

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)



E-mail:

Contraseña:

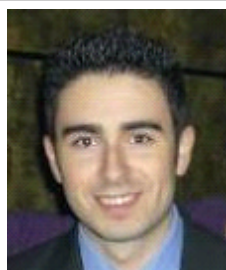
[Deseo registrarme](#) [Entrar](#)

[He olvidado mis datos de acceso](#)

[Inicio](#) [Quiénes somos](#) [Tutoriales](#) [Formación](#) [Comparador de salarios](#) [Nuestro libro](#) [Charlas](#) [Más](#)

✂ Estás en:

[Inicio](#) [Tutoriales](#) Monitoriza tu cache con Jconsole activando Jmx con Ehcache integrado con Sp...



DESARROLLADO POR:
[Alberto Barranco Ramón](#)

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/JEE

[Ver tutoriales de Alberto Barranco Ramón](#)

[Catálogo de servicios Autentia](#)

ADSL 20Mb + Llamadas Internet Máxima Velocidad **15,95** €/mes

Fecha de publicación del tutorial: 2011-12-21



Share |

0

[Regístrate para votar](#)

Monitoriza tu cache con Jconsole activando Jmx con Ehcache integrado con Spring

0. Índice de contenidos.

- 1. Introducción
- 2. Entorno
- 3. Activando Jmx para Ehcache mediante Spring
- 4. Monitorizar la aplicación con jConsole
 - 4.1 Gestión de la caché desde jConsole
- 5. Conclusiones
- 6. Información sobre el autor

1. Introducción

Si lo que estás buscando es una forma de monitorizar tu caché, saber cuantos elementos se están cacheando, y poder realizar operaciones sobre esa caché, o simplemente como empezar a utilizar jConsole en tu aplicación este tutorial quizás pueda ayudarte.

El proyecto de ejemplo sobre el que vamos a realizar el tutorial está usando **Ehcache** para gestionar la caché y **Spring** para la inyección de dependencias.

2. Entorno

- Hardware: Portátil MacBook Pro 15' (2.0 GHz Intel i7, 8GB DDR3 SDRAM, 500GB HDD).
- AMD Radeon HD 6490M 256 MB
- Sistema Operativo: Mac OS X Snow Leopard 10.6.7
- Software: Eclipse Helios

Últimas Noticias

- [iiiTerrakas en Antena 3!!!](#)
- [AdictosAlTrabajo, en la lista final de candidatos a los III Premios Focus al Conocimiento Libre](#)
- [TERRAKAS - Estrenamos el segundo capítulo!!!](#)
- [SoLiMadrid celebra su primer Encuentro de Empresas](#)
- [Mi paso por el Global Day Of Coderetreat 2011](#)

[Histórico de NOTICIAS](#)

Últimos Tutoriales

- [jQuery Mobile: Desarrollo de aplicaciones y portales web para dispositivos móviles](#)
- [Simulación de navegación manteniendo la sesión con wget](#)

3. Activando Jmx para Ehcache mediante Spring




Hasta ahora si tenemos activada la caché en nuestra aplicación tendremos declarado en algún applicationContext algo parecido a lo siguiente:

```
view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02.   <beans xmlns="http://www.springframework.org/schema/beans"
03.         xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:context="http://www.springframework.org/schema/context"
04.         xmlns:util="http://www.springframework.org/schema/util"
05.         xmlns:cache="http://www.springframework.org/schema/cache"
06.         xmlns:tx="http://www.springframework.org/schema/tx"
07.         xmlns:p="http://www.springframework.org/schema/p"
08.         xmlns:aop="http://www.springframework.org/schema/aop"
09.         xsi:schemaLocation="
10.             http://www.springframework.org/schema/beans
11.             http://www.springframework.org/schema/beans/spring-beans-
3.1.xsd
12.             http://www.springframework.org/schema/context
13.             http://www.springframework.org/schema/context/spring-
context-3.0.xsd
14.             http://www.springframework.org/schema/util
15.             http://www.springframework.org/schema/util/spring-util-
3.0.xsd
16.             http://www.springframework.org/schema/cache
17.             http://www.springframework.org/schema/cache/spring-
cache.xsd
18.             http://www.springframework.org/schema/aop
19.             http://www.springframework.org/schema/aop/spring-aop-
3.0.xsd
20.             http://www.springframework.org/schema/tx
21.             http://www.springframework.org/schema/tx/spring-tx-
3.0.xsd"
22.   >
23.
24.   <context:annotation-config />
25.   <cache:annotation-driven/>
26.   <tx:annotation-driven/>
27.
28.   <bean id="cacheManager" class="org.springframework.cache.ehcache.EhCache
29.         p:cache-manager-ref="ehcache" />
30.
31.   <bean id="ehcache" class="org.springframework.cache.ehcache.EhCacheMana
32.         p:config-location="classpath:ehcache.xml" />
```



El objetivo de este tutorial no es explicar como configurar la caché, para eso tenéis el tutorial **Spring cache abstraction** de mi compañero **Jose Manuel** en donde se explica paso a paso la configuración de **Ehcache**.

