



Infinispan

Martin Genčúr

JBoss Quality Engineering, Red Hat

Agenda

- Configuring Infinispan
- Monitoring via JMX
- Example application
- Tasks



Configuring Infinispan

■ CacheContainer level

- Configuration common to all caches under the same CacheContainer
- Defines communication inside the cluster
- Allows for setup of executors for eviction, asynchronous communication
- Fluent

Java API

GlobalConfigurationBuilder / GlobalConfiguration from package org.infinispan.configuration.global

Example

```
GlobalConfiguration glob = new GlobalConfigurationBuilder()
    .transport()
    .addProperty("configurationFile", "jgroups.xml")
    .build();

//also .nonClusteredDefault(), .clusteredDefault()
```



Configuring Infinispan

- Cache level

- Might be different for each cache under the same CacheManager

Java API

ConfigurationBuilder/Configuration from package org.infinispan.configuration.cache

Example

```
Configuration loc = new ConfigurationBuilder()
    //eviction(), expiration(), loaders(), locking(), ...
    .clustering().cacheMode(CacheMode.LOCAL)
    .expiration().lifespan(500).maxIdle(200).wakeUpInterval(10)
    .build();
```

```
EmbeddedCacheManager manager = new DefaultCacheManager(glob, loc, true);
```

```
BasicCache<String, Object> myCache = manager.getCache(CACHE_NAME);
```



XML Configuration

- since version 7.0, the schema is similar to WildFly's configuration

```
<infinispan>

  <jgroups>
    <stack-file name="udp" path="jgroups-udp.xml"/>
  </jgroups>

  <cache-container default-cache="default" statistics="true" shutdown-hook="REGISTER"
    listener-executor="listener"
    eviction-executor="eviction">
    <transport stack="udp" cluster="infinispan-cluster"
      node-name="Jalapeno" />
    <jmx domain="funky_domain" duplicate-domains="true" />
    <distributed-cache name="dist_cache"
      mode="SYNC" owners="3"
      segments="1000" l1-lifespan="600000">
      <state-transfer timeout="120000" />
    </distributed-cache>
  </cache-container>

</infinispan>
```

<https://github.com/infinispan/infinispan/blob/master/core/src/main/resources/schema/infinispan-config-7.1.xsd>



JMX (Java Management Extensions)

- De facto standard for management and administration of Java applications
- Needs to be enabled in Infinispan on both cache manager level and cache level
- Various JMX GUI - JConsole, VisualVM

JConsole: `$JAVA_HOME/bin/jconsole`

- MBeans

```
jboss.infinispan:type=Cache,name="___defaultcache(local)",manager="default",component=Cache"
```

- MBeans available locally/remotely

```
service:jmx:rmi:///jndi/rmi://hostname:1090/jmxrmi
```



JConsole

The screenshot shows the Java Monitoring & Management Console (JConsole) interface. The title bar reads "Java Monitoring & Management Console". The menu bar includes "Connection", "Window", and "Help". The address bar shows the path: "pid: 6505 jboss-modules.jar -mp /home/mgencur/containers/eap6/ER4/jboss-eap-6.0/modules -jaxpmodule javax.xml...".

The main window has several tabs: "Overview", "Memory", "Threads", "Classes", "VM Summary", and "MBeans". The "MBeans" tab is active. On the left, a tree view shows the hierarchy of MBeans under "org.infinispan.carmart.tx". The "Cache" MBean is expanded, showing "DefaultCacheManager" and "Cache". The "Cache" MBean is further expanded to show "Statistics", which is selected. Below "Statistics", a list of attributes is visible: "removeHits", "misses", "averageReadTime", "evictions", "removeMisses", "numberOfEntries", "elapsedTime", "hits", "timeSinceReset", "stores", "hitRatio", "readWriteRatio", "averageWriteTime", "statisticsEnabled", and "Operations".

The right pane displays the "MBeanInfo" for the selected MBean. It contains a table with the following data:

Name	Value
Info:	
ObjectName	org.infinispan.carmart.tx:type=Cache,name="carcache(local)",ma...
ClassName	org.infinispan.interceptors.CacheMgmtInterceptor
Description	General statistics such as timings, hit/miss ratio, etc.

Below the MBeanInfo, there is a "Descriptor" section with an empty table structure:

Name	Value
------	-------

The status bar at the bottom shows the path: "pid: 6505 jboss-modules.jar -mp /home/mgencur/containers..."



Demo

- <https://github.com/qa/pv243-2015-infinispan-seminar>
- Branches infinispan-00 – infinispan-04-complete-querying
 - Branch infinispan-03-prepare-querying is a starting point for Task 3
- Infinispan repository
<https://github.com/infinispan/infinispan>



Task 0

- **Get familiar with the web app**
 - After deploy the app is available at localhost:8080/infinispan-seminar
- How to make the cache accessible from the application?
- How to store a list of cars so that we can list them on a web page (safely)?
- How to get statistics like a number of entries in the cache?



Task 1 - Configuration

- Configure the cache according to the following requirements:
 - Cache parameters available via JMX (need to activate them)
 - One-node cluster (local mode)
 - Max. 4 entries in the cache
 - When more than 4 entries stored in the cache, the entry which was not used for longest time will be removed from the cache and stored in a local file system
- Verify via JMX that the cache always contains max. 4 entries (via JConsole)
- Hints
 - configuration located in CacheContainerProvider
 - eviction, passivation
 - <https://docs.jboss.org/infinispan/7.1/configdocs/>
 - http://infinispan.org/docs/7.1.x/user_guide/user_guide.html



Task 2 – Synchronous events

- Modify LocalStatisticsProvider so that
 - events about created, modified and removed entries are collected **synchronously**
 - the overall number of events is displayed on the initial screen of the application
- Hints
 - Listeners, AtomicInteger
 - <https://docs.jboss.org/infinispan/7.1/configdocs/>
 - http://infinispan.org/docs/7.1.x/user_guide/user_guide.html



Task 3 - Querying

- Switch to tag infinispans-03-prepare-querying
- Add search logic to CarManager
 - Create a query using QueryFactory, which will search for all cars matching one of these parameters: color, brand or country (might match just one of those parameters)
- Hints
 - QueryFactory
 - http://infinispans.org/docs/7.1.x/user_guide/user_guide.html#_infinispans_query_dsl



Task 3 - Querying

- Switch to tag infinispn-03-prepare-querying
- Add search logic to CarManager
 - Create a query using QueryFactory, which will search for all cars matching one of these parameters: color, brand or country (might match just one of those parameters)
- Hints
 - QueryFactory
 - http://infinispn.org/docs/7.1.x/user_guide/user_guide.html#_infinispn_s_query_dsl

