



An Oracle White Paper  
August 2011

# Configuring Fusion Middleware to Scale for Oracle Student Learning

## Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Introduction .....	1
Overview .....	2
OSL Deployment Architecture.....	3
Test Overview .....	4
WebLogic Node Allocation .....	5
Recommendations .....	5
JVM Configuration.....	6
Recommendations .....	6
Observations on JVM.....	7
Caching in the Middle Tier .....	7
Recommendations .....	7
Observations on Caching in the Middle Tier .....	8
Scenarios Considered.....	9
Additional Recommendations: Database Configuration.....	10
Conclusion .....	10
Appendix: Components of an OSL deployment environment .....	11

## Introduction

Oracle Student Learning (OSL) is an enterprise-class software solution for K-12 schools that has been designed to support contemporary, 21st century paradigms of learning.

A deployment of OSL may need to support thousands of concurrent users. Hence, the deployment architecture must be able to sustain acceptable response times as and when the number of concurrent users grows.

This paper is based on a scalability and performance case study that involved testing the performance of OSL with tens of thousands of concurrent users. The testing focused mainly on the performance and scalability of the WebLogic middleware tier in the deployment architecture, and how to configure and deploy the middleware. The principal purpose of this paper is to present deployment recommendations for scaling up to support more than 10,000 concurrent OSL users.

## Overview

The Oracle Student Learning application is based on Oracle's latest Fusion technology stack using Oracle Database 11g and WebLogic Server 11g (WLS). When serving thousands of concurrent users, OSL requires multiple middle tier nodes as well as multiple database nodes. There are many ways to deploy and configure applications so that users see current information and can update information without blocking each other. The testing evaluated caching of data, coordinating of information between middle tier nodes, optimizing of memory allocations in the Java virtual machines (JVMs), and other aspects of the middle tier deployment and configuration.

The main result was a determination that OSL scales linearly when there is no cache coordination between middle tier nodes and static data (the curriculum framework) is cached in the middle tier while transactional data is not. We found that an Intel Xeon Processor 5600 Series core can support approximately 500 concurrent active users given the use cases for a system configured with approximately 500,000 students and 50,000 teachers.

The conclusions and recommendations reached from this testing are described in detail below, starting with an overview of the deployment architecture followed by a description of the test cases, and then laying out the JVM, caching and garbage collection parameters for the middle tier. We conclude with some recommendations for the database.

## OSL Deployment Architecture

The following diagram depicts the components involved in the OSL deployment as it was tested and represents a typical enterprise deployment architecture. The testing focused on the OSL WebLogic Cluster. See the **Appendix: Components of an OSL deployment environment** for an explanation of the terms in the diagram.

