Oracle® Fusion Middleware

High Availability and Disaster Recovery Guide for Oracle WebLogic Server and Coherence

14c (14.1.1.0.0)
F18298-03
February 2024

ORACLE

Oracle Fusion Middleware High Availability and Disaster Recovery Guide for Oracle WebLogic Server and Coherence, 14c (14.1.1.0.0)

F18298-03

Contents

Preface

The high availability (HA) features of WebLogic Server, Coherence, and Oracle Database provide an integrated solution for building maximum availability architectures that span data centers in distributed geographical locations. This document describes three supported WebLogic Server Maximum Availability Architectures (MAA), and provides design considerations that you can use to achieve high availability and disaster recovery solutions.

Audience

This document is intended for administrators, developers, and others whose role is to configure and manage Oracle WebLogic Server and Coherence architectures for high availability and disaster recovery.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following Oracle Fusion Middleware documents:

- *Administering Zero Downtime Patching Workflows*
- *Developing JTA Applications for Oracle WebLogic Server*
- *Developing JDBC Applications for Oracle WebLogic Server*
- *Administering Oracle Coherence*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction to WebLogic Server High Availability and Disaster Recovery

Oracle WebLogic Server, together with Oracle Coherence and Database, includes features that you can use for building maximum availability architectures (MAA) that span data centers in distributed geographical locations.

- About High Availability and Disaster Recovery in WebLogic Server
  Using both high availability and disaster recovery solutions ensures that applications are available when they are needed.
- Terminology
  Learn a comprehensive list of common terms that apply to Oracle WebLogic Server and Coherence high availability and disaster recovery.
- WebLogic Server High Availability Components and Features
  Oracle WebLogic Server provides components and features that work in conjunction with Oracle Coherence and Oracle Database high availability features to provide maximum availability, reliability, and application stability during planned upgrades or unexpected failures.
- Coherence High Availability Components and Features
  Oracle Coherence provides components and features that work in conjunction with Oracle WebLogic Server and Oracle Database high availability features to provide maximum availability, reliability, and application stability during planned upgrades or unexpected failures.
- Oracle Database High Availability and Disaster Recovery
  Oracle WebLogic Server provides strong support for integrating with the high availability (HA) and disaster recovery features of Oracle Database. Integrating with these HA and disaster recovery features minimizes database access time while allowing transparent access to rich pooling management functions that maximize both connection performance and application availability.
- Load Balancers
  Load balancers provide high availability by ensuring that if one web server goes down, requests are routed to the remaining web servers that are up and running.

- Supported MAA Architectures
  WebLogic Server supports three primary maximum availability architecture (MAA) solutions that can be used to protect an Oracle WebLogic Server system against downtime across multiple data centers.
- Potential Failure Scenarios
  Potential failure scenarios range from unexpected full and partial site failures to maintenance outages.

About High Availability and Disaster Recovery in WebLogic Server

Using both high availability and disaster recovery solutions ensures that applications are available when they are needed.

Typically, a high availability solution provides redundancy in one data center. However, production deployments in a data center also need protection from unforeseen disasters and natural calamities. Disaster recovery solutions provide a recovery strategy for applications and data by setting up a standby site at a geographically different location than the production site. Application data, metadata, configuration data, and security data are replicated periodically to the standby site.

Oracle WebLogic Server and Coherence include an extensive set of high availability features, such as server clustering, server migration, cluster integration, Active GridLink, load balancing, failover, backup and recovery, rolling upgrades, and rolling configuration changes, which protect a deployment from unplanned downtime and help to minimize planned downtime. Disaster protection for Oracle databases that are included in your configuration is provided through Oracle Data Guard and Oracle Real Application Clusters (Oracle RAC).

Using the high availability and disaster recovery features of Oracle WebLogic Server, Oracle Coherence, and Oracle Database, you can design and build maximum availability architectures (MAA) that span data centers in distributed geographical locations. Oracle Maximum Availability Architecture (MAA) is Oracle's comprehensive architecture to reduce downtime for scheduled outages, and to prevent, detect and recover from unscheduled outages. The major benefits of these integrated solutions are faster failover or switchover, increased overall application availability, data integrity, reduced human error and risk, recovery of work, and local access of real-time data. See *Best Practices Blueprints for High Availability* at `https://www.oracle.com/database/technologies/high-availability/maa.html`.

Terminology

Learn a comprehensive list of common terms that apply to Oracle WebLogic Server and Coherence high availability and disaster recovery.

- Active-active: An active-active solution deploys two or more active servers to improve scalability and provide high availability. In active-active deployments, all instances handle requests concurrently. When an entire domain or site fails, transactions can be recovered by an active server in a different domain either collocated in the same site or on a different site.
- Active-passive: An active-passive solution involves setting up and pairing a standby site at a geographically different location with an active (production) site. The standby site may have equal or fewer services and resources compared to

the production site, although Oracle recommends configuring symmetrical topology and capacity at both production and standby sites. Having different number of nodes or capacity can cause inconsistencies at the functional and performance levels.. Application data, metadata, configuration data, and security data are replicated periodically to the standby site. The standby site is normally in a passive mode; it is started when the production site is not available. This model is usually adopted when the two sites are connected over a WAN, and network latency does not allow clustering across the two sites.

- Domain pair: A domain pair consists of an active and a passive domain. In an active-active application infrastructure tier with WebLogic domain pairs, the infrastructure tier spans two sites and each site contains a primary active domain and a secondary passive domain. The primary domains at each site are independent domains and do not have to be configured with a symmetric topology, however the domain pair must be symmetrical. For example, if Domain A is the primary (active) domain on Site 1 and Domain B is the primary (active) domain on Site 2, then there must be a Domain B as the secondary (passive) domain on Site 1, and there must be a Domain A as the secondary (passive) domain on Site 2. That is, the pair of domains at each site must be symmetrical, even though the domains themselves can be unique.
- WebLogic Server cluster: A Weblogic Server cluster is a collection of WebLogic Server server instances running simultaneously and working together to provide increased scalability and reliability. In a cluster, most resources and services are deployed identically to each Managed Server, enabling failover and load balancing.
- Coherence cluster: A Coherence cluster is a collection of Java Virtual Machines (JVM) processes, called Coherence servers, that run Coherence. A Coherence cluster consists of multiple Coherence server instances that distribute data in-memory to increase application scalability, availability, and performance. Application data is automatically and transparently distributed and backed up across cluster members.
- Stretch cluster: A stretch cluster is a cluster in which nodes can span data centers within a proximate geographical range, usually with guaranteed, relatively low latency networking between the sites. Stretch clusters are also referred to as extended clusters.
- High availability: High availability is the ability of a system or device to be available when it is needed. A high availability architecture ensures that users can access a system without loss of service. Deploying a high availability system minimizes the time when the system is down, or unavailable, and maximizes the time when it is running, or available.
- Disaster recovery: Disaster recovery is the ability to safeguard against natural or unplanned outages at a production site by having a recovery strategy for applications and data at a geographically separate standby site.
- Switchover: Switchover is the process of reversing the roles of the production site and the standby site. Switchovers are planned operations done for periodic validation or to perform planned maintenance on the current production site. During a switchover, the current standby site becomes the new production site, and the current production site becomes the new standby site.
- Failover: Failover is the process of making the current standby site the new production site after the production site becomes unexpectedly unavailable (for example, due to a disaster at the production site).
- Latency: Latency is the time that it takes for packets to travel from one cluster to another, and can be a factor in 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.

- Metropolitan area network (MAN): A MAN is a telecommunications or computer network that spans an entire city or campus. The MAN standard for data communication specified in the IEEE 802.6 standard is called distributed-queue dual-bus (DQDB). With DQDB, networks can extend up to 20 miles (30 km) long and operate at speeds of 34–155 Mbit/s. A stretch cluster topology is appropriate in a MAN.
- Wide Area Network (WAN): A WAN is a telecommunications or computer network that extends over large geographical distances and between different LANs, MANs and other localized computer networking architectures. Wide area networks are often established with leased telecommunication circuits. Distance and latency of a WAN need to taken into consideration when determining the type of topology you can configure.

WebLogic Server High Availability Components and Features

Oracle WebLogic Server provides components and features that work in conjunction with Oracle Coherence and Oracle Database high availability features to provide maximum availability, reliability, and application stability during planned upgrades or unexpected failures.

These WebLogic Server components and features are described in the following sections.

- WebLogic Server Zero Downtime Patching
- Clustering
  WebLogic Server clusters provide scalability and reliability for your applications by distributing the work load among multiple instances of WebLogic Server.
- Singleton Services
  Singleton services are services that must run on only a single Managed Server instance of a cluster at any given time, for example JMS and the JTA transaction recovery system. WebLogic Server allows you to automatically monitor and migrate singleton services from one server instance to another.
- Session Replication
  Session replication is a feature of WebLogic Server clusters that is used to replicate the data stored in a session across different server instances in the cluster.

- Transaction and Data Source Features
  WebLogic Server features such as Active Gridlink data sources, JDBC TLogs and No TLog, and Logging Last Resource help to provide high availability in WebLogic Server configurations.

WebLogic Server Zero Downtime Patching

WebLogic Server Zero Downtime Patching (ZDT Patching) provides an automated mechanism to orchestrate the rollout of patches while avoiding downtime or loss of sessions. It reduces risks and downtime of mission-critical applications that require availability and predictability while applying patches.

Using workflows that you define, you can patch or update any number of nodes in a domain with little or no manual intervention. Changes are rolled out to one node at a time, allowing a load balancer to redirect incoming traffic to the remaining nodes until the node has been updated.

The ZDT custom hooks feature identifies certain points, referred to as extension points, in a patching workflow where additional commands can be executed to modify the rollout. A user can specify an extension to be run at one or more predefined extension points in the workflow that is executed either on the Administration server node, or on a remote node. See Modifying Workflows Using Custom Hooks in *Administering Zero Downtime Patching Workflows*.

You can use ZDT Patching to update Coherence applications while maintaining high availability of the Coherence data during the rollout process.

For an overview of the features in ZDT Patching, see Introduction to Zero Downtime Patching in *Administering Zero Downtime Patching Workflows*.

Clustering

WebLogic Server clusters provide scalability and reliability for your applications by distributing the work load among multiple instances of WebLogic Server.

For scalability, the capacity of an application deployed on a WebLogic Server cluster can be increased dynamically to meet demand. You can add server instances to a cluster without interruption of service—the application continues to run without impact to clients and end users.

In a WebLogic Server cluster, application processing can continue when a server instance fails. You cluster application components by deploying them on multiple server instances in the cluster—so, if a server instance on which a component is running fails, then another server instance on which that component is deployed can continue application processing. See Understanding WebLogic Server Clustering in *Administering Clusters for Oracle WebLogic Server*.

