Orchestrating a brighter world

NEC brings together and integrates technology and expertise to create the ICT-enabled society of tomorrow. We collaborate closely with partners and customers around the world, orchestrating each project to ensure all its parts are fine-tuned to local needs. Every day, our innovative solutions for society contribute to greater safety, security, efficiency and equality, and enable people to live brighter lives.
Index

1. WebOTX Application Server features
   - Features for operability improvement
   - Features for reliability improvement
   - Features for development easiness

2. License and line-up
What is WebOTX Application Server?

Java EE5 compliant highly reliable application server enabling 24x7 non-disruptive operation

“WebOTX Application Server” delivers the “Responsiveness”, “Speed”, and “Reliability” required for enterprise systems in the networked information society. WebOTX provides latest Java execution environment with mainframe-class reliability.
1. WebOTX Application Server features

- Features for operability improvement
- Features for reliability improvement
- Features for development easiness
Automated registration process of setting information

High reliability and availability by easy operation in cloud environment

- One time registration of server information enables automatic registration for server configuration change hereafter.
- Automatic scale out is realized, hence administrator’s effort can be reduced.

1. Automatic registration setting
   - Issues server registration notification to distributed management server when domain is started.

2. VM image creation
   - VM image distribution
   - Regardless of server number, executes procedure 1 only.
   - Executes profile setting only one time (connection information with distributed management server etc).

3. VM image distribution
   - Other application or component
   - OS
   - WebOTX AS
   - Application

Greatly reduces cloud operation cost
Easy system configuration in the cloud environment

Automation of domain environment configuration by setup/script generation feature of domain

- Script generated solving dependency/condition between parameters
- Domain replication can be done easily

Greatly reduces operation cost by reducing effort for infrastructure construction
Usability improvement in the cloud environment

Manage distributed App server suite as single virtualized App server

- Consolidated application deployment and configuration change to all App servers.
- Easy addition of App server for scale out systems.
- Uniform monitoring of App server operating status, and event notification that occurred in App server. Easy integration with monitoring software such as MasterScope products etc.

Consolidated monitoring to simplify management and reduce cost

Significant cost reduction for construction/operation by uniform management of all servers
Nondisruptive configuration change

Availability improvement utilizing parallel process coordination

- Change Java VM option (Java heap size, Java system property, etc.) by highly reliable Transaction Processing (TP) monitoring function without shutdown of business operation.
- Administrator’s work is reduced as complex administration operation (shutdown operation and executing transaction confirmation, etc.) before configuration change is not required.

Only 1 procedure is enough

WebOTX

Prepare for stopping …request completion

Shutdown Operation

Configuration Change

Start operation

4 procedures needed

Other App server

App server stop

App server start

Users

Service utilization with old configuration

Service utilization with new configuration

Service utilization with old configuration

Service utilization with new configuration

Enables configuration tuning while continuing services
Failure analysis support (business App monitoring)

Automatic information collecting function in case of failure is enhanced

Collect failure information automatically
- Collect information in case of abnormal process as well as stall and slowdown
- Output queue stagnation number to log when queue stagnation occurs

Minimize Need for Problem Replication to Collect Detail Data, and rapidly determine failure cause
Failure analysis support (memory shortage)

Supports rapid investigation of cause of memory leak

Regarding OutOfMemoryError error by memory shortage, which was difficult to be investigated in production environment, collects heap information for investigation. (*)

- From collected heap information, object of memory leak cause can be specified.

<table>
<thead>
<tr>
<th>rank</th>
<th>size(bytes)</th>
<th>object count</th>
<th>class name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>476232</td>
<td>4447</td>
<td>myclass</td>
</tr>
<tr>
<td>2</td>
<td>462976</td>
<td>1686</td>
<td>byte[]</td>
</tr>
<tr>
<td>3</td>
<td>453952</td>
<td>1223</td>
<td>java.lang.Class</td>
</tr>
</tbody>
</table>

When OutOfMemoryError occurs, automatically collected. Cause can be investigated without trying reproduction.

*: JVMTI feature of J2SE 5.0 is utilized. J2SE 1.4 can not be utilized.
Memory shortage detection or GC monitoring feature are provided as a WebOTX standard feature.

- GC occurrence number and thread dump information can be referred in real time.
- By using administration tool, report can be exported based on detailed status.

Simplified analysis of bottleneck in Java VM or tuning point
Failure analysis support (information collection)

Efficient information collection in case of emerging problems

Information collection feature for efficient collection during failure

- Automatically collects logs, setting files, etc required for failure analysis with minimum operation
- Compresses collected data in 1 file automatically

Rapidly resolves problem by collecting failure analysis information with easy operation
Failure analysis support (integrated monitoring)

Breaks down monitoring items. Fulfilling monitoring point for tuning.

