

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.

Co-Chair of Austin Groovy and Grails User Group

Help maintain various plugins – Quartz, GVPS (Video), Struts1
anywhere I can do pull requests for bug fixes (you should too!)

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

Grails-Quartz plugin

Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Clustering With Terracotta

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 Ehcache

Open source with most features free

Commercial if you need multiple active mirror groups

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

Grails-Quartz 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

Clustering With Terracotta

Terracotta Primary Uses

Distributed HTTP Sessions

EhCache and 2nd Level Hibernate Cache

Distributed Quartz scheduling

Clustering With Terracotta

Terracotta Commercial Only Features

Big Memory

Terracotta 'Enterprise' bundle

Extra security features with console, roles, etc

Quartz Location API

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

Grails-Quartz plugin

Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Clustering With Terracotta

Problem: Evenly distribute load across many servers

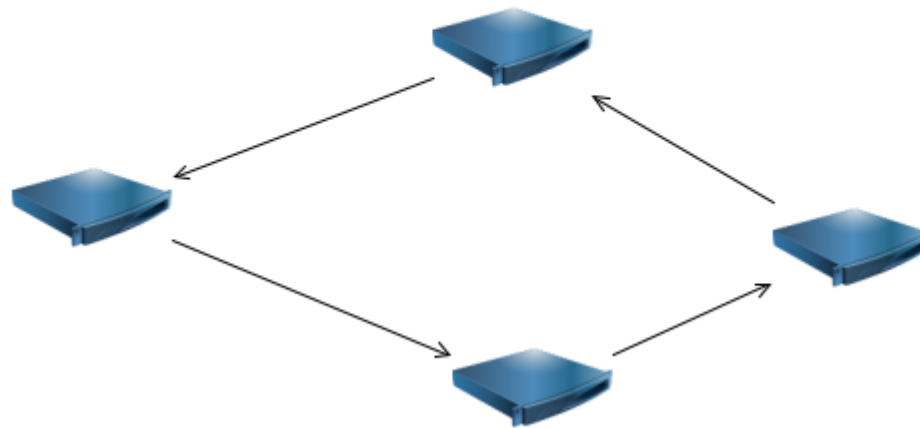
Many techniques to handle this

HTTP Session Ring Type on most web containers

Clustering With Terracotta

Typical Web Session Cluster

Traditional Session Ring



Session data sent in a ring, in case one node fails
session is not lost

Clustering With Terracotta

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)

Clustering With Terracotta

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

Grails-Quartz plugin

Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Clustering With Terracotta

Terracotta Web Sessions

Easy setup

Session visibility

Scale

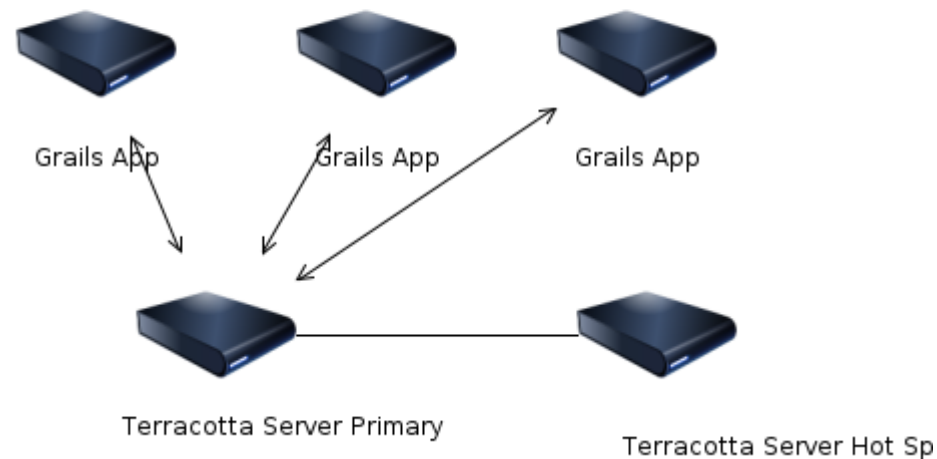
In Memory Speed

Efficient

Clustering With Terracotta

Typical Terracotta Cluster

Typical Terracotta Cluster



Session data sent to central server

Incremental changes only sent – much faster

Clustering With Terracotta

Terracotta Web Sessions Setup

Tomcat 6 / 7 – add valve to context.xml

Websphere, Jboss, others

Add filter to web.xml

Clustering With Terracotta

Terracotta Web Sessions Setup

Tomcat 6 / 7 – add valve to context.xml

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

Clustering With Terracotta

Terracotta Web Sessions Setup

Copy runtime jar to container lib directory

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

Clustering With Terracotta

Terracotta Web Sessions Setup

JBoss – add valve to context.xml

```
<Valve  
  className="org.terracotta.session.Terracotta  
  Jboss51xSessionValve"  
  tcConfigUrl="localhost:9510">
```