Singleton Services

Singleton services are services that must run on only a single Managed Server instance of a cluster at any given time, for example JMS and the JTA transaction recovery system. WebLogic Server allows you to automatically monitor and migrate singleton services from one server instance to another.

WebLogic Server features such as server and service migration, persistent data stores, and leasing make singleton services such as JMS and JTA highly available in a WebLogic Server cluster. See Singleton Services.

Session Replication

Session replication is a feature of WebLogic Server clusters that is used to replicate the data stored in a session across different server instances in the cluster.

WebLogic Server provides three methods for replicating HTTP session state across servers in a cluster: in-memory replication, JDBC-based persistence, and Coherence*Web. See Session Replication.

Transaction and Data Source Features

WebLogic Server features such as Active Gridlink data sources, JDBC TLogs and No TLog, and Logging Last Resource help to provide high availability in WebLogic Server configurations.

- Active GridLink data sources use Fast Connection Failover to provide rapid failure detection of Oracle Real Application Clusters (Oracle RAC) nodes, and failover to remaining nodes for continuous connectivity. For design considerations when using Active Gridlink in high availability architectures, see Data Sources. See Using Active GridLink Data Sources in *Administering JDBC Data Sources for Oracle WebLogic Server*.
- Transaction logs in the database (JDBC TLogs) store information about committed transactions coordinated by the server that may not have been completed. WebLogic Server uses the TLogs when recovering from system crashes or network failures. See Using Transaction Log Files to Recover Transactions in *Developing JTA Applications for Oracle WebLogic Server*.
- No transaction TLog writes (No TLog) where you eliminate writes of the transaction checkpoints to the TLog store. See XA Transactions without Transaction TLogs Write in *Developing JTA Applications for Oracle WebLogic Server*.
- Logging Last Resource (LLR) transaction optimization, which is a performance enhancement option that enables one non-XA resource to participate in a global transaction with the same ACID (atomicity, consistency, isolation, durability) guarantee as XA. See Logging Last Resource Transaction Optimization in *Developing JTA Applications for Oracle WebLogic Server*.

  These features work with Oracle Data Guard which replicates databases to make transaction logs needed for recovery to be highly available. See Introduction to Oracle Data Guard in *Oracle Data Guard Concepts and Administration*.

Coherence High Availability Components and Features

Oracle Coherence provides components and features that work in conjunction with Oracle WebLogic Server and Oracle Database high availability features to provide maximum availability, reliability, and application stability during planned upgrades or unexpected failures.

- Coherence Persistence and Clusters
- Coherence Federated Caching

- Coherence GoldenGate Hot Cache
- Coherence Persistence and Clusters
- Coherence Federated Caching
- Coherence GoldenGate HotCache

Coherence Persistence and Clusters

Coherence persistence is a set of tools and technologies that manage the persistence and recovery of Coherence distributed caches. Cached data is persisted so that it can be quickly recovered after a catastrophic failure or after a cluster restart due to planned maintenance. Persistence and federated caching can be used together as required. See Persisting Caches in *Administering Oracle Coherence*.

When an application asks for an entry to the Coherence cache, if the entry does not exist in the cache and does exist in the database, then Coherence updates the cache with the database value. This is called Read-Through caching. See Read-Through Caching in *Developing Applications with Oracle Coherence*.

Coherence clusters consist of multiple Coherence server instances that distribute data in-memory to increase application scalability, availability, and performance. Application data is automatically and transparently distributed and backed up across cluster members. See Configuring and Managing Coherence Clusters in *Administering Clusters for Oracle WebLogic Server*.

Coherence Federated Caching

The Oracle Coherence federated caching feature replicates cache data asynchronously across multiple geographically distributed clusters. Cached data is replicated across clusters to provide redundancy, off-site backup, and multiple points of access for application users in different geographical locations.

Federated caching supports multiple replication topologies. These include:

- Active-passive: Replicates data from an active cluster to a passive cluster. The passive site supports read-only operations and off-site backup.
- Active-active: Replicates data between active clusters. Data that is put into one active cluster is replicated at the other active clusters. Applications at different sites have access to a local cluster instance.
- Hub and spoke: Replicates data from a single hub cluster to multiple spoke clusters. The hub cluster can only send data and the spoke clusters can only receive data. This topology requires multiple geographically dispersed copies of a cluster. Each spoke cluster can be used by local applications to perform read-only operations.

See Federating Caches Across Clusters in *Administering Oracle Coherence*.

Coherence GoldenGate HotCache

The Oracle Coherence GoldenGate HotCache feature detects and reflects database changes in cache in real time. Third-party updates to the database can cause Coherence applications to work with data that can be stale and out-of-date. Coherence GoldenGate HotCache solves this problem by monitoring the database and pushing any changes into the Coherence cache in real time. It employs an efficient push model that processes only stale data. Low latency is assured because the data is pushed when the change occurs in the database.

In Maximum Availability Architectures, when the database is replicated to a secondary site during failover, the database changes are reflected to the cache using GoldenGate HotCache.

See Integrating with Oracle Coherence GoldenGate HotCache in *Integrating Oracle Coherence*.

Oracle Database High Availability and Disaster Recovery

Oracle WebLogic Server provides strong support for integrating with the high availability (HA) and disaster recovery features of Oracle Database. Integrating with these HA and disaster recovery features minimizes database access time while allowing transparent access to rich pooling management functions that maximize both connection performance and application availability.

> **Note:**
>
> For the most up-to-date details about the specific database versions that are supported with this release of WebLogic Server, see the Oracle Fusion Middleware Supported System Configurations page on Oracle Technology Network.

Oracle WebLogic Server and Coherence take advantage of the HA database features described in this section. The integration of all these products contributes to managing and orchestrating the failover and switchover of the Oracle Database, and makes the failover of the database fast and automatic.

- Oracle Data Guard ensures high availability, data protection, and disaster recovery for enterprise data. It provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production Oracle databases to survive disasters and data corruptions. Oracle Data Guard maintains these standby databases as transactionally consistent copies of the primary database. If the primary database becomes unavailable because of a planned or an unplanned outage, then Oracle Data Guard enables you to switch any standby database to the production role, thus minimizing the downtime associated with the outage. See Introduction to Oracle Data Guard in *Data Guard Concepts and Administration*.
- Oracle Active Data Guard is a comprehensive solution to eliminate single points of failure for mission critical Oracle Databases. It prevents data loss and downtime by maintaining a synchronized physical replica (standby) of a production database (primary). If there is an outage, client connections quickly failover to the standby and resume service. Active Data Guard achieves the highest level of data protection through deep integration with Oracle Database, strong fault isolation, and unique Oracle-aware data validation. System and software defects, data corruption, and administrator error that affect a primary are not mirrored to the standby. Idle redundancy is eliminated by directing read-only workloads and backups to active standby databases for high return on investment. See Getting Started with Oracle Data Guard in *Data Guard Concepts and Administration*.
- Oracle Data Guard broker logically groups these primary and standby databases into a broker configuration that enables the broker to manage and monitor them together as an integrated unit. It sends notifications to WebLogic Active GridLink which then makes new connections to the database in the failover site, and

coordinates with Oracle Clusterware to fail over role-based services. See Oracle Data Guard Broker Concepts in *Data Guard Broker*.

- Oracle Real Application Clusters (Oracle RAC) is a clustered version of Oracle Database that allows running multiple database instances on different servers in the cluster against a shared set of data files, also known as the database. The database spans multiple hardware systems and yet appears as a single unified database to the application. See Introduction to Oracle RAC in *Real Application Clusters Administration and Deployment Guide*.
- Oracle Clusterware manages the availability of instances of an Oracle RAC database. It works to rapidly recover failed instances to keep the primary database available. If Oracle Clusterware cannot recover a failed instance, then the broker continues to run automatically with one fewer instance. If the last instance of the primary database fails, then the broker provides a way to fail over to a specified standby database. If the last instance of the primary database fails, and fast-start failover is enabled, then the broker can continue to provide high availability by automatically failing over to a pre-determined standby database. See Introduction to Oracle Clusterware in *Oracle Clusterware Administration and Deployment Guide*.
- Oracle GoldenGate is a high-performance software application that uses log-based bidirectional data replication for real-time capture, transformation, routing, and delivery of database transactions across heterogeneous systems. Oracle GoldenGate allows for databases to be in active-active mode. Applications that use Oracle GoldenGate must have tolerance for data loss due to the asynchronous nature of Oracle GoldenGate replication. See Oracle GoldenGate Administration Overview in *Administering Oracle GoldenGate*.
- Oracle Database Global Data Services (GDS) streamline the delivery of database services on a global scale, which is key to deploying databases in MAA environments. These technologies oversee replication and failover while performing load balancing within and across data centers, optimizing resource utilization and streamlining database management practices in a distributed database environment. GDS works by enabling a Global Service across Oracle Real Application Clusters (RAC) and single-instance Oracle databases interconnected via Oracle Data Guard, Oracle GoldenGate, or any other replication technology. Client access to this distributed infrastructure is completely transparent. GDS implementations are easy to apply to Oracle WebLogic Server with minimal changes. See Introduction to Global Data Services in *Oracle Database Global Data Services Concepts and Administration Guide*.
- Application Continuity (AC) is available with the Oracle RAC, Oracle RAC One Node and Oracle Active Data Guard options that masks outages from end users and applications by recovering the in-flight database sessions following recoverable outages. Application Continuity enables replay, in a non-disruptive and rapid manner, of a database request when a recoverable error makes the database session unavailable. The request can contain transactional and nontransactional calls to the database and calls that are executed locally at the client or middle tier. After a successful replay, the application can continue where that database session left off. See Ensuring Application Continuity in *Real Application Clusters Administration and Deployment Guide*.

WebLogic Server Active GridLink integrates with the Oracle Database features like Application Continuity and Global Data Services to provide the highest possible availability. Application Continuity will replay transactions when encountered with unplanned database outages. End-user applications will not receive errors or even know that there have been outages. Active GridLink, Application Continuity, and Data Guard provide protection for planned and unplanned database outages in highly available environments.

These technologies oversee replication and failover while performing load balancing within and across data centers, optimizing resource utilization and streamlining database management practices in a distributed database environment.

Load Balancers

Load balancers provide high availability by ensuring that if one web server goes down, requests are routed to the remaining web servers that are up and running.

There are two types of load balancers: global load balancers and local load balancers. Load balancers can be hardware devices such as Big IP, Cisco, Brocade, and so on—or software applications.

A global load balancer is used when you have multiple sites that need to function as the same logical environment. Its purpose is to distribute requests between the sites based on a pre-determined set of rules. Global load balancers are typically used in disaster recovery deployments.

A local load balancer, such as Oracle HTTP Server, is used to distribute traffic within a site. In a typical deployment, at least two Oracle HTTP Server instances are configured in the web tier to provide high availability. See Oracle HTTP Server High Availability Architecture and Failover Considerations in *Administering Oracle HTTP Server*. A web tier with Oracle HTTP Server is not a requirement; you can route traffic directly from the hardware load balancer to the WebLogic Server instances in the application tier. However, a web tier provides several advantages, such as faster fail-over in the event of a WebLogic Server instance failure and HTTP redirection, which is why it is recommended as part of the supported MAA architectures.

