FUSION MIDDLEWARE

An Oracle White Paper
September 2014

Oracle WebLogic Server Integration with Oracle Database 12*c*

Disclaimer

Introduction

Oracle WebLogic Server 11g introduced Active GridLink for Real Application Clusters (RAC). In conjunction with Oracle Database, this powerful software technology simplifies management, increases availability, and ensures fast connection failover, runtime connection load balancing and affinity capabilities.

Tight integration between Oracle WebLogic Server 12c (12.1.2) and Oracle Database 12c enhances these capabilities with improved availability, better resource sharing, inherent scalability, ease of configuration and automated management facilities in a global cloud environment. Oracle WebLogic Server is the only application server with this degree of integration to Oracle Database 12c. All content covered in this paper applies not only to Oracle WebLogic Server 12.1.2 but also to Oracle WebLogic Server 12.1.3.

This white paper explains how this unique database, clustering, and application server technologies work together to enable higher availability, scalability and performance for your business. We start by introducing Oracle Active GridLink for RAC with attention to ease of configuration, manageability, and performance. Then we describe the impact of Oracle WebLogic server on several leading features of Oracle Database12c such as Multitenant Database (PDB), Database Resident Connection Pool (DRCP), Application Continuity, and Global Data Services (GDS).

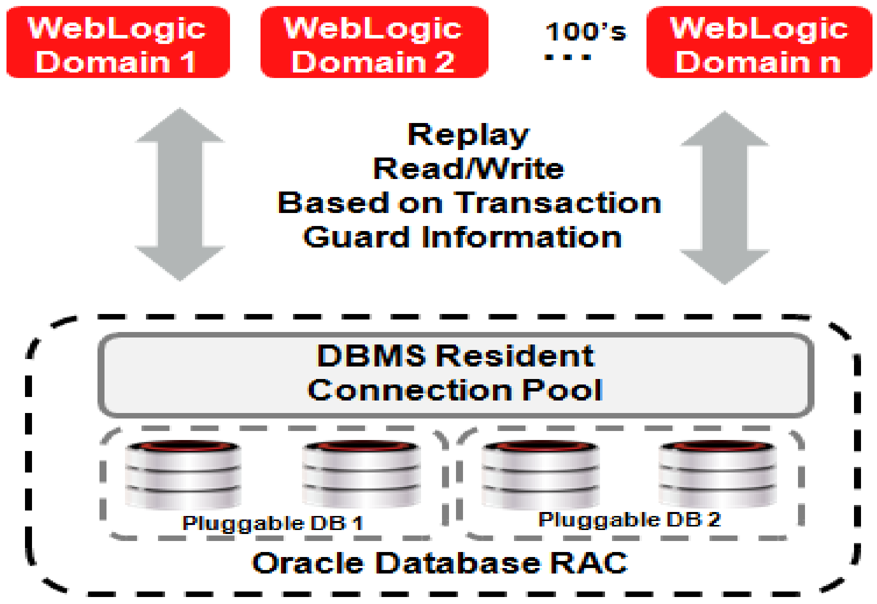


Figure 1: A visual depiction of how Oracle WebLogic Server 12c integrates with Oracle Database 12c

Active GridLink for RAC

Oracle WebLogic Server 11g (10.3.4) introduced a single data source implementation to support Oracle RAC called Active GridLink for RAC. This data source responds to Fast Application Notification (FAN) events to provide Fast Connection Failover (FCF), Runtime Connection Load-Balancing (RCLB), and graceful shutdown of RAC instances. XA affinity is supported at the global transaction ID level. Active GridLink for RAC is implemented as the GridLink data source within Oracle WebLogic Server.

Through deeper integration with Oracle RAC, this single data source implementation in Oracle WebLogic Server supports the full and unrestricted use of database services as the connection target for a data source. Active management of the connections in the pool is based on static settings configured on the connection pool itself (min/max capacity, timeouts, etc.) and real time information that the connection pool receives from the RAC Oracle Notification Service (ONS) subsystem, which advises the client of any state changes within the RAC cluster.

The Universal Connection Pool (UCP) Java library has been integrated with Oracle WebLogic Server. It is utilized by the WebLogic GridLink data source to enable Fast Connection Failover, Runtime Connection Load Balancing and Affinity features.

Simplified Configuration of Oracle RAC and WebLogic Server

This new implementation simplifies the integration of Oracle RAC database with Oracle WebLogic Server through the GridLink data source approach, which in turn reduces the configuration and management complexity required to use Oracle RAC. Oracle also supports WebLogic Multi data source configurations for RAC environments. Upgrades from WebLogic Multi data sources to Grid Link data sources are straightforward. They simply involve creating a single Grid Link data source with the same JNDI name as the Multi data source. This approach improved throughput while reducing the number of configuration artifacts to maintain.

