# Session共享

tomcat本身提供的HttpSession实现是通过StandardSession来实现(catalina.jar中)，StandardSession实现了序列化接口。该实现只能在同1个JVM访问，跨JVM不能访问。如果用session复制且后台服务器数量很多，则消耗服务器性能(session同步开销巨大)。

# memcached-session-manager

google提供了开源解决方案，将session同步到memcached

google开源网站:<http://code.google.com/p/memcached-session-manager/>

对上述网页的翻译可以参考: <http://wenku.baidu.com/view/bc741bc8da38376baf1fae1c.html>

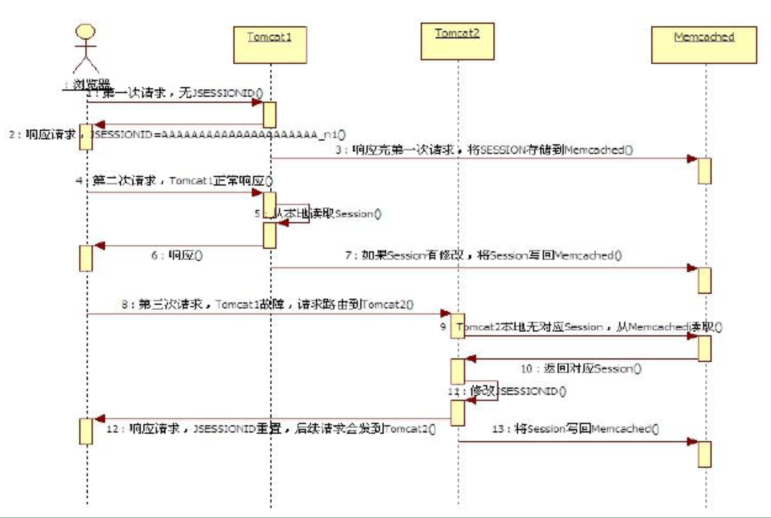
具体实施方案:<http://www.iteye.com/topic/1125301>

## 源码解读

它提供的jar包是通过MemcachedBackupSession来实现HttpSession，该类由于是继承StandardSession，当然也就实现了序列化接口，可以存入到memcached。

如果要通过request.getSession().setAttribute(“key”,对象)将对象存入到memcached，对象本身和属性均要实现序列化接口。

## 流程图



具体测试单点故障的时候，修改的JSESSIONID与tomcat对应的jsessionid是一致的。

## 实施步骤

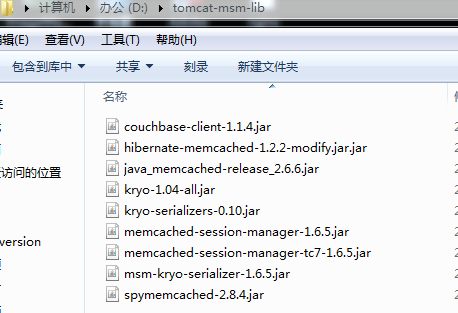
### Nginx配置负载均衡算法

此处实施的时候，采用的是粘性session(轮询算法也测试过，没有问题)

请参考《Nginx安装、配置文档.docx》

### Tomcat-lib拷贝

此处实施采用的是tomcat7，将相关jar包拷贝到tomcat/lib下，请见截图



### Tomcat-context.xml配置

<Manager className="de.javakaffee.web.msm.MemcachedBackupSessionManager"

memcachedNodes="n1:192.168.1.135:11211"

requestUriIgnorePattern=".\*\.(ico|png|gif|jpg|css|js)$"

transcoderFactoryClass="de.javakaffee.web.msm.serializer.kryo.KryoTranscoderFactory"

/>

相关的参数配置，可查阅相关资料。

## 注意事项

memcached-session-manager可能存在的问题

有网友反映,memcached使用LRU算法来处理缓存。但是毛病在于它不是全局的，而是基于slab。 这样你会发现在运行一段时间后，最早访问的那些用户的SESSION会莫名其妙的丢失，即使他们上一秒还有过操作