Best Practices for Oracle FMW Webcenter Portal 11g Multi Data Centre Deployment

Oracle Maximum Availability Architecture White Paper
August 2013

Maximum Availability Architecture

Oracle Best Practices For High Availability

ORACLE

Requirements 2
Definitions 2
Supported Products 2
Topology Requirements 5
Networking Requirements 7
Validation and Testing 9
The Active-Active Topology 10
Global Load Balancer Tier 12
The HTTP Server/Local Load Balancer Tier 12
WebLogic Domains 12
The Application Tier 12
The Database Tier 13
Configuring Oracle Webcenter Portal Active-Active 13
Setting up the first Site 13
Setting up the second Site 14
Configuring Custom Applications 15
Configuring External Resources 16
Configuring the Identity Store 16
Configuring the Policy Store (OID) 18
Configuring the Policy Store (DB) 20
Configuring the Java Object Cache 21
Configuring other External Services 25
Configuring and Managing the Load Balancers 27

Configuring the Local Load Balancer 28
Configuring the Global Traffic Manager 28
Configuring and Managing Session Replication 29
MAN Replication 29
No Replication between Sites 40
Configuring and Managing Database Access 41
Database Configuration 41
Lifecycle Management 43
Configuration Management 44
Patching and Upgrades 44
Topology changes 44
Supporting Other Applications 45
Clusterability 45
Access to External Resources 45
Synchronizing Local Resources 46
Filesystem resources 46
Local caches 46
Appendix A: Network Monitoring Tools 47
Monitoring Interfaces in Linux 47
Adding Delay and Packet loss to an Interface 48
Using iperf to monitor bandwidth 49
Appendix B: Load Testing Tools 50
Oracle Application Testing Suite 50

Apache Jmeter 50
References 51

Introduction

There are multiple advantages to having two active Middle Tier sites. Two separate data center deployments can, of course, protect against local unplanned outages while still fully utilizing all available resources. Both sites can serve users while also acting as failover sites for each other. It is also true that for many applications, much of the processing takes place in the Application or Middle tier, much more so than access to the Database. This may be because of a large Presentation layer or because of efficient caching mechanisms in the application layer. Thus, keeping each Middleware site local to different segments of users may also provide better performance and distribution of resources over one central site.

This paper discusses the best practices for deploying the Application tier into two distinct but simultaneously active sites. It also covers the steps necessary to achieve this configuration.

This paper's focus is on running Oracle WebCenter Portal in an Active-Active configuration. Both Oracle Webcenter Portal sites share the same back-end database. They also optionally are deployed with other products. For required products that cannot be deployed Active-Active some guidance is provided as to how to deploy these in an Active-Passive configuration. The result is a hybrid model where all products are deployed at both sites but only a subset of these products are active and serving requests.

Finally, this paper discusses how such an environment can support seamless failover. That is, if one of the two sites fails, users will automatically be redirected to the surviving site and can continue their work. Thus, a multiple site configuration also provides greater Availability of the application tier.

Requirements

This paper outlines a multi data center deployment of Oracle WebCenter Portal. However, for this configuration to be achievable, a **very specific** set of requirements must be met. These requirements include products supported in this configuration, the specific of the topology and, most importantly, the networking requirements.

The specific requirements are outlined in more detail in this section. Readers should ensure that these requirements are met before proceeding with the remainder of this paper.

Definitions

The following terms are used throughout this paper and are briefly defined here:

Site

A Site is an independent stack of Oracle products, hardware and other software. Sites are physically separated from each other. A Site can serve user requests from the Web Tier to the Middleware Tier to the Database Tier. Sites may have dependencies on each other or on external products.

Servers or Managed Servers

Independent, WebLogic application containers. Servers will usually run as part of a Cluster. And the Cluster will be part of a Domain.

Domain

A WebLogic Domain. A Domain will include an AdminServer and Clusters of Servers.

Machine

A physical or virtual environment running an operating system such as Linux or Windows.

Supported Products

This paper outlines the steps to deploy Oracle WebCenter Portal products in an Active-Active configuration with a cluster of Oracle WebCenter Portal servers at each Site both serving user requests. Note that only Oracle WebCenter Portal is supported in this configuration and, even so, only a subset of products within the Oracle WebCenter Suite are supported in this manner.

Products that cannot be deployed Active-Active must be deployed Active-Passive.

The table below outlines different products and the configurations that they can be deployed into.

Product	Can be deployed Active-	Can be deployed

	Active?	Active-Passive?
WebCenter Portal		
WebCenter Portal: Spaces Application	Yes	Yes
WebCenter Custom Applications	Yes	Yes
Activity Graphs	No (Single site support only)	Yes
Analytics	No (Single site support only)	Yes
Announcements	Yes	Yes
Blog	Yes	Yes
Discussions server	Yes	Yes
Lists	Yes	Yes
Mail	Yes	Yes
Pagelet Producer	Yes	Yes
People Connections	Yes	Yes
Personalization	Yes	Yes
Polls	Yes	Yes
Portlets	Yes	Yes
Recent Activities	Yes	Yes
Wiki	Yes	Yes
WebCenter Content		
WebCenter Content Server (Documents)	No (Shared File Storage Required)	Yes
Oracle SOA		
Worklist	No (BPM/SOA Worklist)	Yes
Identity Management		
OID/LDAP	Yes (sharing a DB)	Yes

As can be seen in the table, there are a few products considered as part of Oracle WebCenter Portal that cannot be deployed Active-Active. These are discussed here.

Oracle WebCenter Content Server

Oracle WebCenter Content Server is often used in conjunction with Oracle WebCenter Portal. In fact, the two products are almost inseparable. However, there are restrictions in Content Server that do not allow it to be deployed Multi Datacenter. This means that a Multi Datacenter WebCenter Portal configuration must rely on a single instance of Oracle Content Server, deployed either at one of the two Sites or at a third location.

Activity Graph/Analytics

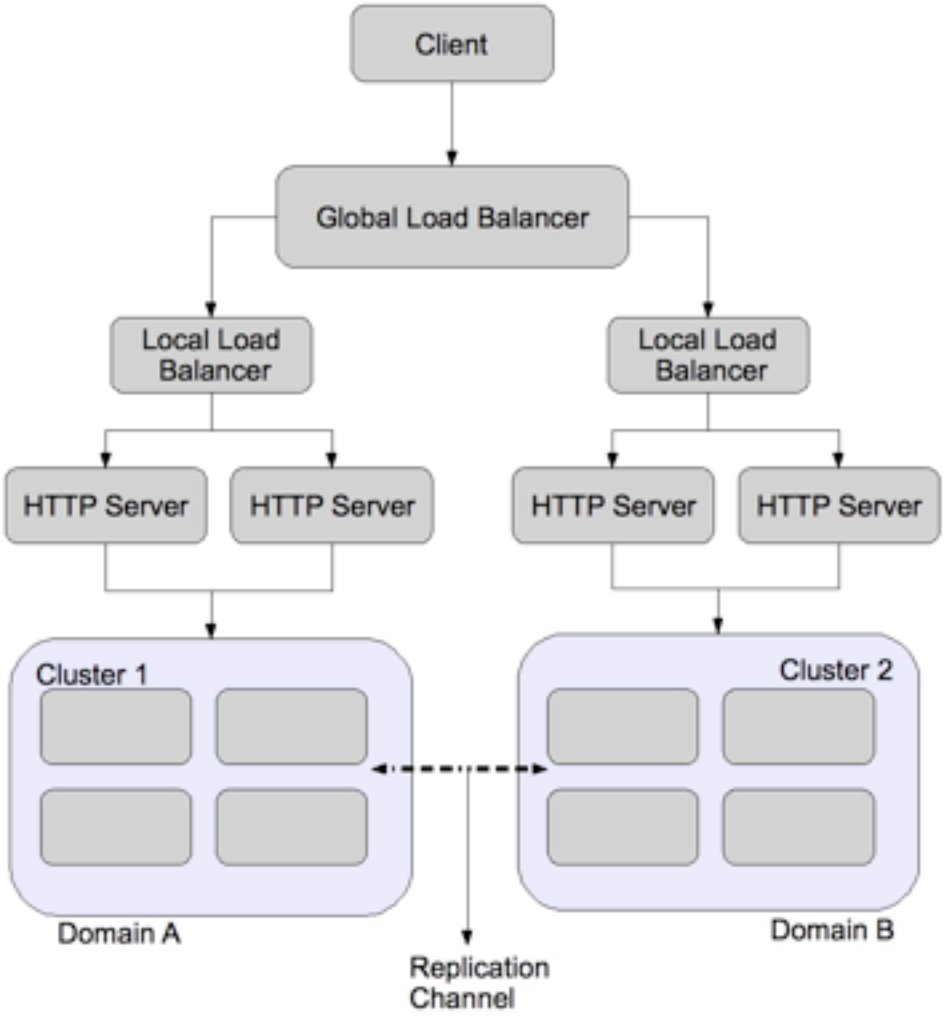
Only one instance of Activity Graph can exist at any one time. This restriction arises because Activity Graph acts as a statistics collector. Multiple instances of Activity Graph would lead to conflicts in the data collection and analysis.

Since both Oracle WebCenter Portal Domains share the same database schemas, only one Activity Graph can exist across **both** Sites. It may be possible to run one instance of Activity Graph at one of the Sites only.

Worklist

The Worklist feature of Oracle WebCenter Portal relies on Oracle's BPEL Server. This is a product outside of Oracle WebCenter Portal's product suite. As with Content Server, external products such as SOA BPEL must reside at one site and be deployed Active-Passive

Topology Requirements


Client
Global Load Balancer
Local Load Balancer
Local Load Balancer
HTTP Server
HTTP Server
HTTP Server
HTTP Server
Cluster 1
Cluster 2
Domain A
Domain B
Replication Channel

The Multi Data Center topology has the following features:

Two, Active Middle Tier Sites

Two distinct Middle Tier sites are each able to fulfill requests from users. The two sites are running the same application built on products that are supported in Active-Active (see next section for the list of supported products). Any user can switch from one site to another and have all their persisted state be available. Each site has its own entry point, consisting of an HTTP Server or Load Balancer.

Two Separate Domains

Each Site is an independent WebLogic Domain that includes the Servers at that Site. The Domain does not include any Servers outside of that Site.

Optional Cross-Domain Session State Replication

The two Domains may be configured to replicate session state synchronously to each other. This has the advantage of having the User state available at each Domain. This has the disadvantage of potential performance implications.

Optional (but recommended) Object Cache Replication

Servers will share an Object Cache in order to reuse objects and save trips to the Database. This is usually configured to Servers within a Domain but the Object Cache is a peer-to-peer cache and can be configured across Domains.

