An Oracle White Paper
July 2013

Oracle WebLogic Suite 12c (12.1.2) Technical White Paper

Introduction

Oracle WebLogic Suite 12.1.2 is a significant new release of the flagship edition of the Oracle WebLogic Server product line. Oracle WebLogic Suite is an integrated solution for building on-premise cloud application infrastructures that span web server, application server and data grid technology tiers. It includes Oracle WebLogic Server, the #1 application server for developing and deploying applications across conventional and cloud environments, Oracle Coherence Enterprise Edition, Oracle Web Tier, Oracle TopLink, Oracle Java Mission Control and Flight Recorder, and other component technologies.

Important business trends and requirements have shaped the Oracle WebLogic Suite offering. Businesses and CIOs increasingly place a high value on IT cost reduction, rapid introduction of new and improved products, and increased productivity. For years, Oracle customers have used Oracle WebLogic Suite to meet these requirements. They use Oracle WebLogic Suite as an application infrastructure for rapid application development, on-demand provisioning of resources for application deployment, efficient management, and high application service levels for both mobile and desktop clients. In effect these customers have built their own private cloud environments. Cloud computing is formalizing the approach to building such infrastructures for both publicly hosted and private, on-premise deployments, and IT organizations are looking to application infrastructure technologies to provide the building blocks of a cloud computing platform solution for productive development, efficient management, and high application service levels, all with low cost of ownership.

Oracle WebLogic Suite 12.1.2 builds on the capabilities of prior releases to meet these business requirements. Through pre-existing and new features and product integration capabilities, it delivers a highly productive, modern application development platform, an efficient and integrated management framework with native cloud management capabilities for low cost of ownership, and a mission critical cloud platform for maximum application performance, availability and scalability to diverse clients. These capabilities are described in the following sections of this document.

Licensing

In Fusion Middleware 12.1.2, Oracle provides a single integrated infrastructure licensed as Oracle WebLogic Suite. Oracle WebLogic Suite contains the following major server side runtime components:

- Oracle Web Tier – including Oracle HTTP Server, Oracle iPlanet, and web server plug-ins for creating a web server tier to front end Oracle WebLogic Server deployments.

- Oracle WebLogic Server - the #1 application server for developing and deploying applications across conventional and cloud environments. Oracle WebLogic Suite includes rights for all Oracle WebLogic Server features, including development and management tools, clustering, and Active GridLink for RAC.
- Oracle Coherence Enterprise Edition– the industry's most predictable, reliable, scale-out in-memory data grid.
- Oracle Java SE Suite – the market share leading Oracle JDK (HotSpot), with new capabilities first introduced in Oracle JRockit, and built-in JDK support from Oracle. Oracle JDK provides the Java VM for running Oracle WebLogic Server and Oracle Coherence. Oracle WebLogic Suite also supports other JDK implementations.

Also included in Oracle WebLogic Suite are the following:

- Oracle TopLink – based on the EclipseLink Java Persistence API reference implementation, with new features such as TopLink Data Services added in version 12.1.2
- Choice of IDEs - Oracle Enterprise Pack for Eclipse, the leading Eclipse-based IDE for developing Oracle WebLogic Server applications; Oracle JDeveloper, the strategic IDE for Oracle Fusion Middleware application development; Oracle NetBeans, the open source IDE supporting the latest Java EE APIs.
- Oracle Application Development Framework – an MVC-based runtime framework for building applications with Oracle JDeveloper or Oracle Enterprise Pack for Eclipse that run on Oracle WebLogic Server.
- Oracle iAS Enterprise Edition – including license rights to Oracle Containers for Java (OC4J) and related component technologies such as Oracle Forms and Reports.

Oracle WebLogic Suite 12.1.2 licenses can be extended with additional licenses for the Oracle Coherence Grid Edition Option, and Oracle Enterprise Manager Management Packs – Oracle WebLogic Server Management Pack Enterprise Edition and the Cloud Management Pack for Oracle Fusion Middleware. For detailed licensing information on Oracle WebLogic Suite, refer to the Oracle Fusion Middleware 12.1.2 licensing documentation.

Note that some of the individual product components licensed and included in Oracle WebLogic Suite 12.1.2 are physically packaged and delivered in separate installation programs to align with typical usage. For example, the key Java application infrastructure technologies – Oracle WebLogic Server (including Oracle TopLink) and Oracle Coherence – are packaged in the same installation program and are installed on the JDK installation that you have created on your systems. Web Tier technologies are packaged separately and typically installed on separate systems. For detailed installation instructions on Oracle WebLogic Suite product components refer to the applicable Oracle Fusion Middleware 12.1.2 product documentation.

Oracle WebLogic Suite Topology Overview

The following diagram provides a high-level summary of a typical Oracle WebLogic Suite production deployment topology:

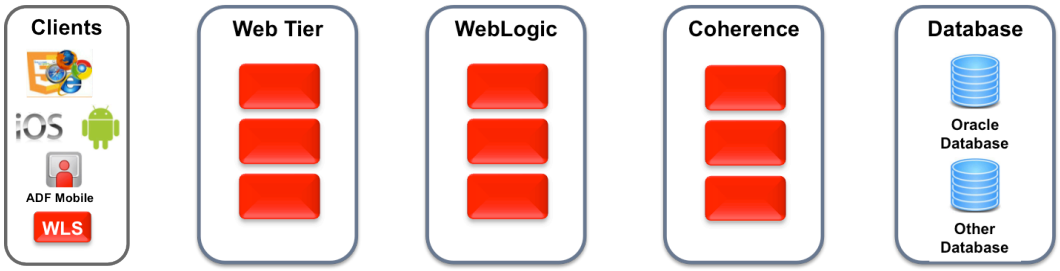


Figure 1: Typical Oracle WebLogic Suite Production Deployment Topology

In this deployment topology, there are clients, including browser clients, native mobile clients, or Oracle WebLogic Server Java clients accessing applications deployed on Oracle WebLogic Server. Typically these clients connect to Oracle WebLogic Server managed servers and clusters through web server proxies hosted on a separate set of web server systems, denoted as the Web Tier above. Web server plug-ins running in the web servers forward client requests to Oracle WebLogic Server applications running in the WebLogic tier of the topology above.

These applications typically access and update data stored in a database, usually an Oracle Database, but other third-party databases are supported as well. Application data being used for application processing is stored in memory in the WebLogic tier in the form of Java objects. In order to achieve better performance and scalability of data-intensive applications, it is often beneficial to move in-memory application data from Oracle WebLogic Server, into a separate Oracle Coherence data grid tier, to provide distributed caching, event, analytics and compute processing services to Oracle WebLogic Server applications. Both Oracle WebLogic Server and Oracle Coherence have unique capabilities to integrate effectively with the database tier, especially with Oracle Database and Oracle Database Real Application Clusters.

This high-level summary omits the more detailed capabilities of the infrastructure, including the security and load balancing functionality provided by the Web Tier, the development, management, performance and reliability features in the WebLogic Tier, and the massive scaling with built-in reliability and failover offered in the Coherence Tier. These capabilities are described in more detail throughout this document. In summary, we believe that Oracle WebLogic Suite provides the industry's leading application infrastructure implementation across the web server, application server and data grid tiers.

In Oracle WebLogic Suite 12.1.2 we have taken these capabilities to the next level. Oracle WebLogic Suite 12.1.2 simplifies the implementation of cloud application infrastructures that span the web server, application server and data grid tiers by delivering:

- A *modern development platform* for productive application development including significant updates to Maven integration, new WebSocket protocol support, and TopLink Data Services for simplifying development of applications that provide RESTful interfaces to enterprise data.
- *Native cloud management* capabilities that enable efficient management of cloud infrastructures using consistent, proven tooling across the web server, application server and data grid tiers. Standardization on Oracle Universal Installation (OUI) and OPatch technologies provides a unified installation and patching framework for cloud provisioning. Extensions to the WebLogic Management Framework enable the consistent use of proven WebLogic management capabilities for managing the web server and data grid tiers. And we continue to make enhancements to the WebLogic Management Framework through enhancements such as Elastic JMS.
- A *mission-critical cloud platform* that meets the most demanding requirements for cloud application reliability, availability, scalability and performance across the web server, application server and data grid tiers. Dynamic Clusters, Oracle Database 12c integration, and Exalogic optimizations are examples of new features we have added to enhance service levels provided to applications.

The next sections of this document describe the capabilities of Oracle WebLogic Suite 12.1.2 in each of these areas, highlighting where applicable the new features added in Oracle WebLogic Suite 12.1.2. The document first describes the modern development platform capabilities of Oracle WebLogic Suite, focusing on development of Oracle WebLogic Server applications. The next section of the document describes the native cloud management capabilities of Oracle WebLogic Suite, again focusing on Oracle WebLogic Server management capabilities, but also describing how these capabilities can be applied to manage the Web Tier and Coherence layers of the typical Oracle WebLogic Suite deployment architecture. The following section describes the service levels that are provided by the Oracle WebLogic Suite mission-critical cloud platform, and how those service levels are delivered.

Modern Development Platform

Oracle has made significant improvements in Oracle WebLogic Suite to provide a modern development platform that simplifies development of applications, including cloud applications, and that makes application developers more productive. Key development capabilities are described below.

