TABLE OF CONTENTS
Table of Contents ................................................................. 2
Introduction ........................................................................... 3
  What is my.oracle.com ? .................................................... 3
  Goals for My Oracle ....................................................... 4
  My Oracle Focus ............................................................ 4
Infrastructure Overview ....................................................... 8
Behind the Oracle Firewall .................................................... 9
  OracleAS Web Cache Cluster ......................................... 9
  Portal Middle Tier .......................................................... 10
  Portal Database Server .................................................. 12
Login.Oracle.Com Architecture ........................................... 13
  SSO Server Middle-Tier .................................................. 13
  SSO Server Database .................................................... 14
Provider.Oracle.Com Architecture ....................................... 15
Oracle Intranet ..................................................................... 17
  Reliability and Availability ............................................... 17
Configuration Settings ......................................................... 18
Performance Logging Analysis ............................................. 20
  Daily Unique Visitors Per Day .......................................... 20
  Hourly Unique Visitors .................................................. 21
  Page Views Per Day ....................................................... 22
  Page Views Per Hour ...................................................... 23
  Top Ten Pages on My Oracle ............................................ 23
  Omniture and Portal Performance logs ......................... 24
Conclusion ............................................................................. 25
INTRODUCTION

Today Oracle is realizing increased productivity through its enterprise portal, My Oracle. My Oracle is specifically designed to be the single source of interaction with Oracle’s corporate information and the focal point for employees conducting their day-to-day business. My Oracle provides Oracle employees with access to the applications and tools they need to do their job. It also acts as a single source of truth to the information they need.

My Oracle is based on Oracle Application Server (OracleAS) Portal. OracleAS Portal combines a rich, declarative environment for creating a portal web interface, publishing and managing information, accessing dynamic data, and customizing the portal experience along with an extensible framework for any web-based technology including J2EE-based application access and Web Services.

This paper is an updated version of the ‘Portal Implementation Case Study: My Oracle’ published in May 2003. This paper is intended for readers who are interested in deploying an enterprise class portal. This paper covers the major technical and infrastructure components that have been implemented for My Oracle.

What is my.oracle.com?

My Oracle (http://my.oracle.com) is an Oracle hosted service that provides a view of all the relevant information and applications that any Oracle user might need. Oracle employees also use My Oracle as their online workspace (Figure 1). It provides a single source of interaction with Oracle corporation’s information, applications, and resources. It is the focal point for any Oracle employee conducting their day-to-day business. My Oracle has more than 43,000 Oracle employee users. Employees obtain key business information for starting their workday. It has a growing number of business applications for all types of users. The My Oracle infrastructure can support up to 60,000 logins per hour, or 1,440,000 logins per day. At the time of writing 500,000 non-employee users were also registered to use My Oracle.

Figure 1: Employee View of My Oracle's Default Home Page
Goals for My Oracle

- Provide employees with a single point of access to productivity tools and information from Oracle and its partners, tailored to their specific needs
- Provide a vehicle for employees – especially sales and marketing focused – to find the latest information on Oracle’s products and services
- Disseminate important internal and external information to employees, and increase information usage, at a lower cost
- Provide a better portal product to our customers, through developing internal requirements that are fed back into Oracle’s development teams
- Showcase OracleAS and validate customer experience with our products

My Oracle Focus

The focus of My Oracle is to support Oracle employees:

- Focus on business content and applications rather than general purpose information
- Improved integration through Single sign-on (SSO) to internal applications that improve employee productivity
- Continued focus on reliable and scalable architecture
- Provide an improved My Oracle at a lower cost to the company
- Allows easy development, testing, and deployment of applications

Focus on business content and applications rather than general-purpose information

The focus of My Oracle is to improve employee productivity through providing them with the latest information on Oracle’s products and services, as well as providing intuitive means to access applications that employees need on a daily basis (Figure 2).

Marketing and sales content is organized using a taxonomy developed in cooperation with the content owners (Figure 3). There are 4 high level page-groups and sub-pages are within these page groups. Items are organized using a “Bill of Materials” that consists of attributes, categories, perspectives and item types.
The refocusing of My Oracle is related to direct feedback from Oracle employees. A survey of the My Oracle user community (see Table 1) showed that additional searching and single sign on capabilities, and access to productivity applications would be of great value. With these results in mind, new searching capabilities and additional SSO applications have been added.
Table 1: Top six My Oracle Features Requested by Oracle Employees

