-processing rules are subscribed processes the event and notifies the requesting application of the processing results.

Pub/Sub architecture makes the interfaces between the platform, applications and devices asynchronous, thereby enabling them to function independently. It permits subscription or unsubscription of applications and devices in/from the platform services, which means that a service platform is implemented in/from which a large number of applications and devices can subscribe and unsubscribe freely.

4.2 Concept of Architecture for Scalable Event Processing

The CEP operations can be divided into the part composed of simple “stateless” processing and that composed of complex “stateful” processing.

For example, let us assume CEP operations to “issue notification when events A and B are transmitted within an interval of less than a minute.” In this case, processing for extracting events A or B from a large number of events can be run only by monitoring the events being received at every moment. This processing therefore belongs to stateless processing. On the other hand, in order to discover that events A and B are transmitted within an interval of less than a minute, it is necessary to save events A and B occurring in preceding minute (this is referred to as a state) in memory. This processing therefore belongs to stateful processing.

In general, the stateless processing is capable of high-speed operation because it monitors only the received events. In contrast, the stateful processing needs to save the states and inquire past states, so its processing is slower than for the stateless

processing.

Considering the above conditions, the SCTXPF divides a CEP processing into a part of stateless processing and that of stateful processing, and allocates the optimum amount of CPU resources to each part to enable large-capacity CEP processing operations.

4.3 Architecture

Fig. 1 illustrates the architecture of the SCTXPF [1]. The SCTXPF is composed of event processors (EPs), an EP controller (EP-CTL) and event dispatchers (or simply, dispatchers).

The stateless processing is carried out by the dispatchers and the stateful processing is carried out by the EPs. The EP-CTL divides an operation of CEP into stateless and stateful processing operations. The detailed functions of each component are as described below.

(1) EP-CTL

The EP-CTL allocates CEP rules received from applications to the EPs. In addition, the EP-CTL generates event dispatch rules so that events that are subjects of processing of the CEP rules are transferred to the EPs to which the CEP rules are set, and the event dispatch rule is set to a dispatcher.

(2) EPs

Each EP processes events according to the CEP rules set by the EP-CTL. The CEP rule set to the EP is identical to the CEP rule received from the application, but the EP is required to execute only stateful processing because the stateless processing is executed by the dispatchers as described below.

To implement CEP, every EP is equipped with a rule engine that searches the CEP rules matching every received event and a state manager that retains the states.

(3) Dispatchers

Each dispatcher filters and dispatches events according to the dispatch rules set for it by the EP-CTL. Dispatching refers to distributing each of the filtered events to the EPs to which the CEP rules for processing the event are set.

In order to execute filtering and dispatching of events, every dispatcher has a rule engine that searches the dispatch rules matching every received event. The dispatchers are not equipped with a state manager because, unlike the EPs, they do not execute stateful processing.

As the SCTXPF does not share states between multiple EPs and multiple dispatchers as shown in Fig. 1, the processing capability can be expanded easily by increasing or decreasing the number of servers without resulting in reciprocal influence. This allows the SCTXPF to be regarded as a scalable architecture as discussed in section 3.

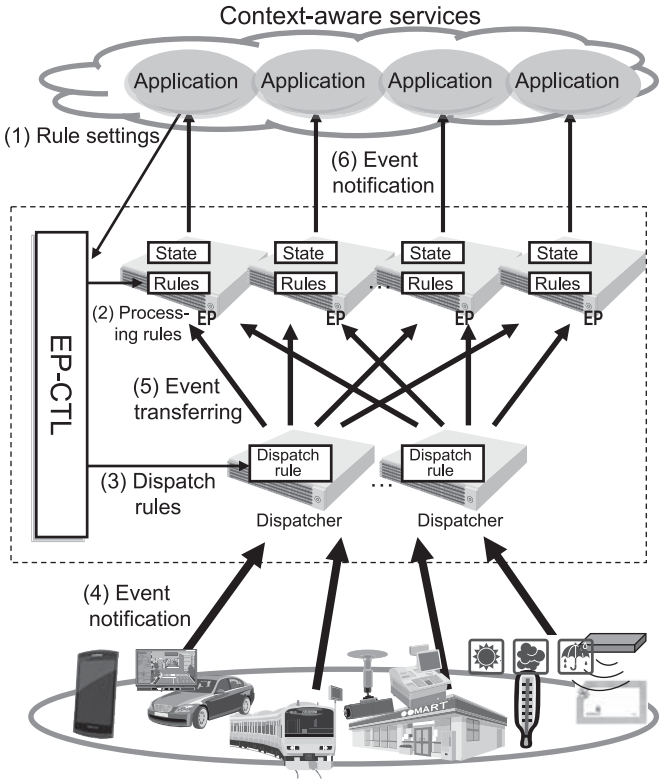

Fig. 1 SCTXPF architecture.

4.4 Event Processing

The representative event processing executed by the event processors (EPs) of the SCTXPF includes the following operations.

(1) Sequential pattern detection

This function detects generation of multiple events specified in a CEP rule in the specified sequence within a certain interval. It can be used to detect passages by people and vehicles on routes or abnormal power consumption patterns of home electric appliances.

(2) AND pattern detection

This function detects generation of all of the multiple events specified in a CEP rule within a certain interval (not in a specified sequence as in sequential pattern detection).

(3) OR pattern detection

This function detects generation of some of the multiple events specified in a CEP rule.