Supported MAA Architectures

WebLogic Server supports three primary maximum availability architecture (MAA) solutions that can be used to protect an Oracle WebLogic Server system against downtime across multiple data centers.

MAA architectures span data centers in distributed geographical locations. Oracle MAA is Oracle's best practices blueprint based on proven Oracle high availability technologies, expert recommendations and customer experiences. The goal of MAA is to achieve optimal high availability for Oracle customers at the lowest cost and complexity.

See the following topics for details and design considerations for the WebLogic Server and Coherence supported MAA architectures:

- Active-Passive Application Tier with Active-Passive Database Tier
- Active-Active Application Tier with an Active-Active Database Tier
- **Active-Active Stretch Cluster with an Active-Passive Database Tier**

Potential Failure Scenarios

Potential failure scenarios range from unexpected full and partial site failures to maintenance outages.

The design considerations and recommendations provided in this document apply to the following potential failure scenarios:

- Full site failure - With full site failure, the database, the middle-tier application server, and all user connections fail over to a secondary site that is prepared to handle the production load.
- Partial site failure - In the context of this document, partial failures are at the mid-tier. Partial site failures at the mid-tier can consist of the entire mid-tier (WebLogic Server and Coherence), WebLogic Server only failure, Coherence cluster failure, or a failure in one instance of Oracle HTTP Server when two instances are configured for high availability.
- Network partition failure - The communication between sites fails.
- Maintenance outage - During a planned maintenance all components of a site are brought down gracefully. A switchover will take place from one site to the other.

2

Common Design Considerations for High Availability and Disaster Recovery

Oracle provides recommended design considerations and best practices for the Maximum Availability Architecture (MAA) solutions supported for Oracle WebLogic Server and Coherence. MAA architectures span data centers in distributed geographical locations. The goal of MAA is to achieve optimal high availability for Oracle customers at the lowest cost and complexity.
Topics in this chapter include:

- Global Load Balancer
- Web Tier
- WebLogic Server
- Coherence
- Database

The recommendations in this chapter apply to all of the WebLogic Server and Coherence supported MAA architectures. Recommendations that are specific to a particular architecture are provided in the subsequent chapters as follows:

- Active-Active Pair Topology Design Considerations
- Active-Passive Topology Design Considerations
- Active-Active Stretch Cluster Topology Design Considerations
- Global Load Balancer
- Web Tier
  Configuring a web tier is optional in the supported WebLogic Server MAA architectures. Web tier products such as Oracle HTTP Server (OHS) and Oracle WebLogic Server Proxy Plug-In are designed to efficiently front-end WebLogic Server applications. OHS and WebLogic Server Proxy Plug-in can be used with other WebLogic Server high availability features.
- WebLogic Server
  WebLogic Server features such as clustering, singleton services, session replication, and others can be used together with Coherence and Oracle Database features to provide the highest level of availability.
- Coherence
  Coherence features such as federated caching, persistence, and GoldenGate Hot Cache can be used together with WebLogic Server and Oracle Database features to provide the highest level of availability.
- Database

Global Load Balancer

When a global load balancer is deployed in front of the production and standby sites, it provides fault detection services and performance-based routing redirection for the two sites. Additionally, the load balancer can provide authoritative DNS name server equivalent capabilities.
In the event of a primary-site disaster and after the standby site has assumed the production role, a global load balancer is used to reroute user requests to the standby site. Global load balancers such as F5 –BigIP Global Traffic Manager (GTM) and Cisco –Global Site Selector (GSS) also handle DNS server resolution (by off loading the resolution process from the traditional DNS servers).

During normal operations, the global load balancer can be configured with the production site's load balancer name-to-IP mapping. When a DNS switchover is required, this mapping in the global load balancer is changed to map to the standby site's load balancer IP. This allows requests to be directed to the standby site, which now has the production role.

This method of DNS switchover works for both site switchover (planned) and failover (unplanned). One advantage of using a global load balancer is that the time for a new name-to-IP mapping to take effect can be almost immediate. The downside is that an additional investment must be made for the global load balancer. For instructions for performing a DNS switchover, see Manually Changing DNS Names in *Disaster Recovery Guide*.

Web Tier

Configuring a web tier is optional in the supported WebLogic Server MAA architectures. Web tier products such as Oracle HTTP Server (OHS) and Oracle WebLogic Server Proxy Plug-In are designed to efficiently front-end WebLogic Server applications. OHS and WebLogic Server Proxy Plug-in can be used with other WebLogic Server high availability features.

You can configure Oracle HTTP Server in one of two ways: as part of an existing Oracle WebLogic Server domain or in its own standalone domain. In the WebLogic Server and Coherence supported MAA architectures, the Oracle HTTP server instances are configured as separate standalone domains, where you can configure and manage the Oracle HTTP Server instances using WLST offline commands, independently of the application tier domains.

The `mod_wl_ohs` module handles the link to Managed Servers. You configure `mod_wl_ohs` by routing requests of a particular type, such as JSPs, or by routing requests destined to a URL to specific Managed Servers.

Oracle HTTP Server (OHS) has two failure types: process failures and node failures. An individual operating system process may fail. A node failure can involve failure of the entire host computer that OHS runs on.

- In a process failure, Node Manager protects and manages OHS processes. If an OHS process fails, Node Manager automatically restarts it.
- In a node failure, the load balancer in front of OHS sends a request to another OHS instance if the first one doesn't respond or URL pings to it indicate that it has failed.

- If a Managed Server in a cluster fails, the `mod_wl_ohs` module automatically redirects requests to one of the active cluster members. If the application stores state, state replication is enabled within the cluster, which enables redirected requests access to the same state information.

For more information about Oracle HTTP Server and WebLogic Server Proxy Plug-Ins, see:

- Introduction to Oracle HTTP Server in *Administering Oracle HTTP Server*
- Overview of Oracle WebLogic Server Proxy Plug-In in *Using Oracle WebLogic Server Proxy Plug-Ins*

WebLogic Server

WebLogic Server features such as clustering, singleton services, session replication, and others can be used together with Coherence and Oracle Database features to provide the highest level of availability.

The following sections provide the design considerations for these WebLogic Server features in a supported WebLogic Server MAA architecture:

- Clustering
- Singleton Services
- Session Replication
- Data Sources
- Security
- Storage
- Zero Downtime Patching

Clustering

A WebLogic Server cluster consists of multiple WebLogic Server server instances running simultaneously and working together to provide increased scalability, reliability, and high availability. A cluster appears to clients as a single WebLogic Server instance. The server instances that constitute a cluster can run on the same machine, or be located on different machines. You can increase a cluster's capacity by adding additional server instances to the cluster on an existing machine, or you can add machines to the cluster to host the incremental server instances. Each server instance in a cluster must run the same version of WebLogic Server.

WebLogic Server supports two types of clusters:

- Dynamic clusters - Dynamic clusters consist of server instances that can be dynamically scaled up to meet the resource needs of your application. When you create a dynamic cluster, the dynamic servers are preconfigured and automatically generated for you, enabling you to easily scale up the number of server instances in your dynamic cluster when you need additional server capacity. Dynamic clusters allows you to define and configure rules and policies to scale up or shrink the dynamic cluster.

  In dynamic clusters, the Managed Server configurations are based off of a single, shared template. It greatly simplifies the configuration of clustered Managed Servers, and allows for dynamically assigning servers to machine resources and greater utilization of resources with minimal configuration.

  Dynamic cluster elasticity allows the cluster to be scaled up or down based on conditions identified by the user. Scaling a cluster can be performed on-demand (interactively by the administrator), at a specific date or time, or based on performance as seen through various server metrics.

  When shrinking a dynamic cluster, the Managed Servers are shut down gracefully and the work/transactions are allowed to complete. If needed, singleton services are automatically migrated to another instance in the cluster.
- Static clusters - In a static cluster the end-user must configure new servers and add them to the cluster, and start and stop them manually. The expansion and shrinking of the cluster is not automatic; it must be performed by an administrator.

In most cases, Oracle recommends the use of dynamic clusters to provide elasticity to WebLogic deployments. The benefits of dynamic clusters are minimal configuration, elasticity of clusters, and proper migration of JMS and JTA singleton services when shrinking the cluster.

However, there are some instances where static clusters should be used. One such instance is - if you need to manually migrate singleton services. Dynamic clusters do not support manual migration of singleton services.

Singleton Services

A singleton service is a service running on a Managed Server that is available on only one member of a cluster at a time. WebLogic Server allows you to automatically monitor and migrate singleton services from one server instance to another.

Pinned services, such as JMS-related services and user-defined singleton services are hosted on individual server instances within a WebLogic cluster. To ensure that singleton JMS or JTA services do not introduce a single point of failure for dependent applications in the cluster, WebLogic Server can be configured to automatically or manually migrate them to any server instance in the cluster.

Within an application, you can define a singleton service that can be used to perform tasks that you want to be executed on only one member of a cluster at any give time. Automatic singleton service migration allows the automatic health monitoring and migration of user-defined singleton services.

Singleton services described in the following sections include:

- Server and Service Migration
- Data Stores
- Leasing

- Server and Service Migration
- Data Stores
- Leasing

Server and Service Migration

Oracle WebLogic Server supports two distinct types of automatic migration mechanisms:

- Whole server migration, where a migratable server instance, and all of its services, is migrated to a different physical machine upon failure. When a failure occurs in a server that is part of a cluster that is configured with server migration, the server is restarted on any of the other machines that host members of the cluster. See Whole Server Migration in *Administering Clusters for Oracle WebLogic Server*.
- Service migration, where failed services are migrated from one server instance to a different available server instance within the cluster. In some circumstances, service migration performs much better then whole server migration because you are only migrating the singleton services as opposed to the entire server. See Service Migration in *Administering Clusters for Oracle WebLogic Server*.

Oracle recommends to use the Service Migration feature instead of using Server Migration. Service Migration provides the same High Availability protection, by utilizing less resources. For example, the floating IPs required by Server Migration are not needed in Service Migration, and less memory or cpu resources are used as only the critical services are migrated instead of migrating the entire WebLogic server.

Both whole server and Service migration require that you configure a database leasing table. See Leasing.

Instructions for configuring WebLogic Server to use server and service migration in an MAA environment are provided in Using Whole Server Migration and Service Migration in an Enterprise Deployment in *Enterprise Deployment Guide for Oracle SOA Suite*.

Data Stores

There are two kinds of persistent data stores for Oracle WebLogic Server transactions logs and Oracle WebLogic Server JMS: database-based and file-based.