<table>
<thead>
<tr>
<th>Top 6 requested features</th>
<th>(%) Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Sign-on to Applications &amp; Content sites</td>
<td>20%</td>
</tr>
<tr>
<td>Oracle Intranet Search</td>
<td>18%</td>
</tr>
<tr>
<td>Oracle Employee Search</td>
<td>18%</td>
</tr>
<tr>
<td>Access to My Calendar</td>
<td>16%</td>
</tr>
<tr>
<td>Oracle Employee Search</td>
<td>16%</td>
</tr>
<tr>
<td>Access to My e-mail</td>
<td>12%</td>
</tr>
</tbody>
</table>

**Improved integration through Single sign-on (SSO) to internal applications that improve employee productivity.**

Application and content integration within My Oracle is being achieved in two steps. The first step involves providing SSO links to the internal sites (e.g. Oracle employee’s e-mail and calendar, Self-Service 11i Applications and Oracle’s Issue Tracking system). The second step involves providing portlets that display a summary from their respective applications. For example, an Oracle Collaboration Suite enabled Calendar portlet will soon be offered on My Oracle (Figure 4):

Single sign-on (SSO) is the act of logging in once, then getting access to multiple servers or applications using a single authentication credential. Users employ a single enterprise username and password to connect to multiple databases applications. Users with a single password tend to use better, harder-to-break passwords because they have only one to remember. Business benefits of single sign-on may include a reduction in customer Support calls due to lost or forgotten passwords and increased security. Please see the ‘Oracle Application Server Security and Single Sign-On’ whitepaper to learn more about SSO.

**Continued focus on reliable and scalable architecture**

My Oracle continues to be built using the portal framework of Oracle Application Server. OracleAS Portal provides a scalable architecture for dynamically generating customized pages that can meet the performance requirements of enterprise class...
portals. The deployment model supports both single-box and multi-tier configurations on a broad set of hardware platforms and operating systems.

**Provide an improved My Oracle at a lower cost to the company**

Five Oracle employees on a part-time basis manage My Oracle. Focus on employee content and improvements in Oracle's Portal technology in part helped to reduce the number of people that are required to offer My Oracle.

**Allows easy development, testing, and deployment of applications**

The modular, portlet-based architecture of OracleAS Portal makes it easy for creators of My Oracle services to develop, test, and deploy portal content and services. In order to group a set of related portlets together, OracleAS Portal has a notion of a provider. Providers may be used to arrange portlets into meaningful groups. For example, portlets that access marketing collateral may be grouped under a marketing provider.

From the user's perspective, portlets summarize, promote, or provide basic access to an information resource. The information resources themselves can take on many forms and be used for many purposes. Portlet output can be either in HTML, or XML/XSL. Please see the 'Oracle Application Server Portal 10g (9.0.4) Technical Overview' on OTN Portal Center to learn more about Providers and Portlets.

**Technical Overview**

There are multiple portals hosted within the same architectural layout including portal.oracle.com, portalcenter.oracle.com and oraclepartnernetwork.oracle.com. The infrastructure described here has been designed to handle a very large amount of traffic. Each of these services have quite large user populations.
INFRASTRUCTURE OVERVIEW

The current architecture as pictured in Figure 5 shows three distinct elements to the my.oracle.com setup. There is my.oracle.com, login.oracle.com, and provider.oracle.com

- My Oracle is the physical portal infrastructure which handles the requests for pages and passes requests for logins and portlets to the other components when needed.
- login.oracle.com is the centralized SSO Server used by My Oracle and other services within the Oracle infrastructure. login.oracle.com provides a common login service for all Oracle websites.
- provider.oracle.com is a separate entity for dealing with requests for content from providers registered within the My Oracle repository.

The SSO solution for My Oracle runs on separate hardware to scale the solution for the following reasons:

- Security
- Administration
- Tuning
BEHIND THE ORACLE FIREWALL

The following sections will step into each of the major component areas in detail, starting with the load balancing solution and working through each of the logical layers of the components.

The Load Balanced Front-End

All traffic from the public internet passes through the Oracle firewall and is received and responded to by a BIG-IP load balancing router (LBR) from F5 Networks Inc. (www.f5.com) to load-balance requests among the Web servers located in the middle-tier of the Demilitarized Zone (DMZ). To ensure high availability and fault tolerance, a second BIG-IP router is in place and active should the primary router fail.

In computer networks, a DMZ is a computer host or small network inserted as a "neutral zone" between a company's private network and the outside public network. It prevents outside users from getting direct access to a server that has company data. A DMZ is an optional and more secure approach to a single firewall and effectively acts as a proxy server as well.

In a typical DMZ configuration for a small company, a separate computer (or host in network terms) receives requests from users within the private network for access to Web sites or other companies accessible on the public network. The DMZ host then initiates sessions for these requests on the public network. However, the DMZ host is not able to initiate a session back into the private network. It can only forward packets that have already been requested.

