Trend of J2EE Servers in Open Source Software

Noriyuki Hayashi†, Shunsuke Yura, Koji Ito, Tohru Shimma, Yukihisa Nishizawa, and Shigeru Nagai

Abstract

Open source software (OSS), which was made famous by Linux, has extended to the field of Web application servers: the platform for applications on the World Wide Web. This article introduces the trends of OSS J2EE (Java 2 Platform, Enterprise Edition) servers based on J2EE specifications, the industry standard specification for Web application servers. It also mentions the activities of NTT Information Sharing Platform Laboratories aimed at introducing OSS-J2EE servers in enterprises.

1. What is a J2EE server?

The World Wide Web network is becoming important for enterprises as a platform of information provisioning, electronic shopping services for individuals, and electronic commerce for enterprises. The Web application server is a software platform for Web-based applications. It improves the productivity and reusability of application software that must meet frequently varying business requirements, and it improves the reliability, availability, security, and scalability to provide stable services. Therefore, it is recognized as a key platform for Web-based applications.

J2EE (Java 2 Platform, Enterprise Edition) is the standard set of specifications for the functions required for a Web application server. Major software vendors provide Web applications servers that conform to the J2EE specifications (called J2EE servers for short) as commercial products. **Table 1** shows some typical systems that use commercial J2EE servers. They are used not only in applications requir-

† NTT Information Sharing Platform Laboratories
Musashino-shi, 180-8585 Japan
E-mail: hayashi.noriyuki@lab.ntt.co.jp

Table 1. Examples of commercial applications of J2EE servers.

Company	J2EE product	Product type	Features
Toyota Motor Corporation	IBM WebSphere	Parts management system [1]	• High scalability (managing 2–3 TB of data) • High reusability (by using EJB)
Bank of Tokyo-Mitsubishi	IBM WebSphere	Internet banking for individuals/ corporations [2]	• High reusability (meets XML/SOAP standards)
eBay	IBM WebSphere	Online auction system [3]	• High scalability (millions of goods sold per day)
Matsui Securities	BEA WebLogic	Online trading system [4]	• High reliability, high availability (24 hours, 365 days) • High development efficiency (developed in 4 months)
NTT East, NTT West	BEA WebLogic	L-mode gateway system [5]	• High availability, reliability (based on Tuxedo*)
NTT DoCoMo	BEA WebLogic	Stock management system [6]	• High availability, reliability (based on Tuxedo*)
VISA	BEA WebLogic	Payment infrastructure [7]	• High scalability (handles $1 trillion in payments) • High availability, reliability (based on Tuxedo*)

* BEA's transaction processing monitor.
EJB: Enterprise Java Beans XML: extensible markup language SOAP: simple object access protocol

ing high productivity, high reusability, and high scalability, but also in mission critical applications[*1] in cooperation with transaction processing monitors[*2].

2. OSS-J2EE servers

There are several problems with using commercial J2EE servers. One is the high cost of license fees. Another is vendor lock-in: after an initial purchase from one vendor, customers often have no choice but to continue buying compatible products from the same vendor. This limits product choice. Another problem is that the source code is confidential (**Table 2**).

Open source software (OSS) J2EE servers, whose source code is openly available if the accompanying license agreement[*3] is followed, can be used free of charge, so they are attracting attention. There are many OSS-J2EE server products, such as JBoss[*4] and JOnAS[*5]. Moreover, the ASF (Apache Software Foundation), which is developing the widely used Apache OSS-Web server, also offers a product called Tomcat that focuses on functions for creating Web pages and is also continuing the development of a product called Geronimo, which implements all the functions of the J2EE specifications.

One advantage of OSS-J2EE servers is that problems can be dealt with promptly through the exchange of information with the development community. Moreover, the functions of these servers are level with those of commercial products. For example, JBoss 4.0 has acquired J2EE authorization just like them (**Table 3**).

Some examples of OSS-J2EE servers being introduced in the mission critical field are shown in **Table 4**. Users have reported not only reduced costs but also higher performance. Although the number of OSS-J2EE servers in this field is still small compared with commercial J2EE servers, the use of OSS-J2EE servers is expected to grow from now on as it becomes a realistic choice.

On the other hand, OSS-J2EE servers lack some peripheral functions required at the time of system configuration compared with commercial products. The present situation is inadequate, although it can be handled by combining two or more OSSs in a complementary manner. Moreover, there is a shortage of engineers who can perform system design and a

Table 2. Problems with commercial J2EE servers.