Active GridLink: Summary of Key Features

WebLogic GridLink data source has been integrated with Fast Connection Failover from the Universal Connection Pool to:

- Rapidly Detect Failures
- Abort and remove invalid connections from the connection pool
- Perform graceful shutdown for planned and unplanned Oracle RAC node outages
- Adapt to changes in topology, such as adding or removing nodes
- Distribute runtime work requests to all active Oracle RAC instances, including those rejoining a cluster

WebLogic GridLink data sources and JDBC connection pools leverage the runtime connection load balancing functionality provided by an Oracle RAC database through FAN notifications to improve throughput and more efficiently use resources.

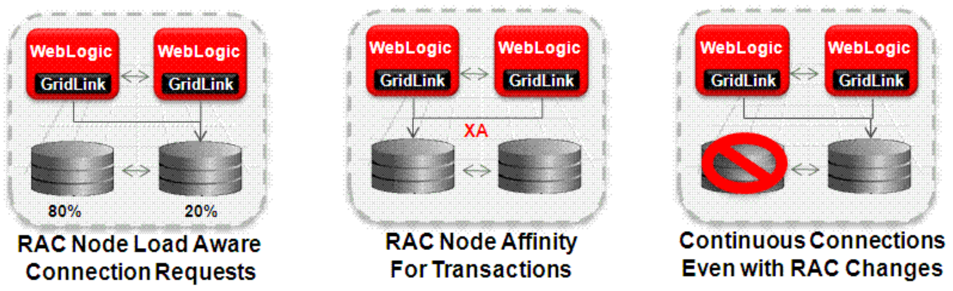

WebLogic
GridLink
WebLogic
GridLink
80%
20%
RAC Node Load Aware Connection Requests
WebLogic
GridLink
WebLogic
GridLink
XA
RAC Node Affinity For Transactions
WebLogic
GridLink
WebLogic
GridLink
Continuous Connections Even with RAC Changes

Figure 2: Active GridLink for RAC Load Balancing/XA Affinity/Failover

In Oracle WebLogic Server 11g (10.3.6), Active GridLink for RAC supports Oracle Database 11g features such as Application Continuity for read-only operations and WebSession Affinity.

Application Continuity is a general purpose, application-independent infrastructure that recovers work in the event of an outage and masks many system, communication, and hardware failures and hangs. This version only supports read-only operations.

Web Session Affinity is a new policy option for GridLink data sources that directs database operations made under a web session to the same RAC instance. This technique improves database utilization and boosts overall application performance.

Oracle WebLogic Server Integration with Oracle Database 12c

Delivering an integrated technology platform has always been a central component of Oracle's overall strategy. To that end, Oracle WebLogic Server 12c (12.1.2) supports and has been fully certified with Oracle Database 12c, particularly with respect to Application Continuity, Transaction Guard, Database Resident Connection Pool, Multitenant Database, and Global Data Services. Oracle WebLogic Server is the only application server in the market that provides this level of the integration with Oracle Database. Table 1 shows which capabilities of Oracle Database 12c have been integrated with Oracle WebLogic Server 12.1.3, 12.1.2, 12.1.1, and 10.3.6.

DATABASE FEATURE	WLS 10.3.6/12.1.1/ 12.1.2/12.1.3 11G DRIVER 11GR2 DB	WLS 10.3.6/12.1.1/ 12.1.2/12.1.3 11G DRIVER 12C DB	WLS 10.3.6/12.1.1 12C DRIVER 11GR2 DB	WLS 12.1.2/ 12.1.3 12C DRIVER 11GR2 DB	WLS 10.3.6/12.1.1 12C DRIVER 12C DB	WLS 12.1.2/ 12.1.3 12C DRIVER 12C DB
JDBC Replay (read/write)	No	No	No	No	Yes (read/write with Active GridLink no XA)	Yes (read/write with Generic data sources and Active GridLink no XA)
Multitenant Database (PDB)	No	Yes	No	No	Yes	Yes
Dynamic Switching between PDBs	No	No	No	No	No	Yes (no XA)
Database Resident Connection Pooling (DRCP)	No	No	No	Yes	No	Yes
Oracle Notification Service (ONS) Auto configuration	No	No	No	No	No	Yes (Active GridLink Only)
Global Database Services (GDS)	No	Yes (Active GridLink Only)	No	No	Yes (Active GridLink Only)	Yes (Active GridLink Only)
JDBC 4.1 (ojdbc7.jar files and JDK 7)	No	No	Yes	Yes	Yes	Yes

Table 1: Oracle WebLogic Server Certification with Oracle Database 12c

Application Continuity and Transaction Guard

As the key offering in Oracle Database 12c, Application Continuity is a general purpose, application-independent software utility that recovers work during an outage and masks system, communication, and hardware failures from users. Oracle Database is the first database management system to provide a generic infrastructure for *at-most once execution semantics* in case of failures. This type of execution means that the operation must execute once, partially, or not at all. For example, adding or deleting an appointment from a calendar generally uses at-most-once semantics.