The LBR is configured to add its own cookie to any request for content, to ensure that a subsequent request for content is directed to the same middle-tier that served the previous request. This is called server affinity, and it is necessary, because we maintain four separate file caches (see the section later on the Portal Cache) on each of the middle-tier boxes and routing a second and subsequent request to a different box could result in previously fetched content being re-fetched, leading to performance loss.

OracleAS Web Cache Cluster

The next layer in the My Oracle request route is OracleAS Web Cache. There are two OracleAS Web Cache machines. Each machine is installed with two OracleAS Web Cache 10g instances. Each OracleAS Web Cache is configured to use dynamic caching directives.

OracleAS Web Cache is a fully integrated, intelligent cache that enables the highest level of performance by minimizing unnecessary re-generation of portal pages and portlet content. OracleAS Portal takes full advantage of OracleAS Web Cache's in-memory caching technology. OracleAS Web Cache combines caching, compression, and assembly technology to accelerate the delivery of both static and dynamically generated portal content. It also provides back-end Web server load balancing, failover, and surge protection features for improved performance and availability.

The portal repository issues dynamic caching and invalidation messages to the OracleAS Web Cache clusters to ensure that the most up-to-date version of the content is stored in OracleAS Web Cache.

### OracleAS Web Cache Machine Specifications

<table>
<thead>
<tr>
<th>Model</th>
<th>Dell 1550 Rackmount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CPUs</td>
<td>2</td>
</tr>
<tr>
<td>CPU Speed</td>
<td>1.2GHz</td>
</tr>
<tr>
<td>Operating System</td>
<td>Red Hat Enterprise Linux,</td>
</tr>
<tr>
<td></td>
<td>version 3.4</td>
</tr>
</tbody>
</table>
Portal Middle Tier

The Middle-tier servers contain the components that handle the page generation and content from local and remote providers. There are four machines configured and set up identically. The portal middle-tier consists of these elements:

- Oracle HTTP Server
- Mod_PLSQL 9.0.4
- OC4J 9.0.4
- Parallel Page Engine (PPE) 9.0.4
- Portal Cache

For the My Oracle deployment, no external service can ever call the middle-tiers directly. They go through a virtual IP or a URL that exists only in the LBR. The LBR routes the request directly to the web-cache cluster which in turn passes the request on to a middle-tier. This provides a higher level of security and the ability to control specific types of hacker attacks in one central location. The following sections summarize the important features of the middle tier:

Oracle HTTP Server

The middle tier includes an Oracle HTTP listener that handles all the incoming HTTP requests to the portal. If the incoming request asks to display a portal page, the listener hands the request to OC4J running the Parallel Page Engine (PPE). The PPE splits up all the portlet requests that make up the page and sends them off in parallel to wherever the portlets are served from. If the request asks to execute a database portlet or a PL/SQL procedure, the listener hands it to mod_plsql to communicate to the database. As each of the requests come back, the Parallel Page Engine reassembles the page and returns it to the user. If one of the portlets is too slow in returning, the Parallel Page Engine returns the rest of the page once the timeout duration specified in the provider has elapsed.

Portal Cache

The Portal Cache is the persistence component of the My Oracle caching armory and is a key component in supporting very high request rates. The Portal Cache uses the file system on the middle-tier to store fully assembled pages as well as page definitions and rendered content from individual portlets. Retrieving content from the cache is much faster than having the database and providers re-generate it and the parallel page engine re-assembles it into pages for every request. So if the page definition hasn’t changed from the last time the user requested it, then the cached information will be served up very quickly. Furthermore, if the portlet content hasn't changed from the last time the user requested it, the Portal Cache serves up the already assembled page.

The Portal Cache entries can be configured on a page-by-page, or portlet-by-portlet basis to check if a refresh is required (Validation-based Caching) or to refresh periodically (Expiry-based Caching). My Oracle uses Expiry-based caching for some portlets and pages.

### OracleAS Web Cache Machine Specifications

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Caches/machine</td>
<td>2</td>
</tr>
<tr>
<td>Total Physical RAM</td>
<td>4096 MB</td>
</tr>
<tr>
<td>Power Supply</td>
<td>Dual</td>
</tr>
</tbody>
</table>
My Oracle also takes advantage of Invalidation based caching, which allows the repository to send an invalidation SOAP/HTTP message to the cache when and only when the content is no longer valid. This means that we reduce the overhead of involved with a validation-based cache 'ping' which occurs every time the content is requested to see if it is still valid. We also reduced the reliance on expiration based caching, which whilst effective and cheap in terms of processing overhead, can be a little too coarse grained for some of the more dynamic page caching requirements.

**Portal User and System Level Caching**

OracleAS Portal supports user customizations of pages and portlets. It also filters out portlets and content items that a user is not authorized to view. Because of these customizations and security features, the view of a page (e.g. the My Oracle Home Page) can vary from one user to the next. Portal's caching is designed to allow content to vary on a per-user basis, even though the physical URL being cached is the same across users.

