



论坛首页 → Java编程和Java企业应用版 → 企业应用 → J2EE 集群之failover小点子

全部 Hibernate Spring Struts iBATIS 企业应用 设计模式 DAO 领域模型 QQ Tomcat SOA JBoss Swing Java综合



最成熟稳定甘特图控件,支持Java和.Net

浏览 4085 次

主题: J2EE集群之failover小点子

该帖已经被评为良好帖

作者

正文

s929498110

等级:

性别:

文章: 122

积分: 140

来自: 开封

我现在离线

发表时间: 2011-08-21 最后修改: 前天

< >

猎头职位: [北京: ITeye网站诚聘Java高级搜索工程师](#)

J2EE 集群不太了解的人首先可以看看附件里面的《解开J2EE集群的神秘面纱》，讲的挺好的。

J2EE 的服务器集群主要的就是**负载均衡** 和**失败转移** 这些。

负载均衡这个话题都烂大街了，随处可以找到相关的帖子或博文，我也就不谈了。但是这些帖子中大部分都只谈了负载均衡，顶多再说一下 Tomcat 的 HttpSession 复制（失败转移的一种解决方案吧）。更有甚者，直接决定“集群中服务器节点宕机丢失的部分 HttpSession 不碍事”。。。。

我的感觉就是，

像 Tomcat 的 HttpSession 交叉复制，如果集群中服务器过多对性能的影响肯定非常之大。

像 JBoss、WebLogic 等的服务器链式 HttpSession 复制，如果一个服务器节点宕机，此节点的下一个服务器节点就得负责两个服务器的用户请求。是不是有点怕人奥。

至于“集群中某个节点宕机就宕机，HttpSession 丢失无所谓”这样的观点也是挺匪夷所思的，毕竟后台中2000个 HttpSession 同时丢失的后果不是那么容易承担的。

为了解决上面这些问题，我自己琢磨了一套方案，感觉挺不错的（不知道网上是不是有类似“轮子”，反正我是没有搜索到）。就发出来共享一下，嘿嘿

我想到的是**缓存服务器和Web服务器双向备份HttpSession**，这个。。。。咱语文挺烂的，表达的不好。

双向备份就是集群中多个Web服务器分散保存 HttpSession，同时在缓存服务器中也分散保存这些 HttpSession（每个 HttpSession 正常情况下只需要复制一次），至于怎么分散各位且听我慢慢说来。

我使用了三个缓存服务器（Memcached），三个Web服务器（Tomcat）进行思路实现和测试。

先说说思路吧：三个缓存服务器和三个Web服务器如下分布

Tomcat_ATomcat_BTomcat_C

Memcached_AMemcached_BMemcached_C

相关文章:

- 基于memcached的SNA实现
- 扩展Tomcat 6.x，使用memcached存放session信息
- 关于rails大容量网站部署的性能讨论

推荐群组: [xmemached用户组](#)

[更多相关推荐](#)

Tomcat_A 的 HttpSession 分布备份在Memcached_B 和 Memcached_C 中
Tomcat_B 的 HttpSession 分布备份在Memcached_A 和 Memcached_C 中
Tomcat_C 的 HttpSession 分布备份在Memcached_A 和 Memcached_B 中

我的构思就是这样：在服务器正常运行情况下，使用 Stick-Session 方式，同一个 HttpSession 由负载均衡器判断 HttpSession 归属交给同一个服务器执行，这样一来，正常情况下，服务器节点的运行就是完全独立的、就好像是单台服务器运行一样。HttpSession 复制备份也是隐式的，通过哈希码将这个 HttpSession 隐式备份在相应哈希码分布的 Memcached 上。

假设运行中 Tomcat_A 突然宕机了（我咔嚓关了Tomcat_A，嘿嘿），那么负载均衡器就会无法将请求分发给 Request 相对应的 Tomcat_A，负载均衡器便会把请求分发给其他的Tomcat_B 或 Tomcat_C，这两台服务器会接收到原属于绑定在 Tomcat_A 上的请求，他们没有相应的 HttpSession 啊，然后他们就主动从 Memcached 集群中寻找这个 HttpSession。如果找到的话，他们便主动把这个 HttpSession “收录”为自己的，即将 JSESSIONID 修改为自己的。这样一来，原属于 Tomcat_A 的 HttpSession 会随机分布给其他没有宕机的服务器节点。