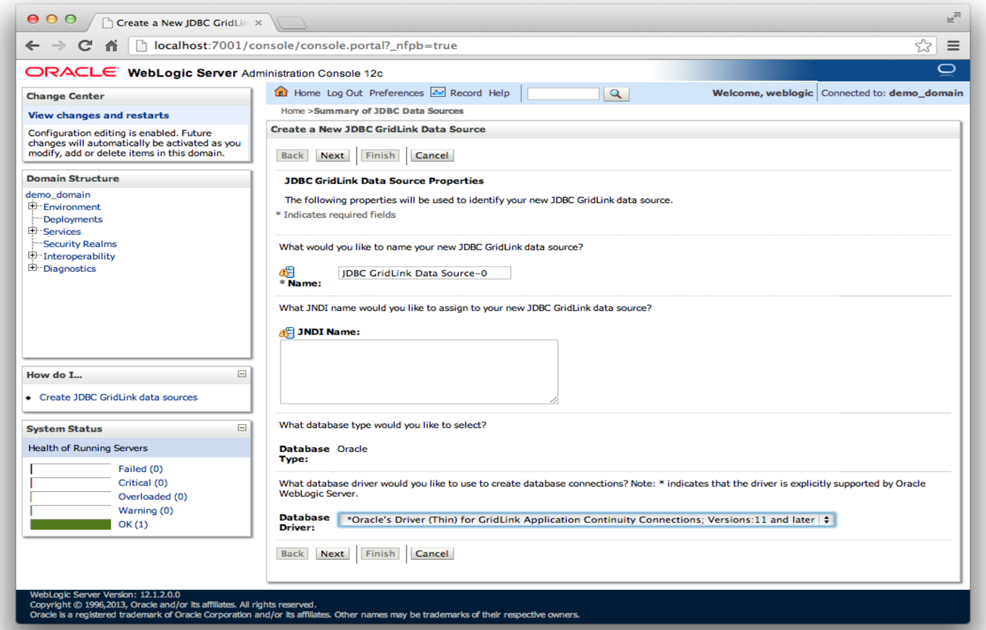
Application Continuity requires minimal effort to enable transaction replay with Oracle JDBC driver. The semantics assure that end-user transactions are executed on time and at-most-once. The only time an end user should see an interruption in service is when the outage is such that it is not possible to recover.

Database conversation interruptions are masked from users. Application Continuity allows requests to be retried safely elsewhere in the system without risk of duplication, increasing application availability, boosting developer productivity and improving the user experience.

Application Continuity will only replay recoverable errors, meaning those errors that occur following planned and unplanned outages of foregrounds, networks, nodes, storage devices, and databases. While applications sometimes receive error codes that don't reveal the status of the last operation submitted, Application Continuity reestablishes database sessions and resubmits the pending work for recoverable errors. It does not resubmit work following call failures due to nonrecoverable errors such as when invalid data values are submitted.

Transaction Guard supplies a unique logical transaction identifier (LTXID) for each database transaction. This identifier can be used to query the Commit outcome of the transaction as well as to ensure that the transaction is applied only once. Transaction Guard is used by Application Continuity and automatically enabled by it, and it can also be enabled independently. It prevents transactions that are replayed by Application Continuity from being applied more than once.

To use this feature in Oracle WebLogic Server, configure an Active GridLink data source by choosing the JDBC replay driver oracle.jdbc.replay.OracleDatasource. (This driver supports FAN/Fast Connection Failover and Runtime Load Balancing for FAST error detection and smart rebalancing.)


Create a New JDBC GridLin
localhost:7001/console/console.portal?_nfpb=true
ORACLE WebLogic Server Administration Console 12c
Home Log Out Preferences Record Help
Welcome, weblogic
Connected to: demo_domain
Change Center
View changes and restarts
Domain Structure
demo_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Create JDBC GridLink data sources
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (1)
Home >Summary of JDBC Data Sources
Back
Next
Finish
Cancel

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Create a New JDBC GridLink Data Source

JDBC GridLink Data Source Properties

The following properties will be used to identify your new JDBC GridLink data source.

* Indicates required fields

What would you like to name your new JDBC GridLink data source?

* Name: JDBC GridLink Data Source-0

What JNDI name would you like to assign to your new JDBC GridLink data source?

JNDI Name:

What database type would you like to select?

