mod_cluster

A new httpd-based load balancer

Brian Stansberry

JBoss, a division of Red Hat

Agenda

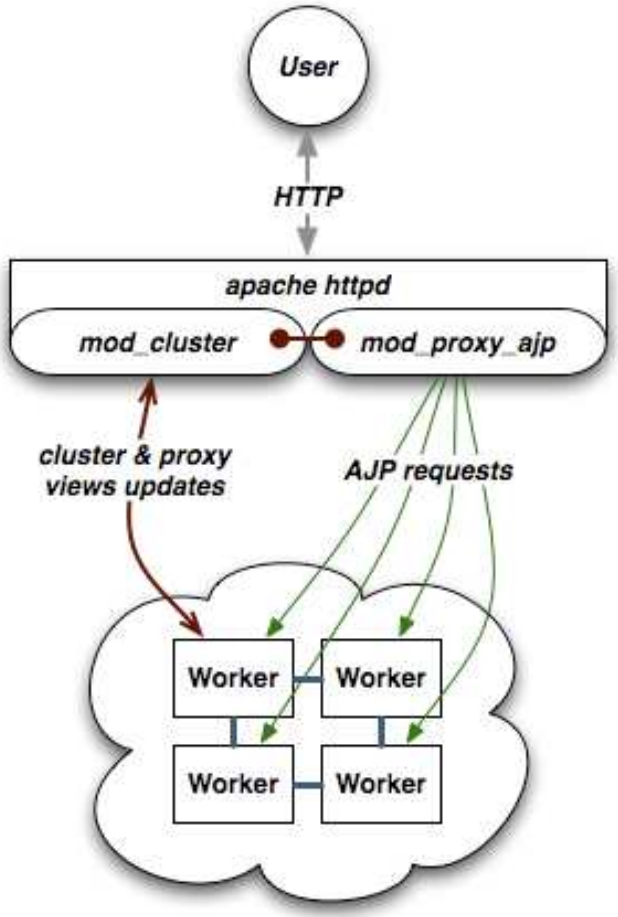
- **Who is Brian Stansberry?**
  - **Principal Software Engineer at Red Hat**
  - **Technical Lead for JBoss Application Server Clustering**
    - Part of JBoss' overall clustering team
  - **Contributor to mod_cluster**
- **What is he going to do today?**
  - **Provide overview of mod_cluster**
  - **Describe key benefits**
  - **Give some brief info on how to obtain it and set it up**
  - **Demo**
  - **Q&A**

What is mod_cluster?

- Set of modules for Apache httpd and a Tomcat-based webserver
  - Apache httpd-2.2.8+
  - JBoss AS 5.0.0.GA+, JBoss Web 2.1.1+, Tomcat 6
- Allows httpd to act as a load balancer in front of Tomcat-based web servers
  - similar to mod_jk and mod_proxy_balancer
- JBoss.org project
  - http://www.jboss.org/mod_cluster
  - LGPL
- Current release is 1.0.0.Beta4
  - First Release Candidate expected this month

Architecture

- **User requests proxied to backend server using AJP**
  - **HTTP/HTTPS also supported**
  - **Request handling on Java side not affected by mod_cluster**
- **Key difference – back channel *from* backend server *to* httpd**
  - **Lifecycle information**
  - **Load balancing information**
  - **Uses HTTP or HTTPs**



Key Advantages

- **Configuration**
  - **httpd side does not need to know cluster topology in advance**
  - **Very little configuration on the httpd side**
  - **Dynamic, not static**
- **Improved Load Balancing**
  - **Main calculations done on the backend servers, where more information is available**
- **Fine grained webapp lifecycle control**
  - **Undeploy an app from a running node without 404s**

Dynamic Configuration

- Backend servers register themselves with httpd during startup
- Backend servers register applications as they are deployed
- No static topology configuration on httpd side
  - No more workers.properties
  - No more uriworkermap.properties
- Optional: httpd servers advertise themselves to backend servers via multicast
  - No topology configuration at all

No more workers.properties

Eliminate redundant boilerplate config

workers.properties

```
worker.list=lb

worker.lb.type=lb
worker.lb.balance_workers=node1,node2

worker.node1.type=ajp13
worker.node1.host=192.168.2.1
worker.node1.port=8009
worker.node1.lbfactor=1

worker.node2.type=ajp13
worker.node2.host=192.168.2.2
worker.node2.port=8009
worker.node2.lbfactor=1
```

