JAVA缓存应用：OSCache使用指南

　OSCache是当前运用最广的缓存方案，JBoss,Hibernate,Spring等都对其有支持，下面简单介绍一下OSCache的配置和使用过程。

**安装过程**

·从http://www.opensymphony.com/oscache/download.html下载合适的OSCache版本，我下载的是oscache-2.0.2-full版本。

·解压缩下载的文件到指定目录，从解压缩目录取得oscache.jar文件放到/WEB-INF/lib或相应类库目录中，jar文件名可能含有版本号和该版本的发布日期信息等，如oscache-2.0.2-22Jan04.jar。

·如果你的jdk版本为1.3.x,建议在lib中加入Apache Common Lib 的commons-collections.jar包，如jdk是1.4以上则不必。

·从src或etc目录取得 oscache.properties 文件，放入src根目录或发布环境的/WEB-INF/classes 目录，如你需要建立磁盘缓存，须修改oscache.properties 中的cache.path信息。win类路径类似为c:\app\cache，unix类路径类似为/opt/myapp/cache。

·拷贝OSCache标签库文件oscache.tld到/WEB-INF/classes目录。

·现在你的应用目录类似如下：

$WEB\_APPLICATIONWEB-INFliboscache.jar

$WEB\_APPLICATIONWEB-INFclassesoscache.properties

$WEB\_APPLICATIONWEB-INFclassesoscache.tld

·将下列代码加入web.xml文件中

程序代码：

<taglib>

<taglib-uri>oscache</taglib-uri>

<taglib-location>/WEB-INF/classes/oscache.tld</taglib-location>

</taglib>

·为了便于调试日志输出，须加入commons-logging.jar和log4j-1.2.8.jar到当前类库路径中，在src目录加入下面两个日志输出配置文件：

log4j.properties 文件内容为：

log4j.rootLogger=DEBUG,stdout,file

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=[start]%d{yyyy/MM/dd/ HH:mm:ss}[DATE]%n%p[PRIORITY]%n%x[NDC]%n%t[THREAD] n%c[CATEGORY]%n%m[MESSAGE]%n%n

log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=oscache.log

log4j.appender.file.MaxFileSize=100KB

log4j.appender.file.MaxBackupIndex=5

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern=[start]%d{yyyy/MM/dd/ HH:mm:ss}[DATE]%n%p[PRIORITY]%n%x[NDC]%n%t[THREAD] n%c[CATEGORY]%n%m[MESSAGE]%n%n

log4j.logger.org.apache.commons=ERROR

log4j.logger.com.opensymphony.oscache.base=INFO

commons-logging.properties文件内容为：

org.apache.commons.logging.Log=org.apache.commons.logging.impl.Log4JCategoryLog

**oscache.properties文件配置向导**

cache.memory：值为true或false，默认为在内存中作缓存，如设置为false，那cache只能缓存到数据库或[硬盘](http://product.it168.com/list/b/0218_1.shtml)中，那cache还有什么意义。

cache.capacity：缓存元素个数。

cache.persistence.class：持久化缓存类，如此类打开，则必须设置cache.path信息。

cache.cluster相关：为集群设置信息。

例如：

cache.cluster.multicast.ip为广播IP地址

cache.cluster.properties为集群属性

**OSCache的基本用法**

cache1.jsp内容如下

<%@ page import="java.util.\*" %>

<%@ taglib uri="oscache" prefix="cache" %>

<html>

<body>

没有缓存的日期: <%= new Date() %><p>

<!--自动刷新-->

<cache:cache time="30">

每30秒刷新缓存一次的日期: <%= new Date() %>

</cache:cache>

<!--手动刷新-->

<cache:cache key="testcache">

手动刷新缓存的日期: <%= new Date() %> <p>

</cache:cache>

<a href="cache2.jsp">手动刷新</a>

</body>

</html>

cache2.jsp执行手动刷新页面如下

< %@ taglib uri="oscache" prefix="cache" % >

<html>

<body>

缓存已刷新...<p>

<cache:flush key="testcache" scope="application"/>

<a href="cache1.jsp">返回</a>

</body>

</html>

你也可以通过下面语句定义Cache的有效范围,如不定义scope,scope默认为Applcation程序代码：

<cache:cache time="30" scope="session">

...

</cache:cache>

**缓存过滤器CacheFilter**

你可以在web.xml中定义缓存过滤器，定义特定资源的缓存。

<filter>

<filter-name>CacheFilter</filter-name>

<filter-class>com.opensymphony.oscache.web.filter.CacheFilter</filter-class>

<init-param>

<param-name>time</param-name>

<param-value>60</param-value>

</init-param>

<init-param>

<param-name>scope</param-name>

<param-value>session</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>CacheFilter</filter-name>

<url-pattern>\*.jsp</url-pattern>

</filter-mapping>

上面定义将缓存所有.jsp页面，缓存刷新时间为60秒，缓存作用域为Session。

注意：CacheFilter只捕获Http头为200的页面请求，即只对无错误请求作缓存，而不对其他请求（如500,404,400）作缓存处理。