Infinispan

Martin Genčúr

JBoss Quality Engineering, Red Hat

Agenda

- Configuring Infinispan
- Monitoring via JMX
- Demo
- Tasks

Configuring Infinispan

- CacheManager level
  - Configuration common to all caches under the same CacheManager
  - 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()
```

CacheManager configuration – cont'd

XML

Tag <global>, podtagy <transport>, <globalJmxStatistics>, <xxxExecutor>

Example

```
<global>
      <globalJmxStatistics enabled="true"
                            jmxDomain="org.infinispan"
                             cacheManagerName="SampleCacheManager"/>

      <transport clusterName="infinispan-cluster"
                  nodeName="Node-A">

          <properties>
             <property name="configurationFile" value="jgroups.xml"/>
          </properties>

      </transport>
</global>
```

https://github.com/infinispan/infinispan/blob/master/core/src/main/resources/config-samples/sample.xml
https://github.com/infinispan/infinispan/blob/master/core/src/main/resources/schema/infinispan-config-5.2.xsd

Configuring Infinispan

- Cache level
  - Might be different for each cache under the same CacheManager

Java API

ConfigurationBuilder/Configuration z balíčku 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);
```

Cache configuration – cont'd

XML

Tags <default>, <namedCache>

Example

```
<namedCache name="distributedCache">

      <clustering mode="distribution">

         <sync/>

         <hash numOwners="3"
               rehashWait="120000"
                rehashRpcTimeout="600000"/>

         <l1 enabled="true"
              lifespan="600000"/>

      </clustering>

</namedCache>
```

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
```

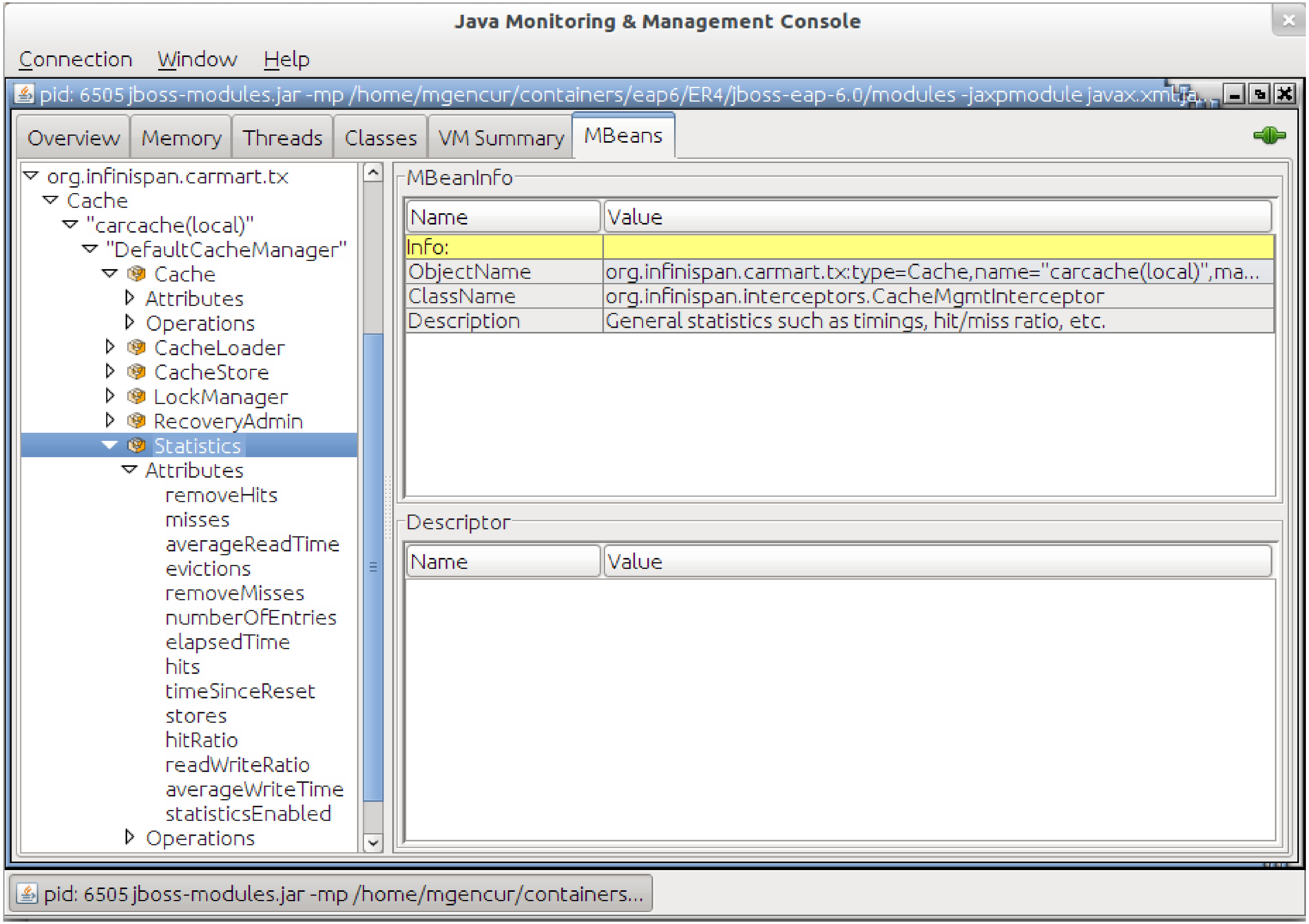
- MBeans

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

- MBeans available locally/remotely

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

JConsole



pid: 6505 jboss-modules.jar -mp /home/mgencur/containers/eap6/ER4/jboss-eap-6.0/modules -jaxpmodule javax.xml.ja...