Database Type: Oracle

What database driver would you like to use to create database connections? Note: * indicates that the driver is explicitly supported by Oracle WebLogic Server.

Database Driver: *Oracle's Driver (Thin) for GridLink Application Continuity Connections; Versions:11 and later

WebLogic Server Version: 12.1.2.0.0
Copyright © 1996,2013, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Figure 3: Configuring Application Continuity in Oracle WebLogic Server

Applying Application Continuity in Oracle WebLogic Server

When Oracle WebLogic Server application first requests a connection, the data source will call "begin" statement so that JDBC operations are remembered through final Commit. When the connection is returned to the pool an "end" statement is issued. Other important considerations are listed below:

- Using Initialization Callbacks – Initialization Callbacks are used to initialize the session and maintain the same state of the session at replay. The callback is executed at every connection check out from the pool and at each successful reconnect following a recoverable error at replay. Use the same callback at run time and at replay to ensure that exactly the same initialization that was used when the original session was established is used during the replay. If the callback invocation fails, replay is disabled on that connection.

- Setting Replay Timeout – To configure the timeout use the replay-initiation-timeout MBean parameter. The timeout starts from the "begin" request to the call of "end." When the value of replay-initiation-timeout is 0, no timeout is set. The default value is 3,600 seconds. The recommended setting is to match the HTTP timeout value.

- Application Continuity supports read and read/write transactions. XA transactions are not supported. To support transactions using non-XA drivers such as LLR and JTS configure a WebLogic data source selecting Oracle JDBC driver for Application Continuity.

- Applications should set `autocommit=FALSE` to prevent breaking the transaction semantics and disabling Application Continuity in their environment.

- Disabling Replay on a per Connection basis – Obtain a connection from the configured Oracle WebLogic Server Active GridLink data source, cast this connection to oracle.jdbc.replay.ReplayableConnection, and then call setReplayableEnabled(false) on this connection.

JDBC Replay with Oracle Driver and Database

The JDBC Replay Driver stores JDBC operations that affect each JDBC object's internal state, along with the arguments to those operations. For example, a Connection object stores all the Statement, PreparedStatement, and CallableStatement objects that it has created, along with all the ResultSet objects that they have created.

In order to conserve memory consumption, the JDBC Replay Driver purges recorded operations as necessary. Applications properly close ResultSets and Statements following use, via standard JDBC calls.

The JDBC Replay Driver purges the stored operation history and all replay-specific objects (such as checksums, cursor replay context) that are related to a Statement or a ResultSet. This purge includes PreparedStatement and CallableStatement when the Statement or ResultSet is closed and there is no bounding active transaction. These objects will also be purged if the connection is closed.

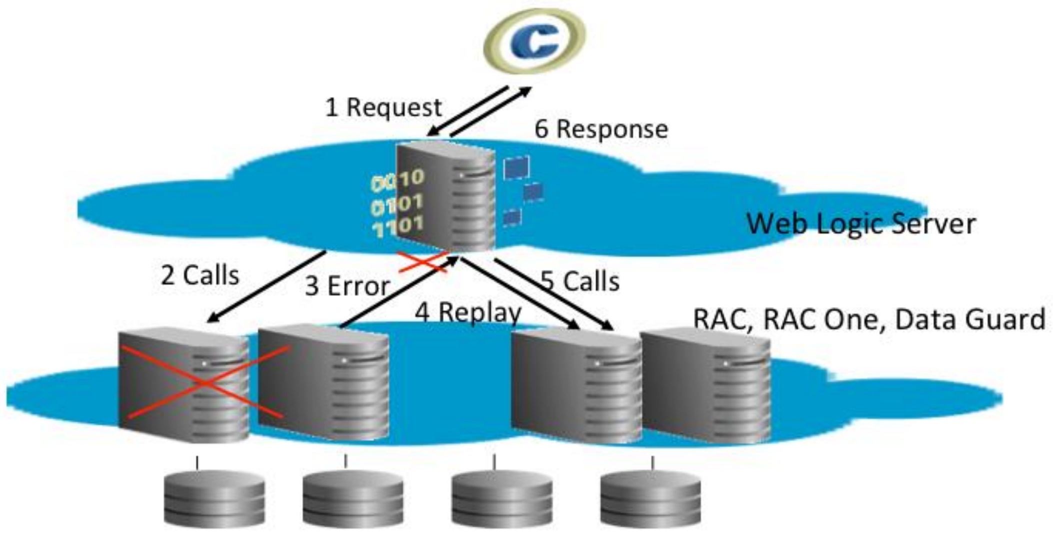


Figure 4: How Application Continuity Works

As shown in Figure 4, client application requests are passed to Oracle WebLogic Server and then on to Oracle Database using the JDBC replay driver.

