

Oracle Maximum
Availability Architecture

Best Practices for Oracle Fusion Middleware SOA 12c Multi Data Center Active-Active Deployment

Oracle Maximum Availability Architecture

ORACLE | DECEMBER 2017





Contents

Paradigms for Designing a Multi Data Center Active-Active Deployment for Oracle

Fusion Middleware SOA	5
Availability: RTO and RPO	5
Performance	6
Administration	6
Latency, Jitter, Packet Loss and Bandwidth Across Sites	7
Requirements	15
Topology	15
Network	15
Shared Storage vs. Database for Transaction Logs and Persistent stores	15
Load Balancers	15
Oracle Fusion Middleware SOA Components and Versions in Scope	15
Hardware Resources and Capacity Utilization	16
Topology Model for an Oracle Fusion Middleware SOA Active-Active Multi Data Center	
Deployment	17
Database Tier	17
Load Balancers and Web Servers	18
Application Layer	19
Characteristics of the Design	21
Other Resources	23
Configuring the Oracle Fusion Middleware SOA Active-Active Topology	24



Configuring Load Balancers and Global Load Balancers for Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment	24
Configuring Oracle HTTP Server for Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment	26
Configuring the Application Tier of an Oracle Fusion Middleware SOA AA DR System for a Stretched Cluster	27
Configuring Data Sources for Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment	36
Composite and MDS Deployments and Updates: Oracle Coherence Configuration In-memory soa	39
Setting Appropriate Timeouts for Synchronous and Asynchronous Operations	44
Session Replication Implications	47
Optimizing Oracle Net Services Performance	48
Configuring I/O Buffer Size in the Database Server	49
Configuring I/O Buffer Size on the Oracle Fusion Middleware Nodes	49
Configuring Session Data Unit	50
Failures in Different Tiers and Switchover/Failover Behavior	53
Failure in All OHS Instances in One Site	53
Failure in All Oracle WebLogic Server SOA Servers in One Site	53
Administration Server Failure	53
Database Failures: Data Guard Switchover and Failover	55
Performance and Scalability Implications for an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment	56



Capacity Usage and Planning	56
Start Latencies	57
Average Active Time for Transactions and Transaction Recovery	60
Summary	62
Appendix A: File Adapter Locks and Muxers	63
Appendix B: Configuring in-place restart for JMS JDBC persistent stores	64
Appendix D: Oracle Service Bus Considerations	67
Load balancer considerations	67
Application Tier considerations	67
OSB Performance in a Stretched Cluster	68
References	74



Introduction

Business continuity is a key requirement for many e-business operations. Downtime of mission-critical applications translates directly into reduction in productivity, service quality, and lost revenue. Mission-critical application services require both a local high availability solution and a disaster recovery solution. A local high availability solution provides redundancy in one data center. Additionally, applications need protection from unforeseen disasters, natural calamities, and downtime that can affect an entire data center. An effective disaster that disables an application service is not necessarily one that destroys the whole data center (e.g. flood, fire), but is more likely to disable one particular type of resource. For example, a failure of corporate gateways or ISP network connections, a spread of viruses to all HTTP listener nodes, a miss configuration, a power outage, or an incorrect patch could all lead to days of complete loss of services. The same applies to planned outages: a network infrastructure update, a firewall upgrade, etc. may have similar downtime effects in a datacenter. In a Service Oriented Architecture (SOA) multiple corporate systems may depend on a unique service provider. As the adoption of these architectures grows, so does the need for failure and downtime protection not only in the scope of a single machine, but also against events that may bring down a group of machines, an entire room or an entire building. Traditional disaster protection systems use a model where one site is running while another site is on standby in prevention of possible failover scenarios (also called Multi Data Center Active-Passive or Active-Passive Disaster Protection). Such approaches usually incur increased operational and administration costs, while the need for continuous use of resources and increased throughput (i.e. avoiding situations where the standby machines are idle) have increased through the years. IT systems' design is increasingly driven by capacity utilization and even distribution of load, which leads to the adoption of disaster protection solutions that use, as much as possible, all resources available (called Multi Data Center Active-Active or Active-Active Disaster Protection).

This paper describes the recommended Active-Active solutions that can be used for protecting an Oracle Fusion Middleware 12c SOA system against downtime across multiple locations (referred to as SOA Active-Active Disaster Recovery Solution or SOA Multi Data Center Active-Active Deployment) It provides the required configuration steps for setting up the recommended topologies and guidance about the performance and failover implications of such a configuration.



Paradigms for Designing a Multi Data Center Active-Active Deployment for Oracle Fusion Middleware SOA

There are multiple factors that can drive the design of a Multi Data Center Deployment. The following are usually considered:

Availability: RTO and RPO

Disaster Recovery designs need to minimize the Recovery Point Objective (RPO) and the Recovery Time Objective (RTO) metrics. RPO measures the amount of data that can be lost in the event of a failure while RTO measures the time the system will be unavailable should a failure occur.

In most Multi Data Center Active-Active Deployments or Active-Active Disaster Recovery systems the reality is that the database tier is usually active in only one site. There are alternatives to this approach (Oracle Real Application Clusters on Extended Distance Clusters, Cross-Site Caching (Oracle GoldenGate), and database replication (Streams)) however these solutions are either very demanding in terms of infrastructure required or require specific data types and rules with which not all applications are compliant.

The main advantage of a Multi Data Center Active-Active Deployment system as compared to traditional Multi Data Center Active-Passive Disaster Recovery design is that in the event of complete middle tier failure in one site (all middle tier servers in one location), the system can fulfill requests because there are middle tiers in the peer site that remain available. In other words, RTO and RPO for Multi Datacenter Active-Active Deployments are null in this type of scenario. For this, the middle tier servers in the alternative location need to be able to sustain the combined load of all locations. The appropriate capacity planning must be done to account for such scenarios. Depending on the design, requests from end clients may need to be throttled when only one site is active. Otherwise, sites must be designed with exceeding power, hence partially defeating the purpose of constant and efficient capacity usage.

When a failure occurs in the database tier, both Multi Data Center Deployment Active-Active and Multi Data Center Active-Passive present similar RTO and RPO since the database is the driver for recovery and in both cases it is active only in one site and passive in the other. The only advantage of Multi Data Center Active-Active Deployment systems is that an appropriate Data Source configuration can automate the failover of database connections from the middle tiers, reducing RTO (the recovery time is decreased because restart of the middle tiers is not required)¹.

¹ The Oracle WebLogic Servers may need to be restarted depending on different aspects. When using database leasing, Oracle WebLogic Servers may shut down if the database remains unavailable (during switchover or failover).

Performance

Besides the common performance paradigms that apply to single-datacenter designs, Oracle Fusion Middleware SOA Multi Data Center Active-Active systems need to minimize the traffic across sites to reduce the effect of latency on the system's throughput. In a typical Oracle Fusion Middleware SOA System, besides database access (for dehydration, metadata access, and other database read/write operations that custom services that participate in the system may perform), communication between the different tiers can occur mainly over the following protocols:

- » Incoming HTTP invocations from Load Balancers (LBR) or [Oracle HTTP Servers](#) (OHS) and HTTP callbacks
- » JNDI/RMI and JMS invocations between Oracle WebLogic Servers
- » Read/write requests to file systems for file/FTP adapters

For improved performance, all of the above should be restrained, as much as possible, to one single site. That is, servers in SiteN ideally should just receive invocations from Oracle HTTP Servers in SiteN. They should make JMS, RMI and JNDI invocations only to servers in SiteN and should get callbacks generated by servers only in SiteX. Additionally, servers should use storage devices that are local to their site to eliminate contention (latency for NFS writes across sites may cause severe performance degradation).

There are additional types of invocations that may take place between the different SOA servers that participate in the topology:

- » **Oracle Coherence notifications:** Oracle Coherence notifications need to reach all servers in the system to provide a consistent composite and metadata image to all SOA requests, whether served by one site or the other.
- » **HTTP session replications:** some Oracle Fusion Middleware SOA components use stateful web applications (such as Composer, Workspace, etc.) that may rely on session replication to enable transparent failover of sessions across servers. Depending on the usage patterns and number of users this may generate a considerable amount of replication data. Replication and failover requirements have to be analyzed for each business case, but ideally session replication traffic should be reduced across sites as much as possible.
- » **LDAP/policy/identity store access:** Access to policy and identity stores is performed by Oracle WebLogic Server infrastructure and SOA components for authorization and authentication purposes. In order to enable seamless access to users from either site, a common policy or identity store view needs to be used. Ideally each site should have an independent identity and policy store that is synchronized regularly to minimize invocations from one site to the other. Alternatively, both sites can share the same store. The impact of sharing the store will depend on the type of store and the usage pattern by the SOA system.

Administration

Another key aspect of the design and deployment of an Oracle Fusion Middleware SOA Multi Data Center Deployment is the **administration overhead** introduced by the solution. In order to keep a consistent reply to requests, the sites involved should use a configuration such that the functional behavior of the system is the same irrespective of which site is processing those requests. Oracle Fusion Middleware SOA keeps its configuration and metadata in the Oracle database. It is for this reason that Multi Data Center Active-Active Deployments with a unique active database guarantee consistent behavior at the composite and metadata level (there is a single source of truth for the involved artifacts). The Oracle WebLogic Server configuration, however, is kept 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 a Multi Data Center Active-Active Deployment system is very small as compared to any active-passive approach where constant replication of configuration changes is required.

Latency, Jitter, Packet Loss and Bandwidth Across Sites

The overall network throughput of an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment system is primarily driven by two factors: the length of the route that the requests have to take between the different sites (mainly for database access) and the interaction between the TCP reliability and congestion control protocols. Regardless of the speed of the processors where Oracle Fusion Middleware SOA runs or the efficiency of the software, it takes a finite amount of time to manipulate and “present” data from one site to the other. Two important measurements of time intervals in network transmission systems are referred to as **latency** and **jitter**. Network latency is the amount of time it takes for a packet to be transmitted end-to-end across a network, and it is composed of multiple variables (the type and number of switches between sites, the type of cabling, etc.) Latency in a network is measured either one-way (the time from the source sending a packet to the destination receiving it), or round-trip (the one-way latency from source to destination plus the one-way latency from the destination back to the source). Round-trip-time (RTT) latency is used more frequently because it provides a more realistic figure of the delay (accounts for traffic in both directions) and can be measured with the *ping* utility in most systems. Jitter is a term that refers to the variance in the arrival rate of packets from the same data flow. Both latency and jitter have a negative impact on applications with communications across sites. They are critical for the appropriate behavior of an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment. Jitter, however, is typically more relevant in systems with extremely low latency. Thus, latency is effectively the main aspect that must be controlled in a Multi Data Center Active-Active Deployment. The main causes of latency are:

- » propagation/distance delay
- » serialization
- » data protocols
- » routing and switching
- » queuing and buffering

Of all of the above causes, distance delay is typically the most relevant one. Distance delay is the minimum amount of time that it takes the electrical signals that represent bits to travel on a physical wire. Optical cable sends bits at about $\sim 5.5 \mu\text{s}/\text{km}$, copper cable sends it at $\sim 5.606 \mu\text{s}/\text{km}$, and satellite sends bits at $\sim 3.3 \mu\text{s}/\text{km}$. Distance delay can have a significant impact on the performance of an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment because multiple network round trips (mainly from the Oracle Fusion Middleware SOA servers to the SOA database) are required to complete each composite instance. Tests conducted have shown that an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment’s performance (where Oracle WebLogic Server SOA servers use a database in a different site) degrades considerably when latency exceeds 5-10 milliseconds. The graphs in Image 1 and Image 2 show the throughput (transactions per second) and average transaction active time for a Fusion Order Demo (FOD) Oracle Fusion Middleware SOA system that uses a database on a different site with different latencies between sites:

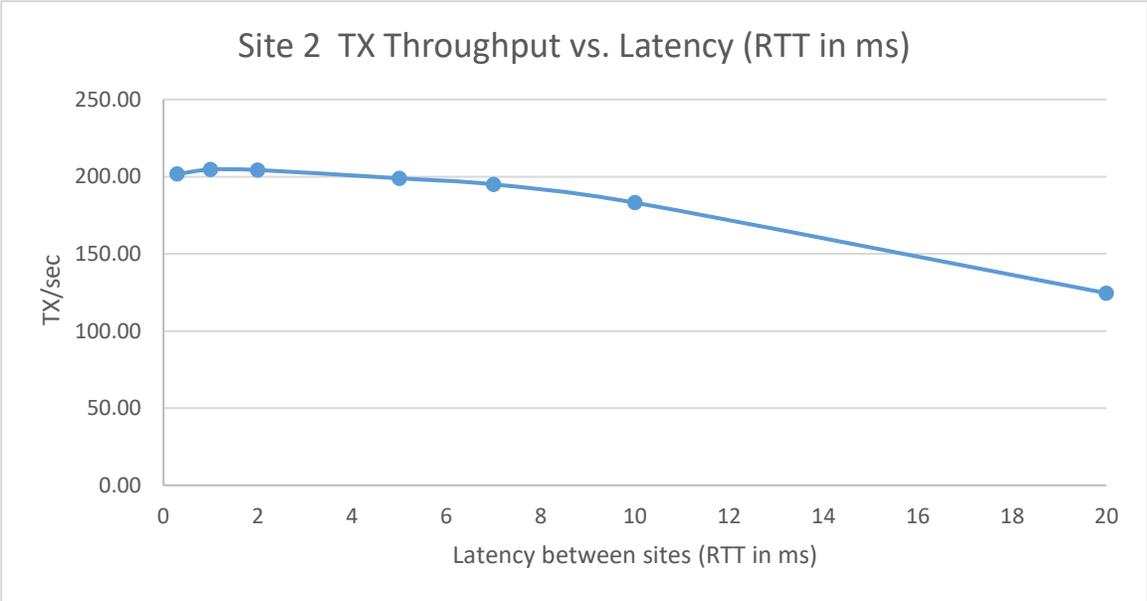


Image 1: Evolution of throughput with different latencies (RTT in msec.) between sites (FOD)

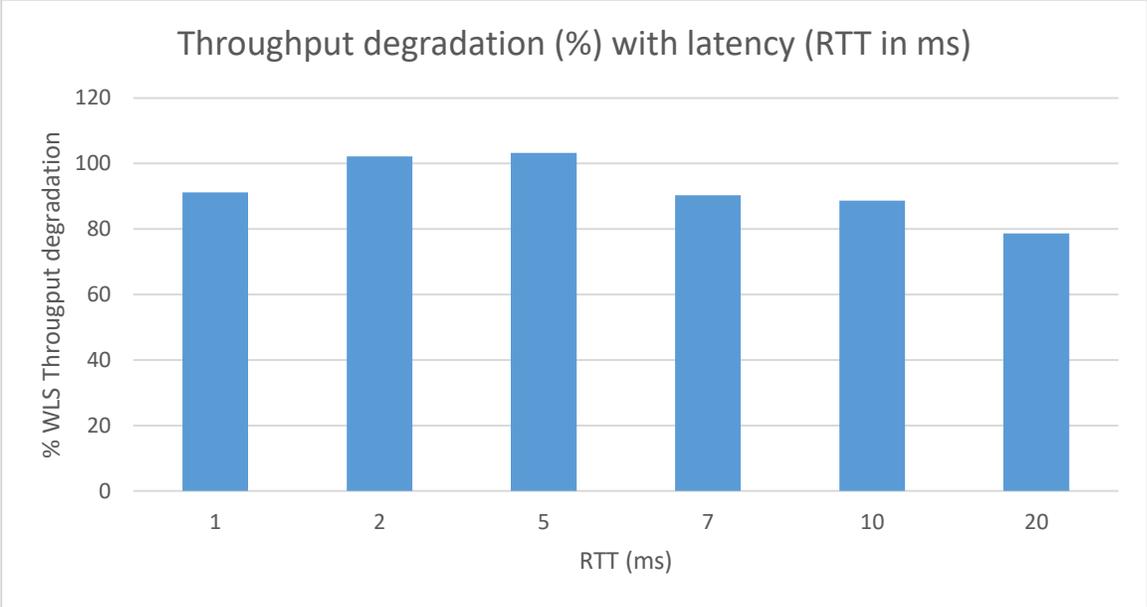


Image 2: Evolution of the time that a transaction remains active as the latency (RTT in msec.) between the SOA servers and the database is increased



Image 3 shows the degradation observed in the overall system's throughput (both sites working together) for different latencies. Observe that for a latency of around 20 milliseconds RTT the throughput decreases almost 25%.

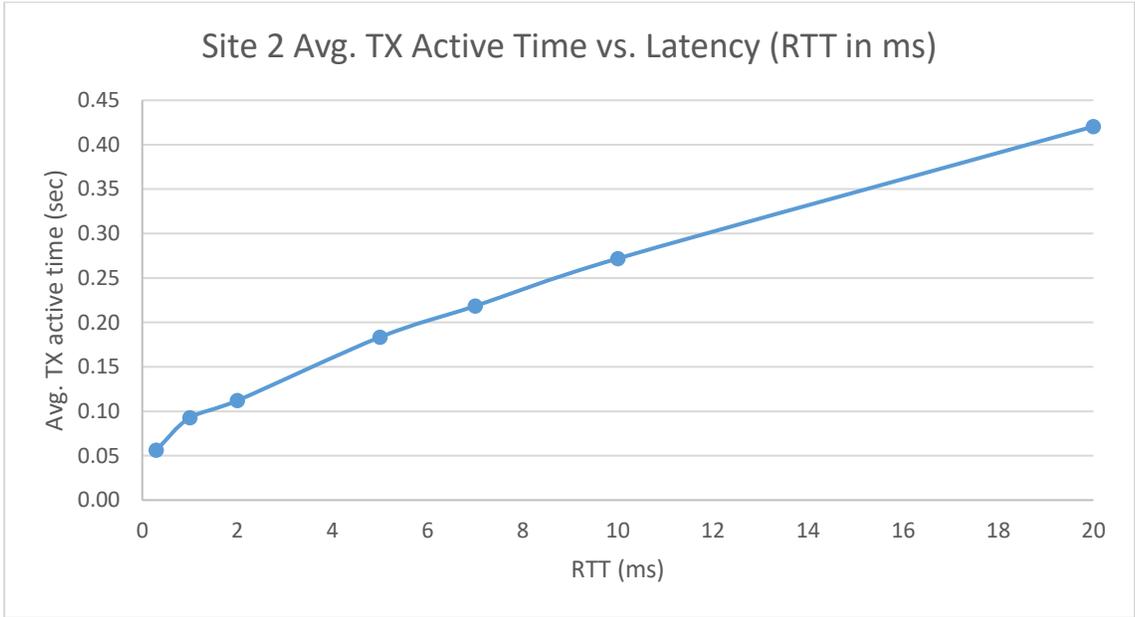


Image 3: Throughput degradation for different latencies (RTT in msec.)

Image 4 shows the degradation in total transactions/sec processed by the system (both sites working together) for different latencies. Observe that for a latency of 20 milliseconds RTT the TX/sec throughput decreases almost 25%.

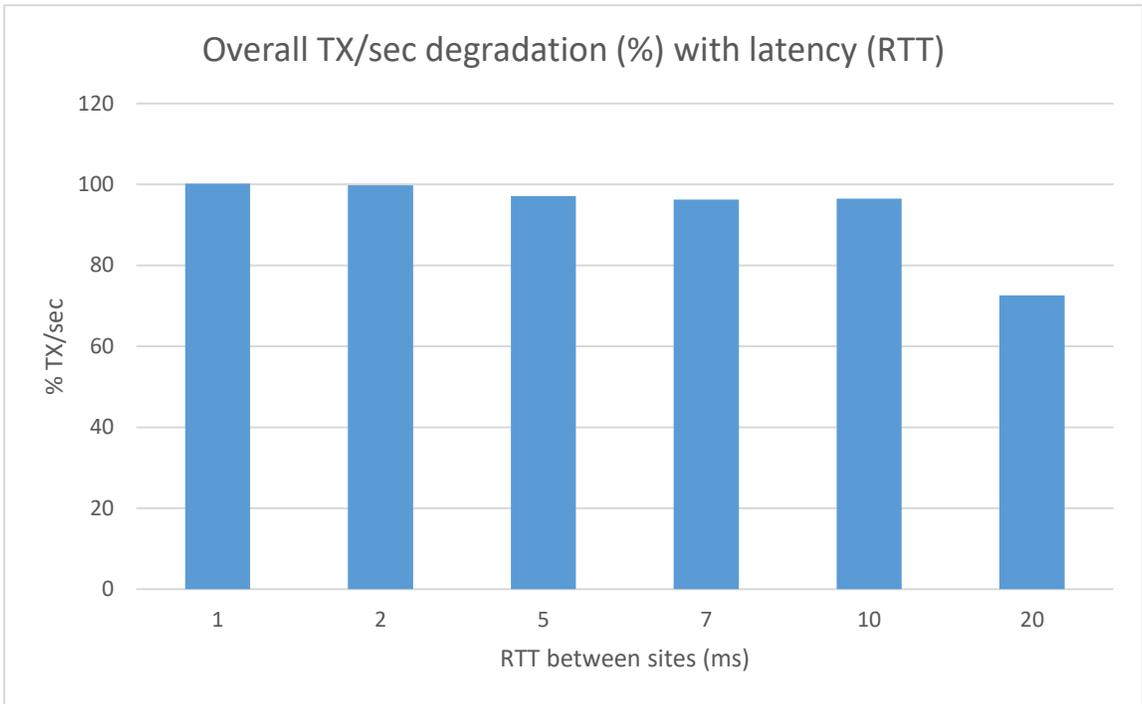


Image 4 Overall TX/sec degradation for different latencies (RTT in msec)

Image 5 shows the overall JMS/sec throughput degradation for the system (both sites working together) for different latencies between sites (RTT in milliseconds). For latencies of 20 ms (RTT) the JMS throughput decreases more than 25%.

