Oracle WebLogic on Shared Storage: Best Practices

Oracle Maximum Availability Architecture White Paper
May 2013

Maximum Availability Architecture

Oracle Best Practices For High Availability

Introduction

This paper explores the best practices for an enterprise application deployment of Oracle Weblogic Server in which software and configuration reside on shared storage. A directory structure is presented which is both suitable for shared access and flexible enough to accommodate hybrid models of both local and remote storage. Best practices are also presented for deploying custom applications. This paper is applicable to Oracle Fusion Middleware 11g only.

Advantages of Shared Storage

Shared storage is an external physical disk, usually hosted on a dedicated storage server, which can be equally accessed by all the client machines. The software on shared storage is used to support a deployment across multiple physical client machines.

This configuration has the advantage of providing a centralized location for management of software and configuration. Provisioning across a large number of physical servers is simplified, as new software does not need to be installed and configured for each new host. Shared locations can also store common configuration, common binaries and artifacts that need to be immediately accessible in the case of machine failure and failover. Shared artifacts accessible to all members of a cluster, are also requirements in some applications. The shared model also removes many dependencies on particular physical machines and thus is suitable for environments where these need to be interchangeable.

Finally, shared storage can simplify Backup and Recovery operations by providing built-in abilities such as snapshots and replication that can provide access to earlier incarnations of filesystem objects.

Structure of this paper

This paper presents a shared storage model in which all of the client machines access the same binaries and the same cluster directories. The domain home can either be on local disk or on shared disk.

The first section of this paper covers the directory structure guidelines for a Fusion Middleware installation that resides on shared storage. The directory structure is based on the separation of the three broad types of Fusion Middleware artifacts: Binaries, Domain directories and Shared directories. Also covered is an overview of the range of shared storage options and recommended configurations.

The second section of this paper presents a worked example. We walk through the steps to install WebLogic Server and configure the shared storage directories and considerations specific to a custom Enterprise application. This includes how to partition the FMW directory structure onto different types of shared storage or local disk in order to optimize management and performance. In addition, procedures are provided to scale out the Domain to additional nodes.

Another worked example shows how a custom application can be deployed using only one Domain Home that is shared across the entire cluster and is used to run all servers. The advantage of this configuration is that it allows for faster provisioning. This option is available if the application allows it.

The last sections of this paper address considerations and best practices for a custom application including lifecycle of the topology, managing configuration, and managing deployments and, finally, troubleshooting shared storage issues.

Overview

The basic physical architecture discussed throughout this paper is that of one or more physical machines accessing the same storage server on the network. The Host machines are Linux or Windows for example. The storage server is another device such as a NAS or a SAN that is making its storage available.

These host machines will have their own local storage but also be able to mount storage volumes that reside on the external storage server.

Understanding how to lay out Oracle Fusion Middleware software and files on shared storage, first involves understanding the different types of artifacts in an installation and their properties.

Oracle Fusion Middleware Artifacts

An Oracle Weblogic Installation consists of different types of artifacts. Each of these will have different lifecycle characteristics and different requirements of whether they are shared or not shared as well as different read/write profiles.

Sharing requirements

An artifact will either need be Shared, Not-shared (or private) or Available.:

Shared: The artifact resides on shared storage and can be *simultaneously* viewed and accessed by all client machines.

Available: Also on shared storage but only one client machine will view and access the artifact at a time. If this machine fails, for example, another client machine can then access the artifact.

Not-shared: The artifact can reside on local storage and never needs to be accessible by other machines.

Read/Write profile

An artifact will also have specific read/write requirements:

Read-Only: This artifact is rarely altered but only read at runtime.

Read-Write: This artifact is both read and written to at runtime.

Artifact Characteristics

With the above in mind, we can introduce the different types of artifacts in Oracle Fusion Middlware and their characteristics. A fuller description of each of these types of objects follows in the next section:

	Sharing	*Read/Write*

Binaries	Shared	Read-Only
Managed server Domain Home	Not-Shared	Read-Write
Adminserver Domain Home	Available	Read-Write
Cluster Files	Available	Read-Write
Nodemanager Configuration	Not-Shared	Read-Write
Application specific files	Shared	Read-Write

Directory structure

The recommended shared directory structure for Fusion Middleware artifacts has the following properties:

Separation of Binaries and Config

The top-level division is between binaries in the Fusion Middleware Home (FMW_HOME) and the Configuration directories under a directory such as /admin.

The binaries include the Oracle WebLogic Home, the Oracle JDK, Coherence binaries and any required Fusion Middleware binaries. Binaries are objects that will be either primarily or exclusively be read and rarely written to.

The Configuration directories include all Domain homes, application files and nodemanager configuration. These directories will be both read and written at runtime.

Shared Binaries

All machines use the same set of binaries to run processes. Binaries are installed once and then only modified during patches and upgrades.

Local or Shared Managed Server Domain Homes

Managed Server Domain Homes are used to run managed servers on the local host machine. For performance or other reasons it may be desirable to have these domain home reside on local storage of the host machine. However, these can also be on the shared storage in their own separate volumes (one volume per machine). Further, if the application permits this, all client machines can also use the same Domain Home to contain their managed servers.

A Domain Home for AdminServer

The Domain Home from which the AdminServer runs is contained in a separate volume that can be mounted on any machine. This allows the AdminServer to be readily available if the machine that was hosting it has failed or is otherwise unavailable.

Shared Cluster files

Files and directories that might need to be available to all members of a cluster are separated into their own directories. These will include JMS files, Transaction logs and other artifacts that belong to only one member machine of a cluster but might need to be available to other machines in the case of failover.

Local NodeManager Directory

Each client machine will have its own nodemanager and nodemanager configuration directories. These can also optionally reside on local storage since a nodemanager is tied to one physical machine.

Shared Application specific files

Any files that must be both shared AND accessed concurrently (read/write) will be application dependent. No files that are part of the Oracle FMW WebLogic infrastructure have this requirement but some FMW products (such as Oracle WebCenter Content) and custom applications may require shared files on disk.

