

# Memcached + Tomcat7（2 台）实现 session 共享

## 1、下载安装 Memcache

测试 memcache 是否连通

telnet localhost 11211（默认安装端口）

测试 memcache 是否工作

```
set abc 0 0 5
```

```
12345
```

```
get abc
```

## 2、下载需要的 jar 包，放在 tomcat7 lib 目录下



spymemcached-2.10.3.jar



xmlpull-1.1.3.1.jar



xpp3\_min-1.1.4c.jar



xstream-1.4.6.jar



memcached-session-manager-1.9.3.jar



memcached-session-manager-tc7-1.9.3.jar



msm-xstream-serializer-1.9.3.jar

### 3、配置 tomcat7

#### Tomcat7/Conf/Server.xml 配置添加

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="tomcat1">
```

#### omcat7/Conf/Context.xml 配置添加

```
<Manager className="de.javakaffee.web.msm.MemcachedBackupSessionManager"
    memcachedNodes="n1:localhost:11211"
    lockingMode="auto"
    sticky="false"
    requestUriIgnorePattern= ".*\.(png|gif|jpg|css|js)$"
    sessionBackupAsync= "false"
    sessionBackupTimeout= "100"
    transcoderFactoryClass="de.javakaffee.web.msm.serializer.xstream.XStreamTranscoder
Factory"
/>
```

### 4、Tomcat7 中放入测试包



ROOT.rar

### 5、启动 tomcat，进行测试访问

#### 黏性 Session

安装在 Tomcat 上的 MSM 使用本机内存保存 session，和 StandardManager 一样。另外，当一个请求结束时，session 会被送回 Memcached 进行备份。当下一次请求开始时，本地 Session 可用，直接服务，请求结束后，session 又被送回 Memcached 备份。

当集群中的一个 Tomcat 挂掉，下一次请求会被路由到其他 Tomcat 上。负责处理此请求的 Tomcat 并不清楚 Session 的信息。此时它会从 Memcached 查找该 Session，更新该 Session 并将其保存在本机内容。此次请求结束，session 被修改，送回 Memcached 备份。

**注意：**当使用多台 tomcat 时，一定要使用 non-sticky（非粘性）模式

#### Manager 标签属性说明

##### 1.className 必须

类名：de.javakaffee.web.msm.MemcachedBackupSessionManager

## 2.memcachedNodes 必须

memcached 节点：此属性应该包含所有运行的 memcached 节点或者 membase bucket 的 uri 地址,每一个 memcached 节点的属性定义格式为<id>:<host>:<port>,多个节点定义直接使用空格或者逗号分隔,形如:

memcachedNodes="n1:app01:11211,n2:app02:11211",如果只有单个的

memcached 节点,则<id>是可选项,只需配置<host>:<port>即可,形如:

memcachedNodes="localhost:11211".

如果我们配置的是 membase,那么从 1.6.0 版本开始,我们可以配置指定一个或者多个 membase bucket uris,形如: http://host1:8091/pools,http://host2:8091/pools。Bucket 名称和密码通过属性 username,password 来定义。membase buckets 连接需要遵循 memcached 协议,传输数据通过二进制流方式。

## 3.failoverNodes 可选项

故障转移节点:可选项,对非黏性 session 不可用,属性必须包含 memcached 节点集群的所有 ids。节点 id 之间用空格或者逗号分隔。

## 4.username 可选项

从 1.6.0 版开始使用,并且是可选的。用来进行 membase bucket 或者 SASL 验证,密码可以为空。

## 5.password 可选项

从 1.6.0 版开始使用,并且是可选的。用来进行 membase bucket 或者 SASL 验证,密码可以为空。

## 6.memcachedProtocol 可选项

定义 memcached 协议,默认使用 text 文本,出属性指明 memcached 使用的存储协议。只支持 text 或者 binary。

## 7.sticky 可选项

定义 session 方式为黏性或非黏性,默认为 true,多个 tomcat 时需使用非黏性

## 8.lockingMode 可选项

只有非黏性 session 才使用,默认值为 none

none: 从不对 session 进行锁定

all: session 将一直被锁定,知道请求结束

auto: 对于只读请求,session 将不会被锁定,如果是非只读请求,则 session 会被锁定

uriPattern:<regex>: 通过正则表达式的方式来对请求 uri 以及查询字符串进行匹配,只有匹配上的才会被锁定。

## 9.requestUriIgnorePattern 可选项

此属性是那些不能改备份 Session 的请求的正则表达式。如果像 css,javascript,图片等静态文件被同一个 Tomcat 和同一个应用 上下文来提供,这些请求也会通过 memcached-session-manager。但是这些请求在一个 http 会话中几乎没什么改变,所以他们没必要 触发 Session 备份。所以那些静态文件没必要触发 Session 备份,你就可以使用此属性定义。此属性必须符合 java regex 正则规范。

如: ".\*\.(png|gif|jpg|css|js)\$"

### **10.sessionBackupAsync** 可选项

指定 Session 是否应该被异步保存到 Memcached 中。 如果被设置为 true, backupThreadCount 设置起作用, 如果设置 false, 通过 sessionBackupTimeout 设置的过期时间起作用。

### **11.backupThreadCount** 可选项

用来异步保存 Session 的线程数, (如果 sessionBackupAsync="true")。默认值为 cup 的内核数。

### **12.sessionBackupTimeout** 可选项

设置备份一个 Session 所用的时间, 如果操作超过时间那么保存失败。此属性只在 sessionBackupAsync="false"是起作用。默认 100 毫秒

### **13.operationTimeout** 可选项

从 1.6.0 版开始使用, 默认值为 1000

### **14.sessionAttributeFilter** 可选项

此属性是用来控制 Session 中的那个属性值保存到 Memcached 中的正则表达式。正则表达式被用来匹配 Session 中属性名称。如

sessionAttributeFilter="^(userName|sessionHistory)\$" 指定了只有"userName"和"sessionHistory"属性保存到 Memcached 中。依赖于选择的序列化策略。

### **15.transcoderFactoryClass** 可选项

此属性值是创建序列化和反序列化 保存到 Memcached 中的 Session 的编码转换器的工厂类名。这个指定的类必须实现了 de.javakaffee.web.msm.TranscoderFactory 和提供一个无参的构造方法。例如其他的有效的实现在其他 packages/jars 中提供如: msm-kryo-serializer,msm-xstream-serializer 和 msm-javolution-serializer. 默认为 de.javakaffee.web.msm.JavaSerializationTranscoderFactory

### **16.copyCollectionsForSerialization** 可选项

默认值为 false。

### **17.customConverter** 可选项

自己定义特殊的类注册到 kryo 自定义转换器中, 实现序列化

### **18.enableStatistics** 可选项

用来指定是否进行统计。 默认值为 true。

### **19.enabled** 可选项

指定 Session 保存到 Memcached 中是否可用和是否可以通过 JMX 进行改变。只用于粘性 Session。 默认值为 true。