Oracle Application Server 10*g* Release 2 (10.1.2): Overview of Oracle HTTP Server Components

An Oracle White Paper
January 2005

Oracle Application Server *10g*: Overview of Oracle HTTP Server Components

ABSTRACT

Oracle Application Server 10g HTTP Server (OHS) provides a web server and also the necessary infrastructure to create dynamic applications.

Based on the proven Apache 1.3 and Apache 2.0 Infrastructure, Oracle HTTP Server adds significant enhancements – Perl (via mod_perl, cgi), C (via CGI, and FastCGI), C++ (FastCGI), PHP, Oracle Single Sign On, mod_security and of course the popular Oracle language - PLSQL. It can also be a proxy server - both forward and reverse.

This paper provides an overview of these powerful features of Oracle HTTP Server (OHS).

INTRODUCTION

Oracle Application Server 10*g* Release 2 is a fully featured application server consisting of a very large number of sub-products. Nearly all of these sub-products depend on the HTTP server since HTTP is the dominant protocol in which Application Servers communicate to outside entities.

This paper focuses on the features of Oracle HTTP Server, called OHS. It is based on the well-known Apache HTTP servers and in this release has versions that are based on both the Apache 1.3 and Apache 2.0 series of HTTP servers.

OHS: HIGH LEVEL FEATURE OVERVIEW

A high level summary of all the features and components available within Oracle HTTP Server is given below:

The Web Server

- Based on Apache - OHS is based on the proven Apache web server. There are two OHS versions. One is based on Apache 2.0 code and the other on Apache 1.3 code.
- Security and Single Sign On- OHS supports SSL/TLS, basic authentication, and different levels of authorization. It also supports declarative model of single sign on.
- Virtual Hosts - Enables ISP to host several customers off of a single instance of web server, and configure them differently.
- WebDAV - Supports Oracle repository in addition to file based store for the content. Enables MS Office or other DAV clients to edit files on server.
- Proxy Server and URL Rewriting allows quick reorganization of site without any impact on externally visible URLs.
- Plug-in components now enable IIS, SunONE, and Apache web servers to be used to front-end Oracle Application Server 10*g* or OC4J. The OC4J Plug-In supports routing and load balancing over SSL directly to OC4J.
- mod_security provides an "application firewall" and prevents intrusions

against user application program vulnerabilities.

The 'Supporting' Application Server

- PLSQL Stored Procedures can now be accessed easily from a browser.
- PSP [PLSQL Server Pages] allows PLSQL to be used as a scripting language with HTML.
- Perl support is provided through mod_perl, which eliminates the need to restart the Perl interpreter each time.
- PHP support is provided when a user plugs mod_php into OHS
- Server Side Includes provides a standard mechanism to include headers/footers.
- C/C++ Support is now available through FastCGI, which keeps the processes alive, thus avoiding the startup cost.
- Dynamic Monitoring Service to monitor OHS or instrument applications

OHS: THE WEB SERVER

The Modular Architecture

The architecture of the web server is extremely modular. The core web server [for http protocol] is very small; all capabilities are implemented as modules that 'plug in,' and are invoked at the appropriate place during the request lifecycle [Figure 1].