Better Load Balancing

- **Problem: Load Balancer lacks info needed to make optimal balancing decisions**
  - **Aware of: number of requests, number of sessions, bytes sent/received, response times**
  - **Ignorant of: critical backend server metrics, e.g. CPU utilization, available memory, DB connection pool usage**
  - **Ignorant of: activity of other load balancers**
- **Solution: Backend servers periodically tell httpd how much load each can handle**
  - **“Load Balance Factor”: number between 1 and 100**
  - **Load balancer uses relative factors to make decisions**
  - **Backend server uses configurable set of metrics to derive the factor**

Load Metrics

- A particular metric that a backend server tracks to help decide how much “load” it is under
  - e.g. Heap utilization, CPU utilization
- Give weights to multiple metrics to come up with an overall load factor
  - e.g. 25% CPU, 25% request count, 50% Session count
  - More than one metric supported in JBoss AS only
- Multiple readings of metrics go into load factor; older readings decline in importance
- Highly configurable
  - Pick the metrics and weights that are relevant to your application

Available Load Metrics

- **Web tier utilization:**
  - **Active sessions, busy connections, bytes sent, bytes received, request count**
- **System utilization:**
  - **CPU utilization, system memory usage, heap usage, number of threads**
- **JCA Connection Pool Utilization**
- **Generic version to listen to any metric exposed via JMX**
- **You can write your own**

Installation – httpd side

- Two downloads – httpd side and Java side
  - http://www.jboss.org/mod_cluster/downloads/
- httpd downloads available for many architectures
  - Linux x86, x86_64, IA64
  - Solaris 9 Sparc, 10 x86
  - Windows 32, 64
  - HP-UX i64, 9000/800
  - Can of course be built from source
- Download is a full httpd distribution
  - You could also copy the required .so files from the distribution to your existing httpd install

Configuration – httpd side

Minimal config – add following to httpd.conf:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule slotmem_module modules/mod_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule advertise_module modules/mod_advertise.so

Listen 192.168.2.3:6666
<VirtualHost 192.168.2.3:6666>
  <Directory />
     Order deny,allow
     Deny from all
     Allow from 192.168.2.
  </Directory>

KeepAliveTimeout 60
MaxKeepAliveRequests 0
AdvertiseGroup 224.0.1.105:23364
</VirtualHost>
```

Installation – Java side

- Single java binary download can be used in both JBoss Web/Tomcat and in JBoss AS 5
- JBoss Web/Tomcat
  - Copy contents of the distribution's JBossWeb-Tomcat folder into your install
  - Adds 3 jars to $CATALINA_HOME/lib
- JBoss AS 5
  - Copy the distribution's mod_cluster.sar folder to JBoss' deploy dir

Configuration – JBossWeb/Tomcat

- **Add a LifecycleListener so mod_cluster is aware of lifecycle events**
- **Add a jvmRoute to give this node a name**

$CATALINA_HOME/conf/server.xml

```
<Server>
  <!-- ... -->
  <Listener className="org.jboss.modcluster.ModClusterListener" advertise="true"/>
  <!-- ... -->
  <Service name="jboss.web">
    <Connector protocol="AJP/1.3" port="8009"
               address="192.168.2.1" redirectPort="8443"/>
    <Engine name="jboss.web" defaultHost="localhost" jvmRoute="node1">
      <!-- ... -->
    </Engine>
  </Service>
</Server>
```

Configuration – JBoss AS

- The server.xml config is similar to Tomcat/JBoss Web shown on last slide.
- Hooks into JBoss Microcontainer that's at the core of JBoss AS
  - Allows a much richer set of configuration options than can be specified via server.xml
  - E.g. multiple load metrics
- See docs on http://www.jboss.org/mod_cluster for details

“Clustered” Mode

- **Java side can operate in 2 modes, depending on whether the Java servers are able to exchange messages with each other**
- **Non-clustered mode – each backend server independently communicates with each httpd server**
  - **Lot of connections if there are a lot of servers**
- **Clustered mode – JBoss AS “all” config only**
  - **JGroups used to communicate between backend servers**
  - **One “master” server elected to communicate with httpd**
    - HA – if master fails another takes over

Demo

- **Topology**
  - **Single Apache httpd instance**
  - **Two JBoss AS instances – cluster01 and cluster02**
  - **WAR initially only deployed on cluster01**
  - **Zero static topology configuration:**
    - AS instances not configured to know about httpd
    - httpd not configured to know about AS instances
- **Demo Application**
  - **WAR**
  - **Client GUI that generates load, tracks load balancing**
  - **Available in the mod_cluster Java binary download**
    - demo/ dir

For more info:

- **Main mod_cluster site**
  - **http://www.jboss.org/mod_cluster**
  - **Links to docs, downloads, forums, dev lists, etc**
- **Tech lead for mod_cluster, Jean-Frederic Clere, will be speaking at ApacheCon next week**
- **Community participation very welcome**
- **Questions?**