Web Sessions, Ehcache, and Quartz Clustering

Ryan Vanderwerf

Chief Architect

ReachForce

www.reachforce.com

My Background

Currently building a Grails and Cloud based infrastructure for ReachForce

Architected a Grails solution for Developerprogram.com that allows rapid deployment of Developer Program portals for all kinds of companies, specializing in the mobile industry.

My Background

Built Java and Linux based webcasting for events such as SXSW,

built telecom software, and ASP's for the financial sector
Worked with Java since 1996, and built server-side applications ever since

Enticed into the Groovy and Grails space by speakers at the early NFJS conferences

Clustering With Terracotta

What Will We Cover?

- Clustering basics
- Reasons for clustering
- Types of clustering
- HTTP session clustering
- Townsend camera demo
- Ehcache & 2nd level cache
- Colorcache demo
- Quartz overview
- Quartz1 plugin
- Quartz2 plugin
- Grails 2 Quartz job demo
- Grails 2 Ehcache/Session/Quartz Demo

Clustering Basics

Not a group of nuts or grapes!

Software and hardware working together

Different levels of clustering for different purposes

Clustering With Terracotta

Many acting as one

We mean many servers, applications, and data grouped together

Covers caching of data for fast access across all nodes

Load balancers are the front line of most clustered setups

Clustering With Terracotta

Terracotta

Owners of Quartz and Ehache

Open source with most features free

Commercial if you need multiple active mirror groups

What Will We Cover?

Clustering basics

Reasons for clustering

Types of clustering

HTTP session clustering

Townsend camera demo

Ehcache & 2nd level cache

Colorcache demo

Quartz overview

Quartz1 plugin

Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Clustering With Terracotta

Terracotta

TCP based server – simplified networking

Central Primary Controller with hot spare

TIM Modules OSGi based for integration into many kinds of apps

Terracotta Primary Uses

Distributed HTTP Sessions

EhCache and 2nd Level Hibernate Cache

Distributed Quartz scheduling

Terracotta Commercial Only Features

Big Memory

Terracotta 'Enteprise' bundle

Extra security features with console, roles, etc

Clustering With Terracotta

Problem: Evenly distribute load across many servers

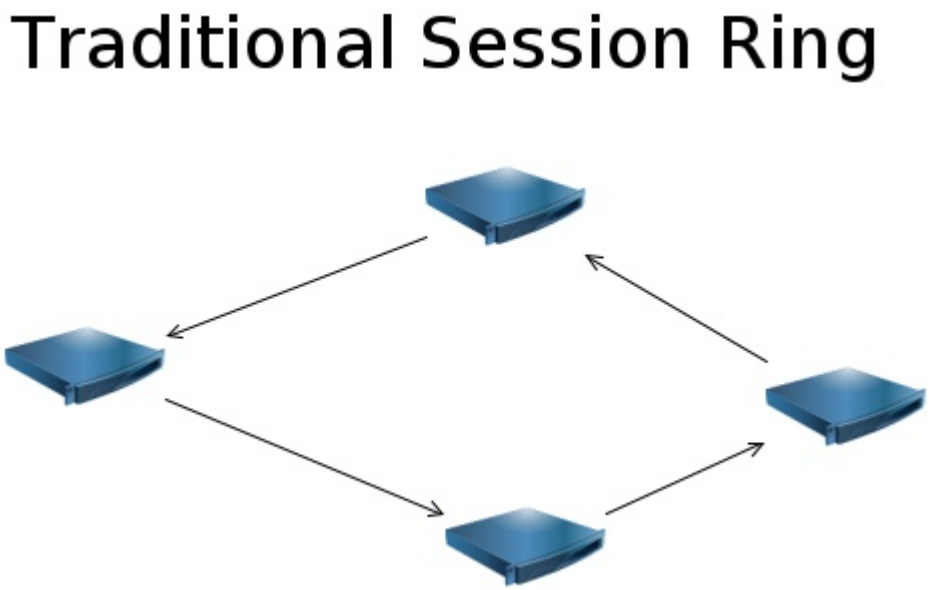
Many techniques to handle this

HTTP Session Ring Type on most web containers

What Will We Cover?

- Clustering basics
- Reasons for clustering
- Types of clustering
- HTTP session clustering
- Townsend camera demo
- Ehcache & 2nd level cache
- Colorcache demo
- Quartz overview
- Quartz1 plugin
- Quartz2 plugin
- Grails 2 Quartz job demo
- Grails 2 Ehcache/Session/Quartz Demo

Typical Web Session Cluster



Session data sent in a ring, in case one node fails

session is not lost

Other Techniques

Use load balancer with sticky source ip

Doesn't give balanced load – due to large IP blocks

sticking to one server (and large ISPs that use proxys)

Other Techniques

Save sessions in database

Too slow – serialization in and out of DB has dreadful performance

Clustering With Terracotta

What Will We Cover?