Problems	Details
High cost	Large license fee (millions of yen per CPU). Maintenance cost (10 to 20% of license fee).
Vendor lock-in	At the time of system configuration, the product choice is limited to a specific vendor. It depends on the support organization of a development vendor. Bug fixing also depends on the vendor's initiative.
Restrictions on expansion	Since the source code is confidential, expansion depends on the vendor's initiative.

Table 3. Features of OSS-J2EE servers.

Features	Details
Low cost	System introduction cost can be much lower than when using a commercial product.
Vendor-free	System configurations combined with various middleware products, including business ones, are possible.
High extendibility	Source code can be examined and changed; functions can be added.
Cooperation with the development community	The OSS development community offers quick bug fixing and collection of opinions and exchange of technical know-how.
Degree of functional sufficiency	Functions equivalent to commercial products are offered. (JBoss4.0 acquired J2EE authorization)
High availability	Cluster composition equivalent to a commercial product is possible.

*1 Mission critical application: An application requiring high availability and durability because any problems with the system cause major financial damage.

*2 Transaction processing monitor: Software product for managing a series of business transactions.

*3 License agreement: An agreement covering the use, alteration, and re-distribution of an OSS product. There are various kinds of license agreements.

*4 JBoss: An open source J2EE server being developed by the JBoss Organization (http://www.jboss.org/). It is a very popular server and established a record with two million downloads per year in 2002.

*5 JOnAS: Open source implementation of the J2EE specification. It is a pure Java implementation of this specification that relies on the JDK (Java development kit). It is part of the ObjectWeb open source initiative, which was launched in collaboration with several partners.

Table 4. Examples of OSS-J2EE servers in use.

Company	J2EE product	Outline	Details
Weather.com	Tomcat	Weather Information site [8] (50 million pages per day depend on the weather.)	**Mission:** Cost reduction **Approach:** Uses Linux, PostgreSQL, and Tomcat (75 pure Web site servers) **Benefits:** Slashed IT costs by one-third, increased Web site processing capacity by 30% (can handle more than 50 million pages per day)
MoneySquare Japan	Tomcat Seasar	Online foreign exchange trading service [9] (The system can be stopped for a few minutes on Monday morning.)	**Mission:** Cost reduction **Approach:** Uses Linux, PostgreSQL, and Tomcat **Benefits:** Slashed software license fees by about nine million yen, supports 10,000 accounts, 10,000 trades per month
Norwegian Air Shuttle	JBoss	Flight operations system [10] (flight operations, customer management, and seat reservation data management)	**Mission:** Short-term development **Approach:** Uses JBoss **Benefits:** Reduced total cost, development in about 2.5 months

Table 5. Problems with OSS-J2EE servers.

Problems	Details
Lack of peripheral functions	Greater dependence on other OSS products for peripheral functions than with commercial products.
Lack of know-how about modeling the system	Lack of interoperability, information about the performance estimation and tuning
Lack of skilled personnel	Lack of engineers who can maintain the system during frequent upgrades

shortage of operators and maintenance personnel who have sufficient know-how to maintain the system when there are frequent upgrades (**Table 5**).

3. Our activities related to J2EE servers

NTT Information Sharing Platform Laboratories is accumulating know-how for designing systems based on J2EE servers (**Fig. 1**). This section overviews activities being conducted in our Laboratories as a whole.

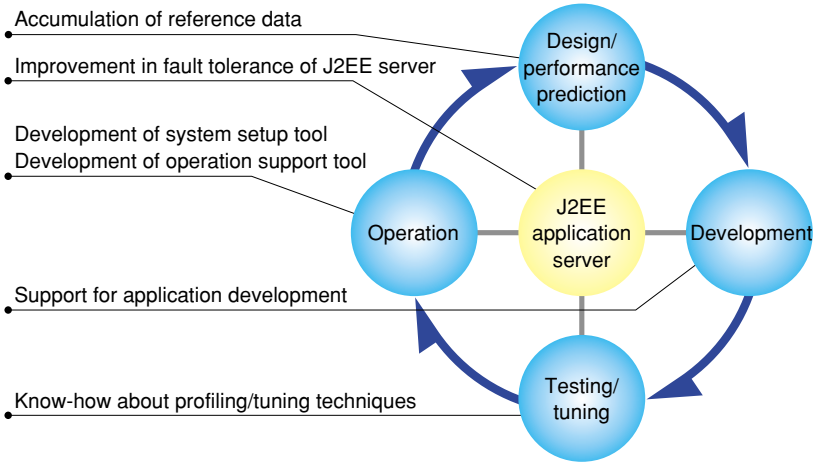


Fig. 1. Our activities related to J2EE.