Java EE Standards Support

In order to maximize productivity and reduce risk in the usage of middleware infrastructure, full adoption of standards is key. Standards also help to leverage developer talent, provide interoperability support, and prevent vendor lock-in to proprietary technologies. Since Oracle WebLogic Server 12.1.1, Oracle WebLogic Server has been built on and supports Java Enterprise Edition Version 6 (Java EE 6).

The primary focus of Java EE 6 is providing features that improve developer productivity. Java EE 6 accomplishes this by enabling broader use of annotations, simplifying component reuse and integration across components, reducing the amount of code required to implement a given level of functionality, and reducing the amount of XML required to deploy a Java EE application. The capabilities and simplifications provided in Java EE 6 enable developers to focus application development efforts on implementing business functionality, instead of working with the Java EE infrastructure. A partial list of Java EE 6 enhancements includes:

- Servlet 3.0 adds support for annotations, asynchronous processing of client requests, and web fragments which simplify use of third-party frameworks with Java EE 6.
- JSF 2.0 adds support for annotations, and provides new page authoring enhancements such as JSF templates. Templates enable construction of similar pages within an application from a common template, to facilitate rapid development of applications with a common look and feel.
- EJB 3.1 no-interface views allow developers to specify an enterprise bean using a bean class, without requiring developers to write a separate business interface. EJB 3.1 also allows EJBs to be packaged directly in WAR files, simplifying packaging and deployment of applications using EJBs.
- Contexts and Dependency Injection (CDI) is a new API in Java EE 6. CDI bridges the gap between the web tier and enterprise tier of Java EE 6 applications by enabling EJBs to replace JSF managed beans in JSF applications. This simplifies the development of web applications that access transactional resources in Java EE applications.
- JPA 2.0 adds enhancements to Object/Relational Mapping and the Java Persistence Query Language, and provides a new Criteria API for constructing object-based queries. EclipseLink, which is the foundation of Oracle TopLink included in Oracle WebLogic Server, is the reference implementation for JPA 2.0.
- Bean Validation is another new API in Java EE 6, enabling reuse of validation logic across multiple components of an EE 6 application, avoiding the complexity of duplicating validation logic across application components.
- The Java API for RESTful Web Services (JAX-RS) is a new API that standardizes the development of lightweight RESTful services within Java EE applications.

Oracle WebLogic Server 12.1.2 provides full compatibility and support for Java EE 6 applications developed on Oracle WebLogic Server 12.1.1, and incorporates user feedback. Oracle WebLogic Server 12.1.2 (and Java EE 6) also provides compatibility with Java EE applications developed on prior Oracle WebLogic Server versions. From a developer's perspective, support for the latest standards and specification ensures that skills are effectively leveraged and are transferrable to future projects.

Web Services

Oracle WebLogic Server has a comprehensive and fully standards compliant Web Services infrastructure that supports and extends the Java EE 6 standards. The primary Java programming model supported is the developer friendly annotation based JAX-WS Java EE 6 standard. However for backwards compatibility JAX-RPC is also supported. For quality of service a complete set of WS-* standards including WS-Policy, WS-PolicyAttachment, WS-Security, WS-SecureConversation, WS-SecurityPolicy, WS-Trust and more are supported. Complementing this implementation is rich support of Oracle Web Services Manager for policy management and policy enforcement Web services.

WebSocket

WebSocket is a new protocol standard that is supported by standard HTML5 clients and is rapidly emerging as an important protocol for web applications. The WebSocket protocol allows a standard client, for example the latest browser versions, or mobile clients, to create a persistent, bi-directional connection with an application server. The client uses HTTP to request that a WebSocket connection be established. Having created the WebSocket connection, this connection can be re-used for bi-directional data exchange between clients and server applications. This is in contrast to the traditional HTTP approach where HTTP header information must be exchanged before data is transmitted between client and server.

The combined characteristics of low overhead, persistent bi-directional connectivity, and standardization in HTML5 clients make WebSocket an attractive protocol for building a new generation of applications. For example, gaming applications, or stock trading applications, or applications using chat interfaces can be developed to deliver lightweight data updates to mobile or desktop clients without incurring the full overhead of an HTTP request, or the generation and rendering of a new web page. In such applications, data updates received from the server may be dynamically updated or rendered in JavaScript clients running on the end user's device. From the perspective of the end user, they experience better response time on a standard client. From the perspective of the application server, there are dramatic scalability benefits because less processing is being performed on the server. Such benefits make deployment and scaling of new data-intensive, mobile-ready application types possible.

A concern is often raised regarding the promised scalability of such new application types. While it is recognized that there are scalability benefits from reduced overhead in data exchanges

between clients and servers, there are questions about how many open socket connections can be supported on an individual server instance. Will WebSocket architectures scale to support large numbers of concurrent clients?

Initial internal testing has shown that Oracle WebLogic Server 12.1.2 instances can maintain up to 60,000 WebSocket connections per server instance. We do not claim that this means all applications will be able to scale up to 60,000 connections per server instance. But we do not believe that scalability of WebSocket connections in Oracle WebLogic Server will be a barrier to delivery of innovative WebSocket applications in most cases.

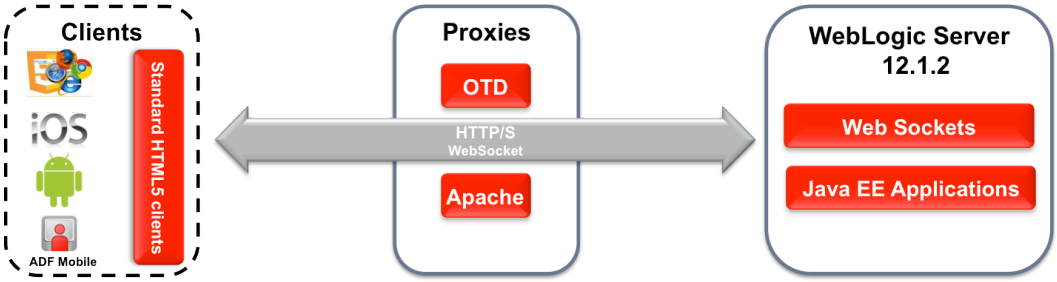


Figure 2: WebSocket Protocol Support in
Oracle WebLogic Server 12.1.2 and Proxies

Oracle WebLogic Suite 12.1.2 supports the WebSocket protocol in Oracle WebLogic Server 12.1.2 and Apache Web Server plug-ins for Web Tier support. Additional support is also provided in Oracle Traffic Director and is available on Exalogic systems.

Oracle TopLink and TopLink Data Services

Oracle TopLink 12.1.2 is fully integrated into the Oracle WebLogic 12.1.2 infrastructure. Oracle TopLink, based on the open source Eclipse Persistence Services Project (EclipseLink), is an advanced object-persistence framework that provides runtime capabilities that reduce development and maintenance efforts and increase enterprise application functionality. TopLink is designed for use in wide range of Java EE and Java SE architectures. TopLink's most popular persistence services include:

- Relational: For persistence of Java objects using the standard Java Persistence API (JPA) specification to a relational database accessed using Java Database Connectivity (JDBC). TopLink provides advanced features for all leading databases with specialized support for Oracle Virtual Private Database, XML DB XMLType, flashback, and stored procedures and functions with Oracle Database.
- TopLink Grid: For integration with Oracle Coherence that supports scaling JPA applications up onto large clusters and leveraging the processing power of the grid to parallelize queries for Oracle Coherence cached objects.

- XML: For conversion between Java objects and XML documents using Java Architecture for XML Binding (JAXB) and Service Data Objects (SDOs). EclipseLink's provided SDO solution, a core component of the Oracle's SOA infrastructure, is the SDO 2.1.1 reference implementation.
- NoSQL: This new feature provides a persistence API to assist developers with mapping their entities to NoSQL data structures and support for translating queries and transaction operations into native database operations. This first release of the NoSQL support includes platform support for MongoDB and Oracle NoSQL, as well as being customizable for other platforms.

Most significant for development of applications servicing HTML5 or mobile clients, TopLink Data Services support available in prior versions has been expanded to include RESTful Data Services support in Oracle TopLink 12.12. TopLink Data Services now makes it easy to expose JPA entities over REST so that HTML5 clients using JavaScript, or mobile clients, can easily retrieve and manipulate data stored in databases via Oracle WebLogic Server using REST. JPA application developers can expose RESTful interfaces to existing entities declaratively, with no additional programming required. Customizable bindings to for both JSON and XML are supported. Powerful capabilities, such as listening on Oracle Database notifications, and updating RESTful interfaces automatically, are also available. Sample applications illustrate how to use TopLink Data Services with the new WebSocket protocol support provided in Oracle WebLogic Server.

Integrated Development Environment (IDE) Support

Oracle WebLogic Suite offers a broad choice of IDEs for application developers. Oracle JDeveloper 12.1.2 has been updated to support development of Oracle WebLogic Server 12.1.2 applications. It offers a complete development environment with code editing, testing, debugging, profiling and support the Oracle Application Development Framework (ADF) for rapidly building and deploying enterprise grade applications on Oracle WebLogic Server.

For those Java EE developers that prefer the Eclipse integrated development environment, Oracle WebLogic Suite offers Oracle Enterprise Pack for Eclipse (OEPE), providing all of the tools necessary for Java EE development and deployment on Oracle WebLogic Server. Oracle Enterprise Pack for Eclipse 12.1.2 includes support for full Java EE 6 application development, support for WLST script authoring, and support for development of Oracle Coherence applications, including support for the new Grid Archive (GAR) packaging provided in the new Managed Coherence Servers feature described in later sections of this document.