Keeping persistent stores in the database provides the replication and high availability benefits inherent in the underlying database system. With JMS, TLogs and the application in the same database and replication handled by Oracle Data Guard, cross-site synchronization is simplified and the need for a shared storage sub-system such as a NAS or a SAN is alleviated in the middle tier. See Database.

However, storing TLogs and JMS stores in the database has a penalty on system performance. This penalty is increased when one of the sites needs to cross communicate with the database on the other site. Ideally, from a performance perspective, shared storage that is local to each site should be used for both types of stores and the appropriate replication and backup strategies at storage level should be provisioned in order to guarantee zero data loss without performance degradation. Whether using database stores will be more suitable than shared storage for a system depends on the criticality of the JMS and transaction data, because the level of protection that shared storage provides is much lower than the database guarantees.

You can minimize the performance impact of database stores, especially when there is a large concurrency, by using techniques such as global hash partitions for indexes (if Oracle Database partitioning is available). For recommendations about minimizing the performance

impact, see Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment in *Enterprise Deployment Guide for Oracle SOA Suite*.

In active-active and active-passive topologies, keeping the data stores in the database is a requirement. Oracle recommends keeping WebLogic Server stores such as JMS and JTA stores, in a highly available database such as Oracle RAC and connecting to the database using Active GridLink data sources for maximum performance and availability.

In the case of an active-active stretch cluster, you can choose between keeping the data stores in a shared storage sub-system such as a NAS or a SAN, or in the database. However, using database-based stores is recommended also in stretch clusters, because high availability and cross-site replication is automatically provided by the underlying Oracle RAC database and Data Guard.

Oracle recommends keeping WebLogic Server stores such as JMS and JTA stores, and leasing tables in a highly available database such as Oracle RAC and connecting to the database using Active GridLink data sources. Storing the stores and leasing tables in the database provides the following advantages:

- Exploits the replication and other high availability aspects inherent in the underlying database system.
- Enhances handling of disaster recovery scenarios. When JMS, the TLogs and the application are in the same database and the replication is handled by Data Guard, there is no need to worry about cross-site synchronization.
- Alleviates the need for a shared storage sub-system such as a NAS or a SAN. Usage of the database also reduces overall system complexity since in most cases a database is already present for normal runtime/application work.

Leasing

Leasing is the process WebLogic Server uses to manage services that are required to run on only one member of a cluster at a time. Leasing ensures exclusive ownership of a cluster-wide entity. Within a cluster, there is a single owner of a lease. Additionally, leases can failover in case of server or cluster failure which helps to avoid having a single point of failure. See Leasing in *Administering Clusters for Oracle WebLogic Server*.

WebLogic Server provides two types of leasing functionality, non-database consensus leasing and high availability database leasing. In high availability or disaster recovery scenarios, Oracle recommends the use of database leasing.

For database leasing we recommend the following:

- A highly available database such as Oracle RAC and Active GridLink (AGL).
- A standby database, and Oracle Data Guard to provide replication between the two databases.

WebLogic Server includes an option to automatically create WebLogic cluster database leasing tables. This option automatically detects that a leasing table is missing, detects the database type, and then finds and runs the appropriate default DDL file to create the table. See High Availability Database Leasing in *Administering Clusters for Oracle WebLogic Server*.

When using database leasing, Oracle WebLogic Servers may shut down if the database remains unavailable (during switchover or failover) for a period that is longer

than their server migration fencing times. You can adjust the server migration fencing times as described in the following topics in *Administering Clusters for Oracle WebLogic Server*:

- Migratable Server Behavior in a Cluster
- Cluster Master Role in Whole Server Migration

Session Replication

WebLogic Server provides three methods for replicating HTTP session state across servers in a cluster:

- In-memory replication - Using in-memory replication, WebLogic Server copies a session state from one server instance to another. The primary server creates a primary session state on the server to which the client first connects, and a secondary replica on another WebLogic Server instance in the cluster. The replica is kept up-to-date so that it may be used if the server that hosts the servlet fails.
- JDBC-based persistence - In JDBC-based persistence, WebLogic Server maintains the HTTP session state of a servlet or JSP using file-based or JDBC-based persistence. For more information on these persistence mechanisms, see Configuring Session Persistence in *Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server*.
- Coherence*Web - Coherence*Web is not a replacement for WebLogic Server's in-memory HTTP state replication services. However, you should consider using Coherence*Web when an application has large HTTP session state objects, when running into memory constraints due to storing HTTP session object data, or if you want to reuse an existing Coherence cluster. See Using Coherence*Web with WebLogic Server in *Administering HTTP Session Management with Oracle Coherence*Web*.

Depending on the latency model, tolerance to session loss, and performance, you should choose the method that best fits your requirement.

- When the latency is small, such as in MAN networks (stretch cluster topology), Oracle recommends WebLogic Server in-memory session replication. However, if a site experiences a failure there is the possibility of session loss.
- When the latency is large (WAN networks), Active-Active, or Active-Passive topologies, and when your applications cannot tolerate session loss, Oracle recommends database session replication.

In most cases, in-memory session replication performs much better than database session replication. See Failover and Replication in a Cluster in *Administering Clusters for Oracle WebLogic Server*.

Data Sources

WebLogic Active GridLink data sources integrate with Oracle RAC databases and Oracle Data Guard to provide the best performance, high scalability and the highest availability. The integration with Oracle RAC enables Active GridLink to do Fast Connection Failover (FCF), Runtime Load Balancing (RCLB) and Affinity features. Active GridLink can handle planned maintenance in the database without any interruptions to end-users while allowing all work to complete.

You can configure your Active GridLink URL to minimize the time to failover between databases. See Supported AGL Data Source URL Formats in *Administering JDBC Data Sources for Oracle WebLogic Server*.

Security

It is important that you determine your security needs and make sure that you take the appropriate security measures before you deploy WebLogic Server and your Java EE applications into a production environment. See Ensuring the Security of Your Production Environment in *Securing a Production Environment for Oracle WebLogic Server*.

Storage

The Oracle Fusion Middleware components in a given environment are usually interdependent on each other, so it is important that the components in the topology are in sync. Some of the storage artifacts that you need to take into consideration in an MAA environment are classified as static and dynamic.

- **Static artifacts** are files and directories that do not change frequently. These include:
  - home: The Oracle home usually consists of an Oracle home and an Oracle WebLogic Server home.
  - Oracle Inventory: This includes `oraInst.loc` and `oratab` files, which are located in the `/etc` directory.
- **Dynamic or runtime artifacts** are files that change frequently. Runtime artifacts include:
  - Domain home: Domain directories of the Administration Server and the Managed Servers.
  - Oracle instances: Oracle Instance home directories.
  - Application artifacts, such as `.ear` or `.war` files.
  - Database artifacts, such as the MDS repository.
  - Database metadata repositories used by Oracle Fusion Middleware.
  - Persistent stores, such as JMS providers and transaction logs. As a best practice for High Availability and Disaster recovery, Oracle recommends storing these in database persistent stores. See Data Stores.
  - Deployment plans: Used for updating technology adapters such as file and JMS adapters. They need to be saved in a location that is accessible to all nodes in the cluster to which the artifacts are deployed.

For maximum availability, Oracle recommends using redundant binary installations. Each node should have its own Oracle home so that when you apply Zero Downtime Patches, only servers in one node need to come down at a time.

For recommended guidelines regarding shared storage for artifacts such as home directories and configuration files, see Using Shared Storage in *High Availability Guide*.

Zero Downtime Patching

Zero Downtime Patching (ZDT Patching) provides continuous application availability during the process of rolling out upgrades, even though the possibility of failures during the rollout process always exists. In an MAA environment, Oracle recommends

patching one site at a time, and staggering the update to the other site to ensure that the sites remain synchronized. In the case of a site failure scenario, allow for the failed site to resume before resuming ZDT Patching.

When using ZDT Patching, consider the following:

- Rollout shuts down one node at a time, so the more nodes in a cluster, the less impact it has on the cluster's ability to handle traffic.
- If a cluster has only two nodes, and one node is down for patching, then high availability cannot be guaranteed. Oracle recommends having more than two nodes in the cluster.
- If you include a Managed Server on the same node that includes the Administration Server, then both servers must be shutdown together to update Oracle home.
- Two clusters can have servers on the same node sharing an Oracle home, but both clusters need to be shutdown and patched together.
- If your configuration contains two Oracle homes, then Oracle recommends that you create and patch the second Oracle home on a nonproduction machine so that you can test the patches you apply, but this is not required. The Oracle home on that node must be identical to the Oracle home you are using for your production domain.

See Introduction to Zero Downtime Patching in *Administering Zero Downtime Patching Workflows*.

Coherence

Coherence features such as federated caching, persistence, and GoldenGate Hot Cache can be used together with WebLogic Server and Oracle Database features to provide the highest level of availability.

The following sections provide the design considerations for Coherence in the supported MAA architectures.

- Coherence Persistent Cache
- Coherence Federated Caching
- Coherence GoldenGate Hot Cache

- Coherence Persistent Cache
- Coherence Federated Caching
- Coherence GoldenGate Hot Cache

Coherence Persistent Cache

Cached data is persisted so that it can be quickly recovered after a catastrophic failure or after a cluster restart due to planned maintenance. In multi data center environments, Oracle recommends using Coherence persistence and federated caching together to ensure the highest level of protection during failure or planned maintenance events.

Persistence is only available for distributed caches and requires the use of a centralized partition assignment strategy. There are two persistence modes:

- Active persistence - cache contents are automatically persisted on all mutations and are automatically recovered on cluster/service startup.

- On-demand persistence - a cache service is manually persisted and recovered upon request using the persistence coordinator.

See Persisting Caches in *Administering Oracle Coherence*.

Coherence Federated Caching

You can use Coherence federated caching in active-active and active-passive topologies (not stretch clusters). Before doing so, consider these ramifications:

- Coherence data reaches the other site at some point in an ordered fashion (in Coherence, ordering is per Coherence partition), even after network partition or remote cluster outage.
- The remote site may read stale data for a period of time after the local site is being updated.
- Update conflicts are possible, and we identify these and call out to an application-specific conflict resolver.

Coherence federated caching implements an eventual consistency model between sites for the following reasons:

- The data center can be anywhere; the location is not constrained by latency or available bandwidth.
- Tolerance for unavailability of the remote data center or cluster is extremely desirable. Note that it is very hard to tell the difference between communications being down and a remote cluster being down, and it is not necessary to differentiate.
- Full consistency in active-active configurations requires some sort of distributed center concurrency control, as well as synchronous writes. This can have a significant impact on performance and is not desirable. Instead, where consistency matters, you can use stretch clusters with synchronous replications. In this case, it is reasonable to assert a maximum latency between data centers, with guaranteed bandwidth.

See Federating Caches Across Clusters in *Administering Oracle Coherence*.

