Service Execution Platform "WebOTX" To Support Cloud Computing

KATOU Masayuki

Abstract

The trend toward reductions in IT investments due to the current economic climate has tended to focus our attention on cloud computing because it does not require initial capital investment. Nevertheless, even in the cloud environment, the needs of business system infrastructures are essentially the same. The WebOTX operational history extends over more than a decade as a service execution platform that can effectively execute business systems. This paper describes its main features of high reliability and operability, together with a description of the function enhancements that are featured in the latest version.

1. Introduction

Business systems have previously been built individually inside enterprises but, at the present time an increasing number of business systems are being created as a result of the increasing trend in the spread of cloud computing. This may be seen with the cases in which enterprises build their business systems based on data centers that are external to the enterprise by procuring the required amount of resources externally for a required period without needing to be the owners of dedicated IT resources such as (PaaS and IaaS). Trends are also
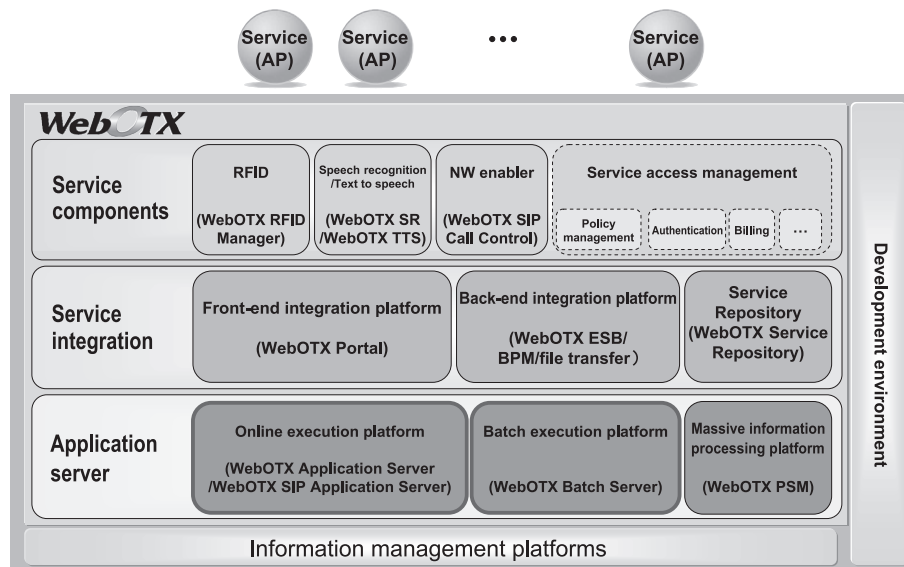

Fig. 1   Outline of service execution platform WebOTX.

evident in cases in which enterprises utilize the services of outside providers instead of building and operating their own proprietary business systems (SaaS).

On the other hand, even in cloud environments, the needs of business systems are universal. The key requirements for long periods of stable system operation are still the high availability, reliability and flexibility that can deal effectively with the sudden changes in traffic and operability standards that can operate complicated systems efficiently. Their importance is expected to increase further in the cloud environment in which the businesses of multiple enterprises coexist in the same data center and also on the same servers.

Major vendors offer application server products as suitable infrastructures for the operation of business systems. The targets of application servers are in general online businesses but the businesses of enterprises usually include the batch jobs concurrently with online operations. The users do not often become conscious of batch operation because most of these do not have a user interface and are executed automatically during the nighttime. However, their importance is not inferior to the online operation. Rather, since their processing should be completed without fault in a limited period of time, they need an infrastructure that is equivalently robust to that of online jobs.

The WebOTX that we introduce in this paper has functioned for more than a decade as a service execution platform featuring high reliability and high operability and it is also positioned as the infrastructure of RIACUBE/SP, which is the SaaS execution platform of NEC. Recently, we have added the WebOTX Batch Server (hereafter called "WebOTX BS") that provides the fundamental functions for batch processing to the line of WebOTX products including the existing WebOTX Application Server (hereafter "WebOTX AS") (**Fig. 1**). In the following sections, we will describe the benefits provided by these products in the cloud environment.

2. WebOTX Application Server (WebOTX AS)

(1) Outline of WebOTX AS

WebOTX AS is an application server that is compliant to Java EE. In addition to providing Java standard specifications, we have applied the mission critical technology that has been cultivated with mainframes in order to improve its reliability and unique operability. In this section, we will discuss the functions and values that these functions mani-

fest in the cloud environment.

(2) High Reliability

Among the many high-reliability functions of WebOTX AS, this subsection describes the multi-process function and multiplicity control function.