One Database

The Servers on both Sites are configured to write to **the same Database**. This Database may be local to one of the two Sites. In this case, it is recommended to have a backup of the Database at the other Site. An Active-Passive Dataguard configuration, for example, may be used for this.

Non Active-Active products

Both Sites may also rely on products that are not supported in an Active-Active configuration. This may include an external Identity Management installation, for example. A decision needs to be made as to where these installations will exist –at one of the two Sites or at a third location.

Networking Requirements

For the two sites to perform sufficiently well, it is important that sufficient resources be allocated in the form of hardware and processing power. This is true for any environment and so such considerations are assumed to be taken care of and are outside the scope of this paper.

An enterprise environment should also be structured so that network issues and latency are not a hindrance to performance. In an Active-Active environment outlined in this paper, each of the

Sites is not self-sufficient but relies on communication with the other Site, either through direct communication at the Middle Tier or through communication with the database or both.

These considerations and requirements are discussed in more detail throughout this paper. But the basic requirements are outlined here.

Middle Tier network requirements

Although the two Sites operate fairly independently, there are two instances where they may need to communicate directly with each other. In the first case, if cross-cluster session replication is enabled then servers at one Site are communicating directly with servers at the other Site. Secondly, servers will have a shared Object Cache that spans the two sites. Because replication is synchronous, there should be sufficient bandwidth and network stability for this to occur reliably. This paper also outlines some guidelines on the network round-trip time between the two sites.

Specifically, a network RTT time of >10ms is deemed to be too large to accommodate a reasonable response time for the applications if session replication is enabled.

Database-Application network requirements

There is only one Database, which must reside at one of the two Sites. This means that the other Site will be accessing a remote database. The network used for the applications on one Site to access the remote Database may be the same network, with the same characteristics.

The network requirements depend strongly on the amount and type of activity on the database. *Nevertheless, because a latency of as little as 8ms RTT can double the wait time for typical database activity, it is highly recommended that the network distance not exceed this.* Again, testing of the application may produce a smaller or larger limit than this.

Database-Standby network requirements

The Primary Database must also send Redo activity to the standby Database. If this occurs synchronously, then this also must be considered.

From MAA tests on Dataguard 10g databases it was discovered that "A Data Guard environment between New York and Montreal (up to 330 miles apart *with 10ms RTT latency)* using synchronous transport mode can provide zero data loss with minimal performance impact (less than 5%) for production databases generating redo data at rates up to 4MB/sec. "

This is from a test on Oracle 10g, however. On 11g, we can assume that the performance is at least as efficient if not more.

Overall network requirements

Although the Middleware and Database tiers have been separated out in the discussion above, most likely the network will be the same network in all cases. The overall recommendation then is that ***the network RTT time between the two sites not exceed 10ms***.

This is not an absolute requirement. For some intense applications with high workloads and strict requirements this value may be less. For some applications with low activity and low user traffic and lightweight tasks this value may be able to be higher. In all cases, **the application should be tested either in a simulated environment or the actual environment to ensure that response times are acceptable.**

Validation and Testing

The preceding requirements are meant to act as general guidelines. In all cases, it is still strongly recommended that applications be tested under realistic load scenarios in order to determine whether the response time is acceptable. In particular, the application should be deployed and the following tests should at least be performed:

User Response Time on First Site

The First site will consist of some hybrid of Active components and Passive components. This combination should be tested to ensure that user response time on this Site is acceptable. This test should be performed while users are also active on the second Site.

User Response Time on Second

The Second site will consist of some hybrid of Active components and Passive components. This combination should be tested to ensure that user response time on this Site is acceptable. This test should be performed while users are also active on the First Site.

Failover of Passive Components

Each of the Passive components should be able to fail over independently. For example, the passive WebCenter Content can become the active WebCenter Content. These components should be failed over and the User Response Time tests above should be repeated.

Full Site Failure

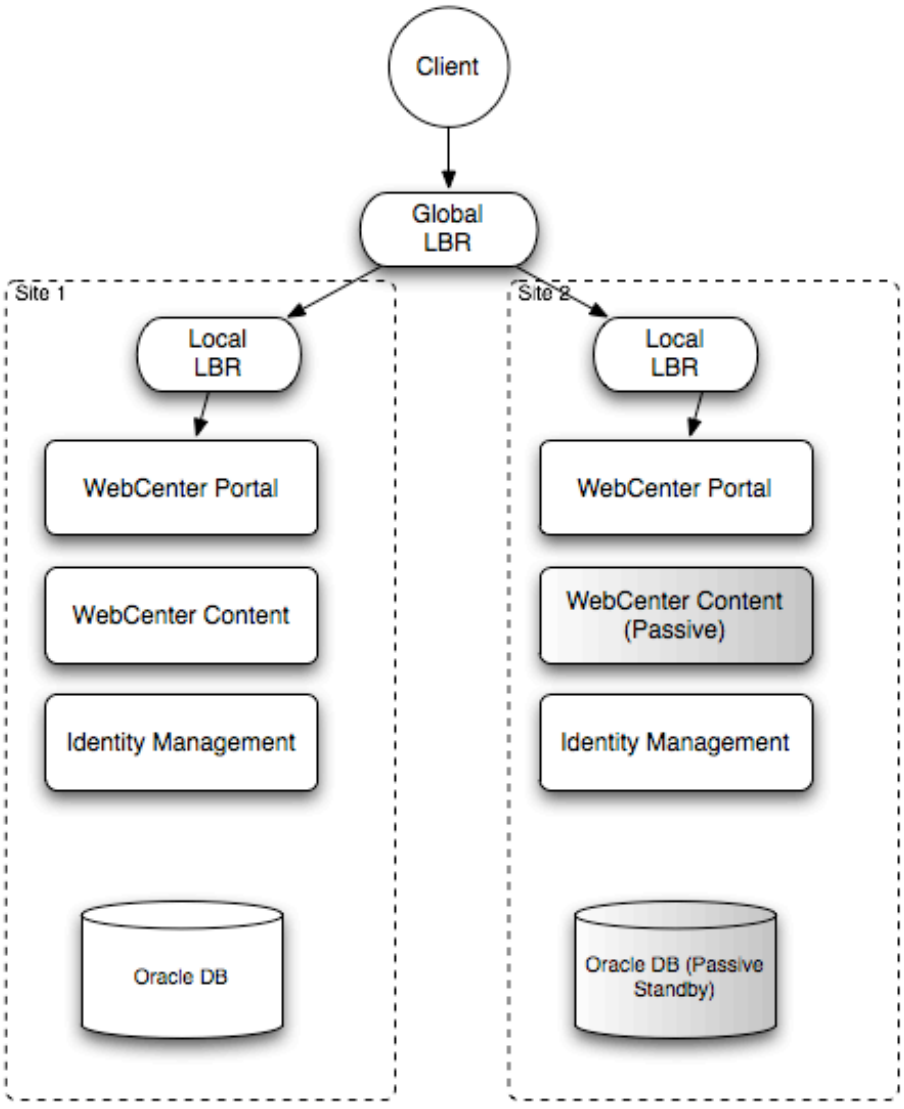
An Active-Active configuration also can tolerate loss of one of the two Active Sites. In the case that one of the two Sites is completely unavailable, the other Site can become completely Active.

That is, all Active-Passive components are now made Active on the surviving site with the result that the surviving site is completely functional and independent.

The Active-Active Topology

The topology discussed in this paper is one in which there are two separate sites although they are accessed by one entry point – a global load balancer configured at each of the sites which directs traffic based on some specified criteria. Then, each site has its own local entry point – a local load balancer. The local load balancer distributes requests to multiple HTTP Servers. Finally, the local HTTP Servers allocate requests to specific servers. The two environments share one database.

The diagram below shows how such a topology might be configured:



More detail on each of the tiers is provided below

Global Load Balancer Tier

The Global Load Balancer is a load balancer configured to be accessible as an address by users of all the sites. This needs to be mapped to a DNS name that is accessible to any client regardless of the site to which they will be eventually connecting.

The Global Load Balancer (GLBR) directs traffic to either site based on a configured criteria. This criteria can be client-IP for example. This should be used to create a Persistence Profile which allows the LBR to map users to the same site upon initial and subsequent requests.

The GLBR maintains a pool which consists of the addresses of all the local load balancers. In the case of failure of one of the sites, users automatically are redirected to the surviving active site.

The HTTP Server/Local Load Balancer Tier

At each site, a Local Load Balancer receives the request from the Global Load Balancer and then directs the request to the appropriate HTTP server or the local managed servers.

In either case, the Local Load Balancer is configured with a persistence method such as Active Insert of a cookie in order to maintain affinity and ensure that clients are directed appropriately.

In the case of HTTP Servers, the WebLogic Plug-in should be installed and enabled so that an ongoing request is mapped back to the Webcenter Portal server that holds the session information.

WebLogic Domains

Each site is configured with a FMW installation. Each site is a separate WebLogic Domain.

The main topology assumed in this paper is one in which each site is **its own separate Domain(s) and installation.** Each site has a separate AdminServer and cluster of Oracle Webcenter Portal Servers. This is the Multiple Domain model.

The Application Tier

The Application tier consists of a cluster of Oracle Webcenter Portal servers. Specifically this includes Oracle Webcenter Spaces, Portlet providers and Custom Webcenter Portal servers. Other products may also be running either in the same Domain as Oracle WebCenter Portal or in a separate Domain on the same Site.

Oracle WebCenter Portal will be running Active-Active. Other products will be either running Active-Active or Active-Passive depending on whether this is supported.

Each of the Oracle WebCenter Portal clusters is serving requests. The Application tier at both sites should be configured identically. This includes the number of available servers.

The Database Tier

The Application tiers all share the same Database. This may be one database instance for example or a RAC Cluster. The main assumption, however, is that both sites are accessing the same data.

In this configuration, we assume that both sites are accessing the same database and the database is setup in a Dataguard configuration. This allows failover of the database as well.

Configuring Oracle Webcenter Portal Active-Active

This section covers the steps to configure the Active-Active WebCenter Portal environment. This includes setting up the domain and applications for the identical sites.

Setting up the first Site

Creating the environment

The first site requires no special configuration. An existing site can be used. We'll assume that the installation and configuration consists of the following steps:

1) Install Binaries

   The binary installation will consist of a WebLogic Server home and an Oracle Home for Oracle Webcenter Portal.

2) Provision the Database

   The database should be created, if it does not exist already.

3) Seed the Database

   Use the Repository Creation Utility (RCU) to install the schemas required for Oracle Webcenter Portal into the Database.

   The correct services should also be created

4) Configure a Cluster environment

   The environment should consist of two or more machines, each machine running a member of an Oracle Webcenter Portal cluster. In this document, we discuss Oracle Webcenter Spaces, the Portlet Providers and any custom Webcenter Portal servers.

