

Ehcache 2 缓存区配置详解

第 1 章 Ehcache 简介

EhCache 是一个开放源码的，基于标准的高速缓存系统。Ehcache 可以显著提高应用性能，降低数据库负载，简化应用扩展。Ehcache 健壮、功能齐全，也历经了众多应用考验，使得它成为使用最广泛的基于 Java 的缓存系统。

Ehcache 可以支持从进程内的一个或多个节点的部署方式到进程内与进程外混合、高达 TB 大小的高速缓存。

Ehcache 目前由 Terracotta 公司维护，使用 Apache 2 许可证。

Ehcache 截止目前最新版本为 2.6。支持多种方式缓存：

- Standalone 模式。嵌入应用进程内。单点，多节点间无沟通。
- Replication 模式。嵌入应用内部，通过 RMI 或 JGroup 或 JMS 进行节点同步。
- Cache Server 模式。作为独立缓存服务器，提供 REST 与 Webservice 接口供访问。
- Distributed Caching 模式。采用 Terracotta Server Array 实现高可用的分布式缓存。

Standalone 与 Replication 均是较传统的使用方式，很多场景下难以满足动态基础设施环境下应用弹性的要求。Cache Server 使得缓存服务可以容易的进行水平扩展，但是基于 REST 与 Webservice 的访问方式降低了缓存访问的效率，不太适合对缓存实时性要求较高的场景。Distributed Caching 模式提供了进程内与进程间缓存较理想的结合模式，支持水平扩展，高可用，对网络依赖程度较低，可以很好适应应用弹性伸缩，是动态基础设施条件下较理想的缓存模式。

第 2 章 Ehcache 2 缓存区配置

Ehcache 默认配置文件在 Ehcache 客户端的 classpath 根目录下，名为 ehcache.xml。典型的配置文件如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="ehcache.xsd"
         updateCheck="false" monitoring="autodetect"
         dynamicConfig="true" name="example">
  <defaultCache
    maxElementsInMemory="10000"
    eternal="false"
    timeToIdleSeconds="120"
    timeToLiveSeconds="120"
    overflowToDisk="true"
    diskSpoolBufferSizeMB="30"
```