Clustering With Terracotta

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>
```

Clustering With Terracotta

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>
```


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

Grails-Quartz plugin

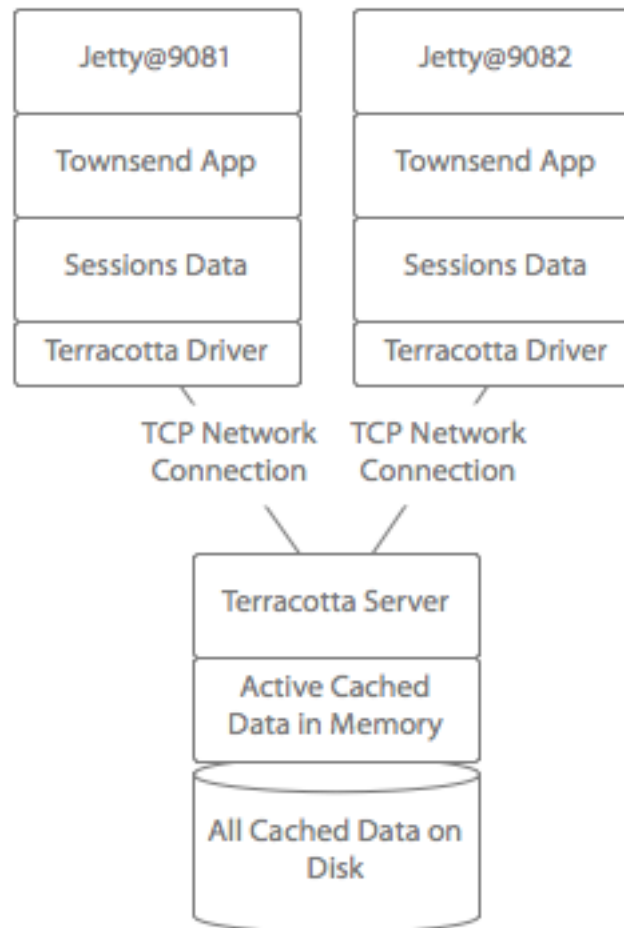
Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Clustering With Terracotta

Terracotta Townsend App Demo




Clustering With Terracotta

Terracotta Townsend App Demo



Clustering With Terracotta

Terracotta Townsend App Demo




TERRACOTTA

Server host:

JMX port:

☐ Connect automatically

 [Connect...](#)