   The datasources should be configured to support Dataguard failover. See the next section for more information.

5) Configure HTTP Server

Two or more HTTP Servers should be configured to route requests appropriately that are received from the Local Load Balancer.

For more detailed steps on configuring this Enterprise environment, please refer to the *Enterprise Deployment Guide for Oracle Webcenter Portal.*

Datasource configuration

For an Oracle DataGuard database it is recommended to use Active GridLink in order to take advantage of Fast Connection Failover and Runtime Connection Load-Balancing.

This involves the separate steps of creating necessary workload services on both the Primary and Standby databases and then configuring the JDBC Connection URL when configuring Active GridLink.

The result is that in the event of a database role change in the DataGuard configuration, both of the Webcenter Portal sites can seamlessly failover to the new Primary.

The steps for configuring this are provided in detail in the whitepaper *Oracle WebLogic Server and Highly Available Oracle Databases: Oracle Integrated Maximum Availability Solutions*.

Setting up the second Site

As much as possible, the second Site should be identical to the first. Ideally, the second site should have the same number of Servers as the first Site. The strongest requirement is that the datsources on the second site be identical to the datasources on the first site.

The second Site can be initially configured by copying the first site. After that, parameters need to be changed to reflect the new environment.

There is no requirement to create the second site immediately. A second site can be added much later as well.

Provision the binaries

This can be done either via a fresh install, by copying or cloning the binaries from the first site. The second site should have the exact same version of all Oracle products.

Cloning the first Site

The second site can be created manually to be identical to the first site but we recommend using Oracle's Test to Production scripts to create the second site.

Use the `copyConfig` script to create an archive of the domain. Then extract the Move Plan. This is a file that contains parameters that can be edited. Then, the archive is copied over to the second site and the command `pasteConfig` is used to extract the archive to create the second domain. The following sections provide recommendations on which parameters should be altered.

More usage detail on the Test to Production scripts can be found in the *Oracle Fusion Middleware Administrators Guide.*

Domain name and Server names

The second site can be created using the same or a different Domain name as the first. There is no requirement that Domain names be the same so **it is recommended** to use two distinct names. When enabling Domain security, later in this paper, only Global Trust supports identical domain names. Cross-Domain security does not since the domain name is used to uniquely identify the domain.

The server and Cluster names can remain identical however. This is not an absolute requirement but is recommended in order to facilitate the cloning of one site to the other.

Listen Addresses

Listen addresses should only be changed if it is required because of different server names at the second site. This usually takes place in two locations: 1) The Listen address of the Managed Servers and 2) The Node Manager Listen address configured on the specific Machines.

Datasources

The datasources should remain the same. If a different physical address for the database is required then the datasources should be changed but otherwise the second Site **should be using the same database and pointing to the same schemas.**

Configuring Custom Applications

Once the sites have been configured, built-in applications such as Oracle Webcenter Spaces should be already deployed.

For Custom applications, the following steps are necessary in order to configure the application properly for an Active-Active environment:

Create an MDS Schema for the Application

This schema will be shared by the Framework application at both sites. To create the schema, use the Repository Creation Utility and create a new schema of type AS Common Schemas -> Metadata Services.

For more information see *The Oracle Fusion Middleware Administrator's Guide for Oracle Webcenter Portal.*

Provision Custom Application Servers at both Sites

A cluster of Managed Servers needs to be created at both sites. This can be done by extending the Oracle Webcenter Portal domain and adding support for the Custom Portal template. The

instructions for doing this are found in the *Oracle Fusion Middleware Installation Guide for Oracle Webcenter Portal.*

Create datasources as necessary at both Sites

Create datasources at both sites to the MDS schema created earlier as well as to any other datasources required for the application.

Deploy to the first Site

The deployment to the first site are the same as any standard deployment. If the application is distributed, ensure that it has been configured correctly to enable replication (if required) as `replicated-if-clustered`.

Deploy to the second Site

On the second domain, perform the deployment using the same steps as the first domain, but take care to specify the same information for the target metadata repository that was used on the initial deployment. Since this is the same application**, it must use the same MDS partition**.

Application redeployment

Note that if the application needs to be redeployed on either site this will have an impact on the second site. For example, any resources and customizations stored in MDS are shared across both sites. For more information, consult the *The Oracle Fusion Middleware Administrator's Guide for Oracle Webcenter Portal.*

Configuring External Resources

Once the two sites are configured, their external resources should also be configured. In this case, external means anything that is external to the Webcenter Spaces or the Custom Webcenter application. This includes Portlet producers, Discussion Server, SOA worklists as well as LDAP based Identity and Policy stores.

Most of these resources can be configured as shared or as local resources. This section covers how to configure these resources to ensure they remain synchronized and/or support access by multiple Sites.

Configuring the Identity Store

Both sites should share the same Identity store so that users from one site can seamlessly move from one site to the other. Alternatively, each site could have an independent identity store but they need to be synchronized. We'll only cover the steps for the former case here.

Since the Identity store resides in a Database, the Identity management products can themselves run as Active-Active. An instance of each Identity Server (for example, Oracle Internet

Directory) can run at each Site. The rest of this section covers how to configure an Identity Store that is shared between the two Sites.

To ensure that both sites are sharing the same store, first configure an external identity store for one site, then ensure that the second site is configured identically. The second site should not have to load or configure any additional information into the store.

In this topology, we configure both sites to share the same Oracle Internet Directory (OID) for example. The steps to do this are as follows:

1) First, configure one Site to set the Identity store to be OID. The detailed steps to do this are not covered here.
2) Ensure that the second site is configured identically. In particular:
    a. The OID Provider has been added to the list of providers for the Domain and has been set as `'SUFFICIENT'`
    b. The Provider details are configured so that both sites are using the exact same LDAP store.
    c. In particular, ensure that the User Base DN and Group Base DN are the same on both sites.
    d. If any additional users and groups were created (for example, an Administrators group) or any configuration was done with `ldapadd` when configuring the first Domain, this does not need to be repeated for the second Domain.
    e. Any Client side Role authentication **does need to be repeated** for the second Domain. For example if a Group was added under Global Roles in the security realm, this needs to be also created on the second site.

The provider details are listed in the table below. These values must be configured to be the same on both sites:

Parameter	Value	Description
Host:		The LDAP server's server ID
Port:		The LDAP server's port number (for example, 3060)

Parameter	Value	Description
Principal:		The LDAP user DN used to connect to the LDAP server (for example, cn=orcladmin)
Credential:		The password used to connect to the LDAP server
User Base DN:		Specify the DN under which your Users start (for example, cn=users,dc=example,dc=com)
Group Base DN:		Specify the DN that points to your Groups node (for example,cn=groups,dc=example,dc=com)
Use Retrieved User Name as Principal	Checked	Must be turned on
All Users Filter:	(&(uid=*)(objectclass=person))	Search to find all users under the **User Base DN**
User From Name Filter:	(&(uid=%u)(objectclass=person))	
User Name Attribute:	uid	

Configuring the Policy Store (OID)

Configuring a shared policy store is also essential to both sites providing the same experience to users on either site. The steps for configuring a shared policy store also involve configuring the policy store correctly for one site and then configuring the second site to use the already existing policy store. The complete set of steps for configuring the shared policy store for both sites is given below. The server in this example is Oracle Internet Directory:

Set a node in the Directory server

First, create a uniquely named node in the Directory server where the Policy store will be loaded. An example of how to do this is below:

1) Create an LDIF file, named myjps.ldif for example, with the following contents:

```
dn: cn=jpsroot_wc

cn: jpsroot_wc

objectclass: top

objectclass: OrclContainer
```

2) Load the information into LDAP using the ldapadd command as follows:

```
OIDHOST> ORACLE_HOME/bin/ldapadd -h ldap_host -p
ldap_port -D cn=orcladmin -w password -c -v -f myjps.ldif
```

Reassociate the Domain Policy store for the first Site

The first time the Domain Policy store is relocated from being local to residing in a Directory Server, Policies and Credentials must be migrated or created in the Directory Server. To reassociate the policy store for the first Site, use the WLST shell as follows:

1) Connect to the Administration Server of the first Domain:

```
WLST> connect("AdminUser","AdminUserPassword",t3://hostname:port)
```

2) Execute the Reassociate Command:

```
WLST>
reassociateSecurityStore(domain="MyFarm",admin="cn=orcladmin",
password="orclPassword",ldapurl="ldap://LDAPHOST:LDAPPORT",servert
ype="OID", jpsroot="cn=jpsroot_wc")
```

Note that in the command in Step 2, for the parameter 'domain', do NOT use the domain name of either of the Sites. Instead, use a name that is suitable as the name for all the sites since this will be a policy store that will be equally shared.

Reassociate the Domain Policy store for the second Site

The second site needs to be configured to point to the same policy store as the first site but does not need to reload the data. This is accomplished with the `join="true"` option of the policy store re-association. Using the WLST shell:

1) Connect to the Administration Server of the second Domain:

```
WLST> connect("AdminUser","AdminUserPassword",t3://hostname:port)
```

2) Execute the Reassociate Command:

```
WLST>
reassociateSecurityStore(domain="MyFarm",admin="cn=orcladmin",
password="orclPassword",ldapurl="ldap://LDAPHOST:LDAPPORT",servert
ype="OID", jpsroot="cn=jpsroot_wc",join="true")
```

The Domain parameter should be the same one that was used in the previous step above for the first Domain.

Configuring the Policy Store (DB)

Configuring a shared policy store can also use the Oracle Database as the policy store. For the sake of completeness, we include the steps for setting up a shared policy store in the Database. The steps for configuring a shared policy store also involve configuring the policy store correctly for one site and then configuring the second site to use the already existing policy store:

Create the OPSS schema and datasource

Use the Repository Creation Utility to create an OPSS schema in the database. This schema will be used by both sites to store the shared policy store.

Create an identical datasource at each site which references the newly created schema. The datasource should be created as non-XA and with no global transaction support.

Reassociate the Domain Policy store for the first Site

The first time the Domain Policy store is relocated from being local to residing in a Directory Server, Policies and Credentials must be migrated or created in the Directory Server. To reassociate the policy store for the first Site, use the WLST shell as follows:

1) Connect to the Administration Server of the first Domain:

```
WLST> connect("AdminUser","AdminUserPassword",t3://hostname:port)
```

2) Execute the Reassociate Command:

```
WLST>
reassociateSecurityStore(domain="MyFarm",servertype="DB_ORACLE",da
tasourcename="mydatasource",jpsroot="cn=jpsroot_wc")
```

Note that in the command in Step 2, for the parameter 'domain', do NOT use the domain name of either of the Sites. Instead, use a name that is suitable as the name for all the sites since this will be a policy store that will be equally shared.