Finally, the Oracle NetBeans IDE also supports development of Oracle WebLogic Server 12.1.2 applications. Oracle NetBeans is an open source IDE implementation that leads the industry in supporting the latest Java EE specifications. It supports both Oracle Glassfish Server, based on the Glassfish open source the Java EE reference implementation, and Oracle WebLogic Server. Developers who have experimented with development of Java EE 6 applications using NetBeans

and Glassfish will experience a natural evolution to use of NetBeans for development of Oracle WebLogic Server applications.

Lightweight Application Server

While Oracle WebLogic Server provides sophisticated runtime features that are extremely valuable in production, not all of these features are relevant during the development process. Developers have asked for packaging and development usage options that can mask these more sophisticated product features in order to create a more "lightweight" developer experience. We continue to invest in delivering such features to developers.

Since Oracle WebLogic Server 11g, Oracle has provided two primary types of software distributions. The first is a full installation program, including both Oracle WebLogic Server and Oracle Coherence, that can be used for both development and production purposes, and is fully supported by the patching and other production support tools provided with Oracle WebLogic Server. In addition, we have provided a zip distribution specifically for development purposes. This lightweight distribution is 179 MB, and is intended to provide a more lightweight download and installation process for those developers who prefer to install software from a zip file. The zip distribution supports the major development platforms – Windows, Linux and Mac OS X. Note that the zip distribution is supported for development purposes only and is not supported in production. It is also not patchable using the new patching tools provided in Oracle WebLogic Server 12.1.2.

Developers have a choice of service startup with both types of installation. Starting a lightweight server without EJB, JMS and JCA services is accomplished by specifying -DserverType="wls" or -DserverType="wlx" as a startup system property. Developers of Web applications, for example, can choose to work with a container in which unnecessary services are not started.

Internal applications, such as the Oracle WebLogic Server administration console, are deployed upon demand to further accelerate server startup and to minimize resource consumption. The Oracle WebLogic Server administration console disables the change center in development mode as well as eliminating confirmation dialogs, thereby allowing configuration changes to be made quickly.

Oracle WebLogic Server Maven Plug-in

Apache Maven is an open source tool for automating software development and build operations. Apache Maven provides a structured approach for executing a sequence of operations such as application compilation, deployment and test execution in Maven projects. Application developers and organizations have been adopting Maven to increase efficiency, reliability and productivity of development, build and QA processes. Software vendors can simplify the use of their products with Maven by providing plug-ins that can be used to control the execution of standard Maven operations with the vendors' products. Maven plug-in "goals" drive product-

specific features and APIs to accomplish standard Maven project functions like application compilation, etc.

Beginning in Oracle WebLogic Server 11g, a WebLogic Maven plug-in was provided to support application deployment operations through support of deployment goals in the WebLogic Maven plug-in. In Oracle WebLogic Server 12.1.1, WebLogic Maven plug-in support was expanded to include support of additional goals required for application development within Maven projects. Goal support included domain creation, starting and stopping of servers, execution of WLST scripts to modify server configurations, and compilation of applications, in addition to deployment of applications. The Maven goal support provided in Oracle WebLogic Server 12.1.1 provides rich support for the majority of lifecycle operations that would be required by developers using Maven.

The Maven support in Oracle WebLogic Server 12.1.2 takes Maven support one step further. Maven is designed primarily to support Maven project execution by referencing software modules stored in a Maven repository, either a local repository on the user's system, a remote shared private repository, or public repositories such as the Maven Central Repository. The Maven plug-in support provided in Oracle WebLogic Server 12.1.1 works by referencing a local product installation rather than use of repositories. In Oracle WebLogic Server 12.1.2, we have provided a new sync plug-in that can be used to populate a local or shared private Maven repository with the appropriate Oracle WebLogic Server components, including patched components. Where possible WebLogic Maven plug-in goals will reference modules in the repository by default.

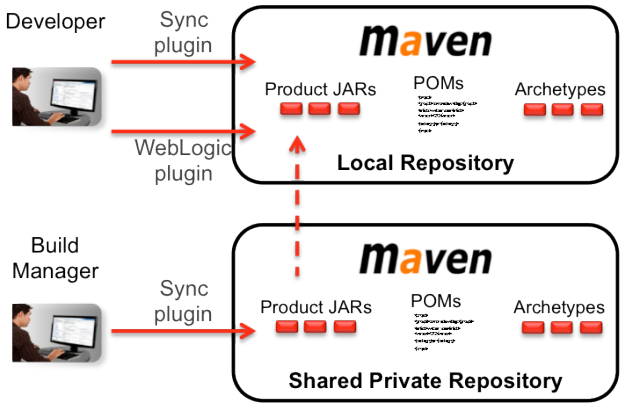


Figure 3: Use of the new Maven Sync Plug-in
For Populating Maven Repositories

This approach, which is also supported by Oracle Coherence and Oracle TopLink and will be adopted more broadly in future Oracle Fusion Middleware releases, makes it easier for organizations to use consistent software versions and patch levels across development projects, and to use consistent approaches for referencing software provided by different software products.

WebLogic Maven archetypes are also provided to facilitate creation of web applications, web/EJB applications, Web Services and MDB applications.

Oracle WebLogic Server Ant Tasks

Oracle WebLogic Server continues to provide a full set of Ant tasks that allow for lifecycle, packaging, code compilation, resource configuration and applications operations. These Ant tasks can also be used to generate automation scripts, thus simplifying and reducing human error for repetitive as well as complex tasks.

Automation of Development and Test Environments

Oracle WebLogic Server provides a set of tools that allows developers (as well as administrators) to automate various aspects of their daily work in order to avoid performing repetitive tasks.

For example, developers need to be able to quickly reset and re-create test and development environments. Repeatedly performing theses tasks is costly and error prone. Using domain templates, the Domain Template Builder, the Configuration Wizard, and the WebLogic Scripting Tool (WLST), automation scripts can easily be created for generating new domains, populating those domains with resources such as JDBC and JMS resources, deploying applications, and any other operation that can be accomplished with the Administration Console. Such scripts can be authored with standard editors, or with IDE support using Oracle Enterprise Pack for Eclipse. The Administration Console can also be used to record actions into a reusable WLST script, which provides a basis for customization, simplifying the script creation process.

Application Classloading

Classloading is a complex, and often misunderstood area of application servers. Fortunately, Oracle WebLogic Server provides several mechanisms that simplify configuration of application classloading.

First, Oracle WebLogic Server provides a way of sharing libraries across applications, which simplifies the deployment of such applications. This is useful in cases where the libraries themselves evolve at a different rate than the applications. It also removes the need to deploy the libraries with each application.

Next, Oracle WebLogic Server provides for application-level libraries. These libraries are deployed with each application and loaded through the standard Java EE classloading hierarchy. This means that these libraries are not shared by other applications deployed on the server - each application receives its own copy of the library. This is useful if applications deployed on the same server need to use different versions of the same library.

Oracle WebLogic Server has also simplified classloading through support of a filtering classloader. This classloader does not load classes itself, but prevents classes from being loaded by the parent classloader. The filtering classloader allows applications to override system level

classes, which is difficult to accomplish with other application servers. This capability is important for use cases where open source software components used in applications may conflict with different versions of those software components embedded in Oracle WebLogic Server. For instance, your application may need to make use of a different version of Xerces, Spring, Ant, or Commons-Logging. To accomplish this the filtering classloader would be configured to filter those classes from the system classpath. Instead those classes would be bundled and loaded from the application libraries.

Oracle WebLogic Server provides a Web-based Classloader Analysis Tool for analyzing and resolving classloading issues. In cases like that described above, when open source software used in applications and in Oracle WebLogic Server may conflict, the tool helps to detect such class conflicts, and proposes filtering classloader configurations to resolve them. We recommend that application development projects using open source software libraries use this tool in their development processes for early detection and resolution of classloading issues.

This tool should also be used when upgrading applications, or upgrading existing applications to newer versions of Oracle WebLogic Server, if these applications use open source software. In such upgrade cases, open source software versions used either in the server or in applications may themselves be upgraded. This may cause versions of open source software used in applications to diverge from the versions used in the server, potentially creating class conflict issues that did not exist before the upgrade. Use of the Classloader Analysis Tool will help to identify and resolve such issues during the upgrade process.

Native Cloud Management

Over many releases, Oracle WebLogic Server has provided powerful management tooling that simplifies management of application environments, and ultimately reduces the cost of maintaining the underlying application infrastructure. Native cloud management in Oracle WebLogic Suite 12.1.2 refers to management tooling enhancements that are built-in to (or native to) the Oracle WebLogic Server infrastructure, and that have been extended, in many cases, to provide consistent management tooling across the Web Tier and Coherence Tiers of the typical Oracle WebLogic Suite deployment topology. Native cloud management in Oracle WebLogic Suite 12.1.2 will enable management of massively scaled cloud infrastructures by providing common, consistent and therefore more efficient tooling across the web server, application server and data grid tiers.

Unified Installation and Patching

The underlying installation and patching technologies used by application infrastructure products becomes more important as users plan and implement cloud infrastructures. Simplified provisioning of new instances is one of the primary goals of cloud infrastructures, and therefore

cloud administrators will expect to have large numbers of installations to manage and maintain through patches. Use of consistent installation and patching tools simplifies management of such cloud environments.