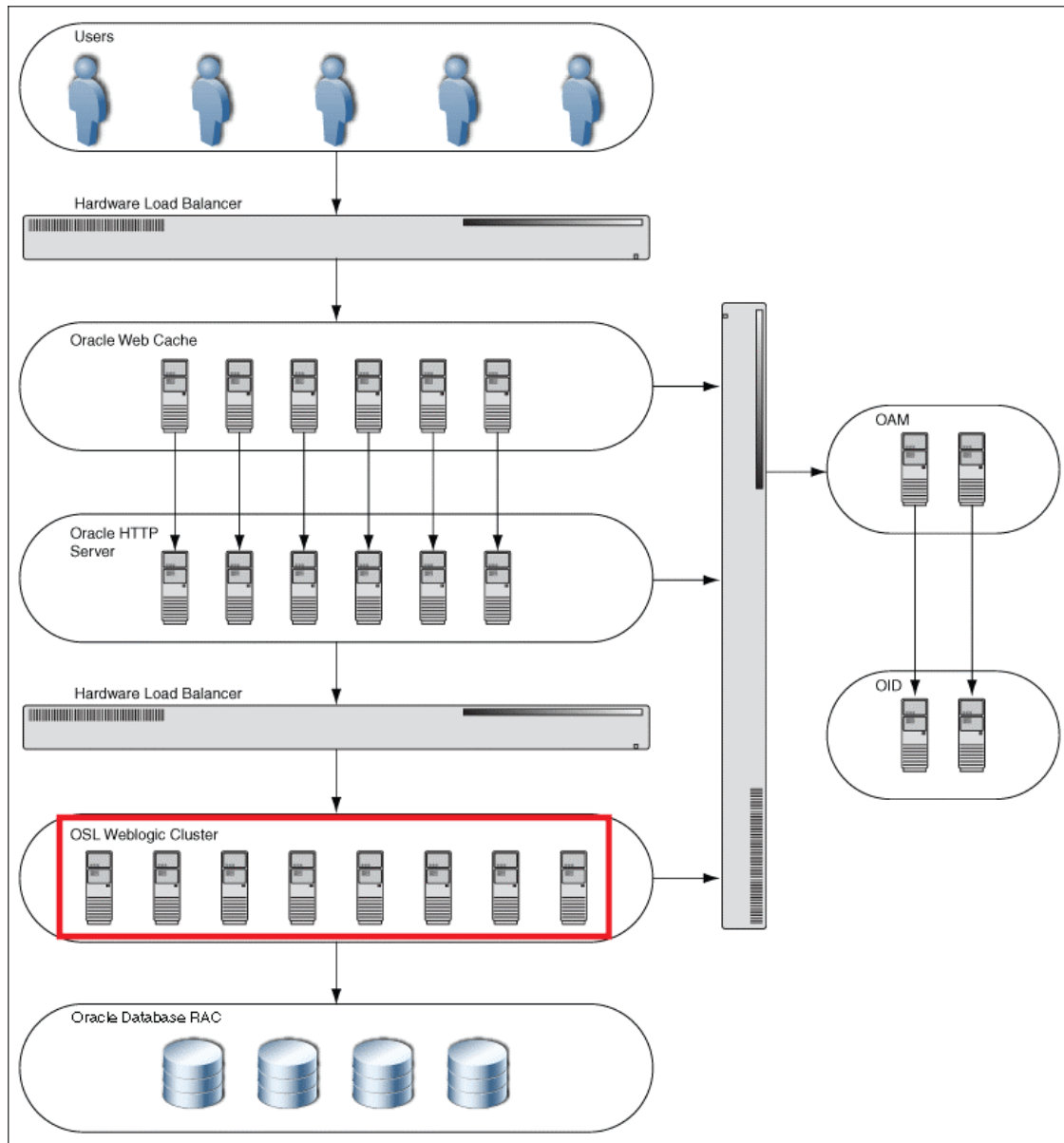


Figure 1: OSL Deployment Architecture

The OSL WebLogic Cluster was deployed on servers with 4-core Intel Xeon Processor 5600 Series using ESX Hypervisor virtual machines (VMs) and Sun JVM. Each WebLogic node (VM) was allocated 4 cores and configured with 24GB of memory. The JVMs were deployed using Sun JVM JDK 1.60\_21 and configured with 20GB heap space.

## Test Overview

**Table 1: Performance and scalability test operations** outlines the basic test cases and relative loads used during the performance testing.

**Table 1: Performance and scalability test operations**

User	Operation	Description	Percentage (%) of test volume
Teacher	Read	A teacher logs in to OSL, and views the <b>Observation</b> and <b>Curriculum</b> tabs.	35%
Teacher	Write	A teacher logs in to OSL, selects a class, and creates a learning item.	5%
Student	Read	A student logs in to OSL, and views the <b>My Home, My Progress,</b> and <b>My Observations</b> tabs.	60%

Testing started with 2,000 virtual users, scaled up to 16,000 virtual users, and finished with more than 25,000 concurrent virtual users. All tests ran for multiple hours to ensure that the tests represented ongoing load and not just initial load. The system was configured with 500,000 students and corresponding numbers of parents, teachers, and administrators.

### Result: Linear Scalability

Oracle Student Learning was found to scale linearly. The middle tier nodes were found to comfortably support 2,000 concurrent users each with no noticeable degradation as nodes were added.