- The JDBC replay driver issues each call in the request.
- A FAN unplanned or planned interruption or recoverable error occurs. FAN/FCF then aborts the dead physical session.
- Application Continuity begins the replay and does the following:
    - Replaces the dead physical session with a new clean session and rewires FAN in case a later error occurs during or after replay
    - Prepares for replay by using Transaction Guard to determine the outcome of the in-flight transaction, if one was open
    - Optionally, calls back using a labeling callback or reconnect callback for the initial state
    - Rebuilds the database session, recovering the transactional and non-transactional states and validating at each step that the data and messages seen by the client driver are the same as those that the client may have seen and potentially based a decision on
    - Ends the replay and returns to runtime mode
    - Submits the last queued call
- This is the last call made when the outage was discovered. During replay, only this call can execute a commit. A commit made mid-way through rebuilding the session will cause replay to be aborted (excluding autonomous transactions).
- The response is returned to the application.

If replay was successful, the application can continue with the problem masked. If replay was unsuccessful, the application will handle the original error.

Database Resident Connection Pools

Mid-tier data sources create many idle connections to accommodate high user demand. The cost of creating and destroying these connections is expensive. Database Resident Connection Pooling (DRCP) allows multiple web-tier and mid-tier data sources to share Oracle Database server processes and sessions (together known as pooled servers), enabling better sharing of database resources and greater application scalability. It scales best when the database connections are not always in use.

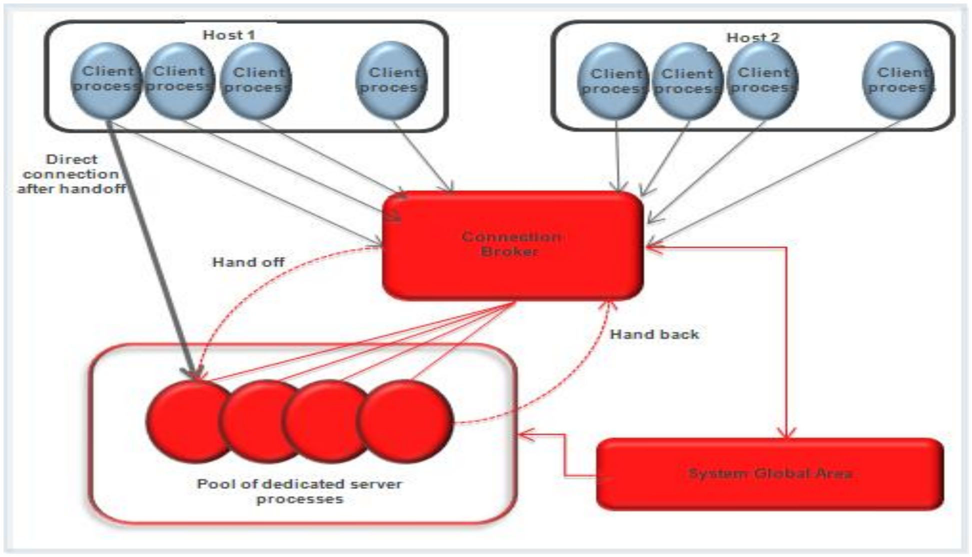


Figure 5: Database Resident Connection Pool

When the pooled connections are in use, they are equivalent to dedicated connections. Upon requesting a connection from the Oracle WebLogic Server data source, the appropriate pooled connection is handed-off. Oracle WebLogic Server directly communicates using the pooled connection for all database activity. The pooled connection is handed back when the data source releases it.

WebLogic Integration with DRCP works with Active GridLink for RAC and with generic data sources using Oracle Database 12c and JDBC Driver 12c. Connections from the data source connection pool are placeholders in an "unattached" state. When a connection is reserved by an application the connection is first associated with a database session by invoking attachServerConnection() on the connection object. When the connection is returned to the pool it is detached from the session by invoking detachServerConnection().

When the pooled connection is used in an XA transaction, the container retains the session for the life of the transaction.

DRCP guarantees that the pooled connections are never shared across different users. However, it does allow connections to be shared and pooled across different instances of the same application. Even for the same user, DRCP maintains a logical grouping of the pooled servers based on the "connection classes" chosen by the applications. A connection class is a logical name supplied by a client when a pooled connection is requested. It indicates that the client is willing to reuse a pooled connection that was used by other clients using the same logical name.

For example, if there are ten Oracle WebLogic Server data sources all pointing to the same database, and each data source has an initial capacity of 10 connections, then 100 connections/sessions will be created at startup. If they aren't all active all the time, then that is a waste of connections. With DRCP they can all share a single connection pool with a configurable number of connections. As with any type of virtualization, it only works if you have spare capacity and everyone doesn't try to use all of the resources at the same time.

