Java-based Application Server

JAVA ISLAND DUEL

BERNHARD BABLOK AND ULRICH WOLF

Java applications servers are as common as espressos in a coffee bar. We present a brief overview of some of the available servers and give some tips on selection criteria. It all depends on the service

Unfortunately *applications server* is not a protected trademark. Which means that nowadays any application which can create dynamic websites is an applications server. In these, the presentation layer is not a core task of the applications server, especially since an HTML interface represents only one of many options for the user interface in modern systems.

In my opinion the following definition is more apt: *An applications server makes available a generic processing environment for applications logic*. The services offered by an app-server represent an important evaluation criterion for selection. The strengths and weaknesses of various application servers are a result of their respective origins. Database manufacturers who are leaping onto the e-commerce bandwagon do of course also provide powerful solutions for the persistence layer, while an ORB manufacturer shines in the field of distributed objects.Important services provided by app-servers are:

- *Resource management*, e.g. connection-pooling of database connections
- *Persistence* of objects
- *Authentication*, *Authorisation*, *secure connections*
- Support for distributed objects, perhaps by means of RMI or CORBA
- Finding objects (*naming*)
- Support for (distributed) *transactions*

Applications servers which meet the J2EE specification (*Java 2 Enterprise Edition*) must provide corresponding standardised implementations for the listed services. In particular they must supply a complete Enterprise JavaBeans (EJB)-Container. Details of EJB-Containers can now be looked up in the *Coffee-Shop* .

Anyone who has to access their data, which may be stored on a large computer, will in any case also still need the corresponding support from the applications server.

Management of applications servers

Smooth operation of critical business applications depends not only upon these services, but above all on the management of the server in day-to-day operation. If the server lies in something of a DMZ, a nice CORBA-supported admin-console will be no help at all (unless a, usually unspeakably slow, HTTP tunnelling of the CORBA connections is possible).

Automated server management is possible, via integration in SNMP-based tools or proprietary solutions (such as Tivoli). Syslog support is sadly uncommon, although no platform-specific code would be necessary for it. This is also the case with log files, in which log reports are provided with numerical IDs, which considerably simplifies parsing, compared to text-based logs.

Another important criterion is the platform availability and the consumption of resources. Pure Java implementations run, theoretically anyway, on all platforms, but sometimes the devil is in the detail. Sometimes the installation fails on Linux simply because a corresponding installation routine is missing. If individual parts of the app-server are not implemented in Java (e.g. parts of CORBA services), platform-specific versions are necessary.

Consumption of resources can in fact be controlled via correspondingly large computers and the aforementioned load-balancing. But a complete development environment (with server, IDE and database) should still run at a reasonable speed on an individual PC. This is particularly important for mobile developers, who rarely have 512 MB of memory available on their notebook.

Frameworks and development tools

Anyone who develops applications for app-servers is keen to have as little as possible to do with the details of implementation. Appropriate development tools simplify and accelerate the process here. Interfaces for existing IDEs such as Borland's JBuilder should also have their place in the decision-making process.

In the commercial domain, there is also a requirement that standard logic, such as for address management, should not have to be constantly re-implemented. The manufacturer of the app-server will supply a framework along with it, or will keep to existing standards, so that the integration of solutions from third party suppliers poses no problems. An advantage of following standards is that one is not tied to the manufacturer of the server.

Brief overview of major servers

In the following sections, there is a brief overview of the principal applications servers. For reasons of time, it has not always been possible to install and research all features, so it was necessary to rely on information from the manufacturer. These details should be verified before use.

If there is no statement on Linux compatibility, the product did not enter into our selection (unless it was explicitly a pure Java server). Since many servers run under Unix derivatives such as Solaris or AIX, the picture might look very different in a few months or even weeks.

Enhydra Enterprise

The future version 4 of the open source *Enhydra Enterprise Application Server* (at the moment the fourth alpha version is being tested) offers, firstly, complete support for the J2EE standard, and secondly through various work groups it implements a range of tools, which greatly simplify the development of HTML/XML-based applications. Sub-components are taken over from other open source projects, so for example JOnAS is included in Enhydra Enterprise. Enhydra Enterprise is presented in more detail in the context of a *Coffee-Shop*.