```
▽ org.infinispan.carmart.tx
  ▽ Cache
    ▽ "carcache(local)"
      ▽ "DefaultCacheManager"
        ▽ Cache
          ▷ Attributes
          ▷ Operations
        ▷ CacheLoader
        ▷ CacheStore
        ▷ LockManager
        ▷ RecoveryAdmin
        ▼ Statistics
          ▽ Attributes
              removeHits
              misses
              averageReadTime
              evictions
              removeMisses
              numberOfEntries
              elapsedTime
              hits
              timeSinceReset
              stores
              hitRatio
              readWriteRatio
              averageWriteTime
              statisticsEnabled
          ▷ Operations
```

MBeanInfo

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.

Descriptor

Name	Value

pid: 6505 jboss-modules.jar -mp /home/mgencur/containers...

Demo

- git://github.com/qa/pv243.git

  (available through a web browser at https://github.com/qa/pv243)
- Directory lesson05-infinispan-updated
- Tags infinispan-01 – infinispan-05-complete
  - Tag infinispan-05-base is a starting point for Task 3
- Download

  http://download.jboss.org/jbossas/7.1/jboss-as-7.1.1.Final/jboss-as-7.1.1.Final.zip
- Infinispan repository

  https://github.com/infinispan/infinispan

Task 0

- **Get familiar with CarMart app**
- 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 the number of entries in the cache?

Task 1 - Configuration

- Configure the cache according to the following requirements:
  - Configuration located in JBossASCacheContainerProvider
  - 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 oldest entry will be removed from the cache and stored in a local file system
- Verify via JMX that the cache contains max. 4 entries at any given moment (JConsole)
- Hints
  - eviction, passivation
  - http://docs.jboss.org/infinispan/5.1/configdocs
  - https://docs.jboss.org/author/display/ISPN51/Home

  Note: When finished, the statistics should be properly displayed in the application

Task 2 – Asynchronous events

- Modify LocalStatisticsProvider so that it
  - **asynchronously** gets events about newly created, visited and removed entries
  - is able to display the overall number of these events on the initial screen of the application, the information is currently obtained explicitly (synchronously)

- Hints
  - Listeners, AtomicInteger
  - http://docs.jboss.org/infinispan/5.1/configdocs
  - https://docs.jboss.org/author/display/ISPN51/Home

Task 3 - Querying

- Add search logic to CarManager class
  - Create a query using QueryBuilder, which will search for all cars matching one of these parameters: color, brand or country (might match just one of those parameters)
  - *Advanced: Create a query using Apache Lucene's BooleanQuery, which will return only cars matching all of the parameters (color, brand and country), ignoring fields that were left empty*

- Hints
  - QueryBuilder, keyword
  - https://docs.jboss.org/author/display/ISPN/Querying+Infinispan
  - http://docs.jboss.org/hibernate/stable/search/reference/en-US/html_single