Es importante decir que el soporte para **Jmx** por defecto no está activado, además de estar en un paquete distinto del core de Ehcache, (en concreto net.sf.ehcache.management), tal y como podéis ver en la **documentación oficial**. Para activarlo debéis incluir los siguientes beans:

```
view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02.   <beans xmlns="http://www.springframework.org/schema/beans"
03.         xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:context="http://www.springframework.org/schema/context"
04.         xmlns:util="http://www.springframework.org/schema/util"
05.         xmlns:cache="http://www.springframework.org/schema/cache"
06.         xmlns:tx="http://www.springframework.org/schema/tx"
07.         xmlns:p="http://www.springframework.org/schema/p"
08.         xmlns:aop="http://www.springframework.org/schema/aop"
09.         xsi:schemaLocation="
10.             http://www.springframework.org/schema/beans
11.             http://www.springframework.org/schema/beans/spring-beans-
3.1.xsd
12.             http://www.springframework.org/schema/context
13.             http://www.springframework.org/schema/context/spring-
context-3.0.xsd
14.             http://www.springframework.org/schema/util
15.             http://www.springframework.org/schema/util/spring-util-
3.0.xsd
16.             http://www.springframework.org/schema/cache
17.             http://www.springframework.org/schema/cache/spring-
```

-  Varias cuentas de gmail en Chrome
-  Apache Camel, primeros pasos
-  Tomcat Cluster con JSF

Últimos Tutoriales del Autor

-  EpubLib, una librería Java para leer Epub
-  Cambiando el plugin de eclipse para Maven, de IAM a m2eclipse.
-  Autentia en la javacup
-  Mi experiencia en Autentia
-  Cómo listar una entidad en wuija con prefiltrado

Síguenos a través de:



Últimas ofertas de empleo

- 2011-09-08  Comercial - Ventas - MADRID.
- 2011-09-03  Comercial - Ventas - VALENCIA.
- 2011-08-19  Comercial - Compras - ALICANTE.
- 2011-07-12  Otras Sin catalogar - MADRID.
- 2011-07-06  Otras Sin catalogar - LUGO.

```

18. cache.xsd          http://www.springframework.org/schema/aop
19.                   http://www.springframework.org/schema/aop/spring-aop-
20. 3.0.xsd            http://www.springframework.org/schema/tx
21.                   http://www.springframework.org/schema/tx/spring-tx-
22. 3.0.xsd"           >
23.
24.     <context:annotation-config />
25.     <cache:annotation-driven/>
26.     <tx:annotation-driven/>
27.
28.     <bean id="cacheManager" class="org.springframework.cache.ehcache.EhCache
29.         p:cache-manager-ref="ehcache" />
30.
31.     <bean id="ehcache" class="org.springframework.cache.ehcache.EhCacheMana
32.         p:config-location="classpath:ehcache.xml" />
33.
34.     <bean id="mbeanServer" class="org.springframework.jmx.support.MBeanServ
35.         <property name="locateExistingServerIfPossible" value="true" />
36.     </bean>
37.
38.     <bean class="net.sf.ehcache.management.ManagementService"
39.         init-method="init">
40.         <constructor-arg ref="ehcache" />
41.         <constructor-arg ref="mbeanServer" />
42.         <constructor-arg value="true" />
43.         <constructor-arg value="true" />
44.         <constructor-arg value="true" />
45.         <constructor-arg value="true" />
46.     </bean>

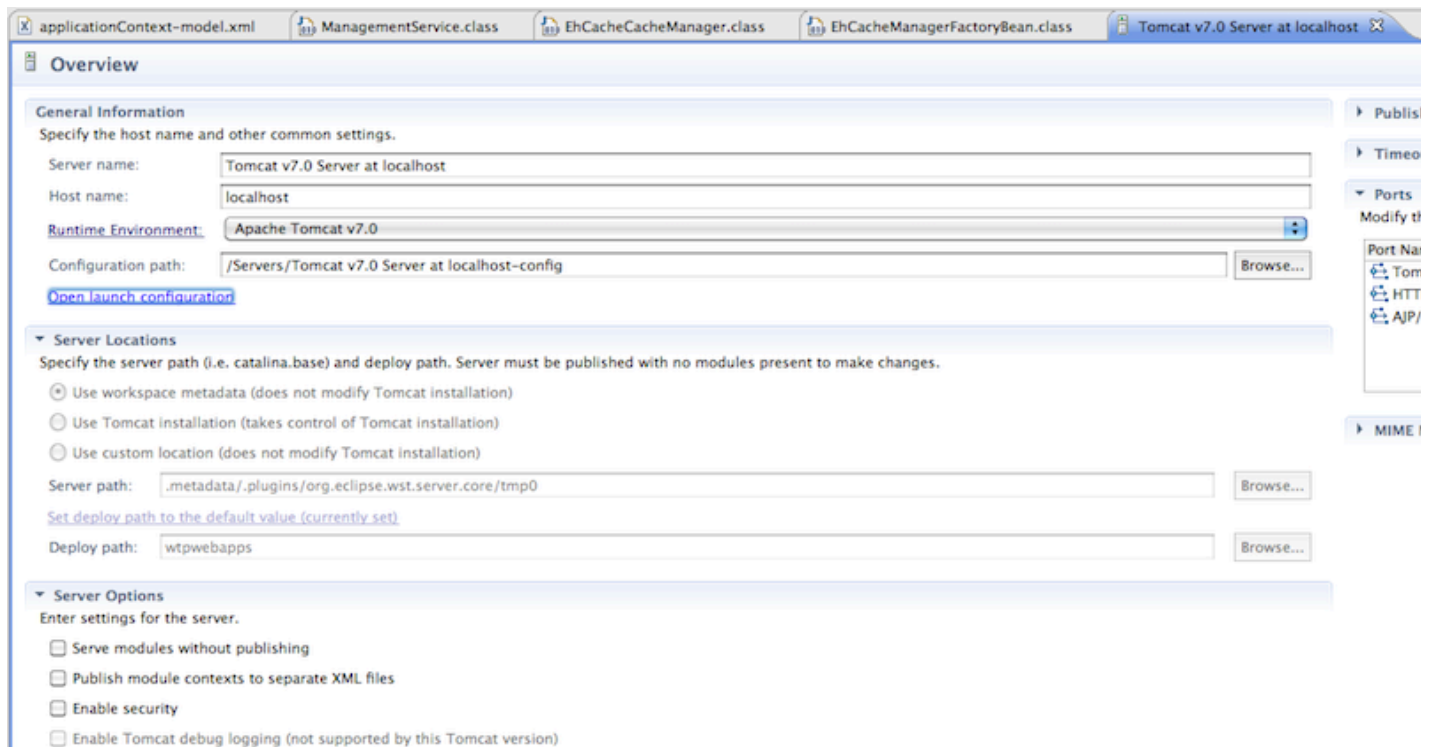
```