Reassociate the Domain Policy store for the second Site

The second site needs to be configured to point to the same policy store as the first site but does not need to reload the data. This is accomplished with the `join="true"` option of the policy store re-association. Using the WLST shell:

3) Connect to the Administration Server of the second Domain:

```
WLST> connect("AdminUser","AdminUserPassword",t3://hostname:port)
```

4) Execute the Reassociate Command:

```
WLST>
reassociateSecurityStore(domain="MyFarm",servertype="DB_ORACLE,dat
asourcename="mydatasource",jpsroot="cn=jpsroot_wc",join="true")
```

The Domain parameter should be the same one that was used in the previous step above for the first Domain.

Configuring the Java Object Cache

Oracle Webcenter Portal uses the Java Object Cache to improve performance. The cache is a write-through cache used to store local objects as well as retrieved external data. Processes read from this in-memory cache instead of reading from the database, in most cases improving performance. The cache is also used to synchronize the cached Metadata Repository among different Servers.

In a typical configuration, the Java Object cache is configured as a distributed cache among all members of the cluster. These synchronized caches allow for the sharing of objects.

In an Active-Active scenario, two separate caches are writing to the same database. This allows for the possibility of either of the sites ending up with stale data. There are only two solutions to this:

1) Disable the Java Object Cache

   The Java Object Cache is not required from a product functionality perspective. Disabling the cache will ensure that both sites are in sync since they now both must retrieve their data from the same database.

   However, this can have an unnecessary performance impact.

2) Configure a Distributed Cache that spans both clusters

   By having both clusters participate in the same distributed cache, all the inconsistency issues disappear.

   This does mean that there is additional network traffic between the two sites, in addition to the replication traffic. However, in our tests it appears that Java Object Cache traffic only contributes a small overhead on the already existing replication traffic.

We recommend the second option. The rest of this section provides the steps for configuring a multi-cluster distributed Java Object Cache.

Plan Java Object Cache Addresses and Ports

Each member of the cluster needs to have a listen address and port. If there are multiple network interfaces on the machine, choose a listen address that is assigned to that interface. The port can be any unused port. Here, we choose to put the Java Object Cache at port 9988.

Each member of the cluster is configured with the listen address and port of all other members of the cluster. If the clusters are ever either relocated or expanded, the java cache will need to be reconfigured to match the new configuration.

Update the javacache.xml file

The Java Object Cache is enabled by modifying the javaacache.xml file and restarting the servers. The following steps must be done for each cluster.

1) Go to the Domain Home of the Administration Server of the Cluster. In a distributed configuration there might be multiple Domain Homes on many machines. The only one we are concerned with is in this section is the Administration Server Home. All changes made here will be automatically propagated to the other Homes.
2) Specifically, go to $DOMAIN_HOME/config/fmwconfig/servers/<server_name>
3) For each server in the cluster, we need to modify the javacache.xml file found in the directory specified in Step 2.
4) Back up the old javacache.xml.
5) Create the new javacache.xml as follows:

```
?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<cache-configuration
xmlns="http://www.oracle.com/oracle/ias/cache/configurati
on11" max-objects="5000" max-size="10" private="false"
cache-dump-path="jocdump" system="false" clean-
interval="60">

    <communicationService enabled="true">

        <v2 ssl-config-file=".sslConfig" init-retry="300"
init-retry-delay="2000" enable-ssl="false" auto-
recover="false">

            <packet-distributor enable-router="false"
startable="true" dedicated-coordinator="false">

                <listener-address host="thishost"
port="9988" ssl="true"/>
```

```
                <distributor-location host="thishost"
port="9988" ssl="true"/>
                <distributor-location host="host2"
port="9988" ssl="true"/>
                <distributor-location host="host3"
port="9988" ssl="true"/>
                <distributor-location host="host4"
port="9988" ssl="true"/>
            </packet-distributor>
        </v2>
    </communicationService>
    <diskCache size="10" ping-interval="60"/>
    <logging override-parent="false"
location="javacache.log" default-level="SEVERE"/>
    <dms enabled="false"/>
</cache-configuration>
```

6) In the above file, replace "thishost" with the address of the local machine. The other hostnames (host1, host2, etc) are the addresses of all the other hosts across both clusters. Ensure that all hosts in the extended cluster (all machines across both sites) are listed. The listener-address element is only needed if the machine has multiple interfaces.

7) Copy the same file to all directories of all the servers in the cluster. The only thing that should be different is the listen-address. The rest of the file should be identical in all the server directories.

Finally, after this has been done, repeat the above steps for the second, remote cluster. That is, go to the Administration server Domain Home of the remote Domain and modify all the javacache.xml files. They should also look identical to the files on the first cluster with the exception of the listen address.

Start or Restart all the servers to effect the changes and distribute the xml files to all the config directories.

Validate the Java Object Cache is running

In order to check that the Java Object Cache has been configured correctly and is running on all the servers, we can run a utility called Cache Watcher. This utility is another instance of a Java object Cache and can report on its peers.

The utility can be run on any machine in the cluster. It uses an existing javacache.xml file for its configuration.

To run Cache Watcher:

```
$ java -classpath
$FMW_HOME/oracle_common/modules/oracle.javacache_11.1.1/cache.
jar:$FMW_HOME/oracle_common/modules/oracle.odl_11.1.1/ojdl.jar
oracle.ias.cache.CacheUtil watch –
config=$DOMAIN_HOME/config/fmwconfig/servers/<server_name>/jav
acache.xml
```

where FMW_HOME is the directory where Fusion Middleware was installed and <server_name> is the name of a server running on this machine.

The utility will start and produce output as follows:

```
INFO: JOC is initialized from oracle.ias.cache.CacheUtil.main,
ver=null, distribute=true, vid=39, coordinator=0, discover
list=[[host1:9988] segID=1, SSL, [host2:9988] segID=1, SSL,
[host3:9988] segID=1, SSL, [host4:9988] segID=1, SSL]

cache>
```

To list all the visible caches, type 'lc' for 'list caches':

```
cache> lc

View Id: 39

My ProcessId: 14630_host1

Distributor Table:

     ip/port                 position  processId

============================================================

#1.  10.210.30.245:9988      0         1900_host1  Coordinator

#2.  10.210.30.247:9988      2         19762_host2

#3.  10.210.30.247:9988      2         19769_host3

#4.  10.210.30.248:9988      3         9770_host4
```

Here we can see if all the other hosts are visible. If they are, then the Java Cache has been configured correctly. If any hosts are missing, check the configuration file for any errors. The server may also need to be restarted in order to pull the javacache.xml file from the Admin Server Domain Home.

Configuring other External Services

Oracle Webcenter Portal relies on connections with other external services. These are either services within Oracle Webcenter Portal Suite such as Portlets or Discussion Servers or external services such as SOA or Presence Servers.

Both Sites can either share most external services or, alternatively, each Site can also have its own local set of Services. The options and best practices for each of the services is covered in this section.

Configuring Portlet producers

Portlet producers should be local to each Domain. The address of the Producers will usually be an address that allows load balancing to a cluster of Producers. This address can be the local HTTP Server or Load Balancer at the Site.

However, since the configuration of the Producer address is stored in the Metadata Repository and both sites share the same Metadata Repository, the configuration of the two sites must be identical. This means that the Producer address for the two sites is the same, although we want each one to be local. The way of resolving this is to have separate DNS servers at each Site in order to have the same address map to multiple hosts. (If that is not possible, then local /etc/hosts file can also be modified to provide different mappings for each server.) Configuring the Providers for both sites then involves the following steps:

1) Configure Site 1 so that the host ProviderHost.mycompany.com (for example) maps to the local HTTP Server or Load Balancer.
2) Configure Site 2 so that the host ProviderHost.mycompany.com (for example) maps to the local HTTP Server or Load Balancer.
3) Ensure that the Providers on both sites are up and running.
4) On either Site 1 or Site 2 configure the Provider address either through Fusion Middleware Control or through WLST:
    a. Connect to the Admin Server of either Site 1 or Site 2
    b. Issue the command:
    ```
    registerOOTBProducers(producerHost='ProviderHost.mycompany.com',producerPort=80, appName='webcenter')
    ```
    c. For the Port use the port number of the local load balancer.

5) Step 4 only needs to be done once at either of the two sites in order to configure both sites.

Configuring SOA worklist

The Oracle Webcenter Portal may also require a SOA installation in order to provide Worklist services. In order to retain consistent user mapping, however, the SOA installation should be mapped to the same Identity store as Oracle Webcenter Portal.

For a shared installation, the location of the SOA install should be configured once, on either Site 1 or Site 2. Both Sites will be able to equally access the SOA services.

Configuring Content Management

Unlike SOA services, the location of the Oracle Webcenter Content services must be shared.

For a shared installation, the address of the Content services should be configured once, on either Site 1 or Site 2. Both Sites will be able to equally access the services.

Content servers cannot run as Active-Active across the two Sites. The reason is that multiple Content servers accessing the same database must also synchronize additional metadata files on disk. It is assumed that separate Content servers on the two sites would not be able to have a shared disk location.

Configuring other external services

For configuring all external services, the following guidelines should be followed when trying to determine if external resources should be configured locally – with an instance at each Site - or should be shared by the two Oracle Webcenter Portal installations.

Stateless services

For external services that hold no application state, the services can be either shared or local. This would include, for example, an SMTP gateway or an external HTTP resource. In this case, the service might still be local for performance reasons.

In the case of local resources, care should still be taken to ensure that the address of the local resource is either stored locally or, if stored in shared configuration, can resolve to a local address.

Services such as Discussion Server or Wiki server are, in this architecture, stateless and can be configured locally just like the Portlet Providers. Although they do hold state in the Database, the Database is already a shared resource.

Stateful services

Stateful services must, in most cases, be shared. In this architecture, the most important stateful service is the Database which is shared by the two Sites. The Identity and Policy store may also undergo changes during runtime and are also shared.

If a stateful resource must be local then care should be taken so that the two local resources are synchronized with each other so that inconsistencies do not arise if a user migrates from one Site to another.

Managing Filesystem resources

In an Oracle Webcenter Portal installation, no disk artifacts need to be shared by the two installations. Although, each installation maintains its local logfiles, for example, no runtime state is persisted to disk. So, each site can have its own binaries and its own domain configuration files.

If a change is made to either Domain, then the same change should be made to the second Domain of course. But this can be accomplished manually and does not usually occur during runtime.

Any Custom Webcenter Portal applications which persist anything to local disk will need to ensure that these files do not need to be synchronized between the two sites.

Managing caches

Any local cluster caches need to be disabled or synchronized across the two sites in order to avoid generating inconsistent views. In a write-through cache there is no danger of actual data inconsistency but, in some cases, users may be viewing and acting on an older view of the underlying data.