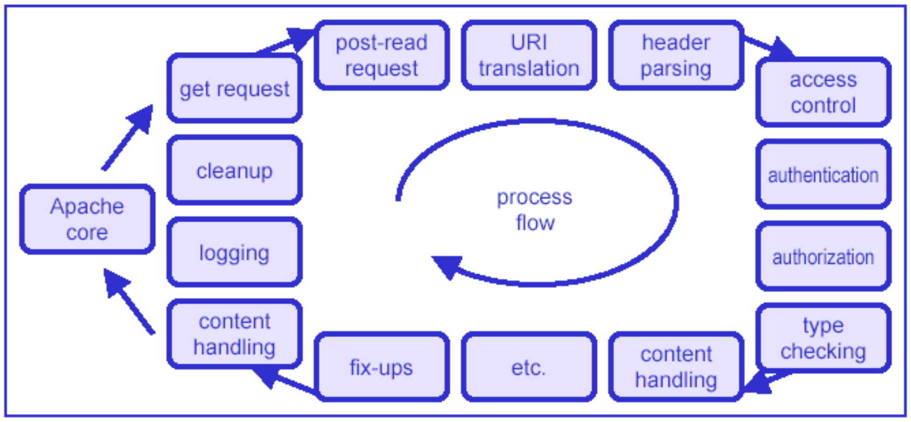


Figure 1: HTTP Request-Response Cycle

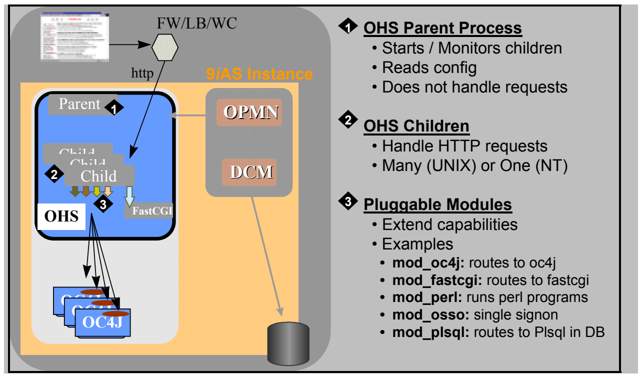
The modules (i.e. their API) are automatically invoked at the appropriate place in the request lifecycle.

Customers can add their own modules to the Oracle web server to supplement OHS functionality, as required. Oracle support may request that user modules be removed from the configuration if they feel it will interfere with troubleshooting.

This introduction to modules is useful when understanding configuration settings and the associated errors.

The Process Architecture

OHS, at startup, starts the parent process. This process loads the entire configuration and the associated modules, and spawns a pre-configured number of child processes. (On NT, it is a single child with multiple threads).


FW/LB/WC
http
9iAS Instance
Parent 1
Child
2
3
OHS
FastCGI
OC4J
OPMN
DCM
1 OHS Parent Process
• Starts / Monitors children
• Reads config
• Does not handle requests
2 OHS Children
• Handle HTTP requests
• Many (UNIX) or One (NT)
3 Pluggable Modules
• Extend capabilities
• Examples
• mod_oc4j: routes to oc4j
• mod_fastcgi: routes to fastcgi
• mod_perl: runs perl programs
• mod_osso: single signon
• mod_plsql: routes to Plsql in DB
Figure 2: Oracle HTTP Server (Web Server Component) Architecture

The parent never listens to any http request. Its sole job is to make sure the children are alive or spawned anew as the load may request it.

Each child process at any given time deals with a single HTTP request. The children determine who should take the next request based on a [user configurable] mutual exclusion (mutex) mechanism.

Configuring OHS

Configuration of OHS is done through the Oracle Enterprise Manager (OEM) thin client using a graphic user interface (GUI). OEM enforces configuration syntax and makes use of wizards to group commands in a task oriented manner. OEM minimizes the chance of typing errors or incorrect syntax causing difficult to detect configuration errors.

The precedence hierarchy of the different settings is shown in Figure 3. These multiple levels of settings are very useful when it comes to flexible deployments that ISPs desire.

Moreover, understanding them will help configure things correctly through OEM.

1. Server Settings, apply to the parent process - in general to a given OHS instance.
2. Default Host Settings, which apply to the default web site that OHS is hosting
3. Virtual Host Settings, which apply to the specific virtual host. (See later for more details on Virtual Hosts).
4. Directory, Location, or File Settings, which apply to all files in a specific directory, or all URLs of a certain kind, or, even to all URLs that map to a specific file.
5. Directory settings can also be applied by having a special file (.htaccess) contain those settings in that directory.

Figure 3: Precedence hierarchy

Security and Single Sign On

The Web Server component of Oracle HTTP Server provides the standard web server security features - encryption, authentication, and authorization.

Regarding encryption, OHS allows for standard SSLv3 as well as TLS support. In addition, it allows one to share server certificates with Oracle 10*g* DB and other Oracle products.

OHS does provide groups based authorization schemes for both static and dynamic applications. However, these groups have to be defined in flat file.