WebOTX AS incorporates a control mechanism based on TP monitoring designed to perform a single operation in multiple processes. As a result, even when a single process stops due to a fault in Java VM, etc., the processing can be allocated to another process currently being run so that the operation as a whole can be continued without interruption. Each process is monitored by the monitoring function and automatic recovery is possible so that return to the original status is possible in a short period even in the case of a fault. With SaaS, there are cases in which an operation is run for multiple tenants. When WebOTX AS is applied to these cases, the executed process can be isolated per tenant so that, even if a tenant develops a fault, the service can be continued without affecting other tenants. Since the provision of SLA for users leads directly to the value of the services offered by the service providers like the functions of their services, the high-reliability function of WebOTX brings about important advantages for the providers deploying services in the cloud environment (**Fig. 2**).

WebOTX AS also has a multiplication function that executes multiple processes in a single operation. This function can vary the multiplicity of operations according to fluctuations in the server load that result from the number of requests, thereby enabling flexible operations. For example by using a small number of CPU resources under low load while increasing the multiplicity to maintain the TAT under high load.

(3) Openness

With the V8.1 available since June 2008, WebOTX AS obtained compliance with the Java EE 5 specification for the first time as a Japanese application server. Java EE is the *de facto* standard specification adopted by many application
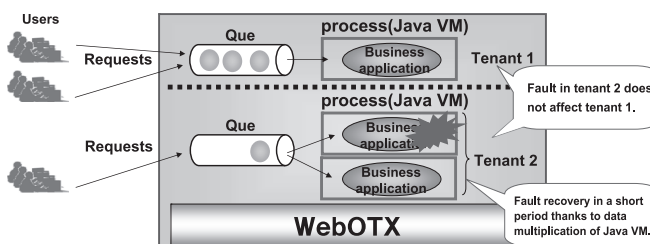

Fig. 2   Fault localization with multi-process system.

vendors. Java EE 5 is a version featuring high ease of use (development productivity) because it is a result of improvements to the main functions including Servlet, JSP and EJB based on the lessons we have gained from the previous version, J2EE1.4 which had massive and bloat characteristics. Furthermore, the additional introduction of JAX-WS2.0 has made it possible to describe web services easily and to significantly improve inter-operability between applications in the cloud environment. We render WebOTX AS compliant to these standard specifications and offer compatibility with previous versions with the intention of protecting the assets of our customers and partners.

(4) Enhanced Operability and Latest Functions
One of the common issues of Java EE-based application servers is the question of the reliability of using Java VM. Java VM is offered free of charge as part of the JDK (Java Development Kit) by Sun Microsystems (purchased by Oracle Corporation in January 2010), the developer as well as by multiple vendors. However, as new technologies are invested in short cycles, difficulties are sometimes encountered due to insufficient absorption of these technologies.
For example, the way an application is installed often causes unexpected issues, such as extreme degradation of performance due to garbage collection (GC) or the sudden shutdown of Java VM itself.
Analysis of these faults is generally not easy. WebOTX AS monitors Java VM behavior patterns from various perspec-

tives thus enabling analyses of bottlenecks and tuning points. In addition, if an abnormal event such as slowdown occurs, WebOTX supports the analysis of operational faults by means of the auto collection function of stack trace and heap information and that for collecting the number of job queuing requests.
The latest version, WebOTX AS V8.3, has had some of its functions enhanced, assuming its use in the cloud environment. One of the enhanced functions is WebOTX Cluster. This function is capable of batch deployment/settings of operations to a large number of grouped application servers via a single outlet.
This arrangement not only reduces the time taken for system building but also prevents faults due to operational mistakes. Furthermore, the setting information that is managed by the integrated management of WebOTX Cluster can be used to add servers for outscaling with a simple operation, so the total construction/operation costs may thus be reduced (**Fig. 3**).
Also, the non-disruptive setting modification function is also provided with V8.3. This enhances the TP monitoring function described above and enables modifications of various Java VM settings including the heap size and system properties, without stopping the running of the operation. The setting change does not require complicated operations such as confirmation of transactions being executed, so the burden imposed on the administrator can be greatly reduced.
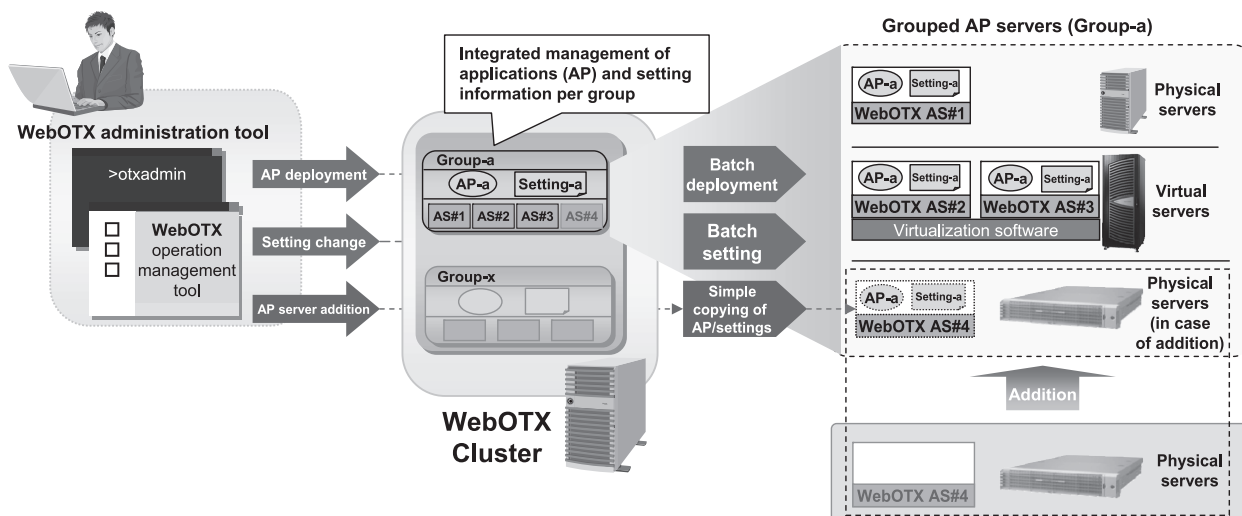