Si os fijáis hemos añadido los 2 últimos beans. El primero de ellos **mbeanServer** es donde vamos a tener registrados todos los **MBeans** y es el encargado de gestionar e interactuar con ellos. Al segundo bean, al que podemos llamar **gestor**, si os fijáis le estamos pasando como parámetro en el constructor el **mbeanServer** ya que el **gestor** es notificado cuando se crea o se destruye cualquier caché y actualiza el **mbeanServer** en consecuencia. El encargado de notificar dichas modificaciones en la caché sería el otro bean que se le está pasando por constructor que es **ehcache**, que si os fijáis, es precisamente el bean que se encarga de gestionar las regiones de caché, y por lo tanto conocerá cuando se crean nuevas o se destruyen viejas. Si queréis saber más sobre los Mbeans os dejo esta [página de referencia](#).

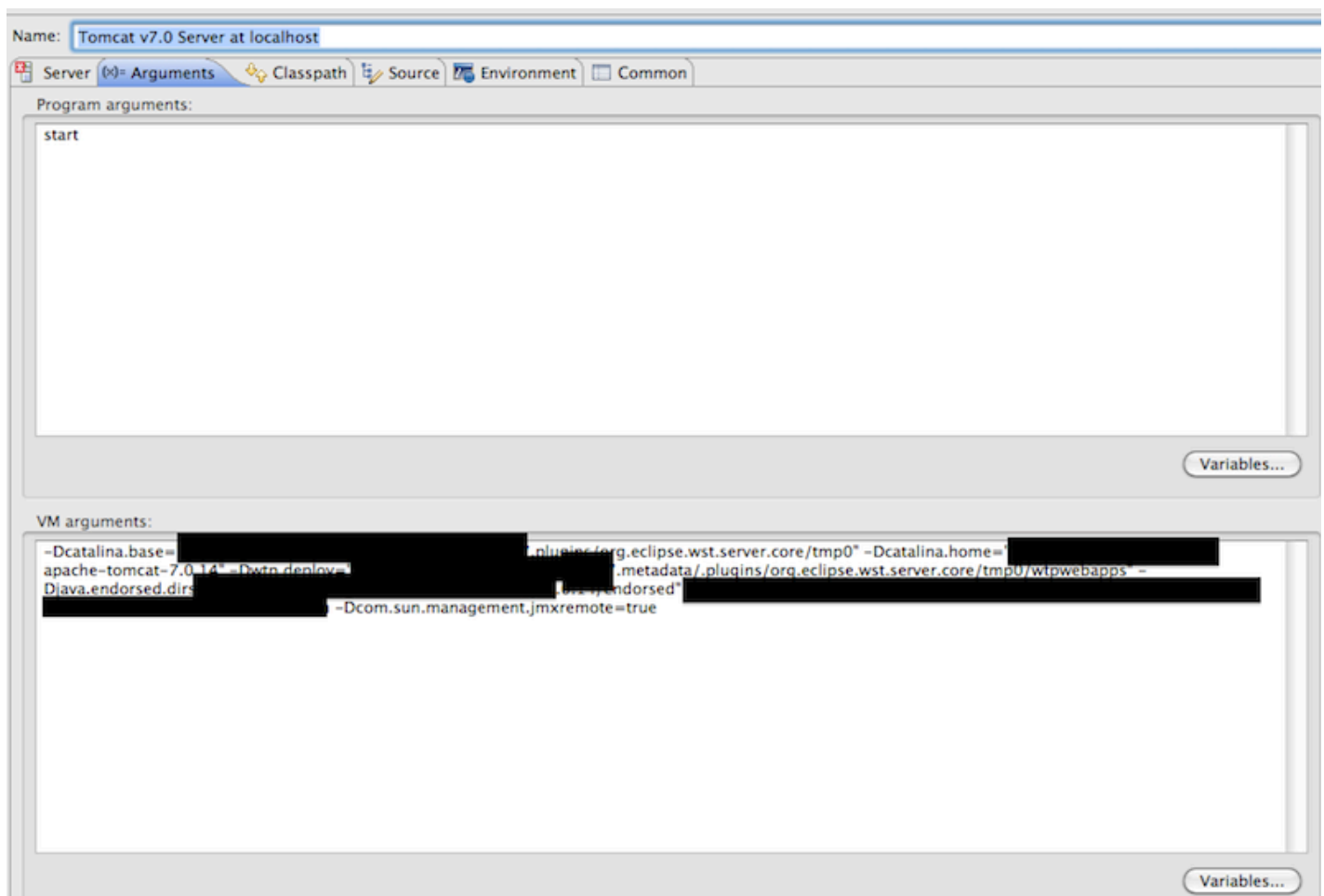