Clustering basics

Reasons for clustering

Types of clustering

HTTP session clustering

Townsend camera demo

Ehcache & 2nd level cache

Colorcache demo

Quartz overview

Quartz1 plugin

Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Terracotta Web Sessions

Easy setup

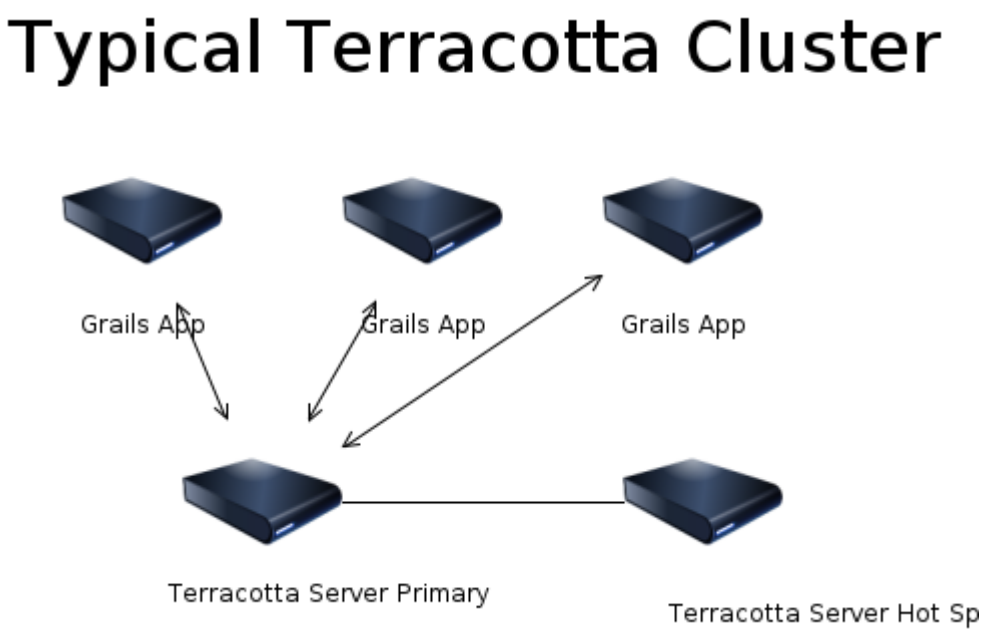
Session visibility

Scale

In Memory Speed

Efficient

Typical Terracotta Cluster



Session data sent to central server

Incremental changes only sent – much faster

Terracotta Web Sessions Setup

Tomcat 6 / 7 – add valve to context.xml

Websphere, Jboss, others

Add filter to web.xml

Terracotta Web Sessions Setup

Tomcat 6 / 7 – add valve to context.xml

```
<Valve
className="org.terracotta.session.TerracottaTomcat
60xSessionValve"
tcConfigUrl="localhost:9510">
```

Terracotta Web Sessions Setup

Copy runtime jar to container lib directory

common/terracotta-toolkit-1.1-runtime-2.1.0.jar

Terracotta Web Sessions Setup

JBoss – add valve to context.xml

```
<Valve
className="org.terracotta.session.TerracottaJboss51xSessionValve"
tcConfigUrl="localhost:9510">
```

Terracotta Web Sessions Setup

Jetty, Weblogic, Websphere

```
<filter>
  <filter-name>terracotta</filter-name>
  <!-- The filter class is specific to the application server. -->
  <filter-class>org.terracotta.session.<container-specific-
class></filter-class>
  <init-param>
    <param-name>tcConfigUrl</param-name>
    <param-value>localhost:9510</param-value>
  </init-param>
</filter>
```

Terracotta Web Sessions Setup

Jetty, Weblogic, Websphere

```
<filter-mapping>
  <!-- Must match filter name from above. -->
  <filter-name>terracotta</filter-name>
  <url-pattern>/*</url-pattern>
  <!-- Enable all available dispatchers. -->
  <dispatcher>ERROR</dispatcher>
  <dispatcher>INCLUDE</dispatcher>
  <dispatcher>FORWARD</dispatcher>
  <dispatcher>REQUEST</dispatcher>
</filter-mapping>
```

What Will We Cover?

Clustering basics

Reasons for clustering

Types of clustering

HTTP session clustering

Ehcache & 2nd level cache

Colorcache demo

Quartz overview

Quartz1 plugin

Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

What Will We Cover?