Prior to Oracle WebLogic Suite 12.1.2, Oracle provided some integration of installation technologies across the suite of products. For example, Oracle TopLink is always installed as part of Oracle WebLogic Server 11g and 12.1.1, and Oracle WebLogic Server 11g and 12.1.1 installers included Oracle Coherence. However this integration was incomplete. Most notably, Oracle WebLogic Server used installation and patching technologies that were developed prior to the acquisition of the product by Oracle. This meant that Oracle WebLogic Server patching and installation technologies were different from the standard Oracle Universal Installer (OUI) and OPatch patching tools.

In Oracle WebLogic Suite 12.1.2, we have unified the installation and patching technologies across the components of Oracle WebLogic Suite. Oracle WebLogic Server, Oracle Coherence, Oracle TopLink, Oracle HTTP Server, Oracle ADF, Oracle JDeveloper and Oracle Enterprise Pack for Eclipse have all standardized on OUI and OPatch installation and patching technologies. This means that cloud administrators can use consistent technologies to deliver simplified and efficient installation and maintenance of cloud infrastructures based on Oracle WebLogic Suite.

WebLogic Management Framework

The WebLogic Management Framework comprises management capabilities that are delivered as part of Oracle WebLogic Server. The WebLogic Management Framework has been essential to delivering an efficient management environment that reduces cost of ownership. In Oracle WebLogic Suite 12.1.2, we have continued to extend the functionality provided by the WebLogic Management Framework, and the scope of the framework has been extended to cover management Oracle HTTP Server and Oracle Coherence.

The WebLogic Management Framework consists of the components below, which are described in the following paragraphs:

- Domain Configuration, Including Administration and Managed Servers
- Clusters
- Deployment Infrastructure
- Node Manager
- Configuration Wizard and Domain Template Builder
- WebLogic Administration Console and Oracle Fusion Middleware Control
- WebLogic Scripting Tool (WLST)
- WebLogic Diagnostic Framework, Java Mission Control and Java Flight Recorder

- RESTful Management Services

Domain Configuration, Including Administration and Managed servers

The fundamental unit of Oracle WebLogic Server configuration and management is the domain. A domain consists of a set of managed servers - Oracle WebLogic Server instances or processes that host applications - and an administration server that controls the configuration of all the servers within the domain. A domain contains one and only one administration server. A domain may comprise a single development server that combines administration server and managed server functions, or may comprise an administration server and many managed servers. Administration servers distribute domain configuration information to managed servers, and protocols built on top of standard JMX are used to ensure reliable communication of configuration changes throughout the domain. There is no runtime requirement for the administration server to be available – if the administration server is not available, applications deployed on managed servers will continue to provide application services uninterrupted. Managed servers running in these domains may or may not be clustered.

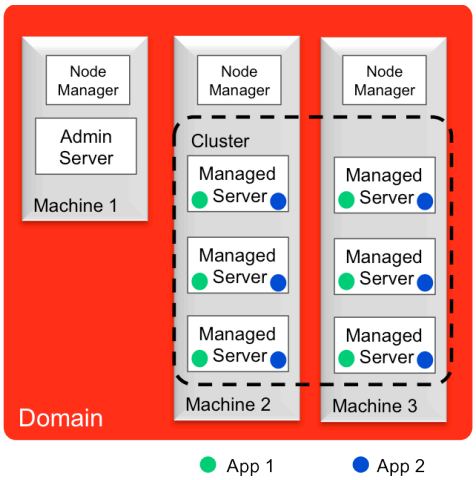


Figure 4: Domain Configuration with Administration Server, Clustered Managed Servers, Applications and Node Managers

Clusters

Oracle WebLogic Server clusters provide clients with a single logical view into applications that are distributed over multiple managed servers, and provide scalability and availability benefits for mission-critical applications. However, clusters also offer manageability benefits as well. Clustered servers are intended to be homogeneous or uniform – that is, they should run the same version of Oracle WebLogic Server, they should contain identical configurations (with a few exceptions discussed later in this document), and they should have the same applications

deployed. Many management operations such as resource configuration and deployment can be targeted at clusters to improve management efficiency.

Deployment Infrastructure

The Oracle WebLogic Server deployment infrastructure ensures that deployment archives such as WARs, EARs, RARs and JARs are made available to managed server instances that will host the applications, and are prepared for application execution. In clustered configurations, Oracle WebLogic Server deployment ensures that deployment is successfully completed on all servers within the cluster. Multiple staging modes for distributing deployment archives to managed servers are supported. Advanced deployment capabilities are also supported, such as production redeployment, which enables updates to application versions without interruption of application services to existing clients.

Node Manager

The node manager is an optional component of the WebLogic Management Framework. The node manager is a lightweight process designed to monitor and control server lifecycle operations, such as server startup and shutdown, for server instances on a given machine. The configuration model for the node manager in previous releases was to share node manager instances across domains. In Oracle WebLogic Server 12.1.2, the default configuration is a node manager instance per domain on each managed server machine. The Admininstration Server can control the startup and shutdown of server instances through the node manager, and the node manager can be configured to perform certain operations like auto-restart of failed server instances.

Configuration Wizard and Domain Template Builder

The Configuration Wizard is a GUI and command-line tool for initial creation of Oracle WebLogic Server domains, and domains for Oracle Fusion Middleware products that are built on Oracle WebLogic Server. The Configuration Wizard creates domains based on templates. The related Domain Template Builder enables end-user creation of templates themselves such that once a domain configuration is created, it can be captured in a Configuration Wizard template for recreation of that domain on different machines. This tooling provides an easy to use mechanism for initial domain creation, and a powerful capability for complex domain recreation and replication across cloud environments.

WebLogic Administration Console and Oracle Fusion Middleware Control

Oracle WebLogic Suite 12.1.2 provides two consoles for managing Oracle WebLogic Server domains at runtime. The primary console used by Oracle WebLogic Server administrators is the WebLogic Administration Console. The Administration Console is a Web Application that runs on the administration server. It provides full functionality for managing domain configuration, starting and stopping of servers, application deployment and monitoring of the domain environment. In addition Oracle WebLogic Suite 12.1.2 includes Oracle Fusion Middleware

Control, which is the strategic console for management of individual Oracle Fusion Middleware product domains. The scope of Oracle WebLogic Server management functions provided in Fusion Middleware Control is not yet equivalent to that provided by the WebLogic Administration Console, but significant areas are covered, such as server monitoring, lifecycle management, application deployment, and common configuration requirements.

WebLogic Scripting Tool (WLST)

The WebLogic Scripting Tool (WLST) is a Jython-based scripting environment that can be used to automate all aspects of Oracle WebLogic Server administration. This is a popular tool for automating Oracle WebLogic Server management, permitting, for example, modification of "empty" domains using WLST scripts to replicate a desired configuration, or executing a uniform set of management tasks across multiple domains to achieve a consistent change in the configuration of all of the target domains. As mentioned in the prior section on development, WLST scripts can be created manually, or by recording actions taken in the WebLogic Administration Console, or by authoring scripts in Oracle Enterprise pack for Eclipse.

WebLogic Diagnostic Framework, Java Mission Control, and Java Flight Recorder

The WebLogic Diagnostic Framework (WLDF) is a monitoring and diagnostic framework designed to collect and surface metrics that relate to server and application performance. Using WLDF you can create, collect, archive and analyze diagnostic data generated by a running server and applications. WLDF watches can be configured to send notifications based on metrics and rules you define. WLDF data can be graphically displayed and viewed in the Administration Console to enable administrators to monitor critical server metrics.

WLDF is also engineered to work with two related features of the Oracle JDK – Java Mission Control and Java Flight Recorder. Java Flight Recorder is designed to retain, in a rolling in-memory buffer, diagnostic information about server process execution over the past several minutes, including event data generated by WLDF. The Flight Recorder buffer can be persisted to disk at any time for post-incident analysis using the Java Mission Control GUI, enabling analysis of what was occurring in the server leading up to the incident in question. WLDF and Flight Recorder provide unique diagnostic capabilities, especially for problems that may recur periodically, but that are difficult to reproduce for analysis. Using WLDF and Flight Recorder you can, for example, configure a watch based on a server metric that is associated with the problem. Resulting WLDF notifications can trigger the capture of a WLDF diagnostic snapshot, including the persisted Flight Recorder buffer. This approach automatically captures a detailed picture of what was occurring in the server at the instant of the event, and what had been occurring in the server leading up to the event, hopefully revealing the underlying cause of the problem.

RESTful Management Services

A new management capability in recent Oracle WebLogic Server releases is RESTful Management Services. Oracle has recognized that use of RESTful interfaces is an increasingly

popular means to accessing application data. In the case of RESTful Management Services, this means providing RESTful access to Oracle WebLogic Server monitoring data, including monitoring servers, clusters, applications, and data sources in a domain. The feature must be explicitly enabled, and access to the interfaces is restricted based on standard administrative roles. We believe this feature will useful for building lightweight management interfaces that can be used to monitor whether applications, servers, and server resources are running or healthy. JSON, XML, HTML representation formats are supported.