Por último comentar que **Jmx** es una tecnología cuya arquitectura se divide en tres capas o tres niveles. Del más bajo al más alto tendríamos: **Nivel de instrumentación** en donde se situarían los MBeans, **nivel de agente** donde se situaría el servidor de mbeans (mbeanServer) y un tercer y último nivel de gestión donde estaría el último bean que hemos llamado el **gestor**.

4. Monitorizar la aplicación con jConsole

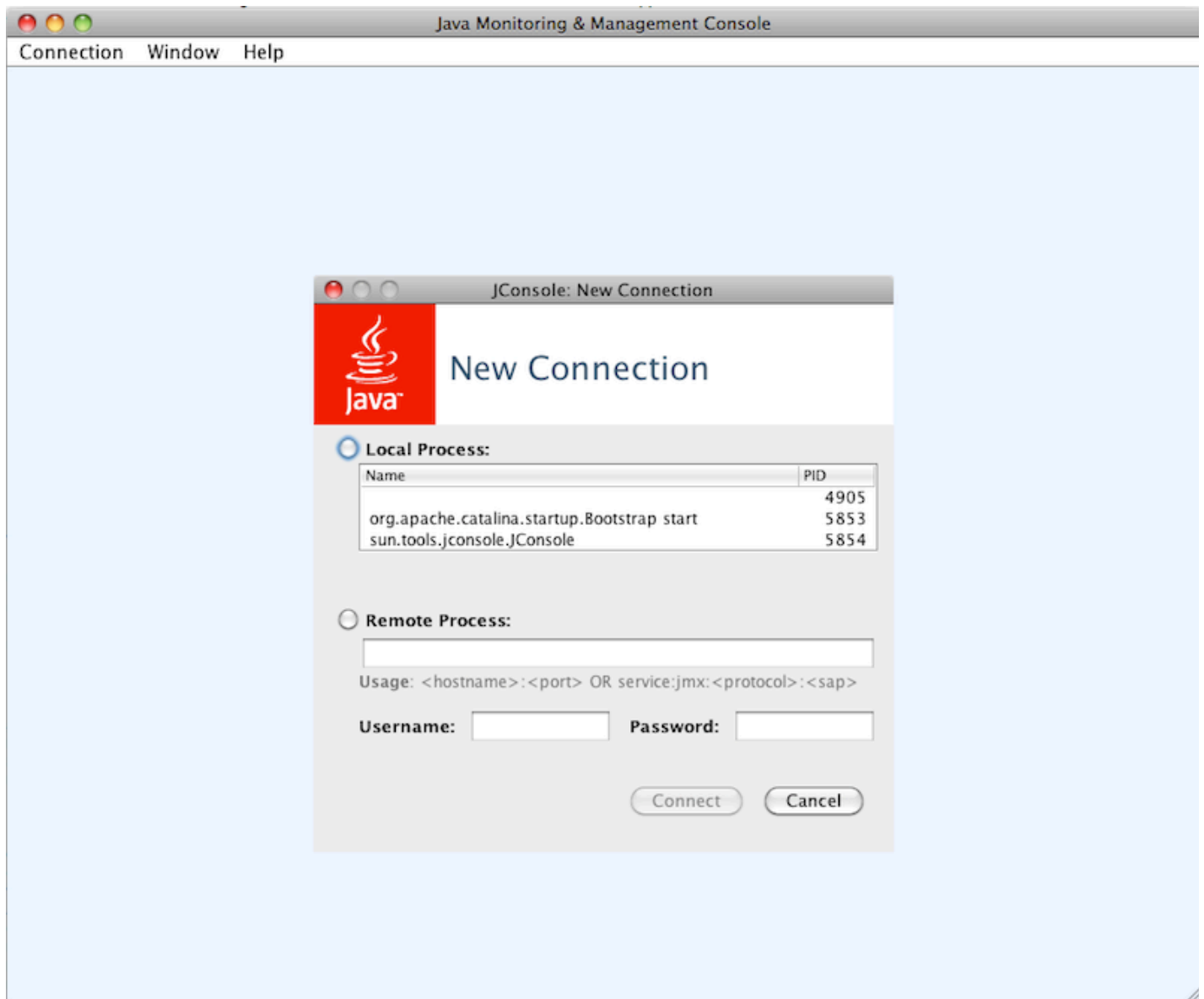
Para que una aplicación sea monitoriable por jConsole debemos de añadir a la máquina virtual el siguiente parámetro: **-Dcom.sun.management.jmxremote=true**. En mi caso la aplicación se despliega con Tomcat luego tenemos que editar el servidor para tal fin. Vamos a ver como se hace esto. Lo 1º paramos el servidor si es que lo tenemos arrancado. Hacemos click en la pestaña de **Servers** de Eclipse y doble click en el Apache Tomcat para poder acceder a la configuración. Nos saldrá una ventana como la siguiente:



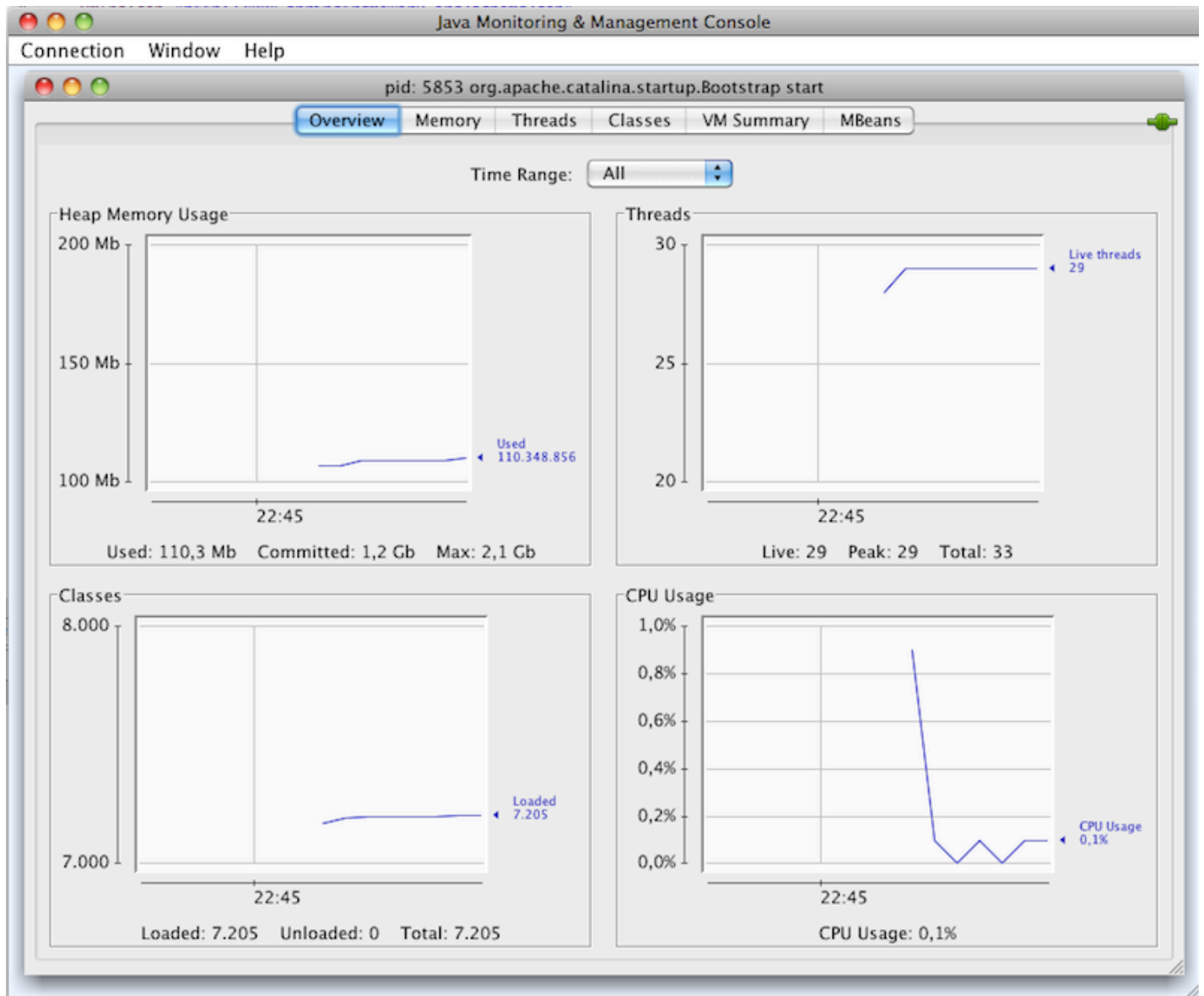
Tenemos que hacer click en **open launch configuration** y luego en la pestaña **Arguments** añadiremos `-Dcom.sun.management.jmxremote=true` tal y como muestra la siguiente foto