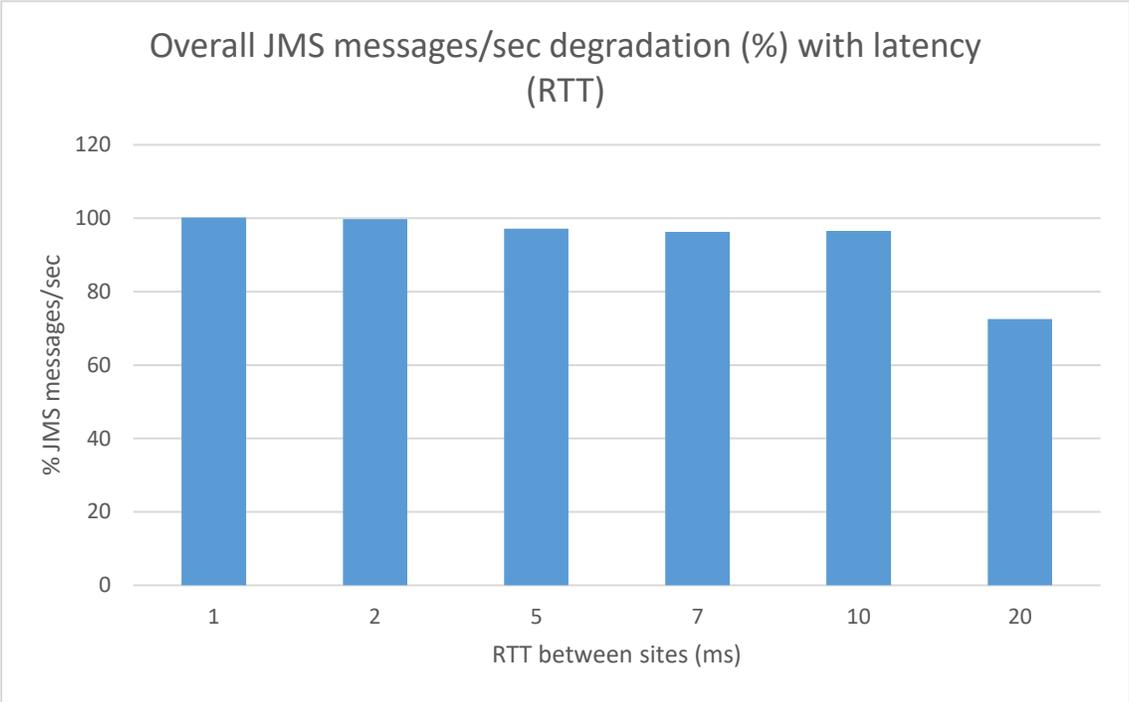


Image 5 Overall JMS messages/sec degradation for different latencies (RTT in ms) between sites

Image 6 shows the degradation in total transactions/sec processed by Site2 only (when both sites working together) for different latencies between sites, compared with the transactions/sec processed by Site1 servers during the same test. Observe that for a latency of 20 milliseconds (RTT) the transaction throughput in Site2 decreases about 35%.

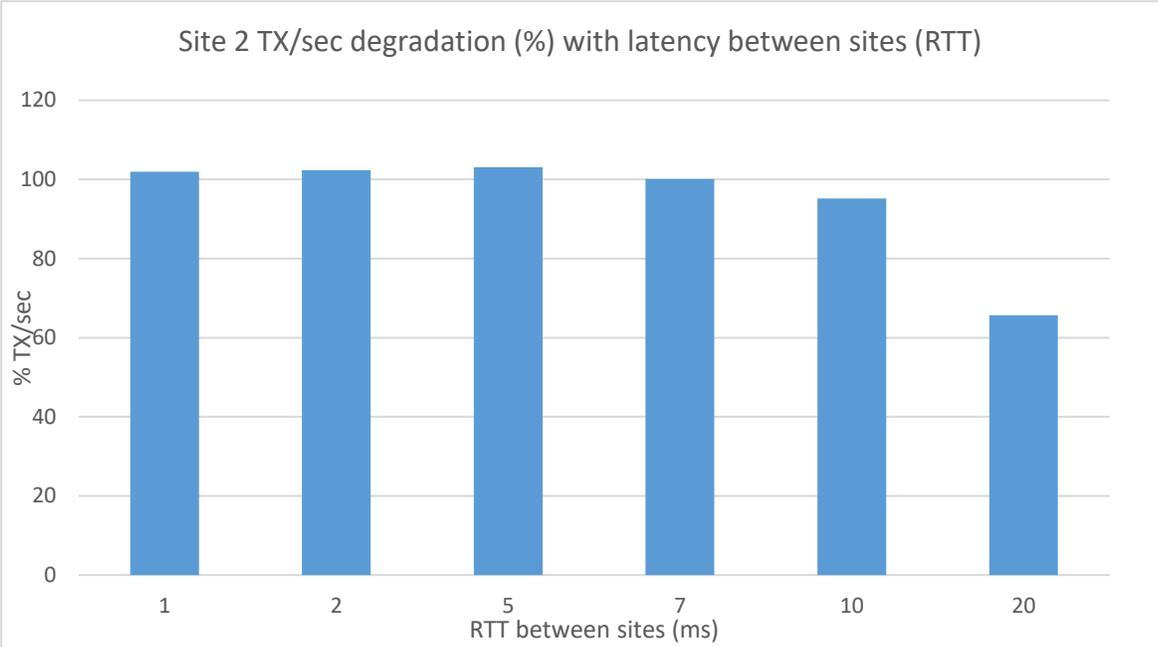


Image 6 Site2 TX/sec degradation compared with the TX/sec processed by Site1 servers during the same test, for different latencies between sites (RTT)

The following image shows the total number of active connections for a server in Site1 and a server in Site for the SOA datasource during an stress load, for different latencies between sites (RTT in milliseconds). Note that for a server in Site2, there are more active connections in the datasource because they are active during more time, and the number is increased with the latency between the servers and the database. This increment must be taken in account when tuning the size of the datasource, than may need to be adjusted for high latencies,

and has an effect also in recovery times for a site 2 server when a failure occurs in the database: more connections need to be recreated and more number of transactions need to be recovered.

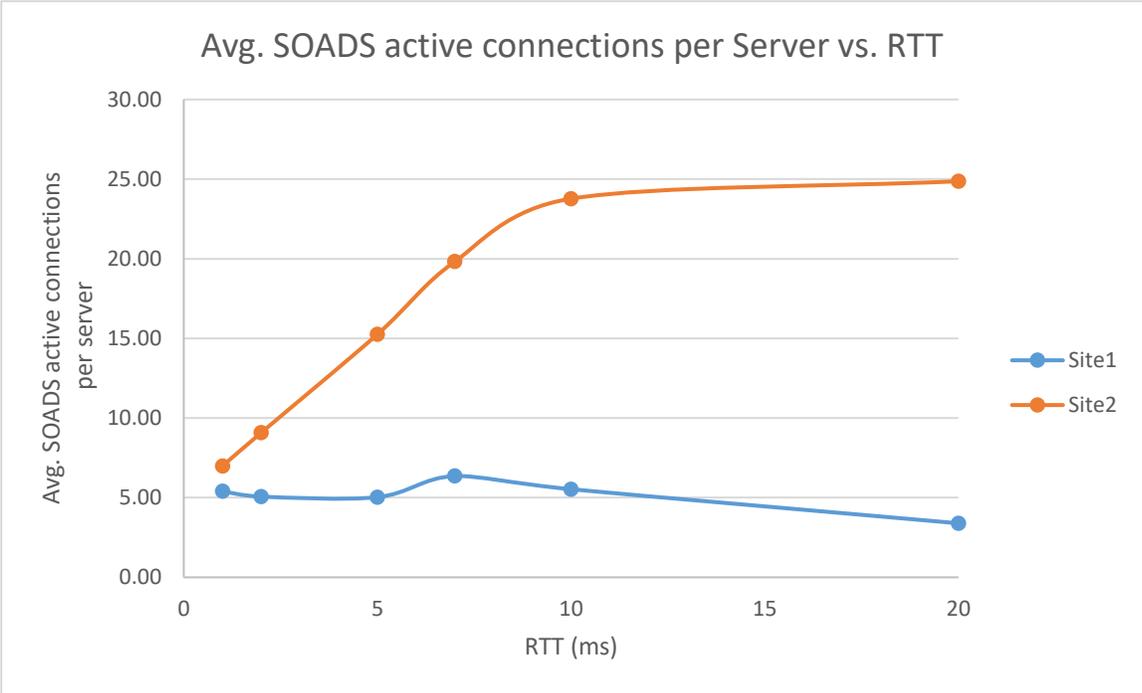


Image 7 Average SOA Data Source active connections during a load test for a server in Site1 and a for a server in Site2

Image 8 shows the increase in time taken to deploy a composite in a Site2 server compared to the the time taken to deploy in a Site1 server (SOA server and database in the same site). The time is higher in the servers of Site2 due to the higher latency to the database. When deploying a composite (first version or updates to newer versions) in the a Multi Data Center Deployment the composite may be deployed earlier in Site1 servers than in Site2 servers, although it is not activated until is is available in all members of the cluster. Refer to “Composite and MDS Deployments and Updates: Oracle Coherence Configuration” section for more details.

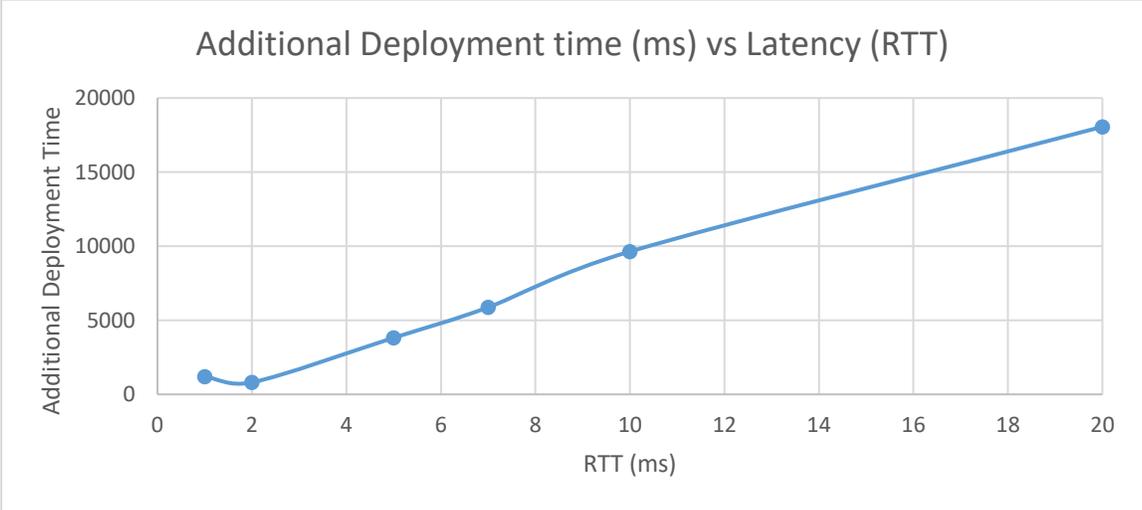


Image 8: Additional time (msecs) consumed for deploying composites with increasing latencies (RTT in msecs.) when the SOA database resides on a different site



When Oracle Data Guard is configured between the two sites, the effect of the latency between sites in the database performance depends on the Data Guard protection mode used. Oracle Data Guard can be configured in Maximum Availability, Maximum Performance, or Maximum Protection mode. When it is configured for Maximum Performance (default), redo data is written to the standby database asynchronously with respect to transaction commitment, so primary database performance is unaffected by the time required to transmit redo data and receive the acknowledge from a standby database. This protection mode offers slightly less data protection than maximum availability mode and has minimal impact on primary database performance.

When Data Guard is configured for Maximum Availability or Maximum Protection, the transactions do not commit until all redo data needed to recover those transactions has been received by the standby database. This protection mode provide higher level of data protection but the effect in the performance is higher.

Note that the effect of latency is orthogonal to the bandwidth between two sites (that is, it will affect equally large or small messages and payloads). For example: if a SOA server executes a SQL database query that requests 100 rows of the CUBE_INSTANCE, MEDIATOR_INSTANCE and DLV_MESSAGES tables, one row at a time, over a link with a latency of 60 ms, it takes approximately 6 seconds ($60 \text{ ms} * 100 \text{ turns}$) to complete the transaction *independently of the amount of data in each row*. The same query executed by a user on a LAN connected to the same database server takes less than 2-3 ms to be completed, as the latency due to distance across the LAN is insignificant. This is irrespective of the size of each row. Bigger rows can be retrieved with better bandwidth, but the overall transaction takes the same amount of time.

With all of the above in mind and provided the performance penalties observed in many tests, Oracle recommends **not to exceed 10 msec of latency (RTT)** for SOA Multi Data Center Active-Active systems when the latency affects database communications. Systems may operate without issues, but the transaction times will increase considerably. Latencies beyond 10 msec (RTT) will also cause problems in the Coherence cluster used for deployment and JTA and web services timeouts for most common composites. This makes the solutions presented in this paper suitable primarily for Metropolitan Area Networks with low latency between sites (for example, the distance from San Francisco to Boston is around 4330 kms and typical latencies are approximately 30-40 msec).

Requirements

Topology

The analysis and recommendations included in this paper are based on the topology described in the “Topology Model for an Oracle Fusion Middleware SOA Active-Active Multi Data Center Deployment” section. Each site locally uses an Oracle Fusion Middleware SOA Enterprise Deployment Topology (separation of WSM-PM servers, shared storage and directory structure, etc.). The system requirements are those specified in the Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite.

Additionally, the following requisites must be met:

Network

Latency between the two sites used in the design should not be higher than 10 msec RTT. The bandwidth requirements will vary based on the type of payloads used by each SOA system.

Shared Storage vs. Database for Transaction Logs and Persistent stores

The topology addressed in this paper was tested using database-based persistent stores for Oracle WebLogic Server transactions logs and Oracle WebLogic Server JMS persistent stores. Storing transaction logs and persistent stores in the database provides the replication and high availability benefits inherent from the underlying database system. With JMS, TLOG, and SOA data 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 (they still apply for the Administration Server’s failover, deployment plans, and some adapters like File Adapter). 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. This penalty applies also to JMS Servers that use AQ destinations. From a performance perspective, a 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.

Additionally, due to the criticality of persistent stores in the overall state of WLS servers, it is recommended to use Test Connections on Reserve for the pertaining Data Sources and also, configure in-place restart for the pertaining JMS Server and Persistent stores. Refer to Appendix B for details on configuring in-place restart.

Load Balancers

Load balancers from any vendor are supported as long as the load balancer meets the requirements listed in [Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite](#) section 2.2.3. The global load balancer should allow rules based on the originating server’s IPs (an example is provided for F5 Networks).

Oracle Fusion Middleware SOA Components and Versions in Scope

This document is based on Oracle Fusion Middleware SOA 12.2.1 PS3 (12.2.1.3). Any later release should also work in similar configurations. The Oracle Fusion Middleware SOA components verified are:

- 
- » Oracle BPEL
 - » Oracle Mediator
 - » Oracle Rules
 - » Oracle EDN
 - » Oracle Technology Adapters File, Database, and JMS Adapter
 - » Oracle Service Bus

This document provides the configuration details and results for static (configured) clusters. Dynamic clusters are out of the scope of this document.

Hardware Resources and Capacity Utilization

A Multi Data Center Active-Active Deployment is usually designed to make effective use of resources available in multiple sites. However, the appropriate capacity planning needs to be done to account for failover scenarios between the two sites. If an entire site loses the middle tiers, the other must be designed to sustain the added load, or the appropriate request throttling and rejection mechanisms must be enabled (typically in the GLBR). Otherwise, cascade failures (where the failover causes such an overhead on the available site that it is rendered unresponsive) may occur. This implies that during normal operation the middle tier nodes must remain underutilized to an extent that will vary depending on the capacity that needs to be available in failover situations.

Topology Model for an Oracle Fusion Middleware SOA Active-Active Multi Data Center Deployment

Image 9 depicts the main pieces (without details on specific routing or Oracle WebLogic Server domain aspects) of the Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment addressed in this paper.

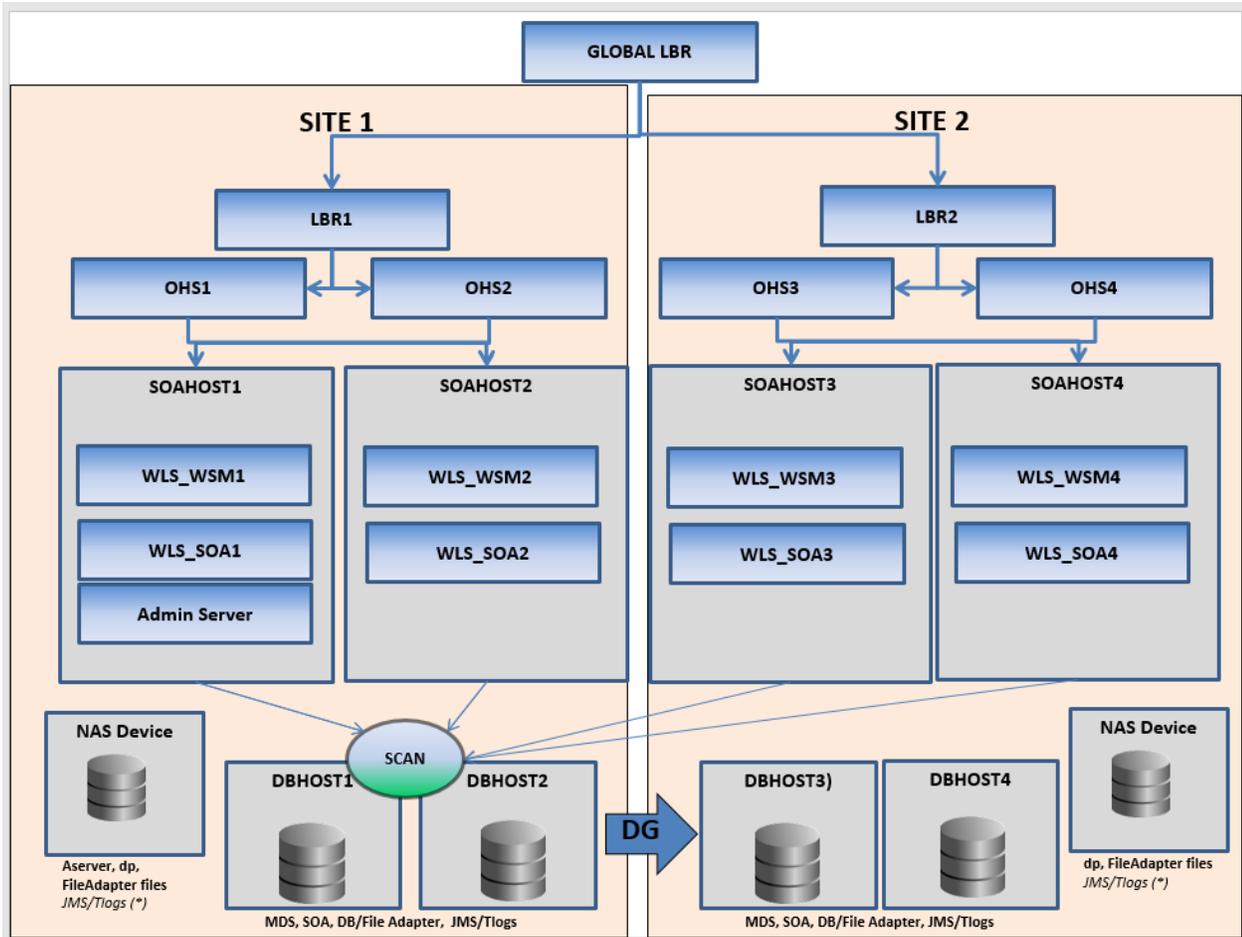


Image 9: Components in an Oracle Fusion Middleware SOA Active-Active Multi Datacenter Deployment

In Image 9, there are two separate sites (Site1 and Site2 for future reference in this document) that are accessed by one unique access point: a global load balancer which directs traffic to either site (each vendor provides different routing algorithms). Each site has its own local access point – a local load balancer. The local load balancer distributes requests to multiple Oracle HTTP Servers (OHS). Finally, the local HTTP servers allocate requests to specific Oracle WebLogic Servers hosting Oracle Fusion Middleware SOA components (service engines, adapters, and infrastructure). The two environments share one unique database that is accessed CONCURRENTLY by servers in both sites. The following sections provide details for each tier.

Database Tier



The synchronicity requirements and data types used by the different Oracle Fusion Middleware SOA Suite components limit the possible approaches for the Oracle Fusion Middleware SOA database in a Multi Data Center Active-Active deployment. This document addresses only a solution where the Oracle Fusion Middleware SOA database uses Data Guard to synchronize an active database in Site1 with a passive database in Site2. Although other Active-Active approaches may work they have not been tested and certified by Oracle and are out of the scope of this document. In this configuration we assume that both sites where Oracle Fusion Middleware SOA is deployed access the same database (as well as the same schemas within that database), and the database is set up in a Data Guard configuration. Data Guard provides a comprehensive data protection solution for the database. It consists of a standby site at a geographically different location than the production site. The standby database is normally in passive mode; it is started when the production site (called “production” from the database activity point of view) is not available². The Oracle Database is configured in each site in an Oracle Real Application Cluster (RAC). Oracle RAC enables an Oracle database to run across a cluster of servers in the same data center, providing fault tolerance, performance, and scalability with no application changes necessary. Oracle Fusion Middleware Tier

Load Balancers and Web Servers

The Global Load Balancer (GLBR) is a load balancer configured to be accessible as an address by users of all of the sites and external locations. The device provides a virtual server which is mapped to a DNS name that is accessible to any client regardless of the site they will be connecting to. The GLBR directs traffic to either site based on configured criteria and rules. These criteria can be based on the client’s IP for example. This should be used to create a Persistence Profile which allows the LBR to map users to the same site upon initial and subsequent requests. The GLBR maintains a pool which consists of the addresses of all the local load balancers. In the event of failure of one of the sites, users are automatically redirected to the surviving active site.

At each site, a Local Load Balancer receives the request from the GLBR and directs requests to the appropriate HTTP server. In either case, the Local Load Balancer is configured with a persistence method such as Active Insert of a cookie in order to maintain affinity and ensure that clients are directed appropriately. To eliminate undesired routings and costly re-hydrations, the GLBR is also configured with specific rules that route callbacks only to the LBR that is local to the servers that generated them. This is useful also for internal consumers of SOA services. These GLBR rules can be summarized as follows:

- » If requests come from Site1 (callbacks from the SOA servers in Site1 or endpoint invocations from consumers in Site1) the GLBR routes to the LBR in Site1.
- » If requests come from Site2 (callbacks from the SOA servers in Site2 or endpoint invocations from consumers in Site2) the GLBR routes to the LBR in Site2.
- » If requests come from any other address (client invocations), the GLBR load balances the connections to both LBRs.
- » Additional routing rules may be defined in the GLBR to route specific clients to specific sites (for example, the two sites may provide difference response time based on the hardware resources in each case).

² The Oracle Active Data Guard Option available with Oracle Database 12c Enterprise Edition enables you to open a physical standby database for read-only access for reporting, for simple or complex queries, or sorting while Redo Apply continues to apply changes from the production database. Oracle Fusion Middleware SOA does not support Oracle Active Data Guard because the SOA components execute and update information regarding SOA composite instances in the database as soon as they are started.



Application Layer

Each site runs from an Oracle Fusion Middleware SOA installation that is “local” to that site (that is, in a file system located nearby the servers). Each local topology uses an Oracle Fusion Middleware SOA Enterprise Deployment Topology for maximum availability and security. Other topologies based on the required high availability principles are allowed. The Oracle WebLogic Server Domain model used in this paper uses one single domain and one single cluster for Oracle Fusion Middleware SOA Suite components. This model is also known as a Stretched Cluster. In this topology, all servers (WSM-PM and SOA) are part of a unique Oracle WebLogic Server Domain. They are managed with a single Administration Server that resides in one of the two sites. Each site uses the same database for persistent stores, and the cross-site synchronization is based on Data Guard. For contention and security reasons it is not recommended to use shared storage across sites. Disk mirroring and replication from Site1 to Site2 and vice versa would be used to provide a recoverable copy of the artifacts in each site. A unique Oracle WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control provide a central administration point for all the servers. The SOA servers in both sites are part of a unique cluster (SOA_Cluster) and so are the WSM-PM ones (WSM_Cluster). The Coherence cluster used for composite deployments and MDS updates is also the same one for the two sites. A single RAC database is used for SOA and all Oracle WebLogic Servers point to the same SOA and MDS schemas. Image 10 describes the topology.