This summary of WebLogic Management Framework functionality and capability is not intended to be exhaustive. Rather it is intended to illustrate the scope of native management capabilities provided with Oracle WebLogic Server. These capabilities have been proven through years of use in customer production environments, and have been applied to creation and management of private clouds and public cloud environments as well. The Oracle Java Cloud Service, which is built on Oracle WebLogic Server and is part of the overall Oracle Cloud offering, uses and leverages these same capabilities. The next sections of this document describe how we have extended the scope of the WebLogic Management Framework to cover management of other components of Oracle WebLogic Suite 12.1.2, to provide the foundation for native cloud management of full Oracle WebLogic Suite 12.1.2 infrastructures.

Managed Coherence Servers

Managed Coherence Servers is a new feature of Oracle Coherence and Oracle WebLogic Suite 12.1.2. It provides the option for Oracle Coherence instances to be managed by the WebLogic Management Framework. Note that this is an optional capability – Oracle Coherence users who wish to manage Oracle Coherence clusters without use of the WebLogic Management Framework are not obligated to use it. However we believe that users who wish to use Oracle WebLogic Server and Oracle Coherence in combination, or users who wish to use an Oracle-supplied management infrastructure for Oracle Coherence will adopt Managed Coherence Servers.

The fundamental design approach behind Managed Coherence Servers is to run Oracle Coherence clients or servers within an Oracle WebLogic Server managed server process – either a managed server with Java EE applications deployed, or a managed server with no Java EE applications deployed. This design approach enables the Oracle Coherence instance running within the managed server to leverage all of the benefits of the WebLogic Management Framework, including the following functions:

- Managed Coherence Servers configurations can be created using the Configuration Wizard, and replicated as required from user-created domain templates.
- Domains containing Managed Coherence Servers can be started and stopped using standard WebLogic Management Framework lifecycle tools. For example Oracle Coherence instances and clusters can be started via the Administration Console or WLST scripts communicating through node managers.

- Node managers can be configured to automatically restart Managed Coherence Server instances.
- Configuration changes to running servers can be executed through the Administration Console, or through JMX, or through WLST scripts.
- Oracle Coherence applications can be deployed through WebLogic Management Framework deployment infrastructure. A new deployment archive type has been created called the Grid Archive (GAR) that is recognized by the WebLogic deployment infrastructure, to prepare applications for execution on Managed Coherence Servers instances.
- Managed Coherence Servers and clusters can be managed using the Administration Console, JMX, Oracle Fusion Middleware Control, Oracle Enterprise Manager and WLST.

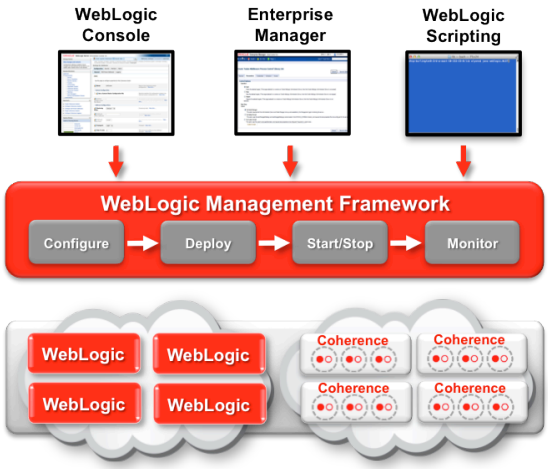


Figure 5: Managed Coherence Servers – Unified Management for Oracle WebLogic Server and Coherence

Managed Coherence Servers supports a diverse set of Oracle WebLogic Suite topologies, including standalone Oracle Coherence clusters (with no Java EE applications deployed), clustered Java EE applications with storage disabled Oracle Coherence instances accessing storage enabled Oracle Coherence instances all running in a single Oracle Coherence cluster, multiple Java EE clusters accessing data in a shared Oracle Coherence cluster, and other configurations.

For users concerned about the overhead associated with running Oracle Coherence within a managed server, we believe this overhead is minimal. Internal measurements indicate that the incremental memory consumption from a managed server is approximately 70MB – 80MB, or approximately 2% of a 4 GB heap. We believe Managed Coherence Servers is a powerful capability that Oracle WebLogic Suite users will rapidly adopt.

Oracle HTTP Server Management

Prior versions of Oracle HTTP Server used a separate framework for configuring and managing Oracle HTTP Server instances. In this 12.1.2 release, the Oracle HTTP Server instances are managed through the WebLogic Management Framework as follows:

- Oracle HTTP Server configurations can be created in "Oracle WebLogic Server domains" or in "Standalone domains". These domain configurations can be created with the Configuration Wizard.
- Oracle WebLogic Server domain configurations support the ability to manage Oracle HTTP server instances through Fusion Middleware Control or WLST scripts running on an Oracle WebLogic Server administration server. Such configurations require full FMW infrastructure installations that include not only Oracle WebLogic Server and Oracle Coherence but ADF and Fusion Middleware Control as well. Process control of Oracle HTTP Server is managed through the node manager. Domain configurations of this type may also include managed servers hosting Java EE applications and Managed Coherence Servers.
- Standalone domain configurations are intended as configurations for system (non-Java) components only, such as Oracle HTTP Server. Use of the standalone domain configuration option would be appropriate when use of Fusion Middleware Control is not desired or required. Standalone domains can be managed using the WLST command line and node manager.

Because the same WebLogic Management Framework technology is being used to manage Oracle WebLogic Server, Oracle TopLink, Oracle Coherence and Oracle HTTP Server, Oracle WebLogic Suite offers the unique ability to provide a consistent, proven native management framework across the major elements of the typical Oracle WebLogic Suite topology. Such an architecture offers compelling manageability benefits, including reliability, efficiency and reduced cost of ownership, especially for customer infrastructures, like private cloud infrastructures, that must scale to meet business user demands.

Elastic JMS

In addition to extending the WebLogic Management Framework to support additional products beyond Oracle WebLogic Server, we continue to improve to the WebLogic Management Framework to increase the efficiency of managing Oracle WebLogic Server itself. An example of such an improvement delivered in Oracle WebLogic Server 12.1.2 is a new configuration capability named Elastic JMS, which simplifies JMS configuration, and scaling of JMS configurations.

Prior to Oracle WebLogic Server 12.1.2, JMS server and JMS store (either file store or database store) configurations could only be targeted at individual managed servers. Furthermore, subdeployments, or the targeting of queues or topics to JMS servers, also required JMS server-specific configuration. What this meant is that even when clustered JMS server configurations

were being created, the administrator was required to execute server-specific configuration steps to configure JMS in these clusters. While such configurations can be achieved with the Administration Console or WLST, creating such configurations is inherently more complex than desirable, especially when scaling JMS configurations to large numbers of managed servers, JMS servers and subdeployments.

Elastic JMS provides a new option to target configuration of both JMS servers and JMS stores to a cluster[1]. This greatly simplifies the process of JMS cluster configuration. In a cluster of N managed servers, only one JMS Server and one JMS store must be targeted, instead of N. Configuration of subdeployments is also simplified, as subdeployments can be targeted at the single, clustered JMS Server. Not only the creation of these configurations, but also the scaling of these configurations is also simplified. As managed servers are added to the clustered configuration, the configuration of JMS servers, stores and subdeployments are automatically scaled to match the new number of managed servers. The end result is more efficient management of JMS configurations, and a significant enhancement to the WebLogic Management Framework for managing cloud environments.

Oracle Enterprise Manager Cloud Control

All of the native cloud management features and capabilities described in the preceding sections can be supplemented through use of Oracle Enterprise Manager Cloud Control. Oracle Enterprise Manager is the strategic management platform across the entire Oracle product portfolio, providing end-to-end management of the Oracle stack from applications to disk. Oracle Enterprise Manager Cloud Control provides support for Oracle WebLogic Server through two Management Packs.

Oracle WebLogic Server Management Pack Enterprise Edition provides unique value by providing management and monitoring support across multiple Oracle WebLogic Server domain configurations, all through a single management interface. For cloud infrastructures that are expected to scale to large numbers of domains, providing a central console for monitoring of all domains in the environment enables more efficient management at scale. Performance monitoring and diagnostics capabilities enable monitoring of application services, and the ability to drill down and identify underlying problems if service levels are not being met. Configuration management capabilities enable administrators to track and maintain configurations across the environment, ensure compliance with standards, and minimize downtime resulting from inappropriate configurations. Lifecycle management enables automation of tasks such as

[1] Elastic JMS does not yet include support for advanced JMS features such as Unit of Order, Unit of Work, Store and Forward services, and Server and Service Migration.

distributed installation and patching which are essential to maintaining a robust cloud infrastructure. This brief description only begins to illustrate the breadth of functionality provided.

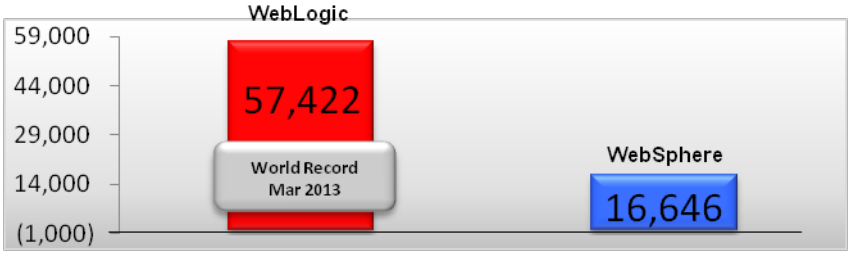
Oracle Cloud Management Pack for Fusion Middleware is an add-on to Oracle WebLogic Server Management Pack Enterprise Edition. It enables IT administrators to create a self-service private cloud infrastructure, where business users are granted the ability to provision Oracle WebLogic Server domains for development, QA or production purposes. Simplified management interfaces make it easy for business users to perform basic management tasks required for self-service management of services they have created. Metering capabilities are provided to enable IT organizations to charge back end users based on predefined metric indicating their consumption of cloud resources. Users who are developing strategies for managing cloud infrastructures should evaluate the capabilities of Enterprise Manager Cloud Control.

