

# Web Application Server SHIFT OPERATION

JOCHEN LILICH

WWW-Applications servers accelerate the development of more powerful Web applications and simplify their operation. What do applications servers do, how do they work and how can one decide on a specific one?

## Overview

- Zope for example : page 16
- ArsDigita Community System: page 23
- Java-based servers: page 26
- jBoss, a free EJB server page 28
- Commercials: Websphere and Weblogic: page 30
- Cocoon/ XML page 32

"Up to date", "interactive" and "individual" are the requirements for a modern web launch. And static HTML pages don't cut it, not by a long way. "Just In Time" means a page is only compiled at the moment when a visitor invokes it. Behind the classically-presented content such as text and graphics, there is also a program logic to be laid down which takes care of the production of the pages. Web development is teamwork. For clear division of responsibilities, content, layout and functionality are managed separately. To separate layout, templates are usually used, in which the content is then inserted at specified places. Editors do not have the option of altering the layout according to their own taste, but concentrate solely

on content. And developers care about program logic and nothing but.

Common script languages such as PHP, ASP or Perl do not offer this clear separation, because the Web logic is embedded in the HTML pages. This is where web applications servers come in. Also referred to as Middleware - they sit in the middle, between the application realised by a web service, and the Web server, which accepts the calls from visitors and the database, which manages the components of the Web site. The term for this is "Three-Tier-Architecture".

At least 50 different products crowd the applications servers market. Amongst the proprietary servers; the best-known are BEA Weblogic, the Oracle Application Server and IBM WebSphere. However in the Open Source world things are really moving now. In particular Zope, Enhydra and Midgard have already made names for themselves. A selection of other OSS applications servers can be found in the Info-box.

## How goes it?

So how does this type of applications server (from now on called "AS" for short) work? As usual, it all

starts with the web query from the visitor to the site. By meshing between itself and the AS the Web server recognises that the query applies to the Web application and passes it on to a colleague. From the URL the AS sees which page and thus which part of the application should be invoked. This page must now be produced. To do this, layout and the program logic of the page are needed. The layout is either statically assigned or is selected dynamically on the basis of criteria such as browser model, day of the week, time of year or stored user preferences. By executing the program logic the content-related elements are combined to make a complete page in the selected layout. The result is now usually an HTML page, which is sent back to the enquiring browser. In the future the data could also be supplied in a more powerful format such as XML (more on this later).

## High demands

Once a website is big enough for it to be worth using an AS, the size of the operation also imposes greater demands on the system:

An AS must be capable of processing a page query at the same rate as a simple static HTML file would be issued. Nothing is more annoying than potential customers who leave the site because of long waiting times. And an AS system must also run stably and reliably. The proven modules Linux and Apache deliver the goods in this respect. The best solution for both problems, though, is called clustering.

## XML – at last

In Web applications data often come together from very different sources: texts in ASCII form or

as text processing documents, graphics with additional information, customer data from the merchandise information system and so on. So it was not long before the lack of a standard, flexible data format became intolerable. The remedy arrived in the form of Extensible Mark-up Language - XML.

Anxiously awaited as a saviour for data gatherers, XML is now broadly implemented for the platform- and application-independent representation of information - but mostly only on the provider side. As soon as the browser technology is able to represent XML data agreeably with the aid of stylesheets, it will also be able to show off its advantages on the user side.

## Spoilt for choice

How then do you choose an applications server? In the case of large projects the software must support distributed development, as far as possible with version management and access protection (Locking).

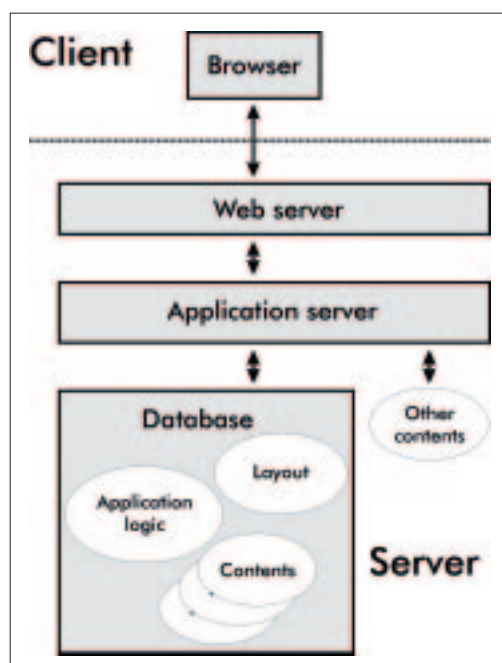
As a programming language, Java is a preferred, along with Perl and PHP, but other languages like Tcl or Python have also become favourites with Web developers. For many applications, development is greatly accelerated if one can make use of flexible tools and ready-made application modules. Components for easy database access, connection pooling, session management and browser interaction can be found in the form of Enterprise Java Beans, or as Perl modules in the CPAN.

If many data sources and thus also data forms are used, open interfaces and standards such as XML are one criterion. Flexible and powerful data administration must be possible at all times.

The fact remains that in the field of "Web applications" a great deal is still in flux. An applications server is a puzzle made of so many individual pieces, themselves subject to heavy modifications (XML, Java, etc.), that this is where a system with open interfaces and standards conformity has the best chance of survival. Which is the best argument for using open source software here, too. ■

## The author

*Jochen Lillich is a technical director of the Linux system house TeamLinux in Graben near Karlsruhe. When he is not sitting at the keyboard of a computer, then he is usually at that of a piano.*



Web application servers usually have a 3-tier architecture ■

## Info

[1] XML W3C Draft: <http://www.w3.org/XML/> [2] Aquarium: <http://aquarium.sourceforge.net/> [3] Ariadne: <http://www.muze.nl/software/ariadne/> [4] Enhydra: <http://www.enhydra.org/> [5] IxpServ: [http://www.commandprompt.com/products\\_1xp.1xp](http://www.commandprompt.com/products_1xp.1xp) [6] Midgard: <http://www.midgard-project.org/> [7] rmdms: <http://rmdms.sourceforge.net/> [8] Zope: <http://www.zope.org/> [9] <http://www.linuxde.org> [10] Application Servers and Linux: <http://www.linuxplanet.com/linuxplanet/reports/1146/1/> [11] CPAN: <http://www.cpan.org>