DRCP boosts the scalability of Oracle Database and Oracle WebLogic Server since persistent connections to the database are held at minimal cost. Database resources are only used by the active connections. Administrators can scale the usage by controlling and setting the pool size.

Multitenant Database

Oracle Database implementations typically fall into the following categories:

- Small to medium size databases that each support a separate application
- Business databases in which each module of an application uses a separate database
- Multitenant databases with multiple copies of the same database (one per tenant)
- Multiple tenant databases with different collections of schemas to support separate tenants

Multitenant Database implementations allow multiple distinct databases in a single larger database installation. The Container Database (CDB) feature in Oracle Database 12c minimizes the overhead of these multi-database configurations by consolidating them into a single database with multiple Databases (PDB) in a single Container Database. The benefits of this type of “in-database virtualization” include the following:

- The ability to upgrade Oracle Database transparently to later versions within the context of a single Container Database
- The ability to run multiple versions of Oracle Database (tenant DBs) inside of a Container Database
- More efficient use of hardware resources
- Unified security management
- Greater density and scalability in the middle and data tiers, leading to better resource utilization
- Support for multiple tenants in a single database

When the client connects to a particular container (PDB) it issues ALTER SESSION SET CONTAINER. This new driver will make sure the container is a valid/existing one and that the current user has the correct privileges to connect to that container. If these checks are successful, the user data in the session is updated with the new PDB name and ID.

The usual connection to a PDB will be through a service. Every PDB will have a default service when it is created, much like the Database service that exists for a singleton database today. Other services can be created for the PDB explicitly. When an application connects using a service, the PDB attribute is used to set up the correct PDB context in the database session.

Oracle WebLogic Server integrates with Multitenant databases with two different configurations. One configuration would be to have a single data source per PDB. If a new PDB is added to the CDB a new data source needs to be configured in WebLogic Server. The benefits of this model are to provide scalability and elasticity at the database level.

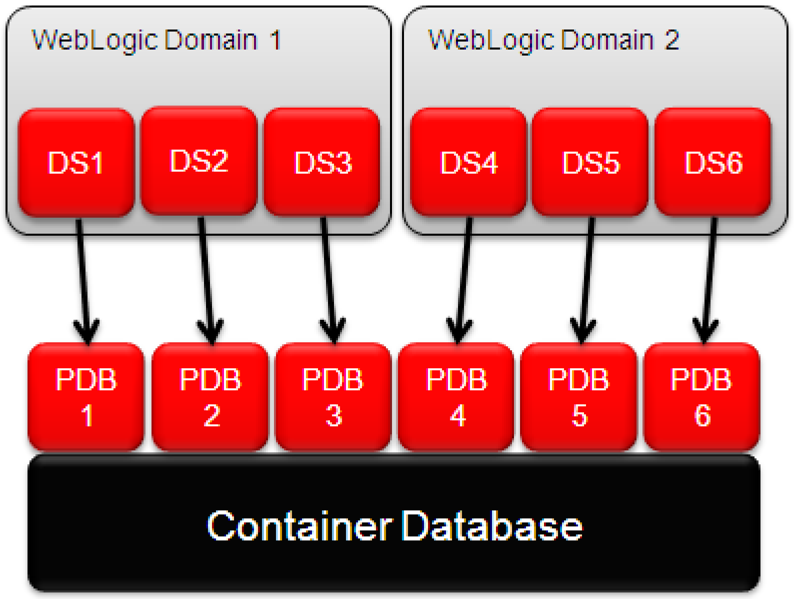


Figure 6: Oracle WebLogic Server configuration showing one data source per Multitenant Database.

A second configuration is to have a single data source to pool connections to multiple Multitenant databases. However, connecting for the first time to a PDB, or switching between the PDBs is not transparent to the application. To enable switching between PDBs, a best practice is to register a callback that changes the connection with the ALTER SESSION SET CONTAINER, sets other properties, and then labels the connection. Any subsequent request for a connection to the same PDB would not need to be altered. This model would provide the true benefits of a pooled connection, including scalability and elasticity.