Mission Critical Cloud Platform

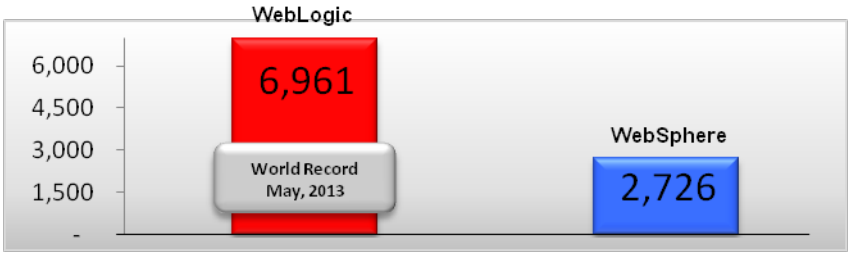
Providing a modern, productive development platform and efficient, native cloud management capabilities is necessary, but not sufficient, to meet the business demands that will be placed on cloud infrastructures. In nearly every business, the quality of service experienced by the customer or the end user is one of the most important determinants of success. Applications and services deployed on cloud infrastructures must provide the highest service levels in terms of performance, availability, scalability and reliability. They must rely on the characteristics of the cloud platform to support these service levels. The goal of Oracle WebLogic Suite 12.1.2 is to provide the leading application service levels at the lowest risk and cost.

Proven to Outperform

Better application infrastructure performance in the middle tier means not only delivering the best possible performance to users, but also enabling reduced investment in hardware resources. Oracle WebLogic Server is the proven performance leader in the application server marketplace and continues to set world records in multiple categories as measured by the industry-standard SPECjEnterprise 2010 benchmark. The charts below highlight key recent results:



Highest Overall EjOPS: 3.4X more workload than IBM



Highest Overall EjOPs per Processor Virtualized: 2.2X more workload per processor than IBM;

Oracle WebLogic Server beats IBM WebSphere Application Server in multiple categories

(source: http://www.spec.org/jAppServer2010/results/).

These results reflect years of product optimizations to meet ever-growing performance requirements, and are also consistent with the outstanding performance levels achieved by our customers. Your cloud applications can benefit from these performance capabilities that are built into Oracle WebLogic Server.

Work Managers and Server Performance

One of the important capabilities of Oracle WebLogic Server is not just that it outperforms other servers in the marketplace. In addition to raw performance capability, Oracle WebLogic Server provides self-tuning capabilities that enable users to achieve optimized performance on Oracle WebLogic Server, without complex, iterative system tuning procedures. The self-tuning capabilities of Oracle WebLogic Server are based on Work Managers.

In order for an application server to perform optimally, it must be configured with the correct number of application request threads. Because the number of request threads is largely dependent on hardware, applications, and the underlying JVM, it is difficult to determine the exact number of threads needed in production to achieve optimal application throughput. In Oracle WebLogic Server there is a single thread pool, in which all types of work are executed. Oracle WebLogic Server uses Work Managers to prioritize work based on policies defined against runtime metrics, including the actual time it takes to execute a request and the rate at which requests are entering and leaving the pool.

Work managers can be configured at the domain, application, or module level using the Administration Console. The various policies and constraints that can be employed are:

- Fair-share-request: Specifies the average thread-use time required to process requests, which the work manager then uses to ensure that requests for different modules get their specified share of compute time.
- Response-time-request: Specifies a response time goal in milliseconds, which the work manager then uses to throttle different types of requests to prevent starvation of certain requests.

- Context-request: Assigns request classes to requests based on context information, such as the current user or the current user's group, which can then have additional constraints applied.
- Max-threads-constraint: Prevents the server from scheduling additional requests of this type until the number of concurrent executions falls below the limit. The server then schedules work based on the fair-share or response-time goal.
- Min-threads-constraint: Guarantees the number of threads the server will allocate to affected requests to avoid deadlocks.
- Capacity: Causes the server to reject requests when it has reached its capacity.

Combining the preceding policies and constraints make for a self-tuning, dynamically adjusting thread pool. The thread pool will schedule work as well as increase and decrease the number of threads to achieve optimal throughput, subject to concurrency constraints. Work managers monitor the overall throughput, and uses collected data to determine if the thread count must be adjusted to achieve optimal performance. The end result is optimized performance, and superior application services, with reduced administration overhead required for manual, iterative tuning.

Oracle WebLogic Server Clustering

Although individual Oracle WebLogic Server managed servers are highly performant, the overwhelming majority of Oracle WebLogic Server applications are deployed on multiple servers to improve application service levels.

Oracle WebLogic Server clusters consist of multiple managed servers running simultaneously and working together to provide increased throughput, scalability and reliability. From a client's perspective, a cluster appears to be a single Oracle WebLogic Server instance. The server instances that constitute a cluster can run on the same machine, or be located on different physical or virtual machines. Cluster capacity can be scaled by increasing the number of servers in the cluster.

The key benefits of clustering are to provide:

- Scalability: Applications deployed on Oracle WebLogic Server clusters can be scaled to meet application performance demands. Many customers use clusters with 2-4 managed servers, and Oracle generally recommends careful stress testing when scaling clustered Java EE applications beyond 32 managed servers. That said, there is no specific technical limit on cluster scaling and customers have run in production with larger cluster configurations. Oracle WebLogic Server clusters can be scaled to meet the overwhelming majority of application performance and throughput requirements, and to exceed the performance levels of other application server technologies.
- High-availability: Applications deployed in clusters can provide continuous application processing through a number of failover mechanisms, even in the event of a managed server or machine failure.

Key capabilities of clustering include:

- Session failover: With clusters, HTTP session and EJB session state is automatically replicated from the primary managed server that the client is accessing, to a secondary server within the cluster. In the event of a failure on the primary server, clients can be transparently reconnected to the secondary server, where they can continue application processing without interruption or loss of session state.
- Load balancing: In order for applications to scale properly, work and associated network communications must be evenly distributed across the compute and network resources. Oracle WebLogic Server provides load balancing at multiple levels to ensure that the appropriate assignment of work takes place.

Oracle WebLogic Server has the ability to cluster the various Java EE application resource including Servlets, JSPs, EJBs, RMI objects, JMS Destinations and JDBC resources.

Server and Service Migration

An advanced feature of Oracle WebLogic Server clustering is Server and Service migration. This is a cluster failover technology that is applicable to singleton services deployed in a cluster.

Singleton services are services that exist on only one node of the cluster at any given point of time. One such example is a JMS service, where the service must ensure that only one copy of the application message is persisted to storage for every persistent send operation from the producer. Unfortunately, problems with the managed server or machine that hosts the singleton service can render the service unusable or make it unavailable until the problem is corrected. The current set of singleton services that are currently offered by Oracle WebLogic Server can be grouped into the following broad categories:

- Persistent store service
- JMS-related services
- JTA-related services
- User-defined singleton services

These services are susceptible the type of failure scenario described above. To address this susceptibility and the potential impact on application service levels, Oracle WebLogic Server offers a solution called migration. Migration in Oracle WebLogic Server is the process of moving a clustered Oracle WebLogic Server instance, or a subsystem component running on a clustered instance, elsewhere in the event of failure. The process of moving an entire server instance from one physical machine to another upon failure is called whole server migration (WSM). Moving only the affected subsystem services from one server instance to another running server instance is called service migration.

In whole server migration, when a migratable server becomes unavailable for any reason, for example, if it hangs, loses network connectivity, or its host machine fails, the server instance is automatically migrated. The migrated server is automatically restarted on the same machine if possible. If the migratable server cannot be restarted on the machine where it failed, it is migrated to another machine. In addition, an administrator can manually initiate migration of a server instance.

Service Migration provides the necessary infrastructure to migrate only the failed subsystem services from an unavailable server instance to an available server instance. Although manual migration of services is possible, the Automatic Service Migration (ASM) framework proactively monitors the health of singleton services and automatically migrates failing services to available server instances in the cluster, reducing the time taken to perform the migration and increasing the overall availability of these services.

Rolling Upgrade

Rolling upgrade is the process of upgrading a running Oracle WebLogic Server cluster with a one-off patch, or Patch Set Update (PSU), without shutting down the entire cluster or domain, allowing for uninterrupted service. During the rolling upgrade of a cluster, each server in the cluster is individually upgraded and restarted while the other servers in the cluster continue to host your application. This process can be manually scripted, or for patch application, can be automatically orchestrated using the Oracle WebLogic Server Management Pack Enterprise Edition. Rolling upgrade provides yet another mechanism by which Oracle WebLogic Server clusters provide uninterrupted service, even in the face of practical requirements such as server maintenance.