Clustering basics

Reasons for clustering

Types of clustering

HTTP session clustering

Townsend camera demo

Ehcache & 2nd level cache

Colorcache demo

Quartz overview

Quartz1 plugin

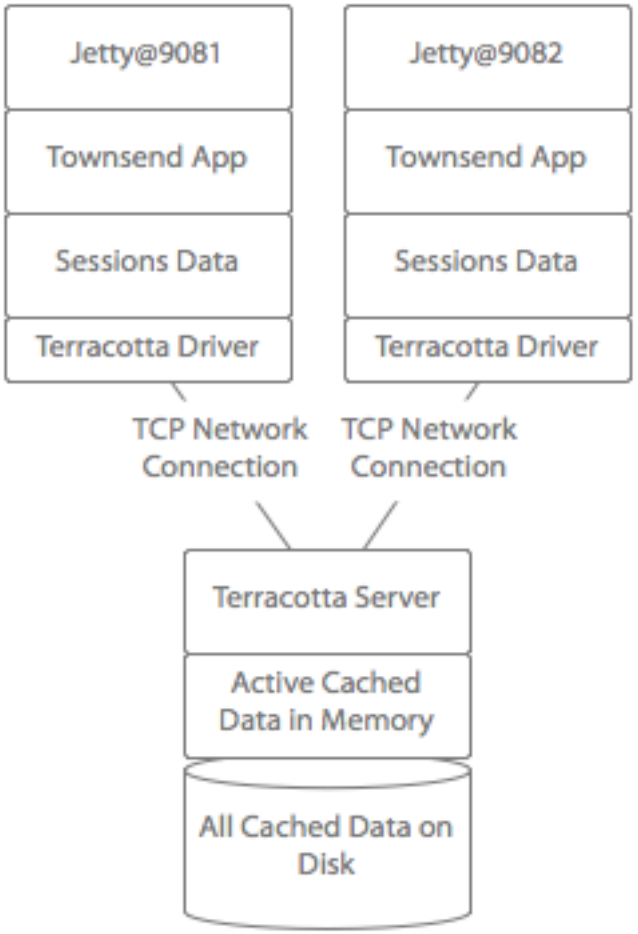
Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Clustering With Terracotta

