Apache Tomcat

Tomcat Clustering: Part 3 – Clustering

Mark Thomas, 15 April 2015
Introduction

• Apache Tomcat committer since December 2003
  – markt@apache.org
• Tomcat 8 release manager
• Member of the Servlet, WebSocket and EL expert groups
• Consultant Software Engineer @ Pivotal
• Currently focused on Apache Tomcat 9
Terminology
Reverse Proxy

bz.apache.org
httpd instance

Bugzilla (Main)
(bz.apache.org/bugzilla)
httpd instance

Bugzilla (AOO)
(bz.apache.org/ooo)
httpd instance

Bugzilla (SpamAassassin)
(bz.apache.org/SpamAssassin)
httpd instance
Load-balancing

www.apache.org
geoip

Europe
www.eu.apache.org

US
www.us.apache.org
Clustering

Load-balancer

Instance 01
Replicated Session State

Instance 02
Clustering

• Load-balancing plus
• Node failure is transparent to users
• This transparency comes at a (usually significant) cost
Putting it all together
Clustering
Agenda

• When to cluster
• Clustering components
• Configuration choices
• Debugging
• Demonstration
  – Time permitting
• Questions
When To Cluster

• Ideally, never
  – Adds configuration complexity
  – Requires additional processing
  – Debugging is lot harder

• What do you really need?
  – Load-balancing plus sticky sessions
  – If a node fails, sessions will be lost

• Clustering should be the last resort
Components

Cluster

- Manager
- Membership
- Channel
- Receiver
- Sender
- Interceptors
- Listeners
- Valves
- Deployer
Components

• Cluster
  – Container for all cluster related configuration
  – May be placed within Engine or Host

• Manager
  – Controls how data is replicated between nodes

• Channel
  – Communication between cluster nodes
Components

- Membership
  - Tracks which nodes joining and leaving the cluster

- Sender
  - Sends cluster messages to other nodes

- Receiver
  - Receives messages from other nodes

- Interceptors
  - Valves for cluster messages
Components

• Listeners
  – Receive notifications of cluster messages
  – Managers support LifecycleListener
  – Standard session listeners remain available

• Valves
  – Inserted into the request processing pipeline

• Deployer
  – Cluster wide deployment of web applications
Configuration Choices

Manager

• Two options
  – Delta manager
  – Backup manager
Configuration Choices

Manager

• Delta manager
  – Default
  – Replicates every change to every node
    ▪ Maximum reliability
  – Network traffic proportional to the square of the number of nodes
    ▪ Doesn’t scale to large numbers of nodes
  – Fail-over can be to any node
Configuration Choices

Manager

• Backup manager
  – Sessions have a primary node and a backup node
    ▪ Need to use sticky sessions
  – Backup node selected on a round-robin basis from all other nodes
  – There is NOT a single backup node
  – Every node knows the primary node and backup node for every session
  – Network traffic proportional to the number of nodes
  – Failover is more complicated
Configuration Choices

Manager

Node A
Primary Sessions: 30*A
Backup sessions: 10*B’, 10*C’, 10*D’

Node B
Primary Sessions: 30*B
Backup sessions: 10*A’, 10*C’, 10*D’

Node C
Primary Sessions: 30*C
Backup sessions: 10*A’, 10*B’, 10*D’

Node D
Primary Sessions: 30*D
Backup sessions: 10*A’, 10*B’, 10*C’
Configuration Choices

Manager

Node A
Primary Sessions: 30*A
Backup sessions: 10*B’, 10*C’, 10*D’

Node B
Primary Sessions: 30*B
Backup sessions: 10*A’, 10*C’, 10*D’

Node C
Primary Sessions: 30*C
Backup sessions: 10*A’, 10*B’, 10*D’

Node D
Primary Sessions: 30*D
Backup sessions: 10*A’, 10*B’, 10*C’
Configuration Choices

Manager

• Node D fails

• Sessions will be distributed to other nodes
  – As soon as node failure is detected

• If new node was the backup
  – It becomes the primary
  – A new backup node is selected
  – Session is copied to new backup node
Configuration Choices
Manager

• If new node was not the backup
  – It becomes the primary
  – The backup node remains the same
  – Session is copied from the backup node

• Sessions are re-distributed amongst remaining nodes
Manager

Node A
Primary Sessions: 40*A
Backup sessions: 20*B', 20*C'

Node B
Primary Sessions: 40*B
Backup sessions: 20*A', 20*C'

Node C
Primary Sessions: 40*C
Backup sessions: 20*A', 20*B'

Node D
Primary Sessions: 30*D
Backup sessions: 10*A', 10*B', 10*C'
Configuration Choices

Membership

• Two options
  – Multicast
  – Static
Configuration Choices

Membership

• Multicast membership
  – Requires multicast to be enabled on the network
  – Can be difficult to debug problems
  – Scales more easily

• Static
  – Simple to debug
  – Adding nodes gets time consuming as cluster grows
Configuration Choices

sendOptions

• Delta manager
  – channelSendOptions on Cluster

• Backup manager
  – mapSendOptions on Manager

• Synchronous or asynchronous
sendOptions

• Synchronous
  – Request processing does not complete until session data has been sent
  – What is meant by sent?
    ▪ On the TCP stack
    ▪ Received by the other node
    ▪ Processed by the other node
  – Next request to a different node will see updated sessions
Configuration Choices

sendOptions

• Asynchronous
  – Request processing continues while session data is sent
  – Next request to a different node may or may not see updated sessions
Configuration Choices

Summary

- **Manager**
  - Delta or Backup
  - (Sticky sessions)

- **Membership**
  - Multicast or static

- **Send options**
  - Synchronous or asynchronous
Debugging
Cluster Configuration

• Need to know
  – Session ID
  – Current route
  – Which node handled the request

• I use a simple JSP page that shows all of the above
Debugging
Cluster Configuration

• Quickly test behaviour is as expected
  – Is the route correct for the current node
  – Is load-balancing happening as expected
  – Is fail-over happening as expected

• Keep in mind how reverse proxy handles failed nodes
Debugging
Application Problems

• Just like trying to debug any other application problem
  – But harder

• Can the issue be replicated in a non-clustered environment?

• Approach depends a lot on the application
Debugging

Application Problems

• Network / failover issues
  – Look at the access logs (need session IDs)
  – Look at error logs
  – May need to look at network traffic

• Application issues
  – Logging, logging and more logging
  – Need to be able to fine tune logging
Questions