Clustering

Clustering LAMS 2

This page is a collection of notes on load balancing and clustering LAMS on JBoss 4.0.2. There are 2 levels of clustering that we touch on, load balancing and session replication. There are a few things to overcome to get full session replication working (which we'll mention later), so for the most part we concentrate on load balancing. That is, several JBoss nodes sitting behind a load balancer, and writing to the same database.

Also note that this is more of a functional setup in order to run tests rather than the last word.

Overview

Roughly speaking, the steps below will set up a load balancer using Apache and mod_jk, providing sticky sessions to a group of JBoss nodes.

Setup the first node

1. Make a copy of the 'all' server configuration and name it 'node1'. Deploy LAMS here.

   If using Linux...

   In ${JBOSS_HOME}/bin/run.conf, set -Djava.net.preferIPv4Stack=true in JAVA_OPTS
   See http://wiki.jboss.org/wiki/IPv6 and http://www.experts-exchange.com/Software/Server_Software/Application_Servers/Java/Jboss/Q_22677495.html for reference.

2. Move lams.ear to /var/lib/lams/lams.ear
3. Link ${JBOSS_HOME}/server/node1/deploy/lams.ear to /var/lib/lams/lams.ear
4. In ${JBOSS_HOME}/server/node1/deploy/jbossweb-tomcat55.sar/META-INF/jboss-service.xml, set UseJK to true in order to enable sticky sessions from the Tomcat side.
5. In ${JBOSS_HOME}/server/node1/deploy/jbossweb-tomcat55.sar/server.xml, give your node a jvmRoute name:

   ```
   <Engine name="jboss.web" defaultHost="localhost" jvmRoute="node1">
   ```

6. Delete ${JBOSS_HOME}/server/node1/deploy/deploy.last/farm-service.xml to disable farming
7. In ${JBOSS_HOME}/server/node1/deploy/tc5-cluster-service.xml and local-service.xml, set CacheMode to REPL_SYNC. We want entity cache replication to be synchronous so our hibernate objects are in a consistent state.
8. (Optional - is this required?) In ${JBOSS_HOME}/server/node1/deploy/jbossweb-tomcat55.sar/context.xml, set distributable flag to true:

   ```
   <Context ... distributable="true">
   ```

9. Start up LAMS by doing cd ${JBOSS_HOME}/bin; ./run.sh -c node1 -b ${IP_ADDRESS}

   Make sure you specify the IP address in order for JBoss' auto discovery to work.

Share common files

Setup a network share for /var/lib/lams - this directory will contain lams.ear (application files + lams-www uploaded files). In terms of the configuration keys, this share will contain the EARDir, ContentRepositoryDir, TempDir, and DumpDir. The network user will need write access to these folders.

This path will need to be the same on all node servers, since all nodes use the same value from the database.

I've found that I needed to mount the lams.ear directly in the deploy folder, then link /var/lib/lams to the mount. The other way around jboss would continually undeploy and deploy lams.ear for some reason.

Setup the second node

1. Unzip a new JBoss instance on a new machine
2. Copy the 'node1' directory from the first node into ${JBOSS_HOME}/server/ directory, rename 'node2'
3. Set preferIPv4Stack=true if Linux
4. Mount the network share and link location of EARDir to ${JBOSS_HOME}/server/node2/deploy/lams.ear
5. Modify ${JBOSS_HOME}/server/node1/deploy/jbossweb-tomcat55.sar/server.xml and change jvmRoute="node2"
6. Start LAMS as above.

You should see something like the following on startup:

```
2008-10-08 14:38:46,184 INFO  [org.jboss.ha.framework.interfaces.HAPartition.DefaultPartition] Number of
cluster members: 2
2008-10-08 14:38:46,185 INFO  [org.jboss.ha.framework.interfaces.HAPartition.DefaultPartition] Other members: 1
2008-10-08 14:38:46,185 INFO  [org.jboss.ha.framework.interfaces.HAPartition.DefaultPartition] Fetching state
(will wait for 30000 milliseconds):
2008-10-08 14:38:46,186 INFO  [org.jboss.ha.framework.interfaces.HAPartition.DefaultPartition] New cluster view
for partition DefaultPartition: 1 ([172.20.100.22:1099, 172.20.100.28:1099] delta: 0)
```

This indicates that it has found node1. Similarly, node1 should show:

```
2008-10-08 14:42:18,442 INFO  [DefaultPartition] New cluster view for partition DefaultPartition (id: 1, delta:
1) : [172.20.100.22:1099, 172.20.100.28:1099]
2008-10-08 14:42:24,749 INFO  [TreeCache] viewAccepted(): new members: [172.20.100.22:33451, 172.20.100.28:
32821]
2008-10-08 14:42:24,752 INFO  [TreeCache] locking the tree to obtain transient state
2008-10-08 14:42:24,754 INFO  [TreeCache] returning the transient state (140 bytes)
```

Setup the load balancer

Note that the jvmRoute define in server.xml **must** be the same as the worker.**nodeX**.*

1. Setup Apache with mod_jk
2. Edit httpd.conf, add the following with paths modified as required

```
LoadModule jk_module /usr/lib/apache2/modules/mod_jk.so
JkWorkersFile /etc/libapache2-mod-jk/workers.properties
```

3. Set JkMount in your virtual server:

```
JkMount /lams* loadbalancer
```

4. Configure the workers.properties:

```
worker.list=loadbalancer

worker.node1.port=8009
worker.node1.host=172.20.100.22
worker.node1.type=ajp13
worker.node1.lbfactor=1

worker.node2.port=8009
worker.node2.host=172.20.100.28
worker.node2.type=ajp13
worker.node2.lbfactor=1

worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1,node2
worker.loadbalancer.sticky_session=1
worker.loadbalancer.method=S
```

Resources

JBoss manual chapter on clustering, http://docs.jboss.org/jbossas/jboss4guide/r4/html/cluster.chapt.html
Tomcat documentation on clustering, http://tomcat.apache.org/tomcat-5.5-doc/cluster-howto.html
Mod jk worker documentation, http://tomcat.apache.org/connectors-doc/reference/workers.html

Problems for LAMS 2

Configuration - config values are currently read from a Map static to each node. This needs to be cached across the nodes or read directly from the database in order for changes to config values to work. As a workaround, accessing the 'edit configuration details' updates the Map.

SessionMap - uses a unique counter static to each node, but since we have sticky sessions this isn't a problem.

SessionManager - same as for SessionMap.

The above classes will probably need to be changed to use jboss cache if we want session replication/failover/fault tolerance.

Cluster start up

To ensure that the cluster starts correctly, it must be started in this sequential order:

1. Start MySQL
2. Start JBoss nodes
3. Start Apache

Make sure that each of these services is **fully** started before starting the next one.