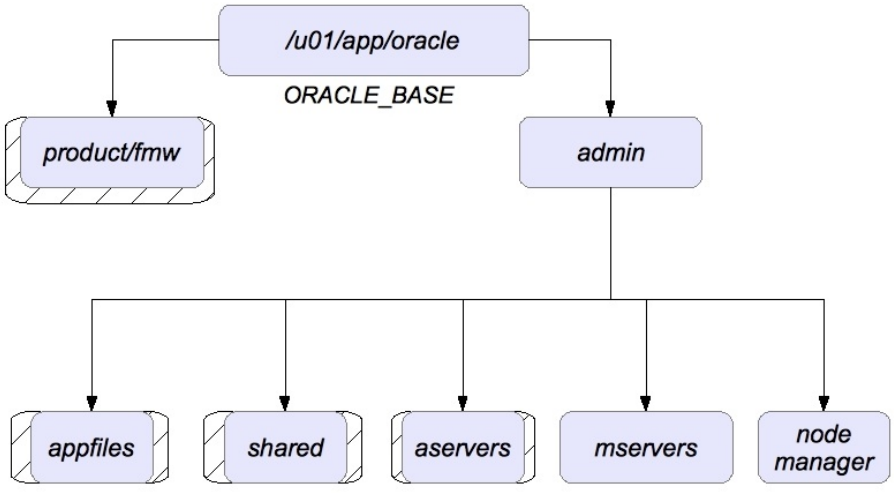

Fig. 1 Top level Directory and Mountpoints

The diagram above shows the top-level directory structure. All product and application files exist under a directory which we will designate as **$ORACLE_BASE**.

All of the binaries reside under the **$ORACLE_BASE/product/fmw** directory. This is a separate volume on shared storage that is mounted by all machines in the cluster.

A hashed box in the diagram in Figure 1 designates the mount points. For example, the directory /u01/app/oracle/product/fmw is an individual local mount point.

In the diagram above, the **mservers** and **nodemanager** directories are on local storage and are used to run local managed servers and the machine's nodemanager. As discussed earlier, these directories can also optionally reside on shared storage but, if so, do not need to be mounted by more than one node.

The AdminServer domain home resides under the **aservers** directory. This is a separate directory that can be independently mounted by any machine.

More detail on the directories under the **admin** directory are shown below:

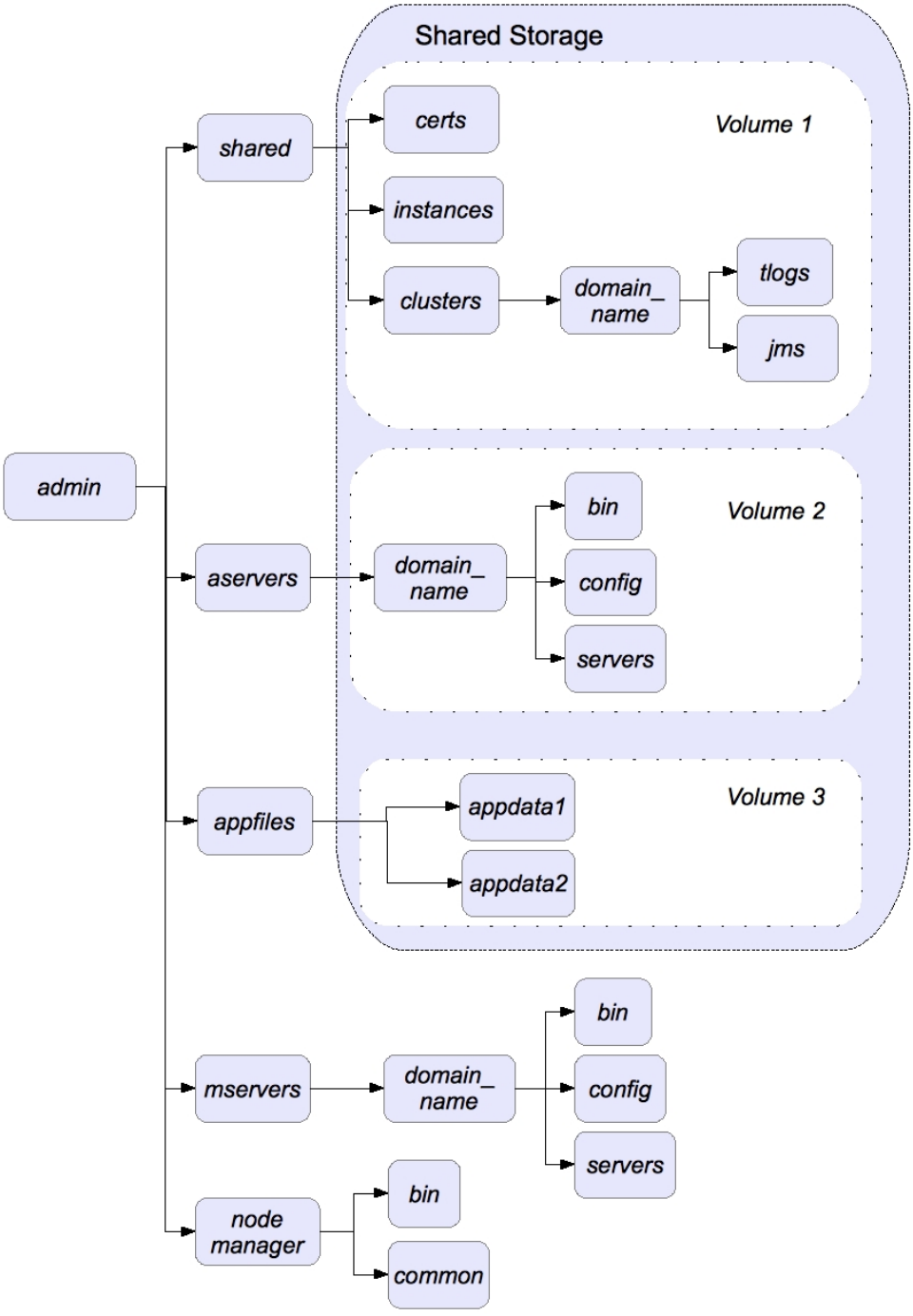
Shared Storage
Volume 1
Volume 2
Volume 3
admin
shared
certs
instances
clusters
domain_
name
tlogs
jms
aservers
domain_
name
bin
config
servers
appfiles
appdata1
appdata2
mservers
domain_
name
bin
config
servers
node
manager
bin
common

Fig. 2 Directories under the admin directory

Each client machine mounts the shared volume (Volume 1) in order to access shared artifacts such as Persistent Stores, Transaction logs and Certificate and keystores.

The **aserver** volume (Volume 2) is mounted by the machine that is running the AdminServer for the domain.

The **appfiles** volume (Volume 3) is mounted by all machines running applications that require a store for concurrently accessed files.

The **mservers** and **nodemanager** directories do not need to be on shared storage. Optionally, the **mservers** and **nodemanager** directories can also reside on shared storage. The mserver directory can be private to each machine or (as seen in the example in this paper) can be shared by one or more machines.