Two levels of caching can be specified:

- **System level caching** - Allows users to share a single cache entry. System level caching requires that an object be accessible by all users (i.e., the object does not enforce access privileges) and is not customizable. A big advantage to specifying system level caching for an object is that a single copy of the object can be cached and shared by all users. Since no customizations of the My Oracle pages (except the Home Page) or portlets is allowed, this caching option provides the most optional performance, while consuming less Web Cache resources. Since content on all the pages - except the Home page - changes infrequently, page caching is set at ‘Cache Page Definition and Content at System Level for 480 Minutes’.

- **User level caching** - The cache entries stored are unique for that user only. User level caching must be used for objects that can be customized or that enforce their own access privileges. For security reasons, Web Cache caches user level entries on a per-session basis only (session level caching); when a user logs out, his or her Web Cache entries are no longer accessible. User level Portal Cache entries, which are accessed only through the Parallel Page Engine, are protected from unauthorized access and can therefore be shared across multiple sessions for the same user. For example, if a user logs out of My Oracle and then returns the next day, he or she will use valid cache entries from the previous day's session that are still stored in the Portal Cache. For the Home page, page caching is set at ‘Cache Page Definition and Content for 480 Minutes’ on the Home page. The Home page is treated slightly differently, since users can customize their Favorites portlet. In addition, content on the home page, especially the news stories, tends to change at least twice a week.

**OC4J**

Each machine runs two instances of OC4J for redundance purposes. The OracleAS Process Management Service (OPMN) will detect and correct any OC4J process death. Running two separate instances allows the middle-tier to continue to handle requests while the dead process is restarted should that event occur. The OC4J count is managed using the parameter numProcs which can be found in $ORACLE_HOME/opmn/conf/opmn.xml
Importantly, the middle-tier machines are not installed in a cluster as this promotes unnecessary chatting between machines. The OracleAS Web Cache cluster will hand requests on to the middle-tier machines as they are received. The server affinity discussed earlier will ensure that the request is returned to the correct middle-tier.

As an operation best practice, we restart one of the four middle-tier machines each day using a CRON script.

### MidTier Machine Specifications

<table>
<thead>
<tr>
<th>Model</th>
<th>Sun Ultra 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CPUs</td>
<td>2</td>
</tr>
<tr>
<td>CPU Speed</td>
<td>450MHz</td>
</tr>
<tr>
<td>Total Physical RAM</td>
<td>1024 MB</td>
</tr>
</tbody>
</table>

### Portal Database Server

The Portal repository resides within the internal Oracle network behind the DMZ. This means that only the middle-tier machines are allowed to make requests directly to it. If a request comes from anywhere else, it is denied.

### DB Machine Specifications

<table>
<thead>
<tr>
<th>Model</th>
<th>Sun Enterprise 6500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CPUs</td>
<td>14</td>
</tr>
<tr>
<td>CPU Speed</td>
<td>450MHz</td>
</tr>
<tr>
<td>Total Physical RAM</td>
<td>14336 Mb</td>
</tr>
<tr>
<td>Database Size</td>
<td>40 Gb</td>
</tr>
<tr>
<td>SGA</td>
<td>270 Mb</td>
</tr>
<tr>
<td>Operating System</td>
<td>Solaris 2.8</td>
</tr>
<tr>
<td>Database Version</td>
<td>9.0.1.3</td>
</tr>
<tr>
<td>Disk Configuration</td>
<td>EMC Tower (2.2Tb)</td>
</tr>
</tbody>
</table>

For high availability of the databases Oracle Dataguard is installed and a hot standby database runs on a second, identical database server (this second server is also used for other applications, so only a portion is allocated to running the My Oracle standby database). The synchronization of the log files is intentionally delayed for a period of 3 hours. This way, if there ever is any corruption, the bad information is not replicated to the failover system.

For additional reliability, the file system is mirrored on the servers' EMC storage systems, and standard database backups are also taken. The backups are first written to disk, and then streamed to tape - this has less impact on database performance and allows the tape backup to proceed at its own pace.
LOGIN.ORACLE.COM ARCHITECTURE

The SSO solution for My Oracle runs on separate hardware for the following reasons:

- Security for the SSO servers can be much tighter than that for the My Oracle and provider.oracle.com machines
- Administration of SSO servers should be separate from that of My Oracle. Oracle's internal Global IT division are solely responsible for the maintenance and management of the SSO architecture.
- Tuning the SSO Repository on a separate machine from the Portal repository allows for separate tuning parameters to be set in line with the requirements of the SSO application profile.