Ahora solo tenemos que ejecutar jConsole. jConsole es una aplicación gráfica con la que contamos desde la JDK 1.5 para monitorizar nuestras aplicaciones. Está en el directorio donde tenemos instalada la JDK. ¿Cómo? ¿Qué no te acuerdas donde tienes instalada la JDK? a mi también me ha pasado, si escribes **whereis java** en un terminal el gnomio que vive dentro de tu ordenador lo busca rápidamente y te lo sopla por pantalla. jConsole está dentro del directorio **bin**, así que una vez en el directorio **bin** ejecutamos `./jconsole`.



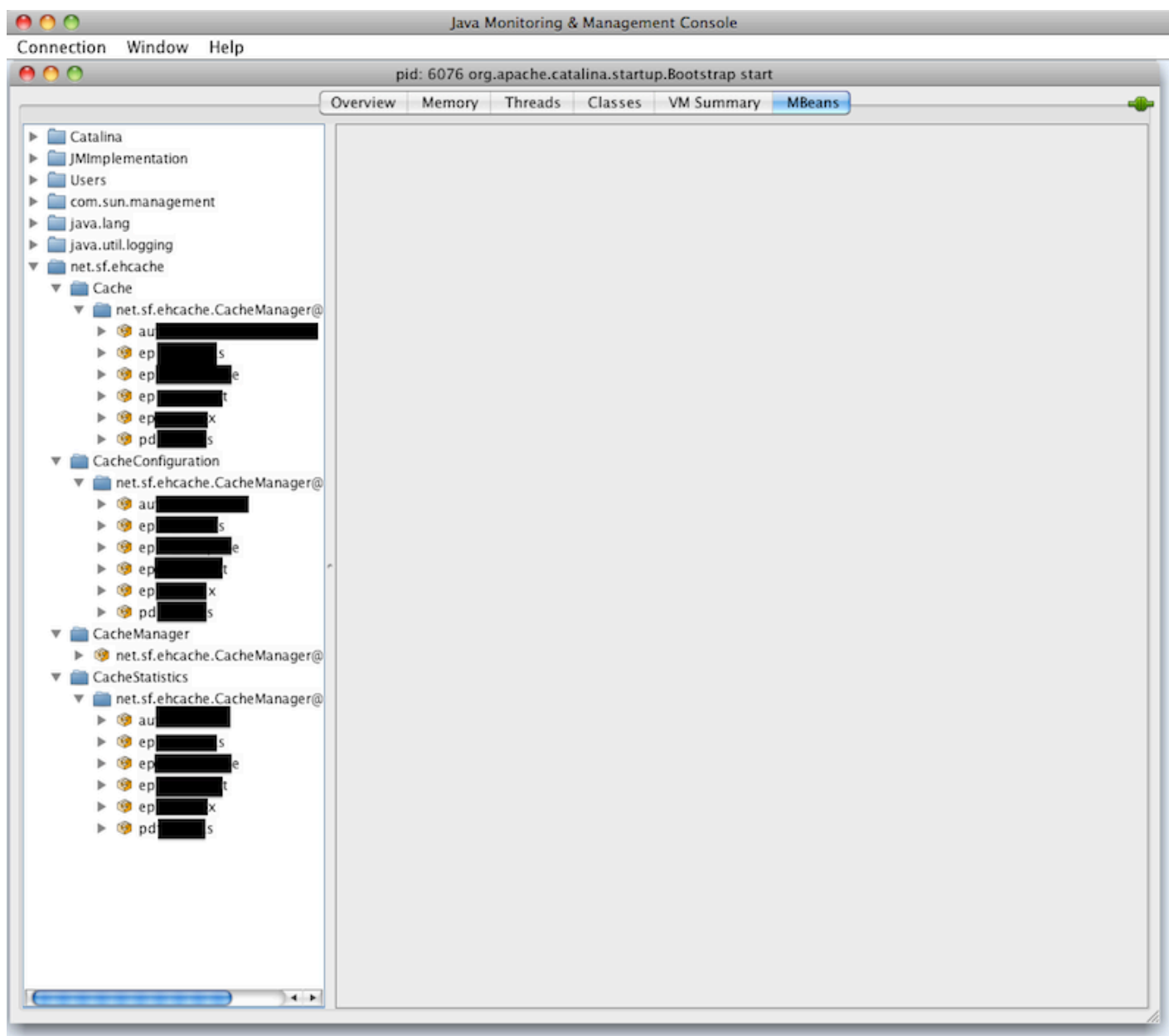
Como puedes ver, ahí tenemos nuestra aplicación arrancada, pinchamos en **Local process** y después en **org.apache.catalina.startup.Bootstrap start** y le damos a Connect. Deberíamos ver una pantalla como la siguiente



