Oracle Application Server 10*g* Release 3 (10.1.3.1.0) Overview of Oracle HTTP Server

An Oracle White Paper
October 2006

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

EXECUTIVE OVERVIEW

Oracle Application Server 10*g* 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.

The Oracle HTTP Server (OHS) provides key infrastructure for serving the Internet's HTTP protocol. OHS is used to return responses for both process-to-process and human generated requests from browsers. Key aspects of OHS are its technology, its serving of both static and dynamic content and its integration with both Oracle and non-Oracle products.

Technology - OHS is based on the proven, open source technology of both Apache 1.3 and Apache 2.0. OHS versions based on Apache 2.0 now provide the ability to accommodate the newest version of the Internet Protocol, IPv6. In addition OHS now provides, via the open source product mod_security.

Static and Dynamic Content - OHS serves static content directly or via standard interfaces such as WebDAV standard. Great flexibility is provided in dynamic content generation and many languages, such as Java, C/C++, Perl, PHP and PLSQL are provided for content generation.

Integration - While OHS has standalone deployment options it can also be deployed in a highly integrated manner with Oracle clustering, monitoring, Single Sign On or Web Caching technology. In addition, Oracle offers plug-ins (Proxy, OC4J, and OSSO) for integration of the Oracle Application Server with non-Oracle HTTP Servers.

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

OHS: HIGH LEVEL FEATURE OVERVIEW

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

- 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 10g or Oracle Application Server Container for J2EE (OC4J).

- mod_security provides an "application firewall" and prevents intrusions against user application program vulnerabilities.
- PLSQL Stored Procedures can now be accessed easily from a browser.
- PSP [PL/SQL Server Pages] allows PL/SQL 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 via mod_php module of 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.

ORACLE HTTP SERVER ARCHITECTURE

Based on Apache – HTTP v1.1 Support

OHS is based on the open source technology of both Apache 1.3 and Apache 2.0. With such a proven code base, Oracle HTTP Server provides Oracle Application Server customers with the stability, flexibility, and scalability required of a Web server.

Apache 2.0 Support

Along with Apache 1.3, Oracle HTTP Server is now based on Apache 2.0 as well. The version of OHS based on Apache 2.0 is provided as a standalone component and provides all modules that are supported in OHS based on Apache 1.3 except mod_oradav, mod_plsql and mod_dms.

Main benefits of Apache 2.0 when compared to Apache 1.3 are faster execution on Windows OS machines and its ability to accommodate IPv6. Note that Apache 2.0 can proxy from IPv6 addresses to IPv4 addresses.

The Modular Architecture

The architecture of Apache 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]. 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 OHS 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.

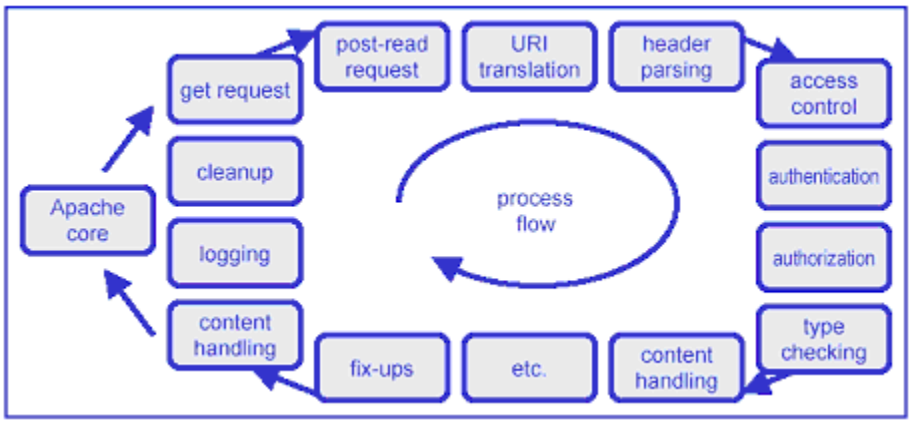

Figure 1: HTTP Request-Response Cycle

Oracle HTTP Server Modules

Modules extend the basic functionality of the Web server, and support integration between OHS and other Oracle Application Server components. Many of the standard Apache modules are included with OHS. Oracle also includes several modules that are specific to Oracle Application Server components. Table 1 below identifies the key Oracle modules shipped in OHS based on Apache 1.3 and Table 2 identifies the key modules from Apache Software Foundation that are shipped with OHS based on Apache 1.3.