SSO Server Middle-Tier

The SSO Server middle-tier machines run the same configuration as the portal middle-tier machines. Web servers were specifically added to service just the SSO Server as other groups outside My Oracle need to share the user repository. With these dedicated web servers, the performance for the authentication service can be tuned and adjusted independently from the portal service. Again, if you were considering building a similar type of OracleAS Portal deployment, then this function could be shared with the same middle-tier machines used to front end the Portal database.

The SSO service is accessed via SSL. The SSL encryption is handled by an accelerator board in the LBR, this is important as the process of encrypting the outbound traffic will add an unreasonable overhead to the webserver, offloading the encryption to the LBR means that My Oracle can be configured to operate in full HTTP mode and rely on the LBR to deal with the encryption.

Techniques for configuring these types of set-ups are discussed in the Oracle9iAS Server Security Guide.

Each middle-tier is running

- Oracle HTTP Server
- mod_plsql 9.0.4
- OC4J 9.0.4

The middle-tiers are running the 10g versions of the listener and the java environment. Unnecessary pieces of the default 10g environment are removed after the installation; these include DAV, OC4J_Demos, OC4J_Portal and OC4J_Wireless. For an in-depth definition of the components removed and further configuration please refer to the document titled 'Configuring a Secure Oracle9i Application Server Environment' on OTN.

A custom login page is used to enable the branding of the SSO Server look and feel. The login.oracle.com one is written using JSP, but this could just as easily have been written in Perl or PHP. Two instances of OC4J are running on each of the middle-tiers for the same redundancy reasons discussed earlier.

There are two middle-tier machines configured as follows.
**SSO MidTier Machine Specifications**

<table>
<thead>
<tr>
<th>Model</th>
<th>Dell 1550 Rackmount</th>
<th>Dell 1650 Rackmount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CPUs</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CPU Speed</td>
<td>1.2GHz</td>
<td>1.4GHz</td>
</tr>
<tr>
<td>Operating System</td>
<td>Red Hat Enterprise Linux, version 2.1</td>
<td>Red Hat Enterprise Linux, version 2.1</td>
</tr>
<tr>
<td>Total Physical RAM</td>
<td>4096 MB</td>
<td>4096 MB</td>
</tr>
<tr>
<td>Power Supply</td>
<td>Dual</td>
<td>Dual</td>
</tr>
</tbody>
</table>

**SSO Server Database**

login.oracle.com uses the core security and authentication features of OracleAS. This means we use the SSO Server to pass the initial authentication requests for logging in to Identity Management and to manage retrieval of usernames and passwords for external applications that may be stored in the SSO schema.

Upon initial authentication the username and password submitted by the user are passed to the Identity Management instance for an LDAP authentication. The encrypted text password entered by the user is run through a one-way encryption and the encrypted string is compared against that retained in the Identity Management repository. If a match is made then Identity Management will return an authenticated token and the SSO may commence the login process.

**OID Machine Specifications**

<table>
<thead>
<tr>
<th>Model</th>
<th>Sun E4500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CPUs</td>
<td>12</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>450MHz</td>
</tr>
<tr>
<td>Total Physical RAM</td>
<td>12288 MB</td>
</tr>
<tr>
<td>Database Size</td>
<td>2 GB</td>
</tr>
<tr>
<td>SGA</td>
<td>270 MB</td>
</tr>
<tr>
<td>Operating System</td>
<td>Solaris 2.8</td>
</tr>
<tr>
<td>Database Version</td>
<td>9.0.1.3</td>
</tr>
</tbody>
</table>

**SSO Machine Specifications**

<table>
<thead>
<tr>
<th>Model</th>
<th>Sun E420R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CPUs</td>
<td>2</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>450MHz</td>
</tr>
<tr>
<td>Total Physical RAM</td>
<td>2048 MB</td>
</tr>
<tr>
<td>Database Size</td>
<td>0.5 GB</td>
</tr>
<tr>
<td>SGA</td>
<td>270 MB</td>
</tr>
<tr>
<td>Operating System</td>
<td>Solaris 2.8</td>
</tr>
<tr>
<td>Database Version</td>
<td>9.0.1.3</td>
</tr>
<tr>
<td>SSO Version</td>
<td>9.0.2.6</td>
</tr>
</tbody>
</table>
PROVIDER.ORACLE.COM ARCHITECTURE

provider.oracle.com is a separate installed architecture used to resolve requests for content from web providers registered within the My Oracle repository. The use of this separated provider architecture allows for the scaling of component hardware according to specific request requirements, without having to worry about the trade off this may cause for other components of the portal architecture, as would be the case if the providers were running from within the My Oracle architecture. The distributed provider architecture also allows us to manage access and security independently from the My Oracle machinery. If for example a developer needs access to the provider.oracle.com architecture it can be provided without having to compromise the security of the My Oracle or login.oracle.com architectures.