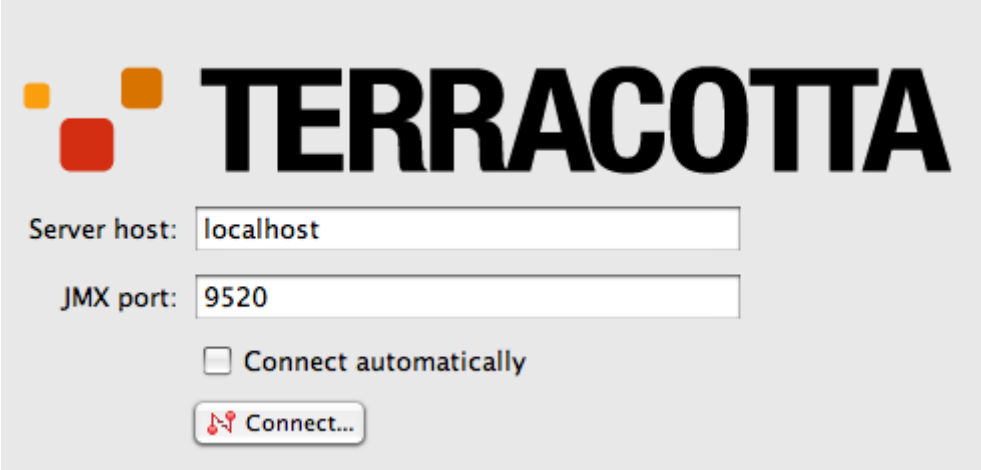
Terracotta Townsend App Demo



Terracotta Townsend App Demo

Clustering With Terracotta

Terracotta Townsend App Demo

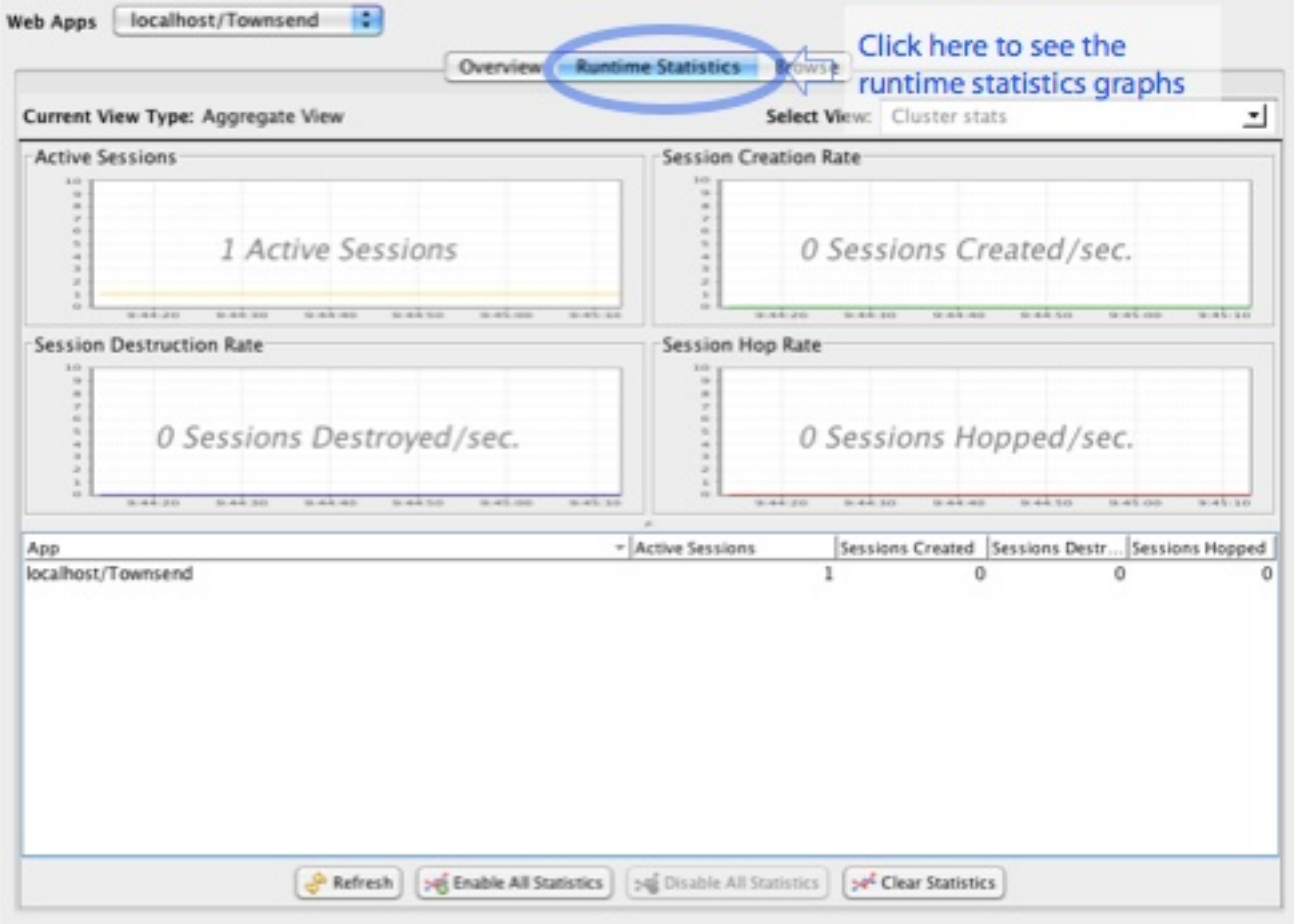


Clustering With Terracotta

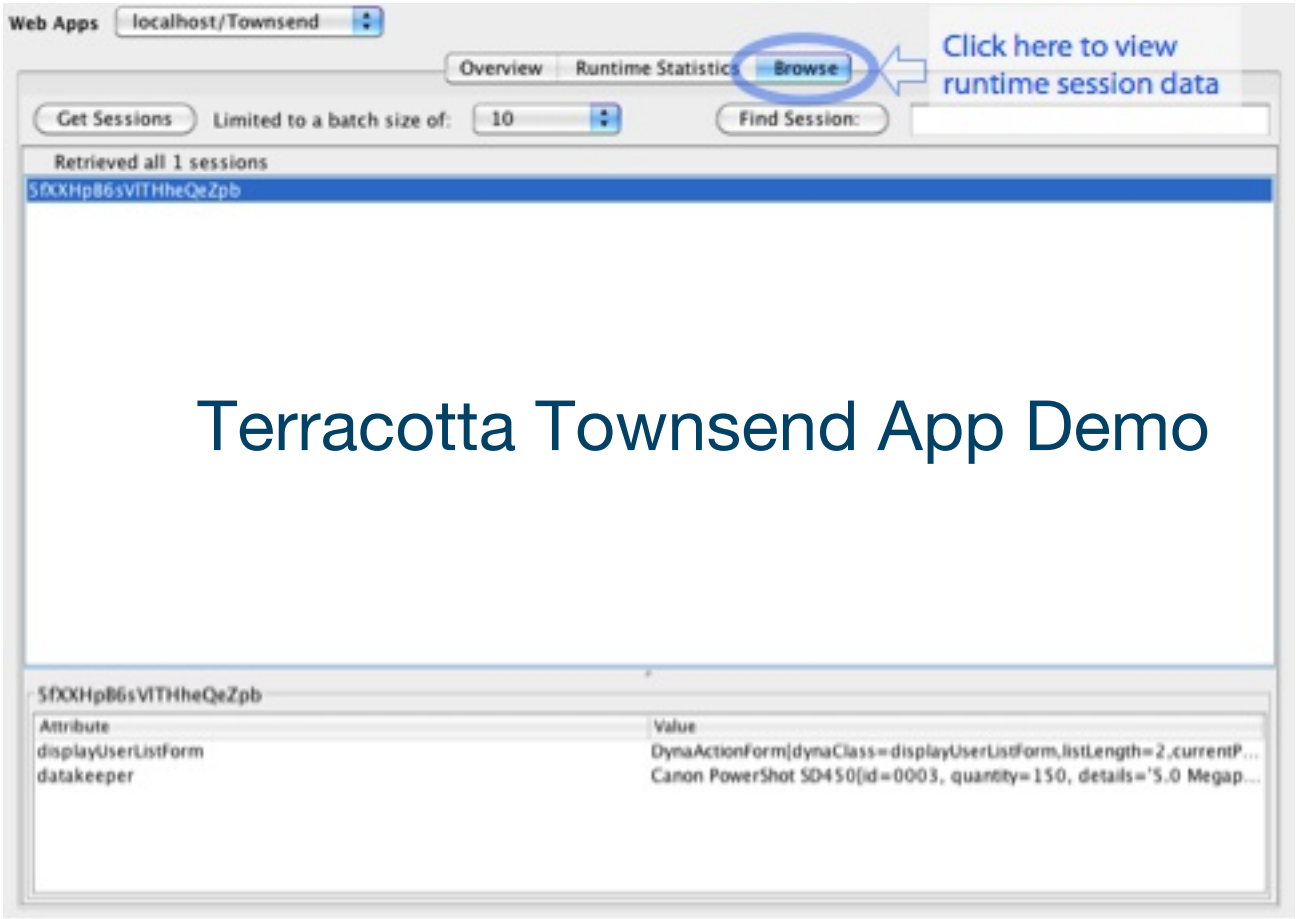
Terracotta Townsend App Demo

Clustering With Terracotta

Terracotta Townsend App Demo



Clustering With Terracotta



What Will We Cover?