```

        maxElementsOnDisk="10000000"
        diskPersistent="false"
        diskExpiryThreadIntervalSeconds="120"
        memoryStoreEvictionPolicy="LRU"/>
<cache name="colors"
        maxElementsInMemory="100"
        maxElementsOnDisk="0"
        eternal="false"
        timeToIdleSeconds="120"
        timeToLiveSeconds="0"
        memoryStoreEvictionPolicy="LFU">
    <terracotta/>
</cache>
<terracottaConfig url="localhost:9510"/>
</ehcache>

```

缓存区使用<cache/>标签进行配置，主要属性以及意义如下：

- name (必填属性)：缓存区名称，用以区别缓存区，必须唯一。
- maxEntriesLocalHeap (必填属性)：设置缓存在本地内存中最大缓存项数量（0 表示无限）。(等效于旧版本中的 maxElementsInMemory 属性)。
在实际使用中，在非分布式部署条件下，无限等效于 Integer.MAX_SIZE (2147483647)。在分布式部署条件下，缓存项数量由 Terracotta Server Array 资源上限决定。
- maxEntriesLocalDisk (必填属性)：设置保存在本地磁盘中的最大缓存项数量。默认为 0，表示数量不限。
- eternal (必填属性)：设置缓存项是否为永久的。如果设置为 true，缓存项的过期设置将被忽略，缓存项永不过期。
- overflowToOffHeap：此属性仅在使用 Ehcache 企业版时有效。设置为 true，缓存将使用非堆内存，非堆内存并非由 Java 管理，与 Java 垃圾回收无关。默认为 false。
- maxBytesLocalHeap：定义保存在 Java 虚拟机堆内存中的缓存项的最大字节数。定义时格式如下<number>k|K|m|M|g|G，其中 k|K 代表千字节，m|M 代表兆字节，g|G 代表吉字节。

举例：maxBytesLocalHeap="2g"将使用 2G 堆内存。

此属性与 maxEntriesLocalHeap 互斥，指定了此属性，则不能指定 maxEntriesLocalHeap。

如果设定过 CacheManager 的 maxBytesLocalHeap 属性，也不能使用此属性。

放入缓存的缓存项将使用 net.sf.ehcache.pool.sizeof.SizeOf 方法计算其字节数。

如果希望在计算字节数时忽略对象树种的某些属性，请参考 net.sf.ehcache.pool.sizeof.annotations.IgnoreSizeOf 注解。

- maxBytesLocalOffHeap：此属性仅在使用 Ehcache 企业版时有效。设置本缓存区使用的非堆内存的大小。

指定此属性，将默认设置 `overflowToOffHeap` 为 `true`。如果特别指定了 `overflowToOffHeap="false"`，将禁止使用非堆内存。

注意：当使用非堆内存时，推荐将 `maxEntriesLocalHeap` 设置为大于等于 100，否则将严重影响性能。也将看到一条警告日志。

非堆内存最小值为 128MB，没有上限。

- `maxBytesLocalDisk`：和 `maxBytesLocalHeap` 属性类似，不过指定的是存储在本地磁盘上的缓存项最大可使用的字节数。
- `timeToldleSeconds`：设置一个缓存项在过期前的闲置时间。即一个缓存项在其过期前，两次访问间隔的最大时间。仅在缓存项为非永久时有效。0 表示不限闲置时间，默认为 0。
- `timeToLiveSeconds`：设置一个缓存项在过期前的生存时间。即从缓存项创建到过期的最大时间。仅在缓存项为非永久时有效。0 表示不限生存时间，默认为 0。
- `diskExpiryThreadIntervalSeconds`：清理保存在磁盘上的过期缓存项目的线程的启动时间间隔，默认 120 秒。
- `diskSpoolBufferSizeMB`：磁盘缓冲区的大小。写入磁盘的内容将缓冲在此区域，使用异步的方式写入磁盘。默认 30MB，每一个缓存区使用独立的缓冲区，如果遇到 `OutOfMemory` 错误时，可以尝试减小这个值。改进磁盘写入性能时，尝试增加这个值。将日志级别开到 `Trace`，当 `DiskStore` 执行磁盘写入时，可以看到对应日志。
- `clearOnFlush`：当 `flush()` 方法调用时，`MemoryStore` 是否清空其内容，默认为 `true`，即清空。
- `statistics`：是否收集统计信息。如果需要监控缓存使用情况，应该打开这个选项。默认为关闭（统计会影响性能）。设置 `statistics="true"` 开启统计。
- `memoryStoreEvictionPolicy`：当缓存项达到 `maxEntriesLocalHeap` 限制时，剔除缓存项的策略。默认为 LRU (Least Recently Used)。其他的策略有：FIFO (First In First Out) 和 LFU (Less Frequently Used)。
- `copyOnRead`：当缓存项被读出时，是否返回一份它的拷贝（返回对象是缓存中对象的拷贝）。默认 `false`。
- `copyOnWrite`：当缓存项被写入时，是否写入一份它的拷贝（写入缓存的是写入对象的拷贝）。默认 `false`。

通过使用 `<persistence/>` 子元素，可以配置缓存区的持久化策略。`<persistence/>` 元素的主要属性如下：

`strategy`：配置缓存区持久化的类型。可选值如下：

- `localRestartable`：仅在使用 Ehcache 企业版时有效。启用 `RestartStore`，拷贝所有的缓存项（包含堆和非堆中的）到磁盘中，此选项提供了缓存快速重启能力以及对磁盘上缓存的容错能力。
- `localTempSwap`：当缓存容量达到上限时，将缓存对象（包含堆和非堆中的）交换到磁盘中。`"localTempSwap"` 并不持久化缓存内容。
- `none`：不持久化缓存内容。
- `distributed`：按照 `<terracotta>` 标签配置的持久化方式执行。非分布式部署时，此选项不可用。

`synchronousWrites`：此属性仅在 `strategy="localRestartable"` 时有意义。默认 `false`。设置为 `true`，缓存写入方法在缓存项成功写入磁盘前不会返回。

至此，缓存区主要配置介绍完毕。

第 3 章 旧版本向新版本迁移

本章总结了 Ehcache1.5 版本向 Ehcache2.6 版本迁移时的一些问题：

- `maxElementsInMemory` 属性被 `maxEntriesLocalHeap` 代替。
- 使用分布式缓存时，缓存区 `overflowToDis` 和 `diskPersistent` 都必须设置为 `"false"`（分布式模式下不支持这个属性），`maxElementsOnDisk` 属性建议设置为大于 0 的值（此时这个属性作用于 Terracotta Server Array 中这个缓存区，并非本地缓存）。