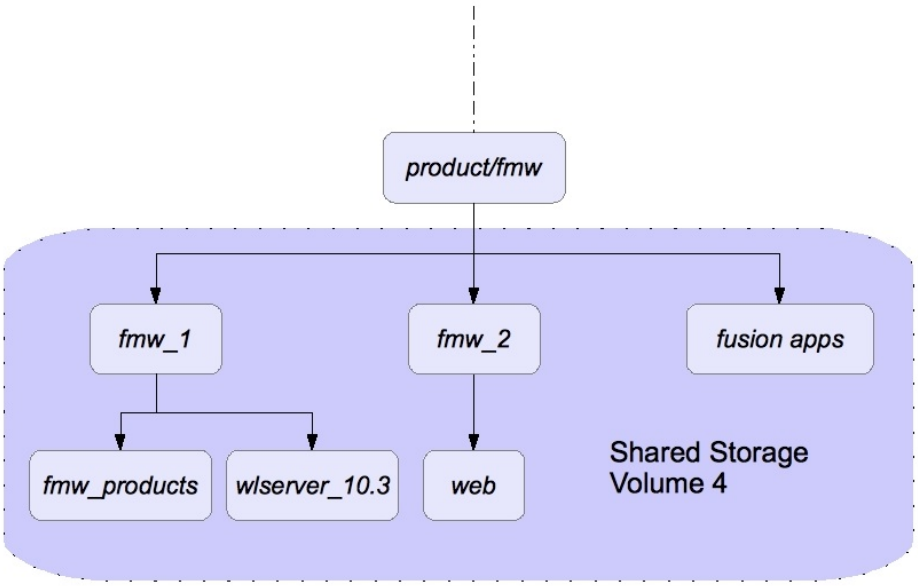

Figure 3. The FMW Directory structure on Shared Storage

The Binaries directory is mounted and shared by all client machines.

A key to some of the directories:

$ORACLE_BASE

This is the top-level oracle directory, in this case /u01/app/oracle

$ORACLE_BASE/product/fmw

All FMW product directories will exist under this directory tree. This is mounted by all machines and used to run Oracle WebLogic Server.

$ORACLE_BASE/product/fmw/fmw_install

This is the directory tree for a specific fmw binary install. In practice separate directories under fmw may be required either for separate products or for different versions of FMW.

$ORACLE_BASE/admin

Everything else besides the binaries will exist under this directory tree.

$ORACLE_BASE/admin/mservers

All Managed Server domains are under this directory. This directory can be local on each machine.

$ORACLE_BASE/admin/mservers/*domain_name*

Here the domain name is an actual domain name such as 'MyDomain'

$ORACLE_BASE/admin/shared

This directory acts as a single mount point and container for different types of shared artifacts.

$ORACLE_BASE/admin/shared/instances

Directory for oracle instances. Part of the web tier.

$ORACLE_BASE/admin/shared/certs

Directory used to store certificates as well as Identity and Trust keystores. The certificates all have unique names. The keystores are shared by all members of the domain and across domains.

$ORACLE_BASE/admin/shared/clusters/*domain_name*

All cluster information is stored here for a specific domain. Optionally, beneath this directory is the clustername.

The further sub-directories here will depend on the application. For some applications this might include directories for JMS and Transaction logs.

$ORACLE_BASE/admin/nodemanager

Nodemanager homes are moved here. In this configuration there is one nodemanager per physical machine. The nodemanager home is located at $ORACLE_BASE/admin/nodemanager/{machine_name}/common. The node manager start scripts are located at $ORACLE_BASE/admin/nodemanager/{machine_name}/bin.

Instructions for achieving this separated nodemanager configuration are described in this paper.

Again, the machine_name directory is **optional** and not necessary if each client machine has the nodemanager directories on local disk.

Shared storage Protocols and Devices

The primary example used in this paper is a Linux client accessing a remote filesystem exposed as NFS v3 or NFS v4 . The range of options, as well as how these devices are configured and mounted is briefly discussed here. Specific Shared Storage vendors and machines are not discussed here.

Some of the options for storage that can be accessed by multiple client machines are illustrated in the diagram below.

	Windows solutions	Linux solutions
Block storage	ISCSI/NTFS (MSCS)	ISCSI/EXT3
Shared Filesystem	CIFS, NFS, ACFS, DBFS	NFS, ACFS, DBFS

Fig 4. Shared storage options for Windows and Linux

Remote Block Devices can be made available through protocols such as ISCSI which allow remote storage to be accessible using many of the same protocols as local SCSI devices. A filesystem appropriate for the client machine such as NTFS for Windows or EXT3 for Linux can then be configured on this device. Although the resulting filesystem can be physically shared, this solution lacks the necessary semantics and protocols to be able to manage multiple client machines accessing the same filesystem and files concurrently.

Microsoft Cluster Services (MSCS) also known as Failover Clusters is noted here because it also is not a solution for managing multiple *concurrent* access to shared artifacts. Failover Clusters works by ensuring that only one client machine is reading/writing to the remote NTFS filesystem and, if this machine fails, automatically making the remote filesystem available to another client machine.

A shared filesystem is both a filesystem and a set of protocols to manage multiple concurrent access. The most common of these protocols is CIFS for Windows and NFS for Unix clients. These solutions consist of a back-end file system such as ZFS or EXT3 along with server processes (such as NFSD) to manage and mediate access.

Oracle's ACFS filesystem solution is slightly different in that instead of the server managing access, access is managed by Oracle Automatic Storage Management (ASM) and Oracle Clusterware. Oracle's DBFS filesystem provides a FUSE (Filesystem in User Space) filesystem interface to the Oracle database allowing the Database itself to manage concurrency.

More information on each of these solutions is provided below as well as their suitability for use as a shared filesystem solution for Oracle Fusion Middleware.

NFS v3

NFS version 3 is perhaps the most common standard for Unix clients that require a remote, shared filesystem. Permissions between the client machine and the storage server are mapped by using the UID of the users. NFS v3 is relatively easy to configure and export on most devices. The client machine can then mount the remote device locally with a command such as the following:

```
$ mount -t nfs storage_machine:/export/nfsshare
/locmntddir -o
rw,bg,hard,nointr,tcp,vers=3,timeo=300,rsize=32768,wsi
ze=32768
```

which mounts the remote filesystem at /locmnntdir. Most of the values specified above are default and so an NFS v3 device can also be mounted as:

```
$ mount -t nfs storage_machine:/export/nfsshare
/locmntddir -o nointr,timeo=300
```

The right values of rsize, wsize and timeo will depend on attributes of the network between the client machine and the storage server . For example, a higher timeout value will help to minimize timeout errors.

The attributes `noac` and `actimeo` are not used here. These are used to disable attribute caching and may be necessary when objects such as Oracle data files are placed on NFS. They are not necessary, however, for most WebLogic applications or specific FMW applications such as SOA, WebCenter or BI. Enabling these parameters will carry performance penalties as well.

Other options include `noatime,` which will provide a small performance boost on read-intensive applications by eliminating the requirement for access-time writes.

NFS v4

NFS v4 has several advantages over NFS v3,. One is support for an authentication protocol such as NIS to map user permissions instead of relying on UIDs. The other is support for the expiration of file locks through leasing.