Clustering basics

Reasons for clustering

Types of clustering

HTTP session clustering

Townsend camera demo

Ehcache & 2nd level cache

Colorcache demo

Quartz overview

Quartz1 plugin

Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Application / Database Caching

Why do I need this?

Take load off database

Increase performance

Ease scalability

Clustering With Terracotta

Solutions

Application / Database Caching

Ehcache

Jboss Cache

Jgroups + roll your own caches

Terracotta

Application / Database Caching

Ehcache

Fine on it's own but....

Hard to debug

Uses Jgroups under the cover but can cause odd

network issues with multicast
No real console to see what is

going on

Application / Database Caching

JBoss Cache

Fine on it's own but....

JbossCache Grails plugin only tested with Grails 1.1

Uses Jgroups under the cover but can cause odd network issues with multicast

No real console to see what is going on

Application / Database Caching

Jgroups + Roll Your Own

JGroups hard to manage

Responsibility to manage cache – why not let a framework help you?

No real console to see what is going on

Complex

Clustering With Terracotta

Application / Database Caching

Terracotta – Ehcache

Integrates into console

Centralized server makes setup and management less complex

Works in Grails 2.x

Fast – only transmits deltas

Application / Database Caching

Terracotta – Ehcache

Open source free edition sufficient for most installations
Great developer console tells you shows full insight of cache
Express install – easy to setup if you follow these slides

Valid Setup Versions

Terracotta – Ehcache

Tomcat Version	Quartz Plugin	Terracotta Version	Quartz TC Cluster ?	Quartz DB Cluster ?	Ehcache Version
6	1.8	3.4.1	Y	N	2.3.2
7	1.8	3.6.x	N	Y	2.5.2
7	2.x	3.7	N	Y	2.6.0
6	2.x	3.7	N	Y	2.3.2

Grails 2.x Setup Checklist

Terracotta – Ehcache

BuildConfig.groovy

Make sure versions of Ehcache-core match TC Install

Make sure TC Toolkit jar match TC install

Make sure ehcache-terracotta jar math TC install

Clustering With Terracotta

Terracotta 3.4.1 Grails 2.x Ehcache Setup – BuildConfig.groovy

```
<!-- delete newer ehcache core, version must match
TC 3.4.1's Ehcache version -->
  grails.war.resources = { stagingDir ->
    delete(file:"${stagingDir}/WEB-INF/lib/ehcache-
core-2.4.6.jar")
  }
  dependencies {
        runtime 'net.sf.ehcache:ehcache-core:2.3.2'
        runtime 'net.sf.ehcache:ehcache-terracotta:2.3.2'
        runtime 'org.terracotta:terracotta-toolkit-1.1-
runtime:2.1.0'
        runtime 'org.terracotta.quartz:quartz-
terracotta:1.2.1'
    }
```

What Will We Cover?

Clustering basics

Reasons for clustering

Types of clustering

HTTP session clustering

Townsend camera demo

Ehcache & 2nd level cache

Colorcache demo

Quartz overview

Quartz1 plugin

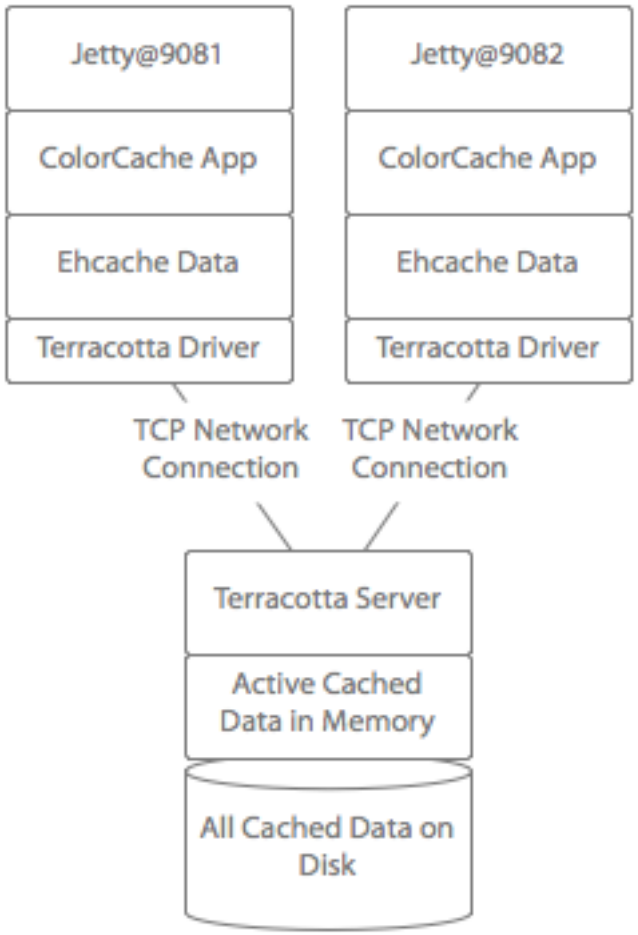
Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