Oracle Webcenter Portal uses the Java Object Cache (JOC) for storing object state in memory. This is a cluster-wide cache but is not synched across two remote clusters. In this case, we create a cache that spans both clusters. If that were not possible the cache might have to be disabled.

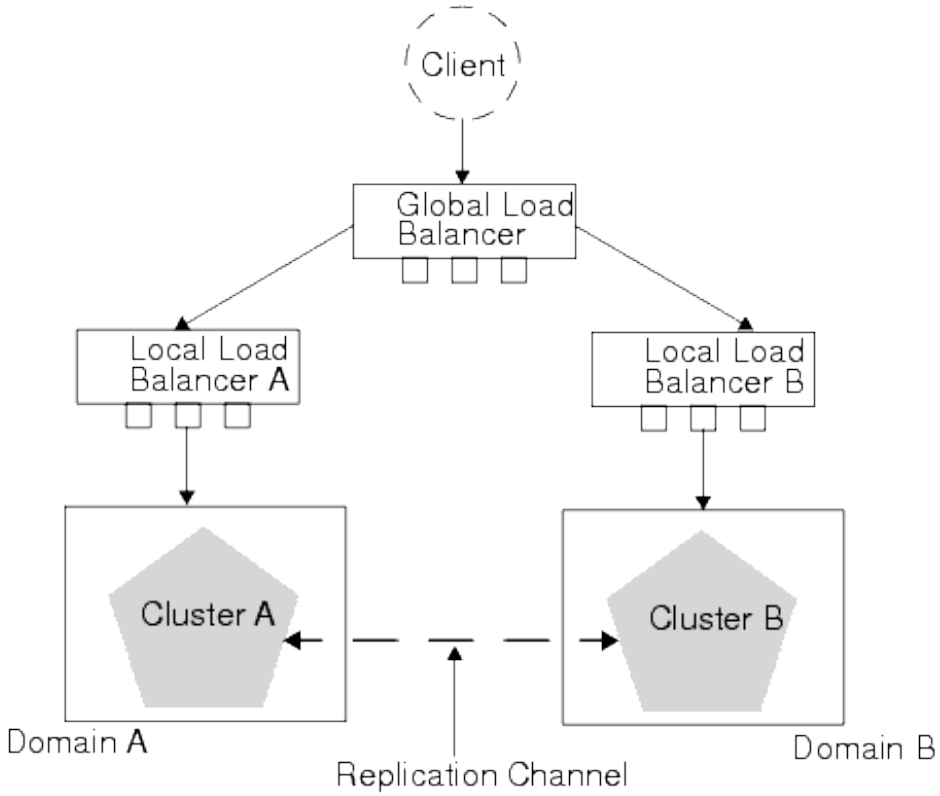
Configuring and Managing the Load Balancers

The diagram below shows the configuration assumed here. There are other topologies possible. The main requirements for the Load balancers are that:

1) A local Load Balancer exists for each site to receive traffic from a Global Load Balancer. The local load balancer then distributes local traffic among the local HTTP servers or local Application servers.

2) A Global load balancer exists that can route users accessing either site. The routing takes place based on a client rule such as their originating IP address.

Each of the local load balancers is routing requests to two or more HTTP servers. This is the recommended Enterprise Deployment Configuration and allows the use of the WebLogic-plugin to dynamically balance requests across the cluster.



Configuring the Local Load Balancer

The local Load Balancer is receiving requests from the Global load balancer and sending requests to the HTTP Servers. The HTTP servers themselves should be configured according to the instructions provided in the *Enterprise Deployment Guide for Oracle WebCenter Portal*.

Configuring the Global Traffic Manager

The Global Traffic Manager next needs to be configured to route to each of the Load Balancers configured on the sites.

Configuring and Managing Session Replication

One of the key features of an Active-Active configuration is the ability to seamlessly failover users from one site to another. This includes maintaining the same session at one site as at the other. .

Here, we consider two options for managing session replication:

1) **MAN Replication**

   In MAN Replication, session state is sent synchronously to a server on the remote cluster. This has the advantage of providing no loss of session state if configured correctly. The disadvantage is a reliance on a reliable network between the clusters.

2) **No Remote Replication**

   Alternatively, loss of sessions can be accepted as the cost of failover. This has obvious disadvantages which are discussed in more detail later in this paper. The advantage is that there are no additional requirements on the system.

Not considered here are the options of

1) **Asynchronous DB Replication**

   Asynchronous DB Replication relies on local in-memory session replication supplemented by an asynchronous flush of session state to the database. The advantage is that no network is required between the two sites. The disadvantage is that session state is not guaranteed to be current. This configuration may be discussed in the future but is out of scope for this paper.

2) **Synchronous Database Replication**

   This involves both sites using synchronous Database replication and sharing their session tables in the database. This configuration may be discussed in the future but is out of scope for this paper.

3) **Single Domain Replication**

   This involves using only one Domain across the two sites instead of two and relying on in-memory replication from site to site. This configuration is not discussed here.

MAN Replication

If the two sites are connected via a fast network, then direct Cluster to Cluster session replication can be used. This will be the case in a Metropolitan Area Network (MAN) or Local Area Network (LAN) environment.

In this configuration, sessions created on a server on Site 1, for example, are immediately transmitted to a secondary server on Site 2. Likewise, sessions created and updated on Site 2 are sent to Site 1. A user can failover at any time from Site 1 to Site2 and will be directed to the session state on the existing Secondary.

If network issues arise between the two clusters, then replication will again default to local replication. More detail on this is provided on the section on MAN Replication best practices.

Configuring MAN Replication

This section will walk through the steps to configure a cluster on Site 1 to replicate session state to and from a cluster on Site 2. If you need to replicate more than one cluster – for example a Cluster of Oracle Webcenter Spaces and a Cluster of a Custom Portal application then most of the steps here will have to be repeated.

Enabling Trust

Before beginning, security must be configured between the two remote clusters. There are two options to do this, enabling Global Trust or setting up Trusted users for Trust. If the latter option is chosen then the two Domain names must NOT be identical. This is the case because users are identified also by their domain name.

Both options are documented in FMW documentation. Here we cover the steps to enable Global Trust between the two domains:

1) On the first Domain, select the Domain name from the left-hand Pane.
2) Go to **Security->General** and click on **Advanced** to show more options.
3) In the **Credential** text field, enter a password for the Domain.
4) Save the changes.

Then repeat the above steps in the second Domain. Ensure that the same password is used on both Domains. These steps are sufficient to enable Trust between the two domains.

Configuring Network Channels

In a default configuration, each managed server is configured with one channel – the Default channel. This channel is used for all cluster broadcast messaging, for replication messages and as the endpoint for client HTTP requests. In order to have greater control over cluster replication, including the ability to monitor it and control it, it can be configured to occur on a separate dedicated channel.

In the case of replication across a MAN, the machines involved can also optionally be configured on a separate network interface. This section provides the steps for configuring a replication network channel for the managed servers.

Note that configuration of a custom replication channel is optional. Replication can occur over the default channel as well.

For a specific managed server on either Site, configure a network channel as follows, using the Administration Console:

1) After selecting **Environment->Servers** from the left-hand pane, select the specific Server.
2) Select the **Protocols** tab and then the **Channels** tab
3) Click on **New** to configure a new Channel
4) Give the Channel a name, for example 'RepChan'. This same name will be used when creating all the channels on the servers in this Cluster. Specify the **Protocol** as t3 and click **Next**
5) Configure a **Listen address** and **Port** for this Channel. The Listen address can be the same as the default channel or can also be listening on a different interface. Ensure that the Listen address/Port combination however is unique on this machine. Click **Next**.
6) Ensure that only the box **Enabled** is checked. Uncheck the **HTTP Enabled** box and any other options. We want to restrict how this Channel is used. Click **Next.**
7) Optionally configure Client certificates for this channel. Then click **Finish**.

The above steps should be repeated for all of the Servers in the Cluster on both Sites. Keep the following in mind:

- The Channel Name should be the same for all Servers in a Cluster on a specific Site.
- The Channel Name can be the same on both sites but this isn't required.
- Each Managed Servers Channel Listen Address/Port should be an address that is reachable by members of the Remote cluster. The Channel is the address where remote replication calls will be received.

Enabling MAN Replication

Once the Channels are configured, MAN Replication can be enabled between the two clusters.

For each Cluster, configure the Cluster as follows, using the Administration Console:

1) Select **Environment->Clusters** from the left-hand pane
2) Select the Cluster that will be enabled for MAN Replication
3) Select the **Configuration->Replication** tab

4) For **Cross-Cluster Replication Type**, select 'MAN (synchronous) HTTP Session State Replication'

5) For the **Remote Cluster Address** enter the address of the Remote cluster in the following format: `t3://host1:port1,host2:port2` ensuring that all the servers in the remote cluster are selected. For the Host and Port use the address of the remote Server's replication channel.

6) In the Field **Replication Channel**, enter the name of the replication channel that was configured on each server in this Domain (not the remote Domain)

7) Click **Save** and Restart all Servers.

The above procedure should be followed on both sites.

Monitoring MAN Replication

Once configured, MAN Replication can be monitored by monitoring the replication channel that was configured on each server. For example, to view the network activity for a specific Server:

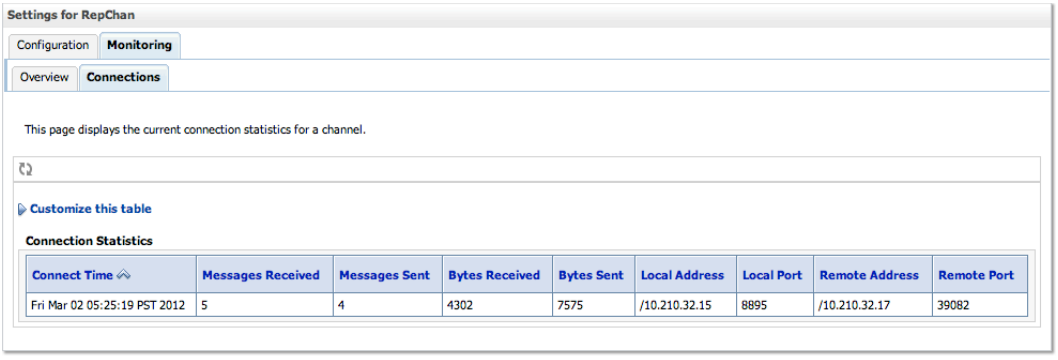
1) Select the Servername from **Environment->Servers**

2) Select the **Protocols** tab and then the **Channels** tab

3) Select the replication channel from the table

4) Select the **Monitoring->Connections** tab

If the remote servers are up and running, this table will show the local and remote replication addresses along with a summary of how much network traffic is being sent back and forth.