Fig. 3　Batch Deployment/Setting Using WebOTX Cluster

3. WebOTX Batch Server (WebOTX BS)

(1) Background of Batch Processing Using Java

With the open systems, too, the mainstream of the languages used in batch processing has previously been COBOL or C. Batch processing is required to process a large amount of transactions over the limited period of nighttime, high performance is the most wanted requirement. However, Java is disadvantageous in performance compared to COBOL and C due to the presence of Java VM overhead during program execution and Java VM also consumes a considerable amount of memory resource. Therefore, Java has been used only for online processing in most past cases.

However, the trend of using Java also for batch processing has been growing recently. This is because of the following three reasons.

Firstly, the improvement of HW performance has made the Java VM overhead less noticeable than before.

The second reason is a reduction in the number of COBOL users. In the changing trend from mainframes to open computing, the opportunities for building business systems using COBOL have tended to be reduced. This has led to the elimination of COBOL from the technician education programs, thus causing a reduction of numbers as well as ageing of the COBOL engineers. While Java allows development and execution for anyone provided that there is a PC, preparation of the COBOL environment is difficult; this is also one of the factors causing the reduction in the number of COBOL engineers in training.

The third reason is related to the development productivity. Java has already gained the position of a mainstream language in the development of online systems. If different languages are used in online processing and batch processing, engineers familiar with all of them are necessary so the development cost is increased. On the other hand, if both online processing and batch processing can be described in Java, it is not necessary to duplicate the training of developers. The libraries created for online processing can also be used in the batch processing, so the total development productivity can be improved.

(2) Outline of WebOTX-BS

WebOTX BS is an execution environment for Java batch processing based on Spring Batch that has an established reputation as an open source batch-processing framework. Compared to Spring Batch, which is the original, WebOTX BS has enhanced the four functions of performance (memory usage efficiency), reliability, operability and productivity. Each of the enhanced functions is detailed in the following.

(3) Performance Enhancement

With the batch job execution of Spring Batch, Java VM is launched for each job. This means that the heap, stack area and Java standard libraries are loaded for each job so that their memories should be consumed according to the number of jobs.

On the other hand, WebOTX BS provides the batch container for the running of multiple jobs, and each job runs as a thread on the container. This design employing multiplexed execution of jobs enables a significant reduction in memory usage.

With WebOTX BS, the memory usage amount does not increase even when the job multiplicity is increased. Even when typical jobs are run with a multiplicity of around 8, the memory usage is less than 30% of Spring Batch. **Fig. 4** shows an actual comparison of memory usage with Spring Batch.

In normal operation, WebOTX BS is launched in advance, interlocking with the startup of servers so that the processing such as the above described library loading is not required before the start of job execution.

This design considerably decreases the time required before job startup compared to Spring Batch.