Coherence GoldenGate Hot Cache

Within a single Coherence cluster with a mix of data, where some of the data is owned by the database and some of it is owned by Coherence, you can use both Coherence Read-Through cache and Coherence GoldenGate Hot Cache.

The choice between HotCache and Read-Through cache comes down to (1) whether Read-Through may lead to stale reads if the database is updated behind Coherence's back and (2) whether the real-time nature of HotCache is preferred for other reasons. There can also be situations where both HotCache and Read-Through are used together, for example to push real-time updates via HotCache, but then to handle the case where data was removed due to eviction or expiration.

See Integrating with Oracle Coherence GoldenGate HotCache in *Integrating Oracle Coherence*.

Database

Oracle Database provides several features such as Oracle Data Guard, Oracle Real Application Clusters (Oracle RAC) and others that can be integrated to provide high availability of the database in MAA architectures. See Oracle Database High Availability and Disaster Recovery. Regardless of the topology, the goal is to minimize the time that it will take for switchover and failover of your databases.
To achieve high availability of your database for both planned and unplanned outages, Oracle recommends using an active-passive configuration with a combination of the following features:

- Oracle RAC as the highly available database. See Introduction to Oracle RAC in *Real Application Clusters Administration and Deployment Guide*.
- Oracle Data Guard because it eliminates single points of failure for mission critical Oracle Databases. It prevents data loss and downtime by maintaining a synchronized physical replica of a production database at a remote location. If the production database is unavailable for any reason, client connections can quickly, and in some configurations transparently, failover to the synchronized replica to restore service. Applications can take advantage of Oracle Data Guard with little or no application changes required. See Introduction to Oracle Data Guard in *Data Guard Concepts and Administration*.

  In a supported WebLogic Server MAA architecture, Oracle recommends using the Oracle Data Guard maximum availability protection mode. This protection mode provides the highest level of data protection that is possible without compromising the availability of a primary database. It ensures zero data loss except in the case of certain double faults, such as failure of a primary database after failure of the standby database. See Oracle Data Guard Protection Modes in *Oracle Data Guard Concepts and Administration*.

  > **Note:**
  >
  > Oracle Data Guard can only be used in active-passive configurations, but guarantees zero-data loss.

- Oracle Active Data Guard, an option built on the infrastructure of Oracle Data Guard, allows a physical standby database to be opened read-only while changes are applied to it from the primary database. This enables read-only applications to use the physical standby with minimal latency between the data on the standby database and that on the primary database, even while processing very high transaction volumes at the primary database. This is sometimes referred to as real-time query. See Opening a Physical Standby Database in *Oracle Data Guard Concepts and Administration*.

  An Oracle Active Data Guard standby database is used for automatic repair of data corruption detected by the primary database, transparent to the application. In the event of an unplanned outage on the primary database, high availability is maintained by quickly failing over to the standby database. An Active Data Guard standby database can also be used to off-load fast incremental backups from the primary database because it is a block-for-block physical replica of the primary database.

  Oracle Active Data Guard provides a far sync feature that improves performance in zero data loss configurations. An Oracle Data Guard far sync instance is a remote Oracle Data Guard destination that accepts redo from the primary database and then ships that redo to other members of the Oracle Data Guard configuration. Unlike a standby database, a far sync instance does not have data files, cannot be opened, and cannot apply received

redo. These limitations yield the benefit of using fewer disk and processing resources. More importantly, a far sync instance provides the ability to failover to a terminal database with no data loss if it receives redo data using synchronous transport mode and the configuration protection mode is set to maximum availability. See Using Far Sync Instances in *Oracle Data Guard Concepts and Administration*.

- Oracle Data Guard broker as a distributed management framework that automates and centralizes the creation, maintenance, and monitoring of Data Guard configurations. Some of the operations Data Guard Broker can perform is the creation, management, monitoring of the Data Guard configurations, invoking switchover or failover to initiate and control complex role changes across all databases in the configuration, and configuring failover to occur automatically. See Oracle Data Guard Broker Concepts in *Data Guard Broker*.

  You can enable Oracle Data Guard fast-start failover to fail over automatically when the primary database becomes unavailable. When fast-start failover is enabled, the Oracle Data Guard broker determines if a failover is necessary and initiates the failover to the specified target standby database automatically, with no need for database administrator intervention. See Managing Fast-Start Failover in *Oracle Data Guard Broker*.
- Active GridLink Datasources in WebLogic Server makes the scheduled maintenance process at the database servers transparent to applications. When an instance is brought down for maintenance at the database server, draining ensures that all work using instances at that node completes and that idle sessions are removed. Sessions are drained without impacting in-flight work.
- Application continuity protects you during planned and unplanned outages. Use Application Continuity and Active GridLink for maximum availability during unplanned down events. See Ensuring Application Continuity in *Real Application Clusters Administration and Deployment Guide*.
- Global Data Services (GDS) or Data Guard broker with Fast Application Notifications (FAN) to drain across sites. When you use Active Data Guard, work can complete before switching over to secondary database.

If your configuration requires that you have an active-active database configuration, Oracle recommends:

- Oracle GoldenGate, which allows for databases to be in active-active mode. Both read and write services are active-active on the databases on both sites. See Configuring Oracle GoldenGate for Active-Active Configuration in *Administering Oracle GoldenGate*.

  When using Oracle Golden Gate, Application Continuity and Active GridLink can be used within a site (intra-site) to handle planned and unplanned down database events. Application Continuity *cannot* be used to replay transactions during failover or switchover operations across sites (inter-site). Application Continuity does not support failover to a logically different database –including Oracle Logical Standby and Oracle Golden Gate. Replay has a strict requirement that it applies to databases with verified no transaction loss.

> **Note:**
>
> Because of the asynchronous replication nature of Oracle GoldenGate, applications must tolerate data loss due to network lag.

- Implementing conflict resolution with full active/active for all applications/schema that are using Oracle GoldenGate.
- Designing an environment that requires web affinity to avoid seeing stale data (stick at a site in conversation). Global Data Services (GDS) can provide affinity to the database that is local to the site and manage global services. See Introduction to Global Data Services in *Oracle Database Global Data Services Concepts and Administration Guide*.

When environments require an active-active database, a combination of these technologies can be used to maximize availability and minimize data loss in planned maintenance events.

3

Active-Passive Application Tier with Active-Passive Database Tier

In an active-passive application tier topology, an active site is paired with a passive site that is on standby at a geographically different location.

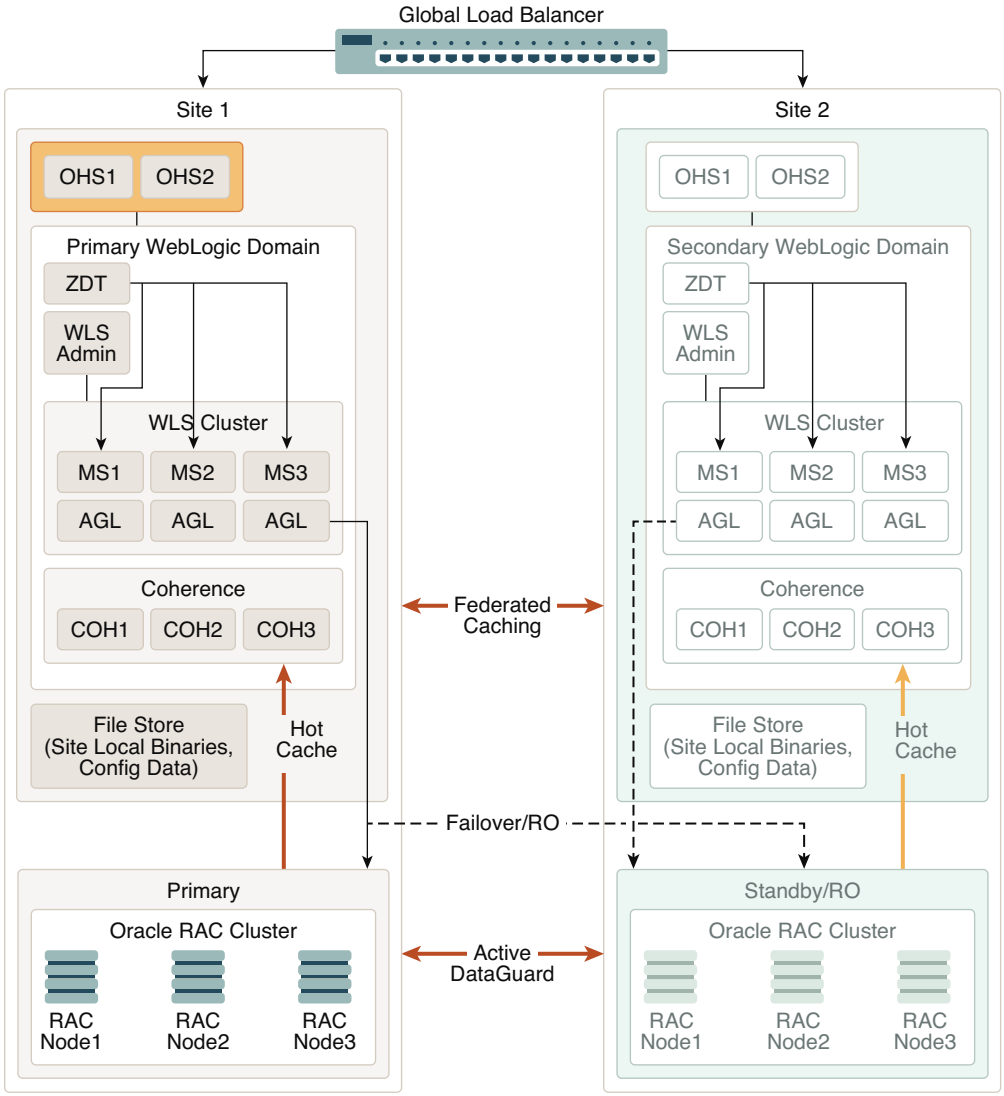
Topics include:

Active-Passive Topology Architecture Description

This MAA architecture consists of an active-passive application infrastructure tier with an active-passive database tier and in which both tiers span two geographically different sites. At the first site, both the application infrastructure tier and the database tier are active. At the second site, the secondary domain is shutdown, and the secondary database is on standby.

Figure 3-1 shows a recommended architecture using an active-passive application infrastructure tier with an active-passive database tier.

Figure 3-1 Active-Passive Application Tier with Active-Passive Database Tier Architecture Diagram



The key aspects of this topology include:

- A global load balancer.
- Two active instances of Oracle HTTP Server (OHS) on Site 1 and two passive instances of OHS on Site 2. OHS can balance requests to the WebLogic Server cluster.
- Two separate WebLogic Server domains configured in two different data centers, Site 1 and Site 2. The domain at Site 1 is active and the secondary domain at Site 2 is shutdown. The configuration of each active-passive domain must be identical. See Active-Passive Topology Design Considerations.