Table 1 Key Oracle Modules in OHS based on Apache 1.3

Module	Version	Description
mod_oc4j	10.1.3.1.0	Provides configurable and intelligent routing for incoming requests from OHS to Oracle Application Server Container for J2EE (OC4J) using AJP 1.3 protocol.
mod_oradav	10.1.3.1.0	This Oracle module is an extended implementation of mod_dav. It enables WebDAV clients to connect to an Oracle database, read and write content, and query and lock documents in various schemas.
mod_ossl	10.1.3.1.0	Enables strong cryptography for Oracle HTTP Server. It is based on the Oracle implementation of SSL, which supports SSL version 3, and is based on Certicom and RSA Security technology.
mod_osso	10.1.3.1.0	Enables single sign-on for Oracle HTTP Server. It examines incoming requests and determines whether the resource requested is protected, and if so, retrieves the OHS cookie for you.
mod_plsql	10.1.3.1.0	Connects Oracle HTTP Server to an Oracle database, enabling you to create Web applications using Oracle stored procedures.

Table 2 Key Non-Oracle Modules in OHS based on Apache 1.3

Module	Version	Description
mod_cgi	1.3.31	Enables the Oracle HTTP Server to run CGI scripts.
mod_fastcgi	2.4.2	Supports the FastCGI protocol, which enables you to maintain a pool of running servers for CGI applications, thereby eliminating start-up and initialization overhead.
mod_perl	1.99	Embeds the Perl interpreter into the OHS. This eliminates start-up overhead and enables you to write modules in Perl. Oracle Application Server uses Perl version 5.8.3.
mod_php	5.1.2	PHP is a widely used, general-purpose, client-side scripting language. On Oracle HTTP Server, PHP support is provided through mod_php and has Oracle database support enabled.
mod_rewrite	1.3.31	Provides a way to manipulate URLs. A rewriting engine based on a regular-expression parser is used to rewrite requested URLs.
mod_security	1.9.2	Increases Web application security by protecting Web application from known and unknown attacks.

Note: This is not the complete list of modules shipped with Oracle HTTP Server. Some key Oracle and non-Oracle provided modules are listed here. For complete list refer *Oracle HTTP Server Administrator's Guide 10g Release 3 (10.1.3.1.0).*

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 [Figure 2]. (On Windows, it is a single child with multiple threads).

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] mutex mechanism.

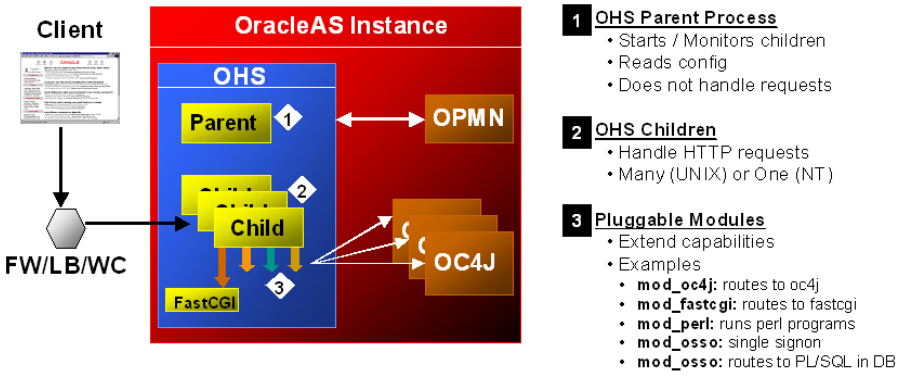

Figure 2: Oracle HTTP Server (Web Server Component) Architecture

ORACLE HTTP SERVER FEATURES

Security

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

Encryption with SSL

Secure Sockets Layer is required to run any Web site securely. Oracle HTTP Server supports SSL encryption based on industry standard, patented, algorithms. The SSL works seamlessly with both Internet Explorer and Netscape browsers. In addition, the infrastructure has been upgraded to share the same wallet information as the database users. Features include:

- **SSL HW Acceleration Support:** SSL encryption is slower when done in software. Dedicated hardware support for this purpose is now supported. OHS supports hardware security modules that use APIs, which conform to the RSA Security, Inc., PKCS #11 specification. Typically, these hardware devices are used to securely store and manage private keys in tokens or smart cards, or to accelerate cryptographic processing.
- **Variable Security per Directory:** 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.
- **OHS to OC4J SSL Support:** OHS and OC4J can communicate using AJP protocol over SSL. Previously, OHS and OC4J used the AJP 1.3 protocol unencrypted, without support for authentication. OHS has been modified to extend support to the AJP 1.3 protocol over SSL providing both encryption and authentication.

Single Sign On

Oracle HTTP Server supports the standard basic authentication features of HTTP servers. The source for the username and password used here is a flat file (with encrypted passwords). In addition, a module, mod_osso, is included to support

single sign on across sites and across applications. This provides for a much better end user experience (they have to login only once), and a much easier development cycle (most of the security is declarative).

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, and here is a sample scenario to understand it better:

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 - such as mod_osso - can decrypt.
4. When OHS receives the request (again), it notices the right cookie has been set - implying the customer has been authenticated - it lets them through.
5. Now 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.

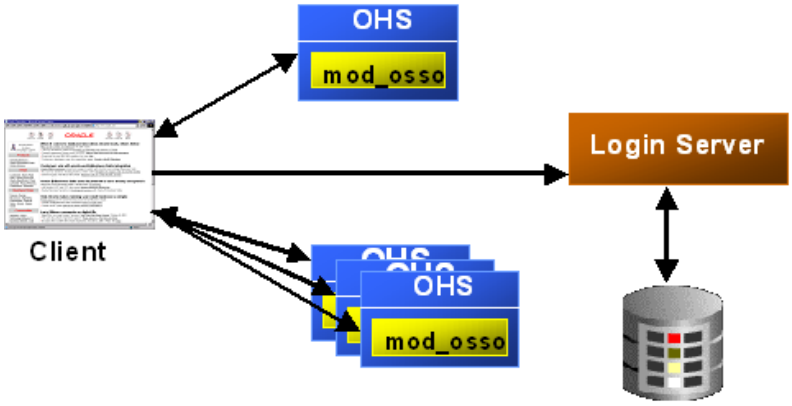

Figure 3: mod_osso and Single Sign On

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.

Application Firewall - mod_security

Oracle HTTP Server includes a module called mod_security 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. 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.

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.

Distributed Authoring and Versioning Support

WebDAV, an IETF standard, is an HTTP based protocol that allows DAV enabled clients, such as MS Office, Windows Explorer, to edit files on a server. The Apache Software Foundation provides a module, mod_oradav, which provides support for file-based storage on the server. In addition to providing this functionality, Oracle HTTP Server enables the server side store to be a database or other repository.

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.

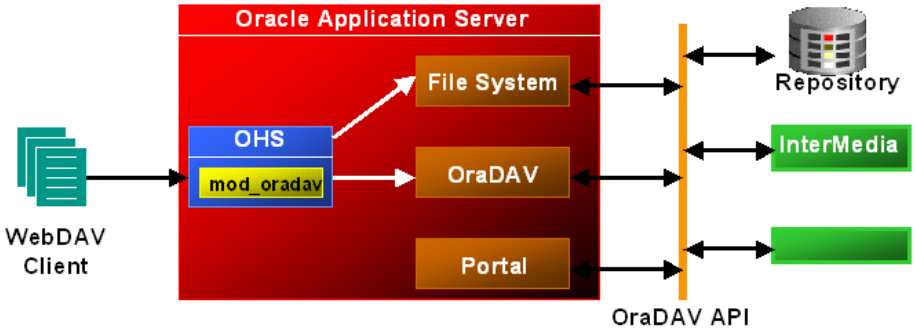

Figure 4: OraDav architecture

Proxy Server and URL Rewriting

The directory structure and URL's of active websites often change. Oracle HTTP Server makes it easy to accommodate these changes by including an engine to support URL rewriting (so that the end users do not have to change their bookmarks). It 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.