ColorCache Demo

Terracotta – Ehcache



ColorCache Demo

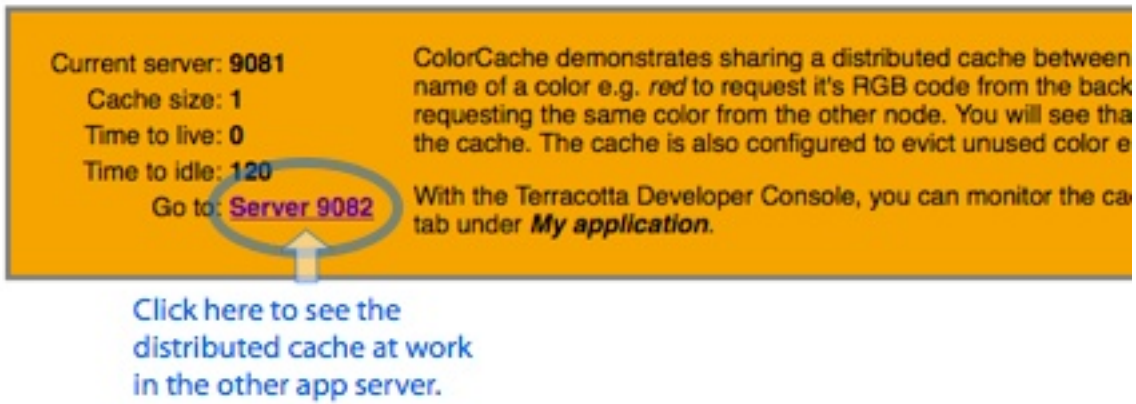
Terracotta – Ehcache

ColorCache Demo

Terracotta – Ehcache

ColorCache Demo

Terracotta – Ehcache



ColorCache Demo

Terracotta – Ehcache

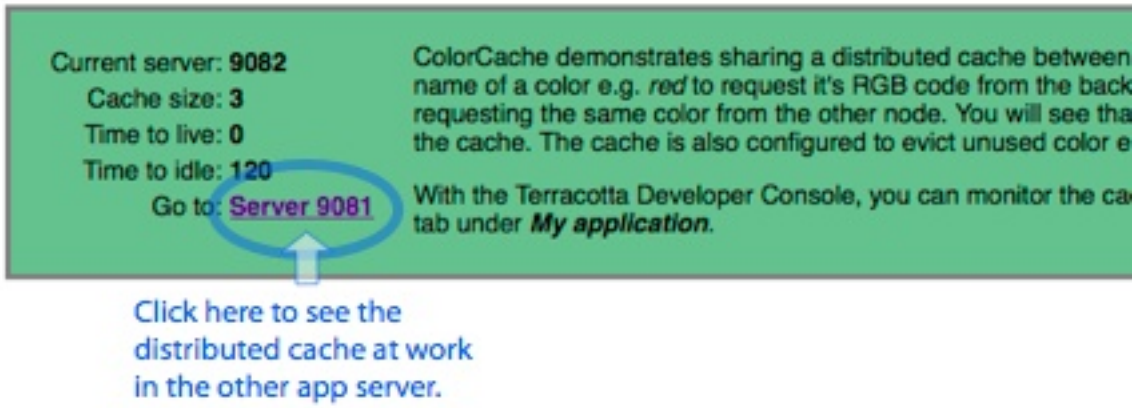
Click on these swatches to load them into the main display.