After a specified time, the storage server releases unclaimed file locks. The default lease expiration time is storage server dependent. Typical values are 45 or 120 seconds. In Sun ZFS systems the default is 90 seconds. In a failover scenario, where a server has failed and another server must claim or access the files owned by the failed server, it is able to do so after the lease expiration time. This value should be set lower than the failover time, so that the new server can successfully acquire locks on the files. For example, this should be set lower than the time for server migration in a cluster using JMS file based persistence .

This differs from NFS v3 where locks are held indefinitely. The locks held by a process that has died have to be released manually in NFS v3.

An NFS v4 filesystem can be mounted on Linux similarly to NFS v3:

```
$ mount -t nfs4 storage_machine:/export/nfsv4share /locmntddir
-o nointr,timeo=300
```

CIFS

For Windows storage, Common Internet File System (CIFS) also known as SMB is commonly used. CIFS supports optimistic locking for greater performance in low concurrency scenarios.

In optimistic locking, the client (Machine 1) claims a file lock and notifies the CIFS server. The client can, from then on, assume it holds the lock. The Client does not have to renew a lease, as in NFS v4. If another client (Machine 2) requests the lock, the server notifies Machine 1 to request the lock be released.

In a failover scenario, the CIFS server will not request a lock to be released by Machine 1 until a second Client requests the lock. If Machine 1 has failed, the server will not receive a response from Machine 1 and, after a timeout period, will go ahead and release the lock.

Permissions can be mapped using Windows Active Directory Authentication. A CIFS share can be exposed as an NFS v3 or v4 share as well by most storage servers. The server reconciles the different locking mechanisms automatically.

Accessing a CIFS share involves mapping a remote drive of the form `\\storage_machine\export\cifsshare` to a local drive such as `E:`

ACFS

A shared storage installation can also be successfully performed on Oracle's ASM Cluster Filesystem (ACFS) and was found to be a highly performant solution. Guidelines and usage for ACFS, however, is not covered in this paper.

DBFS

The Database Filesystem (DBFS) is a FUSE (Filesystem in UserSpace) front-end to the Oracle Database. Files are stored as CLOBS in the Oracle database and accessed by multiple remote clients. Since the files are database objects, they are managed with all the transaction control and semantics that a database provides. This paper does not cover the details of implementing FMW Middleware on DBFS.

The following table summarizes the different types of shared storage options and their support within this paper:

Shared Storage	Usage within this paper
NFS V3	Recommended
NFS V4	Recommended
CIFS	Recommended
ACFS	Outside the scope of this paper
DBFS	Outside the scope of this paper
ISCSI/NTFS ISCSI/EXT3	NOT Recommended as an option for multiple concurrent access (for shared application data, for example)

Windows Client Support

In the sections that follow, commands are given for Linux machines. The same procedure can be used in configuring Windows clients. Where the Windows equivalent command is obvious – such as in creating directories or editing files – no Windows equivalent is given. In a few cases, the Windows procedure differs enough and a note is added to the end of that section.

Step-by-Step Example: Installing and Configuring a topology for FMW applications (suitable for Exalogic)

This Example walks through the steps to install Oracle FMW WebLogic into a shared environment, distributed across one or more volumes. It is assumed that a custom Enterprise application will be deployed into this environment.

The Topology

The topology consists of:

Two or more application nodes.

The assumption is that more application nodes can be added at anytime and quickly provisioned as described here.

Shared Storage.

The shared storage device holds all binaries, all domain files, all configuration files.

Provisioning still requires a few local files such as `beahomelist` and Oracle Inventory. The primary reason for these files to be on local disk is to allow coexistence with other local installations.

The topology otherwise has nothing on local disk. The shared disk contains all the software and all the applications. This increases portability and eases maintenance. Care must be taken however both in preparing this environment, managing this environment (including what is run from what machine) and in maintaining it through patches and upgrades.

Private, Local Domain Home

The Managed Servers on each machine will run out of a private Domain Home. Although this domain home is on shared storage, it will be mounted by only one node.

The Admin Server maintains its own domain home on shared storage so that it can be mounted on any node.

One NodeManager volume

For simplicity, we'll create the nodemanagers for all the nodes on one volume on shared storage. The nodemanager directories will be separated by machine name.

Admin Server Failover

Because the Admin Server Domain Home resides on shared storage, it can be mounted on any machine. The Admin Server is a singleton and, in order to avoid conflicts, it is recommended that the Admin Server be mounted on only one machine at a time.

Hardware provisioning

Shared disk

Mount the shared disks on all machines that will be running the Oracle Weblogic environment.

After creating the local directory structure down to /admin and to /product/fmw, we need to create the necessary volumes on the shared storage and mount them. The table below shows the different volumes on storage for a two-compute node cluster and where they will be mounted:

Volume (local dir. Path)	Notes	Mounted on:
1. /product/fmw	Binary Install	Node 1, Node 2
2. /mserver	Managed Server Domain Home	Node 1
3. /mserver	Managed Server Domain Home	Node 2
4. /aserver	Admin Server Home	Node 1
5. /nodemanager	All nodemanagers	Node 1, Node 2
6. /shared	Shared artifacts	Node 1, Node 2

Sample mount commands are given below for mounting on Linux systems

Mount the same binary directory on all nodes

```
Node1$ mount storage:/export/binaries /u01/app/oracle/product/fmw –t nfs4
Node2$ mount storage:/export/binaries /u01/app/oracle/product/fmw –t nfs4
```

Mount the individual managed server directory on each node

```
Node1$ mount storage:/export/mserver1 /u01/app/oracle/admin/mserver –t nfs4
Node2$ mount storage:/export/mserver2 /u01/app/oracle/admin/mserver –t nfs4
```

Mount the nodemanager directory on each node

```
Node1$ mount storage:/export/nodemgr /u01/app/oracle/admin/nodemanager –t nfs4
Node2$ mount storage:/export/nodemgr /u01/app/oracle/admin/nodemanager –t nfs4
```

Mount the Admin directory on one node

```
Node1$ mount storage:/export/aserver /u01/app/oracle/admin/aserver –t nfs4
```

Virtual IPs

The primary purpose of Virtual IPs is to allow specific servers to failover. For example, configuring a Floating IP for the AdminServer allows the AdminServer to be started on any host. Clients of the Admin Server can still access the server at the same address.

Configure a Floating IP also for servers that will participate in Automatic Server Migration.

Installing the Application

All software can be provisioned from one node. First, we are installing only the Weblogic Server binaries.

Install a full install of Weblogic Server.

The target for the install should be ORACLE_BASE/product/fmw/fmw_install. Here, fmw_install is a user specified name for this particular installation. From hereon, we will refer to this install directory as FMW_HOME.

Accept the defaults for the location of the Weblogic home and Coherence binaries.

Install any other Application Binaries

The FMW binaries can be installed into the same directory as WebLogic.