Load Balancing - mod_oc4j

Mod_oc4j is the load balancer for the requests going to the Oracle Container for J2EE (OC4J) Instances in Oracle Application Server. It is an OHS module that provides routing between OHS and OC4J. The Oracle Process Manager and Notification Server (OPMN) component of Oracle Application Server keeps mod_oc4j aware of the status of different OC4J processes - thus, mod_oc4j routes only to the processes that are up and running. Mod_oc4j also understands the concepts of Oracle Application Server Cluster and OC4J groups, and routes accordingly to provide as much transparent failover as possible. Mod_oc4j supports AJP over SSL for those cases where secure communication to remote machines is a requirement.

Load Balancing Algorithms

Mod_oc4j provides three distinct kinds of routing: (a) round robin, (b) random and (c) metric based. These load balancing/routing algorithms also have a flavor - affinity based. In this mode (it is the default mode), these algorithms will always route to the local node, except in cases when no process is available on the local node. The random and round robin algorithms have an extra flavor - weight based. In case of weight based, mod_oc4j distributes requests according to the routing weight configured for each host.

Dynamic Discovery

With new dynamic discovery functionality mod_oc4j dynamically builds its routing table, including the list of application contexts and OC4J Instances available. Dynamic discovery helps in the following two areas:

- **Routing Relationship:** The routing relationships between OHSs and OC4Js are established dynamically. All OHSs listen for notifications from OC4Js that they have a routing relationship with. Therefore, each of these OHSs discovers the OC4Js it needs to route to instead of being statically defined in the mod_oc4j configuration file.
- **Mount Point Discovery:** When an application is deployed to OC4J it announces itself via ONS messages that are received by OHS. The routing table in mod_oc4j is then updated. No configuration changes (e.g. mount points) are required in mod_oc4j.conf and processes do not have to be restarted.

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*.

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 PL/SQL, 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 5 shows the Proxy Plug-In in use with Microsoft IIS.

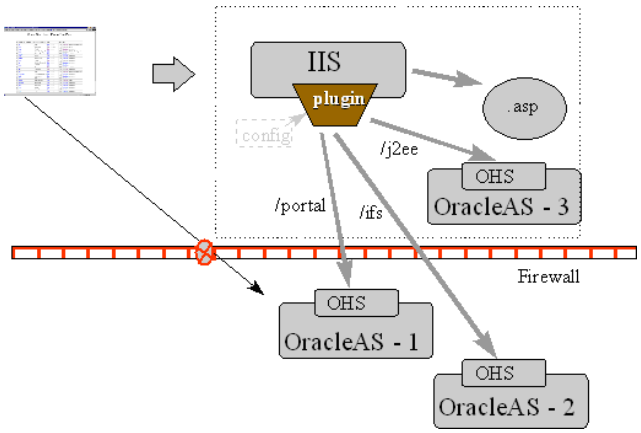

Figure 5: Proxy Plug-In Architecture

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 are available as a Plug-In to IIS, SunONE and Apache servers (mod_onsint is required in order to use mod_oc4j with Apache servers). This plug-in provides the load balancing options, Oracle Application Server Port Tunneling and direct routing (including AJP over SSL) to Oracle Container for J2EE (OC4J). Use of this Plug-In eliminates the need to route requests through OHS to reach OC4J. Requests go directly from the 3RD party server to the correct OC4J for execution.

IPv6 Support

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 it has a

much larger address space (128 bits) and therefore people are confident that IPv6 will not run out of address space.