In the above screenshot, one connection is being made by this network channel. In this case, 10.210.32.15 is the local server and 10.210.32.17 is a server in the remote cluster.

To confirm, also check the Cluster to ensure that a remote server is acting as the replica. To do this:

1) Select 'Clusters' from the left-hand pane in the Admin Console.
2) Select the name of the Cluster.
3) Select the **Monitoring** tab and then the **Failover** tab.

The table will show the location of primary sessions and session replicas both in the local cluster and the remote cluster.

Replication can also be monitored by noting the network traffic between the two sites. For more information on this, consult Appendix A of this document: "Network Monitoring tools"

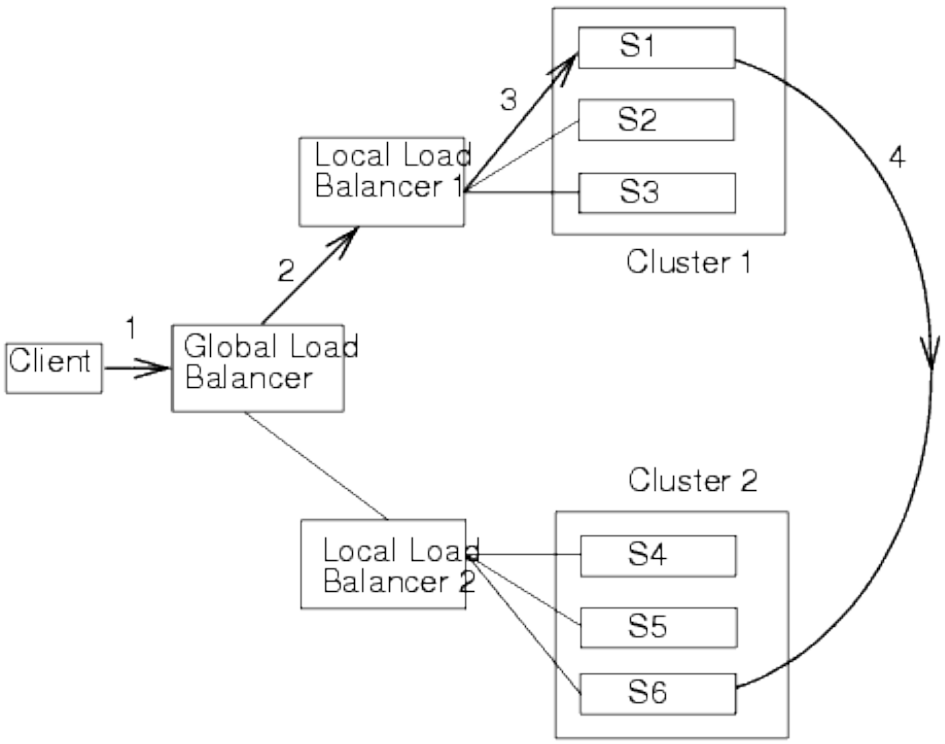
Best Practices for MAN Replication runtime

This section walks through MAN Replication in a bit more detail including how servers should behave during runtime and tuning the configuration.

MAN Replication runtime behavior

During runtime, users will be directed to one of the two Domains. Subsequent requests will also go to that same Domain. The user's session state is stored in memory. Instead of a local secondary, however, the secondary server is a server in the other Domain.

This is illustrated below. The user is connecting to Cluster 1 where session state is being held. But session state is also being replicated to another server that is part of Cluster 2.



All Session state is sent synchronously from Server S1 to Server S6 in the remote cluster. If a custom replication channel was configured then all replication data is sent and received through that channel.

Determining bandwidth

In MAN Replication, session state is transmitted from Cluster 1 to Cluster 2 for users on Cluster 1 and from Cluster 2 to Cluster 1 for all users on Cluster 2.

The amount of required bandwidth will depend on three factors:

1) How much state does each user generate?
2) How active is each user?
3) How many users are there?

We'll examine each of these three factors:

Determining Session size

The amount of data transferred as altered session state will depend on the application. For Webcenter/ADF applications this may be anywhere from 100k to 1MB. A reliable method of determining user session size is to examine the amount of data transferred across the network with one user on the system.

The appendix A of this paper "Network Monitoring Tools" provides more detail on how to examine the network to determine how much session state is being transferred.

Determining User Activity

The activity levels of user is how often users are actually generating state. State is generated whenever users navigate to different pages for example. It is not generated if users are idle or if users are waiting for a background process to complete – such as a load to a backend database.

User activity can be determined by examining the usage patterns of a test sample of users.

Determining User Load

The User Load is the total number of users active on the system. It does not include users who have abandoned their sessions. It does include, however, all users across **both** sites.

Calculating bandwidth

Although it is recommended that the above values be derived empirically, by actual testing, we can get a sense of how much bandwidth would be required for an example environment.

Assuming that Session size (S)= 500k and User Activity (A) = 4 pages/minute and User Load (L) = 100 users then:

```
Bandwidth (B) = S * A * L

= 200 MB/minute = 3.3 MB/sec

= 27 Mbps
```

This is in the range of a T3 line.

Determining Latency

Latency is the time taken for packets to travel from one Cluster to another. This can be a factor of many things including the length of the path between the sites and any layers in between.

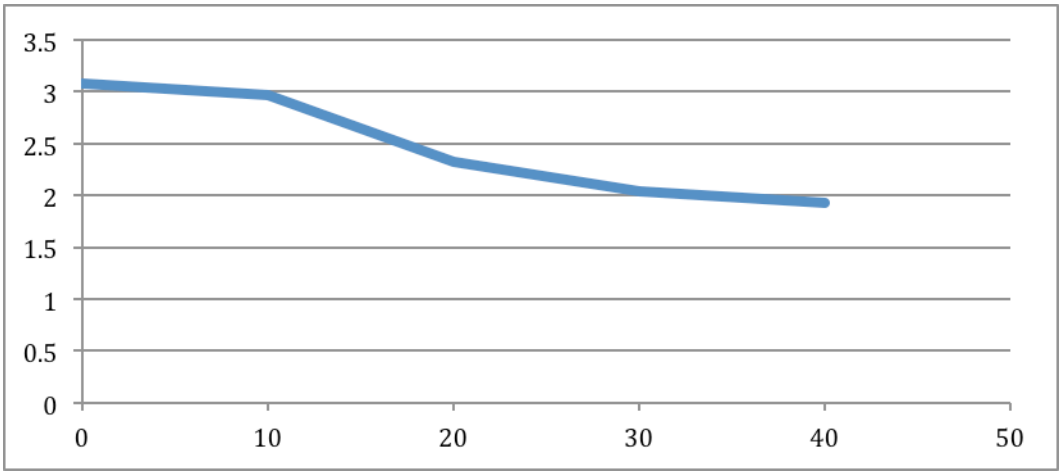
Typically latency is determined by using utilities such as `traceroute` or `ping` to send test packets from one site to another.

The latency or round-trip time (RTT) has a direct effect on the response time that any one user experiences when accessing the system. The effects of high latency can be seen even with only one user on the system.

The graph below shows throughput in Webcenter Spaces (in request/sec) as a function of the measured RTT between the two sites (in milliseconds). In this test, one user repeatedly

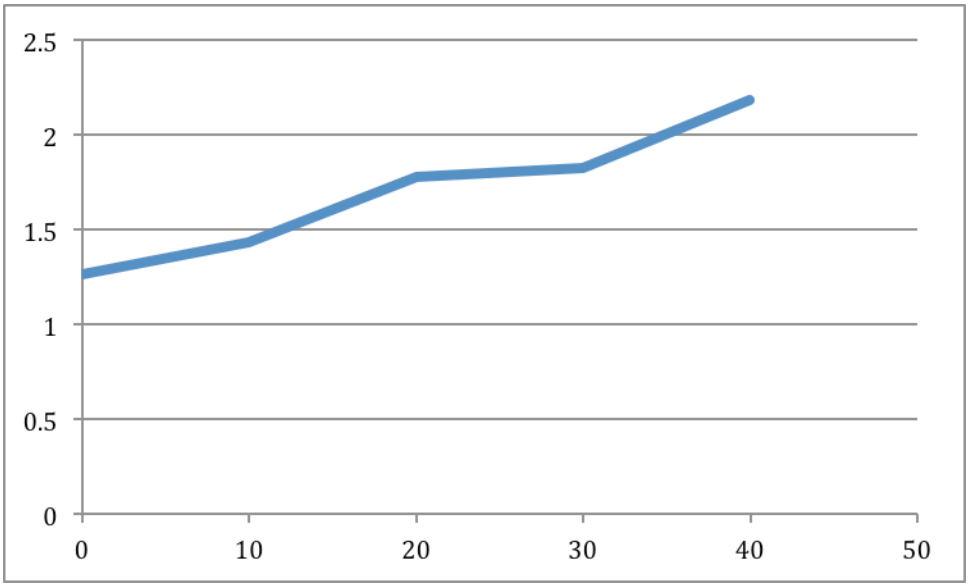
performed 100 runs. Each run consisted of navigating through different Spaces and then updating a Discussion Forum.

There were both large and small requests but the throughput is calculated as average requests/second. A system that started out at 3 requests/sec is 1/3 less efficient when the network latency is above 25ms.



The graph below shows how measured response time (in seconds) goes up rapidly for one user loading a specific page in Webcenter Spaces as a function of the measured RTT (in milliseconds) between the two sites. The page was loaded 100 times for each measure of network RTT time. This is a particularly large page – even with no latency the page still takes over a second to load. So, much of the wait time has nothing to do with waiting on replication.

Nevertheless, by the time we reach 25ms of RTT, we have increased the latency of this one page by 50%. This increase is solely due to the added wait time for replication to occur.



Based on the above tests and others like them, it is recommended that the RTT between the two sites not exceed 25ms as an outer limit and less than 10 ms ideally. Greater RTTs than that are likely to have a severe performance effect on the system.

Setting the TCP Socket Buffer

Knowing the bandwidth and the latency allows for the optimal TCP socket buffer size to be set. This allows the TCP window to be large enough so that the window's inability to accommodate the amount of data in the network at any given time does not limit performance.

For values of B=44 Mbps (T3) and RTT=20ms then,

$B*D = 44 \times 10^{6}\ b/s * 20 \times 10^{-3}\ b = 880 \times 10^{3}\ b = 880\ kb$

This is an example of a Long Fat Network. Window scaling should be used to set the TCP Window size appropriately.

Packet loss and jitter

Tools such as `iperf` should be used to determine if there is any packet loss or any appreciable jitter on the network. This paper assumes that the network has zero packet loss and negligible jitter. If this is not the case, these problems should first be addressed.