ColorCache Demo

Terracotta – Ehcache

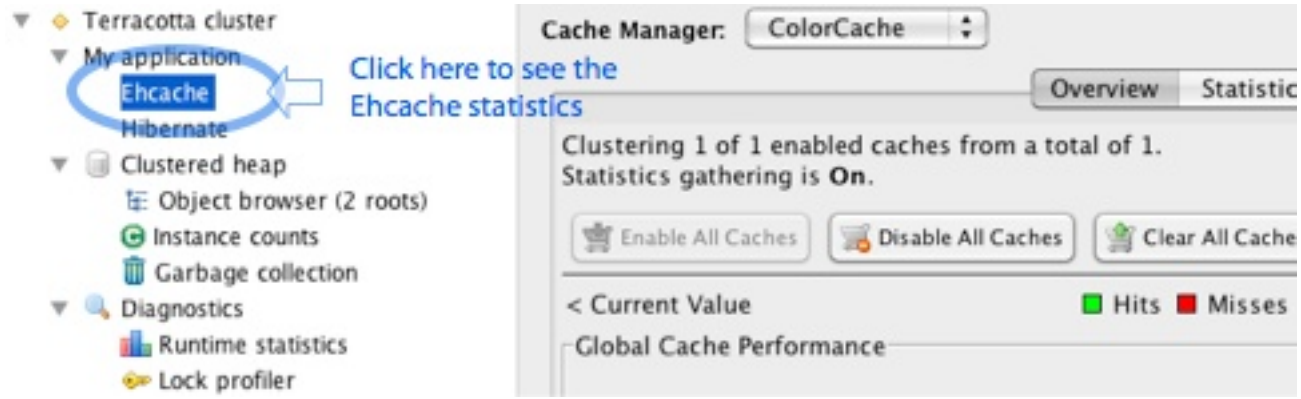
ColorCache Demo

Terracotta – Ehcache



ColorCache Demo

Terracotta – Ehcache

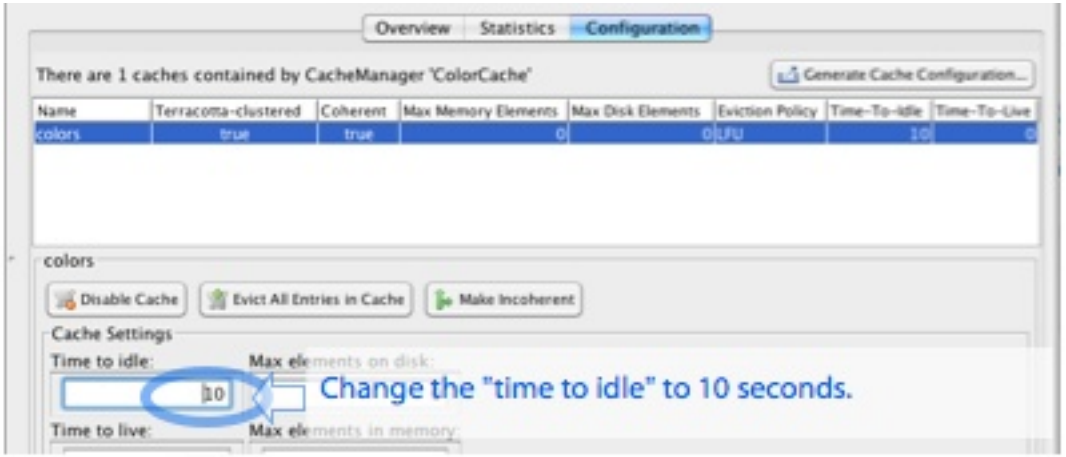


ColorCache Demo

Terracotta – Ehcache

Terracotta – Ehcache

ColorCache Demo



What Will We Cover?

- Clustering basics
- Reasons for clustering
- Types of clustering
- HTTP session clustering
- Townsend camera demo
- Ehcache & 2nd level cache
- Colorcache demo
- Quartz overview
- Quartz1 plugin
- Quartz2 plugin
- Grails 2 Quartz job demo
- Grails 2 Ehcache/Session/Quartz Demo

Clustering Quartz

What Is Quartz?

Open source Java API Used to schedule, persist, and distribute jobs

Great Grails support

Great community support and large usage (thousands)

Most common solution for scheduling execution in Java applications

Clustering Quartz

Quartz Plugin

Integrates Grails with Quartz 1.x

Works best with Clustered Terracotta Option due to bug in Terracotta (https://jira.terracotta.org/jira/browse/QTZ-310)

Yet to be updated to support Quartz 2

Officially supported by SpringSource

Clustering Quartz

Quartz2 Plugin

Integrates Grails with Quartz 2.x

Works best with Clustered JDBCStore due to bug in Terracotta (https://jira.terracotta.org/jira/browse/QTZ-310)

Supports Groovy based JobDetail

Not Officially supported by SpringSource

Supports (nosql) engines like

Mongo or Redis

Clustering Quartz

Why Cluster Quartz

Distribute Load

Scale easily

Handle many batch jobs at once

Persist scheduled work queue in case of crash

Fail-over

What Will We Cover?

Clustering basics

Reasons for clustering

Types of clustering

HTTP session clustering

Townsend camera demo

Ehcache & 2nd level cache

Colorcache demo

Quartz overview

Quartz1 plugin

Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Clustering Quartz

Installing Quartz and Creating a Job

grails install-plugin quartz

grails create-job <jobName>

```
class MyJob {
  static triggers = {
    simple name: 'mySimpleTrigger',
startDelay: 60000, repeatInterval:
1000
  }
  def group = "MyGroup"
  def execute(){
    print "Job run!"
  }
}
```

Installing Quartz and Creating a Job

Cron trigger example

```
class MyJob {
  static triggers = {
    cron name: 'myTrigger',
cronExpression: "0 0 6 * * ?"
  }
  def group = "MyGroup"
  def execute(){
    print "Job run!"
  }
}
```

Clustering Quartz

Installing Quartz and Creating a Job

Dynamic Jobs

```
// creates cron trigger;
MyJob.schedule(String cronExpression, Map params?)
//  creates simple trigger: repeats job repeatCount+1
times with delay of repeatInterval milliseconds;
MyJob.schedule(Long repeatInterval, Integer
repeatCount?, Map params?) )
```

Clustering Quartz

Installing Quartz and Creating a Job

Dynamic Jobs

```
// schedules one job execution to the specific
date;
MyJob.schedule(Date scheduleDate, Map
params?)
//schedules job's execution with a custom trigger;
MyJob.schedule(Trigger trigger)
// force immediate execution of the job.
MyJob.triggerNow(Map params?)
```

Clustering Quartz

Installing Quartz and Creating a Job

Dynamic Jobs

```
// Each method (except the one for custom trigger) takes
optional 'params' argument.
// You can use it to pass some data to your job and then
access it from the job:
class MyJob {
  def execute(context) {
    println context.mergedJobDataMap.get('foo')
  }
}
// now in your controller (or service, or something else):

MyJob.triggerNow([foo:"It Works!"])
```

Clustering Quartz

Prepping Your Environment for Distributed Quartz

How do you want to run the jobs?