3.1 Operation/maintenance support

We offer know-how obtained through cooperation with the OSS Support Center [11] and accumulated, and we have developed a support tool to enable OSS-J2EE servers to be introduced into business easily despite the shortage of engineers who can operate and maintain OSS-J2EE servers.

3.2 Accumulation of reference data and know-how about profiling and tuning techniques

To meet the user's performance requirements, we must understand how OSS-J2EE servers perform when they are used in a system. There are also various tuning parameters that interactively affect server performance. Therefore, these parameters must be adjusted and tuned to enhance the performance while providing reliability and stability even under a heavy load. We aim to improve the performance, stability, and reliability of systems using OSS-J2EE servers for the mission critical field. In addition to verifying the tuning by using a standard benchmark application based on the J2EE specification, we will accumulate those results as reference data for building systems.

(1) Accumulating profiling know-how

Profiling is a technique that identifies bottlenecks that lead to per-

formance problems in the system. This process uses stress test tools to simulate the system under a load and profilers to analyze the system from various viewpoints. We are investigating how to use such tools and gaining useful experience.

(2) Accumulating tuning know-how

To solve a bottleneck identified in the profiling process, it is necessary to tune the parameters not only in the OSS-J2EE servers but also in other software in the system such as the operating system and Java virtual machine. We are investigating tuning techniques and tool usage and accumulating such know-how.

(3) Measuring OSS-J2EE server performance

We are taking the approach of evaluating well tuned OSS-J2EE servers by performing benchmark tests and investigating the effect of tuning parameters on system performance. In addition, performance is being compared with comercial J2EE servers.

3.3 Support for application development

The following issues need to be addressed in application development using OSS-J2EE servers.

- Reacting to rapid changes in business requirements
- Selecting OSS products that need to be used together
- Keeping up with frequently updated OSS products

We support rapid software development by providing the following functions as Eclipse (an OSS integrated development environment) plug-ins [12] or project management server.

(1) Shorter development period

We are shortening the development period by using know-how searching, mailing list management, library management functions, and so on.

(2) Recommended combinations

We designate a specific combination of OSS products as our preferred combination and concentrate our know-how accumulation on it.

(3) Enhanced reusability

We enhance source code reusability by standardizing coding styles by applying coding style management.

(4) Guaranteed source code quality

We guarantee the quality of source code by using a source code quality management function that only accepts source code that meets quality standards.

3.4 OSS system setup tool

Installing, setting up, and upgrading a system based on OSS-J2EE servers and other OSS products are complex tasks. In particular, they are very difficult when the system runs on many clustered server machines. We are developing an OSS system setup tool to enable OSS products in the system to be managed centrally. We are tackling the introduction and spread of OSS J2EE servers.

(1) Support for installing and upgrading OSS products

The tool enables the collective installation and upgrading of OSS products. Major new versions may have modified parameter formats, so the tool provides a function for translating parameter values to reduce manual operation.

(2) Central management of parameter values of OSS products

The tool manages parameter values and provides a graphical user interface for changing them. When some values are changed, it updates the parameter values of related OSS products installed on some server machines collectively.

(3) Testing

The tool also provides a test function for examining whether installed OSS products are communicating with each other correctly. Users can confirm that the system is working correctly when installing OSS products and changing their parameter values.

3.5 Operation support

With OSS products, an operation support tool is indispensable at the time of system configuration. For systems consisting of commercial products, there are several tools such as OpenView (Hewlett Packard) and products that perform integrated management like JP1 (Hitachi), and these are widely used. However, there are no formally supported commercial operation support tools for integrated management of systems based on OSS products. We are conducting research and development of operation support tools by using JMX[*6] technology as a base for spreading J2EE-based systems in the future. Our targets are:

(1) Easy function customization

The platform will make it easy to add required functions, customize some portions, and achieve localization.

*6 JMX (Java management extensions): Java specifications for employing/managing all the resources on the network: both hardware and software. Until now, the monitoring of the operation status of a J2EE server has been restricted. More detailed surveillance is becoming possible with JMX. Specifications for JMX are advancing quickly with the spread of J2EE servers.