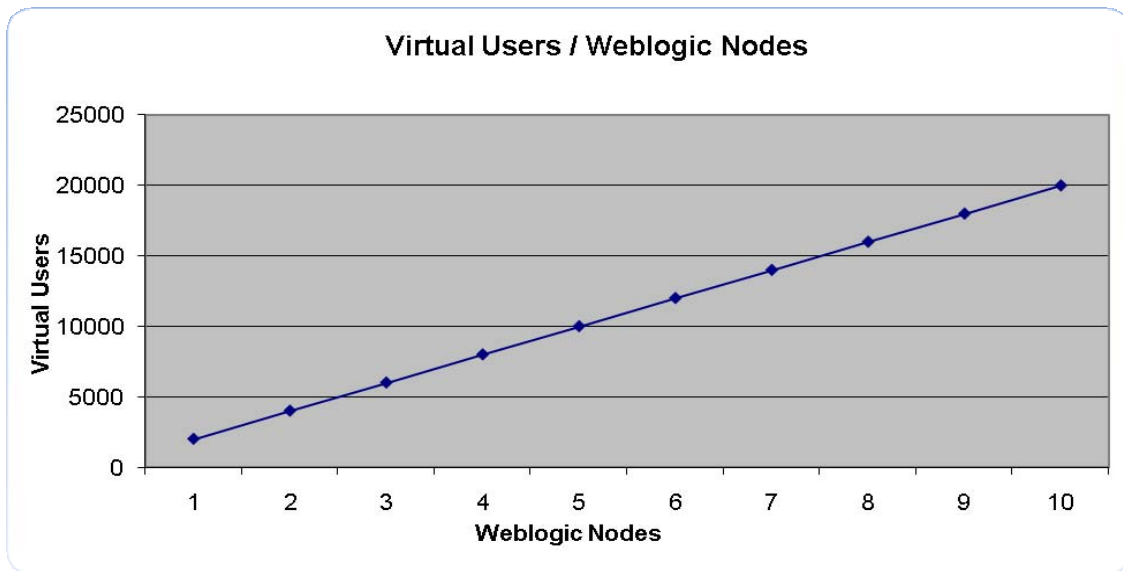


Figure 2: WebLogic nodes

Key to this deployment was configuration of the WebLogic nodes or VMs, including garbage collection within the JVMs and caching in the middle tier.

## WebLogic Node Allocation

This section provides recommendations and observations for configuring middle tier nodes and JVMs for a large-scale OSL deployment.

### Recommendations

To deploy Oracle Student Learning WebLogic nodes in a large-scale environment, we recommend the following:

- Allocate at least 4 cores to each OSL WebLogic cluster node.
- Allocate at least 1 node (of 4-cores) per 2,000 users.
- If you are using VMs:
  - Allocate sufficient resources to the VMs running each OSL WebLogic cluster node so that there is no resource contention between the VMs.
  - Quarantine the OSL VMs and do not over-allocate the hardware such that the VMs comprise more resources than the actual hardware available.

## JVM Configuration

This section lists the recommendations and observations on JVM configuration.

### Recommendations

This section discusses the recommendations for JVM configuration.

- Use Sun JVM rather than Oracle JRockit.  
OSL response times when using Sun JVM 1.6.0\_22 were observed to be 15-20% faster than when using JRockit 4.0.0-1.6.0.
- Allocate 20GB of JVM heap size on each OSL WebLogic cluster node.
- Garbage collection:
  - Use the concurrent mark sweep garbage collector.
  - Set the new generation and old generation heap space ratio to approximately 1:4.
  - Aggressively clean the soft references in the JVM.

For reference, *Table 2: JVM tuning parameters for garbage collection* contains garbage collection tuning parameters for a JVM heap of 20GB, with 4GB being used in the new generation.

**Table 2: JVM tuning parameters for garbage collection**

Parameter	Parameter Value
-XX:MaxNewSize	4000m
-XX:NewSize	4000m
-XX:+UseParNewGC	-
-XX:+UseConcMarkSweepGC	-
-XX:SurvivorRatio	20
-XX:+CMSParallelRemarkEnabled	-
-XX:-BindGCTaskThreadsToCPUs	-
-XX:+DisableExplicitGC	-
-XX:CMSInitiatingOccupancyFraction	70
-XX:+UseCMSInitiatingOccupancyOnly	-
-XX:PermSize	512m
-XX:SoftRefLRUPolicyMSPerMB	10

## Observations on JVM

Using the concurrent-mark-sweep garbage collector provided an increase in the number of concurrent users and more stability to OSL.