The location of APP_HOME will be FMW_HOME/App1 for example. It is assumed that one installation can support multiple domains on multiple machines.

Creating the Domain

The Domain should be created in $ORACLE_BASE/admin/aservers. The application home should be specified as $ORACLE_BASE/admin/aservers/applications.

Note: It is important to keep the applications folder within the aserver volume. This volume will failover as a group and so the applications folder should be kept on the same volume.

Create machines for every physical machine in the cluster. Ensure that when creating multiple managed servers that they all have different Listen Ports. This is to simplify the movement of Managed Servers across machines.

Creating the Managed Server Domain Homes

For Managed Server domain homes, the pack/unpack utility should be used to create a Managed Server home for the local machine and for all subsequent machines.

1. Use pack on Node 1 to create an archive:

```
$ $FMW_HOME/wlserver_10.3/common/bin/pack.sh –
domain=$ORACLE_BASE/admin/aservers/mydomain_name –
template=$ORACLE_BASE/admin/shared/stage/mydomain_template.
jar –template_name=”mydomain_template” –managed=true
```

2. Use unpack on the local node and all subsequent nodes to create a managed server domain:

```
$ $FMW_HOME/wlserver_10.3/common/bin/unpack.sh –
template=$ORACLE_BASE/admin/shared/stage/mydomain_template.
jar –domain=$ORACLE_BASE/admin/mservers/mydomain_name –
```

These steps need to be repeated if the domain is ever extended using the Configuration Wizard. This must be done for all machines in the domain.

Note that it might be necessary to add the parameter overwrite_domain=true to the unpack command in Step 2 above. This will be required if a domain home already exists at that location.

Post-Domain creation

JMS Persistence Store

If the Managed Servers require a JMS Persistence Store for pending JMS Messages, this can be set as a shared location so that recovered servers can access the persistence files at the same location. The location of this store can be set as `$ORACLE_BASE/admin/shared/clusters/domain_name/jms` for all servers in the domain.

The location of this directory can be set for each individual Managed Server through the Administration Console by clicking on **Services->Persistence Store** in the Left Pane and then selecting the individual Persistence Store.

Persistence Store for Transaction Recovery

For some applications, each Server can also store information about committed and uncommitted transactions in a transaction log. This can be used in recovery in order to clean up transactions in progress. This log should also be stored in a shared storage location. The location of this log can be set as

`$ORACLE_BASE/admin/shared/clusters/domain_name/tlogs` for all servers in the domain. To set this location, click on each individual Server, click on its Services tab, and enter the location in the Default Store field on this page.

Configuring and Starting up the Node Manager

Nodemanagers can run from different machines and share the same set of binaries and configuration files. However, there are good reasons to separate nodemanager configuration and logfiles:

- When setting up Trusted communication between Node Managers and the AdminServer, each NodeManager requires its own certificates.
- When configuring Automatic Server migration, node managers may need to be configured with different local network interfaces.
- Although each Nodemanager will create its own lock file and log files, there is no naming scheme to easily determine which logfiles belong to which nodemanager.

In a shared nodemanager configuration, these issues can be solved by overriding the values in the shared nodemanager.properties file with startup parameters when the nodemanager is started.

To simplify this process and avoid confusion, we suggest that a separate StartNodeManager.sh file be created for every physical host and a separate Nodemanager home be created for every physical host. The following sections show how to achieve this.

Create the NodeManager directories

Create a directory to store the configuration files for the nodemanager on this machine. In the examples below, node_name is the listen address of the nodemanager. This will usually be the physical hostname of the machine.

The /common directory will be the Node Manager Home for this machine. The /bin directory will hold the Start script for the Nodemanager on this machine.

```
$ mkdir -p $ORACLE_BASE/admin/nodemanager/node_name
$ mkdir $ORACLE_BASE/admin/nodemanager/node_name/bin
$ mkdir $ORACLE_BASE/admin/nodemanager/node_name/common
```

Copy default files

The /common directory is the new NodeManagerHome and will hold all the nodemanager log and configuration files. By default the NodeManagerHome is $FMW_HOME/wlserver_10.3/common/nodemanager. At this point, the only file in that directory should be nodemanager.domains. Copy this over:

```
$ cp $FMW_HOME/wlserver_10.3/common/nodemanager/*
$ORACLE_BASE/admin/nodemanager/node_name/common
```

Also, we will be customizing the start script for the nodemanager on each machine. So, lets copy this over too:

```
$ cp $FMW_HOME/wlserver_10.3/server/bin/startNodeManager.sh
$ORACLE_BASE/admin/nodemanager/node_name/bin
```

On **Windows**, copy

`$FMW_HOME/wlserver_10.3/server/bin/installNodeMgrSvc.cmd`

and `uninstallNodeMgrSvc.cmd`

to

`$ORACLE_BASE/admin/nodemanager/node_name/bin`

Edit Start script

Edit the file startNodeManager.sh that was copied over. Change the line:

```
NODEMGR_HOME="${WL_HOME}/common/nodemanager"
```

to

```
NODEMGR_HOME="$ORACLE_BASE/admin/nodemanager/node_name/common"
```

Spell out the full directory name rather than typing $ORACLE_BASE.

On **Windows**:

Edit installNodeMgrSvc.cmd to add the line:

```
set
NODEMGR_HOME=$ORACLE_BASE/admin/nodemanager/node_name/com
mon
```

Start the NodeManager

Now, we use this slightly modified script to start the NodeManager on this machine:

```
$ cd $ORACLE_BASE/admin/nodemanager/node_name/bin
$ ./startNodeManager.sh
```

The $ORACLE_BASE/admin/nodemanager/node_name/common directory should now have config and logfiles for this nodemanager.

On **Windows**:

Run
`$ORACLE_BASE/admin/nodemanager/node_name/bin/installNodeMgrSvc.cmd`

Use the Services Panel to start the Service

Configuring an HTTP Server

A Web tier consisting of one or more Oracle HTTP Servers can also be installed onto shared storage and run on different machines. The binary installation is only done once. The Configuration tool is run to create new instances for each distinct HTTP Server that will be run out of this home.

HTTP Server Install

Run the HTTP Server installation and install the binaries into $ORACLE_BASE/product/fmw/fmw_name/web

This location is the Oracle Home for the Web Tier installation.

When creating an instance, edit the default directory path and create the instance in the admin directory tree:

$ORACLE_BASE/admin/shared/instances/instance1

Any created components, such as ohs, should have the same suffix as the instance number. For example, instance1 will have ohs1, instance2 will have ohs2 and so on.. The important thing is that each component name have *a unique name within the domain* not just the instance. The reason for this is so that Enterprise Manager can more easily manage these components later.

Scale-out: Provisioning new Hosts and relocating Managed Servers

In this section, we'll run through the considerations when scaling the shared storage configuration to new hosts as well as relocating Managed Servers manually.