<table>
<thead>
<tr>
<th>PROVIDER.ORACLE.COM Machine Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
</tr>
<tr>
<td>Number of CPUs</td>
</tr>
<tr>
<td>CPU Speed</td>
</tr>
<tr>
<td>Total Physical RAM</td>
</tr>
<tr>
<td>Oracle HTTP Listener</td>
</tr>
<tr>
<td>Jserv Engines</td>
</tr>
</tbody>
</table>

There are many different portlets available to My Oracle from within the provider architecture. These primarily take the form of a piece of java code utilizing the Portal PDK to import the portlet structure and provide its own piece of unique content. This content may be dynamic or static. For performance reasons any content refresh is limited to a lifespan of 30 minutes, thus reducing the load on the portal infrastructure, cached content being much easier and quicker to serve up than dynamic content.

Within the network infrastructure the provider.oracle.com machines are installed on a separate sub-net to the My Oracle machines, this enables us to avoid a common network request problem, as outlined below.

In the example illustrated in Figure 6, Machines M1 and M2 are installed in a different subnet to the P1 & P2 machines thus all requests will travel through the LBR.
If this was not the case and we suppose that P2 was the provider of a portlet that was being used as a component on a page existing in the M2 repository then the following sequence of steps would occur in the request process.

- Page requested by browser from M1
- M1 passes the request to the PPE on M2
- M2 requests a portlet from the provider hosted on P1 through the LBR
- P1 passes the request on to the hosting machine P2
- P2 receives a request from machine M2, which it has knowledge of because they are in the same subnet
- P2 services the request and hands it back to M2 directly, not through the LBR
- M2 ignores the response as it did not request direct contact from P2, it is waiting for a response from the LBR
- The portlet request is never completed and the portlet will hang until the provider timeout is reached and the page returns with a message that the provider timed out.

An alternative to implementing the machines on a separate sub-net would be to either use Network Address Translation (NAT) or configure the LBR to act as a proxy instead of a router.
ORACLE INTRANET

Oracle's online services (www.Oracle.com, Oracle Technology Network (OTN), and My Oracle) use a single user repository to authenticate users, called Oracle's online registration system. This system also handles registrations for the eBusiness Network, AppsNet, and Oracle's Partner Network. There are also a number of internal applications and content providers on the Oracle intranet. These providers are integrated into My Oracle, but are only accessible to Oracle employees. OracleAS Portal's flexible security architecture allows employees and customers to share the same portal while ensuring that each user only sees the information for which they have been authorized. Authorization privileges can be specified on each portal page, for each portlet, or by the applications exposed through the portlets.

Reliability and Availability

As discussed above, all of the middle-tier components have a redundant set of servers to ensure high availability. Oracle Enterprise Manager is used to monitor the system and issue alerts when anything goes wrong.

It is planned for My Oracle to use Real Application Clusters (RAC) from the Oracle10g database. RAC is the evolution of the Oracle Parallel Server (OPS), offering the failover and load-balancing benefits of OPS with considerably less administrative complexity. This implementation of RAC is planned to occur in the second half of 2004.

Every machine has at least two network interface cards (NIC's) to ensure access to the machines is available for administrators at all times. One NIC is employed as the main interface; the other is used for doing online network backups. Both NIC's are on separate switches and subnets. This setup is also exceptionally useful if a Denial of Service attack is suffered on the main NIC. Administration staff can access the target machine through the secondary NIC and deal with the problem.

Performance logging is switched on throughout the whole My Oracle architecture; mod_plsql, PPE, OC4J, Web Cache and DB logging are all utilized to give the fullest picture possible of the health of systems.

The performance logs are switched twice in a 24-hour period and these finalized logs are fetched by a specific log analysis system. The analysis system generates performance reports on a daily basis and retains seven days worth of data, approximately seven million rows of data. The data from these logs is then uploaded into an Oracle 10g database for analysis. The steps for this are documented in the technote 'Performance Monitoring with mod_plsql in Oracle9iAS Portal Release 2' and an overview of the types of reports available can be found in the technote 'Object Access Reporting from the Performance Logs in Oracle9iAS Portal'.

The performance logs are analyzed to check for poorly performing pages, portlets and providers, however there is also a failsafe monitoring system in place utilizing the monitoring capabilities of Oracle Enterprise Manager.

If an issue has arisen either via the performance logs or through database or system alerts, these alerts are emailed to My Oracle site administrators. Oracle's IT staff will be automatically paged and requested to resolve the problem immediately.
Monitoring listener status is an automated process that issues the following series of ramped tests every five minutes to every middle-tier machine.