Dynamic Clusters

Dynamic Clusters is an important new clustering configuration option provided in Oracle WebLogic Server 12.1.2, and is specifically targeted at the requirements of cloud infrastructures and potentially unpredictable application performance requirements. Dynamic Clusters provide a new configuration option to simplify the initial configuration of clusters, and the scaling of clusters to meet application performance requirements. Note that standard “configured clusters”, as supported in prior Oracle WebLogic Server releases, continue to be supported, along with this new clustering configuration option.

Prior to Oracle WebLogic Server 12.1.2, configured clusters required administrators to define specify server-specific configuration information – managed server names, listen ports, and machine mappings - for individual managed servers in a cluster. Therefore, while clusters are intended to be homogeneous with respect to server configuration, configured clusters have required server-specific settings. This is true not only for creating initial cluster configurations, but also for scaling cluster configurations. Scaling configured clusters has therefore been more complex than configuring a single server and requesting N copies of the server in a cluster.

Dynamic Clusters change this. Dynamic Clusters enable users to create cluster configurations based on a server configuration defined in a server template. Having defined such a configuration, the user specifies how many managed servers are to be created in the cluster. Oracle WebLogic Server configuration takes responsibility for automatically generating unique managed server names, machine mappings and listen ports in order to create the number of managed servers required in the cluster. When adding managed servers to a cluster to scale up the configuration, the user only needs to change a single parameter, the number of managed servers in the cluster, and the additional managed servers are available in the configuration to be started.

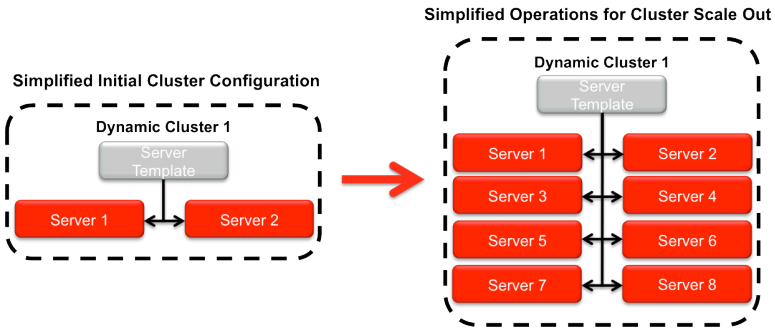


Figure 6: Dynamic Clusters – Elastic Scaling for Cloud Environments

The creation and scaling of Dynamic Clusters configurations can be easily accomplished in standard tools such as the Administration Console, Oracle Fusion Middleware Control, and WLST. Note that Dynamic Clusters use the same underlying runtime implementation and provides the same service levels as standard, "configured" clusters – Dynamic Clusters is a new cluster configuration option, not a new implementation of Oracle WebLogic Server clustering.

Dynamic Clusters have been designed to work with Managed Coherence Servers and Elastic JMS, such that Managed Coherence Servers and Elastic JMS configurations can be configured and scaled with Dynamic Clusters. However, Elastic JMS support in Dynamic Clusters does not yet include support for advanced JMS features such as Unit of Order, Unit of Work, Store and Forward services, and Server and Service Migration. With the exception of users who require these advanced JMS features, we believe the simplicity of initial cluster configuration combined with the inherent scalability of Dynamic Clusters will make this feature broadly popular among Oracle WebLogic Server users, especially users creating new cloud infrastructures.

Oracle WebLogic Server and Oracle Coherence Integration

Oracle Coherence is the industry's leading in-memory data grid solution that enables applications to predictably scale by providing fast, reliable and scalable access to frequently used data. The key to the linear scalability of Oracle Coherence is the partitioning of data across the nodes in the data-grid. In other words, each node does not cache a complete set of the data yet to external users it

appears as a single coherent clustered cache. Oracle Coherence can be used with Oracle WebLogic Server applications to increase application performance, throughput and scalability.

Oracle Coherence provides APIs that enable developers to write applications and leverage the benefits of the data grid in powerful ways. All of the APIs provided by Oracle Coherence Enterprise Edition are available to Oracle WebLogic Suite users. However, in addition to the support for custom application development that is provided, Oracle provides two mechanisms for Oracle WebLogic Server users to leverage Oracle Coherence without any application programming. These mechanisms are:

- Coherence*Web: Support for HTTP session management within Oracle Coherence
- TopLink Grid: Support for JPA applications using TopLink to store JPA entities within Oracle Coherence

Coherence*Web

Coherence*Web is designed to extend and improve the performance of Java EE Web applications that make use of HTTP session state by offloading session management from the application server infrastructure. There are several scenarios where offloading session state management from the application server makes sense. Applications making use of large sessions can quickly limit the scalability of an application, as session data can compete for application server memory usage, and/or large heaps created to retain session data can create other trade-offs such as garbage collection latency. In these cases, Coherence*Web can be used to store HTTP session state in the data grid, including HTTP session state replication. This allows for very fast access of session data in a high availability solution, while properly leveraging the data grid for the work for which it was designed. Coherence*Web works equally well in cases where session state must be shared across multiple applications.

TopLink Grid

Rather than using the typical local cache infrastructure provided by JPA implementations, the TopLink Grid implementation within Oracle WebLogic Server enables TopLink based JPA applications to seamlessly take advantage of Oracle Coherence as a high performance shared cache for JPA entities. The JPA programming model is augmented by a number of annotations enabling developers to easily and quickly put their JPA applications on the grid. The end result is dramatically increased performance and increased scalability for JPA applications.

The Coherence*Web and TopLink capabilities described above, when combined with Managed Coherence Servers and use of the WebLogic Management Framework, provide a level of integration between application server and data grid technologies that is unmatched in the industry. This integration gives users the ability to easily and efficiently utilize Oracle Coherence's scalability, reliability and performance benefits in Oracle WebLogic Suite applications, and in mission-critical cloud infrastructures built on Oracle WebLogic Suite.

JMS Messaging

Oracle WebLogic Server JMS has been designed and developed to provide all the features necessary for the most demanding messaging applications. Above and beyond the JMS specification, Oracle WebLogic Server JMS incorporates value-added reliable messaging features including:

- Unit of Order: Processing messages out of order can be disastrous to the integrity of an application. This feature guarantees that all messages are processed sequentially in the order that they are placed on the destination.
- Unit of Work: Where applications require an even more restricted group than that provided by the Unit of Order feature, Unit of Work can be used to group messages as a unit allowing message consumers to process the collection of messages as a unit. A message within a Unit of Work does not become available for processing by the recipient until all messages scoped within the Unit of Work are all available.
- Store and forward messaging: This construct removes significant complexity from the development of message-based applications by having the framework address reliability concerns. A messaging client should have the option of sending a message once and only once, even if the associated message provider is down. The store and forward messaging feature will store messages destined for a system that was unavailable at the time they were sent until the system becomes available, at which time the messages will be forwarded.
- Distributed destinations: A distributed destination is a set of destinations (distributed queues or topics) that are accessible to clients as if they were as a single, logical destination. Distributed destinations are typically used in high availability environments or where message processing must be load balanced across a set of compute resources.
- JMS Server and Service Migration: Migration in Oracle WebLogic Server is the process of moving a clustered managed server, or a subsystem component running on a clustered managed server, elsewhere in the event of failure. Server and Service migration provide high availability for JMS configurations, as described in more detail in an earlier section of this document.

Oracle WebLogic Server JMS also includes support for the Oracle AQ JMS implementation based on the Oracle Database. Oracle AQ JMS leverages Oracle Database availability and reliability features to provide enterprise-class messaging services. Oracle AQ JMS also enables interoperability with other Oracle AQ JMS applications, and fully supports JTA transactions.

Active GridLink for RAC

Oracle WebLogic Server multi data sources have been supported for use with Oracle Database RAC systems for years, have proven benefits in production, and will continue to be supported.

However, beginning in Oracle WebLogic Server 11g, a new feature for Oracle WebLogic Server and Oracle Database RAC integration was delivered, and is referred to as Active GridLink for RAC.

Active GridLink for RAC, available in Oracle WebLogic Suite, is the preferred mechanism for integrating with Oracle Database RAC systems. Active GridLink for RAC simplifies configuration and optimizes Oracle Database RAC connectivity. The result is higher throughput and faster response time for enterprise Java applications that are involved in intensive database work.

Oracle Active GridLink for RAC brings in the following capabilities to GridLink data sources:

- Runtime Connection Load Balancing – GridLink data sources are made load-aware and notified proactively of the workload on RAC nodes and can balance connections to least loaded nodes. This results in much more even utilization in your RAC environment and improved overall application performance.
- XA Affinity – Transaction affinity for global transactions ensures all the database operations for a global transaction performed on an Oracle RAC cluster are directed to the same Oracle RAC instance. The first connection request for an XA transaction is load balanced using runtime connection load balancing and is assigned an Affinity context. All subsequent connection requests are routed to the same Oracle RAC instance using the affinity context of the first connection.
- Single data source for an entire RAC cluster - if your RAC clusters are large and you have many different domains, GridLink data sources can dramatically simplify configuration.
- Fast Connection Failover - event based notification changes in the RAC cluster and graceful recovery of database connection. This ensures that the connection pool in the GridLink data source contains valid connections without the need to poll and test connections.
- Support for Oracle Single Client Access Name (SCAN) - frequently asked for by our 12c Database customers. A GridLink data source containing SCAN addresses does not need to change if you add or remove Oracle RAC nodes.

