Clustering Information

Contents

Clustering Information

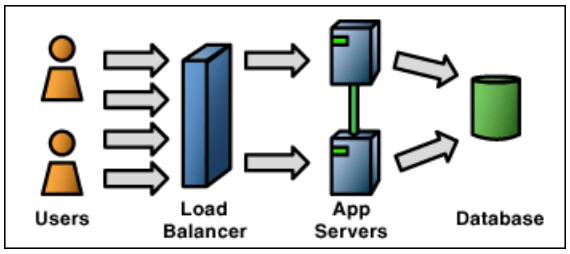
This guide describes the clustering model, which allows the application to run on a group of servers.

The clustering module provides increased capacity and fault-tolerance. Clusters are auto-configuring, have no single point of failure, and provide nearly linear scalability as additional servers are added.

At the heart of the clustering module is a distributed cache powered by *Oracle Coherence*. This allows application state to be shared among cluster members. Servers can join and leave the cluster at will, and cluster node failures are transparently handled.

Cluster Architecture

The best clustering configuration will depend on your traffic needs, existing infrastructure, and budget. One possible clustering configuration is pictured below:



There are three layers to this setup:

- **Load-balancer** Between users and the application servers is a load-balancing device. The device may be hardware-based or software-based. Every user has a session (represented by a unique cookie value) that allows stateful data to be maintained while they are using the application. Each session is created on a particular application server. The load-balancer must be "session-aware," meaning that it inspects the cookie value and always sends a given user's requests to the same application server during a given session. Without session-aware load balancing, the load-balancer could send requests to any application server in the cluster, scrambling results for a given user.
- **Application Servers** In the middle-tier, multiple application servers are set up and the clustering feature is enabled. Caches between the application instances are automatically synchronized. If a particular application server fails, the load-balancer detects this and removes the server from the cluster.
- **Database** All instances in a cluster must share the same database.

Note: You're unlikely to see any performance improvement at all with a cluster of only two nodes.

Clustering Best Practices

Following a few best practices will improve the performance and stability of your clustered installation. Here are a few suggestions based general guidelines for the clustering technology, as well as experience with Jive SBS in a clustered environment.

- Use three nodes or more in your cluster. Two nodes will give you a redundancy benefit, but is unlikely to help at all with performance (in fact, you might see a performance decrease). If you're scaling for capacity, go from a single node to three nodes.
- As a rule of thumb, consider going to a three-node cluster at approximately 12 million page views or 40 thousand users.
- Ensure that the number of nodes in your cluster is greater than what you'll need to handle the load you're getting. For example, if you're at capacity with three nodes, then the cluster will fail when one of those nodes goes down. Provision excess capacity so that your deployment can tolerate a node's failure.

- When the application is deployed on a cluster, its heap size requirements will decrease by as much as 50 percent or more. This is because the cluster management software stores everything in serialized, rather than unserialized, lists. With smaller heap size requirements, you can increase your cache sizes (if you have a large database).
- Clustering tends to be very latency intolerant. To offset this, ensure that everything is connected to a single gigabit switch and that all servers use gigabit network cards.

Setting Up a Cluster

Your license determines whether or not clustering is enabled and how many nodes are supported. To check on the number of clustered servers your license allows, see the license information after logging into the application administration tool.

Note: Before enabling clustering, you should set the system time on each cluster member to match other cluster member's settings as closely as possible, or (preferred) set the cluster members to sync to a centralized time server.

Next, navigate to System > Settings > Caches. Toggle Clustering to Enabled and then Save settings (note that the Clustering option will only be displayed if your license allows it). The server will automatically discover and join the cluster. You'll then see additional options in the cache panel to monitor cluster activity.

Note: When enabling clustering on the first clustered node, there will be a 30-second delay when it starts. This is a necessary component of the clustering protocol.

Clustering FAQ

Do all cluster members need to be on the same local network? Yes. By default, the clustering module uses IP multicast to discover the cluster. This requires all cluster members to be on the same local network or for multicast traffic to be tunneled across multiple networks. See the *Coherence documentation* for more information.

Is it possible to have more than one cluster per physical network? Yes, this is possible. Coherence uses IP multicast to discover other cluster members. So, to isolate different clusters on the same physical network, you simply need to tell each cluster to use different multicast IPs or ports. The address and port can be specified with the *appadd* command (see *Application Management Commands* for more information).

Does searching work when clustering is enabled? Yes, searching works without needing to make any special configuration changes. Each cluster member will maintain its own search index and update it with the latest data (when auto-indexing is enabled).

Do gateways work with clustering? Yes, gateways will work transparently in a cluster.

How do config files work in a cluster? All configuration data (except bootstrap information such as database connection information) is stored in the database. Changing configuration settings on one cluster member will automatically update them on all other cluster members.

How do I set Coherence operational configuration parameters such as multicast port or address? The "instance" file contains the specific settings for each application. See the "appadd" documents for more information.

As a developer, how do I add a cache of my own? All caches are defined in the file WEB-INF/classes/coherence-cache-config.xml. The format of this file is defined and *extensively documented* by Oracle.