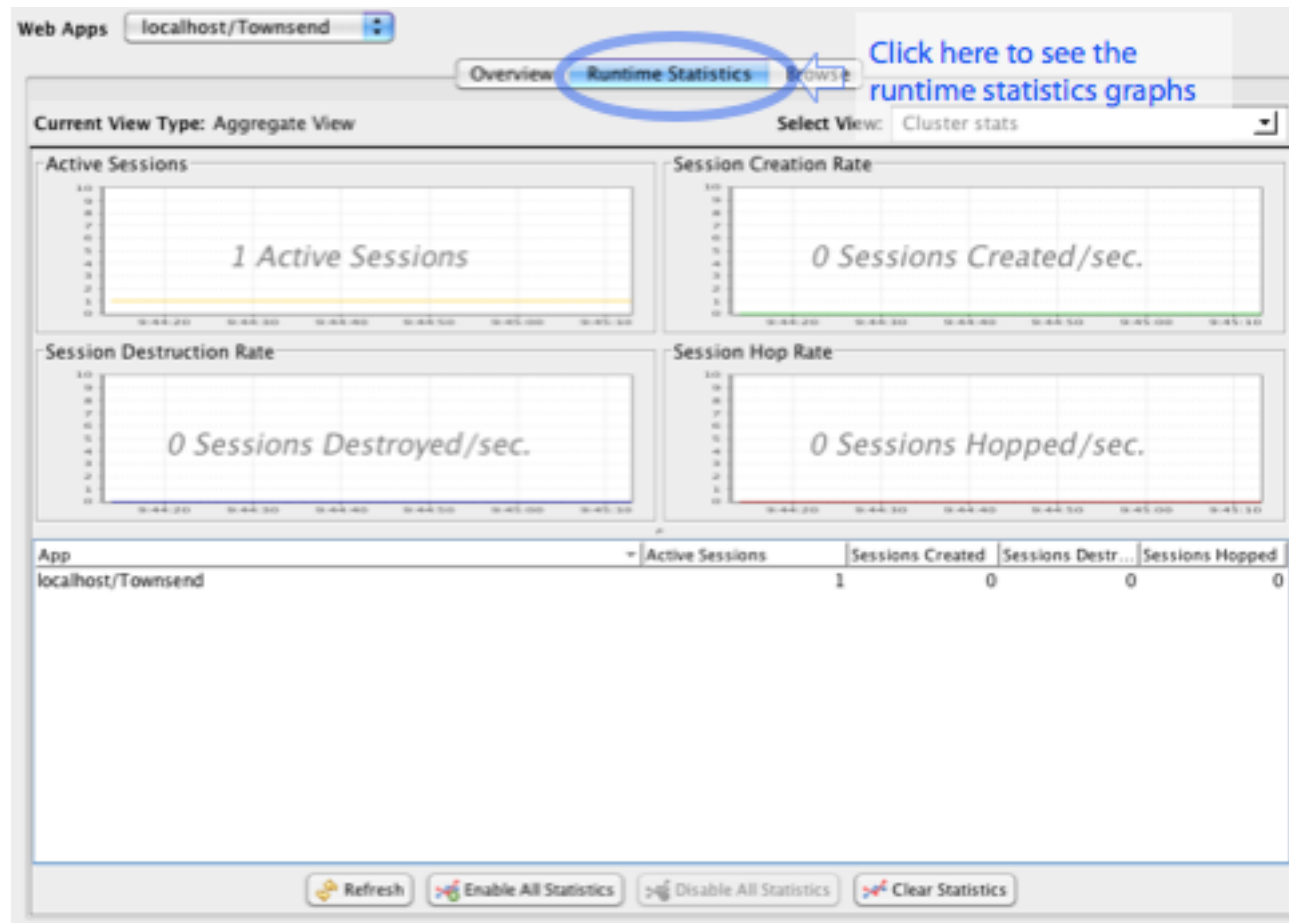
Clustering With Terracotta

Terracotta Townsend App Demo

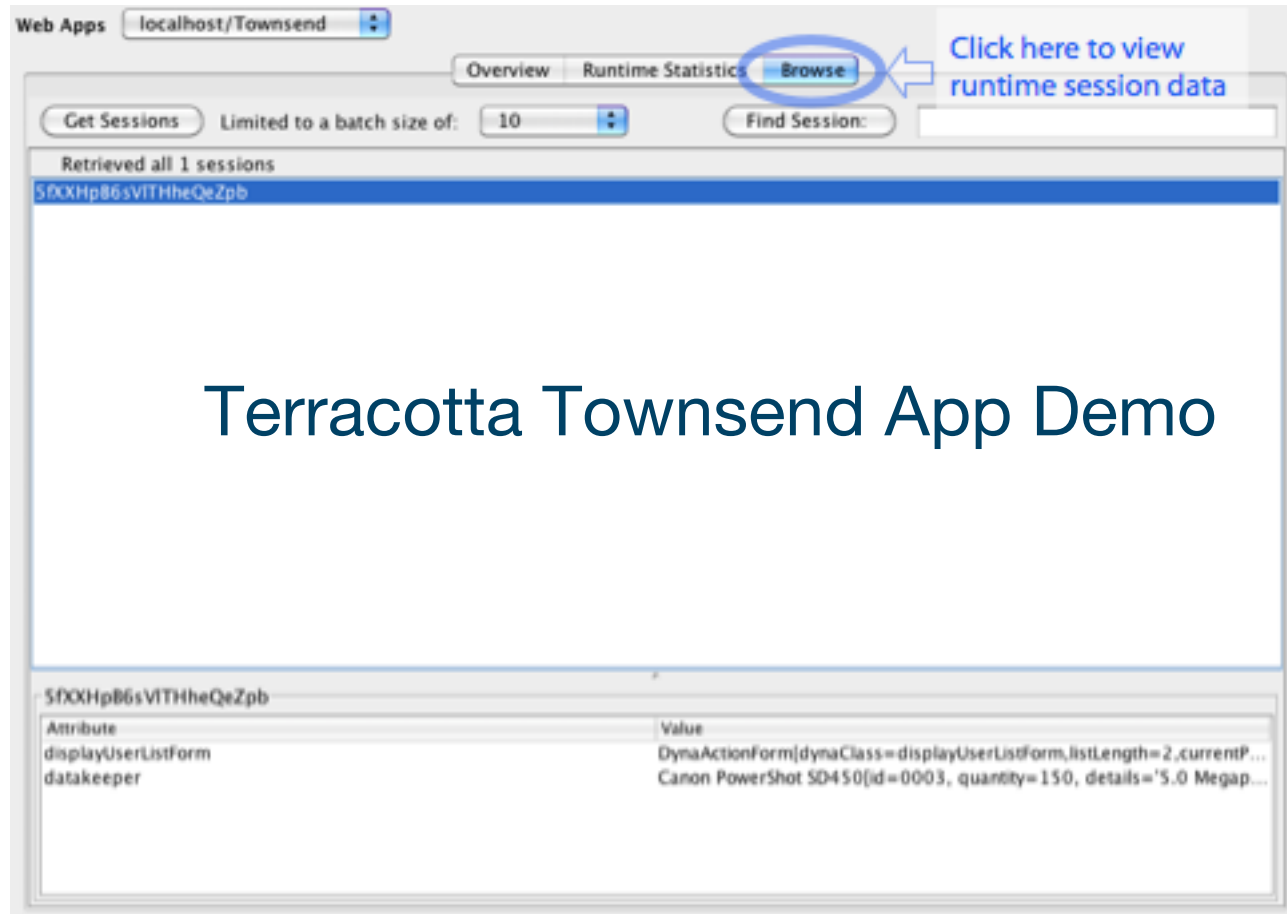


Clustering With Terracotta

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

Grails-Quartz plugin

Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Clustering With Terracotta

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

Clustering With 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 cloud

No real console to see what is

going on

Clustering With Terracotta

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

Clustering With Terracotta

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

Works in Grails 2.x
complex

Fast – only transmits deltas

Clustering With Terracotta

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

Clustering With Terracotta

Valid Setup Versions – Updated!

Terracotta – Ehcache

Tomcat Version	Quartz Plugin	Terracotta Version	Quartz TC Cluster ?	Quartz DB Cluster ?	Ehcache Version
6	2.x	3.7.5	Y	Y	2.6.6
7	2.x	3.7.5	Y	Y	2.6.6
7	2.x	3.6.x	Y	Y	2.5.2
6	2.x	3.6.x	Y	Y	2.5.2

Clustering With Terracotta

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 3.7.5 Grails 2.x Ehcache Setup – BuildConfig.groovy

```
<!-- delete grails ehcache core, version must match  
TC 3.7.5's Ehcache version -->  
  
dependencies {  
    runtime 'net.sf.ehcache:ehcache-core:2.6.6'  
    runtime 'net.sf.ehcache:ehcache-terracotta:2.6.6'  
    runtime  
'org.terracotta:terracotta-toolkit-1.6-runtime:5.5.0'  
    runtime  
'org.terracotta.quartz:quartz-terracotta:2.3.5'  
}
```