Regarding authentication, OHS has added significant enhancements for single sign on - enabling LDAP directory integration via the Oracle Application Server 10*g* Login Server. The integration is done via mod_osso. Below is a sample scenario of its operation:

1. Customer requests the page http://www.foo.com/subscribe/content
2. OHS recognizes that customer has not logged in, and redirects him to Login Server to login.
3. The customer logs in, and Login Server then redirects them to the page they were requesting. In the process, Login Server sets an encrypted cookie that only partner applications, Porgal and mod_osso can decrypt.
4. When OHS receives the request (again), it notices that the cookie has been set - implying the customer has been authenticated and passes the request through.
5. Then the customer requests the page www.bar.com/subscribe/content - a site hosted (possibly) on a different server, but going against the same Login Server.
6. OHS (at the new server) receives the request, notices the cookie is already set and lets the customer through without requesting him to login!

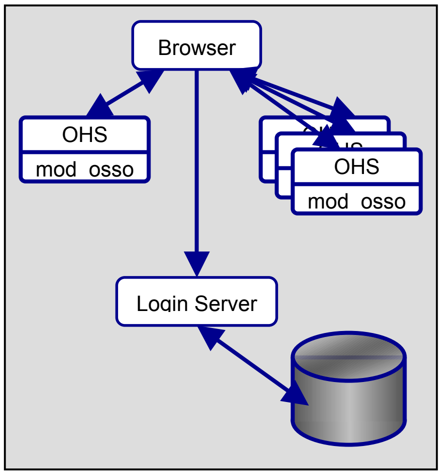


Figure 4: mod_osso and Single Sign On

Thus, the developer is able to deploy a single sign on based application without necessarily having to program anything, and the end user can access multiple sites without requiring multiple logins.

SSL Renegotiation Support

This feature allows an individual directory to be protected by different strength encryption. A common application is to have a directory that requires authentication of a client side certificate for access. An SSL request from the browser for this directory would be redirected to get the client certificate for authorization.

SSL Hardware Accelerator support for nCipher

This feature supports the nCipher hardware accelerator card. SSL requests can be configured to be routed to the Hardware accelerator for processing, significantly increasing the throughput of SSL requests.

Virtual Hosts

As an ISP, it is common to map several hostnames to a machine thus allowing several people to be hosted with limited infrastructure. Now, each of these different hostnames can be mapped to their own site through a configuration setup commonly referred to as Virtual Host.

Almost all configuration settings of the default host are available to the virtual host - i.e. almost no capability is lost in "virtual" hosting. Thus, hosting customers can be provided with control of their own configuration (it doesn't impact the overall server at all), or different configurations can be tested on the same server before making them production.

WebDAV Support

DAV stands for Distributed Authoring and Versioning. This is an IETF standard that uses HTTP as its base protocol. It enables clients (such as MS Office, Dreamweaver or other tools) to edit documents on a WebDAV enabled site. Most implementations of this protocol (ex. mod_dav from Apache organization) allow file system as a backing store on the server for these editable files.

OHS provides mod_OraDav to support the functionality and has enhanced this by making the Oracle database a possible backing store. In addition, it also provides an API that can be easily used to provide any other store as a backing store. Thus, MS Word can be used to directly edit and store files on an OHS powered site - and those office documents may even be stored in a database, without the end user knowing about it.

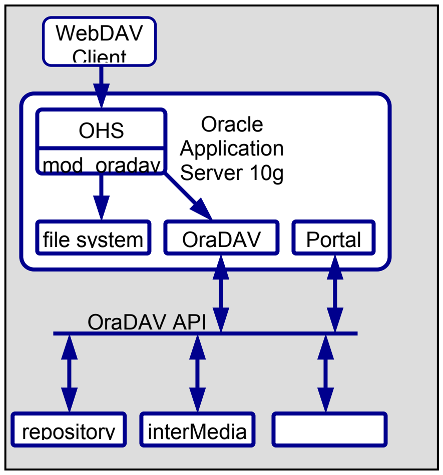


Figure 5: OraDav architecture

Proxy Server and URL Rewriting

Most production web sites are frequently modified. Sometimes this modifications result in changes to the directory structures. To accommodate these changes while allowing existing user bookmarks, etc. to function, Oracle HTTP Server provides a feature allowing URL rewriting so that old URLs will find the appropriate content.