  The domains include:

  – A collection of Managed Servers (MS1, MS2, and MS3) in a WebLogic Server cluster, managed by the WebLogic Server Administration Server in the domain. In this sample, Active Gridlink (AGL) is being used to connect the

Managed Servers to the primary database. (Although a generic data source or multi data source can be used, Active Gridlink is preferable because it offers high availability and improved performance).

The Zero Downtime Patching (ZDT) arrows represent patching the Managed Servers in a rolling fashion. In this architecture, you can use the Zero Downtime Patching feature on the active domain in Site 1 as described in WebLogic Server Zero Downtime Patching. Because the servers are not running in the secondary domain at Site 2, you can use OPatch to patch the Oracle home. When the servers become active, they will point to the patched Oracle home. See Patching Your Environment Using OPatch in *Patching with OPatch*.

- A Coherence cluster (COH1, COH2, and COH3) managed by the WebLogic Server Administration Server in the domain. Coherence persistent caching is used to recover cached data in case of a failure in the Coherence cluster. Read-Through caching or Coherence GoldenGate Hot Cache is used to update cache from the database.

  Using Coherence Hot Cache in this architecture, updates on the active database at Site 1 update the Coherence cache in real time and the database updates are replicated to Site 2. When the data replication occurs on Site 2, HotCache updates the cache in real time. See Coherence GoldenGate HotCache.

  Coherence Federated Caching replicates data from the active cluster to the passive cluster. The passive site supports read-only operations and off-site backup. See Coherence Federated Caching.

- A file storage for the configuration data, local binaries, logs, and so on.
- Two separate Oracle RAC database clusters in two different data centers. The primary active Oracle RAC database cluster is at Site 1. Site 2 contains an Oracle RAC database cluster in standby (passive) read-only mode. The clusters can contain transaction logs, JMS stores, and application data. Data is replicated using Oracle Active Data Guard. (Although Oracle recommends using Oracle RAC database clusters because they provide the best level of high availability, they are not required. A single database or multitenant database can also be used.)
- The secondary WebLogic domain configuration is a replica of the primary domain. A replication technology (storage level replication, rsync, DBFS) is used to copy the middle-tier file systems and other data from the production site's storage to the standby site's storage.

Active-Passive Topology Design Considerations

Consider Oracle's best practice design recommendations for continuous availability in an active-passive application tier topology with an active-passive database tier.

To take full advantage of continuous availability features in an active-passive topology, consider the following:

- All active-passive domain pairs must be configured with symmetric topology; they must be identical and use the same domain configurations such as directory names and paths, port numbers, user accounts, load balancers and virtual server names, and software versions. Host names (not static IPs) must be used to specify the listen address of the Managed Servers. When hostnames between sites are identical (IPs are not), the hostname provides the dynamic ability to start an identically configured server or domain on the recovery site.
- There are no latency considerations in this topology. The only requirement is that stores such as JMS and JTA TLogs are kept in the database.

- In passive mode, WebLogic Server servers are configured but not running. When there is a failure, the WebLogic Server servers in the passive site are brought up and JTA and JMS transactions/messages are recovered. Work then takes place in the second site.
- The passive site must contain an exact copy of the WebLogic domain config of the primary site. A replication technology (storage level replication, rsync, DBFS) is used to copy the middle tier file systems and other data from the production site's storage to the standby site's storage.

  After storage replication is enabled, application deployment, configuration, metadata, data, and product binary information are replicated from the production site to the standby site.
- JMS is supported in this topology. Using database stores for JMS is recommended, because high availability and cross-site replication is automatically provided by the underlying Oracle RAC database and Data Guard.

  In the case of a failover or switchover, if the store and/or JMS server is targeted at the cluster, you must start the same number of servers in the cluster. This process is required because each JMS server + store + destination is specific to the server on which it was running. If you have MyServer1 and MyServer2 in the primary domain, there is a JMS Server + store on each of those servers. It is possible that the queues on those servers contain messages. If you restart only one server, you only recover the messages for that one server.

  The standby domain cannot be running during the replication phase and started only after the initial data center is confirmed as down. This process is necessary to prevent data corruption and to force recovery. If two JMS server/stores are writing/updating the same exact data, unexpected results occur. Also, message recovery only happens on JMS server startup. Then, the JMS server reads the full store contents and recreates destination state.

  Asynchronous replication can result in lost messages (message was written in the primary datacenter but not copied) or duplicate messages (message was consumed/deleted in the primary data center, but remains in the replicated data center), hence using database-based stores is recommended to take advantage of the underlying Data Guard replication.

  See Recommendations for Oracle WebLogic Server Java Message Service (JMS) and Transaction Logs (T-Logs) in *Disaster Recovery Guide*.
- Zero Downtime Patching upgrades your WebLogic homes, Java, and applications in a rolling fashion in the active site. During a planned maintenance and switchover to the passive site and after servers have been started, use Zero Downtime Patching to upgrade WebLogic, Java, or applications. To keep the domains symmetric at both sites, keep upgrade versions on both sites in sync.
- In the active-passive topology, Coherence is in Standby mode, not passive mode like WebLogic Server. The standby site has a backup of the cache data from the active site. With Coherence Federated Cache in active-passive mode, this replication happens asynchronously but it happens constantly.
- In this topology Oracle recommends using Oracle Site Guard to orchestrate the failover/switchover of all site components: Oracle Traffic Director, WebTier, WebLogic Server, Coherence, and the database.

4

Active-Active Application Tier with an Active-Active Database Tier

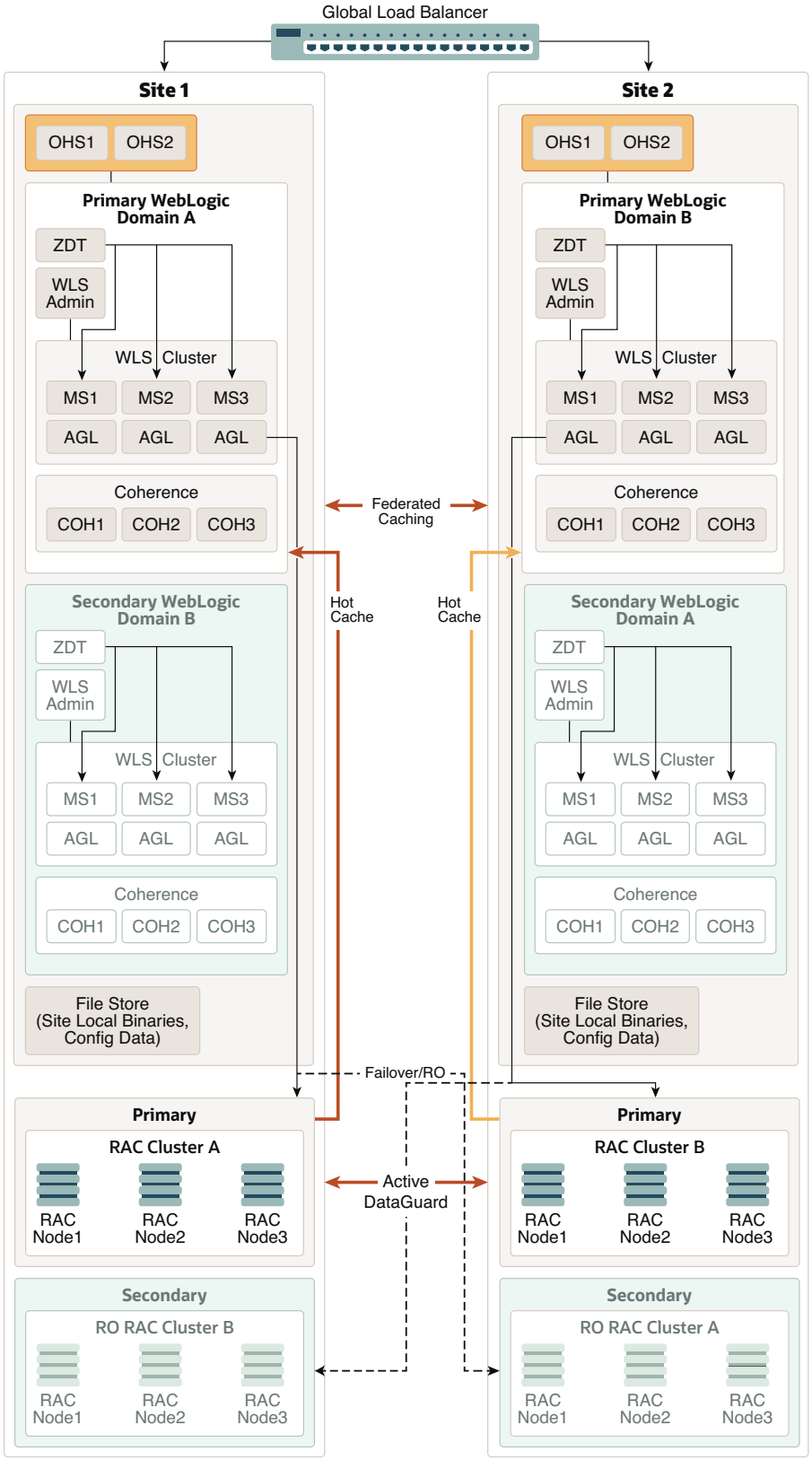
In an active-active application tier topology, two or more active server instances at distributed geographic locations are deployed to handle requests concurrently and thereby improve scalability and provide high availability.

Topics include:

Active-Active Pair Topology Architecture Description

This supported MAA architecture consists of an active-active application infrastructure tier with WebLogic domain pairs used in conjunction with an active-active database tier with Oracle RAC cluster pairs, and in which both tiers span two sites. The application infrastructure tier contains a domain pair with an identical active and passive WebLogic domain on each site, and the database tier contains a database cluster pair, with an identical active and passive database cluster on each site.
Figure 4-1 shows a recommended solution using an active-active application pair infrastructure tier with an active-active database pair tier.

Figure 4-1 Active-Active Application Pair Tier with an Active-Active Database Pair Tier Architecture Diagram



The key aspects of this sample topology include:

- A global load balancer.
- Two active instances of Oracle HTTP Server (OHS) at each site. OHS can balance requests to the WebLogic Server cluster.
- Two identical domain pairs configured in two different data centers, Site 1 and Site 2. Domain A and B are independent domains and do not have to be configured with a symmetrical topology, however the domain pair at each site must be symmetrical. Each site contains a domain pair with one active domain and one passive domain. See Active-Active Pair Topology Design Considerations. The domains include:
  - A collection of Managed Servers (MS1, MS2, and MS3) in a WebLogic Server cluster, managed by the WebLogic Server Administration Server in the domain. In this sample, Active Gridlink (AGL) is being used to connect the Managed Servers to the primary database located in the same site. (Although a generic data source or multi data source can be used, Active Gridlink is preferable because it offers high availability and improved performance).