- Replicate your grails app X times and distribute across that?
- Have separate replicated application the picks up the jobs
  - WAR file under application server

Clustering Quartz

Prepping Your Environment for Distributed Quartz

Standalone application

Just Java classes running scheduler command line?

Standalone plugin?

Custom standalone i.e.

https://gist.github.com/1804182 ?

Clustering Quartz

Terracotta Clustering

No Database setup required

Currently only works with Quartz 1.8 and 'quartz' plugin

Doesn't work with Quartz2 plugin because it implements a different class for the JobDetails interface that is not JobDetailsImpl. Terracotta will throw errors because it assume JosDetailsImpl class is used. JIRA logged at https://jira.terracotta.org/jira/browse/QTZ-310 if you'd like to vote on it

Clustering Quartz

Setting Up Open Source Terracotta Clustering

Download Terracotta 3.4.1 from (This is the last version

that is compatible with Quartz 1.8.x)

Install 'quartz' grails plugin

Create quartz.properties in grails-app/conf or src/java

Clustering Quartz

Setting Up Open Source Terracotta Clustering

Sample quartz.properties file:

```
org.quartz.scheduler.instanceName = MyClusteredScheduler
org.quartz.scheduler.instanceId = AUTO
# Configure ThreadPool
org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool
org.quartz.threadPool.threadCount = 25
org.quartz.threadPool.threadPriority = 5
org.quartz.jobStore.class=org.terracotta.quartz.TerracottaJobStore
# the path below should point to your terracotta config file. This
can can be a URL as well like http://
org.quartz.jobStore.tcConfigUrl = /opt/terracotta-3.4.1/tc-
config.xml
```

Clustering Quartz

Starting Terracotta

Download Terracotta 3.4.1 from

http://terracotta.org/downloads/open-source/destination?name=terracotta-3.4.1_1.tar.gz&bucket=tcdistributions&file=terracotta-3.4.1_1.tar.gz

Copy $TERRACOTTA_HOME/config-samples/tc-config-express-

reference.xml to $TERRACOTTA_HOME/tc-config.xml

Run ./start-tc-server.sh -f /path/to/tc-config.xml

Run ./dev-console.sh

Clustering Quartz

Go Advanced!

Use Job and Trigger Listeners (there is no groovy version of this, you will have to make regular class files) to split large jobs apart into smaller ones. (Cluster Bomb Pattern)

Example applications using distributed jobs:

- Email Campaign Tool
- Data processing
- Email Verification Tool

What Will We Cover?

Clustering basics

Reasons for clustering

Types of clustering

HTTP session clustering

Townsend camera demo

Ehcache & 2nd level cache

Colorcache demo

Quartz overview

Quartz1 plugin

Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Clustering Quartz

Setup and Run Quartz 2 Plugin

Use Job and Trigger Listeners (there is no groovy version of this, you will have to make regular class files) to split large jobs apart into smaller ones.

See Docs at https://github.com/9ci/grails-quartz2

Clustering Quartz

Setup and Run Quartz 2 Plugin

Use Job and Trigger Listeners (there is no groovy version of this, you will have to make regular class files) to split large jobs apart into smaller ones.

See Docs at https://github.com/9ci/grails-quartz2

What Will We Cover?

Clustering basics

Reasons for clustering

Types of clustering

HTTP session clustering

Townsend camera demo

Ehcache & 2nd level cache

Colorcache demo

Quartz overview

Quartz1 plugin

Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Clustering Quartz

Terracotta Demo

2 Nodes CreatePersonJob

Clustering Quartz

Setup and Run Quartz 2 Plugin

Example applications using distributed jobs:

- Email Campaign Tool
- Data processing
- Email Verification Tool

What Will We Cover?

- Clustering basics
- Reasons for clustering
- Types of clustering
- HTTP session clustering
- Townsend camera demo
- Ehcache & 2nd level cache
- Colorcache demo
- Quartz overview
- Quartz1 plugin
- Quartz2 plugin
- Grails 2 Quartz job demo
- Grails 2 Ehcache/Session/Quartz Demo

More Information

https://github.com/9ci/grails-quartz2
http://terracotta.org/downloads/open-source/catalog
https://jira.terracotta.org/jira/browse/QTZ-310
http://grails.org/plugin/quartz
http://grails-plugins.github.com/grails-quartz/
http://quartz-scheduler.org/

Contact Me

Via twitter: https://twitter.com/RyanVanderwerf
Google+/email: rvanderwerf@gmail.com
Blog: http://rvanderwerf.blogspot.com

Copyright Notices