OHS also supports forward and reverse proxy capabilities, thus making it easier to make content served by different servers to appear from one single server. This feature is also extensively used to segregate modem connections that may otherwise tie up processes from the primary application server.

Oracle Application Server 10*g* 3RD Party Web Server Integration

While OHS is quite powerful, many corporations may have a different web server standard. To help them leverage the powerful features of Oracle Application Server 10*g*, several plug-ins are available to integrate 3RD party web servers with Oracle Application Server 10*g*.

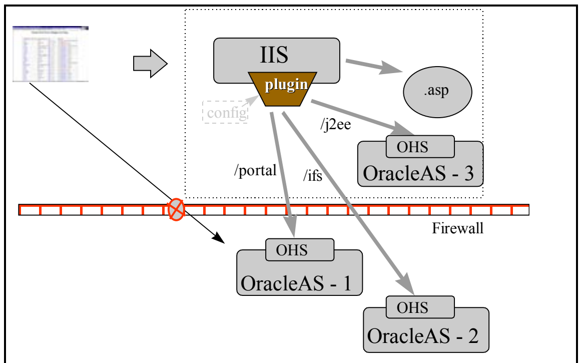


Figure 6: Proxy Plug-In Architecture

Proxy Plug-In

This component "plugs in" into IIS or SunONE servers and proxies the requests over HTTP to Oracle Application Server 10*g*. These web servers thus continue to work just as they did earlier, while still routing the Oracle Application Server 10*g* specific requests to Oracle Application Server 10*g*. This configuration allows the users of 3RD party servers to access all the features of the Oracle Application Server 10*g* server such as PLSQL, FastCGI etc.

The Proxy Plug-In can provide support (i.e. routing) to multiple Oracle Application Server 10*g* installations at the back-end. Figure 7 shows the Proxy Plug-In in use with Microsoft IIS.

OSSO Plug-In

Single sign-on functionality is also available to users of 3RD party servers via the OSSO Plug-In. This solution provides Oracle single sign-on functionality to users who have a reason to use IIS or SunONE as their Web Server, in place of OHS.

OC4J Plug-In

The routing and load balancing features of mod_oc4j will be available as a Plug-In to IIS and SunONE servers. This feature will provide the new load balancing options, OracleAS Port Tunneling and direct routing (including AJP over SSL) to OC4J. Use of this Plug-In eliminates the need to route servlet requests through OHS to reach OC4J. Requests go directly from the 3RD party server to the correct OC4J for servlet execution.

Mod_oc4j is also available for use with Apache web servers from the Apache Software Foundation. Mod_onsint is required in order to use mod_oc4j with Apache servers.

mod_security

mod_security is a plug-in that provides an "application firewall". This means that it can be configured to block attacks against application vulnerabilities such as cross-site scripting attacks or SQL injection attacks. (see for example http://www.securityfocus.com/infocus/1768). These attacks are considered application attacks because they exploit vulnerabilities in user written applications. It should be noted that most of these attacks exploit a lack of parameter checking by application developers and can be fixed by the addition of parameter checking. However, this can be time consuming and costly. Solutions that use mod_security may have better performance and may be much less costly to implement. Note that mod_security is also available on modsecurity.org at no cost if there are issues in a customer relying on mod_security for applications that might be portable between OHS and non-OHS Web servers.

IPv6

The version of the Internet Protocol that is in dominant use today is called IPv4 (Internet Protocol version 4). It has served the Internet community very well over the Internet's explosive growth. An issue with IPv4 is that its IP addresses are only 32 bits in length meaning that only about 4 billion addresses are available for all Internet end points.

An upgraded version of IPv4 has been developed and standardized. It is called IPv6 and people are starting to use this protocol. Its biggest virtue is that is has a much larger address space (128 bits) and therefore people are confident that IPv6 will not run out of address space.

At this time (January, 2005), only a very small fraction of the Internet supports IPv6 and therefore, generally speaking, one cannot directly access an IPv6 address device from most places on the Internet. Of course all devices, servers, etc. cannot migrate simultaneously to IPv4 address and as a result there will be a long transition phase where IPv4 and IPv6 addresses will both exist simultaneously.