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 OHS based on Apache 2.0.

Dynamic Content with PL/SQL - PSP and mod_plsql

Similar in concept to the Java Server Pages, the PL/SQL Server Pages (PSP) module allows PL/SQL 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. 6 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 PL/SQL processes continue to run in the database.

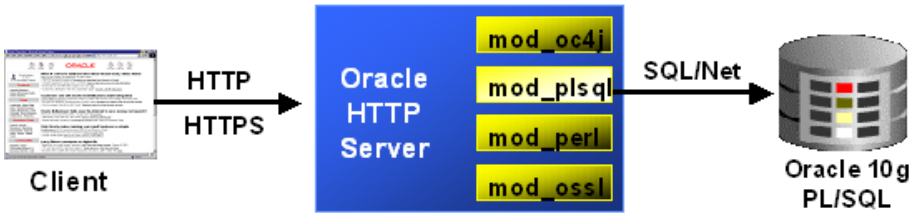

Figure 6: PL/SQL Gateway Architecture

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 PL/SQL procedure in the database.
3. The PL/SQL 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 [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).

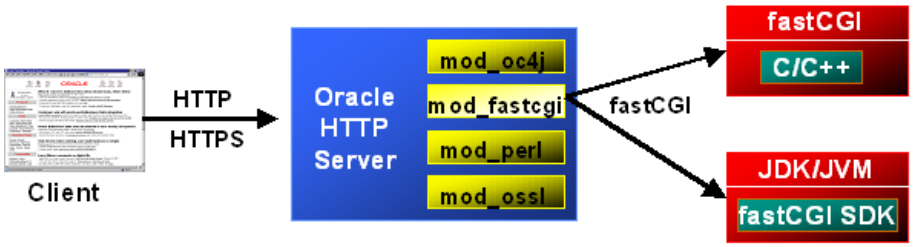

Figure 7: FastCGI Architecture

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.

Dynamic Monitoring Service

Dynamic Monitoring Services (DMS) metrics give runtime performance statistics for both Oracle HTTP Server and OC4J processes. As applications run, DMS collects detailed performance statistics. This data enables you to monitor the duration of important request processing phases and status information. With this information, you can locate performance bottlenecks and tune the application server to maximize throughput and minimize response time.

Oracle Process Manager and Notification Server

Oracle Application Server provides a highly availability infrastructure integration with Oracle Process Manager and Notification Server (OPMN), for process management, death detection, and failover for Oracle HTTP Server processes.

Port Tunneling

Oracle 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 Oracle Application Server 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 Oracle Application Server Port Tunnel process. The port tunnel then forwards communication on to OC4J. Each Oracle Application Server Port Tunnel routes requests to multiple OC4J's. By doing this concentration of connections the customer is only required to open one port per Oracle Application Server Port Tunnel process on the internal firewall rather than one port per OC4J container.

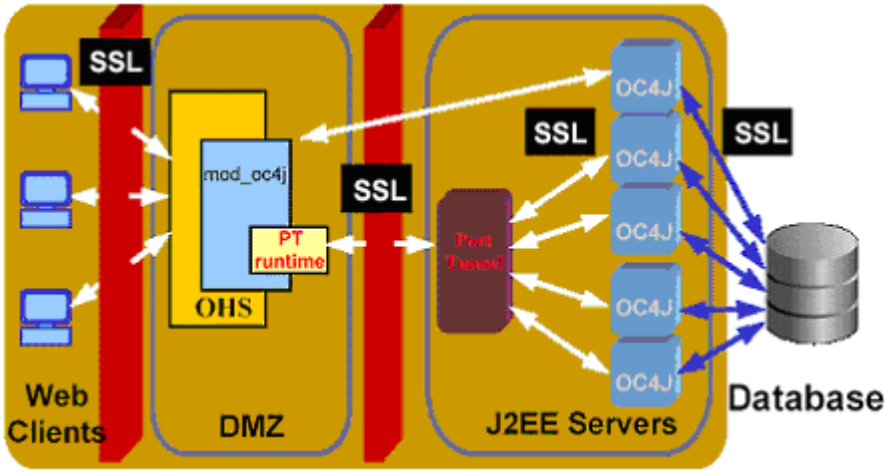

Figure 8: Oracle Application Server Port Tunnel

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

CUSTOMER PROOFPOINTS

"Oracle Application Server is much faster than our previous Java application server. With the move to Oracle Application Server, we now operate on a single, unified platform built on Oracle from the ground up"