For more information consult the appendix A of this paper "Network Monitoring Tools"

Best Practices for MAN Replication failover

There are several minor and major types of failover scenarios. We'll briefly cover them here:

Minor failover refers to failures of individual redundant components of a specific site. In these cases, the site is still active but has been damaged in some way but it is also able to recover seamlessly.

1) **Primary server failure**

   If the Primary server fails, then a new local primary will be appointed. This new Primary will retrieve its session state from the remote Secondary. Users will not notice this failover as their state is retrieved across the network.

2) **Secondary Server failure**

   This refers to failure of the remote secondary. A new remote secondary will be appointed which will begin accepting state from the Primary. Users will not notice this.

Major failover refers to a loss of critical function at one site, complete failure of a site, or critical network failures.

1) **Full Site failure**

   In the case of full site failure, users will all be directed to the surviving site. Since their secondary servers are already at the remote site, users should experience little or no interruption in service. More detail on how this occurs as well as configuration for this failover can be found in the section on "Configuring and Managing Load Balancer configuration"

2) **Local Load Balancer failure**

   In the case of a site failure where the local load balancer has failed but the two sites can still communicate with each other, then all users will be moved to the active site but their secondary servers will be on the remote, inaccessible, site.

   There is nothing wrong with this configuration since state can still be retrieved and accessed in any failover scenario.

3) **Network failures**

   Network failures can occur anywhere but in this case we are most concerned with the failure of the network between the two sites. If this is the only failure, then the global load balancer will still send users to both, active sites. In this case, however, replication will be only local. More detail on this scenario follows in this section.

Network failure scenario

During runtime, session state is being created or managed on the local Primary and then sent to a remote Secondary. This happens synchronously. That is, the user session will wait until the receipt of the state on the remote secondary has been acknowledged.

A long delay in the network can cause this waiting time to be significant. If it becomes too severe, either because of network congestion or because of network failure, the user may continue to wait indefinitely. With lots of users on the system all waiting for the remote network requests to complete, the system may become completely unusable.

To manage this, timeouts can be configured so that the system can respond appropriately to network failures. In this case, replication still continues but occurs to a local server instead of a remote server.

Likewise, the system is capable of re-establishing the connection automatically when the remote site is again available.

Configuring network timeouts

The default timeout period for a server to wait for its remote peer to respond is 240 seconds, or 4 minutes. During this time, all clients will be waiting for a response from the server since the session replication must occur and it is synchronous.

The two values which control this timeout are `ServerMBean.PeriodLength` and `ServerMBean.IdlePeriodsUntilTimeout`. By default these are set to 60000ms and 4 respectively.

The first value is the interval of heartbeats between a server and its peer. The second value is the number of heartbeat periods after which a server is considered unreachable. So, in the default configuration:

```
Period Length * IdlePeriodsUntilTimeout = 60 sec * 4 = 4
minutes
```

These parameters can be set in the Admin Console:

1. Select **Environment-Servers**
2. Click on the specific Server Name
3. Click on the **Tuning** tab
4. Click on 'Advanced' at the bottom of the screen
5. Both `Period Length` and `Idle Periods until Timeout` can be configured here

This needs to be done for all Servers.

Finally, ensure that Replication Timeouts are enabled. This can be found in Cluster Name -> **Replication** -> Advanced. Check the **Enable Replication Timeout** checkbox.

If the total timeout period is set too high, then users may experience an unreasonably long delay when there is a network failure. On the other hand, these network failures should hopefully be infrequent events.

If the total timeout period is set too low then high but temporary network congestion may trigger a failure. The system will then revert to local replication.

Configuring health checks

After a network failure, both sites will start operating independently, replicating their session state to local servers. Once the network link is restored between the two sites, they will automatically begin replicating remotely again.

To configure this, go to Cluster Name -> **Replication** -> Advanced in the Administration Console. Set the parameter **Inter-Cluster Comm Link Health Check Interval**. The default value for this parameter is 30000ms or 30 seconds. This is how often the site will check to see if the remote site is again available. The default should be adequate for most cases.

No Replication between Sites

Another option is to not configure any replication between the two Active-Active sites. The impact of this is, of course, that after a site failover all users will have lost their session state.

The impact of lost session state depends on the application. In the case of Webcenter Spaces and for many ADF applications, lost session state will primarily affect three things:

1) Authentication
2) Open Windows
3) Unsaved entries and configuration

All other state is sent to the database where profile information and general configuration is stored.

Authentication

A session matches a session cookie for authentication. If that session is no longer available, then the user is no longer logged into the system. The first and most obvious impact of losing session state is that the user is not logged on and is returned to the logon screen.

After a site failover, all users will have to re-logon to the system. No data is lost in this event but users will be inconvenienced.

Open Windows

After logging in, the user is taken to the initial application page. All scoped variables have been lost including the current page and the page layout. Getting back to where a user was before the session was lost may involve some work in navigation.

Unsaved entries

Anything that appears on a page, or any selection made but not saved will be lost. In the case of composing a blog entry or editing a wiki, for example, this may be a substantial bit of information. Once saved, the information is stored in the database. Until then, information written on a page is volatile and will be lost if the session is lost.

This may represent actual data loss to a user with many unsaved changes.

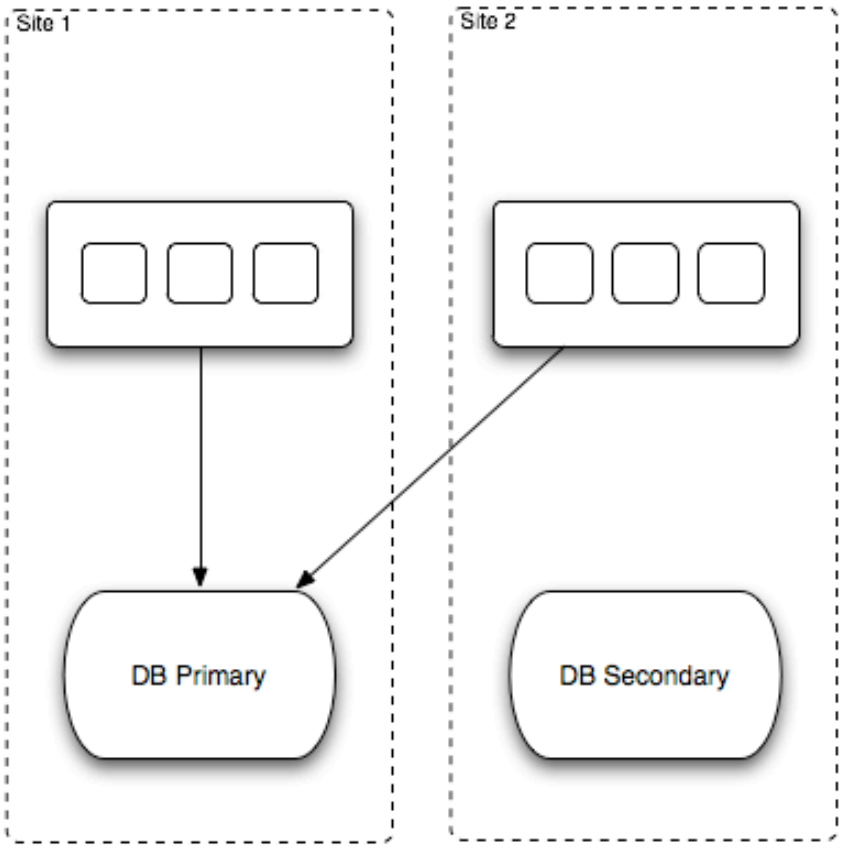
Configuring and Managing Database Access

As mentioned previously, the applications at both Sites are writing to and reading from the same Oracle Database. The configuration of this database can be an Active DataGuard installation where the primary resides at one Site and the standby instance resides at the secondary Site.

Both Sites should be configured with Active GridLink datasources that point to the primary and are configured correctly for DataGuard failover. The details for doing this are not covered here. Please refer to the Oracle MAA whitepaper: *Oracle WebLogic Server and Highly Available Oracle Databases: Oracle Integrated Maximum Availability Solutions*

Database Configuration

The diagram below shows schematically how the Database is configured across the two Sites. For one Site, the Primary Database is local. For the other Site, connections to the Database must go across the network that connects the two Sites.

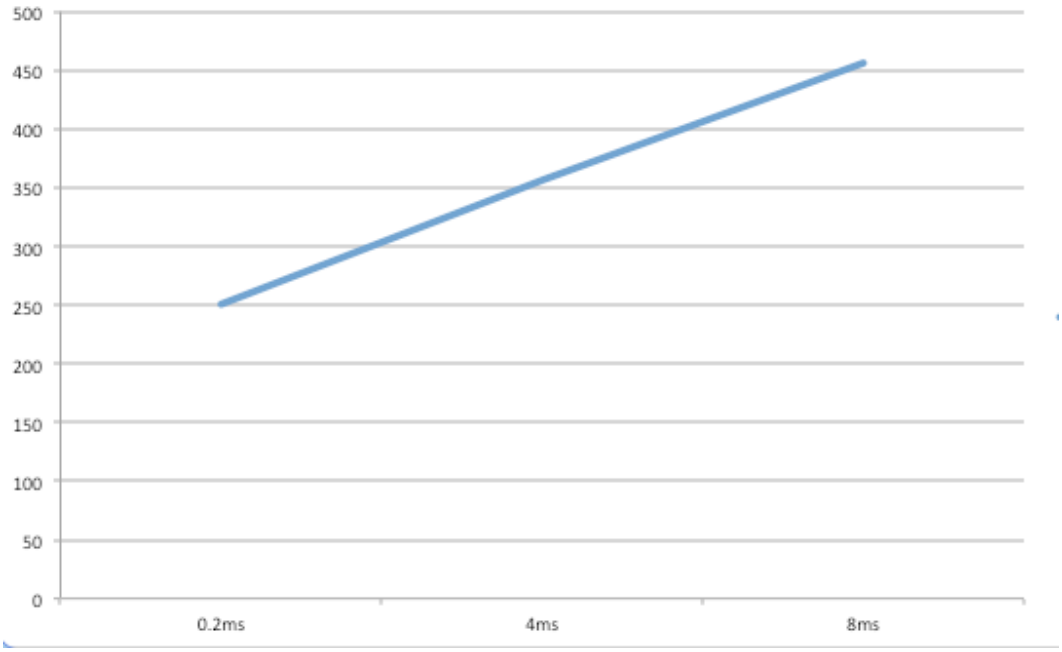


For the remote Site to function satisfactorily, the network connection between the remote site and the active database should be as reliable as possible.

Before beginning to design an architecture to meet specific performance criteria, a typical user load should be tested on a cluster of servers and tested against a database simulated to be remote. This will help identify whether the constraint is on throughput/bandwidth or latency of the remote system.