- Measures application operating status in server in detail
- Seamless integration with administration tool enabling periodical information collection and monitoring

Application monitoring
MasterScope Application Navigator

WebOTX administration platform

Web container
- Web application

EJB container
- EJB application

Transaction service

Web container
- Execution number
- Session number
- Thread number
- Execution time
- Cumulative processing time
- Data transfer number
- Data transfer rate
- Connection number

EJB container
- Execution list
- Count number per status
- Commit number per target
- Rollback number per target

Administrator

By collected information from WebOTX, elaborate tuning is enabled
1. WebOTX Application Server features

- Features for operability improvement
- Features for reliability improvement
- Features for development easiness
Real time monitoring by TP monitor

Facilitate Application Failure Isolation to minimize affected Business App

TP monitor process that is not existing in other vendor’s product is implemented

- System disruption time is minimized by real time monitoring/recovery of abnormal process

![Diagram of TP monitor process with WebOTX AS, Client, process (Java VM), thread, TP monitor, automatic recovery of process failure, business app execution time monitoring, and signal detection of process abnormal termination.]

Stable operation by system disruption time minimization
Queue control

Properly controls overload request

- Controls overload by queue control
- If simultaneous processing ability is over, sends Sorry screen to the browser

1. Controls simultaneous DB access number according to execution multiplicity
2. Queues request over execution multiplicity, and holds temporarily
3. Displays error to guarantee response if request is full in queue
4. Sends freely customized error screen to browser

WebOTX
Process multiplicity setting

Dynamically changes multiplicity according to flow changing time to time

Dynamic change of execution multiplicity (request number) by command

- Ensures response by increasing request in peak time

Schedule operation to automatically put multiplicity change command just before peak time is enabled

Execution process (dynamic addition)

WebOTX

Execution process (Java VM)

Normal: Multiplicity 3

Peak: Multiplicity 6
Priority control

Guarantees response in heavy load by priority control of process

- Controls CPU allocation priority in OS level
- Detailed setting such as priority change on peak time is enabled

Start immediately because high priority process is prior to others

*: Not supported in Linux

App server of other vendors can not enable priority control
Process grouping

Makes business application isolated to minimize affected Business App

Groups application according to business characteristics
- Realizes independent operation by running on different Java VM
- Other app groups are not affected in case of failure

- Groups apps according to business requirement
- Controls multiplicity and queue wait number by group

Other groups are not affected since Java VM is different in case of failure

Works normally if another group request is flooded

App server of other vendors spreads failure/delay to whole instance
Multiple Java VM (1/2)

Minimizes Java VM failure and continues business

Advantages of “1 instance (port), multiple Java VM” implementation

- When Java VM aborts, continues business without instance stoppage
- Wait requests in queue are not affected in case of failure

Application server of other vendors cause instance stoppage due to Java VM failure, leading to server fallback

WebOTX

Process (Java VM)

Biz App

Users

Instance (port)

Queue

Rapidly allocates requests to other process

Multiple Java VM in 1 instance (port)

Process (Java VM)

Biz App

Control service

Users

Instance (port)

Queue

1 Java VM in 1 instance (port)
Advantages of “1 instance (port), multiple Java VM” implementation

- When Java VM aborts, continues business without instance stoppage
- Wait requests in queue are not affected in case of failure

In app server of other vendors, requests in queue are all destroyed
Limited reliability/availability improvement despite SI effort

- Wait requests in queue are not saved even with multiple instances
- Logic setting of instance allocation (including operation during failure) is required
- Required time for failure detection is not shortened

Setting/built-in required for allocation to another instance (port)

Wait requests are all destroyed
Order guarantee of redelivery JMS message

In addition to normal delivery, provides order guarantee of redelivery

- System with higher reliability can be configured
- Application built-in is not required for order guarantee in case of failure
  - Without order guarantee of redelivery
  - More than one message delivered in one time
  - Redelivery message will not be target of next processing (becomes last in line)

- With order guarantee of redelivery
  - Message is delivered one by one
  - Redelivery message will be target of next processing

Guarantees order of redelivery not specified in JMS standard specification.

