Ehcache Monitor

Ehcache Monitor Guide

Table of Contents

---

1. Introduction

This add-on tool for Ehcache is designed to provide enterprise-class monitoring and management capabilities for use in both development and production. It is intended to help understand and tune cache usage, detect errors, and provide an easy to use access point to integrate with production management systems. It provides administrative functionality such as the ability to forcefully remove items from caches.

The package contains a probe and a server. The probe installs with your existing Ehcache cache instance, and communicates to a central server. The server aggregates data from multiple probes. It can be accessed via a simple web UI, as well as a scriptable API. In this way, it is easy to integrate with common third party systems management tools (such as Hyperic, Nagios etc). The probe is designed to be compatible with all versions of Ehcache from 1.4.1 and requires JDK 1.5 or 1.6.

Enabling Statistics

The Ehcache Monitor displays statistics only for caches with statistics enabled. By default, statistics are disabled for caches in Ehcache 2.1 and higher. You can turn statistics on using the *statistics* attribute:

```
<cache name="myCache"
           maxElementsInMemory="1000"
           eternal="true"
           overflowToDisk="false"
           memoryStoreEvictionPolicy="LFU"
           statistics="true"
            />
```

---

2. Installation and Configuration

First download and extract the Ehcache Monitor package.

The package consists of a lib directory with the probe and monitor server jars, a bin directory with startup and shutdown scripts for the monitor server and an etc directory with an example monitor server configuration file.

To include the probe in your Ehcache application, you need to perform two steps:

STEP 1. Add the ehcache-probe-<version>.jar to your application classpath (or war file). Do this in the same way you added the core ehcache jar to your application.

STEP 2. Configure Ehcache to communicate with the probe by specifying the class name of the probe, the address (or hostname), and the port that the server will be running on. This is done by adding the following to ehcache.xml:

<cacheManagerPeerListenerFactory class="org.terracotta.ehcachedx.monitor.probe.ProbePeerListenerFactory"
properties="serverAddress=localhost, serverPort=9889" />

To start the server, run the startup script provided in bin/startup.sh (or bin/startup.bat on Microsoft Windows). These scripts will use the configuration stored in etc/ehcache-monitor.conf. The server port specified in the configuration file should match the serverPort specified in ehcache.xml.

Once the server is running, point your web browser to it to see the status and configuration of the cache(s) being monitored. For example, if the server is running on localhost on port 9889, use the following URL: http://localhost:9889/monitor

---

3. Using the Ehcache Monitor Web-based GUI

The web-based GUI is available by pointing your browser at http://<server-host-name>:<server-port>/monitor. For a default installation on the local machine, this would be http://localhost:9889/monitor

The GUI contains 4 tabs, described as follows:

Cache Managers

This tab shows aggregate statistics for the cache managers being monitored by probes connected to the monitor server. Double-clicking on any cache manager drills down to the detailed Statistics tab for that manager.

Statistics

This tab shows the statistics being gathered for each cache managed by the selected cache manager.

The Settings button permits you to add additional statistics fields to the display. Note: only displayed fields are collected and aggregated by the probe. Adding additional display fields will increase the processing required for probe and the server. The selected settings are stored in a preferences cookie in your browser.

Double-clicking on any cache drills down to the Contents tab for that cache.

Configuration

This tab shows the key configuration information for each cache managed by the selected cache manager.

Contents

This tab enables you to look inside the cache, search for elements via their keys and remove individual or groups of elements from the cache.

NOTE:

The GUI is designed to work in most browsers. Currently it has been tested in Firefox and Safari and is known not to work in Microsoft Internet Explorer. Wider browser support is planned for future release builds.

The GUI is set to refresh at the same frequency that the probes aggregate their statistic samples which is every 10 seconds by default. The progress bar at the bottom of the screen indicates the time until the next refresh.

---

4. Interpreting the Information Provided by Ehcache Monitor

Statistics

The following statistics are collected and aggregated by the probe and the monitor server. The first name is the name used to describe the statistic in the GUI. The second name (in parenthesis) is the name used by the API.

Disk Elements (diskCount) - The number of elements on disks

Eviction (evicted) - The number of elements evicted due to size limitations in the last sample period

Expiration (expired) - The number of elements evicted due to expiration in the last sample period

Hits (hit) - The number of cache hits in the last sample period

Disk Hits (hitDisk) - The number of cache hits served from disk in the last sample period

Mem Hits (hitMemory) - The number of cache hits served from memory in the last sample period

Hit Ratio (hitRatio) - The percentage of accesses in the last sample period that resulted in cache hits.

Mem Elements (memoryCount) - The number of elements in memory

Misses (miss) - The number of cache misses in the last sample period

Puts (put) - The number of cache puts in the last sample period

Removes (removed) - The number of elements removed in the last sample period

Total Elements (totalCount) - The total number of cache elements

Updates (updated) - The number of elements updated in the last sample period

NOTE: All statistics are counted and aggregated by the monitor over a preset sample period. The default sample period is 10 seconds. Thus, for example, a count of hits for a given cache, represents the number of hits recorded in the preceding 10 second sample period.

Configuration

Eternal - indicates whether elements are eternal. If eternal is set to true, timeouts are ignored and the element is never expired.

TTI - indicates the time to idle in seconds for an element before it expires. i.e. TTI is the maximum amount of time between accesses before an element expires. It is only used if the element is not eternal. A value of 0 means that an Element can idle for infinity.

TTL - indicates the time to live in seconds for an element before it expires. i.e. TTTL is the maximum time between creation time and when an element expires. Is only used if the element is not eternal. A value of 0 means that and Element can live for infinity.

Max Mem - indicates the maximum number of objects that will be created in memory.

Eviction - indicates the policy that will be enforced when the maximum elements in memory limit is reached. Possible policies include LRU (Least Recently Used), FIFO (First In First Out) and LFU (Less Frequently Used)

Overflow - indicates whether elements can overflow to disk when the memory store has reached the maximum elements in memory limit.

Max Disk - indicates the maximum number of objects that will be maintained in the DiskStore. Zero means unlimited.

Persistent - indicates whether the disk store persists between restarts of the Java Virtual Machine.

---

5. Using the Ehcache Monitor API

The list of functions supported by the API can be accessed by pointing your browser at http://<server-host-name>:<server-port>/monitor/list. For a default installation on the local machine, this would be http://localhost:9889/monitor/list

This list shows the functions available. In addition to the parameters shown, each function has a format parameter, which can either be xml or txt. The default format is txt.

For example, the getVersion function returns the software version of the monitor server. It can be called as follows:

http://localhost:9889/monitor/getVersion

or, to receive the results as XML:

http://localhost:9889/monitor/getVersion?format=xml

To query the data collected by the monitor server from scripts that can then be used to pass the data to enterprise system management frameworks, commands such as curl or wget can be used.

For example, on a Linux system, to query the list of probes that a local server on the default port is currently aware of, and return the data in XML format, the following command could be used
$ curl http://localhost:9889/monitor/listProbes?format=xml

Additional Notes

- HTTP status code are used to indicate the outcome of the API call. 200 implies success, 500 implies error (also returned when not all required parameters are provided). When the license expires the status code 402 is returned.
- When authentication is enabled via the server config file, 'user' and 'password' parameters should be added to each API call for this. HTTP status code is 401 is used to indicate that authentication is required.
- 'address' and 'port' parameters always have to be specified together, when not specified the method will be called on all the probes that are known to the server.