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

Grails-Quartz plugin

Quartz2 plugin

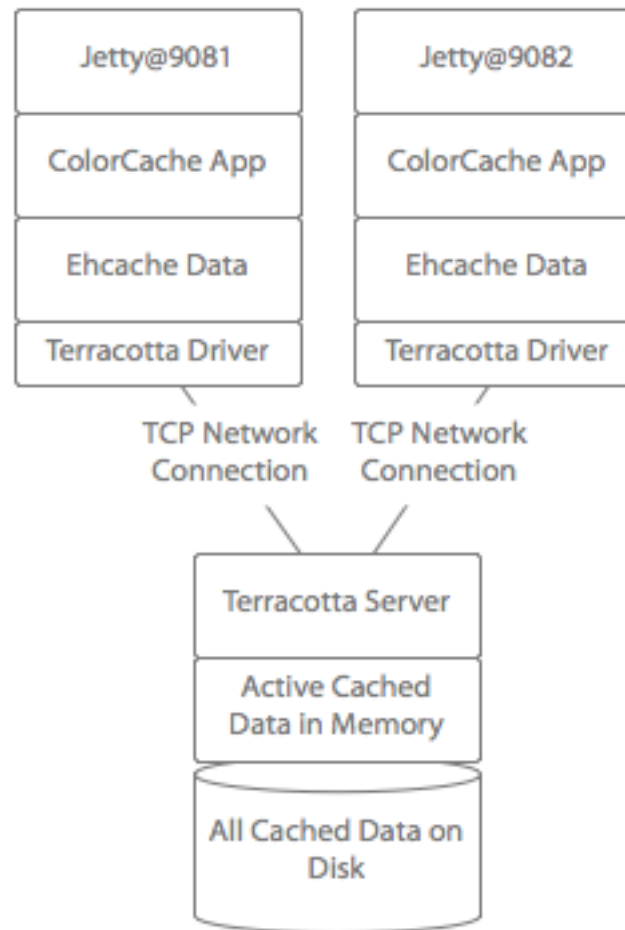
Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Clustering With Terracotta

ColorCache Demo

Terracotta – Ehcache



Clustering With Terracotta

ColorCache Demo

Terracotta – Ehcache

ColorCache

Enter Color Name:

No Color Selected

Clustering With Terracotta

ColorCache Demo

Terracotta – Ehcache

ColorCache

Enter Color Name:

Retrieve Color



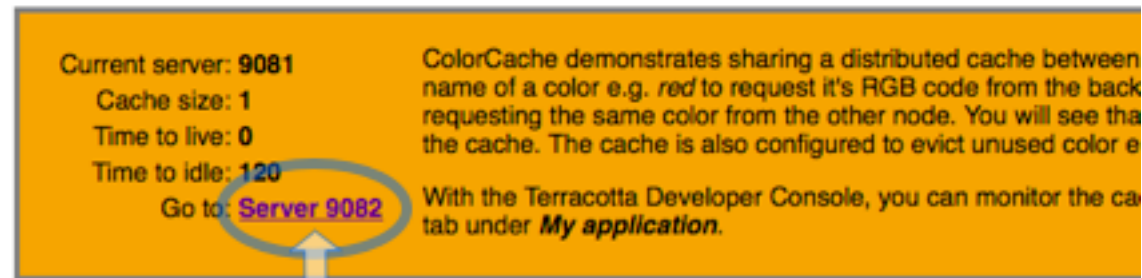
Color 'red' [255,0,0] retrieved in 3673 milliseconds.

Not cached = slow

Clustering With Terracotta

ColorCache Demo

Terracotta – Ehcache



The screenshot shows a yellow rectangular interface for the ColorCache demo. On the left, it displays server status: 'Current server: 9081', 'Cache size: 1', 'Time to live: 0', and 'Time to idle: 120'. Below this is a 'Go to:' label followed by 'Server 9082', which is circled in blue with a blue arrow pointing to it. On the right, there is explanatory text about the demo and a note about the Terracotta Developer Console.

Current server: 9081
Cache size: 1
Time to live: 0
Time to idle: 120
Go to: **Server 9082**

ColorCache demonstrates sharing a distributed cache between name of a color e.g. *red* to request it's RGB code from the back requesting the same color from the other node. You will see that the cache. The cache is also configured to evict unused color e

With the Terracotta Developer Console, you can monitor the ca tab under *My application*.

Click here to see the distributed cache at work in the other app server.