Processing for guaranteeing message order is not required!
Cluster control feature of JDBC connection pool

Improves availability of database connection

Provides standby/load balancing feature of JDBC connection pool

- Hides connection failure, and rapidly recovers from database server failure
- In load balancing feature, switches connection pool every time when JDBC connection is obtained, and do load balancing equally (weighting is possible)

Before

![Diagram showing the transition from before to after using the cluster control feature]

- Application
- WebOTX AS
- Database (primary)
- Database (standby)

Availability is improved by cluster control feature which is not depending on database

*1: License for standby is required
1. WebOTX Application Server features

- Features for operability improvement
- Features for reliability improvement
- Features for development easiness
Productivity improvement with Java EE 5
- easy EJB development -

Simplified programming model without deployment descriptor

- EJB unique interface no longer have to be constructed. In business logic part, it is implemented as a plain old Java object (POJO).
- Complex deployment descriptor causing description mistakes is not required.
- Code (log output code, etc) which was traditionally required to be implanted in lots of business logic part can be injected without changing business logic.
- EJB container traditionally used for examining EJB is not required.

EJB 2.1

EJB container

Home interface

Component interface

Business interface

Deployment descriptor

EJB 3.0

EJB container

Business interface

Deployment descriptor not required

Only Java knowledge is required to implement EJB business logic
Provides more free web service with support for JAX-WS 2.0

- Compared to the web service before, various features are added to expand coverage area
  - Provides perfect mapping to Java code of all XML styles
  - Supports asynchronous execution
  - Provides interrupt processing enabling to enter in communication of all protocols and ports
  - Realizes higher speed communication by optimization of message transfer

- Definition file with tricky and long description is not required thanks to annotation

Before (JAX-RPC 1.1) vs. Java EE 5 (JAX-WS 2.0)

- Only business logic required by using annotation
Productivity improvement with Java EE 5
- POJO that anyone can develop -

Business logic made by POJO from implementation of annotation

By implementing annotation, following components will be made by POJO. As a result, code is largely reduced.

**POJO** - Java object which is not limited by any rule and framework. Can be implemented with **Java knowledge only**

*Web service *EJB *Entity *Client

Everyone can easily develop and examine it
Productivity improvement with Java EE 5
- Parallel development is promoted -

Enabled parallel development through development and test by excluding dependency between components.

- In Java EE 5, web application framework JSF is supported as a standard.
- JSF enhances independency of three components, presentation, business logic, control, enabling parallel development of each expert.
- Technologies used by each component is freely changed.

Technologies used by each component:
- Presentation (JSP)
- Business logic (POJO/EJB)
- Control (Servlet)

Connection test

Productivity is improved by promoting parallel development.
2. License and line-up
## WebOTX Application Server product information

<table>
<thead>
<tr>
<th>Product name</th>
<th>Details</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebOTX Media</td>
<td>WebOTX installation media</td>
<td></td>
</tr>
<tr>
<td>WebOTX Application Server Express</td>
<td>Express edition</td>
<td>Server</td>
</tr>
<tr>
<td>WebOTX Application Server Foundation</td>
<td>Foundation edition</td>
<td>CPU</td>
</tr>
<tr>
<td>WebOTX Application Server Foundation CPU License</td>
<td>Foundation additional CPU license</td>
<td>CPU</td>
</tr>
<tr>
<td>WebOTX Application Server Foundation Virtual CPU License</td>
<td>Foundation additional virtualization CPU license</td>
<td>CPU</td>
</tr>
<tr>
<td>WebOTX Application Server Standard</td>
<td>Standard edition</td>
<td>CU</td>
</tr>
<tr>
<td>WebOTX Application Server Standard CU License</td>
<td>Standard additional CU license</td>
<td>CU</td>
</tr>
<tr>
<td>WebOTX Application Server Standard Virtual CU License</td>
<td>Standard additional virtualization CU license</td>
<td>CU</td>
</tr>
<tr>
<td>WebOTX Application Server Standard for Standby System</td>
<td>Standard edition (standby)</td>
<td>CU</td>
</tr>
<tr>
<td>WebOTX Application Server Standard for Standby System CU License</td>
<td>Standard additional CU license (standby)</td>
<td>CU</td>
</tr>
<tr>
<td>WebOTX Application Server Standard for Standby System Virtual CU License</td>
<td>Standard additional virtualization CU license (standby)</td>
<td>CU</td>
</tr>
<tr>
<td>WebOTX Application Server Enterprise</td>
<td>Enterprise edition</td>
<td>CU</td>
</tr>
<tr>
<td>WebOTX Application Server Enterprise CU License</td>
<td>Enterprise additional CU license</td>
<td>CU</td>
</tr>
<tr>
<td>WebOTX Application Server Enterprise Virtual CU License</td>
<td>Enterprise additional virtualization CU license</td>
<td>CU</td>
</tr>
<tr>
<td>WebOTX Application Server Enterprise for Standby System</td>
<td>Enterprise edition (standby)</td>
<td>CU</td>
</tr>
<tr>
<td>WebOTX Application Server Enterprise for Standby System CU License</td>
<td>Enterprise additional CU license (standby)</td>
<td>CU</td>
</tr>
<tr>
<td>WebOTX Application Server Enterprise for Standby System CU License</td>
<td>Enterprise additional virtualization CU license (standby)</td>
<td>CU</td>
</tr>
<tr>
<td>WebOTX Developer</td>
<td>WebOTX development environment</td>
<td>Server</td>
</tr>
<tr>
<td>WebOTX Administrator</td>
<td>WebOTX administration environment</td>
<td>Server</td>
</tr>
</tbody>
</table>