These are the steps for provisioning the second host as well as any additional hosts in the future.

Mounting shared directories on new Hosts

The shared volumes should be mounted on the exact same directory path by all client machines in the cluster.

Embedded within many WebLogic files are the full path to many targets and mounting the shared filesystem at a different mountpoint may cause unexpected behavior.

Add home to beahomelist

On the new host, create beahomelist if it doesn't exist and add the Middleware Home directory. This file is used by WebLogic installations to detect existing installations on this machine.

```
$ mkdir ~/bea
$ touch ~/bea/beahomelist
```

Create new nodemanager directories

As we did on the first machine, we need to create the NodeManager Home for this new machine.

```
$ cd $ORACLE_BASE/admin/nodemanager
$ mkdir node2
$ mkdir ./node2/common
$ mkdir ./node2/bin
```

Copy over the nodemanager.domains file from the directory of the first machine.

```
$ cp
$ORACLE_BASE/admin/nodemanager/node1/common/nodemanager.domain
s $ORACLE_BASE/admin/nodemanager/node2/common
```

Copy the original Startup file:

```
$ cp $MW_HOME/wlserver_10.3/server/bin/startNodeManager.sh
$ORACLE_BASE/admin/nodemanager/node2/bin
```

And Edit the file to set NODEMGR_HOME to $ORACLE_BASE/admin/nodemanager/node2/common

Finally, start the NodeManager

```
$ ./startNodeManager.sh &
```

Relocate Managed Server

Although it is possible to configure Virtual IPs for all Servers, a Managed Server can also be relocated from one machine to another by manually changing its Listen Address and associated Machine.

1. Shutdown Managed Server2 through the console.

2. Create a new Machine, Node2, and set its NodeManager listen address to the physical address of Node2.
3. Edit Managed Server2. Change its Machine to Node2 and the Listen Address to Node2 as well.
4. Save and Activate all Changes
5. Start the Managed Server2

The new Node Manager on Node2 now controls the Managed Server.

Ensure that the HTTP Configuration files to point to the new location of Managed Server2. This is only necessary if this Managed Server is actually listed in the WebLogicCluster parameter. Otherwise, the cluster will dynamically discover the relocated Cluster member.

Provisioning Additional Hosts and Servers

To Provision additional hosts and new Managed Servers:

1. Use the Clone command (Environment->Servers, Select Server, Select Clone) to create additional Servers from the existing Managed Servers
2. Follow the steps above to provision a new nodemanager and then set the Listen Address of the new Server correctly

Provisioning Additional HTTP Servers

New instances of Web Tier components such as HTTP Servers should be configured using the Configuration tool at $WEB_HOME/bin/config.sh where WEB_HOME will usually be under $ORACLE_BASE/product/fmw/web

Additional instances should be installed into the existing Web Tier configuration home which is under $ORACLE_BASE/admin/instances. Ensure that new components do not share the same name with other components across the domain.

Admin server failover

With AdminServer failover, a Virtual IP is recommended so that the AdminServer location remains at a fixed address for any applications that require it.

To relocate the AdminServer we bring it down, relocate its Virtual IP, then bring it back up.

1. Bring down the Adminserver on the first machine.
2. Relocate the IP. On Linux:

```
a. NODE1> /sbin/ifconfig ethX:Y down
```

```
b. NODE2> /sbin/ifconfig <interface:index>
   <IP_Address> netmask <netmask>
```

3. Bring up the AdminServer on the second machine

Advanced Node Manager Configuration

The advantage of separating Node Managers is that their configuration can be managed separately. This section lists a few caveats and notes when configuring the Node Managers.

Host Name Verification

Node Managers can be configured for Host Name verification. When a Server is enabled with Host Name Verification, the exchanged certificates must include a matching hostname. The procedure to configure this, including generating the shared keystores, is outlined elsewhere, including the Fusion Middleware Administration Guides.

In this topology, there are a few things to consider:

1. A certificate must be generated for every hostname and Virtual Hostname that will use hostname verification.

2. The Identity keystores are shared by all members of the environment and are domain-independent. Thus, the Identity keystore should be placed in a shared location: $ORACLE_BASE/admin/shared/certs

3. The Trust keystore is also shared by everyone and is also domain-independent. It should also be placed in the $ORACLE_BASE/admin/shared/certs directory.

4. Care should be taken when adding the CustomIdentityAlias to each nodemanager. Each nodemanager should have a unique CustomIdentityAlias specified in its nodemanager.properties file which maps to the hostname assigned to that certificate.

Server Migration

In Server Migration, a managed server is configured to restart on another host should a failure occur. For this configuration, the servers which require failover will listen on specific floating IP addresses that are automatically failed over by WebLogic Server migration.

The steps for Server Migration are not documented here. Again, we only note some considerations specific to the shared topology:

1. All nodemanagers involved in Server Migration will have to set network interface properties such as Interface and NetMask. These should be set in the individual nodemanager.properties files of all the nodemanagers involved.

2. The $PATH variable on each machine should point to the nodemanager.domains file for that specific machine. All nodemanagers can share the same wlscontrol and wlsifconfig scripts.

3. The failed Server will restart using the same directories. If older logfiles are needed for crash analysis, note that these will be quickly archived as startup logs are written.

Adding new Domains

As of FMW 11g, running the Configuration Wizard to create new domains will only add nodemanager domain entries in the default NodeManager Home. When new domains are created, the correct domain entry should be added manually to all nodemanagers that will be part of that domain. This involves editing the nodemanager.domains files and adding an entry in the form:

Domain_name=Domain_home

Where Domain_home is the full path to the Domain directory.

Example: One Shared Domain Home

In the topologies discussed in this paper so far, one AdminServer home was created and then a Managed Server home was created for each machine in the cluster.

Oracle WebLogic can also use one Domain Home, shared across all members of the cluster and used to run the entire cluster environment. This topology has the advantage that a new machine can be quickly provisioned and managed servers can be more easily moved from machine to machine. The disadvantage is a single point of failure for the entire cluster. The Domain Home should be backed up frequently but, even so, cluster downtime is to be expected.

The steps to do this are similar to those in the step-by-step example. The differences as well as any special considerations are outlined here in this section.

Create One Domain Home Only

Instead of creating an aserver domain home and then creating managed server homes, we create one domain home on a shared storage volume with the configuration wizard. The pack and unpack utilities are never used. This Domain Home should be backed up immediately, either through storage server snapshots and/or manually.

All Machines mount the same Domain Home

The Domain Home resides on a volume on the shared storage. This same volume is mounted by all machines in the cluster.

AdminServer Availability

The one Domain Home is an Administration Server home already. If one machine fails, the AdminServer's Virtual IP can be migrated to another machine and the AdminServer can be immediately started there.