Clustering With Terracotta

ColorCache Demo

Terracotta – Ehcache



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

Clustering With Terracotta

ColorCache Demo

Terracotta – Ehcache

ColorCache

Enter Color Name:

Retrieve Color



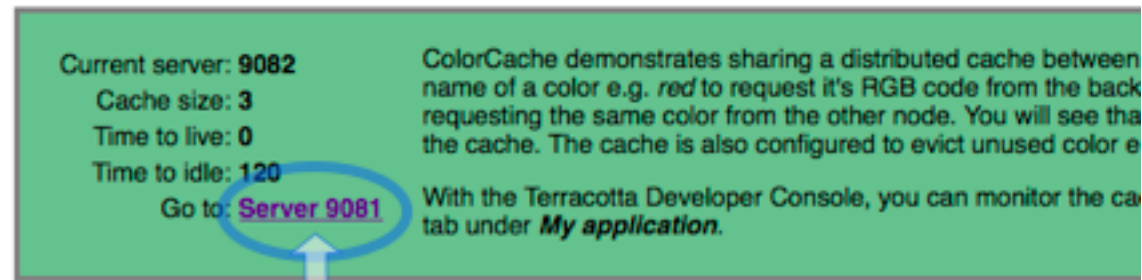
Color 'green' [0,255,0] retrieved in 7 milliseconds.

Cached = fast

Clustering With Terracotta

ColorCache Demo

Terracotta – Ehcache



Current server: 9082
Cache size: 3
Time to live: 0
Time to idle: 120
Go to: [Server 9081](#)

ColorCache demonstrates sharing a distributed cache between name of a color e.g. *red* to request it's RGB code from the back requesting the same color from the other node. You will see that the cache. The cache is also configured to evict unused color e

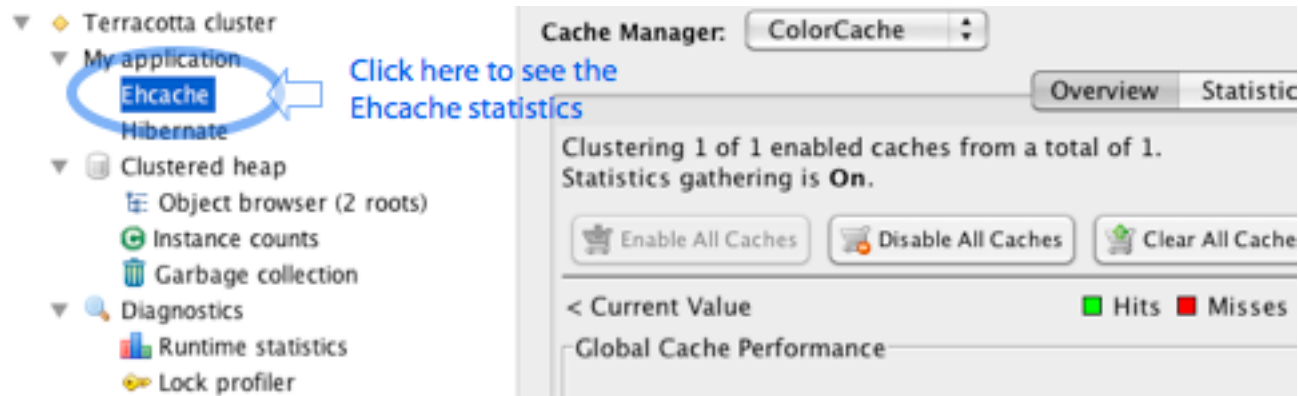
With the Terracotta Developer Console, you can monitor the ca tab under *My application*.

Click here to see the distributed cache at work in the other app server.

Clustering With Terracotta

ColorCache Demo

Terracotta – Ehcache



Clustering With Terracotta

ColorCache Demo

Terracotta – Ehcache



Clustering With Terracotta

Terracotta – Ehcache

ColorCache Demo

The screenshot shows the 'Configuration' tab of the Terracotta Ehcache management interface. At the top, there are tabs for 'Overview', 'Statistics', and 'Configuration'. Below the tabs, a message states 'There are 1 caches contained by CacheManager 'ColorCache'' with a 'Generate Cache Configuration...' button. A table lists the cache details:

Name	Terracotta-clustered	Coherent	Max Memory Elements	Max Disk Elements	Eviction Policy	Time-To-idle	Time-To-Live
colors	true	true	0	0	LFU	10	0

Below the table, there are buttons for 'Disable Cache', 'Evict All Entries in Cache', and 'Make Incoherent'. Under the 'Cache Settings' section, the 'Time to idle' is set to 10, which is circled in blue. A blue arrow points from the text 'Change the "time to idle" to 10 seconds.' to the circled value. Other settings include 'Max elements on disk' and 'Max elements in memory'.