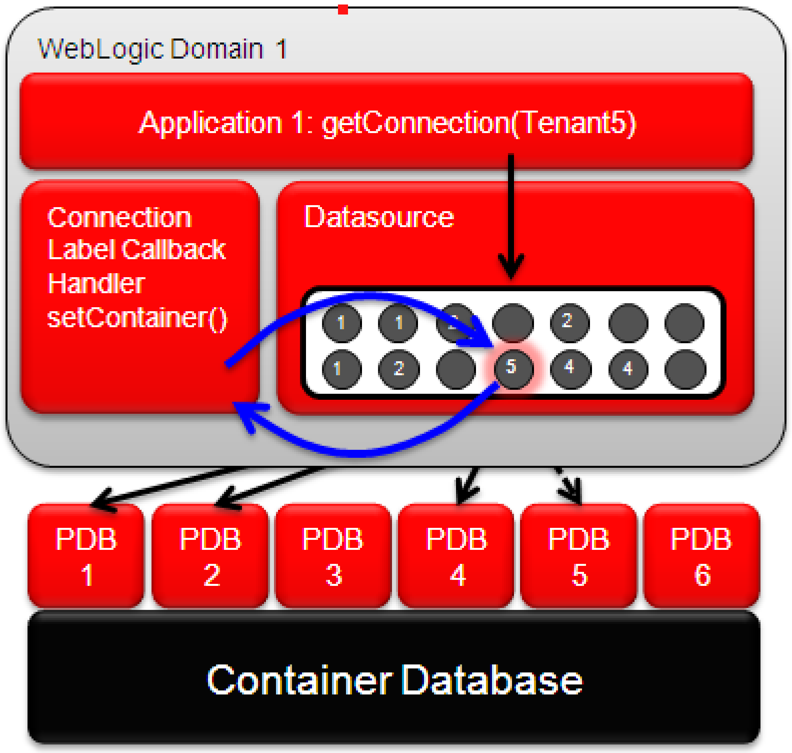

WebLogic Domain 1
Application 1: getConnection(Tenant5)
Connection Label Callback Handler setContainer()
Datasource
1
1
2
2
1
2
5
4
4
PDB 1
PDB 2
PDB 3
PDB 4
PDB 5
PDB 6
Container Database

Figure 7: Oracle WebLogic Server and Multitenant Database

You should be aware of some limitations of PDB switching with Oracle Database 12.1. The following WebLogic Server limitations exist when using tenant switching.

- Oracle RAC Fast Application Notification (FAN) is not supported. Even though FAN is not supported, Active GridLink still provides the benefit of a single data source view of multiple RAC instances and the ability to reserve connections on new instances as they are available without reconfiguration using connection load balancing. If you want to use tenant switching with an Active GridLink data source, “FAN enabled” must be set to false
- Database Resident Connection Pool (DRCP) is not supported
- Application Continuity is not supported.
- Proxy authentication is not supported.
- Recovery of XA transactions in single data source that is used for switching between multiple PDB's is not supported.

Auto ONS

With the initial version of Active GridLink with Oracle Database 11g, you must configure an ONS listener list when FAN is enabled. With Oracle Database 12c the ONS list is optional; the information is automatically provided from the database to the driver. The auto-ONS feature works only with Oracle Database 12c RAC or the new Global Database Service (GDS) feature. It does not work with a single-instance Oracle Database.

Currently, the Oracle WebLogic Admin Console requires ONS list for GridLink data source configurations when FAN is enabled. In Oracle WebLogic Server 12.1.2, the ONS list will be optional if you are using the Oracle Database 12c and JDBC 12c driver.

Validation in the Oracle WebLogic Admin Console will only be done when the data source is targeted during creation or re-targeted during an update. This approach is similar to other values such as the URL. If the administrator forgets to enter an ONS list, the targeting will fail. The ONS list will need to be updated, and the targeting will need to be done again.

The ONS list will be validated during deployment. If the list is empty, the system will verify that the Oracle Database 12c driver is being used.

Global Data Services

Global Data Services (GDS) streamline the delivery of database services on a global scale, which is key to deploying databases in a large cloud architecture. These technologies oversee replication and failover while performing load balancing within and across data centers, optimizing resource utilization and streamlining database management practices in a distributed database environment. GDS works by enabling a Global Service across RAC and single-instance Oracle databases interconnected via Oracle Data Guard, Oracle GoldenGate, or any other replication technology. Client access to this distributed infrastructure is completely transparent. GDS implementations are easy to apply to Oracle WebLogic Server with minimal changes.

The benefits of Oracle Database 12c Global Data Service include the following:

- Central management of database services across a distributed database cloud
- Dynamic migration of services based on load and availability
- Scalability by adding RAC clusters
- Automatic restart of failed services on an available database
- Easy integration with Oracle WebLogic Server via the Active GridLink data source

When configuring the GridLink data source you simply specify a primary local region from which to access the global service, along with the addresses to each region. This configuration enables RAC-like failover for Oracle Database in the cloud. If a region loses its connection to the global database service the reconnection is based on FAN events. There is no need to restart mid-tier components on failure, ensuring business continuity.

GDS is designed to inherently balance the database load across global services. When there is a heavy load on a Read-Only service and the global service is available in another region, the GDS framework will notify the Active GridLink and a new connection will be made to the service in the other region. (The framework cannot load-balance across regions for Read-Write services. This type of processing can only be performed in the primary region.)