    The Zero Downtime Patching (ZDT) arrows represent that the Managed Servers are patched in a rolling fashion. Because both domains are active, you can orchestrate the roll out of updates separately on each site. See WebLogic Server Zero Downtime Patching.
  - A Coherence cluster (COH1, COH2, and COH3) managed by the WebLogic Server Administration Server in the domain. Coherence persistent caching is used to recover cached data in case of a failure in the Coherence cluster. See Coherence Persistence and Clusters.

    Read-Through caching or Coherence GoldenGate Hot Cache is used to update cache from the database. Coherence Hot Cache updates the Coherence cache in real time for any updates that are made on the active database. See Coherence GoldenGate Hot Cache

    Coherence Federated Caching replicates data between the active clusters. In this active-active architecture, you can use the full capabilities of this feature, as described in Coherence Federated Caching. Data that is put into one active cluster is replicated at the other active clusters. Applications at different sites have access to a local cluster instance.
- A file storage for the configuration data, local binaries, logs, and so on.
- Two separate Oracle RAC database cluster pairs in the two different data centers, Site 1 and Site 2. Both sites contain a primary active Oracle RAC database cluster and a standby (passive) Oracle RAC database cluster. On Site 1, Cluster A is the active primary cluster, and Cluster B is in standby mode. On Site 2, Cluster B is the primary Active Oracle RAC cluster and Cluster A is in standby mode. The clusters can contain transaction logs, JMS stores, and application data. Data is replicated using Oracle Active Data Guard. (Although Oracle recommends using Oracle RAC database clusters because they provide the best level of high availability, they are not required. A single database or multitenant database can also be used.)

Active-Active Pair Topology Design Considerations

Consider Oracle's best practice design recommendations in an active-active application pair tier topology with an active-passive database pair tier.

In addition to the design considerations described here, you should also follow the best practices recommended for all supported WebLogic Server and Coherence MAA architectures. See Common Design Considerations for High Availability and Disaster Recovery.

To take full advantage of high availability features in an active-active topology, consider the following:

- Domain A and B are independent domains and do not have to be configured with a symmetrical topology, however the domain pair at each site must be symmetrical. That is, the domain pair must use the same domain configuration such as domain and server names, resource names, port numbers, user accounts, load balancers and virtual server names, and the same version of the software. Host names (not static IPs) must be used to specify the listen address of the Managed Servers. You can use any existing replication technology or methods that you currently use to keep the pairs in sync.
- In this topology network latency is normally large (WAN network). If applications require session replication between sites you must choose either database session replication or Coherence*Web. See Session Replication.
- Server and service migration only applies to Managed Servers in a cluster intra-site (within a site) in an active-active topology. See Server and Service Migration.
- JMS is supported only intra-site in this topology. JMS recovery during failover or planned maintenance is not supported across sites.
- Zero Downtime Patching is only supported intra-site (within a site) in an active-active topology. You can upgrade your WebLogic homes, Java, and applications in each site independently. Keep upgrade versions in sync to keep the domains symmetric at both sites.
- You can design applications to minimize data loss during failures by combining different high availability features. For example, you can use a combination of Coherence federated cache, Coherence HotCache or Coherence Read-Through cache.

  If the Coherence data is backed up in the database and there is a network partition failure, federated caching is unable to perform the replication and data becomes inconsistent on both sites since the Coherence clusters can independently continue doing work. Once the communication between the sites resumes, backed up data is pushed from the database to Coherence via Coherence HotCache or Coherence Read-Through cache, and eventually data in the Coherence cache is synchronized. See Coherence.

5

Active-Active Stretch Cluster with an Active-Passive Database Tier

In an active-active stretch cluster topology, cluster nodes can span data centers within a proximate geographical range, and usually with guaranteed, relatively low latency networking between the sites.

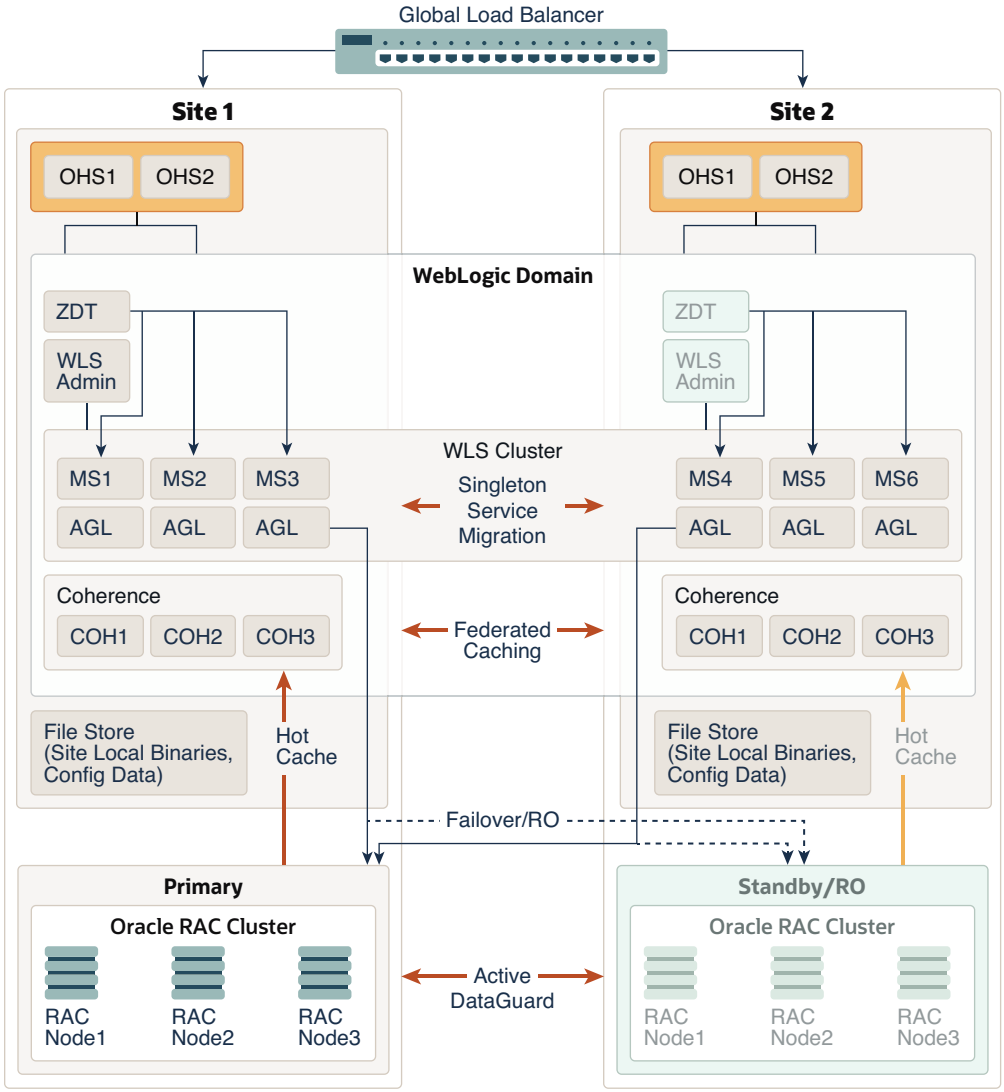
Topics include:

Active-Active Stretch Cluster Topology Architecture Description

This supported MAA architecture consists of an active-active stretch cluster application infrastructure tier with an active-passive database tier and in which the two tiers span two sites. Both sites are configured with a WebLogic Server stretch cluster, and all server instances in each cluster are active. The database tier is active at the first site, but on standby at the second.

Figure 5-1 shows a recommended architecture using an active-active stretch cluster application infrastructure tier with an active-passive database tier.

Figure 5-1 Active-Active Stretch Cluster with an Active-Passive Database Tier Architecture Diagram



The key aspects of this topology include:

- A global load balancer.
- Two active instances of Oracle HTTP Server (OHS) at each site. OHS can balance requests to the WebLogic Server cluster.
- WebLogic Server configured as a cluster that stretches across two different data centers, Site 1 and Site 2. All servers in the cluster are active. The cluster can be either dynamic or static. See Clustering.
- The domain includes:
  - A WebLogic Server cluster that consists of a group of Managed Servers (MS1, MS2, and MS3) at Site 1 and another group of Managed Servers (MS4, MS5, and MS6) at Site 2. The Managed Servers are managed by the WebLogic Server Administration Server at Site 1. In this sample, Active Gridlink (AGL) is being used to connect the Managed Servers to the primary database.

(Although a generic data source or multi data source can be used, Active Gridlink is preferable because it offers high availability and improved performance).

The Zero Downtime Patching (ZDT) arrows represent patching the Managed Servers in a rolling fashion. Because all of the servers in the cluster are active, you can use the full capabilities of this feature as described in WebLogic Server Zero Downtime Patching.

Because all of the servers are in the same cluster, you can use WebLogic Server singleton service migration to recover transactions.

* In whole server migration, a migratable server instance, and all of its services, is migrated to a different physical machine upon failure. See Whole Server Migration in *Administering Clusters for Oracle WebLogic Server*.
* In service migration, in the event of failure, services are moved to a different server instance within the cluster. See Service Migration in *Administering Clusters for Oracle WebLogic Server*.

– A Coherence cluster at each site (COH1, COH2, and COH3) managed by the WebLogic Server Administration Server in the domain. Coherence persistent caching is used to recover cached data in case of a failure in the Coherence cluster. Read-Through caching or Coherence GoldenGate Hot Cache is used to update cache from the database. In this architecture, you can use the full capabilities of Coherence HotCache as described in Coherence GoldenGate Hot Cache.

  Coherence Federated Caching replicates cache data asynchronously across the two sites. In this architecture, you can use the full capabilities of this feature as described in Coherence Federated Caching.

- A file storage for the configuration data, local binaries, logs, domain home, Node Manager directories, shared application-specific files, and shared cluster files. Because this is a stretch cluster, the shared cluster files (TLogs, JMS stores, leasing tables) must all be stored in the database so that they can be shared by servers in the cluster.
- Two separate Oracle RAC database clusters in two different data centers. The primary active Oracle RAC database cluster is at Site 1. Site 2 contains an Oracle RAC database cluster in standby (passive) read-only mode. The clusters can contain transaction logs, JMS stores, and application data. Data is replicated using Oracle Data Guard or Active Data Guard. (Although Oracle recommends using Oracle RAC database clusters because they provide the best level of high availability, they are not required. A single database or multitenant database can also be used.)
- The Oracle WebLogic Server configuration is synchronized across multiple nodes in the same domain by the Oracle WebLogic Server infrastructure. Most of this configuration usually resides under the Administration Server's domain directory. This configuration is propagated automatically to the other nodes in the same domain that contain Oracle WebLogic Servers. Based on this, the administration overhead of an Active-Active stretch cluster system is smaller as compared to any active-passive approach, where constant replication of WebLogic's configuration changes is required.