Migrating Managed Servers

Managed Servers can be migrated to different machines by

1. Shutting the Server down
2. Altering the associated Machine and their Listen Address in the AdminServer Control Panel
3. Starting the Server

Managed Servers can be migrated as above even if there are multiple Domain Homes. The advantage of one Domain Home is that the Managed Server retains the same server directory which contains its logfiles and any customisations made to the server startup scripts.

Adding a New Machine

Provisioning new machines is simplified with the One Domain Home model. The steps to provision a new machine are as follows:

1. Use the Console to create a new Machine and to pre-create any servers that will run on the new machine. If this is extending a cluster, then this can be done quickly by using the Clone command to clone an existing server in the cluster and then altering the Listen Address and Machine.
2. Mount all the volumes including the binaries and Domain Home
3. Provision a new nodemanager for this machine and start it
4. Start the Servers for this new machine.

Custom Applications Best Practices

Any custom application can be deployed into an environment on shared storage. There are only a few things to consider and some best practices to keep in mind.

Application deployment

Applications are deployed as EAR files to servers or read as exploded directories. The archives can be placed in a shared directory such as $ORACLE_BASE/admin/shared/deploy that is available to all members of the cluster.

In a shared services environment, applications should all be deployed as NOSTAGE. This is the default for deploying to AdminServer but the default for deploying to a managed server cluster is STAGE. In NOSTAGE, the application files reside in a location from which all cluster members can initiate a deployment.

Applications can also be deployed as exploded archives. Again, these can be deployed into a shared directory and made accessible as NOSTAGE to all members of the Cluster. In this case, the location of the directory is provided to the deployer where the archive has been exploded. Each member works from the same archive directory. Any concurrency issues that may arise from different members accessing the same files should be resolved within the application itself.

Concurrency issues, however, can be minimized with some best practices, which are covered in the next few sections.

Configuration and State Management

Successfully managing applications involves separating the instance specific and shared configuration of different instances of an application so as to avoid conflicts.

Shared Configuration

Shared configuration is configuration that must be read and updated by all members of the cluster. The shared configuration can go either into a Database or can be managed on shared disk as one or more configuration files.

The most common usecase for a shared configuration file is an initialization file containing parameters which are read once on start-up by each member of the cluster. A shared file should be mainly read-only and only rarely updated in order to minimize concurrency issues. Otherwise a transactional source, such as a Database, should be considered for the shared data OR the application should build in the necessary logic for concurrent access to a file or sections of the file.

We discuss three options for managing shared configuration files on disk and the best practices for each:

1. **Updatable configuration files within an exploded deployment.**

This is a simple option to implement. All servers share an exploded deployment and read/write to a configuration file within that deployment. This option should generally be avoided.

The initial deployment files should be considered as static and read-only. Updating deployment files in any way will make long-term maintenance of the application - versioning, patching, backups - much more difficult. A re-deployment of the application may also overwrite the entire configuration.

2. **Configuration files targeted to a config. directory**

A common configuration directory can be specified by the application. This can be, for example, any absolute path accessible by al members of the cluster. The configuration directory is fixed and accessible by all members of the cluster. Since it is independent of the deployments, it should survive re-deployments of the application.

3. **User-editable config files that redirect to other config files**

This is one more level of indirection. The configuration files do not contain configuration information but pointers to the location of the actual configuration files. Users can edit this file to re-locate the actual configuration file.

The advantage of this is that config can be located anywhere and if it is dynamic can be located outside of the domain directory. For example in
`$ORACLE_BASE/admin/shared/clusters/domain_name`

Individual Configuration

A Server within a cluster may require either individual configuration or private data that is not meant for consumption by other members of the cluster. The location of this configuration should be clearly partitioned so that multiple cluster members may all share the same directory structure.

For example, a Cluster member should store its configuration not in $DOMAIN_HOME/config/configfile.xml but in $DOMAIN_HOME/config/server_name/configfile.xml This allows for the co-existence of multiple servers of the same type in the same Domain Home.

If configuration information is expected to change after the initial configuration, then the configuration should be placed in a location that will always be accessible to the Managed Server even if it is relocated. A JDBC Data Source can be used for this for example.

The name of the running server is available as an Mbean to the application. This can be used in constructing the output directory names or necessary tables in the database.

Cluster directories

Cluster directories are provided for artifacts which either need to be shared by all members of the Cluster or should be available to be owned and accessed by other members of the Cluster. We can also use these directories for Persistent Stores.

Setting JMS Persistence Stores

This can be configured in the **Services->Persistence Stores** Panel on the Admin Console.

Enter the location of any persistence stores used by Clusters in a domain as:

$ORACLE_BASE/admin/shared/clusters/domain_name/jms

Setting Default Persistence Stores

Open the Admin Console, then Environment->Servers->Server_Name. Click on the Services Tab. In the Default Store field, use:

$ORACLE_BASE/admin/shared/clusters/domain_name/tlogs

This will place individual server logs in a shared directory that other servers in the Cluster can use for Transaction recovery.

Dynamic Configuration

Dynamic data, required by all cluster members should ideally go into a Database. If this is not possible, then the dynamic data should go into the clusters subdirectory. For example, Oracle Universal Content Management (UCM) requires shared vault files which can go into `$ORACLE_BASE/admin/shared/clusters/domain_name/ucmvault`

Centralized Logging

It might be useful to centralize all Server and Application logs in one directory. There are a few reasons to do this:

1. One diagnostics location

Centralized logs can simplify the process of diagnosing a problem on the system by providing one directory or common location where all logs can be examined.

2. Separating dynamic files from the domain directory

The logfiles are the most dynamic files in the domain directory. Separating them out can allow a different backup strategy for these artifacts.

3. Supporting multiple domain homes

Centralized logs are useful when Managed Servers may have to be relocated to other domain directories. The log destination will not change even if the Managed Server is relocated to a new domain home.

To change the default log location for a Managed Server, select the Managed Server in **Environment->Servers** and then the Logging Tab. Enter the new logging location. This can be `$ORACLE_BASE/admin/shared/logs` for example.

Lifecycle Best Practices

The previous sections have addressed installation and configuration and application deployment. This section provides recommendations for managing the lifecycle of a shared storage installation.

Multiple Volumes

Having a single source for all binaries, applications and configuration presents the problem that the environment can become compromised by application error, user error or other hardware or software failures. An error on one physical machine, because of the nature of the shared environment, can potentially cause problems across all machines.

One way of minimizing this risk is to use the capabilities of the storage server itself. Most modern storage servers offer the capability to archive or take snapshots of volumes to create new volumes.

For purposes of management it may also be useful to implement different policies for each of the shared volumes.

Volume 1: Binaries

Directory Tree: $ORACLE_BASE/fmw/

This volume consists of application binaries. These are static, should be mostly read-only and thus require infrequent backups. This volume should be backed up/replicated at least whenever a new product is installed into FMW_HOME or the binaries are patched or upgraded.