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

Grails-Quartz 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

Grails-Quartz Plugin

Integrates Grails with Quartz 2.x (yay!)

Works best with Clustered Terracotta Option due to bug in Terracotta (<https://jira.terracotta.org/jira/browse/QTZ-310>) - update – fixed! Will be in next release

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>) fixed in next major Quartz release!

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

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

Grails-Quartz plugin

Quartz2 plugin

Grails 2 Quartz job demo

Grails 2 Ehcache/Session/Quartz Demo

Clustering Quartz

Installing Quartz and Creating a Job

Add plugin to BuildConfig.groovy:

grails create-job <jobName>

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

Clustering Quartz

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 has remote GUI console

Terracotta shows quick status of jobs

Terracotta doesn't require a database to persist jobs

Terracotta vs. JDBCJobStore Clustering
Terracotta Open Source Free

Good support via Terracotta.org forums

Both solutions should have a time server synchronizing
clocks on all machines

Clustering Quartz

Most common and least work to configure

Set up JobStore and Delegate

Configure database clustering prefix

Setup database schema

Works with Grails 'quartz' (Quartz 2.x) or

'quartz2' (Quartz 2) plugin

Clustering Quartz

Terracotta Clustering

No Database setup required

Now works with Quartz 2.x 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> - update fixed for next major Quartz release (api change)

Clustering Quartz

Slick GUI Interface

Excellent failover and hot spare ability, as well as integrating with Hibernate 2nd level cache, and HTTP Session caching

Terracotta Clustering

Open source edition free, works great for most small to medium installs

Commercial version allows more than one active cluster server and role based access to admin console

Clustering Quartz

Setting Up Open Source Terracotta Clustering

Download Terracotta 3.7.5 OS from Terracotta.org

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 be a URL as well like http://
org.quartz.jobStore.tcConfigUrl =
/opt/terracotta-3.7.5/tc-config.xml
```


Clustering Quartz

Starting Terracotta

Download Terracotta 3.7.5 from Terracotta.org

<http://www.terracotta.org/downloads/open-source/>

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

JDBCJobStore Clustering

Simple to configure

Schema to create tables included in distribution

Uses central database to persist jobs, and unique nodeIDs (AUTO will work)

Time server must sync instances regularly, and be within 1 second of each other

Works with Quartz 2.x or Quartz 2.x ('quartz' or 'quartz2' plugin)

Clustering Quartz

JDBCJobStore Clustering

- Sample quartz config highlights (see code samples for complete file)
 - `org.quartz.jobStore.isClustered = true`
 - `org.quartz.jobStore.clusterCheckinInterval = 20000`
 - `org.quartz.jobStore.misfireThreshold = 60000`
 - `org.quartz.jobStore.class =`
`org.quartz.impl.jdbcjobstore.JobStoreTX`
 - `org.quartz.jobStore.driverDelegateClass =`
`org.quartz.impl.jdbcjobstore.oracle.OracleDelegate`

Clustering Quartz

Sample quartz config highlights (see code samples for complete file)

```
org.quartz.jobStore.isClustered = true
```

```
org.quartz.jobStore.clusterCheckinInterval = 20000
```

```
org.quartz.jobStore.misfireThreshold = 60000
```

```
org.quartz.jobStore.class =
```

```
org.quartz.impl.jdbcjobstore.JobStoreTX
```

```
org.quartz.jobStore.driverDelegateClass =
```

```
org.quartz.impl.jdbcjobstore.oracle.OracleDelegate
```

JDBC JobStore Clustering

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

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

Grails-Quartz 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>

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 Ehcache/Session/Quartz Demo

Clustering Quartz

Setup and Run Quartz 2 Plugin

Example applications using distributed jobs:

Email Campaign Tool

Data processing

Email Verification Tool

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 Ehcache/Session/Quartz Demo

Clustering Terracotta

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: @RyanVanderwerf

Google+/email: rvanderwerf@gmail.com

Blog: <http://rvanderwerf.blogspot.com>

Copyright Notices

Page 1 Image: Creative Commons by Striving to a goal
Cluster Bomb Picture: Sascha Grant (Creative Commons)
Screenshots for ColorCache and Townsend property of
Terracotta, Inc.