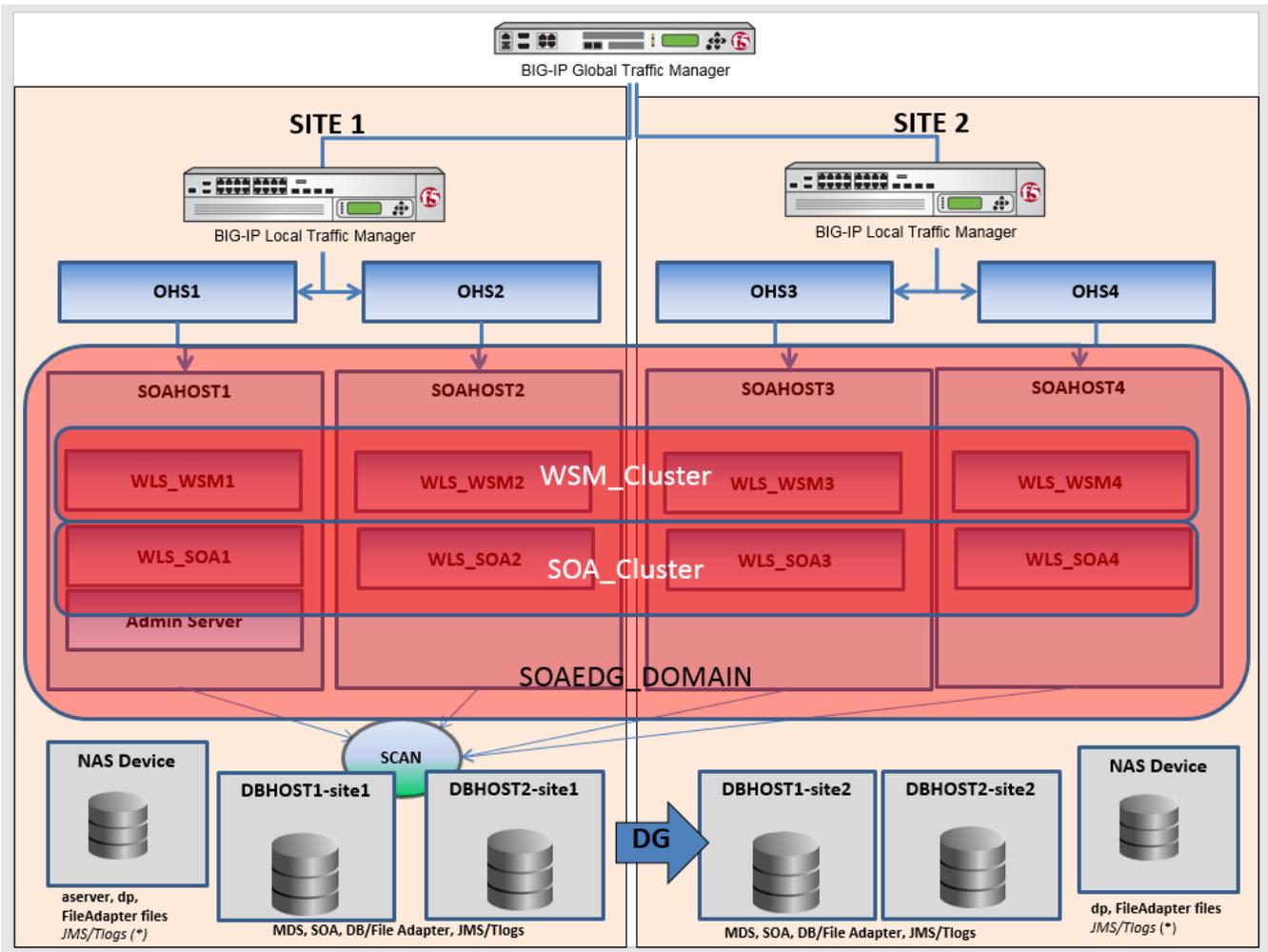


Image 10: Stretched cluster model for Multi Data Center Active-Active Deployment.

Characteristics of the Design

Availability: The stretched cluster design uses an OHS configuration based on a fixed list of servers in each site (instead of the “dynamic” list, provided by the OHS plug-in and used in typical single-location deployments). This is done to eliminate undesired routing from one site to another. This has the disadvantage of slower reaction times to failures in the Oracle WebLogic Servers. The database connection failover behavior and the JMS and RMI failover behaviors are similar to those that take place in a standard Enterprise Deployment Topology. Specifically for BPEL, automatic recovery can be achieved from either site irrespective of the node that originated the instance. There is, at all times, one single CLUSTER_MASTER server, that is, just one server among all the available servers in the Multi Data Center Active-Active Deployment is able to perform automatic recovery. Instances can be recovered equally from Site1 and Site2 should a failure occur on the partner site.

1. From Site1 when Site2 is up if the CLUSTER MASTER resides in Site1
2. From Site2 when Site1 is up if the CLUSTER MASTER resides in Site2
3. From Site1 when Site2 is down
4. From Site2 when Site1 is down

Should a failure occur in Site1 that affects all of the middle tiers, recovery of the Administration Server is required to resume the Oracle Enterprise Manager Fusion Middleware Control and the Oracle WebLogic Server Administration Console. The Administration Server listens in a Virtual IP, and it is unlikely (depending on the customer’s infrastructure) that the Virtual IP used in one site is valid in the other. It usually requires additional intervention to enable the listen address initially available in Site1 in Site2 and vice versa.

Those servers that are remote to the Administration Server take longer to restart than in a regular Enterprise Deployment Topology. 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.

From the RPO perspective, transactions that were halted by a site failure can be resumed in the site that remains available using Automatic Service Migration without manual intervention. Automated 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 needs to be set up between the sites.

Service Migration is recommended and used in this topology design versus Server Migration, that requires Virtual IPs so it may need additional intervention to enable a listen address initially available in Site1 in Site2 and vice versa. This intervention can be automated in pre-migration scripts, but in general the RTO will increase compared to a standard automated server migration (taking place in the scope of single data center) or to Automatic Service Migration, for which VIPs are not required.

Administration: In a Multi Data Center Active-Active Deployment, the Oracle WebLogic Server infrastructure is responsible for copying configuration changes to all the different domain directories used in the domain. The Coherence cluster configured for SOA is in charge of updating all of the servers in the cluster when composites or metadata are updated³. Except for the replication requirement for runtime artifacts across file systems (file adapter,

³ See the sections related to Composite Deployment and MDS Updates for details on the possible effects of latency in the system from the administration perspective.

TLOGS, etc.), a Multi Data Center Active-Active Deployment is administrated like a standard cluster. This makes its administration overhead very low.

Performance: If the appropriate load balancing and traffic restrictions are configured (see following sections) the performance of a stretched cluster with low latency across sites should be similar to that of a cluster with the same number of servers residing in one single site. The configuration steps provided in the following sections are intended to constrain the traffic inside each site for the most common and normal operations. This isolation, however, is non-deterministic (for example, there is room for failover scenarios where a JMS invocation could take place across the two sites). That said, most of the traffic takes place between the Oracle Fusion Middleware SOA Servers and the SOA database. This will be the key to the performance of the Multi Data Center Active-Active Deployment. Image 11 shows the percentage of traffic between a SOA server in Site2 and the different addresses in Site1 during a stress test. Notice that more than 90% of the traffic happens between the servers and the database (also located in Site1).

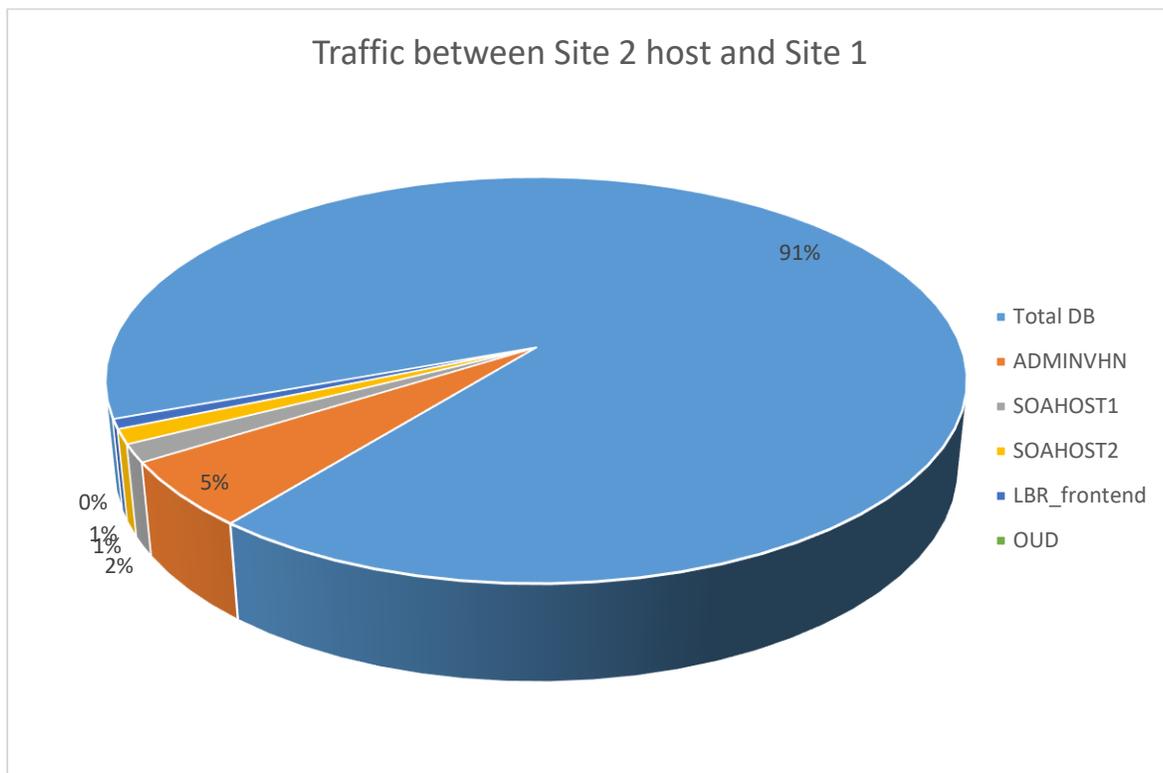


Image 11: Stretched Cluster Model. Traffic percentages between a server in Site2 and the rest of the hosts in Site1

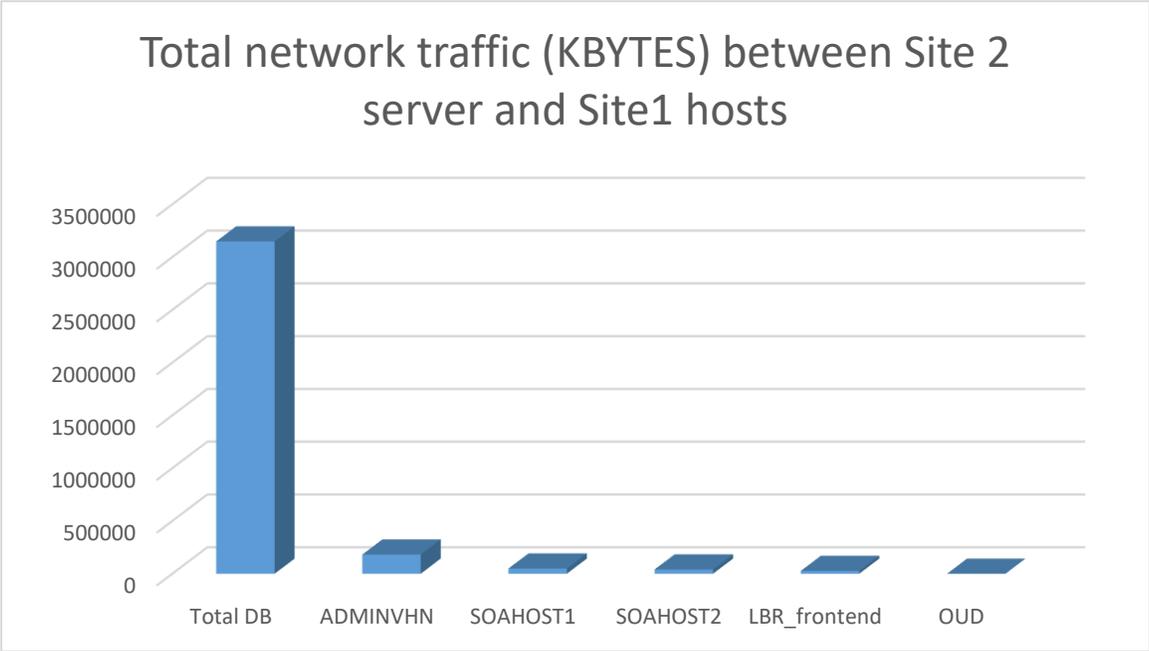


Image 12 Total traffic between a server in Site2 and the rest of the hosts in Site1 during a 15 min duration stress test

Other Resources

The two sites may or may not share other external resources. These resources include LDAP, identity stores, policy stores, external JMS destinations, external web services, etc. The configuration details for these external resources are out of the scope of this document. It is required, however, that these resources are consistent in both sites. Notice that asynchronous callbacks may re-hydrate instances that were initiated in a different site. For these to provide a consistent behavior, the same external resources must be available in both sites (this is also required for automatic recovery purposes: any Oracle WebLogic Server can become a cluster master and perform recovery in either site).

Configuring the Oracle Fusion Middleware SOA Active-Active Topology

The following sections provide the steps for configuring an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment. Basic understanding of the common Oracle WebLogic Server administration tasks as well as familiarity with the procedures and configuration steps included in the [Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite](#) is assumed. The steps are very similar to those described in the guide, but specific configuration changes are applied in different sections of the EDG to minimize traffic across sites. In summary the steps are:

1. Configure GLBRs and LBRs as per the *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite* but with the appropriate rules for local routing.
2. Configure OHS as per the *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite* but with routing restricted to each site.
3. Configure the application tier with special steps for the following:
 - Shared storage/directory configuration
 - Service migration configuration
 - JMS configuration
 - JMS Adapter and File Adapter configuration
 - Data Source configuration
 - Depending on whether the latency between sites is approaching the 10 msec. limit, adjust Oracle Coherence settings, Oracle Net settings, and JTA/Timeout settings.

The sections that follow detail each of these aspects.

Configuring Load Balancers and Global Load Balancers for Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment

As indicated in previous sections, the Global Load Balancer (GLBR) is responsible for performing smart routing of requests between multiple Local Load Balancers. This smart routing is usually done based on the originating request. In an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment it is recommended that you restrain callbacks and invocations that come from servers in a specific site to the same site again. Because the GLBR is typically located in one of the two sites (physically) this also makes the invocations to such a site more efficient. The following procedures provide an example of configuration for F5's products.

Configuring the Local Load Balancer

The Local Load Balancers (LBR) receive requests from the Global Load Balancer and send requests to the Oracle HTTP Servers. Each LBR should be configured as indicated in the [Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite](#). Because all of the components addressed in this document depend on the availability of the Oracle Fusion Middleware SOA Service Infrastructure application, it is recommended that you set the LBRs to monitor the `/soa-infra/` URL⁴ to determine the availability of a SOA server. This will eliminate undesired routings when the SOA Oracle WebLogic Server is RUNNING but the SOA subsystem is not really available (these routings can occur when the monitor is set on the root context `/` URL).

Configuring the Global Load Balancer

The following procedure is specific to F5 BIG-IP Global Traffic Manager (GTM) and LBR. The procedure is provided as an example of the configuration required. Refer to the [F5 knowledge base](#) or to your GTM's specific documentation for details⁵.

1. It is assumed that the appropriate listener already exists in the GTM
2. In the Global Traffic Menu of the F5 Administration Console, create two Data Centers, one for each site participating in the Multi Data Center Deployment configuration (a data center defines the servers and links that share the same subnet on the network). The defaults are appropriate.
3. In the Global Traffic Menu of the F5 Administration Console create a server for each site (assuming one LBR per site) and assign it to the appropriate site (a server defines a specific physical system on the network) as follows:
 - a. Use the address of the first site's Local LBR for this server.
 - b. For Product, select BIG-IP System (single).
 - c. Use the appropriate health monitor for the Server (this may be a TCP monitor or a combination of multiple monitors, depending on the services the Local LBR is running).
 - d. Add as virtual server the address on which the local LBR is listening for SOA requests. For this virtual server, if the latency across sites is high, you may want to use a different monitor depending on the site (a more permissive probe may be needed for high latencies).
4. In the Global Traffic Menu of the F5 Administration Console create a new pool (for future reference we will call it the MDCpool). A pool represents one or more virtual servers that share a common role on the network. A virtual server, in the context of GTM, is a combination of IP address and port number that points to a specific resource on the network.

Use the appropriate health monitor for the server (HTTP or HTTPS exists with the device's factory configuration) according to your system's protocol in the Local LBR (this typically would be HTTP). Assign as members the virtual servers created in the previous steps. This monitor should be the most permissive one of the two monitors used for the sites.

5. In the Global Traffic Menu of the F5 Administration Console create a new Wide IP. A Wide IP maps a fully-qualified domain name (FQDN) to a set of virtual servers that host the domains content as follows:
 - a. Use the FQDN that will be used to access the SOA Multi Data Center Deployment system.
 - b. Add the pool previously created to the Wide IP.
 - c. Enable persistence for the Pool.
 - d. Use Round Robin as the load balancing method.

⁴ Include the backslash at the end of the URL, or the monitor will fail due to the Oracle WebLogic Server redirecting to the front end address.

⁵ The redundancy and DNS server configuration required for providing redundancy for GTM servers is out of the scope of this paper.

With these settings the F5 GTM should round robin request to both sites or datacenters. To do the smart routing required for callbacks, internal web service invocations, etc., define two more pools:

- » A pool containing ONLY the LBR in Site1 (Site1pool)
- » A pool containing ONLY the LBR in Site2 (Site2pool)

Add the following iRule for Global Traffic and assign it to the Wide IP for the system.

```
when DNS_REQUEST {
  if { [IP::addr [IP::client_addr] equals 10.10.10.10/24 ] } {
    pool Site1pool
  } elseif { [IP::addr [IP::client_addr] equals 20.20.20.10/24] } {
    pool Site2pool
  } else {
    pool MDCPool
  }
}
```

Use the appropriate IP address ranges and definitions that apply to each datacenter or site. With this, the system is enabled to redirect requests to each Local LBR based on the originating request's IP. For additional details refer to the F5 GTM documentation at http://support.f5.com/kb/en-us/products/big-ip_gtm/manuals/product/gtm-concepts-11-2-0.html.

Configuring Oracle HTTP Server for Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment

In a stretched cluster model, and to reduce traffic across sites, the Oracle HTTP Servers (OHS) do not use dynamic cluster notifications, instead they get configured with a static list of servers. This has the caveat of slower failure detection (when a SOA server crashes the HTTP server takes more time to detect the failure than with a dynamic list) and requires updates in the configuration if new servers are added. However, it improves the system's performance. The following excerpts from the mod_wl_ohs.conf files in the OHS provide an example of the required configuration for routing to the soa-infra web application.

Site1:

```
# SOA soa-infra app
<Location /soa-infra>
  WLSRequest ON
  WebLogicCluster
  Site2_server1.mycompany.com:8001,Site2_server2.mycompany.com:8001
  DynamicServerList OFF
  ..
</Location>
```

Site2:

```
# SOA soa-infra app
<Location /soa-infra>
  WLSRequest ON
  WebLogicCluster
  Site2_server1.mycompany.com:8001,Site2_server2.mycompany.com:8001
  DynamicServerList OFF
  ..
```

</Location>

Configuring the Application Tier of an Oracle Fusion Middleware SOA AA DR System for a Stretched Cluster

The stretched cluster design is an Enterprise Deployment Topology scaled out to two additional servers in Site2. There are some aspects to consider that can make the system more scalable and that will minimize the possible performance degradation caused by the latency across sites.

The first site is installed as described in the [Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite](#). Variations in terms of installing the Oracle WSM-PM Cluster collocated with the SOA Cluster and other high availability topologies are allowed. The second site (Site2) is configured using the steps provided in [Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite](#) section 24. The following aspects need to be considered:

- » **Binaries/Installations:** The second site uses its own redundant binaries (that is, at least two binary installations should be used per site for high availability).
- » **Paths:** The binary installations, `aserver` domain directory, deployment plans and file adapter directories in Site2 should use the same path as Site1.

Shared Storage: The binary installations, `aserver` domain directory, deployment plans, and file adapter directories reside on shared storage as indicated in the [Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite](#), but each site uses a shared storage local to the site. That is, designs where servers in Site2 need to access shared storage on Site1 and vice versa **should be avoided**. This means that in the Stretched Cluster a “manual” separation of artifacts (file adapter, deployment plans) is done. In this scenario if File persistent stores are used, automatic service migration across sites is not possible because transaction and JMS message recovery is precluded (as opposed to a typical scale out scenario in the context of one single site, where service migration is configured using all machines as candidate). Hence, it is recommended to use JDBC persistent stores. In this scenario, Automatic Service Migration is possible between sites, as long as the JMS and TLOGs data is available both from sites and it is replicated using Data Guard from Site1 to Site2.

- » **Oracle HTTP Server Configuration:** Observe the details in Configuring Oracle HTTP Server for Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment when configuring OHS routing.
- » **SOA Cluster Configuration:** Specify the GLBRs virtual server's address created in the Configuring Load Balancers and Global Load Balancers for Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment sections as callback URL (that is, as front end address for the SOA cluster) that will front end the system.

The Cluster Address is not required to be set with the explicit list of all the servers in the cluster. When it is empty, the Cluster Address value is generated automatically. In this case, ensure that the property “Number of Servers In Cluster Address”, that limits the number of servers to be listed when generating the cluster address automatically, has a value high enough to include all the servers in the cluster .

- » **Automatic Service Migration configuration:** Oracle recommends configuring Automatic Service Migration along with JDBC persistent stores for enterprise deployment topologies. In this scenario, Automatic Service Migration is possible between sites, as long as the JMS and TLOGs data is available both from sites and it is replicated using Data Guard from Site1 to Site2. There are no special considerations for stretched cluster automatic service migration.

However, the time taken for the service migration from Site1 to Site2 can be increased when there are high latencies between sites. This increase is not caused by the time it takes to detect a failure, but rather for the time spent in recovering the messages in the other server, because they are read from the persistent store in the database. The following picture shows the time spent during a service migration from a server in Site1 to

a server in Site2, with a very low of messages in the persistent stores. Note that only for latencies of 20ms (RTT) the time spent in service migration is really increased.