Version 3.1 of the *"Enhydra Application Server"* (without "Enterprise") implements much of the functionality addressed above and has already been introduced in the *Coffee-Shop* (see [14]).

JOnAS Application Server

JOnAS came about as the result of a joint venture between various firms, with the participation of France Telecom. JOnAS implements the latest EJB 1.1 standard. The Object Web Group, which maintains JOnAS and the projects connected with it Joram and Jonathan, consists of developers at French universities and telecommunications firms. Its objective is to build a platform for distributed objects for the telecommunications industry.

jBoss Application Server

jBoss (also open source) was originally called EJBoss, but the abbreviation EJB is a registered trademark of JavaSoft, so the name was simply abbreviated. The jBoss Web site states that some 500 developers world-wide are supporting the further development of jBoss. Version 2.0, the current release, makes available an EJB 1.1 standard environment, and development is progressing on support for the EJB 2.0 specification in the next release.

jBoss is really compact and comes either stand-alone, or bundled with the servlet engines *Tomcat* and *Jetty*. The greatest shortcoming is the lack of documentation but this is being addressed at top speed. A lot of this can already be seen online, though at present it is not yet possible to download the complete documentation. In the next article the project will be presented in more detail.

Exolab Application Server

The Exolab app-server has not yet reached a usable state. The aim is to implement a server which right from the start supports the EJB 2.0 specification. Chief architect of the server is Richard Monson-Haefel, the author of the classic *Enterprise Java Beans*.

Orion Application Server

The application server from the small Swedish firm Evermind Data is fast and, in comparison with other commercial products, a bargain. Each professional installation will cost US$1500. For development and non-commercial use the server is free. Orion, in the current version 1.2.9, already supports parts of the future EJB2 standard, including Message Driven Beans. And parts of the, not yet completely specified, servlet-standard 2.3 are also implemented. The Orion application server comes with its own Web server, but it can also be persuaded to work with Apache.

Silverstream Application Server

The stock market-quoted software house Silverstream, from Massachusetts, is supplies a complete e-commerce suite; the Silverstream application server is a central component of this. Currently the software is available in version 3.5, 3.7 (at beta stage at the time of writing) is scheduled for released at the end of 2000. Worth noting is the integration of various IDEs, including Macromedia DreamWeaver.

Inprise Application Server

Inprise maintains one of the traditional models among Web applications servers. Integration with JBuilder goes without saying in this case. Its roots lie in the Visibroker, from Visigenics, one of the first Object Request Brokers (ORB). Borland bought Visigenics in 1997. The Inprise application server is written in 100 per cent Java. Installation routines are currently available for Red Hat 6.0, it supports the common J2EE standards such as EJB 1.1. At this point we could go on forever. Nearly every company above a certain size which has nailed its colours to the mast of system integration, has a Web application server in its schedule. In our overview a few of those discussed here are listed. Two commercial products with very high market shares, namely IBM Websphere and BEA Weblogic, are presented from page 56 in somewhat more detail. ■

The author

Bernhard Bablok works as a systems programmer in the field of systems-management. When he is not listening to music, cycling or hiking, he spends his time in the world of object orientation.

Info

[1] Lutris Enhydra Enterprise: http://www.enhydra.org [2] IBM WebSphere Application Server: http://www-4.ibm.com/software/webservers/appserv/ [3] BEA Weblogic Application Server 5.1: http://www.bea.com/products/weblogic/server/ [4] Silverstream Application Server: http://www.silverstream.com/ [5] Inprise Application Server: http://www.borland.com/appserver/ [6] Orion Application Server: http://www.orionserver.com [7] JOnAS Application Server: http://www.objectweb.org/jonas/ [8] jBoss EJB-Server: http://www.jboss.org [9] Unify eWave Application Server: http://www.unifyewave.com/overview.htm [10] Exolab Application Server http://openejb.exolab.org [11] Oracle Application Server: http://www.oracle.com/ip/deploy/ias/ [12] IONA iPortal Application Server: http://www.iona.com/products/ [13] JRun Application Server http://www.allaire.com/products/jrun/ [14] Coffee-Shop: Im Zeichen des Otters [Under the sign of the otter], Linux-Magazin 08/2000, p. 128 ff. [15] Richard Monson-Haefel: Enterprise Java Beans, 2nd edition, O'Reilly 2000

■