Apache 2.0 has been transitioned to IPv6 but Apache 1.3 has not. Thus, people that need to accommodate IPv6 need to use the Apache 2.0 version of OHS. At first, it is expected that use of IPv6 by Apache 2.0 versions of OHS will be by using only the IPv4 to IPv6 proxy conversions features. This will allow, for example, IPv6 devices on an internal network to access Internet resources by having their IPv6 addresses be converted to IPv4 addresses before the requests are forwarded to the Internet at large.

Since IPv4 to IPv6 proxy is expected to be the dominant use of OHS when accommodating IPv6 addresses, the OHS version based on Apache 2.0 is only being released in a standalone configuration. In the future, as it becomes practical to expose IPv6 addresses to the Internet at large, the OHS version based on Apache 2.0 will be provided with integrated installation options as well as standalone options.

OHS: THE 'SUPPORTING' APPLICATION SERVER

Dynamic Content with PLSQL - PSP and mod_plsql

Similar in concept to the Java Server Pages, the PLSQL server pages module allows PLSQL to be used as the scripting language within an HTML page. The page gets translated into a stored procedure, and the mechanism described in Fig. 8 then sends the output to the browser.

OHS includes a module (mod_plsql) that enables making requests to database stored procedures from the browser. This is one of the most popular features. In addition, it also provides performance improvement due to the disk-based cache. All PLSQL processes continue to run in the database.

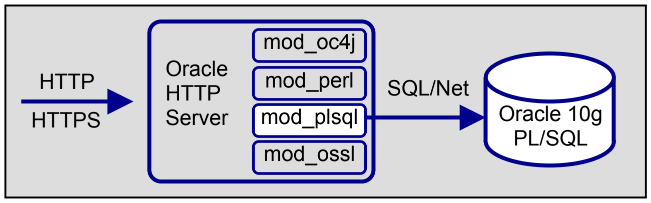


Figure 7: PL/SQL Gateway Architecture

Here is a flow of a request to mod_plsql:

1. OHS receives the request. Depending on the registered modules, it determines which module should handle the request, in this case mod_plsql.
2. mod_plsql connects to the database, prepares the call parameters, and invokes the PLSQL procedure in the database.
3. The PLSQL procedure generates an HTML page using data and stored procedures accessed from the database. The product supplies packages that can be installed in the database to make this task easier.
4. The response is returned to mod_plsql, which sends it back to the browser.

Mod_plsql runs within the OHS child process. Thus, each child process owns the connection to the database and keeps it alive. This connection is not shared across OHS child processes - thus the number of connections for a large site will be dependent on the number of child processes (and other configuration settings).

Dynamic Content with Perl and Server Side Includes

Server Side Include provides an easy way of adding some dynamic, or uniform static content, across all the site's pages. It is typically used for header / footer information. Oracle HTTP Server supports special directives to enable these only for certain types of files or for a given virtual host.

Perl is one of the most common ways scripts and CGI programs are developed for the web. However, the Perl interpreter is large and starting and stopping it is time consuming. OHS optimizes this execution by keeping the Perl interpreter always running and in memory. It also enables extending the web server functionality (see Figure 1) by adding new Perl modules that can process a web request.

Dynamic Content with CGI and FastCGI: Java / C / C++

The ability to write CGI programs is common in almost all web servers. However, OHS adds the ability for these programs to stay alive beyond the request lifecycle. Thus, future requests do not incur the overhead of restarting the CGI program (and the associated database connections!). This results in a significant performance boost.

The framework that enables this is referred to as FastCGI. In addition to C and C++, it can also run Java programs (although with advent of J2EE Java based CGI is not common).

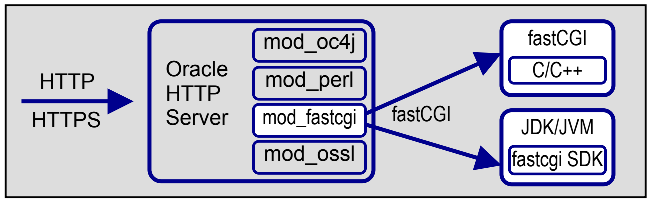


Figure 9: FastCGI Architecture

When a client request comes in, the Web server opens a connection to the FastCGI process that then invokes the application's entry point and sends the result back. The FastCGI process can be on the same machine or different machines, providing a choice of deployment options. Configuration based multiple processes are automatically provided for single threaded FastCGI applications; multi-threaded FastCGI applications are served through a single process.