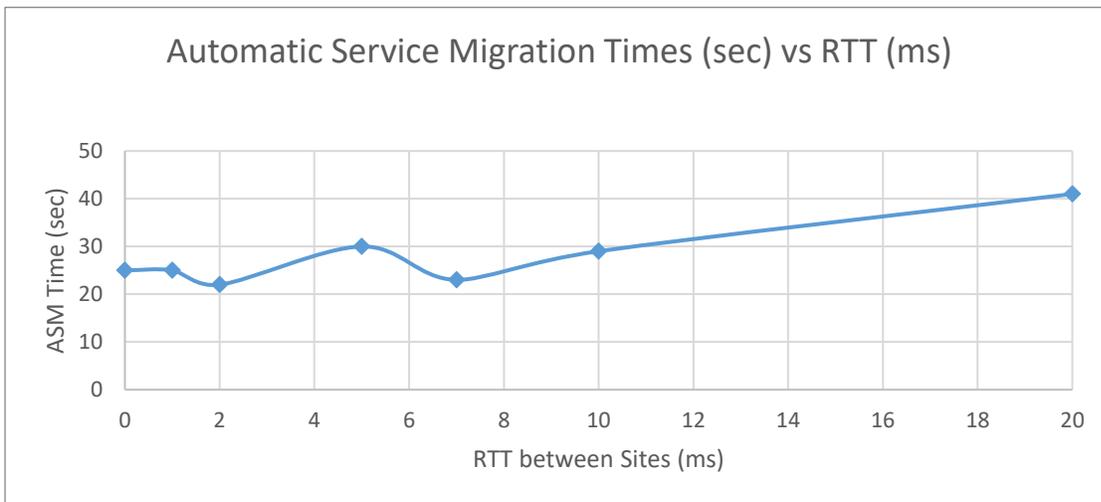


Image 13 Automatic Service Migration times when migration JMS and TLOG services from a server in Site1 to a server in Site2

This increment is higher if the persistent stores have more pending messages. The following image is an example of the Automatic Service Migration time increment when one of the persistent stores has a high number of pending messages (around 8000).

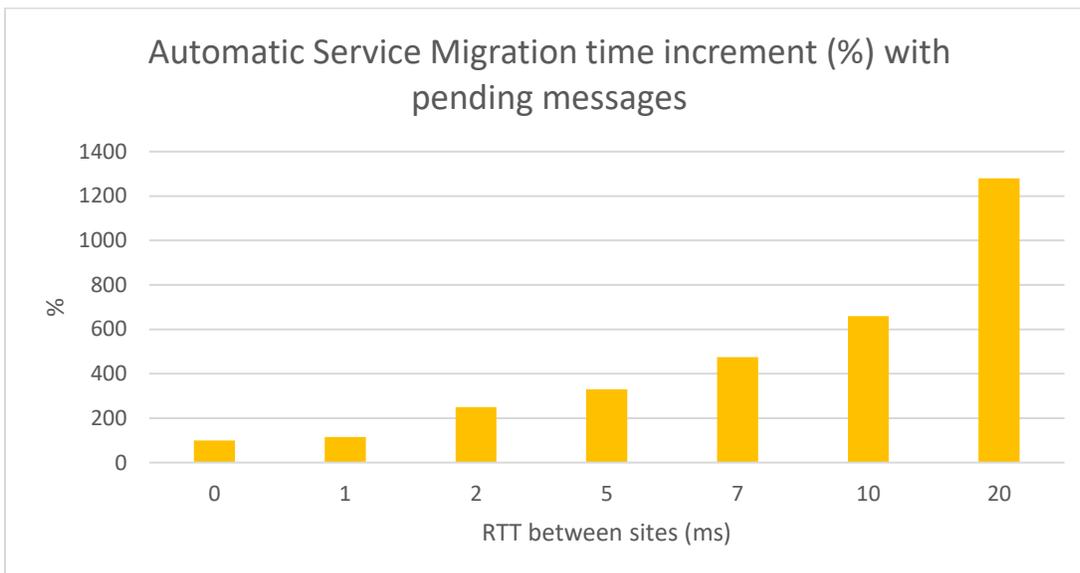


Image 14 Automatic Service Migration time degradation with latency between Sites when migrating services from Site1 to Site2 server with high number of pending messages in a persistent store.

Only for cases where files stores are used, the stretched domain design servers use only those servers in their same site as candidates for migration. Use the steps in the Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite for configuring service migration with these additional

considerations (note that this is not needed when using JDBC stores, where cross-site service migration is possible):

- For each migratable target on Site1, chose only servers in Site1 as candidates.

The screenshot shows the 'Settings for WLS_SOA1' configuration page. The 'Migration' tab is selected. Under 'JMS Service Migration Configuration', the 'Available' list contains WLS_SOA3 and WLS_SOA4, and the 'Chosen' list contains WLS_SOA1 and WLS_SOA2. Under 'JTA Migration Configuration', the 'JTA Migration Policy' is set to 'Failure Recovery'. The 'Available' list contains WLS_SOA3 and WLS_SOA4, and the 'Chosen' list contains WLS_SOA1 and WLS_SOA2.

- For each migratable target on Site2, chose only servers in Site2 as candidates.

Depending on the latency across sites, you may need to increase the Health Check Interval for service migration. The default is 10000 msecs which should be adequate in most cases; however, busy periods and overloads may require using a higher value depending on each case. Notice that this setting affects the health checks for all of the servers in the Stretched Cluster; hence it will increase the time it takes to detect crashes of all of the servers. To increase the Health Check Interval use the Health Monitoring tab for the cluster as shown here:

The screenshot shows the 'Settings for SOA_Cluster' configuration page. The 'Health Monitoring' tab is selected. The configuration fields are: 'Inter-Cluster Comm Link Health Check Interval' set to 30000, 'Health Check Interval' set to 15000, 'Health Check Periods Until Fencing' set to 3, and 'Fencing Grace Period' set to 30000.

» **Transaction Logs and JMS Persistence configuration:**

- For **database JMS and TLOG stores**, servers in the two sites can point to the same database and the same schema. This is the recommended approach for this topology because using TLOGs and JMS stores in the database has the advantage of incorporating the propagation of transaction logs to Site2 using Data Guard. Automatic Service Migration between sites can happen without any additional intervention.

NOTE: This can cause an average performance degradation of around 5-10% for the FOD example (the impact will vary depending on the application or composite type). This effect is aggravated when servers need to access the database in another site. Using Oracle WebLogic Server JMS with database persistence in Multi Data Center Active-Active Deployment will cause an even larger performance impact, especially with large payloads. The benefits of automatic preservation of messages vs. the performance degradation caused should be considered.

- For **file-based persistence** stores use two shared storages, each one local to each site (to enable service migration inside each site). For consistency, and to simplify backup procedures and other administration operations, the same paths can be used in both sites since persistence stores are qualified with the server's name. However, using a different path identifying each site is advisable to facilitate recovery (scenarios where Site2 needs to recover TLOGs from Site1 or vice versa). For example, in Site1 use the following path for the persistence store:

```
ORACLE_RUNTIME/domain_name/soa_cluster_name/tlogs_Site1
```

And in Site2 use:

```
ORACLE_RUNTIME/domain_name/soa_cluster_name/tlogs_Site2
```

In file-based persistence scenario, when a server fails and other servers in the same site remain available, service migration takes care of resuming transactions. When all of the servers in one site are unavailable and another site needs to resume transactions it is necessary to migrate MANUALLY the appropriate services to the available site, as described in the following procedure:

1. Make the appropriate transaction logs and persistent stores available in the new site (either through the appropriate disk replication or backups).
2. Manually migrate the migratable server of the server that has failed to another server in the other site.

- » **Replicated services:** To eliminate cross-site traffic, it is recommended that you use local affinity for JNDI context factory resolution. To do this, set the Default Load Algorithm to "round-robin affinity" (the default is round_robin) or to any "affinity-based" algorithm. This can be done in the General tab for the SOA Cluster Configuration as shown:

The screenshot shows the 'Settings for SOA_Cluster' configuration page in the Oracle WebLogic Server Administration Console. The page has a navigation bar with tabs for 'Configuration', 'Monitoring', 'Control', 'Deployments', 'Services', and 'Notes'. Under the 'Configuration' tab, there are sub-tabs for 'General', 'Messaging', 'Servers', 'Replication', 'Migration', 'Singleton Services', 'Scheduling', 'Overload', 'Health Monitoring', and 'HTTP'. A 'Save' button is located at the top left. Below the navigation, a message states: 'This page allows you to define the general settings for this cluster.' The 'Name' field is set to 'SOA_Cluster'. The 'Default Load Algorithm' is set to 'round-robin-affinity' via a dropdown menu.

- » **JMS Destinations:** Oracle WebLogic Server provides initial context and server affinity for client connections. JMS connection factories can be used further to restrict connections and provide site-affinity for a stretched cluster. Connection factories are replicated objects in the cluster. Clients get JNDI contexts, and from these JNDI contexts obtain a reference to the appropriate connection factory. By setting the default load balancing algorithm for the cluster to any affinity-type protocol (round-robin-affinity, weight-affinity, random-affinity) connection factories' stubs will be generated from a local server as first option. The connection factory client stubs then use the list of all servers that the connection factory is targeted to. By using local affinity at the connection factory level (which is set by default), the connection factory will connect to JMS servers that are local as first preference. It is therefore recommended to use affinity at cluster level (configured in the previous section) and also in the connection factories (which is the default). This will reduce undesired cross-site communication for Uniformed Distributed Destinations (UDDs) and Uniform Distributed Topics (UDTs). This mechanism, however, is not deterministic. For systems that use JMS destinations intensively and with large payloads it is recommended to completely avoid cross-site traffic for JMS invocations. This can be achieved using a selective targeting of subdeployment modules in each site. Consider for example the case where a specific UDD (for example, DemoSupplierQueueUDD) needs to be available in both sites, but the system is using large JMS payloads intensively. Traffic can be deterministically restricted to one site by using separate JMS modules in each site. Follow the following steps for this type of approach:

NOTE: This configuration forces destinations to be isolated (they are just available locally in each server); hence it may not be applicable for some systems where subsequent processing may depend on specific messages properties or when clients are remote to the destinations

1. Using the Oracle WebLogic Server Administration Console select Services > Messaging > JMS Modules and create a separate JMS module for each site. For each module select the servers in each site respectively as targets.

JMS Modules

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

<input type="checkbox"/>	Name ↕
<input type="checkbox"/>	BPMJMSModule
<input type="checkbox"/>	DemoSupplierModuleHAL
<input type="checkbox"/>	DemoSupplierModuleRWS
<input type="checkbox"/>	SOAJMSModule
<input type="checkbox"/>	UMSJMSSystemResource

2. Create a separate subdeployment module for those JMS servers that will host the destinations.

Home Log Out Preferences Record Help Welcome, weblogic Connected to: soaedg_domain

Home > JMS Modules > DemoSupplierModuleRWS > Summary of Deployments > JMS Modules > DemoSupplierModuleHAL

Settings for DemoSupplierModuleHAL

Configuration **Subdeployments** Targets Security Notes

This page displays subdeployments created for a JMS system module. A subdeployment is a mechanism by which JMS module resources (such as queues, topics, and connection factories) are grouped and targeted to a server resource (such as JMS servers, server instances, or clusters).

Customize this table

Subdeployments

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

<input type="checkbox"/>	Name ↕	Resources	Targets
<input type="checkbox"/>	SOAJMSServer 509784671	DemoSupplier:QueueUDD, DemoCF	SOAJMSServer_auto_3, SOAJMSServer_auto_4

3. Create the required UDDs and the pertaining connection factories. For UDDs use the same JNDI name in both sites. For connection factories, specify only local JNDI name. This name will be bound only on the local server instance and will not be propagated to the rest of the cluster. Thus the connection factory stubs will hold a reference only to the local JMS server when establishing a connection.

Settings for DemoCF

Configuration Subdeployment Notes

General Default Delivery Client Transactions Flow Control Load Balance Security

Save

Use this page to define the general configuration parameters for this JMS connection factory, which includes various client connection,

Name: DemoCF

JNDI Name:

Default Targeting Enabled

Advanced

Local JNDI Name:

Save

- Assign the subdeployment created in the previous step to the connection factories and UDDs and activate all changes.

This will guarantee that each server uses only local destinations avoiding cross-site JMS invocations.

- » **JMS Adapter configuration:** The JMS adapter requires configuring specific connection factory properties that include the list of servers available for JNDI context retrieval. In SOA Enterprise Deployment Guide 12c, it is recommended to use the t3 cluster syntax (example: cluster:t3://cluster_name) to simplify the configuration. Using this cluster syntax, the invocation fetches the complete list of members in the cluster at any given time, thus avoiding any dependencies on the initial servers and accounting for every member that is alive in the cluster at that point of time.

Settings for oracle.tip.adapter.jms.DJMSConnectionFactory

General **Properties** Transaction Authentication Connection Pool Logging

This page allows you to view and modify the configuration properties of this outbound connection pool. Properties you modify here are saved to

Outbound Connection Properties

Save

Property Name	Property Type	Property Value
AcknowledgeMode	java.lang.String	AUTO_ACKNOWLEDGE
ConnectionFactoryLocation	java.lang.String	weblogic.jms.XAConnectionFactory
FactoryProperties	java.lang.String	java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory;java.naming.provider.url=
IsTopic	java.lang.Boolean	false
IsTransacted	java.lang.Boolean	false
Password	java.lang.String	
Username	java.lang.String	

Save

For systems that use JMS destinations intensively and with large payloads, besides the considerations explained in the “JMS Destinations” bullet above, it is recommended that the JNDI URL used by the adapter contains a list of “local” servers in each site (that is, it uses only servers in Site1 for the configuration in Site1 and only servers in Site2 for the configuration on Site2). This will guarantee site context affinity⁶. To achieve this configuration follow these steps:

6. Update the Outbound Connection Pool properties for the instance that the adapter will use (as described in the *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*, Section 13.13.1.2) specifying as `java.naming.provider.url` the list of server in Site1). For example:

```
java.naming.provider.url=t3://Site1_server1:8001,Site1_server2:8001
```

7. Save the update in the Administration Console.
8. Copy the generated deployment plan to the mirror location on Site2. For example, from `server1_Site1`:

```
scp /u01/oracle/config/dp/soaedg_domain/SOA_cluster/JMSPlan.xml Site2_server1:/u01/oracle/config/dp/soaedg_domain/SOA_cluster/
```

9. Edit the deployment plan in Site2 and replace the server list with the list of servers in Site2.

Site1 Deployment Plan excerpt:

```
<name>ConfigProperty_FactoryProperties_Value_13243793917130</name>
<value>java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory;java.naming.provider.url=t3://Site1_server1:8001,Site1_server2:8001;java.naming.security.principal=weblogic;java.naming.security.credentials=welcome1</value>
```

Site2 Deployment plan excerpt:

```
<name>ConfigProperty_FactoryProperties_Value_13243793917130</name>
<value>java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory;java.naming.provider.url=t3://Site2_server1:8001,Site2_server2:8001;java.naming.security.principal=weblogic;java.naming.security.credentials=welcome1</value>
```

10. Update the JMS Adapter deployment using the modified deployment plan (the same location in both sites, but effectively a different file). The update will use the deployment plan file in Site1 for the servers in Site1 and the deployment plan file in Site2 for the servers in Site2.

» **File Adapter considerations:** Although the two sites act separately processing files (using separate share storage for each site), by default the same Data Source is used with the same schema and tables for file locking and file mutex. This schema is used for guaranteeing that the same file is processed only by one server at a time and that two adapter instances will not write to the same file concurrently. It is appropriate to use the same database for outbound operations since a unique sequence is used for the mutex that is used

⁶ For consumer scenarios, the JMS Adapter implicitly sets up at least one consumer on each member in the Distributed Destinations so the “locality” effect is mitigated.

to prevent overwriting files. For inbound operations, however, and since the two sites use (by default) the same schema in the same database, “under processing” scenarios could take place. This is because file adapter instances in either site can mark a file name as “blocked”. Because the same file name can be used in both sites, this can block its processing in both locations but the file will be consumed in only one of them. To avoid these situations, there are multiple alternatives:

- Guaranteeing unique file names for input operations across sites (notice that the path is the same since the corresponding jca file is unique in a stretched cluster).
- Using different schemas in the same database to guarantee proper processing.
- Using a separate database for mutex and locks to provide better performance than the previous alternatives and allow total parallel processing without races for locks if the same file is present in both site’s input directory.

NOTE: If due to middle tier failure at site level a server from Site1 needs to be started in Site2 (or the other way around), the Data Source must be updated to point to a local database/schema for improved performance.

In order to use different databases or separate schemas in the SOA database, the appropriate schema owner and tables need to be created and a new Data Source needs to be used for the File Adapter. The following steps provide the details for reconfiguring the adapter.

1. Site1 will use the default schema.
2. For Site2, use a different schema from the existing one (SOINFRA) (if you are using different databases you can actually use the same one). You can reuse other existing schemas (such as the one used for service migration leasing) or create a new one.
3. Connect to the database with the new schema.
4. Use the FILEADAPTER script in Appendix A: File Adapter Locks and Muxers to create the appropriate mutex and lock database objects.
5. Create a new GridLink Data Source for the schema. This must be of type “Oracle’s Driver (Thin XA) for GridLink Connections Versions:Any”.

The screenshot shows a wizard window titled "Create a New JDBC GridLink Data Source". It has navigation buttons: Back, Next, Finish, and Cancel. Below the title is the section "JDBC GridLink Data Source Properties" with a note: "The following properties will be used to identify your new JDBC GridLink data source. * Indicates required fields".

The first question is "What would you like to name your new JDBC GridLink data source?". The "Name" field contains "Fileadapter_site2".

The second question is "What JNDI name would you like to assign to your new JDBC GridLink data source?". The "JNDI Name" field contains "jdbc/Fileadapter_site2".

The third question is "What database type would you like to select?". The "Database Type" is set to "Oracle".

The fourth question is "What database driver would you like to use to create database connections? Note: * indicates that the driver is explicitly supported by Oracle WebLogic Server.". The "Database Driver" is set to "*Oracle's Driver (Thin XA) for GridLink Connections Versions:11 and later".

At the bottom, there are navigation buttons: Back, Next, Finish, and Cancel.

6. Enable FAN for the Data Source. You can copy the JDBC URL and ONS properties from the existing SOA schemas.
7. Target the Data Source to the SOA_Cluster.
8. Make a backup of the deployment plan for the File Adapter.

- ```
cp /u01/oracle/config/dp/soaedg_domain/FileAdapterPlan.xml
/u01/oracle/config/dp/soaedg_domain/FileAdapterPlan.xml.orig
```
9. Update the inbound Data Source with the JNDI name used in the new one.  

```
jdbc/Fileadapter_Site2
```
  10. Save the update in the Administration Console.
  11. Copy the generated deployment plan to the mirror location on Site2. For example:  

```
scp /u01/oracle/config/dp/soaedg_domain/FileAdapterPlan.xml
```
  12. Site2\_server:/u01/oracle/config/dp/soaedg\_domain/FileAdapterPlan.xml
  13. Revert to the original file in Site1  

```
cp
/u01/oracle/config/dp/soaedg_domain/FileAdapterPlan.xml.orig
/u01/oracle/config/dp/soaedg_domain/FileAdapterPlan.xml
```
  14. Access the Deployments screen in the Oracle WebLogic Server Administration Console and update the File Adapter deployment with the modified Deployment plan. Site1 will use the original Data Source while Site2 will use the new one.

## Configuring Data Sources for Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment

The Data Sources used by Oracle Fusion Middleware SOA Suite should be configured to automate failover of connections in case there is failover or switchover of the active database. A Data Guard database role transition (where a standby database becomes the new primary database) can be performed without restarting Oracle WebLogic Servers<sup>7</sup>. The following Data Sources need to be configured properly to automate this failover.

- » EDNDataSource
- » EDNLocalTxDataSource
- » LocalSvcTblDataSource
- » mds-owsm
- » mds-soa
- » opss-audit-DBDS
- » opss-audit-viewDS
- » opss-data-source
- » OraSDPMDDataSource
- » SOADDataSource
- » SOALocalTxDataSource
- » WLSSchemaDataSource

Additionally, any custom Data Source (used for persistent stores, leasing Data Sources, DB Adapter, etc.) should also be configured to automate failover. The [Oracle WebLogic Server and Highly Available Oracle Databases: Oracle Integrated Maximum Availability Solutions](#) paper contains all of the details required for configuring Data Sources for transparent fail over in disaster recovery scenarios for the SOA database. Beyond the required

---

<sup>7</sup> Refer to

Database Failures: Data Guard Switchover and for details on service migration implications when a database role change takes place.

database configuration (services, Oracle Net, and Data Guard), and since the default recommendation is to use GridLink Data Sources, the modifications on the middle tier should typically be limited to:

- » Updating the ONS configuration to include both production and standby site ONS.
- » Updating the JDBC URL to include the appropriate services in both sites

For database connection pool sizing, consider that the total number of processes in the database must sustain the addition of the pools of both sites. The following is a sample JDBC URL for the SOA Data Source in an Oracle Fusion Middleware SOA Multi Data Center Active-Active configuration where the database uses Data Guard:

```
jdbc:oracle:thin:@
(DESCRIPTION=
 (CONNECT_TIMEOUT=15)
 (RETRY_COUNT=5)
 (RETRY_DELAY=5)
 (TRANSPORT_CONNECT_TIMEOUT=3)
 (ADDRESS_LIST=
 (LOAD_BALANCE=on)
 (ADDRESS=(PROTOCOL=TCP) (HOST=scanSite1.mycompany.com) (PORT=1521))
)
 (ADDRESS_LIST=
 (LOAD_BALANCE=on)
 (ADDRESS=(PROTOCOL=TCP) (HOST=scanSite2.mycompany.com) (PORT=1521))
)
 (CONNECT_DATA=(SERVICE_NAME=soaedg.us.oracle.com))
)
```

In addition to the Data Sources configuration, the JDBC URL of the OPSS Security Stores needs to be updated to include the appropriate services in both sites. This connection string is configured in the file `jps-config.xml` and `jps-config-jse.xml`, located in the folder `ASERVER_HOME/config/fmwconfig/`.

This connection string sets `LOAD_BALANCE` to OFF at the `DESCRIPTION` level (default value). This means that the address lists are attempted in the order provided in the list. Hence, the second `ADDRESS_LIST` is attempted only if the first one is not available. Additionally, `LOAD_BALANCE` is explicitly set to ON within each `ADDRESS_LIST`. This means that the client will connect to one of the 3 scan IPs of the `ADDRESS` in a random order, so that the connections are load balanced between the 3 scan IPs of the scan address.

The timeout and retry parameters provide resiliency in the switchover/failover operations, during which the database may be unavailable for a few minutes.

You can provide this connect string directly in the datasource and jps related files. Make sure you do not use blank spaces or line breaks when entering the connect string.

Alternatively to entering the connect string directly in the datasource and jps files, you can use a TNS alias in the jdbc url and point to a `tnsnames.ora` containing an alias mapped to a connection string like the one above. For details on using a TNS alias, refer to the [Configuring Data Sources for Oracle Fusion Middleware Active-Passive Deployment](#) section in the [Oracle Fusion Middleware Disaster Recovery Guide 12.2.1.4](#).



## Composite and MDS Deployments and Updates: Oracle Coherence Configuration

Oracle Fusion Middleware SOA Suite uses Oracle Coherence for propagating both composite deployments and metadata (MDS) updates across a cluster. Oracle recommends using unicast for Coherence communications in context of the Oracle FMW SOA Suite EDG. Unicast is most often used when multicast networking is undesirable or unavailable in an environment or when an environment is not properly configured to support multicast. This is typically the case for systems that span a MAN. The Well Known Addresses (WKA) feature is a Coherence mechanism that allows cluster members to discover and join a cluster using unicast instead of multicast. All multicast communications are disabled for the coherence cluster if WKA is enabled. Oracle Fusion Middleware products and components default to a Well Known Address (WKA) list so Oracle Coherence uses unicast for communications in the SOA Cluster.