Volume 2: Shared Configuration

Directory Tree: $ORACLE_BASE/admin/shared

The clusters directory in this tree is the location of the most active data on disk, read from and written to by different cluster members. The importance of the data is application dependent. In some cases, shared data may be transient; in other cases data here might be vital for a successful recovery. This directory tree should be backed up more frequently than either of the two other volumes.

Volume 3: AdminServer Domain Home

Directory Tree: $ORACLE_BASE/admin/aservers

This includes the most important Domain Home in any installation. This directory tree should not change much during normal runtime but is altered during administrative operations such as any changes to the domain configuration, adding new domains or applications or altering other runtime configuration.

This directory tree should be backed up at least after any administrative operations.

Backup and Recovery

A backup strategy should take into account the different volatility of the different artifacts as pointed out in the previous section. Storage servers can create clones or copies of active volumes and these can be used as backups.

A backup should be taken immediately after the initial configuration of the system for the Binaries Volume and the Configuration Volume. If one of these volumes becomes corrupt, the primary volume can be dismounted and the backup volume can be mounted.

Note that backups should not be created by copying the bits from one volume to another. This can result in errors in file permissions not being set correctly. Backups should be created using server-side mechanisms.

Patching and Upgrades

In order to minimize downtime during patches and upgrades, the multiple volume solution can also be used to perform out-of-place patching. In this scenario:

1. Storage volumes are copied to create new volumes
2. Patch is applied to the copied volumes
3. Servers are shutdown. For a Rolling Patching scenario, this can happen one machine at a time. All processes on that machine including servers and nodemnagers are shutdown.
4. Old volume is dismounted
5. New volumes are mounted
6. All processes are restarted

Troubleshooting Shared Storage

There are a few extra considerations to take into account when deploying and running Oracle WebLogic on Shared Storage instead of a local or dedicated device. Some of these common problems are covered here.

Authentication and Permissions issues

The Storage Server managing the shared storage is an independent device with its own users and authentication. The method by which local users and groups and permissions are mapped to the device and across machines needs to be configured correctly. This also varies depending on the storage protocol such as NFS v3, NFS v4 or CIFS.

Identity Management and Role Mapping

Access management involves the ability to mount and read/write to a remote filesystem and map permissions successfully. These privileges are usually storage server specific and are not covered in this paper.

Identity management involves the ability of multiple client machines and the storage server to allow correct read/write privileges to different types of users. In NFS v3, users on the client machines and the storage server can be mapped using nothing more than UIDs. This default mapping means that one should ensure that UIDS are consistent across all the clients using this storage server. This can be either done manually or through NIS. The only exception to this rule is the special userid 0 or root. Root must be manually mapped to a user on the storage server otherwise it is usually mapped to a user such as 'nobody.'

The recommended way to manage identity is to delegate the authentication to an external source such as an LDAP Server or NIS or Active Directory. If the storage server can enroll in the same Directory Service as the client machines then authentication and privileges can all be centrally controlled. Note that if this is configured it may then not be possible for local users to access the remote storage device.

File Permissions

If the users are mapped correctly using role mapping or through a Directory service then file permissions should be transparent. There are two cases to note here:

Setuid

Installation of Fusion Middleware should not require configuring setuid. It is worth noting though that setuid issues usually point to an issue in how the role mapping was configured as outlined in the previous section. In particular, check that root has been given sufficient privileges on the Storage Server, which does not usually occur by default.

Extended attributes

Most client machines and storage servers support ACL and extended file attributes. However, there may be a mismatch between how each device handles the extended attributes. This will often show up as warnings or errors when performing a `'cp -p'` command. If this occurs, the best option is to disable acl when the remote storage is being mounted. This can be done by adding the option `noacl` to the list of mount options in the mount command.

General File Locking Issues

Different storage protocols handle locks differently.

NFS v3

In NFS v3 locks can remain even after the owner of the lock has disappeared. When this occurs, in the case of an abrupt crash, these locks must be released manually.

In NFS v4, locks will automatically expire after a fixed amount of time if the lock owner has not renewed its lease on the lock.

File locking issues can be detected by examining the Server logs, especially following an abrupt shutdown if a Server has problems restarting afterwards. A message such as the following might appear in the logs:

```
java.io.IOException: Error from fcntl() for file locking,
Resource

temporarily unavailable, errno=11
```

These can be temporarily resolved by removing the lock on the affected file by either:

1. Copying the file, renaming it and copying it back. For example:

```
$ mv locked_file locked_file_tmp

$ cp locked_file_tmp locked_file
```

2. Manually releasing the locks on the storage server. This will depend on the storage device and the protocol. On CIFS for example, open locks can usually be inspected and managed through an interface on the storage server.

If problems persist, Oracle WebLogic does provide the ability to disable file locking for different types of artifacts. This is not recommended however unless all other options have failed.

To disable file locking for the Default File Store for example, select the Advanced section of the Default Store section of a specific Server under **Configuration->Services**. De-select the **Enable File Locking** checkbox.

Similarly, file locking can be disabled for a Custom File Store, a JMS Paging File Store, or a Diagnostics File Store.

NFS v4

In NFS v4, locks will automatically expire after a fixed amount of time if the lock owner has not renewed its lease on the lock.

If a lock has not expired after a failover has occurred then the lease expiration time should be decreased. The steps for configuring this parameter depend on the storage server that is used.

In Sun ZFS servers for example, this can be set with a parameter found as part of the NFS configuration on the server.

Specifically, in the ZFS console, under **Configuration->Services->NFS** the parameter `'Grace Period'` controls the NFSv4 Lease timeout.

If faster failover times are required then this parameter can be lowered. Raising this parameter makes NFS v4 behave more like NFS v3 where locks are held indefinitely.

Oracle HTTP Server Performance and Locking

If Oracle HTTP Server is not as performant as usual when residing on shared storage, the following are configurations that can be tried.

Move OHS DocumentRoot

Moving DocumentRoot to a local filesystem will reduce much of the writing that OHS does on the network filesystem. This can be changed in the httpd.conf file.

Set Locking to Semaphores

By default, lockfiles used for socket serialization are created on the filesystem. These can be replaced with semaphores as follows:

In httpd.conf

- Add the line AcceptMutex sysvsem and comment out any other AcceptMutex lines
- Comment out the LockFile directive if it appears

Move LockFile to local filesystem

Alternatively, the lockfile can be moved to a local filesystem using the LockFile directive

One of the two above LockFile solutions should also be used if an error appears when starting Oracle HTTP Server with the message 'No Locks Available' or a similar locking error.

Oracle White Paper Title:
May 2013
Author: Richard Delval
Contributing Authors: Pradeep Bhat, Shailesh Dwivedi
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