- File request – a known file (JPG) is requested from My Oracle, it must be returned within a given time frame to pass this stage of the test.
- CBUF Test – the string 'http.p?cbuf=test' is issued to the mod_plsql component of My Oracle. An HTML response of test should be received within a given performance threshold. Failure to do so would once again trigger a performance alert
- IsItWorking Servlet Test – OracleAS ships with a servlet called IsItWorking which is requested by the /servlet alias of Oracle HTTP Server. An HTML response should be received within a given performance threshold. Failure to do so would once again trigger a performance alert
- PPE Servlet Test – A page id from the portal is requested by the parallel page engine servlet, a suitable response within a given time frame is expected.

If the first four tests are passed then the tool moves onto performing an integrated test, executed via the LBR which consists of the following steps:

- Navigate to http://my.oracle.com
- Login
- Customize the screen to add a new portlet.
  - View the new page.
- Customize the screen to remove the new portlet.
- View the new page.
- Logout.

If all the tests are passed then the tool will sit idle for 5 minutes before starting again. If any of the steps fail, then a warning is issued and the testing steps are restarted. If the error reproduces on three occasions then the alerting process mentioned above is put into place.

Internally there are other less specific tests that get done on a regular basis. For example we regularly run iostat and vmstat on our unix servers.

**Configuration Settings**

The configuration of the components within OracleAS for My Oracle are generally left at the defaults they ship with, however some components are either shut down or removed for security purposes whilst others are tuned for performance reasons.

For a single mid-tier on My Oracle the following setting are dependent on the site usage, site content, and modes of caching used by content providers. This can only be used as a reference, and the actual settings for a site have to be determined separately on a per-site basis.

My Oracle Configuration for Oracle HTTP Server (OHS) (httpd.conf)

- KeepAlive Off
- MinSpareServers 30
- MaxSpareServers 100
• StartServers 30
• MaxClients 250
• MaxRequestsPerChild 2000

My Oracle configuration for mod_plsql (dads.conf)
• DAD level NLS_LANG is configured to match the database NLS_LANG, avoiding unnecessary unsupported requirements to have charset conversions for response being fetched through mod_plsql
  ○ PlsqlNLSLanguage AMERICAN_AMERICA.UTF8
• The same setting is done for the SSO DAD as well
• PlsqlIdleSessionCleanupInterval is defaulted to 15 minutes

Database TNS Listener is configured for DEDICATED connections. MTS mode has additional overheads which we wish to avoid. If your portal has Multi-Threaded Server (MTS) enabled MTS, you should try running without it to see if a performance improvement is achieved. Typically there is little need to use MTS with the connection pooling and reuse model of mod_plsql.
PERFORMANCE LOGGING ANALYSIS

Through using the performance logging features of mod_plsql and the PPE it is possible to obtain detailed log entries that may be analyzed. For a full description of how to implement this logging service, please see the technote 'Portal Performance Monitoring in Oracle Application Server Portal', which is available on Portal Center. In addition, customers may consider third-party reporting technologies from companies such as Omniture or WebTrends for Portal Usage Tracking.

Daily Unique Visitors Per Day

In order to determine how My Oracle is being used, Oracle licensed SiteCatalyst from Omniture (www.omniture.com). SiteCatalyst is a remotely hosted, subscription-based reporting solution for real-time web site analysis. By providing reports that contain powerful insights into the visitor interaction and usage behavior of My Oracle, SiteCatalyst allows us to determine which areas to focus on. SiteCatalyst provides an additional level of statistics to that which can be obtained through analyzing the Portal logs.

![Graph of Daily Unique Visitors per day](image)

Figure 7: Daily Unique Visitors per day

The chart in Figure 7 indicates the quantity of unique daily views per day for the week April 4 to 10, and the week prior. Since employees are required to login, this chart also shows successful logins over a 7-day period. A successful login is composed of one or more successful calls to wwptl_login.login_url and wwsso_app_admin.ls_login, terminated by a successful call to wwssec_app_priv.process_signon. Success is defined as returning an HTTP status code of less than 400. Since the number of calls to the first two functions can vary, the time reported is a weighted sum based on the frequency of the 3 calls over the time period specified. If your system is properly configured, the average value should be 1.6-1.8 seconds.

The chart is showing that there is significantly less demand for login access during the weekend and that demand is higher at the beginning of the week than the end of the week.
Hourly Unique Visitors

The chart in Figure 8 drills down further on the daily unique visitor data to show the unique visitors for a given hour, and therefore the successful unique logins. The time measurement is based on the timestamp recorded on Omniture’s servers. Their machines are based in the PST timezone (GMT-8), so it reasonable to concur that the majority of the logins are from the US rather than other timezones around the world.