Coherence is sensitive to delays in cluster formation and responsiveness to heartbeats from members that are part of the cluster. With latencies ranging from 5/10 msec RTT no issues were detected for different operations involving the Coherence infrastructure used by SOA (cluster creation, new nodes joining cluster across sites, MDS updates, and composite deployments).

For high latencies above 10 msec RTT, Coherence may report errors both in cluster formation and also in keeping members updated. Expect messages like the following on the servers that come up first in such situations:

```
<TIME> <Warning> <com.oracle.coherence> <BEA-000000> <TIME/159.625 Oracle
Coherence GE 12.2.1.3.0 <Warning> (thread=Cluster, member=n/a): This
Member(Id=0, Timestamp=TIME, Address=X.X.X.X:34483, MachineId=40262,
Location=machine:HOSTNAME,process:5863,member:WLS_SERVER, Role=CLUSTER_NAME)
has been attempting to join the cluster using WKA list [/site1_server1:9991,
/site1_server2:9991, /site2_server1:9991, /site2_server2:9991] for 51
seconds without success; this could indicate a mis-configured WKA, or it may
simply be the result of a busy cluster or active failover.>
```

If there are multiple failed tries for joining the cluster, the nodes trying to join will reports messages like the following:

```
<TIME> <Warning> <com.oracle.coherence> <BEA-000000> <TIME Oracle Coherence GE
 12.2.1.3.0 <Warning> (thread=Cluster, member=n/a): Received a discovery
 message that indicates the presence of an existing cluster:
Message "NewMemberAnnounceReply"
 {
 FromMember=Member(Id=1, Timestamp=TIME, Address=x.x.x.x:38785,
MachineId=40260, Location=machine:HOSTNAME,process:269986,member:WLS_SERVER,
Role=CLUSTER_NAME)
 FromMessageId=0
 MessagePartCount=0
 PendingCount=0
 Delivered
 MessageType=8
 ToPollId=0
 Poll=null
 Packets
 {
 }
 Service=ClusterService{Name=Cluster, State=(SERVICE_STARTED, STATE_JOINING),
Id=0}
 ToMemberSet=null
 NotifyDelivery=false
 ToMember=Member(Id=0, Timestamp=TIME, Address=10.133.41.137:34483,
MachineId=40262,
Location=machine:HOSTNAME,process:5863,member:THIS_WLS_SERVER,
Role=CLUSTER_NAME)
 PrevSentTimestamp=TIMESTAMP
 PrevRecvTimestamp=TIMESTAMP
 ThisSentTimestamp=TIMESTAMP
 ThisRecvTimestamp= TIMESTAMP
 MaxDeliveryVariance=501
 }>
```

Finally, if the error persists, the new member trying to join will report a failure and the Coherence service will be stopped, and after some time reports the following:

```
<TIME> <Error> <com.oracle.coherence> <HOSTNAME> <WLS_SERVER>
<Logger@2008407832 12.2.1.3.0> <<anonymous>> <> <ID> <1506347752198>
<[severity-value: 8] [rid: 0] [partition-id: 0] [partition-name: DOMAIN] >
<BEA-000000> <TIME Oracle Coherence GE 12.2.1.3.0 <Error> (thread=Cluster,
member=n/a): Failure to join a cluster for 300 seconds; stopping cluster
service.>

<TIME> <Trace> <com.oracle.coherence> <HOSTNAME> <WLS_SERVER>
<Logger@9219882 12.2.1.3.0> <<anonymous>> <> <ID> <1506509839389>
<[severity-value: 256] [rid: 0] [partition-id: 0] [partition-name: DOMAIN] >
<BEA-000000> <[com.tangosol.coherence.component.util.logOutput.Jdk:log] TIME
Oracle Coherence GE 12.2.1.3.0 <D5> (thread=Cluster, member=n/a): Service
Cluster left the cluster>

<TIME> <Error> <CoherenceIntegration> <HOSTNAME> <WLS_SERVER> <[ACTIVE]
ExecuteThread: '7' for queue: 'weblogic.kernel.Default (self-tuning)'\> <<WLS
Kernel>> <> <ID> <1506509839390> <[severity-value: 8] [rid: 0] [partition-
id: 0] [partition-name: DOMAIN] > <BEA-2194507> <The Coherence cluster
service failed to start or failed to join cluster due to
com.tangosol.net.RequestTimeoutException: Timeout during service start:
ServiceInfo(Id=1, Name=TransportService,
..
 at
com.tangosol.coherence.component.util.daemon.queueProcessor.service.Grid.onS
tartupTimeout(Grid.CDB:3)
 at
com.tangosol.coherence.component.util.daemon.queueProcessor.Service.start(Se
rvice.CDB:28)
 at
com.tangosol.coherence.component.util.daemon.queueProcessor.service.Grid.sta
rt(Grid.CDB:6)
 at
com.tangosol.coherence.component.net.Cluster.startSystemServices(Cluster.CDB
:11)
 at
com.tangosol.coherence.component.net.Cluster.onStart(Cluster.CDB:53)
 at com.tangosol.coherence.component.net.Cluster.start(Cluster.CDB:12)
```

In these situations, it is recommended that you adjust the coherence parameters related with networking timeouts:

- » **multicast-listener/join-timeout-milliseconds.** Despite being originally designed for multicast configurations, join-timeout impact in unicast as well. It determines how long the initial node will take to form a cluster, and after that how long each node will spend looking for the cluster before trying to form their own (if they are on the WKA list). If you are on an unreliable network may need to try for a longer period of time to find a cluster (because of lost packets) before starting a new one.
- » **packet-publisher/packet-delivery/timeout-milliseconds.** This timeout should be high, the default value of 5 min should be good, but you can increase it to avoid timeouts when joining the cluster.
- » **packet-publisher/packet-delivery/resend-milliseconds.** The packet resend interval specifies the minimum amount of time, in milliseconds, that the packet publisher waits for a corresponding ACK packet, before resending a packet. This should be a few multiples of the RTT, 10x should be fine.

- » **tcp-ring-listener/ip-timeout and ip-attempts.** These are the amount of time and attempts before determining that a computer that is hosting cluster members has become unreachable. Ip-timeout should be in the area of 10x the RTT or greater, with a minimum of 5s.

Using a Coherence override file allows you to change the default values. Create a file custom-coherence-override.xml as shown here and make sure that this file is consistent and available for all the servers in the cluster:

```
<?xml version='1.0'?>

<!--
This is a custom operational configuration override
-->
<coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="http://xmlns.oracle.com/coherence/coherence-
operational-config"

xsi:schemaLocation="http://xmlns.oracle.com/coherence/coherence-
operational-config coherence-operational-config.xsd"
 >
 <cluster-config>
 <multicast-listener>
 <join-timeout-milliseconds>5000</join-timeout-milliseconds>
 </multicast-listener>
 <tcp-ring-listener>
 <ip-timeout>20s</ip-timeout>
 <ip-attempts>3</ip-attempts>
 </tcp-ring-listener>
 <packet-publisher>
 <packet-delivery>
 <timeout-milliseconds>650000</timeout-milliseconds>
 <resend-milliseconds>200</resend-milliseconds>
 </packet-delivery>
 </packet-publisher>
 </cluster-config>
</coherence>
```

This file is specified in the java property tangosol.coherence.override, that can be set in the EXTRA\_JAVA\_PROPERTIES java property in the setUserOverrides.sh file, for example:

```
EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES} -
Dtangosol.coherence.override=/u01/oracle/config/coherence_custom/custom
-coherence-override.xml"
```

If needed, tcp-ring death detection can be tuned or even disabled. To tune it, the death detection heartbeat interval can be increased. A higher interval alleviates network traffic but also prolongs detection of failed members. The default heartbeat value is 1 second. To change the death detection heartbeat interval, edit the operational override file and add a <heartbeat-milliseconds> element that includes the heartbeat value. For example:

```
<?xml version='1.0'?>
```

```

<coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="http://xmlns.oracle.com/coherence/coherence-operational-
 config"
 xsi:schemaLocation="http://xmlns.oracle.com/coherence/
 coherence-operational-config coherence-operational-config.xsd">
 <cluster-config>
 <packet-publisher>
 <packet-delivery>
 <heartbeat-milliseconds>5000</heartbeat-milliseconds>
 </packet-delivery>
 </packet-publisher>
 </cluster-config>
</coherence>

```

Disabling death detection can alleviate network traffic but also makes the detection of failed members take longer. If disabled, a cluster member uses the packet publisher's resend timeout interval to determine that another member has stopped responding to UDP packets. By default, the timeout interval is set to 5 minutes. Refer to the coherence documentation for changing the packet resend timeout. To disable death detection, tangosol-coherence-override.xml file and add an `enabled` element that is set to false. For example:

```

<?xml version='1.0'?>

<coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="http://xmlns.mycompany.com/coherence/coherence-operational-
 config"
 xsi:schemaLocation="http://xmlns.mycompany.com/coherence/
 coherence-operational-config coherence-operational-config.xsd">
 <cluster-config>
 <tcp-ring-listener>
 <enabled>false</enabled>
 </tcp-ring-listener>
 </cluster-config>
</coherence>

```

Composite deployments (first version or updates to newer versions) are not activated in the SOA servers until they are available in all members of the cluster (servers must be in the RUNNING state and with the soa-infra application in the Activestate). The composite version will be listed by the soa-infra application as Loaded in the SOA server's output file, but invoking this version while it is being deployed to other servers generates a SOA fault (that is, activation is precluded until all servers complete the deployment). In a Multi Data Center Deployment this is especially relevant because remote access to the MDS schemas may cause long delays in activating composites in those servers with higher latency to the database. Planned downtimes or activation windows should be planned based on these "slowest" members.

## In-memory soa

In-memory soa is a feature that leverages the Coherence cache associated with WebLogic Server to run the non-transactional business processes in memory. This improves performance and scalability for these business processes as read and write operations are performed out of the cache. The process state gets written to the database only when faulted, or at regular, deferred intervals using a write-behind thread. The BPEL state information is dehydrated and rehydrated to/from the Coherence cache.

In-memory soa is not supported to be used across stretched clusters, where the usage of the coherence cache must be minimal as it is very sensitive to delays.

## Setting Appropriate Timeouts for Synchronous and Asynchronous Operations

Timeouts may occur in different layers of the Oracle Fusion Middleware stack in a running Oracle Fusion Middleware SOA system. There are timeout periods specified for transactions in the database, for transactions branches, EJB method invocations, web services, etc. Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployments are especially sensitive to timeout settings because multiple operations performed by some servers need to access the database in a different location. Timeouts may need to be increased in these types of systems due to the latencies involved. In the Stretched Cluster model, domain settings are shared for both sites; hence it is required to use timeouts that account for the worst case scenario. Additionally it is required that timeouts are configured in the different layers of the SOA system so that the appropriate embracing tiers behave properly (for example, if the database timeout is set to a value lower than the Global WLS timeout, it may occur that a transaction ID is “removed” from the database before the work on other branches completes). In summary timeouts need to be configured such that:

- » They account for the latency in the system
- » They expire properly in the chain of invocations across different layers

There are different parameters that may be configured for expiring requests in different layers. These are the main ones:

- » **Timeout in the application server invocations** (also known as global transaction timeout). This is the Oracle WebLogic Server's global transaction timeout in seconds for active transactions. It is configured through the Oracle WebLogic Server Administration Console. In the Oracle WebLogic Administration Console select Domain in the Navigation tree on the left, then Services > JTA > Timeout Seconds.

The screenshot displays the Oracle WebLogic Server Administration Console interface. The main content area is titled "Settings for soaedg\_domain" and includes a "Configuration" tab. Within this tab, the "JTA" sub-tab is active, showing a "Timeout Seconds" field with the value "300". The left-hand navigation pane shows a tree view with "soaedg\_domain" expanded to "Services > JTA". The top navigation bar includes "Home", "Log Out", "Preferences", "Record", and "Help".

- » **Timeouts in XA Data Sources:** This is used in Oracle WebLogic Server to set a transaction branch timeout for Data Sources (You may want to set it if you have long-running transactions that exceed the default timeout value on the XA resource). It is configured in the Transaction tab for each XA Data Source.

Settings for SOADDataSource

Configuration | Targets | Monitoring | Control | Security | Notes

General | Connection Pool | Oracle | ONS | **Transaction** | Diagnostics | Identity Options

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.

Save

The transaction protocol for a JDBC data source determines how connections from the data source are handled during transaction processing. This page enables you to define transaction options for this JDBC data source.

Use XA Data Source Interface

Set XA Transaction Timeout

XA Transaction Timeout:

- » **Timeouts for distributed locking in the database** (distributed\_lock\_timeout): this specifies the amount of time (in seconds) for distributed transactions to wait for locked resources. It can be modified with the appropriate ALTER statements or using Oracle Enterprise Manager Database Control as shown:

ORACLE Enterprise Manager 11g Database Control

Cluster Database: soaedg.us.oracle.com >

Initialization Parameters

Current | SPFile

The parameter values listed here are currently used by the running instance(s). You can change static parameters in SPFile mode.

Name: distributed Basic: All Modified: All Dynamic: All Category: All Go

Filter on a name or partial name

Apply changes in current running instance(s) mode to SPFile. For static parameters, you must restart the database.

Add | Reset

| Select                           | Instance | Name                     | Help              | Revisions | Value | Comments | Type    |
|----------------------------------|----------|--------------------------|-------------------|-----------|-------|----------|---------|
| <input checked="" type="radio"/> | *        | distributed_lock_timeout | <a href="#">?</a> | 6         | 700   |          | Integer |

Current | SPFile

In an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment, you should set the distributed locking timeout for the database so that it accounts for the longest running database transactional operation possible (accounting for the delays introduced by the latency across sites.) Once this value is set, configure the XA Data Sources and Global Transaction Timeouts to lower values:

distributed\_lock\_timeout >= XA DS Timeout >= Global Transaction Timeout

Additionally, and for BPEL synchronous processes, there are other parameters that may control timeouts:

- » **Time that a synchronous client should wait for an answer** (syncMaxWaitTime). This property defines the maximum time a request and response operation takes before timing out. If the BPEL process service component does not receive a reply within the specified time, then the activity fails. To modify it using Oracle Enterprise Manager Fusion Middleware Control right click soa-infra then select SOA Administration > BPEL Properties > More BPEL Configuration Properties > syncMaxWaitTime.

The screenshot shows the Oracle Enterprise Manager Fusion Middleware Control interface. The left sidebar displays the navigation tree with 'soa-infra' selected. The main content area shows the 'Application Defined MBeans: BPELConfig:bpel' page. The 'Attributes' tab is active, showing a table of configuration properties. The 'SyncMaxWaitTime' attribute is circled in red.

| Name                            | Description                                                          | Access | Value                   |
|---------------------------------|----------------------------------------------------------------------|--------|-------------------------|
| readOnly                        | If true, it indicates that this MBean is a read-only MBean.          | R      | false                   |
| 24 RecoveryConfig               | Recovery Configuration                                               | RW     | javax.management.openmb |
| 25 RecurringMaxMessageRaiseSize | Number of messages to recover during recurring recovery              | RW     | 50                      |
| 26 RestartNeeded                | Indicates whether a restart is needed.                               | R      | false                   |
| 27 SpecCompliance               | Sets how the implementation is compliant to the spec                 | RW     | suggest                 |
| 28 StartupMaxMessageRaiseSize   | Number of messages to recover during startup recovery                | RW     | 50                      |
| 29 StatsLastN                   | The size of the "most recently processed" request list               | RW     | -1                      |
| 30 SyncMaxWaitTime              | The maximum time a request/response operation will take be...        | RW     | 45                      |
| 31 SystemMBean                  | If true, it indicates that this MBean is a System MBean.             | R      | false                   |
| 32 ValidateXML                  | If set to "true" the engine will apply schema validation for inco... | RW     | false                   |
| 33 Version                      | version of the config file                                           | R      | 11.1.0                  |

- » **Timeouts for EJBs** (applicable especially to BPEL): when the BPEL EJBs methods are involved, this specifies the transaction timeout period in seconds (default is 300 for all BPEL EJBs). It can be modified using the Oracle WebLogic Server Administration Console by selecting **Deployment** then expand soa\_infra application, and then click in the specific BPEL EJB.

|                                      |                                                          |
|--------------------------------------|----------------------------------------------------------|
| <b>Name:</b>                         | BPELDispatcherBean                                       |
| <b>Type:</b>                         | stateless                                                |
| <b>EJB Class Name:</b>               | com.collaxa.cube.engine.ejb.impl.bpel.BPELDispatcherBean |
| <b>Pool Configuration</b>            |                                                          |
| <b>Initial Beans in Free Pool:</b>   | 100                                                      |
| <b>Max Beans in Free Pool:</b>       | <input type="text" value="1000"/>                        |
| <b>Idle Timeout:</b>                 | <input type="text" value="0"/>                           |
| <b>Enterprise Bean Configuration</b> |                                                          |
| <b>Network Access Point:</b>         | <input type="text"/>                                     |
| <b>Run As Principal Name:</b>        | <input type="text"/>                                     |
| <b>Create As Principal Name:</b>     | <input type="text"/>                                     |
| <b>Remove As Principal Name:</b>     | <input type="text"/>                                     |
| <b>Passivate As Principal Name:</b>  | <input type="text"/>                                     |
| <b>JNDI Name:</b>                    | <input type="text"/>                                     |
| <b>Local JNDI Name:</b>              | <input type="text"/>                                     |
| <b>Dispatch Policy:</b>              | <input type="text"/>                                     |
| <b>Transaction Timeout:</b>          | <input type="text" value="300"/>                         |

To elude exceptions and SOA faults in an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment where transactions are dropped before the processes are completed (typically reported as “thread is NOT associated with a transaction”) the recommendation is to set

$$\text{syncMaxWaitTime} < \text{BPEL EJB's transaction timeout} < \text{Global Transaction Timeout}$$

Also, any web services clients that the system exposes must be configured with a timeout that is permissive enough with worst latency (that is, if the invocation takes 3 seconds though Site1, but 10 seconds when it goes though Site2, the latter will set the general limit for invocations).

Finally, if the BPEL system is using specific Request-reply (synchronous) and In-only receive (asynchronous) timeouts, they must be defined based on the worst case scenarios for a SOA invocation in context of the Multi Data Center Deployment topology. This would be the case where the invocation is routed to the site with the highest latency in accessing the SOA database. These settings are part of the BPEL process definition, and since there is a unique MDS/composite repository, they cannot be customized for each site. Refer to the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* for more details on using events and timeouts in BPEL processes.

## Session Replication Implications

There are some applications (especially administration consoles like SOA's composer, Oracle Business Process Management BPM composer, BPM workspace, and the Oracle B2B console) that make intensive use of HTTP session objects. One of the advantages of an Oracle Fusion Middleware SOA Multi Data Center Active-Active

Deployment is the ability to seamlessly failover users from one site to another when they are accessing these consoles. However, session replication across data centers may cause serious performance degradation in the system. Oracle recommends defining two different replication groups (one for each site) to minimize the possibility of replication happening across the two sites. To configure session replication groups, use the Oracle WebLogic Administration Console to access Environment > Servers > Server Name > Cluster SubTab as shown:



For each server in Site1 enter the same replication group name (for example Site1RG). Repeat the operation for servers in Site2 using a common name for all of them but different from the one used in Site1 (for example Site2RG).

**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 others are available in the other, replication will occur across the MAN and will continue for that session even if servers come back online in the same site.

## Optimizing Oracle Net Services Performance

Operating System and Oracle Net tuning play a critical role in data transmission across Metropolitan Area Networks (MAN). However, the effect of these adjustments is more relevant for higher latencies between JDBC clients and servers. In an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment the servers with the highest latency in accessing the database become the drivers for network buffers and Oracle Net adjustments. Some operating system default configurations are not optimized for current Ethernet speeds and it becomes very important to adjust some parameters to make data transfer more efficient. One of the key metrics in determining the optimum configuration is the Bandwidth-delay product. This is the product of network bandwidth and the round trip time/delay of data going over the network. A simple way to determine the round trip time (RTT) is to use a command such as *ping* from one host to another and use the response times returned. Ideally this parameter should be obtained as an average over a few minutes to eliminate momentary deviations. In Linux, this average can be obtained using the command *ping* directly.

```
[orcl@host1_Site1~]$ ping host1_Site2
```

```
64 bytes from host1_Site2 icmp_seq=0 ttl=61 time=7.64 ms
64 bytes from host1_Site2 icmp_seq=1 ttl=61 time=8.43 ms
64 bytes from host1_Site2 icmp_seq=2 ttl=61 time=7.62 ms
64 bytes from host1_Site2 icmp_seq=3 ttl=61 time=7.76 ms
```

```
64 bytes from host1_Site2: icmp_seq=4 ttl=61 time=6.71 ms
```

```
--- host1_Site2 ping statistics ---
```

```
5 packets transmitted, 5 received, 0% packet loss, time 4014ms
```

```
rtt min/avg/max/mdev = 6.715/7.634/8.430/0.552 ms, pipe 2
```

TCP socket buffer settings control how many packets are sent at one time over the network. In some operating systems default settings need to be increased in order to improve utilization of available bandwidth. When network latency is high, larger socket buffer sizes are better to fully utilize network bandwidth. The optimal socket buffer size is two times the size of the Bandwidth Delay Product (BDP) (which would typically be  $RTT \times BW$  since  $RTT$  is  $2 \times \text{delay}$ ) and should be set both for the database server and the Oracle Fusion Middleware nodes. Larger buffers than  $2 \times BDP$  are beneficial in most cases (overall with higher latencies that can present larger deviations in  $RTT$ ) but also consume more memory. Resource utilization should be monitored, and if memory is available increasing the buffer improves the data transmission behavior. The size of the socket buffers should be adjusted both in Oracle Fusion Middleware nodes and database servers.