Please see WebOTX website for support platform information.
http://www.nec.com/global/prod/webotx/
**WebOTX Application Server Line-up**

Select appropriate product according to system scale

<table>
<thead>
<tr>
<th>License</th>
<th>Edition</th>
<th>Unit</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server unit</td>
<td>WebOTX Application Server Express</td>
<td>Simple Server</td>
<td>For small scale system. High speed Java EE execution environment compared to OSS at reasonable price. GUI based administration environment.</td>
</tr>
<tr>
<td>CPU unit</td>
<td>WebOTX Application Server Foundation</td>
<td>Simple Server CPU</td>
<td>For small/middle scale system. Simple and highly reliable system at a low price. Automatic recovery feature of SW failure is deployed as a standard.</td>
</tr>
<tr>
<td>CU unit</td>
<td>WebOTX Application Server Standard</td>
<td>Robust Server CPU</td>
<td>For middle/large scale system. Highly reliable and available system for strict demand for reliability. Cost advantage at multiple cores CPU usage. <em>(Same price for 2 cores through 8 cores per 1 CPU)</em></td>
</tr>
<tr>
<td>CU unit</td>
<td>WebOTX Application Server Enterprise</td>
<td></td>
<td>Sophisticated platform for huge scale system. Lower price than other vendors’ when multiple core CPU is used. <em>(Same price for 2 cores through 8 cores per 1 CPU)</em></td>
</tr>
</tbody>
</table>
(Reference) WebOTX Application Server License

Sample:
How to count the license in 1Server / 2CPU / 4cores model

<table>
<thead>
<tr>
<th>License</th>
<th>WebOTX Application Server</th>
<th>Server</th>
<th>CPU</th>
<th>Core</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server unit</td>
<td>Express</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU unit</td>
<td>Foundation</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CU unit</td>
<td>Standard</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CU unit</td>
<td>Enterprise</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
License system reasonable for multiple core CPU

Overwhelming cost performance beyond the other vendors (Standard/Enterprise)

- Same price for 2 cores through 8 cores (Please consult if more than 9 cores machine is used). More reasonable according to core number.
- Price increase is moderate according to CPU number. More saving if CPU number is increased.
- Low cost infrastructure for large scale system.

**In case of core increase (6 CPU case)**

- Same price for more than 2 cores. WebOTX is saving.

**In case of CPU increase (4 cores case)**

- WebOTX is reasonable according to CPU number.

<table>
<thead>
<tr>
<th>Core number</th>
<th>Socket number</th>
<th>WebOTX</th>
<th>Other vendors’</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

- 1/4 of other vendor at 8 cores / 6CPU case!

<table>
<thead>
<tr>
<th>CPU number</th>
<th>Socket number</th>
<th>WebOTX</th>
<th>Other vendors’</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

- 1/3 of other vendor at 4 cores / 8CPU case!
Virtualization License

License based on physical server which is not affected by virtual server number

- Basic license according to physical server regardless of virtual server
- Additional license according used CPU in physical server unit
- Additional license can be used commonly in Windows/Linux

E.g.) In case of 4 virtual servers consolidated in 1 physical server of 2CPU quad core, and allocates 2 cores for each virtual server. *(WebOTX AS Standard)*

License before:
Standard license (4) + Additional CU license (4)

New license:
Standard license (1) + Additional CU license (2)

Not only HW but SW license can reduce the cost by virtualization

(*)Operation of VMware and Hyper-V is verified
Thank You

Application Service Platform for the age of cloud-computing

For more product information & request for trial license, visit >> http://www.nec.com/webotx/

For more information, feel free to contact us - global@soft.jp.nec.com
Orchestrating a brighter world

NEC