(4) Reliability

In addition to the above described batch container, WebOTX BS uses a management unit called the domain that comprises the batch container. It is capable of setting the maximum number of simultaneous executions to both the
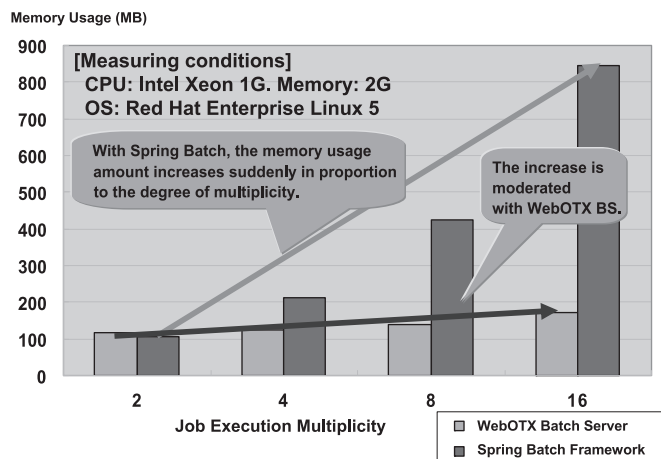

Fig. 4   Comparison of memory utilization efficiency.

batch container and domain. Thanks to this function, even when requests increase suddenly, system overload can be prevented by retaining jobs in the request queues as required. When the batch container is multiplexed, faults related to Java VM can be localized and the effects of faults occurring in specific jobs can be minimized.

(5) Operability

Spring Batch offers only the commands for starting jobs and it is not capable of performing operations such as checking the job status during execution or emergency shutdown. On the other hand, WebOTX BS provides the five functions used frequently in the batch job operation as the basic commands (start, stop, rerun, forced stop and view status), so that users do not have to implement these functions by themselves.

In addition to the above commands, WebOTX BS also offers advanced operation functions such as schedule management and batch job monitoring by utilizing the GUI of MasterScope JobCenter, which is the job management middleware of NEC, based on linkages with this middleware (**Fig. 5**).

(6) High Productivity

Development of batch operations requires specific descriptions of inputs/outputs in/from databases or logics such as repeating or re-running. WebOTX BS offers the processing required in usual batch operations as a framework to support batch operation development.

With Spring Batch, a huge volume of job definitions is necessary for the execution of batch operations, which posed an important barrier to cross for the users. On the other hand, WebOTX BS offers the necessary job definitions in the form of templates for reducing the labor of developers. This allows developers to be freed from troublesome definition operations and focuses on the development of business logic, so that the development productivity is improved greatly. At the same time as the shipment of the new WebOTX BS, we employed SystemDirector, which is NEC's development standard and is compatible with WebOTX BS.

SystemDirector consists of SystemDirector Enterprise that provides the development methodology for upstream processes and the various Java components and SystemDirector BATOOL that provides a high-functionality batch editor and test environment support tool. It offers a development environment equivalent to the online environment (**Fig. 6**).
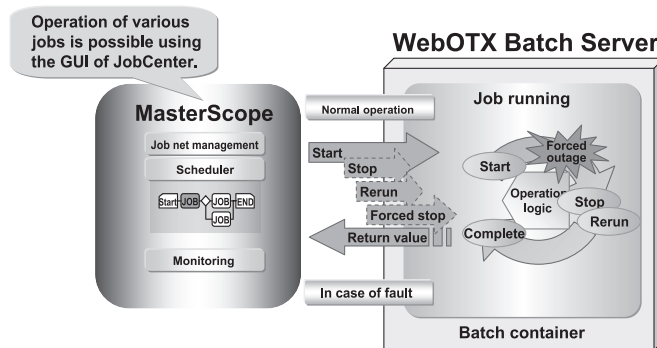

Fig. 5   High operability by linkage with JobCenter.
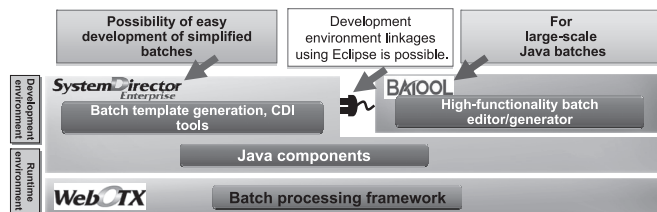

Fig. 6   Provision of a development environment equivalent to the online environment.

4. Conclusion

As was mentioned at the beginning of this paper, the key requirements of business systems in the cloud environment are reliability, flexibility and operability. To meet these expectations, we will continue to enhance the WebOTX product as a core service execution platform for the support of NEC's SaaS platform RIACUBE/SP.

Service Execution Platform "WebOTX" To Support Cloud Computing

Author's Profile

KATOU Masayuki
Senior Manager
3rd IT Software Division
IT Software Operations Unit

The details about this paper can be seen at the following.
Related URL:

http://www.nec.co.jp/WebOTX/