### Configuring I/O Buffer Size in the Database Server

Since the database server primarily writes data to clients, setting the `SEND_BUF_SIZE` parameter on the server-side is typically enough. If the database server is receiving large requests, then also set the `RECV_BUF_SIZE` parameter. It is recommended to set them at the Oracle Net level in the database so that normal TCP sessions such as telnet, ssh, etc. do not use additional memory unless these other protocols are in similar need for optimization. To configure the database server, set the buffer space size in the listener.ora and sqlnet.ora files. In the listener.ora file, specify the buffer space parameters for a particular protocol address or for a description. The following is an example of the settings for a typical Oracle RAC-scan listener configuration:

```
LISTENER_SCAN3 =
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL =
IPC) (SEND_BUF_SIZE=10485760) (RECV_BUF_SIZE=10485760) (KEY =
LISTENER_SCAN3)))
LISTENER_SCAN2 =
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL =
IPC) (SEND_BUF_SIZE=10485760) (RECV_BUF_SIZE=10485760) (KEY =
LISTENER_SCAN2))
)
LISTENER_SCAN1 =
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL =
IPC) (SEND_BUF_SIZE=10485760) (RECV_BUF_SIZE=10485760) (KEY =
LISTENER_SCAN1))
)
```

### Configuring I/O Buffer Size on the Oracle Fusion Middleware Nodes

Unfortunately, as of Oracle 12c JDBC thin clients cannot specify socket buffer sizes, hence buffers need to be adjusted in the operating system. Recent versions of Linux (version 2.6.17 and later) use auto tuning with a 4 MB maximum buffer sizes. Enabling or disabling of auto tuning is determined by the



`/proc/sys/net/ipv4/tcp_moderate_rcvbuf` parameter. If the parameter `tcp_moderate_rcvbuf` is present and has a value of 1 then auto tuning is enabled. With auto tuning, the receiver buffer size (and TCP window size) is dynamically updated (auto-tuned) for each connection. Sender-side auto tuning has been present and unconditionally enabled for many years in Linux kernels. The per connection memory space defaults are set with two element files:

```
/proc/sys/net/ipv4/tcp_rmem - memory reserved for TCP receive buffers
```

```
/proc/sys/net/ipv4/tcp_wmem - memory reserved for TCP send buffers
```

These files contain three values: minimum, initial, and maximum buffer size. They are used to set the bounds on auto tuning and balance memory usage while under memory stress. The maximum values must be larger than the maximum value of  $2 \times \text{BDP}$ . With auto tuning, an excessively large initial buffer wastes memory and can hurt performance. Additionally, the maximum buffer size that applications can request can also be limited with `/proc` variables:

```
/proc/sys/net/core/rmem_max - maximum receive window
```

```
/proc/sys/net/core/wmem_max - maximum send window
```

The following are sample recommendations for TCP optimization in Linux:

- » Use TCP auto-tuning in kernel (2.4.27, 2.6.7)

```
/proc/sys/net/ipv4/tcp_moderate_rcvbuf (1=on)
```

- » Tune TCP Max Memory:

```
/proc/sys/net/ipv4/tcp_rmem and tcp_wmem
- 4096 87380 174760
```

Set the maximum (last value in the example=174760) to a value larger than  $2 \times \text{BDP}$ .

- » Tune the socket window sizes.

```
/proc/sys/net/core/rmem_max and wmem_max
```

Set this to larger than  $2 \times \text{BDP}$ .

- » Ensure that TCP Performance features are enabled (set all of the following to 1).

```
/proc/sys/net/ipv4/tcp_sack
/proc/sys/net/ipv4/tcp_window_scaling
/proc/sys/net/ipv4/tcp_timestamps
```

## Configuring Session Data Unit

Oracle Net sends data in packages with a specific size. Oracle Net waits for these units to be “filled” before sending them across the network. Each of these buffers is called a session data unit (SDU). Adjusting the size of the SDU to the amount of data provided to Oracle Net can improve performance, network utilization, and memory consumption in an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment with higher latencies than 5 msec RTT. The SDU size can be set from 512 bytes to 2 MB. The default SDU for the client and a dedicated server is 8192 bytes. The default SDU for a shared server is 65535 bytes. The actual SDU size used is negotiated between the client and the server at connect time and is the smaller of the client and server values. Configuring an SDU size different from the default requires configuring the SDU on both the client and server computers, unless you are using shared servers. For shared servers, only the client value must be changed because the shared server defaults to the maximum value. Oracle recommends setting the SDU to the maximum value possible (64k) as this has been verified to provide the best results in systems with different latencies. To set the SDU on the Oracle Fusion

Middleware servers, update the JDBC URL connection string in each of the connection pools used by SOA Data Sources. For example:

```
jdbc:oracle:thin:@
(DESCRIPTION=
 (CONNECT_TIMEOUT=15)
 (RETRY_COUNT=5)
 (RETRY_DELAY=5)
 (TRANSPORT_CONNECT_TIMEOUT=3)
 (ADDRESS_LIST=
 (LOAD_BALANCE=on)
 (ADDRESS=(PROTOCOL=TCP) (HOST=scanSite1.mycompany.com) (PORT=1521)
(SDU=65535)
)
 (ADDRESS_LIST=
 (LOAD_BALANCE=on)
 (ADDRESS=(PROTOCOL=TCP) (HOST=scanSite2.mycompany.com) (PORT=1521)
(SDU=65535)
)
 (CONNECT_DATA=(SERVICE_NAME=soaedg.us.oracle.com))
)
```

To set the SDU on the database servers, you can modify the listener configuration file (LISTENER.ORA) (this will set the SDU on a “per connection basis”, or you can set the SDU for all Oracle Net connections with the profile parameter DEFAULT\_SDU\_SIZE in the sqlnet.ora file. The following example sets the SDU to its recommended value for Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment in each of the SCAN listeners for the Oracle RAC Database:

```
LISTENER_SCAN3 =
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL = IPC)
 (SDU=65535) (SEND_BUF_SIZE=10485760) (RECV_BUF_SIZE=10485760) (KEY =
LISTENER_SCAN3))
)

LISTENER_SCAN2 =
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL = IPC)
 (SDU=65535) (SEND_BUF_SIZE=10485760) (RECV_BUF_SIZE=10485760) (KEY =
LISTENER_SCAN2))
)

LISTENER_SCAN1 =
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL = IPC)
 (SDU=65535) (SEND_BUF_SIZE=10485760) (RECV_BUF_SIZE=10485760) (KEY =
LISTENER_SCAN1))
)
```



The benefit of larger SDU values should be noticeable in scenarios where large payloads are transferred to the database. In all cases, the SDU size should be less than or equal than the socket buffer to avoid fragmentation.

## Failures in Different Tiers and Switchover/Failover Behavior

The Oracle Fusion Middleware SOA Multi Data Center Active-Active topology is resilient to failures in any component. Since each site uses the Oracle Fusion Middleware SOA EDG Topology, failures at component level (LBR, OHS instance, Oracle WebLogic Server, database instance) should not cause any disruption because each site provides local redundancy for each component. When an entire tier fails in a site, different aspects need to be considered as discussed in the following sections.

### Failure in All OHS Instances in One Site

OHS instances typically reside in isolated hardware (for security and administration purposes). It may occur that all instances of OHS in one site become unavailable due to a failure in the hardware hosting them. If a site loses all instances, the Oracle WebLogic Server SOA servers may continue processing SOA instances as long as no HTTP callbacks occur. If internally-generated callbacks occur, because the LBR is configured with rules to route to the site originating the callback, they will fail. However, the server may continue processing JMS and locally optimized invocations. If restoration of the lost OHS instances is not possible in the short term, the instances in the other site may be used to route requests to the SOA servers on the “OHS-orphan” site. In the stretched cluster model, this would require setting the DynamicServerList to ON. This change can be applied in a rolling manner to eliminate downtime. Additionally, and because rules have been defined to perform smart routing to each site based on the request’s origin, make sure that the appropriate monitors are set in the GLBR to stop routing to OHS when this type of failure occurs.

### Failure in All Oracle WebLogic Server SOA Servers in One Site

When all the Oracle WebLogic Server SOA servers are down in one site, the other site continues processing requests. The local LBRs should stop routing to the corresponding OHS servers (GLBR should stop routing to that site). From the BPEL perspective if the automatic recovery cluster master was hosted in the failed servers, a new cluster master will arise in the available site. This server can perform automated recovery of instances initiated on the other site.

JMS services and TLOGs will be automatically migrated to the servers in the other site when using Automatic Service Migration along with JDBC persistent stores (refer to the service migration paragraph in the configuration sections above).

### Administration Server Failure

The standard consideration for Administration Server failure scenarios that apply in a single data center topology apply to Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment. Node failures should be addressed with the standard failover procedures described in *Oracle Fusion Middleware High Availability Guide* and *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite* (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). Additionally, the appropriate backup and restore procedures should be deployed to make regular copies of the Administration Server’s domain directory. In case of a failure that affects the site hosting the Administration Server (involving all nodes), it may be required to restart the server in a different site. To do this, follow these steps:

1. Make the backup (or disk mirror/copy) of your Administration Server’s domain directory available in the failover site.

```
scp /u01/orcl/backups/Multi_Data_Center_Deployment_soaedg_domain
```

```
_STRETCHED.gz server1_Site2:/u01/
oracle/config/soaedg_domain
```

2. Restore the aserver/ directory (including both the soaedg\_domain and applications directory) in the failover site so that the exact same domain directory structure is created for the Administration Server's domain directory as in the original site.

```
cd /u01/oracle/config/soaedg_domain/
tar -xzvf
/u01/oracle/config/soaedg_domain/Multi_Data_Center_Deploymen
t_soaedg_domain_STRETCHED.gz
```

3. Modify \$NM\_HOME/nodemanager.domains in the server of Site2 where the Administration Server will be restored to include the ASERVER\_HOME:

```
domain_name=MSERVER_HOME;ASERVER_HOME
```

4. Restart Node Manager in the node where the Administration Server will be restored.
5. Likely, the Administration Server will be 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 hostname resolution system in this subnet so that this VIP will map to the original Virtual Hostname that the Administration Server used as listen address in Site1 (for example, in Site1, ADMINHOSTVHN1 may map to 10.10.10.1 while in Site2 either local /etc/hosts or DNS server will have to be updated so that ADMINHOSTVHN1 maps to 20.20.20.1). All servers will use ADMINHOSTVHN1 as the address to reach the Administration Server). If the Administration Server is front ended with an OHS and LBR as prescribed in the Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite, clients will be agnostic to this change. If clients directly access the Administration Server's listen hostname, they must be updated in their DNS resolution also.

Sample /etc/hosts when Admin Server is running in Site1:

```
10.10.10.1 ADMINHOSTVHN1.mycompany.com ADMINHOSTVHN1
```

Sample /etc/hosts when Admin Server is running in Site2:

```
20.20.20.1 ADMINHOSTVHN1.mycompany.com ADMINHOSTVHN1
```

6. Start WLST and connect to Node Manager with nmconnect and the credentials set for the Administration Server using nmstart. Enter the Node Manager user name and password used in the original site.

```
cd ORACLE_COMMON_HOME/common/bin
./wlst.sh
```

Once you are in the WLST shell:

```
wls:/offline>nmConnect('nodemanager_username','nodemanager_password
, 'ADMINVHN', '5556', 'domain_name', 'ASERVER_HOME', 'PLAIN')
```

```
wls:/nm/domain_name> nmStart('AdminServer')
```

Verify that the Administration Server is working properly by accessing both the Oracle WebLogic Server Administration Console and the Oracle Enterprise Manager Fusion Middleware Control.

## Database Failures: Data Guard Switchover and Failover

The JDBC URL string and ONS configuration provided for the SOA Data Sources should guarantee that reconnection happens automatically when a failover or a switchover takes place at the database level. However, all SOA servers configured with database leasing will shut down themselves if they are not able to update the leasing table. The server goes to FAILED state because a critical subsystem has failed and it is automatically shut down. By default, the time taken for this is variable, depending on when the relevant checks are triggered, but usually takes less than 1 minute.

A database switchover for a SOA system will typically take more than a couple of minutes. During Data Guard switchover or failover of the SOA database, servers will be shutdown automatically and then restarted by the node manager.

The Health Check Interval (default is 10 secs) is the interval to update the leasing table. It can be increased to be more tolerant to database failures, but it will increase also the time it takes to detect server crashes in other scenarios. Two parameters are more suitable to improve the resiliency to the database outages for servers that use database leasing. These parameters are database-leasing-basis-connection-retry-count (1 by default) and database-leasing-basis-connection-retry-delay (1 second by default). Example to change these parameters:

```
./wlst.sh
connect('weblogic','password','t3://ADMINVHN:7001')
edit()
startEdit()
cd('/Clusters/' + 'SOA_Cluster')
cmo.setDatabaseLeasingBasisConnectionRetryCount(10)
cmo.setDatabaseLeasingBasisConnectionRetryDelay(10000L)
save()
activate()
disconnect()
exit()
```

If the product of the retry count \* retry delay is higher than the database outage total time, the servers won't be restarted and they will reconnect to the database once it is up again. Nevertheless, they will have failures in connecting to the database during the time that the database is unavailable.

In case the total number of retries for updating the leasing table are reached and the servers go to FAILED state, they are automatically restarted by the node manager. Two restart are attempted by default, and if the servers can't startup correctly, they will be marked as FAILED\_NOT\_RESTARTABLE. The number of restart attempts performed by the node manager can be tuned as well. In the server's Health Monitoring tab, use the parameter Max Restarts Within Interval to define the number of times that the node manager can restart this server within the interval specified in Restart Interval.

Alternatively, the servers can be gracefully shut down before the switchover operation (does not apply to failover scenarios). Either of the two approaches can be used depending on the load on the system or the business needs.

## Performance and Scalability Implications for an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment

### Capacity Usage and Planning

A Multi Data Center Deployment design needs to account for the throughput penalty introduced by the latency between sites. The worst case scenario will be such that the middle tiers residing in the same site as the active database (Site1 for this example) become unavailable while this database keeps running. In this case, Site2 will have to sustain the load that the two sites were processing together, and it is expected that the average response times will worsen due to the latency because all requests must access a remote database.

The next image shows the transaction throughput degradation (TX/sec) in the system under this situation, for different latencies between sites, compared with the TX/sec value when all the servers are up with the same latency.

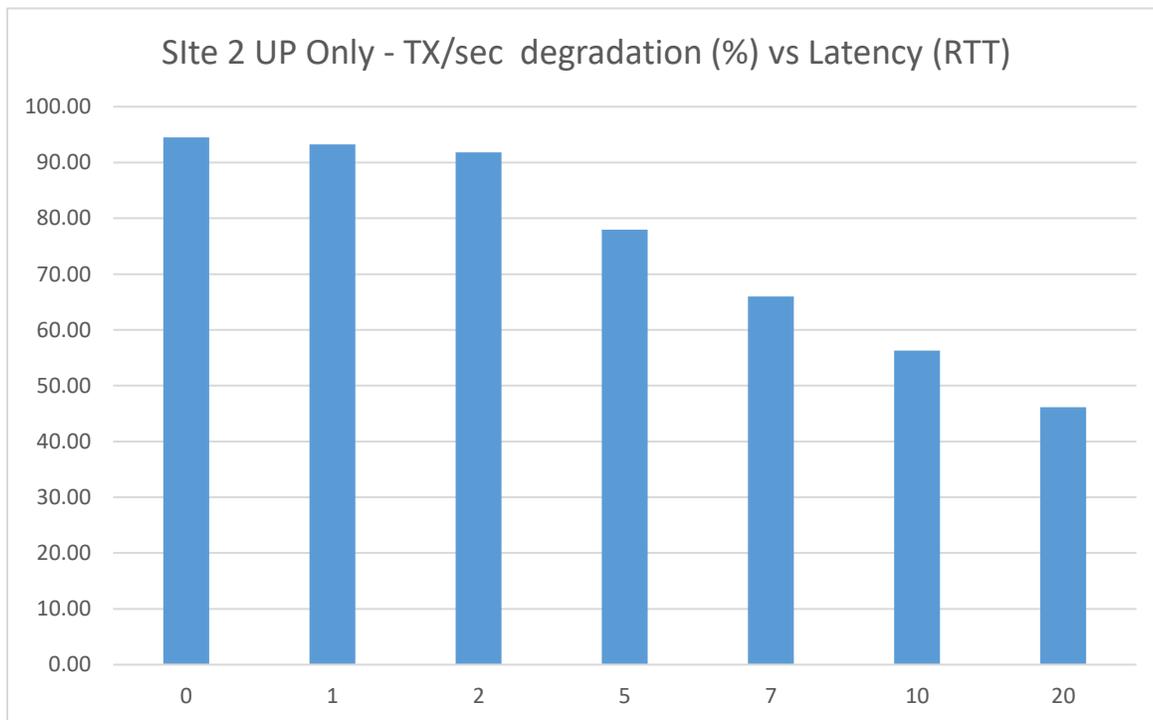


Image 15 Transaction throughput degradation when the Site2 SOA servers only are up and the database resides in the Site1 for different latencies between sites (RTT in ms), compared with the overall throughput of the system (both sites up) with the same latency

For such cases it may be recommended to switch over the database until the middle tiers in Site1 come back online. Site2 will have to have enough spare capacity during normal operation to sustain the added load when a failover from Site1 middle tiers occur. Alternatively, requests will have to be throttled in such a failover mode. The best approach is to throttle requests in the entry points to the system (GLBR or resource adapter access points, such like file system for file/FTP adapters or JMS clients). Requests can be throttled in the GLBR using the appropriate rules

for routing. Similarly, clients producing JMS messages (for JMS adapter), database entries (for database adapter) or Files (to file or FTP adapters) can use the appropriate IP filtering to decrease the load in the system. Defining these rules is out of the scope of this document.

### Start Latencies

A considerable amount of time is spent by servers in creating the connection pools' initial capacity for the different Data Sources used by SOA components. By default most SOA Data Sources use a zero initial capacity for its connection pool. However, to reduce the response time of the system during runtime, it may be advised to increase the initial capacity. For servers that reside in a "remote" site (remote to the SOA database) higher initial pool capacity will cause increased delays in starting the servers. A balanced decision needs to be made between the improved response time during normal operation and the time that it may take a system to recover in order to come up with the ideal initial capacity settings. In stretched clusters, because the initial capacity is set at the Data Source level (connection pool), the settings for initial capacity will affect the start period for all servers in a cluster. Image 15 shows the increased time that it takes to restart a server (compared to a server local to the database) as the latency between the server and the database becomes worse.

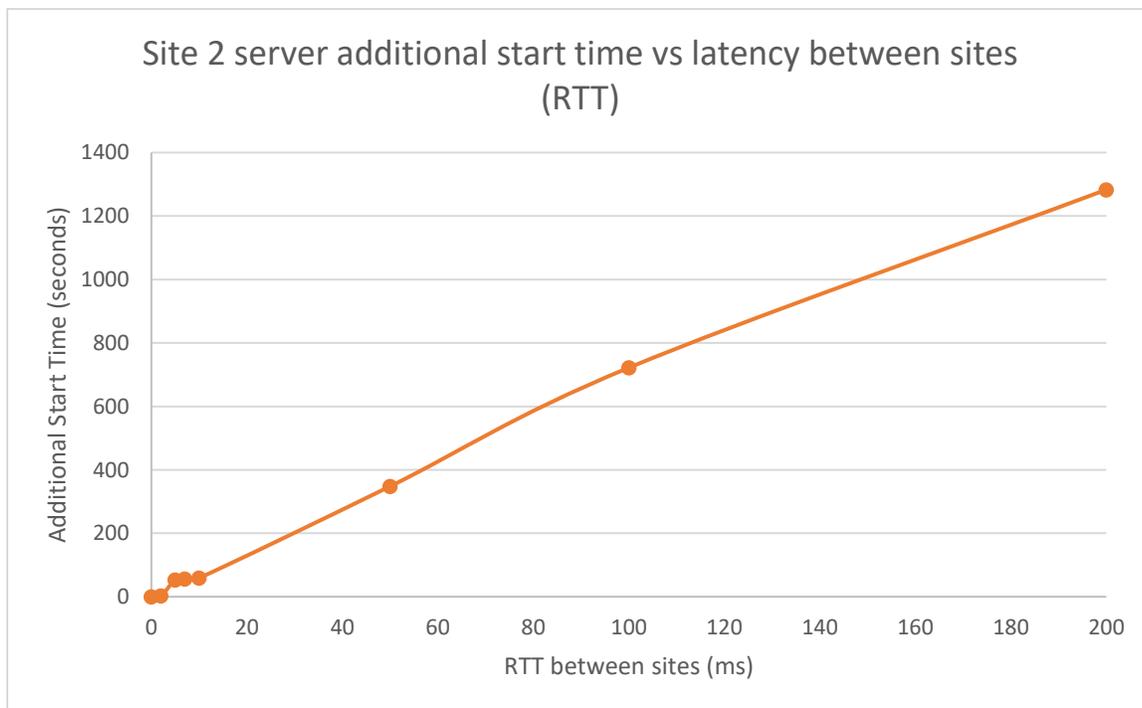


Image 16: Increase in start time (seconds) as latency (RTT in msec) grows for a SOA server using initial capacity=30 for Data Sources)



Site 2 server start time vs latency between sites (RTT)

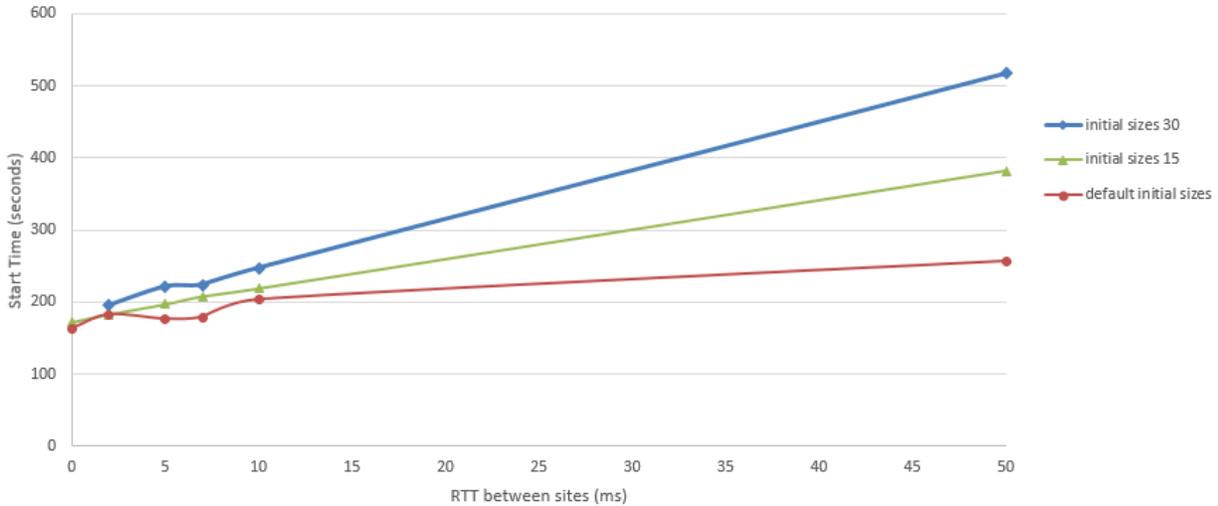


Image 17 Start times (seconds) as latency grows (RTT in msec) for a SOA server in Site2 for different initial size values in all the datasources

% Start Time Penalty for a Site 2 server vs. latency between sites (RTT)

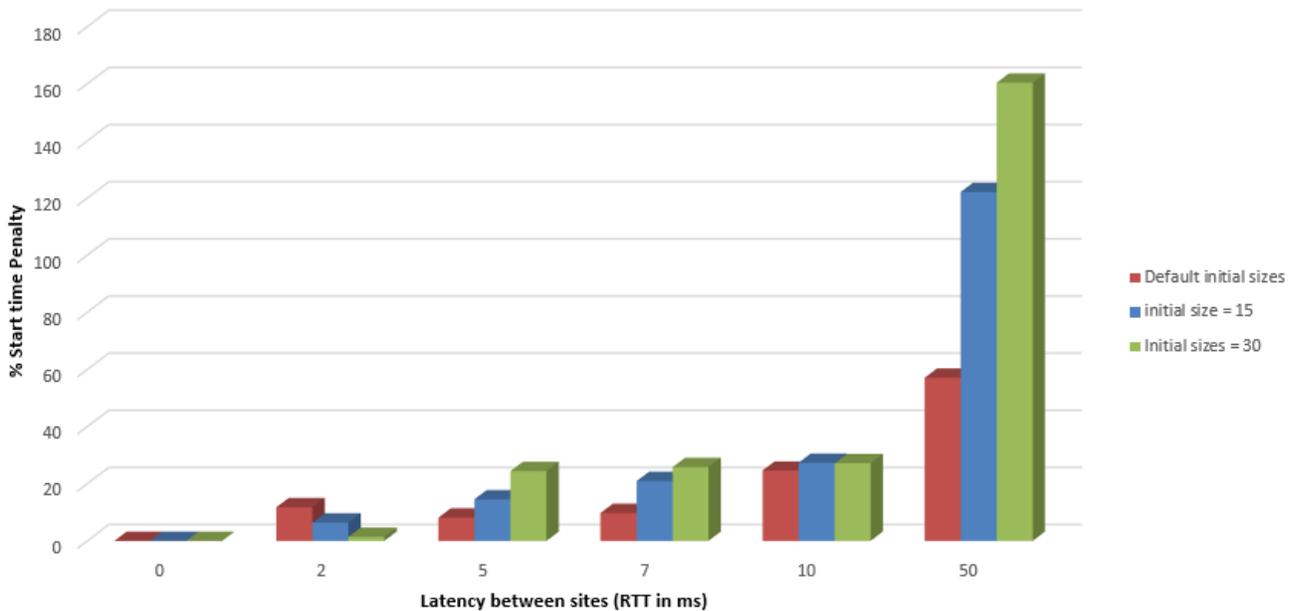


Image 18 Start time penalty (%) for a SOA server in Site2 as latency (RTT in msec) grows, for different initial sizes values in the datasources

The process to restart a SOA server involves bringing up the appropriate Oracle WebLogic Server and also loading and activating the appropriate composites. Once the Oracle WebLogic Server hosting SOA reaches the RUNNING state, nothing prevents OHS from routing to the SOA applications whether composites are available or not.