To understand the effect of latency only on an Oracle WebCenter Portal environment, the following test was performed. One user posted on a Blog in a WebCenter Space. The entry was approximately 200 chars. The latency was measured for the user's POST operation. In this case, latency means the time from then the POST was sent until the first bits of the response are returned from the server.

The graph below shows the results:



The vertical axis shows the User Response Time in milliseconds. The horizontal axis shows the measured RTT between the machine hosting the WebCenter Portal managed servers and the machine hosting the Oracle Database.

As we move the database from 0.2ms away to 8ms, the measured Response Time for the request almost *doubles* from 250ms to almost 500ms. How this translates to user experience depends on the number of database requests in a session and the activities of other users on the system. Nevertheless, even on a quiet system, with no other users and sufficient memory and processing power, the effect of placing the database too far can have a severe impact on the response time.

Lifecycle Management

Although we have covered the initial creation of the two Active sites, just as important is the ongoing maintenance of the sites. This includes day-to-day configuration management as well as topology changes as well as patches and upgrades.

In this section, we try to cover some of the recommendations for keeping the sites synchronized.

Configuration Management

The sites need to stay synchronized in their configuration. This is especially important in the case of servers and their configuration, applications and their configuration and the replication configuration.

The sites are not identical in that the config.xml file of both sites may not match exactly – for example, the sites may have different machine names or domain names – but they should match as much as possible. One way to track diverging sites is to follow the practice of:

1) Compare the config.xml file of both domains and note down the differences
2) Repeat step 1) regularly and note if any new differences have appeared

This will be likely caused by someone making a change to one domain – tuning changes for example - and then forgetting to make the same change to the other domain. Try to reconcile the changes if possible.

Patching and Upgrades

Any changes to Applications must also be synchronized across the two sites. The two most important changes to occur to applications are patches and upgrades.

Because the two sites share the same database schemas, any patches can be treated, with slight modification, as cluster patches are treated. If a patch is rollable across a cluster, then it is rollable across the two sites. Likewise, **any patch or upgrade that requires complete downtime of the site will also require complete downtime of both sites**.

In the case of rollable patches:

1) Roll the patch across Site 1 while Site 2 also remains up.
2) Then roll the patch across Site 2 while Site 1 remains up.

In the case of larger upgrades where the entire system must be down, there are two options:

1) Bring down both sites and apply the upgrade to Site 1 and then Site 2. Afterwards, start up both sites.
2) Bring down both sites and apply the upgrade to Site 1 only. Bring up Site 1. Re-create Site 2 (including binaries and domain) as a copy of Site 1 in the manner that was documented earlier in this paper.

Topology changes

Topology changes include Scaling out the clusters by adding more Servers or adding new machines to the environment.

There are no strict symmetry requirements in the Active-Active setup that requires both sites to be topologically identical. It is, however, a best practice to keep them so.

For example, in the case of MAN replication, if one site has fewer servers than the other, the site with fewer servers will be carrying more session state per server in memory. This is because servers will need to act as secondary servers to more than one remote primary server.

In making topology changes such as adding new servers then, the change can be made first to one site while keeping both sites active and then made to the other site with no interruption of service. The sites will be temporarily asymmetric but this is not critical.

When adding new servers, the following changes should also be made:

1) Update the server list if needed on the HTTP Servers or Load Balancer.
2) Update the Cluster Address of the Remote Cluster for MAN Replication to include the new server.

Supporting Other Applications

This section outlines some of the main considerations when considering whether a custom application will run as Active-Active. This includes Custom Portal applications as well as other custom applications that need to co-exist with Oracle Webcenter Portal.

There is one overall requirement to keep in mind: **Can a User on Site 1, move to Site 2 and have its state be identical?**

This means not only session state and state in the database but also state that may be held by other applications or stored in memory or on filesystems or other forms of storage.

To make this happen there are a few principles to keep in mind:

Clusterability

The application should be clusterable. If it can only run as a singleton because of a resource conflict then the application cannot run Active-Active.

Access to External Resources

If the application requires external resources then this external resource should support concurrent access from either of the two sites.

This includes ensuring that user/credential mapping is the same from the application on either site to the external resource.

Synchronizing Local Resources

In the case of local resources, the resource must either be stateless or the resource data needs to be synchronized between the two sites.

For example, assume each site has a local LDAP. This is only possible if the LDAP is fixed or rarely changes. If any changes occur to the LDAP on one site, such as adding a user, then the same change must occur on the second site, either manually or through some synchronization mechanism.

In some cases, resources are constrained to be local. Two examples follow, that of local filesystem resources and local memory caches.

Filesystem resources

If the application is storing critical runtime state on the filesystem this will need to be available to both sites so that the state is not lost upon failover. Both sites sharing the same storage may not be practical. The solution may be to move the state into a database.

Local caches

As discussed in this paper, local cluster caches may lead to inconsistencies on failover or, in the case, of some in-memory caches – data loss on failover. The only resolution is to not rely on these caches, to transform them into more persistent remote storage or to make them distributed across the two sites.

Appendix A: Network Monitoring Tools

In the course of investigating the topology of an Active-Active installation, several tools were found to be useful in both monitoring and also simulation of the network traffic between the two sites.

These tools are also useful in determining the traffic in the environment empirically in order to aid in bandwidth calculations and network planning.

Monitoring Interfaces in Linux

If the network channel is on its own interface (and the interface is not being used for other traffic) then the interface itself can be monitored to reveal session state traffic or other types of traffic in the network.

One tool for this is **vnstat** - available at http://humdi.net/vnstat/

Vnstat provides a summary of bits and packets either received or transferred through the network interface.

Vnstat can be used to summarize all traffic during a trial run of one user or several users accessing the system. First, vnstat is started, then the run is performed and finally, vnstat is stopped.

A sample run of vnstat is shown below, showing the traffic of one session request hitting a server. 177 kib (or about 181k) was transferred to another server in the cluster.

```
$ vnstat -l -i eth1 --style 4

Monitoring eth1...    (press CTRL-C to stop)

   rx:         0 kbit/s     1 p/s          tx:         0 kbit/s     0 p/s

   rx:         4 kbit/s     3 p/s          tx:        32 kbit/s     5 p/s

   rx:         4 kbit/s     3 p/s          tx:        44 kbit/s     5 p/s

   rx:         0 kbit/s     0 p/s          tx:         0 kbit/s     0 p/s

   rx:         0 kbit/s     0 p/s          tx:         0 kbit/s     0 p/s

   rx:         8 kbit/s    14 p/s          tx:       628 kbit/s    56 p/s

   rx:         0 kbit/s     1 p/s          tx:         0 kbit/s     0 p/s
```

```
 eth1  /  traffic statistics

                           rx         |       tx
--------------------------------------+------------------
  bytes                       4 KiB   |         177 KiB
--------------------------------------+------------------
          max               8 kbit/s  |      628 kbit/s
      average            1.10 kbit/s  |    48.83 kbit/s
          min               0 kbit/s  |        0 kbit/s
--------------------------------------+------------------
  packets                         50  |             142
--------------------------------------+------------------
          max                 14 p/s  |          56 p/s
      average                  1 p/s  |           4 p/s
          min                  0 p/s  |           0 p/s
--------------------------------------+------------------
  time                    29 seconds
```

Adding Delay and Packet loss to an Interface

Once replication has been isolated to an interface, another tool allows us to simulate delays and packet loss on that interface. The tool is **netem** and it should already be a part of the Linux distribution:

http://www.linuxfoundation.org/collaborate/workgroups/networking/netem

A sample of netem is shown below. Note that netem only affects inbound packets. For monitoring session replication, netem should be run on the server receiving session state. Or it can be run on all servers since the end-to-end characteristics of the data will still be affected.

You run netem as root. To perform basic operations such as create packet loss and delay, you create a qdisc, or scheduler, and then modify the properties of that scheduler. Finally, delete the scheduler to return the system to normal.

Some examples are shown below.

```
# Create a qdisc on eth1 and set a delay of 10ms

$ /sbin/tc qdisc add dev eth1 root netem delay 10ms


# Modify the qdisc to also add a packet loss of 0.1%

$ /sbin/tc qdisc change dev eth1 root netem loss 0.1%


# Remove the qdisc altogether

$ /sbin/tc qdisc del dev eth1 root
```

netem can be used to test the effects of network reliability on session replication or other network traffic.

Using iperf to monitor bandwidth

The tool iperf can be used to examine the available bandwidth on a network and its reliability. It can be found at:

http://sourceforge.net/projects/iperf/

Iperf is capable of measuring bi-directional bandwidth by setting up a client and a server component on each end of the network.

In an active-active cluster, replication traffic is being both sent and received from the remote cluster. Iperf is capable of performing a simulation of this and calculating the available throughput.

Appendix B: Load Testing Tools

For testing of the ADF applications, two testing tools were used, Oracle's Application Testing Suite and JMeter from the Apache Software Foundation.

Oracle Application Testing Suite

The components of Oracle ATS can be downloaded from the following location:

http://www.oracle.com/technetwork/oem/downloads/index-084446.html

Tests are created using a tool called OpenScript which allows the creation of ADF tests by recording an actual run. These tests are then packaged and loaded onto another tool called Oracle Load Tester. There, the test can be run and configured to run multiple times and with different numbers of virtual users.

Apache Jmeter

Apache Jmeter is a popular testing platform. It can also be used to run ADF tests but some additional configuration is required.

Recording a Run

The steps to record a run involve configuring a Jmeter proxy server where Jmeter can record the details of a user run. The steps to do this are documented in many places including Oracle's Note 1334214.1: How to do Performance Testing with Oracle Webcenter and Apache Jmeter

This will create a valid Jmeter test plan but you may still encounter problems with getting the test plan to work in an ADF application.

Setting ADF variables

Unfortunately, ADF makes use of parameters passed in HTTP POST and in the URL. And these parameters will change from session to session. So, these parameters need to be converted to variables.

The steps to do this were worked out by an Oracle consultant and can be found at: http://one-size-doesnt-fit-all.blogspot.co.uk/2010/04/configuring-apache-jmeter-specifically.html

References

1. Oracle Maximum Availability Architecture Web site
   http://www.otn.oracle.com/goto/maa
2. MAA Best Practices – Oracle Fusion Middleware
   http://www.oracle.com/technetwork/database/features/availability/fusion-middleware-maa-155387.html
3. Enterprise Deployment Guide for Oracle Webcenter Portal
   http://www.oracle.com/technetwork/middleware/webcenter/portal/documentation/index.html

Oracle White Paper Title:
August 2013
Author: Richard Delval
Contributing Authors: Pradeep Bhat, Richard Nessel, Eric Pollard, Jeni Ferns

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

Oracle is committed to developing practices and products that help protect the environment

0109