The OSL application is built on Oracle's Application Development Framework (ADF) and relies on ADF security for maintaining authorization policies. ADF security uses Oracle Platform Security Services (OPSS), which uses soft caches. The cache size grew significantly with increasing numbers of users in the system, resulting in frequent, full garbage collection cycles. Because the environment can become inaccessible to users during full garbage collection cycles, it is important to aggressively clean the JVM soft references.

## Caching in the Middle Tier

Applications often cache data in the middle tier to speed response time in providing information to the user (by saving the roundtrip time of accessing the database). Various caching scenarios were tested. This section discusses recommendations for, and observations about, caching in the OSL WebLogic cluster.

### Recommendations

Oracle Student Learning was found to perform best and scale best when only the relatively static information (Curriculum Framework) is cached in the middle tier and there is no attempt to synchronize or coordinate that cached data between the multiple middle tier nodes. Transactional data should not be cached in the middle tier.

The following are the recommended caching strategies for a multiple-node middle tier OSL deployment.

- Do not use middle tier cache synchronization or coordination for OSL.
- Disable Java Persistence API Level 2 (JPA L2) cache when OSL is deployed in a cluster.
- Cache the Curriculum Framework data in the application tier. There are certain entities in OSL (Curriculum Framework tree) that seldom change once they are correctly configured. Moreover the volume of Curriculum Framework data is typically very large. We recommend caching this data in the application tier, so that repeated database queries for these entities are avoided, and we have created a configuration property to enable this.

**Table 3: Java Persistence API L2 cache configuration** indicates the recommended configuration parameters for deploying the OSL application in WebLogic cluster.

**Table 3: Java Persistence API L2 cache configuration**

File name	Property	Value	Use
osl_configuration.properties	osl.lt.model.cache.isolation.enable	true	Disables L2 cache
	osl.lt.web.enableFrameworkDataCaching	true	Application caching of Framework data
persistence.xml	eclipselink.cache.shared.default	false	Use client-isolated cache for cached instances.

**Required: Restart middle tier when Curriculum Framework is updated**

With this configuration, the Curriculum Framework data is cached at the application level in the LT Web and middle tier nodes do not synchronize. Therefore, any update to the Curriculum Framework data, whether through the Administration screens or Data Loading Services, will not automatically be propagated to the other middle tier nodes. The result will be that users accessing OSL from other middle tier nodes will not see the updated information. Hence, with this configuration, the OSL application (middle tier) must be restarted whenever any change to the Curriculum Framework occurs so that all users will see the changes.

Restarting the middle tier will log out any users actively working in the system. In a production environment, changes to the Curriculum Framework are expected to be extremely infrequent (less than once a year). In addition, as school systems are normally geographically collocated, it is expected that system maintenance time would be available in the “middle of the night” for the few minutes required to restart the middle tier nodes.

**Observations on Caching in the Middle Tier**

Caching is often used to improve performance and throughput of applications by keeping information that users need in the middle tier so that the database need not be queried frequently. Low response time can be achieved if the information being requested is found in the middle tier cache rather requesting it from the database. The effectiveness of such caching depends on the application.

Given the high number of concurrent users, it is clear that the OSL application requires multiple middle tier WebLogic nodes.

The recommendation to cache static data and not to cache transactional data removes the need to coordinate between nodes and allows OSL to scale linearly. Testing showed no significant difference in the ability for a single node to scale with full caching. Testing also showed that the addition of cache coordination, when adding fully cached mid-tier nodes, had a significant negative effect on performance.

## Scenarios Considered

Based on the tests performed, we considered the following alternatives before presenting our recommendations:

### Full caching

OSL was originally configured to cache all data, both static and transactional, in the WebLogic middle tier. However, one of the first tests conducted was to evaluate performance of the application when only static data is cached. In this case, the curriculum framework data, which is changed infrequently, is cached in the WebLogic server, but transactional data is not. A user request or transaction that requires such transactional data necessarily requires a roundtrip communication to the database.