Hourly Unique Visitors are defined as a visitor's first visit to My Oracle in an hour period. While one person may visit My Oracle and view the home page three times, the Apps page twice, and several other pages once each, the Unique Visitors report records that person as one "unique visitor" for the time allotted.

We use the Daily and Hourly Unique Visitors report to:

- See the number of different people that viewed My Oracle during any given hour.
- See when peak/off-peak times are - know when to make changes to My Oracle.
- See recent traffic patterns and how announcements are affecting unique visitors to My Oracle.

The peak login time is between 6 am and 10 am with logins tailing off through the day.
The chart in Figure 9 indicates the number of times the My Oracle home page was viewed over a 7-day period. The Page Views report displays statistics for the entire site. A page view is counted every time someone visits the My Oracle home page. This includes pages that are loaded as a result of the reload and back buttons. We use this report to:

- See recent traffic patterns for your entire site and each individual page.
- Identify anomalies - such as hosting problems.
- Determine the best downtime for applying patches and rolling our updates/new features.
- Measure visits to promotional, sign up, or order pages for marketing effectiveness.

The chart confirms the view that there is significantly less demand for My Oracle during the weekend and that demand is higher at the beginning of the week than the end of the week.

Note there is more traffic at the end of the weekend (Sunday), than at the beginning (Saturday). This is due to Oracle employees logging into My Oracle from time zones (e.g. Asia, Europe) that are ahead of the recording time zone, PST (GMT-8).
Page Views Per Hour

The chart in Figure 10 drills down further on the Page Views data to show the unique successful Home page views for a given hour. Combining this report with the Daily Unique Visitors and Hourly Unique Visitors report, indicates that in order to minimize the performance impact on users, any changes to the My Oracle pages should occur either on a Tuesday or Thursday after 4pm.

Top Ten Pages on My Oracle

The chart in Figure 11 shows that there is a large number of Oracle employees logging in and viewing the home page. What is interesting about this chart is that very few employees navigate to the other employee pages. In fact, this follows the classic portal navigation model. This model follows a ‘landing pad’ approach where
all users hit the home page, find what they are looking for, and then visit another site.

**Omniture and Portal Performance logs**

As noted earlier, there are two ways to assess portal usage and performance: Use the Portal Performance Logs for portal performance. They can also be used for tracking portal usage. The Portal Performance logs need to be parsed to extract the necessary data, and that data extracted to produce charts for determining portal usage. Third party solutions, such as Omniture's SiteCatalyst, provide immediate information on portal usage, but no information on portal performance.

The Portal Performance Logs are included within OracleAS Portal as described in the technote 'Portal Performance Monitoring in Oracle Application Server Portal'. These capabilities thus require no additional software purchase. We still use the Portal Performance Logs to maintain a history of the of the CPU costs of generating content from within a web provider versus the additional framework costs versus the overall page generation costs. We also use the Portal Performance Logs to determine the fastest and slowest running portlets. Omniture does not currently provide this level of portal performance information.

The advantage of Omniture SiteCatalyst over locally stored usage statistics (e.g. the Portal Performance Logs) is that the information is current, due to Omniture's hosted solution. When we looked at means to collect usage statistics and building our solution based off the Portal reports, this was a key advantage. We've found the Omniture SiteCatalyst licensing costs to be very reasonable and cost efficient for us.

In summary, the Portal Performance Logs are useful for answering technical/performance questions. Third party solutions, such as WebTrends and Omniture, are useful for answering business related questions, such as portal usage.
CONCLUSION

My Oracle demonstrates the use of portal technology to enhance an organization's productivity. As a portal that delivers compelling business content and applications to Oracle employees, My Oracle helps employees find the information and tools they need to do their daily work. Flexible security enables the same portal infrastructure to also serve external users, and a robust architecture allows the portal to scale to large numbers of users. The most important part is that everything used to produce My Oracle is the exact same software customers use and buy in Oracle Application Server.

The upshot of the performance data in this document is that in April 2004 despite handling in excess of 10,000 unique logins and serving in excess of 34,000 unique page requests the middle tier CPU (even at peak page load of ~2,600 page views around 10am) never exceeded an average 6% utilization and the repository server never exceeded an average 17% utilization whilst still returning an average response time of 0.25 seconds.

The next challenge for My Oracle and the underlying OracleAS Portal technology is to ensure that the speed of delivery of pages is uniform across the globe. The performance figures within the document are all based on in-server generation and delivery times, they do not account for the network latency or bandwidth restrictions between the My Oracle servers and the user's browser.

OracleAS Portal scales and performs exceptionally well for us. Contact an Oracle Sales Representative to arrange a demonstration¹ or download² Oracle Application Server for free right now!

¹ http://www.oracle.com/products/buy/