5. Real–Time Information Provision Service for ITS Car Navigation Systems

NEC has developed a demonstration system that provides real-time information for an ITS (Intelligent Transport System) car navigation system by applying the event processing system introduced in this paper.

This demo system emulates a service that distributes real-time information to vehicles being driven all over Japan. The distributed information includes that in the neighborhood of the current location of each vehicle and electronic coupon information according to history of the electronic coupon usage of each user.

Fig. 2 shows the configuration of the demo system. Event load generators on the extreme left generate events simulating events transmitted by the vehicles. The event types include the vehicle location information, remaining battery information (assuming EVs) and coupon usage information. Events are transmitted by setting the event generation rate, assuming that the total number of vehicles running the system all over Japan is 50 million and that every vehicle transmits an event per minute.

The CEP system introduced in this paper corresponds to the area enclosed by the broken-line rectangle in Fig. 2. The CEP detects the traveling direction of each vehicle from its location information events using the sequential pattern detection

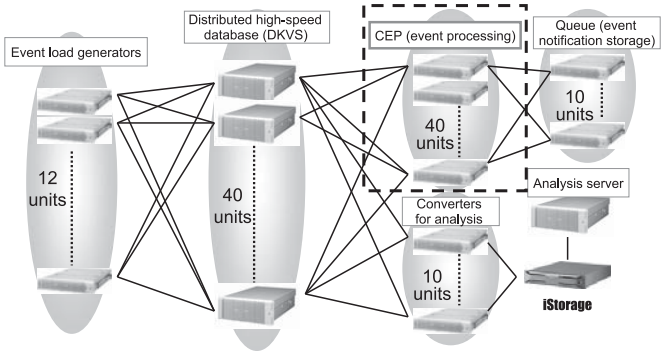

Fig. 2 Demonstration system configuration.

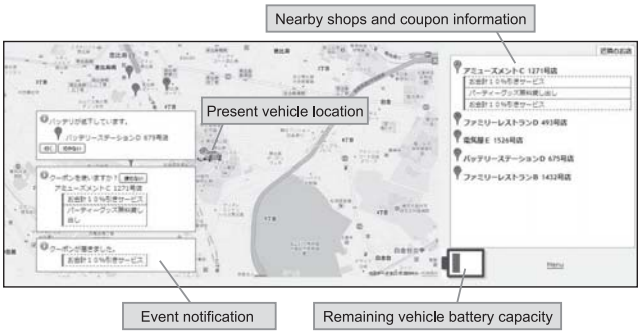

Fig. 3 Demonstration display of ITS car navigation.

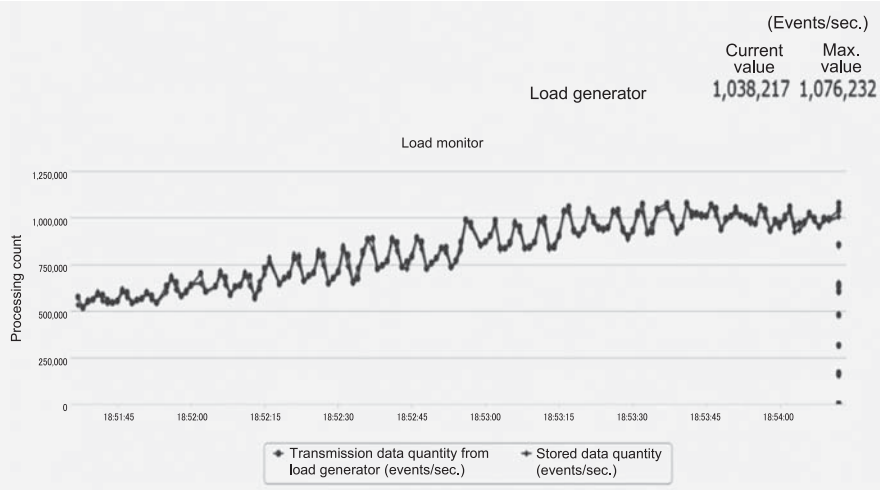

Fig. 4 Graph of load processing amount in demonstration.

function, and notifies the vehicle of shop location information in the traveling direction. It also notifies nearby recharging stations based on the remaining battery capacity of each vehicle (**Fig. 3**).

Fig. 4 is a graph of the event processing capability of the demo system. The graph shows that the system achieves a processing capability of one million events per sec., which means that the required processing capabilities shown in Table can be satisfied. As this system is scalable, the event processing capability can be increased further by increasing the number of servers.

6. Conclusion

This paper described an event processing platform technology which realizes real-time, scalable and large volume M2M services. We believe that the adoption of such a technology will make it possible to provide an event-processing platform compatible with the M2M services, which are expected to increase both in number and scale.

Reference

1) K. Isoyama, et al. “SCTXPF: Scalable Context Delivery Platform,” International Conference on Communications (ICC) 2011 Workshop on Embedding the Real World into the Future Internet, 2011

Authors' Profiles

ISOYAMA Kazuhiko
Assistant Manager
Service Platforms Research Laboratories

SATO Tadashi
Assistant Manager
Service Platforms Research Laboratories

KIDA Koji
Principal Researcher
Service Platforms Research Laboratories

YOSHIDA Makiko
Senior Manager
Service Platforms Research Laboratories

Information about the NEC Technical Journal

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

Link to NEC Technical Journal website

Japanese

English

Vol.6 No.4 “Network of Things”

Vol.6 No.4
December, 2011

Special Issue TOP