FastCGI can play multiple roles - that of responder, where it produces the response to an HTTP request, or of authorizer, where it accepts or declines the authorization request to the web server.

Mod_oc4j

Mod_oc4j is a module that plugs into Oracle HTTP Server (OHS) and routes to all OC4J instances. It works in conjunction with DCM and OPMN to keep its routing table updated so that it load balances across only live OC4J processes in an OC4J instance. This module uses AJP to communicate to the OC4J instances locally or to any OC4J instance that is part of the farm. Mod_oc4j supports AJP over SSL for those cases where secure communication to remote machines is a requirement.

Port Tunneling

The Application Server 10*g* Port Tunneling feature reduces the number_of ports required to communicate to multiple OC4J processes to ONE. The diagram below shows an Oracle Application Server 10*g* configuration using OracleAS Port Tunneling. The process acts as a communications concentrator for connections between OHS and OC4J's.

OHS does not connect directly to OC4J. Rather, OHS connects to an OracleAS Port Tunnel process. The port tunnel then forwards communication on to OC4J. Each OracleAS Port Tunnel routes requests to multiple OC4J's. By doing this concentration of connections the customer is only required to open one port per OracleAS Port Tunnel process on the internal firewall rather than one port per OC4J container.

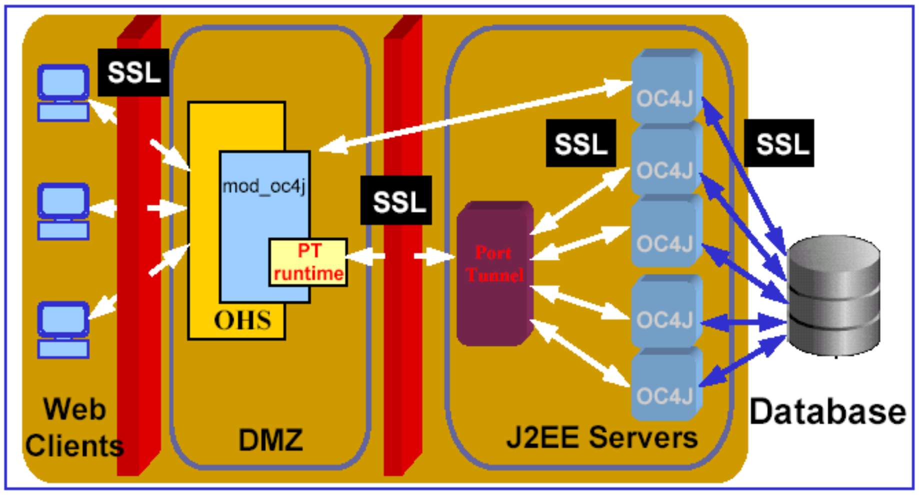


Figure 11: OracleAS Port Tunnel

The communication between OHS and OracleAS Port Tunnel can be encrypted using SSL. Authentication will be done when these connections are established using SSL Client Certificates.

SUMMARY

Oracle HTTP Server is first, a web server. Based on the proven Apache technology, it provides all the necessary features for a full functioned enterprise site. It provides support for WebDAV, with Oracle Database 10*g* as a backing store. Plug-In components provide Oracle Application Server 10*g* functions such as Single sign-on and mod_oc4j load balancing, to be used with other web servers such as IIS, SunONE, and Apache.

Second, it contains many application server components - it provides an ability to write dynamic web applications in several languages - PLSQL, Perl, PHP, Server Side Include, C/C++ etc.

Third, it provides the clustering framework for Oracle Application Server 10*g* - The new infrastructure enables quick *distributed deployment* of J2EE applications. It also *monitors* all processes in a *cluster for failure* and transparently updates the routing table for optimal *load balancing* and minimum runtime impact on requests.

Oracle Application Server 10*g* - Technical Overview of Web Server and Modules
January 2005
Authors: John Lang, Bruce Lowenthal
Contributing Authors:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com