Starting with SOA 12c, there is a new feature name lazy loading. If requests for a composite arrives before the composite is active, then the HTTP requests are put on hold until the required artifacts are available and the composite reaches the active state. However, composites that include JCA bindings, EJB, and ADF binding cannot be lazy loaded. SOA has been enabled to reject the requests for these composites with a “503 Service unavailable” HTTP code. OHS should automatically retry the request on other available servers and process it successfully (if any other servers have completed the loading of the composite). The following excerpt from an OHS server routing to a SOA server that has not completed loading composites shows the expected behavior.

OHS routes to a SOA Server that is RUNNING but has not loaded the composites yet:

```
[TIMESAMP] [weblogic:debug] [pid 1671:tid 140542142777088]
ApacheProxy.cpp(2570): [client CLIENT_IP:45584]
<005NfNA0CA36qI9_VdP5ic0000Pv00008^> got a pooled connection to server
'SERVER1_IP/8001/8001' from general list for '/soa-
infra/services/soaFusionOrderDemo/OrderBookingComposite/orderprocessor_cl
ient_ep', Local port:13056

[TIMESTAMP] [weblogic:debug] [pid 1671:tid 140542142777088]
BaseProxy.cpp(611): [client CLIENT_IP:45584]
<005NfNA0CA36qI9_VdP5ic0000Pv00008^> Entering method
BaseProxy::sendRequest
```

SOA returns a 503 error because the composite is not loaded yet:

```
[TIMESTAMP] [weblogic:debug] [pid 1671:tid 140542142777088] URL.cpp(850) :
[client CLIENT_IP:45584] <005NfNA0CA36qI9_VdP5ic0000Pv00008^>
URL::parseHeaders: CompleteStatusLine set to [HTTP/1.1 503 Service
Unavailable]

[TIMESTAMP] [weblogic:debug] [pid 1671:tid 140542142777088]
ApacheProxy.cpp(288): [client CLIENT_IP:45584]
<005NfNA0CA36qI9_VdP5ic0000Pv00008^> sendResponse() : r->status = '503'
```

OHS marks the server as unavailable until next proxy plugin update and retries on a different server:

```
[TIMESTAMP] [weblogic:debug] [pid 1671:tid 140542142777088]
BaseProxy.cpp(218): [client CLIENT_IP:45584]
<005NfNA0CA36qI9_VdP5ic0000Pv00008^> Marking SERVER1_IP:8001 as
unavailable for new requests

[TIMESTAMP] [weblogic:error] [pid 1671:tid 140542142777088] [client
CLIENT_IP:45584] <005NfNA0CA36qI9_VdP5ic0000Pv00008^> *****Exception
type [FAILOVER_REQUIRED] (Service Unavailable) raised at line 241 of
BaseProxy.cpp

[TIMESTAMP] [weblogic:debug] [pid 1671:tid 140542142777088]
ap_proxy.cpp(682): [client CLIENT_IP:45584]
<005NfNA0CA36qI9_VdP5ic0000Pv00008^> got exception in sendResponse phase:
FAILOVER_REQUIRED [line 241 of BaseProxy.cpp]: Service Unavailable at
line 682
```

```

[TIMESTAMP] [weblogic:debug] [pid 1671:tid 140542142777088]
ap_proxy.cpp(687): [client CLIENT_IP:45584]
<005NfNA0CA36qI9_VdP5ic0000Pv00008^> Failing over after FAILOVER_REQUIRED
exception in sendResponse()

[TIMESTAMP] [weblogic:error] [pid 1671:tid 140542142777088] [client
CLIENT_IP:45584] ap_proxy: trying POST /soa-
infra/services/soaFusionOrderDemo/OrderBookingComposite/orderprocessor_cl
ient_ep at backend host SERVER1_IP/8001, client CLIENT_IP/45584, total
tries 1; got exception 'FAILOVER_REQUIRED [line 241 of BaseProxy.cpp]:
Service Unavailable'; state: reading status line or response headers from
WLS; failing over

[TIMESTAMP] [weblogic:debug] [pid 1671:tid 140542142777088]
ap_proxy.cpp(556): [client CLIENT_IP:45584]
<005NfNA0CA36qI9_VdP5ic0000Pv00008^> attempt #1 out of a max of 5

[TIMESTAMP] [weblogic:debug] [pid 1671:tid 140542142777088]
ApacheProxy.cpp(2575): [client CLIENT_IP:45584]
<005NfNA0CA36qI9_VdP5ic0000Pv00008^> general list: trying connect to
'SERVER2_IP'/8001/8001 at line 2575 for '/soa-
infra/services/soaFusionOrderDemo/OrderBookingComposite/orderprocessor_cl
ient_ep'

[TIMESTAMP] [weblogic:debug] [pid 1671:tid 140542142777088]
URL.cpp(1805): [client CLIENT_IP:45584]
<005NfNA0CA36qI9_VdP5ic0000Pv00008^> URL::Connect: Connected successfully

```

OHS fails over the request transparently to the client and marks the servers that are loading composites as “bad” (for 10 seconds, the default MaxSkipTime period in the OHS configuration) before retrying that bad server. Because composites are loaded from a remote database, the total amount of time that it takes a SOA server to be effectively available after a restart will be affected mainly by the Data Source pool’s initial capacity (for the server to be available to OHS) and the composite’s sizes (for the SOA system to stop rejecting request with 503 codes).

During the SOA server startup, WebLogic JMS destinations don’t advertise themselves in JNDI until their owning services are ready. When Automatic Service Migration is used, some delay (seconds) is expected until JMS destination are available in the JNDI tree, as the services need to sync up with cluster leasing before they can start. During that interval, JMS Adapter can return jndi resolution errors trying to connect to a JMS resource. If this causes errors in composites, consider to configure Fault handling in the JMS adapter to retry in case of JNDI binding failures.

### Average Active Time for Transactions and Transaction Recovery

In an Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment, transactions remain active for longer periods of time in the site with higher latency to the database. Transactions that remain active for longer periods are more likely to be affected by failures. Image 18 shows the evolution of the average time that transactions remain active for the FOD example as the latency between Site2 and the database is increased.

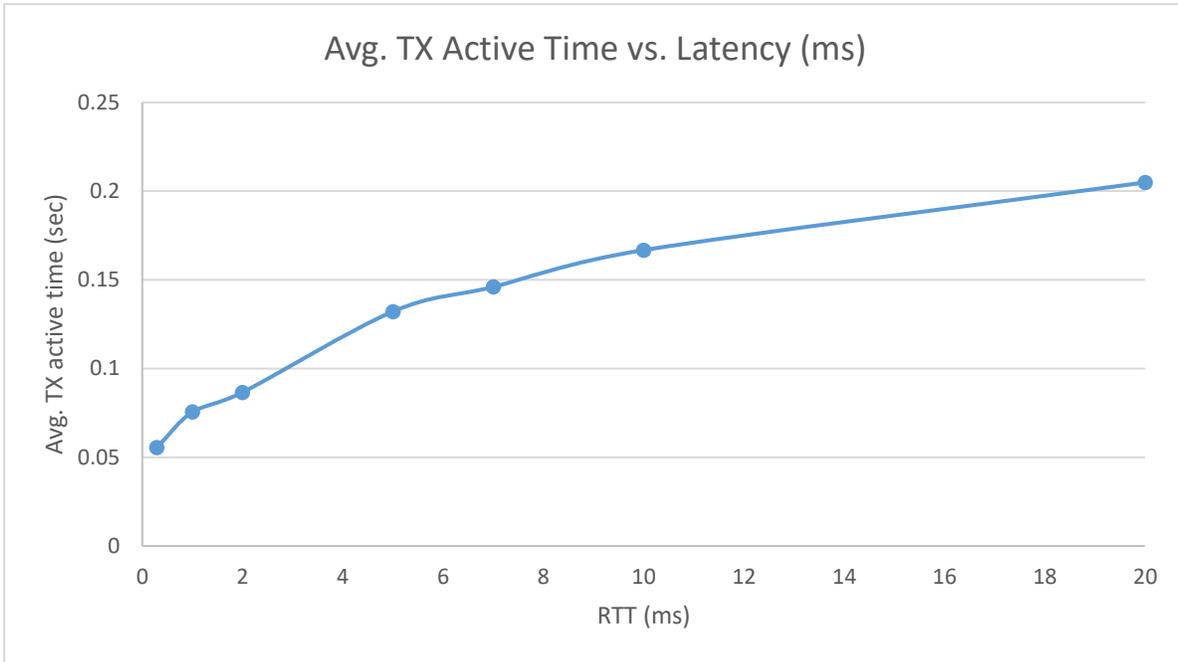


Image 19: Average time that transactions remain active in a Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment with FOD as the latency (RTT in msec) across sites increases

It is important that the appropriate transaction logs protection mechanisms are provided when zero data loss is a business requirement. In these cases, using a JDBC persistent store for JMS and transactions logs is recommended for protection. For single server failures, service migration should take place and recovery will happen automatically. When an entire middle tier in one site is lost, JMS and TLOGS jdbc persistent stores are available for Site2 also so the services can be automatically migrated to the other site. In a Stretched Cluster system the read and resume of logs should be transparent.

When using filestores for JMS and TLOGs, a constant replica of the appropriate persistent stores is need to be set between sites. Otherwise, Automatic Service Migration is prevented from one site to the other, and the services must be manually migrated from Site1 to Site2 after making the transaction logs and persistent stores available in the new site.



## Summary

Oracle Fusion Middleware SOA Suite can be used in multi-datacenter deployments where multiple Oracle Fusion Middleware SOA servers are actively processing requests using a central database located in one of those datacenters. This configuration uses a single Oracle WebLogic Server domain model where servers in all sites participate in the same cluster (also known as Stretched Cluster) and rely on Data Guard to provide protection for the SOA database. The network latency between sites needs to be sufficiently low to overcome the performance penalty introduced by the delay in invocations and to eliminate possible inconsistencies in deployment and runtime scenarios. Oracle recommends using these topologies in Metropolitan Area Networks with latencies between SOA servers and the database below 10 ms (RTT). It is recommended that for each SOA multi data center the deployment, payload, composite types, and throughput requirements are analyzed, and parameters such as timeouts are fine tuned to ensure optimum fail over and performance of the system.

## Appendix A: File Adapter Locks and Muxers

The following script can be used for creating file adapter mutex and lock tables:

```
CREATE TABLE FILEADAPTER_IN
(
 FULL_PATH VARCHAR2(4000) NOT NULL,
 ROOT_DIRECTORY VARCHAR2(3000) NOT NULL,
 FILE_DIRECTORY VARCHAR2(3000) NOT NULL,
 FILE_NAME VARCHAR2(1000) NOT NULL,
 FILE_ENDPOINT_GUID VARCHAR2(2000) NOT NULL,
 FILE_LAST_MODIFIED NUMBER,
 FILE_READONLY CHAR(1),
 FILE_PROCESSED CHAR(1) DEFAULT '0',
 CREATED NUMBER NOT NULL,
 UPDATED NUMBER ,
 TENANT_ID NUMBER(18,0) DEFAULT -1
);
ALTER TABLE FILEADAPTER_IN ADD CONSTRAINT FILEADAPTER_IN_PK PRIMARY KEY
(FULL_PATH);
CREATE INDEX IDX_ROOT_DIRECTORY ON FILEADAPTER_IN (ROOT_DIRECTORY);
CREATE INDEX IDX_FILE_DIRECTORY ON FILEADAPTER_IN (FILE_DIRECTORY);
CREATE INDEX IDX_FILE_PROCESSED ON FILEADAPTER_IN (FILE_PROCESSED);
CREATE INDEX IDX_FILE_READONLY ON FILEADAPTER_IN (FILE_READONLY);

-- FILEADAPTER_MUTEX

CREATE TABLE FILEADAPTER_MUTEX
(
 MUTEX_ID VARCHAR2(4000) NOT NULL,
 MUTEX_CREATED TIMESTAMP,
 MUTEX_LAST_UPDATED TIMESTAMP,
 MUTEX_SEQUENCE NUMBER ,
 TENANT_ID NUMBER(18,0) DEFAULT -1
);
ALTER TABLE FILEADAPTER_MUTEX ADD CONSTRAINT FILEADAPTER_MUTEX_PK
PRIMARY KEY (MUTEX_ID);
```

## Appendix B: Configuring in-place restart for JMS JDBC persistent stores

To configure in-place restart for JMS JDBC persistent stores, the pertaining JMS server and persistent store must be targeted to a migratable target and this migratable target must use “Restart on Failure”. To configure this, follow these steps:

### » Enable restart on failure for migratable targets:

1. Log in to the WLS Administration Console
2. Click Lock and Edit
3. On the navigation tree on the left, expand Environment and click on Migratable Targets
4. Click on WLS\_SOA1 (migratable)
5. Click on the Migration Tab
6. Check the “Restart On Failure” box at the bottom

The screenshot shows the 'Migration' tab in the WLS Administration Console. The 'Name' field is 'WLS\_SOA1 (migratable)'. The 'Service Migration Policy' is set to 'Manual Service Migration Only'. The 'User-Preferred Server' is 'WLS\_SOA1'. Under 'Constrained Candidate Servers', the 'Available' list contains 'WLS\_SOA1', 'WLS\_SOA2', 'WLS\_SOA3', and 'WLS\_SOA4'. The 'Chosen' list is empty. The 'Pre-Migration Script Path' and 'Post-Migration Script Path' fields are empty. The 'Post-Migration Script Failure Cancels Automatic Migration' checkbox is checked. The 'Allow Post-Migration Script To Run On a Different Machine' checkbox is unchecked. The 'Restart On Failure' checkbox is checked.

Image 20: Migration configuration

7. Click Save
8. Repeat steps 4-7 for all the SOA JMS servers using the appropriate migratable targets (WLS\_SOA1 (migratable), WLS\_SOA2 (migratable) etc depending on the WLS server that the JMS servers is related

to)

9. Activate the changes

» **Re-target the JMS Servers and Persistent Stores:**

In a SOA 12.2.1.3 environment, this configuration is already implemented if the option “Enable Automatic Service Migration” was selected during the domain configuration. For any other case where Automatic Service Migration needs to be configured manually as a post step, follow these instructions to target the persistent stores to the appropriate migratable target:

1. Log in to the WLS Administration Console
2. Click Lock and Edit
3. On the navigation tree on the left, expand Services->Messaging and click on JMS Servers
4. On the JMS Servers table, Click on SOAJMSServer\_auto\_1
5. Select the Targets Tab
6. Select “WLS\_SOA1 (migratable)” as target

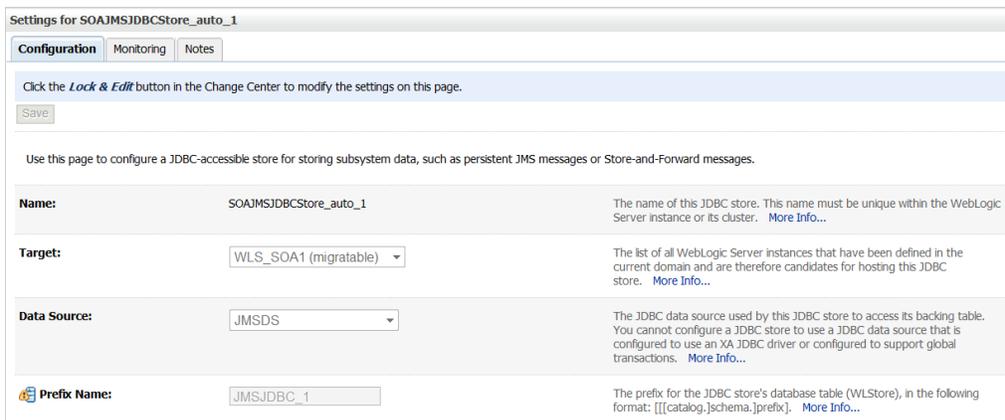


Image 21: Migratable target configuration

7. Click Save (you may ignore the error about the JMS server or SAF agent SOAJMSServer\_auto\_1 not being targeted to the same target as its persistent store)
8. Repeat steps 4-7 for all JMS servers using JDBC persistent stores
9. On the navigation tree on the left, expand Services and click on Persistent Stores
10. Click on the persistent store associated to the SOAJMSServer\_auto\_1 JMS server (the persistent store can be determined from the Services->Messaging ->JMS Servers table)
11. Select “WLS\_SOA1 (migratable)” as target
12. Click Save
13. Repeat steps 10-12 for all the JMS JDBC persistent stores using the appropriate migratable targets (WLS\_SOA1 (migratable), WLS\_SOA2 (migratable), etc.)

14. Activate the Changes
15. Restart the SOA servers

Once in place restart is configured, the behavior of the JMS JDBC persistent store can be verified setting the “-Dweblogic.debug.DebugSingletonServices=true” and “-Dweblogic.StdoutDebugEnabled=true” start properties for the servers. Upon a database failure, the following events should be logged in the server’s out file:

**Add store to migratable target:**

```
<Debug> <SingletonServices> <BEA-000000> <MigratableGroup: adding migratable SOAJMSJDBCStore_auto_1 to group WLS_SOA1 (migratable) Migratable class - weblogic.management.utils.GenericManagedService$ManagedDeployment>
```

**Reach out to DB:**

```
<Info> <Store> <BEA-280071> <JDBC store "SOAJMSJDBCStore_auto_1" opened table "JMSJDBC_1WLStore" and loaded 0 records. For additional JDBC store information, use the diagnostics framework while the JDBC store is open.>
```

**Crash of DB:**

```
<Emergency> <Store> <BEA-280060> <The persistent store "SOAJMSJDBCStore_auto_1" encountered a fatal error, and it must be shut down: weblogic.store.PersistentStoreFatalException: [Store:280065]java.sql.SQLException: Connection has been administratively destroyed. Reconnect. (server="WLS_SOA1" store="SOAJMSJDBCStore_auto_1" table="JMSJDBC_1WLStore");(Linked Cause, "java.sql.SQLException: Connection has been administratively destroyed. Reconnect.")>
```

**Failed Store:**

```
<Error> <Store> <BEA-280074> <The persistent store "SOAJMSJDBCStore_auto_1" encountered an unresolvable failure while processing transaction "BEA1-148701EFAE50DC04072E". Shutdown and restart to resolve this transaction. weblogic.store.gxa.GXAException: weblogic.store.PersistentStoreException: weblogic.store.PersistentStoreFatalException: [Store:280032]The persistent store suffered a fatal error and it must be re-opened>
```

**Deactivate Store:**

```
<Debug> <SingletonServices> <BEA-000000> <MigratableGroup: Going to call migratableDeactivate on SOAJMSJDBCStore_auto_1 for WLS_SOA1 (migratable)>
```

**Reactivate Store:**

```
<Debug> <SingletonServices> <BEA-000000> <MigratableGroup: activating migratable 'SOAJMSJDBCStore_auto_1' for WLS_SOA1 (migratable)>
```

**Ownership lock on table and failure to activate (first try):**

```
<Warning> <Store> <BEA-280076> <Database table "JMSJDBC_1WLStore" for store "SOAJMSJDBCStore_auto_1" is currently owned by "[name={server=WLS_SOA1!host=192.168.48.58!domain=soaexa_domain!store=SOAJMSJDBCStore_auto_1!table=JMSJDBC_1WLStore};random=5074588073335957339;timestamp=1401991210748]". Trying to wait for ownership.>
```

**New try to activate Persistent/ Store:**

```
<Debug> <SingletonServices> <BEA-000000> <MigratableGroup: activating migratable 'SOAJMSJDBCStore_auto_1' for WLS_SOA1 (migratable)>
```

**Successful activation of store:**

```
<Info> <Store> <BEA-280071> <JDBC store "SOAJMSJDBCStore_auto_1" opened table "JMSJDBC_1WLStore" and loaded SOAJMSJDBCStore_auto_1 records. For additional JDBC store information, use the diagnostics framework while the JDBC store is open.>
```

To reduce logging overhead, remove the “-Dweblogic.debug.DebugSingletonServices=true” and “-Dweblogic.StdoutDebugEnabled=true” flags from the server’s start properties once the correct behavior is verified.

## Appendix D: Oracle Service Bus Considerations

All the recommendations and configuration details described in this document for configuring SOA in a multidatcenter Active-Active deployment are valid for OSB as well: Cluster Configuration, Automatic Service Migration configuration, Transaction and Persistence logs configuration, Replicated services or JMS Destinations are applicable to an OSB Cluster also as they are explained in the previous chapters.

There are some specific considerations for Oracle Service Bus that are included below:

### Load balancer considerations

The virtual server created for OSB must follow the same recommendations provided in “Configuring Load Balancers and Global Load Balancers for Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment” section.

For the OSB virtual server, it is recommended that you set the LBRs monitor to the WSIL inspection URL /sbinspection.wsil to determine the availability of an OSB server, to eliminate undesired routings when OSB WebLogic Server are not available.

### Application Tier considerations

#### OSB Singleton services

FTP, File and Mail pollers are OSB services that run only in one server in the cluster. When the global property “OSB Singleton Components Automatic Migration” is enabled, these services act as singleton and they automatically migrate to another server in the cluster when there is a failure or a server shutdown. This server can be in Site1 or Site2. Resources that are polled by these services must be ideally reachable from both sites, or at minimum from the site where the singleton service is running. If the resource is an FTP or a Mail server, they must be accessible regardless the singleton is running in Site1 or in Site2. In the Stretched cluster topology there is no shared storage between sites, so if the resource is a file system folder, it must be available from the Site where the service is running to be processed. Continuous replication of this storage needs to occur to the other site for consistency processing of files.