The majority of the Oracle WebLogic Server customer base uses the Oracle Database. Integration features such as Active GridLink for RAC enable customers to improve application performance, availability and manageability by leveraging the cross-product integration provided by Oracle.

Database 12c Integration

Oracle has recently released Oracle Database 12c, the latest version of the industry's leading database technology, and the database that is most commonly used with Oracle WebLogic Suite applications. Oracle recognizes the critical importance of integrating with the database technologies that host application data, and the customer value that can be delivered by leveraging

the capabilities delivered in enterprise database technology. Oracle continues to enhance the integration of Oracle Database and Oracle WebLogic Server for this reason.

More specifically, Oracle WebLogic Server 12.1.2 has been integrated with Oracle Database 12c to improve the availability, performance, and multi-tenancy capabilities of cloud infrastructures built on these products. The majority of Oracle WebLogic Suite customers will ultimately use Oracle Database 12c. An investment in Oracle WebLogic Suite is an investment in an infrastructure that will leverage the new database capabilities in the following areas:

- Application Continuity: A common issue encountered in application environments is the loss of database connections. Loss of connections may be caused by transient problems such as temporary network outages, or failover that occurs across nodes in a RAC cluster. Even if the causes are transient, and connections can be re-established, loss of such connections causes exceptions, which must be handled by the application or surfaced back to end users, impacting application reliability. Oracle WebLogic Server 12c transparently leverages application continuity to address this issue with both GridLink and generic data sources. It provides continuous application services to end users, even when database connections are temporarily lost. Application Continuity automatically recreates lost connections, and replays database requests in process – including read and write operations - transparent to the application and the end user, providing highly reliable application services to the end user without any programming required by the developer.
- Database Resident Connection Pooling: Maintaining dedicated connections between application servers and databases consumes database resources. In cloud environments, where large numbers of domains, clusters and servers may ultimately be created and connected to database instances, the resource consumption associated with connection management can create scaling problems at the database layer. Scaling issues can occur even if many of the connections are idle at any given time.

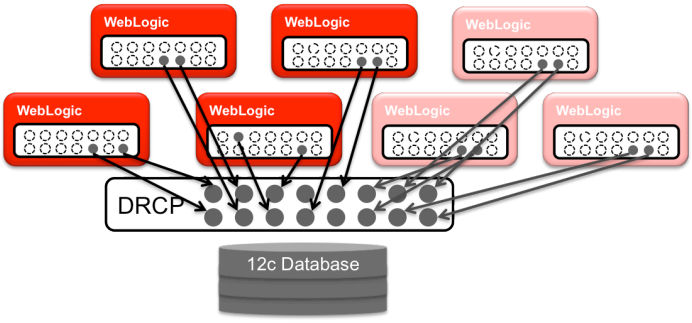


Figure 7: Database Resident Connection Pooling Enables Connection Sharing and Efficient Use of Resources

Database Resident Connection Pooling addresses this problem by creating a reusable pool of connections at the database layer that is shared across application instances. Oracle WebLogic Server 12.1.2 has been integrated to use and support Database Resident Connection Pooling. When application server connections are closed, they are returned to the shared connection pool. This results in much better scaling at the database tier, as application server instances are added, and better scaling of cloud infrastructures as new cloud services are provisioned in the cloud environment.

- Pluggable Databases: Oracle Database 12*c* introduces the concept of a Pluggable Database, which represents a single virtual database that could be dedicated to a specific tenant. Multi-tenant Container Databases can host multiple Pluggable Databases, enabling database consolidation for cloud environments, and providing significant management efficiencies at the database layer, while retaining the data isolation for tenants using dedicated Pluggable Databases. Oracle WebLogic Server 12.1.2 not only supports the use of Pluggable Databases in Oracle WebLogic Server applications, but also leverages Pluggable Database features to provide multi-tenant application support. Using a single datasource within an Oracle WebLogic Server configuration, a multi-tenant application can dynamically switch between tenant-specific Pluggable Databases based on tenant identity, providing isolation of tenant data being accessed from a single multi-tenant application. For organizations seeking to leverage the efficiencies of Pluggable Databases at the database tier, or seeking to leverage multi-tenancy capabilities in cloud applications, Oracle WebLogic Suite 12.1.2 provides the integration to meet their requirements.
- Global Data Services: Global Data Services is feature of Oracle Database 12c that enables the central management of database services in multi-site environments on a global scale. Services that are created and accessed by applications can be migrated across sites based on site workload and availability. Oracle Data Guard or Oracle Golden Gate is used to replicate data across sites to support migration and failover scenarios. As more and more services are added, scalability can be achieved by adding additional RAC clusters to the Global Data Services environment – in effect creating a highly available distributed database cloud that can be leveraged by applications. Oracle WebLogic Server 12.1.2 provides direct integration to Oracle Database 12c Global Data Services via Active GridLink for RAC. GridLink datasource configurations specify primary and secondary regions to access services. As services are migrated, Oracle RAC Fast Application Notification (FAN) events are recognized by GridLink datasources, and connections are automatically re-established to the migrated service. This automatic migration capability allows Oracle WebLogic Suite users to fully leverage Global Data Services.

In summary, Oracle Database 12c contains many powerful new features, and we expect strong adoption of this latest database version in enterprise customers. Oracle WebLogic Suite 12.1.2 provides significant value-added integration capabilities that will enhance the availability, multi-tenancy, and scalability of cloud infrastructures.

Exalogic Elastic Cloud Software Optimizations

Oracle Exalogic Elastic Cloud is designed to provide enterprises with a foundation for secure, mission-critical private cloud capable of unmatched scalability, performance, and manageability. Oracle Exalogic Elastic Cloud Software includes a number of optimizations and enhancements made to the core products within Oracle WebLogic Suite, the essential Java foundation on which Oracle‘s next-generation applications are being developed.

The Exalogic Elastic Cloud Software includes a set of enhancements to Oracle WebLogic Suite, for optimized performance when running on Exalogic hardware. The software optimizations address performance limitations that become relevant when the software is running on Exalogic's high-density computing nodes and fast-networking InfiniBand switches. These software optimizations utilize the benefits of this high-end hardware to the maximum, resulting in a balanced hardware-software engineered system.

Examples of these optimizations are as follows:

- Increased Scalability, Throughput and Responsiveness: Improvements to Oracle WebLogic Server networking, request handling and thread management mechanisms enable better scalability and performance on Exalogic compute nodes connected over the internal InfiniBand network fabric.
- Superior Session Replication Performance: Oracle WebLogic Server session replication mechanism is improved to utilize the large InfiniBand bandwidth that is used between managed servers for parallel connections over the network.
- Active GridLink for RAC: Optimized Oracle WebLogic Server and Oracle Database RAC Integration for more reliable database interaction.
- Oracle Coherence Message Bus: Oracle Coherence integration with the Infiniband Message Bus provides extremely high performance, low latency messaging using Remote Direct Memory Access (RDMA) across Oracle Coherence instances.
- Reduced Response Times for Exalogic to Exadata Communication: In situations where an Exalogic system is directly connected to an Exadata system, using InfiniBand, a “native” InfiniBand networking protocol called SDP is used to interact with the Oracle RAC database on Exadata. This results in low latency for request-response times for calls to the database. The performance gain is most significant when large results sets are transferred from the database. The net effect is applications are able to respond to client requests faster, leading to overall performance gain for enterprise Java applications.
- New in Oracle WebLogic Suite 12.1.2 are optimizations to the JMS subsystem for Exalogic. Internal performance benchmarks have identified that for JMS-intensive applications on Exalogic systems, the CPU processing capability exceeded the I/O capacity of the Exalogic storage subsystem. This created a system bottleneck for JMS messages being persisted to disk. Oracle WebLogic Server 12.1.2 contains an optimization on Exalogic Systems, that

causes JMS messages to be compressed before being persisted to disk, alleviating the storage processing bottleneck and increasing performance of JMS-intensive applications by up to 3x.

The combination of hardware and software results in substantial performance gains for Java-based applications running on Oracle WebLogic Suite 12c, making Exalogic an ideal platform for delivering mission-critical services to cloud applications.

Upgrading to Oracle WebLogic Suite 12.1.2

Oracle WebLogic Server users will experience very high compatibility levels when upgrading applications from prior releases of Oracle WebLogic Server to Oracle WebLogic Server 12.1.2. Application compatibility, domain compatibility and cross-version interoperability are provided to ensure a smooth upgrade path. Users should consult product documentation when planning upgrades.

Oracle SmartUpgrade, which supports migration of Oracle Containers for Java (OC4J) applications to Oracle WebLogic Server, has been updated to support direct upgrade of OC4J applications to Oracle WebLogic Server 12.1.2.

Summary

Oracle WebLogic Suite 12.1.2 is the latest release of the flagship edition of the Oracle WebLogic Server product line. It includes updated versions of Oracle WebLogic Server, Oracle Coherence Enterprise Edition, Oracle Web Tier, Oracle TopLink, Oracle Java Mission Control and Flight Recorder, and other component technologies. Oracle WebLogic Suite 12.1.2 builds on the capabilities of prior releases to provide a modern development platform, native cloud management and a mission critical cloud platform that make it the ideal application platform for customers planning their cloud infrastructures. We look forward to your usage of the product and your feedback.

Oracle WebLogic Suite Technical White Paper
July 2013
Author: Will Lyons
Contributing Authors: Cloud Application Product Management and Marketing team

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

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