Oracle WebLogic Server administrators can configure a GDS data source via the Admin Console as follows:

- Specify connections by denoting:
    - Service name (Global Service Name)
    - Address/port pairs (for various Global Service Managers)
    - GDS Region (new)
- Listeners cannot be tested after configuring a GDS Data source.
- It is not possible to use a single SCAN address, instead of multiple GSM addresses.service name need to be configured. The following is a sample URL:

```
jdbc:oracle:thin:@(DESCRIPTION=
 (ADDRESS_LIST=(LOAD_BALANCE=ON)(FAILOVER=ON)
 (ADDRESS=(HOST=slc02wqh.us.oracle.com)(PORT=2711)(PROTOCOL=tcp))
 (ADDRESS=(HOST=slc02wqh.us.oracle.com)(PORT=2709)(PROTOCOL=tcp)))
 (CONNECT_DATA=(SERVICE_NAME=uniformdg1.pool1.oradbcloud)(REGION=EAST)))
```

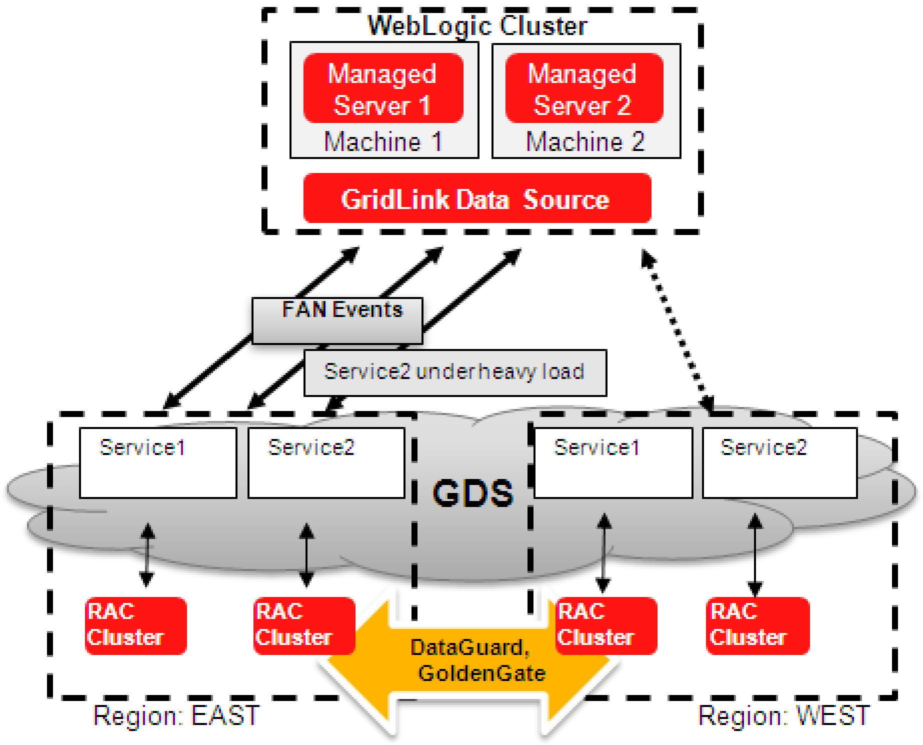


Figure 8: Oracle WebLogic Server and Global Data Services

In order for update operations to be handled correctly you must only define a service for updates on the primary database. Read-Only operations can be directed to another service that is defined on the primary and secondary databases. Since only a single service can be defined on a URL and a single URL on the data source configuration, one data source must be defined for the Update service and another data source must be defined for the Read-Only service. The application must be written so that Update operations go to the Update data source and Read-Only operations go to the Read-Only data source.

Summary

Oracle WebLogic Server is tightly integrated with Oracle Database 12c to provide the highest level of availability, failover, multi tenancy, resource sharing, scalability, ease of configuration and management, all within the context of a cloud like infrastructure. It delivers a complete, best-of-breed data-processing platform to enable higher availability, scalability and performance for your business.

References

Oracle WebLogic Server Active GridLink for Oracle Real Application Clusters (RAC)

http://www.oracle.com/technetwork/middleware/weblogic/gridlink-rac-wp-494900.pdf

Oracle WebLogic Server and Highly Available Oracle Databases: Oracle Integrated Maximum Availability Solutions

http://www.oracle.com/technetwork/middleware/weblogic/learnmore/1534212

Database Resident Connection Pooling (DRCP)

http://www.oracle.com/technetwork/articles/oracledrcp11g-1-133381.pdf

White Paper Title
June 2013
Author: Monica Riccelli, Frances Zhao-Perez

Contributing Authors: David Baum

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com