**入门**

使用场景：Hibernate二级缓存、DAO对象、安全认证、Web Page等等。甚至可以用作SOAP和RESTFul服务端缓存。

集群之间的复制策略：除了自身提供的Terracotta外，还支持RMI、JGroups、JMS。

**主要的类和方法**

**CacheManager**：推荐使用单例模式创建，CacheManager.create()。

**Cache**：都实现了Ehcache接口。

**MemoryStore**、**DiskStore**：两种存储方式。

**Element**：被cache的元素。包含key、value和访问次数。对于不可序列化的Object，不能持久化和在集群间复制。

**性能考量**

尽可能多的使用堆内存，防止GC(垃圾回收)，如BigMemory(the off-heap store)。

**使用模式**

**cache-aside**：先查询缓存，能命中，则直接返回查询结果；否则，查询SOR(如db)，然后将结果缓存。

**cache-as-sor**：将缓存当做SOR使用。

**read-through**：读取数据的时候和cache-aside类似，不过需要实现CacheEntryFactory接口，指定缓存miss时如何读取数据(将显式变成隐式)。

**write-through**：写数据时和cache-aside类似，需要实现CacheWriter接口并将cache配置为write-through或者write-behind。

**write-behind**：启用另一个线程往SOR中写数据。

**动态配置**

timeToLive：最长存活时间(秒)，不管有没有被使用。默认值为0，永不过期。

timeToIdle：最长闲置时间(秒)，多少秒没有被使用就过期。默认值为0，永不过期。

Local Sizing Attributes：maxEntriesLocalHeap，maxBytesLocalOffHeap等等。

memory-store eviction policy。

CacheEventListeners添加和删除。

**BigMemory**

要配合Terracotta使用。它的原理是开辟一个不被JVM GC的heap，因此叫做off-heap。它以字节的方式存储对象，因此需要序列化。在xml配置中，有两个选项用于打开这个特性：overflowToOffHeap，表示是否启用；maxBytesLocalOffHeap，表示最大空间。同时，需要开启JVM选项支持：java -XX:MaxDirectMemorySize=2G

Memory Store --- off-heap Store --- Disk Store

**生命周期**

Flush：从上层刷出到下层，比如从Memory刷到Disk；

Fault：从下层copy到上层。

Eviction：从集群中移除cache实体。

Expiration：缓存过期。过期了的缓存并不马上被flush或者evict，只是表示不可用。

Pinning：强制数据保存在某个层。

**大块加载(Bulk Loading)**

比如在系统启动的时候加载权限块。API方面和普通cache保持一致，但添加了一些bulk方面的方法。

page 136