Active-Active Stretch Cluster Topology Design Considerations

Consider Oracle's best practice design recommendations in an active-active stretch cluster topology with an active-passive database tier.

In addition to the design considerations described here, you should also follow the best practices recommended for all supported WebLogic Server and Coherence MAA

architectures. See Common Design Considerations for High Availability and Disaster Recovery

To take full advantage of high availability features in an active-active stretch cluster, consider the following:

- In a multi data center, active-active stretch cluster environment session replication across data centers can cause serious performance degradation in the system. Oracle recommends defining two different replication groups (one for each site) to minimize the possibility of replication occurring across the two sites.

  > **Note:**
  >
  > Using replication groups is a best effort to replicate state only to servers in the same site, but is not a deterministic method. If one single server is available in one site, and other servers are available in the other site, replication occurs across the MAN and continues for that session even if servers come back online in the same site.

- A stretch cluster uses an Oracle HTTP Server (OHS) configuration based on a fixed list of servers at each site (instead of the dynamic list provided by the OHS plug-in that is used in typical single-location deployments). Using a fixed list of servers eliminates undesired routing from one site to another. A disadvantage, however, is slower reaction times to failures in Oracle WebLogic Servers.
- An active-active stretch cluster only works in the metro latency model. Latency should be no longer than 10-milliseconds round-trip time (RTT).
- For contention and security reasons, Oracle does not recommend using shared storage across sites. Each site uses individual shared storage for JMS and Transaction Logs (TLogs), or alternatively the database is used as a persistent store. Whether a database store is more suitable than shared storage for a system depends on the criticality of the JMS and transaction data, because the level of protection that shared storage provides is much lower than the protection guaranteed by the database. With JMS, TLog, and leasing tables in a Data Guard database, cross-site synchronization is simplified and the need for a shared storage sub-system such as a NAS or a SAN is alleviated in the middle tier. Using TLogs and JMS in the database has a penalty, however, on the system's performance. This penalty is increased when one of the sites needs to cross communicate with the database on the other site (depending on network lag between sites).
- Each site uses individual shared storage. If this is used to store runtime data, disk mirroring and replication from Site1 to Site2, and in reverse, can be used to provide a recoverable copy of these artifacts in each site.
- It is recommended to store the JMS and Transaction Logs (TLogs) in database-based stores. This allows cross-site service migration for the JMS and JTA services, because the TLogs and JMS messages are available in the database for all the servers in the cluster. And, in case of a complete site switchover, the Tlogs and JMS messages are automatically replicated to the other site with the underlying Data Guard replication.
- Both sites are managed with a single Administration Server that resides in one of the two sites. A unique Oracle WebLogic Server Administration Console is used to configure and monitor servers running on both sites. The WebLogic Server

infrastructure is responsible for copying configuration changes to all the different domain directories used in the domain.

- If an Administration Server fails, the same considerations that apply to an Administration Server failure in a single data center topology apply to a multi data center active-active stretch cluster topology. Use the standard failover procedures described in Failing Over or Failing Back Administration Server in *High Availability Guide* to address node failures (that is restarting the Administration Server in another node that resides in the same data center pointing to the shared storage that hosted the Administration Server domain directory). Also, deploy the appropriate backup and restore procedures to make regular copies of the Administration Server domain directory. If there is a failure that affects the site hosting the Administration Server (involving all nodes), you need to restart the server in a different site. To do so, use the existing storage replication technology to copy the Administration Server domain directory available in the failover site. Restore the server/directory (including both the domain and applications directories) in the failover site so that the exact same domain directory structure is created for the Administration Server domain directory as in the original site. Restart Node Manager in the node where the Administration Server is restored.

  Likely, the Administration Server failed over to a different subnet requiring the use of a different virtual IP (VIP) that is reachable by other nodes. Make the appropriate changes in the host name resolution system in this subnet so that this VIP maps to the original Virtual Hostname that the Administration Server used as the listen address in Site1. For example, in Site1, `ADMINHOSTVHN1` maps to `10.10.10.1`, while in Site2 either the local `/etc/hosts` or DNS server has to be updated so that `ADMINHOSTVHN1` maps to `20.20.20.1`. All servers use `ADMINHOSTVHN1` as the address to reach the Administration Server. If the Administration Server is front ended with an Oracle HTTP Server and load balancer, clients are agnostic to this change. If clients directly access the Administration Server listen host name, they must be updated in their DNS resolution also.

  Also, if host name verification is enabled for the Administration Server, update the appropriate trust stores and key stores with new certificates. Use the instructions in Updating Self-Signed Certificates and Keystore on Standby Site in *Disaster Recovery Guide*.

  Verify that the Administration Server is working properly by accessing the Oracle WebLogic Server Administration Console.

- Servers that are remote to the Administration Server take longer to restart than the servers that are collocated. The reason is that all the communications with the Administration Server (for retrieving the domain configuration upon start) and initial connection pool creation and database access is affected by the latency across sites. See Administration Server High Availability Topology in *High Availability Guide*.
- Automatic Server or Service Migration across sites is not recommended unless a database is used for JMS and TLog persistence, otherwise a constant replica of the appropriate persistent stores must be set up between the sites.
- Oracle recommends using Service Migration instead of Server migration. Server migration uses Virtual IPs and in most scenarios the Virtual IPs used in one site are invalid for migration to the other. It requires additional intervention to enable a listen address, which is initially available in Site1 in Site2 and viceversa. This intervention can be automated in pre-migration scripts, but the RTO increases compared to a standard automated server migration (taking place in the scope of single data center). When compared, Service Migration does not require virtual IPs and the RTO is much better than in Server migration.
- JMS and transaction recovery across sites is handled by using service or server migration inside a WebLogic Server stretch cluster. As explained in the Common Design

Considerations: WebLogic Server section, Oracle recommends using Service Migration rather than Server Migration. For server or service migration of JMS or JTA, Oracle recommends using database leasing on a highly available database. When configured with consensus non-database leasing, servers in the stretch cluster could fail to reboot and require the entire domain to be restarted when the environment experiences network partition failure.

- Zero Downtime Patching in an active-active stretch cluster topology orchestrates the updates to all servers in the stretch cluster across both sites. In a stretch cluster, the servers at each site must have their own Oracle Home. When an Oracle Home is updated and servers share the same Oracle Home, all servers must come down and are updated simultaneously. When each server has its own Oracle Home, each server can be patched individually and other servers in the stretch cluster can still service requests. See Zero Downtime Patching.
- In an active-active stretch cluster topology, only stretch the WebLogic Server cluster across the two sites. Coherence should have a Coherence cluster on each site using federated caching to replicate data across the two active sites. When a Coherence cluster is stretched across sites, it is susceptible to split brain.
- Table 5-1 lists the recommended settings to configure database leasing in a stretch cluster topology. See the following MBean descriptions in *MBean Reference for Oracle WebLogic Server*:
  - `ClusterMBean`
  - JDBCConnectionPoolParamsBean

Table 5-1 Recommended Settings for DataBase Leasing in a Stretch Cluster

Configuration Property	MBean/ Command	Description	Recommended Setting
`DatabaseLeasingBasisConnectionRetryCount`	`ClusterMBean`	The maximum number of times that database leasing tries to obtain a valid connection from the data source.	5 (Default value is 1)
`DatabaseLeasingBasisConnectionRetryDelay`	`ClusterMBean`	The length of time, in milliseconds, that database leasing waits before attempting to obtain a new connection from the data source when a connection has failed.	2000 (Default value is 1000)
`TestConnectionOnReserve`	`JDBCConnectionPoolParamsBean`	Enables WebLogic Server to test a connection before giving it to a client.	Enabled. For leasing the data source. The servers remain in a running state during switchover.

Table 5-1 (Cont.) Recommended Settings for DataBase Leasing in a Stretch Cluster

Configuration Property	MBean/ Command	Description	Recommended Setting
-Dweblogic.cluster.jta.SingletonMasterRetryCount	Server start up command	Specifies the retry count for the singleton master to get elected and open its listen ports. The singleton master might not be immediately available on the first try to deactivate JTA.	4 (The default value is 20) Because DatabaseLeasingBasisConnectionRetryCount is set to 5, this property can be decreased to 4. This setting can reduce the time cost during database server booting.

- Figure 5-2 and Figure 5-3 represent results found during benchmark testing that show the degradation observed in the overall system throughput (both sites working together) for different latencies. Figure 5-2 shows that for a latency of around 20-milliseconds round-trip time (RTT), the throughput decreases by almost 25%. Figure 5-3 shows the additional time (msecs) consumed for deploying SOA composites with increasing latencies (RTT in msecs.) between sites in a SOA Active-Active stretch cluster (as compared to a deployment with all servers and database in the same site).

  Considering the data provided and the performance penalties observed in many tests, Oracle recommends not to exceed 10 ms of latency (RTT) for active-active stretch cluster topologies when the latency affects communication between the two sites.

Figure 5-2 Throughput Degradation With Latency in a Stretch Cluster

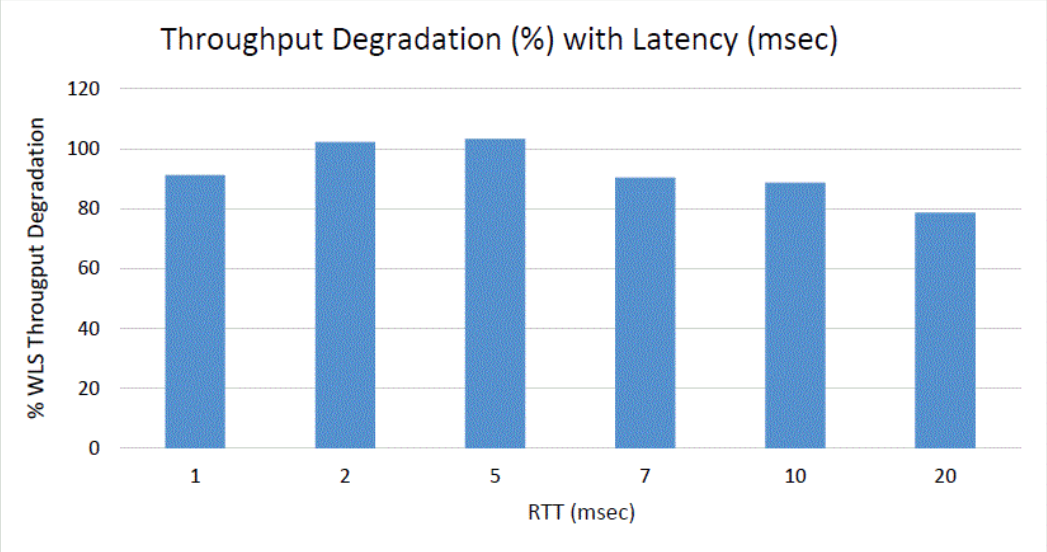


Figure 5-3 Deployment Delay Vs. Latency in a Stretch Cluster