

# Oracle Weblogic

Setup, Configuration, Tuning, and  
Considerations

Presented by: Michael Hogan  
Sr. Technical Consultant at Enkitech



# Overview

- ✦ Weblogic Installation and Cluster Setup
- ✦ Weblogic Tuning
- ✦ Considerations on Transitioning to Weblogic

# Weblogic Installation and Cluster Setup

# Weblogic Installation and Cluster Setup Steps

1. Install Weblogic on machines to be clustered
2. Create the Weblogic Domain on one machine
3. Install Weblogic onto other machines to be clustered
4. Configure the domain by adding Managed Servers and Machines on one machine
5. Copy the domain file structure to the other machines and enroll it to the Weblogic Domain
6. Ensure that all Weblogic Machines are operational
7. Create and configure the Cluster
8. Ensure that the Cluster is operational



# Install Weblogic on machines to be clustered

- ✦ Download and Run the Oracle installer executable (Windows or Linux)
- ✦ Create a new middleware home: [\$ORACLE\_HOME/Middleware]
- ✦ Uncheck to register for security updates (optional)
- ✦ Choose installation type: typical = full install excluding server examples, custom allows de-selecting components [TYPICAL]
- ✦ Choose Product Installation Directories [Choose defaults; all under new middleware home]
- ✦ Installation Summary window lists all items to be installed/performed
- ✦ Click “Next” to install

# Installation Video



# Install Webllogic on machines to be clustered

- ✦ Repeat installation process on each machine that is to be grouped into one or more clusters
- ✦ Installation **must be** at the same location on each machine
- ✦ Installation Note: Examples/samples DO NOT have to be installed

# Weblogic Machine Clustering - Steps

- Pack domain from primary machine
- Unpack domain onto subsidiary machine(s)
- Start Nodemanager on subsidiary machine(s)
- Use WLST to connect and enroll subsidiary machine(s) to primary
  - `connect('login', 'password', 't3://ipordomainofadminserver:7001')`
  - `nmEnroll('full path to domain directory')`
- Define [Managed] Server(s) to point to subsidiary Nodemanager(s)
- Define Machine(s) to point at subsidiary Nodemanager(s)
- Associate Machine(s) to Server(s)
- Create a Cluster and associate with Server(s)

# Clustering Video

# Installation and Clustering Summary

- Installation is more involved than that of other Java Application Servers, but not to a great extent
- Once you have a cluster of machines created, you get:
  - Load balancing
  - Failover
  - Scalability
  - High Availability
  - Centralized Database Access
  - Centralized Application Deployment
  - Centralized Server Management
- for free!



# Weblogic Tuning

# Weblogic Tuning

- Tune Pool Sizes
  - Provide pool sizes (such as pools for JDBC connections, Stateless Session EJBs, and MDBs) that maximize concurrency for the expected thread utilization.
  - A server instance uses a self-tuned thread-pool. The best way to determine the appropriate pool size is to monitor the pool's current size, shrink counts, grow counts, and wait counts.
  - In the case of JDBC connection pools, setting the minimum connection value equal to the maximum connection value (after testing DB load to determine this value) will provide a performance enhancement by helping reduce memory utilization and spiking, and sudden database loads.
- Use the Prepared Statement Cache - The prepared statement cache keeps compiled SQL statements in memory, thus avoiding a round-trip to the database when the same statement is used later.

# Weblogic Tuning

- Optimistic-concurrency with cache-between-transactions work best with read-mostly beans. Using verify-reads in combination with these provides high data consistency guarantees with the performance gain of caching
- Query-caching is a feature that allows the EJB container to cache results for arbitrary non-primary-key finders defined on read-only EJBs. All of these parameters can be set in the application/module deployment descriptors
- Use Local Interfaces or use call-by-reference semantics to avoid the overhead of serialization when one EJB calls another or an EJB is called by a servlet/JSP in the same application
- Tune Messaging Applications - always configure quotas and paging

# Weblogic Tuning

- Adjustments to the min and max Java heap memory values via the Weblogic startup parameters -Xms and -Xmx. For example, using -Xms256m and -Xmx512m would set the min to 256Mb and the max to 512Mb
  - It is recommended that the min and max be set to the same value.
- Choice of JVM (JRokit vs. Sun) will impact performance to a degree, mainly due to the inherent optimizations of the JVM
- Clustering will provide performance benefits by spreading load across multiple servers.
- Use Logging Last Resource Optimization
  - When using transactional database applications, consider using the JDBC data source Logging Last Resource (LLR) transaction policy instead of XA

# Weblogic Tuning

- Tune the Chunk Size - A chunk is a unit of memory that the WebLogic Server network layer, both on the client and server side, uses to read data from and write data to sockets. A server instance maintains a pool of these chunks. For applications that handle large amounts of data per request, increasing the value on both the client and server sides can boost performance
- Tune Connection Backlog Buffering - You can tune the number of connection requests that a WebLogic Server instance accepts before refusing additional requests. This applies primarily to web applications
- Tune HTTP Sessions - Optimize your application so that it does as little work as possible when handling session persistence and sessions. You should also design a session management strategy that suits your environment and application
- Simple is best - Don't setup Weblogic for functionality you don't need!

# Considerations for Transitioning to Weblogic

# Considerations for Transitioning to Weblogic

- ✦ External and Environmental Considerations
- ✦ Application, Development, and Configuration Considerations

# External and Environmental Considerations

- ✦ Server configuration
  - ✦ Weblogic has possibly different demands on a server or network environment than other Application Servers
  - ✦ “Too Many Files Open” - Weblogic handles archives and file resources differently than JBoss and Glassfish. It expands archives on the FS rather than selectively in memory and requires more file handles to be allotted than other AS’ typically do
- ✦ Replacement or inclusion of “external” Web or LB server

# Application, Development, and Configuration Considerations

- ✦ JVM choice/use
  - ✦ Sun JVM is bigger and less optimized for performance over development practicalities
  - ✦ JRokit is smaller and optimized for production environment use
- ✦ Use proper JDBC drivers for datasources and applications
  - ✦ Accepting default XA JDBC drivers can cause issues for applications that don't use XA

# Application, Development, and Configuration

- Session replication - applications must be built, designed and configured to make use of session replication, EJB replication, etc
  - This is usually done by definitions placed into web.xml and weblogic.xml files, along with notations in the source code
  - EJBs must be developed with specific features to be replicated, most notably in being serializable
- Applications must be tailored to make use of Weblogic specific functionality, especially to make use of persistence and failover
  - Failover and load balancing are provided by default, but only in the sense that your application will remain up and available for failures of n-1 nodes

# Considerations on Transitioning to Weblogic

- ✦ Overall:
  - ✦ It is better to develop to take advantage of Weblogic's features, rather than modify Weblogic to handle an unmodified application
  - ✦ Weblogic will most likely operate in a different fashion to other application servers you are currently (or previously) using, especially in regard to resource usage, allocation, and developmental due diligence



Thanks for your time!



Fastest Growing Companies  
in Dallas