### Cache coordination between middle tier nodes

If the middle tier WebLogic server were configured to cache all OSL application data, then some form of cache coordination for transactional data would be required to enable a user that was connected to one server to see changes made by a user connected to another server. The typical example might be that a teacher assigns a learning item to a student. If the student is connected to a different middle tier node, that student might not see the assignment if there were no cache coordination. Cache coordination of transactional data would ensure that changes made on one WebLogic node would be propagated to all of the other nodes.

It was found that enabling cache coordination of clustered WebLogic nodes for OSL added overhead on the individual nodes and began to exhibit significant degradation in OSL performance when several nodes were added. Given the expectation that there could be at least five middle tier nodes, cache coordination would have a negative impact for scaling the Oracle Student Learning application.

### Cache Test Summary

In summary, testing showed the following:

- Performance degraded significantly (longer response times) when no data was cached as compared to caching all application data (static and transactional).
- Testing on a single node showed no significant increase in response time (an increase would indicate a degradation of performance) when the application is configured to cache only static data as compared with fully caching all application data.
- Testing on multiple nodes showed that cache coordination has a significant negative impact on scaling the OSL application when all application data is cached. Even if only static data is cached, cache coordination has a negative impact due to overhead.

## Additional Recommendations: Database Configuration

This section discusses additional observations and recommendations regarding the dataset base configuration. This configuration was specifically targeted during the testing described in this paper. However, there are some pertinent points of interest.

Running OSL against a single 8-core database node (Intel Xeon Processor 5600 Series) should be able to support 10,000 to 11,000 concurrent users and around 5,000 concurrent users in a database node running on a 4-core box. The following are our recommendations:

- Allocate sufficient database nodes to support the queries and transactions from the OSL nodes.
- Allocate approximately one database core for every two WebLogic server cores running OSL. The ratio may be altered based on database load.
- Database nodes should be deployed on the bare machine (not using VMs) and tuned to support a maximum number of queries and transactions.
- Use Oracle Real Application Clusters (RAC) when deploying multiple database nodes.

## Conclusion

Oracle Student Learning is successfully running in environments serving tens of thousands of concurrent users. Configuration of the middle tier is critical to scaling the application in a stable environment. Testing showed that OSL scales linearly when there is no cache coordination between middle tier nodes and when static data (the curriculum framework) is cached in the middle tier while transactional data is not.

## Appendix: Components of an OSL deployment environment

In the OSL deployment environment pictured in *Figure 1: OSL Deployment Architecture*:

- The **Oracle HTTP Server** (OHS) is the Web server component of Oracle Fusion Middleware. It provides an HTTP listener for Oracle WebLogic Server and the framework for hosting static pages, dynamic pages, and applications over the Web.
- The **WebLogic cluster** is the middle tier, where the OSL application is deployed. The WebLogic cluster is a collection of WebLogic application server nodes. An application server is a software framework that provides an environment where applications can run. An application server acts as a set of components accessible to the software developer through an application programming interface (API) defined by the platform itself. Enterprise web applications such as OSL are deployed on application servers such as WebLogic.
- The **Oracle Database Real Application Cluster** is an Oracle Relational Database hosted on one or more nodes, depending on the scalability needs, in which all nodes access the same database.
- The **Oracle Web Cache** is a secure, reverse proxy cache and compression engine that can be deployed between browser and HTTP server to improve the performance of web sites by caching frequently accessed content. Applications like OSL that are based on rich user interfaces may have a lot of static content (such as images) that might benefit from web cache deployment.
- The **Oracle Access Manager** (OAM) is the Oracle Fusion Middleware 11g single sign-on solution. Single sign-on (SSO) enables users, and groups of users, to access multiple applications after authentication.
- The **Oracle Internet Directory** (OID) is the default LDAP mechanism used by OSL Learning Tool components for authentication and authorization.
- A **hardware load balancer** is an appliance device that is used to split network load across multiple servers. The chosen load balancer should support persistence. Persistence is a feature that is required by many web applications including OSL. Once a user has interacted with a particular server, all subsequent requests are sent to the same server, thus persisting to that particular server. It is normally required when the session state is stored locally to the web server as opposed to a database.



Configuring Fusion Middleware to Scale for  
Oracle Student Learning  
August 2011

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

Copyright © 2011, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0611

**Hardware and Software, Engineered to Work Together**