There is no way to manually migrate these services, so it is important to be able to determine in which site are they activated. The OSB server that activates these singleton services has these kind of messages in the log:

```
<TIMESTAMP> <Info> <Cluster> <HOSTNAME> <SERVER_NAME> <[ACTIVE]
ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-tuning) '>
<<WLS Kernel>> <> <2108ed22-dbea-46d4-8a25-2164f202fa6a-00000397>
<1497264860902> <[severity-value: 64] [rid: 0] [partition-id: 0]
[partition-name: DOMAIN] > <BEA-003130> <default-TransportPollers-
MyFilePollerPS successfully activated on server SERVER_NAME.>

<TIMESTAMP> <Info> <Cluster> <HOST_NAME> <SERVER_NAME> <[STANDBY]
ExecuteThread: '17' for queue: 'weblogic.kernel.Default (self-tuning) '>
<<WLS Kernel>> <> <36c0d769-2602-4b0c-aa4c-2ab5d217ae67-000000db>
<1510130044907> <[severity-value: 64] [rid: 0] [partition-id: 0]
[partition-name: DOMAIN] > <BEA-003130>
<Appscoped_Singleton_Service_Initializer successfully activated on server
SERVER_NAME.>
```

Alternatively, the database leasing table can be checked to determine which server is running the OSB singletons.  
Example:

```
SQL> select server, instance from ACTIVE;
```

| SERVER                                                 | INSTANCE                             |
|--------------------------------------------------------|--------------------------------------|
| service.WLS_OSB2                                       | 2589091175644382971/WLS_OSB2         |
| service.WLS_OSB1                                       | -3338612690579104383/WLS_OSB1        |
| service.WLS_OSB3 (migratable)                          | 8011168529714146120/WLS_OSB3         |
| service.WLS_OSB1 (migratable)                          | -3338612690579104383/WLS_OSB1        |
| <b>service.Appscoped_Singleton_Service_Initializer</b> | <b>-4188121437114911506/WLS_OSB4</b> |
| service.WLS_OSB4 (migratable)                          | -4188121437114911506/WLS_OSB4        |
| service.SINGLETON_MASTER                               | -4188121437114911506/WLS_OSB4        |
| service.WLS_OSB4                                       | -4188121437114911506/WLS_OSB4        |
| service.WLS_OSB2 (migratable)                          | 2589091175644382971/WLS_OSB2         |
| service.WLS_OSB3                                       | 8011168529714146120/WLS_OSB3         |

The Aggregator Server is also a singleton service that runs only in one server in the cluster, and automatically fails over another server where there is a failure or a server shutdown. No special considerations need to be applied in this case.

### Configuring Datasources

The Data Sources used by the OSB Clustert should be configured to automate failover of connections in case there is failover or switchover of the active database, as it was explained in the section “Configuring Data Sources for Oracle Fusion Middleware SOA Multi Data Center Active-Active Deployment. The following Data Sources need to be configured properly to automate this failover in a OSB domain:

- » LocalSvcTblDataSource
- » mds-owsm
- » opss-audit-DBDS
- » opss-audit-viewDS
- » opss-data-source
- » OraSDPMDDataSource
- » SOADDataSource
- » wlsbjmsrpDataSource
- » WLSSchemaDataSource

Any other custom datasource should be also configured to automate failover.

### OSB Performance in a Stretched Cluster

In the Stretched cluster topology, OSB's performance degrades less than SOA's when latency between sites increases. The reason is that OSB's use of the database is much less intensive than SOA's, so the latency between Site2 OSB servers and the database is not such a relevant bottleneck. The following image shows a comparison of the traffic from Site2 to Site1 in 3 different scenarios: SOA cluster running FOD application, OSB cluster running a web service routing based scenario and OSB cluster running a JCA DBAdapter based scenario.

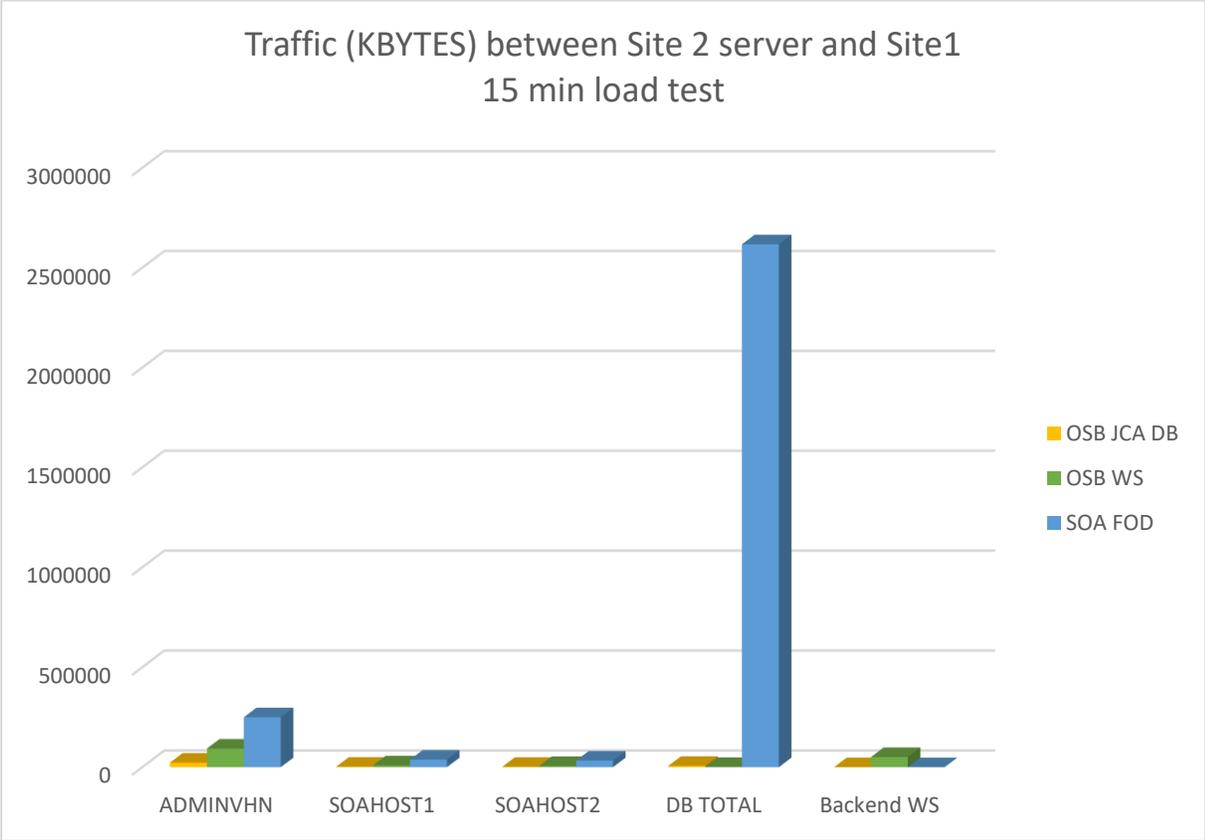


Image 22 Comparison of the total traffic between a middle tier server in Site2 and the hosts in Site1 for different scenarios: SOA running FOD, OSB running a web service routing based scenario, and OSB running a JCA DB based scenario.

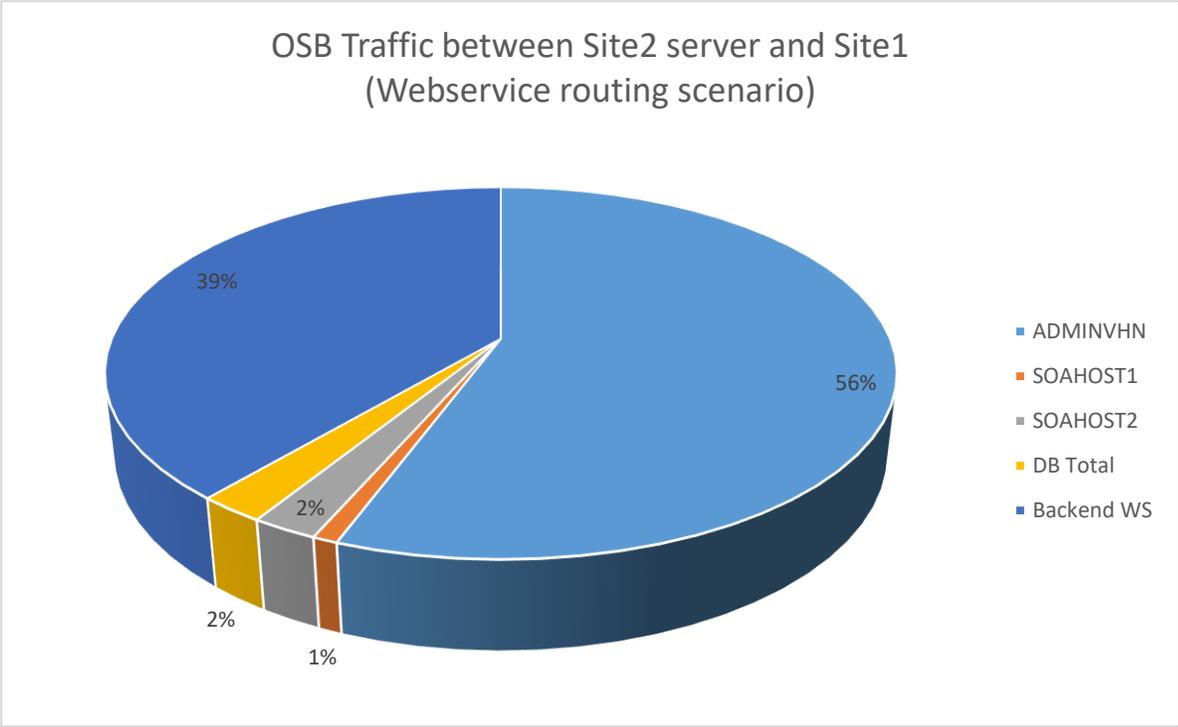


Image 23 Traffic distribution between a server in Site2 and the rest of the servers in Site1 for OSB running a scenario based on web service routing

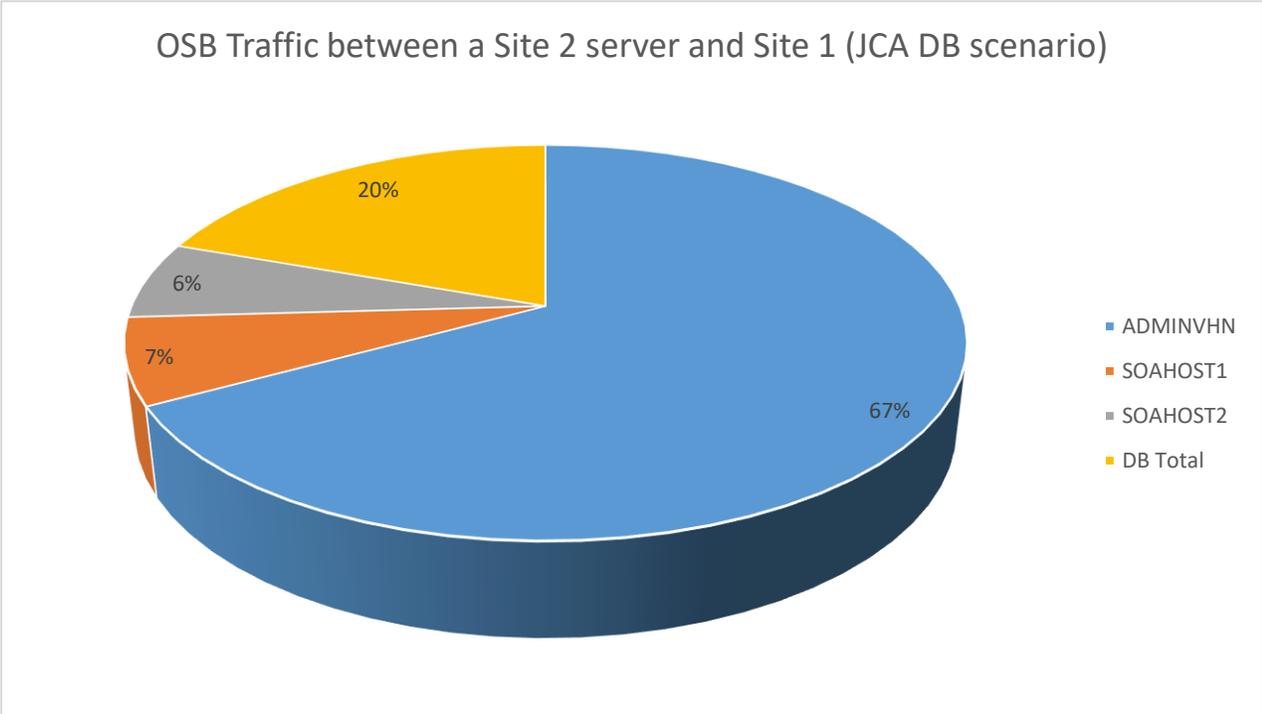


Image 24 Traffic distribution between a OSB server in Site2 and the rest of the servers in Site1 running an scenario based on a JCA DB adapter

With this in mind, the performance degradation for OSB non-transactional scenarios based on pure routing web services is almost neglectable.

However, the latency between sites does not only affect the communication between middle-tiers and the database. OSB backend endpoints (databases or web services) can incur in additional latency when accessing servers in the other site. When this happens, the effect is reflected directly in the Proxy Services' avg. response times:

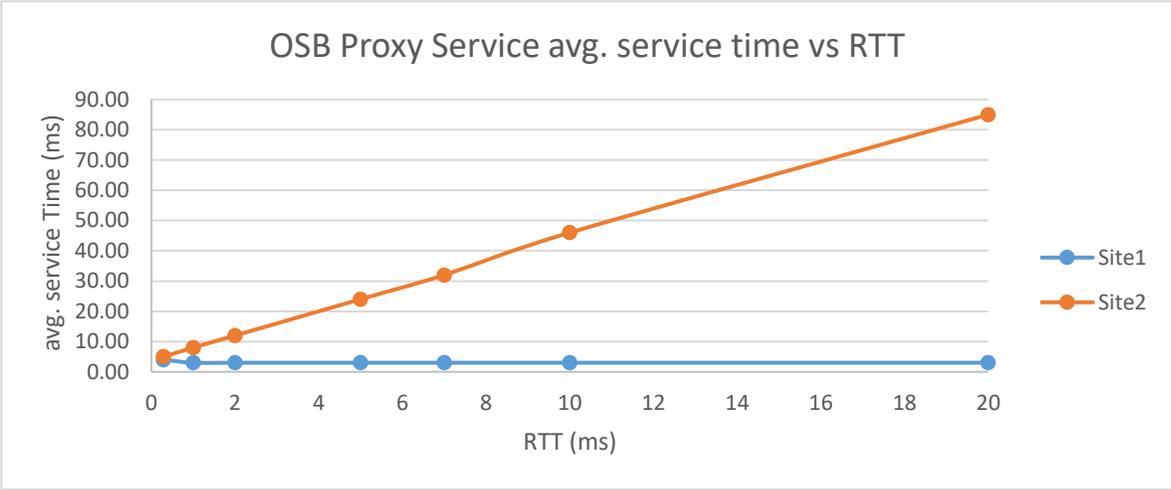


Image 25 Average proxy service time for each site for different latencies, when OSB is running a JCA DBAdapter scenario and DB is in the other Site

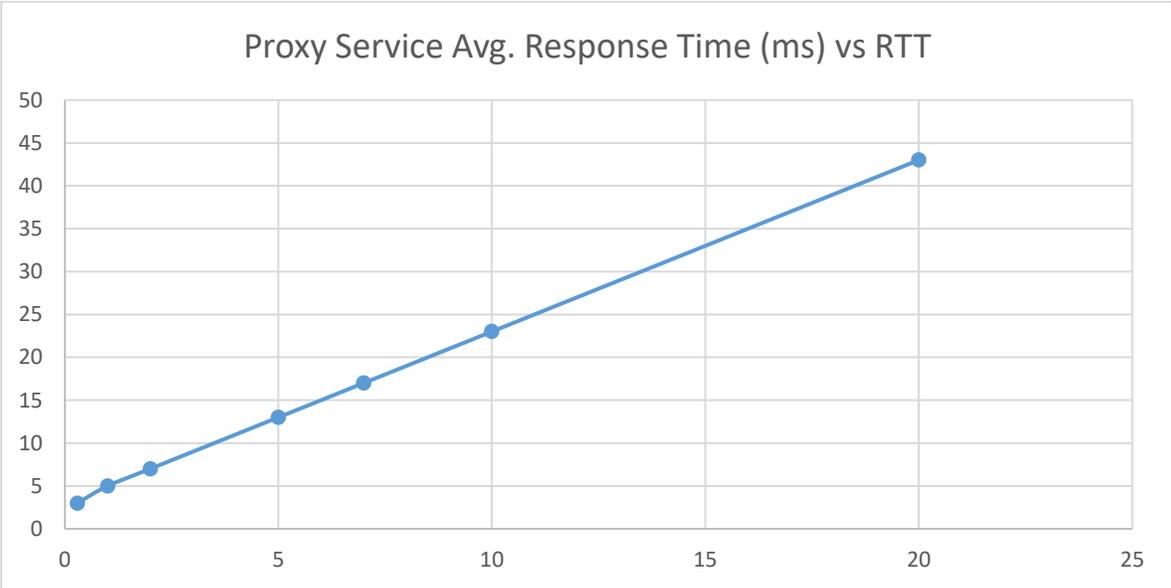


Image 26 Average proxy service time for Site2 server, when OSB is running a web service routing scenario and the backend web services are in the other Site

The following image shows the throughput (number of proxy services request/sec processed by the system (both sites up) for a scenario based in a JCA DB adapter insertion.

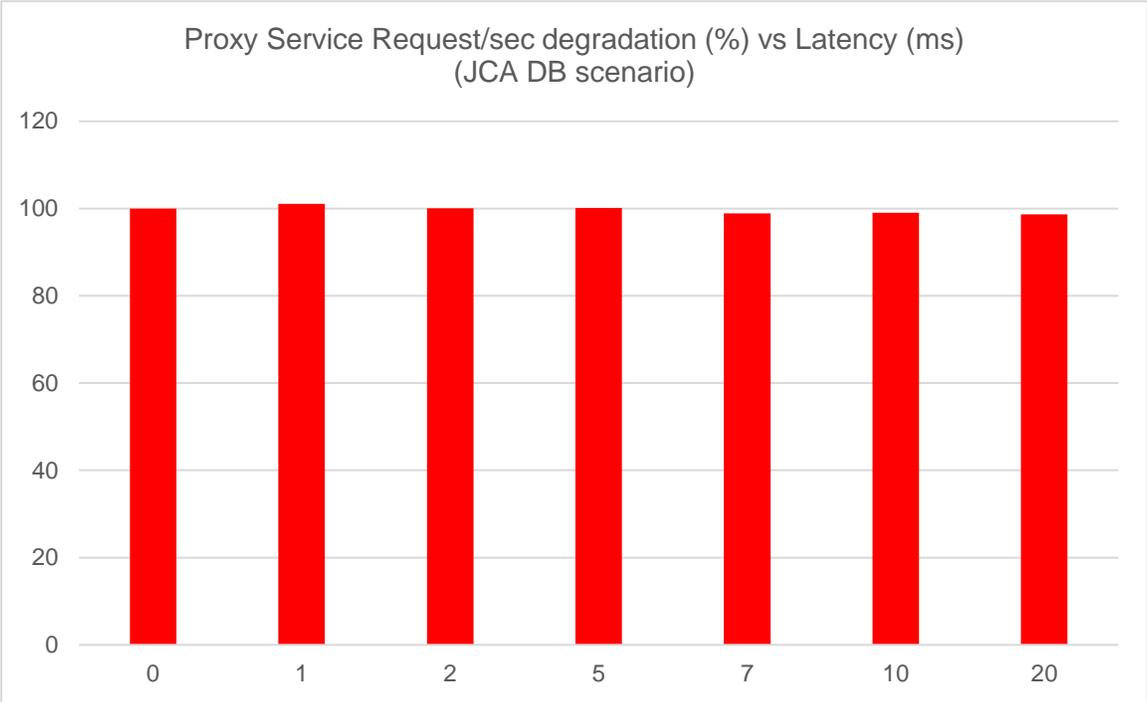


Image 27 Proxy Service request/sec degradation (%) for different latencies (RTT in ms) between sites, for a scenario based in a JCA DBAdapter

Depending on the protocols used by the Proxy and Business services and the location of the endpoints, the load and the size of the payloads, the performance penalty can vary, but in general the performance degradation will be lower than for a SOA system with the same latency between sites.

## References

1. *Best Practices for Oracle Fusion Middleware SOA 11g Multi Data Center Active-Active Deployment*  
<http://www.oracle.com/technetwork/database/availability/fmw11gsoamultidc-aa-1998491.pdf>
2. *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*  
<https://docs.oracle.com/middleware/12213/lcm/SOEDG/toc.htm>
3. *Fusion Middleware Administering Oracle SOA Suite and Oracle Business Process Management Suite*  
<https://docs.oracle.com/middleware/12213/soasuite/administer/toc.htm>
4. *Fusion Middleware Developing SOA Applications with Oracle SOA Suite*  
<https://docs.oracle.com/middleware/12213/soasuite/develop/toc.htm>
5. *Fusion Middleware Administering Oracle Service Bus*  
<https://docs.oracle.com/middleware/12213/osb/administer/toc.htm>
6. *Fusion Middleware Developing Services with Oracle Service Bus*  
<https://docs.oracle.com/middleware/12213/osb/develop/toc.htm>
7. *Oracle Fusion Middleware User's Guide for Technology Adapters*  
[http://docs.oracle.com/cd/E23943\\_01/integration.1111/e10231/toc.htm](http://docs.oracle.com/cd/E23943_01/integration.1111/e10231/toc.htm)
8. *Oracle Fusion Middleware Understanding Technology Adapters*  
<https://docs.oracle.com/middleware/12213/adapters/develop-soa-adapters/TKADP.pdf>
9. *F5's Big IP Global Traffic Manager Documentation*  
[http://support.f5.com/kb/en-us/products/big-ip\\_gtm.html](http://support.f5.com/kb/en-us/products/big-ip_gtm.html)
10. *Oracle WebLogic Server and Highly Available Oracle Databases: Oracle Integrated Maximum Availability Solutions*  
<http://www.oracle.com/technetwork/database/features/availability/wlsdatasourcefordataguard-1534212.pdf>
11. *Best Practices for Active-Active Fusion Middleware: Oracle WebCenter Portal*  
<http://www.oracle.com/technetwork/database/availability/webcenteractiveactive-1621358.pdf>
12. *Fusion Middleware Developing Applications with Oracle Coherence*  
<https://docs.oracle.com/middleware/12213/coherence/develop-applications/toc.htm>



---

CONNECT WITH US

-  [blogs.oracle.com/oracle](https://blogs.oracle.com/oracle)
-  [facebook.com/oracle](https://facebook.com/oracle)
-  [twitter.com/oracle](https://twitter.com/oracle)
-  [oracle.com](https://oracle.com)

## Integrated Cloud Applications & Platform Services

Copyright © 2017, Oracle and/or its affiliates. All rights reserved. This document is provided *for* information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0615

Best Practices for Oracle Fusion Middleware SOA 12c Multi Data Center Active-Active Deployment  
Updated June 2023



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