Se nos muestran 4 gráficas que monitorizan en tiempo real la memoria que está usando la aplicación, los hilos actuales, el uso de la CPU y las clases que tiene cargadas, datos que son críticos a la hora de medir el rendimiento de una aplicación. Os recomiendo cacharrear un poco por las ventanas e investigar por vuestra cuenta.

4.1 Gestión de la caché desde jConsole

En este tutorial nos vamos a centrar en la pestaña **MBeans**. Si lo hemos hecho todo bien al ir a esta pestaña y desplegar **net.sf.ehcache** deberíamos ver algo parecido a lo siguiente.



Tenemos que tener un MBean por cada región de caché que tengamos. En mi caso tengo definido en **ehcache.xml** las regiones de caché con los siguientes nombres: au, eps, epe, ept, epX y pds que como veis están todas en la imagen.

Vamos a ver ahora la información que tenemos disponible de cada una de ellas, por ejemplo, vamos a desplegar la opción de eps, dentro de **CacheStatistics**. Veremos que tenemos 2 opciones: **Attributes** y **Operations**. Si pinchamos en Attributes podemos ver entre otras cosas el número de elementos que tenemos cacheado de este tipo, que como acabamos de arrancar la aplicación es 0.

pid: 6403 org.apache.catalina.startup.Bootstrap start

Overview Memory Threads Classes VM Summary **MBeans**

Attribute values

Name	Value
AssociatedCacheName	ep[REDACTED]s
CacheHitPercentage	0.0
CacheHits	0
CacheMissPercentage	0.0
CacheMisses	0
DiskStoreObjectCount	0
InMemoryHitPercentage	0.0
InMemoryHits	0
InMemoryMisses	0
MemoryStoreObjectCount	0
ObjectCount	0
OffHeapHitPercentage	0.0
OffHeapHits	0
OffHeapMisses	0
OffHeapStoreObjectCount	0
OnDiskHitPercentage	0.0
OnDiskHits	0
OnDiskMisses	0
StatisticsAccuracy	1
StatisticsAccuracyDescription	Best Effort
WriterMaxQueueSize	0
WriterQueueLength	0

A continuación vamos a hacer una petición de elementos eps, para que se almacenen en la caché. Veamos como cambian ahora la información anterior.