(2) Reduction of operating work

The labor of upgrading is reduced by exchanging and adding functions over the network.

(3) Expansion of the server management targets

In addition to grasping the operation of a Java-based application server, integrated surveillance with the existing system is achieved and the server surveillance range is expanded.

4. Future plans

The use of J2EE servers will grow steadily in future and basic systems will enter practical use. NTT Information Sharing Platform Laboratories has been actively conducting research and development on systems towards effective use of OSS-J2EE servers from the viewpoint of application development support, system operation support, and system design support.

References

[1] http://www-6.ibm.com/jp/servers/eserver/zseries/library/jirei/toyota/toyota02.html
[2] http://www-6.ibm.com/jp/e-business/library/casestudies/btm.html
[3] http://www-6.ibm.com/jp/NewsDB.nsf/2001/09102
[4] http://jp.bea.com/customers/jirei/matsui_sec.html
[5] http://jp.bea.com/business/biz_com/200212/index.html
[6] http://jp.bea.com/business/biz_com/200301/index.html
[7] http://www.beasys.com/content/news_events/white_papers/BEA_Visa_ss.pdf
[8] http://www.itmedia.co.jp/enterprise/0405/06/epi06.html
[9] http://itpro.nikkeibp.co.jp/free/NC/TOKU1/20030812/1/
[10] http://www.conduct.no/pub/binary?id=795
[11] N. Kobayashi, H. Shibagaki, N. Uchida, and T. Nishihara, “Open Source Software Research Activities in NTT Laboratories,” NTT Technical Review, Vol. 3, No. 2, pp. 12-14, 2005.
[12] http://www.eclipse-plugins.info/eclipse/index.jsp

Noriyuki Hayashi

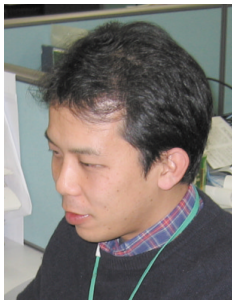
Research Engineer, Software Architecture Project, NTT Information Sharing Platform Laboratories.

He received the B.E. and M.E. degrees in electronics engineering from Hokkaido University, Sapporo, Hokkaido in 1991 and 1993, respectively. He joined NTT in 1993. He has been engaged in R&D of open source software (OSS). His current research interests include operation support for OSS systems. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.

Shunsuke Yura

Engineer, Software Architecture Project, NTT Information Sharing Platform Laboratories.

He received the B.S., M.S., and Ph.D. degrees in information science from the University of Tokyo, Tokyo in 1994, 1996, and 1999, respectively. He joined NTT in 1999. He has mainly been engaged in R&D of service collaboration platforms. He is currently engaged in R&D to support the construction of an OSS system including tuning, installation, and upgrading.



Koji Ito

Engineer, Software Architecture Project, NTT Information Sharing Platform Laboratories.

He received the B.E. and M.E. degrees in electrical engineering from Waseda University, Tokyo in 1996 and 1998, respectively. Since joining NTT in 1998, he has mainly been engaged in R&D and the commercial introduction of security platforms such as regional currency management systems and web access control systems. He is currently engaged in the development of a system that supports application development with OSS.

Tohru Shimma

Engineer, Software Architecture Project, NTT Information Sharing Platform Laboratories.

He received the B.S. degree in electrical engineering from Tokyo University of Science, Tokyo in 1991 and joined NTT Telecommunication Software Headquarters the same year. He has been engaged in R&D of OSS. His current research interests include operation support for OSS systems. He is a member of IEICE.

Yukihisa Nishizawa

Engineer, Software Architecture Project, NTT Information Sharing Platform Laboratories.

He received the B.E. and M.E. degrees in applied physics and physico-informatics from Keio University, Yokohama, Kanagawa in 2000 and 2002, respectively. He joined NTT in 2000. He has been engaged in R&D of OSS. His current research interests include profiling and tuning techniques for OSS systems. He is a member of IEICE.

Shigeru Nagai

Senior Research Engineer, Software Architecture Project, NTT Information Sharing Platform Laboratories.

He received the B.E. degree in communication engineering from Tohoku University, Miyagi, Sendai in 1985 and the M.E. degree in information engineering from Tohoku University in 1987. Since joining NTT in 1987, he has mainly been engaged in a study of a computer mapping system for managing facilities. He is currently engaged in the development of effective ways of using OSS application servers such as JBoss.