—Marty Boos, CIO, Digital River

Digital River

Digital River, a leading global e-commerce outsource provider founded in 1994, offers its clients the ability to cut costs and grow their businesses by using its complete e-commerce systems and services. Digital River relies on Oracle Application Server to host e-commerce operations for thousands of customers, processing up to 40,000 orders and generating 1.5 million dynamic Web pages daily. Among others, clients include Symantec, 3M, Novell, Autodesk, and Staples.com.

Digital River replaced BEA WebLogic and standardized its next-generation hosted e-commerce platform on Oracle Application Server and Oracle Database. "Oracle Application Server is much faster than our previous Java application server. With the move to Oracle Application Server, we now operate on a single, unified platform built on Oracle from the ground up," says CIO Marty Boos. At the same time, switching from BEA WebLogic to Oracle Application Server saved Digital

River 44% on the cost of maintenance and support, and 69% on the cost of future purchases.

The most immediate benefit of the new platform has been the remarkable performance improvement yielded by the Oracle's revolutionary web cache technology. The average page load time shrank two- to fivefold and the average number of SQL database calls to generate a dynamic Web page decreased 100-fold. Boos raves, "With the Web-caching capabilities of Oracle Application Server, we find that we can now achieve all the scalability we need in the middle tier. The bottom line is that we save millions of dollars' worth of server hardware."

Poste Italiane (www.poste.it)

Poste Italiane Group is a multibusiness conglomerate based in Italy that offers core postal and mail delivery services, as well as communication, logistic, and financial services, all over Italy. The Poste Italiane Group includes the following major subsidiaries: the SDA Group, which provides express mail and logistics; Mototaxi city bike couriers; Postecom, managing Internet services; PosteVita and BancoPosta Fondi, offering life insurance and investment solution; and Postel, the European leader for hybrid electronic mail and document processing. Poste Italiane manages complex logistics and communications business processes on an outsourced basis for several industries, including retail, commercial business, and government and public administration segments. It also manages all of Italy's more than 14,000 post offices.

Poste Italiane's Pension Payment System was one of the world's largest and highest transaction-rate mainframe applications, capable of handling 1.5 million pension transactions per day for more than 6 million government employees. Faced with the escalation of maintenance costs and the inflexibility of the mainframe, Poste Italiane migrated the application from the mainframe to the world's largest, most complex, and highest transaction-rate J2EE application on the AIX operating system. Built on servlets, stateless session beans, and entity beans with container-managed persistence, the system was deployed to two Oracle Application Server instances accessing an Oracle Database Real Application Cluster (RAC).

Both the application server instances and the database server are replicated to a remote site for disaster recovery purposes.

With Oracle Application Server, Poste Italiane was able to downsize from their old, expensive, and hard-to-maintain mainframes, while reengineering their business logic to provide wider and simpler system access across the company's many different groups, departments, and employees. Oracle Application Server 10g also provided Poste Italiana with the industry's best performance, scalability, and availability features required to power this highly mission-critical system.

SUMMARY

Oracle HTTP Server is 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, mod_oc4j load balancing and AJP over SSL, to be used with other web servers such as IIS, SunONE, and Apache.

It provides an ability to write dynamic web applications in several languages - PLSQL, Perl, PHP, Server Side Include, C/C++ etc.

With new dynamic discovery feature application mount points and routing relationships are discovered dynamically in the entire cluster. 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 FUSION MIDDLEWARE

Oracle Application Server 10*g*: Overview of Oracle HTTP Server
October 2006
Author: Shail Goel
Contributing Authors: John Lang

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
oracle.com