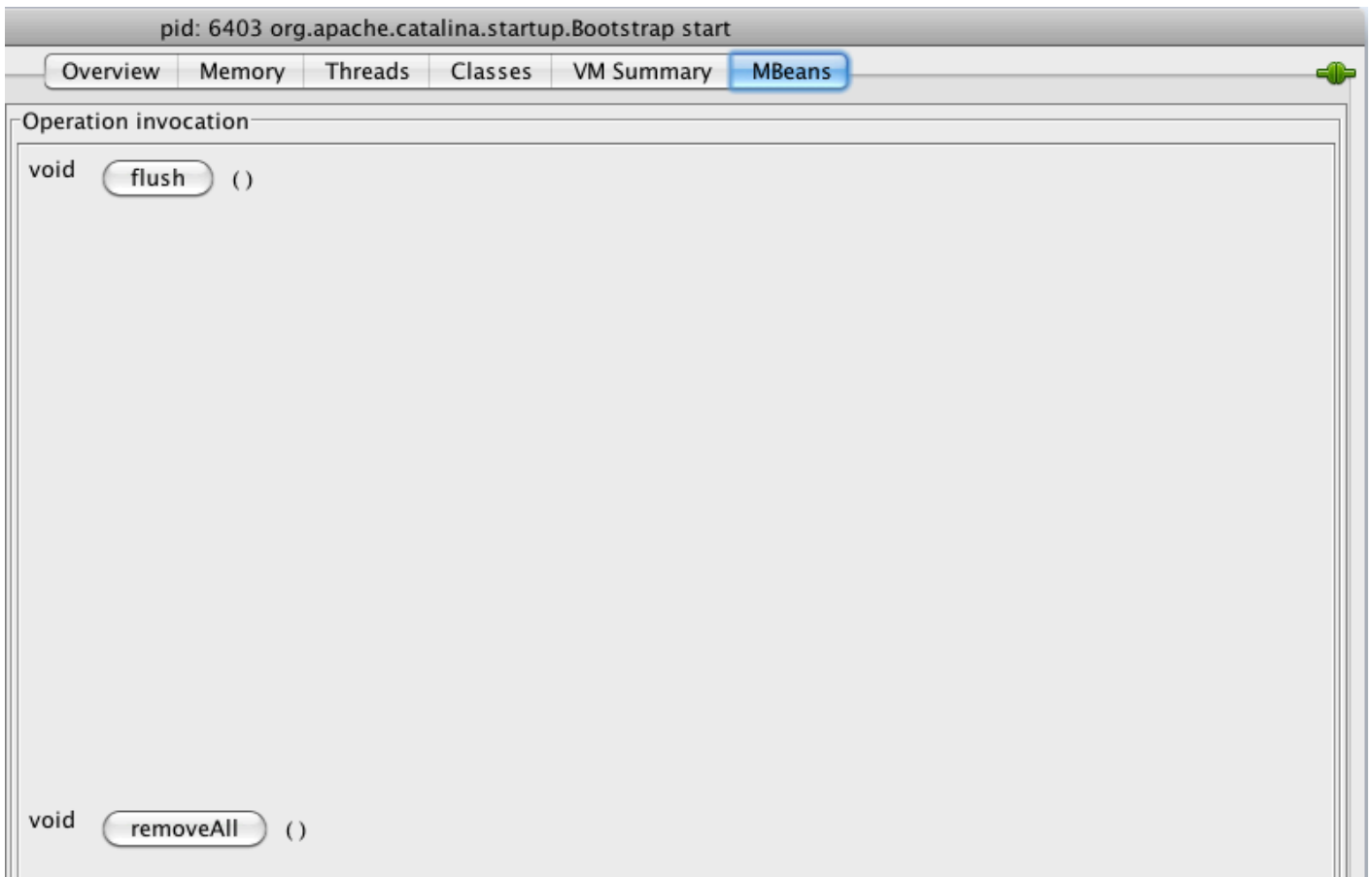
pid: 6403 org.apache.catalina.startup.Bootstrap start

Overview Memory Threads Classes VM Summary **MBeans**

Attribute values

Name	Value
AssociatedCacheName	ep[REDACTED]s
CacheHitPercentage	0.0
CacheHits	0
CacheMissPercentage	0.0
CacheMisses	0
DiskStoreObjectCount	310
InMemoryHitPercentage	0.0
InMemoryHits	0
InMemoryMisses	0
MemoryStoreObjectCount	1
ObjectCount	311
OffHeapHitPercentage	0.0
OffHeapHits	0
OffHeapMisses	0
OffHeapStoreObjectCount	0
OnDiskHitPercentage	0.0
OnDiskHits	0
OnDiskMisses	0
StatisticsAccuracy	1
StatisticsAccuracyDescription	Best Effort
WriterMaxQueueSize	0
WriterQueueLength	0

Una opción que nos ofrece jConsole y que nos puede resultar muy útil es el **limpiado de una región de caché** en concreto. Para hacerlo tenemos que pinchar esta vez en la pestaña de **Cache** y expandir la región de memoria que queramos limpiar, en mi caso eps. A continuación hacemos click en **Operations** y pinchamos en removeAll como se indica en la imagen



Si lo hemos hecho todo bien deberíamos tener de nuevo 0 elementos cacheados de eps y si estamos debugueando la herramienta al volver a solicitarlos podríamos ver como volvería a entrar en nuestro método **@Cacheable** (si es que estamos utilizando anotaciones) y volvería a realizar toda la operativa en vez de devolvernos el resultado desde caché sin pasar por el método.

5. Conclusiones

Monitorizar una aplicación puede ser interesante y nos puede ayudar a **evitar problemas de rendimiento**. Además si monitorizamos la caché podemos acceder a una serie de información valiosísima para ver como se comporta esta en tiempo de ejecución, que elementos son más solicitados, y además tendremos acceso a operaciones como al borrado de regiones de caché lo que me parece bastante interesante.

6. Información sobre el autor

Alberto Barranco Ramón es Ingeniero Técnico en Informática de Gestión y Graduado en Ingeniería del Software por la Universidad Politécnica de Madrid

Mail: abarranco@autentia.com.

Twitter: [@barrancoalberto](https://twitter.com/barrancoalberto)

Autentia Real Business Solutions S.L. - "Soporte a Desarrollo".

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» **Regístrate** y accede a esta y otras ventajas «

COMENTARIOS



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

IMPULSA

Impulsores

Comunidad

[¿Ayuda?](#)

sin clicks

0 personas han traído clicks a esta página

+ + + + + + + +

powered by [karmacracy](#)

Copyright 2003-2011 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