这是服务器节点宕机的情况、
如果缓存器节点宕机了怎么办？

假设 Memcached_A 宕机，那么整个集群中保存在 Memcached_A 中的Session备份就丢失，那么 Tomcat_B 和 Tomcat_C 中的Session就没有了备份。如果这时这两台服务器再挂断的话，Session就真的丢失了（用户感情伤不起啊）。为了保证“整个集群中随时都存有同一个Session的备份”，Tomcat_B 和 Tomcat_C 应该为自身的Session负责。他们应该主动把自己的 Session 再备份一次（可以采用临时新建线程的方式，毕竟宕机不是频繁的么。属于特殊情况），再次备份到其他的可用 Memcached 节点中。

这样，整个程序中可以完全无视单台机器的突然宕机，一分钟内整个集群应该能把宕机服务器的 Session 再备份一次吧！

关于这个想法，网上有一个开源项目叫做 [memcached-session-manager](#)。我试过它，感觉不怎么好用。它只能说实现了部分我的想法吧，没有完全实现。测试中，仍然存在 HttpSession 丢失的情况。并且它有着令人发指的效率问题！

我测试了一下，普通的 JSP 页面处理只耗费 62.3ms（机子挫）。而添加上这个插件之后，普通的 JSP 页面处理需要消耗 2.18s !!! 这真是令人发指的效率！

靠人不如靠自己、我便自己实现了自己的思路。

使用的缓存程序就是 Memcached 2.6，我添加在附件里了。

我使用的 Memcached 客户端是 XMemcached（国产）、支持国产么，并且它的效率据某公司测试证明，比 SynMemcached 略高一点点。

下载地址：[XMemcached](#)

我就是通过替换了 Tomcat 原有的 StandardManager（MemcachedManager替换）、StandardSession（MemcachedSession替换）来实现的，然后加入了隐式的 HttpSession 备份操作。如果需要的话，还会从缓存程序中尝试获取 HttpSession。

我还通过在 MemcachedManager 中向 Context 中加入阀门（Valve）在每次请求之后再根据 MemcachedSession 的属性是否被修改来判断是否更新缓存（事务一致性么，整个请求处理过程是一个原子）。不涉及setAttribute、removeAttribute的操作是不会触发缓存更新的，并且 setAttribute 如果设置重复值也是不会引起缓存更新的。

在 XMemcached 中我添加了DisConnect 事件的监听器、如果某条连接断开（Memcached服务器节点宕机）的话，会触发Web服务器将 Session 再次更新入其他未断开的Memcached服务器节点中。

整个过程实现其实挺简单的、 jar 包我也放在附件中。

最后就说说我实现的这个“无名”插件的效率、 测试中我挺费解的， 添加了这个插件的 Tomcat 居然比正常的 Tomcat 处理JSP页面还快。使用它之后，处理一个正常同样JSP页面只需要 了 57.5ms ，嘿嘿，快了几毫秒（应该是测试数据误差）。

使用这个“无名”插件挺简单的、 就是修改 context.xml ，添加一个：

Xml代码 ☆

```
1. <Manager      className="net.sulin.mem.MemcachedManager"
2.              queueSize="300"
3.              addr="127.0.0.1:11211 127.0.0.1:11212 127.0.0.1:11213"
4.              poolSize="3"
5.              protocol="text"
6.              maxInactiveInterval="30"/>
7.
8. <!--
9.
10. className : 替换的Manager
11.
12. queueSize : 我采用了异步队列更新 HttpSession 缓存，队列用于存储需要更新的HttpSession对象，如果值太小的话。队列空间不够用
13.
14. addr : 就是 Memcached 的服务器地址和端口了，每个之间用空格隔开（XMemcached就是这样处理的，我不想费事）
15.
16. poolSize : XMemcached 客户端与 Memcached 服务器的连接池，据XMemcached 作者说即便是高并发也 30 之内吧，因为这是用NIO处理的，不需要太多连接。
17.
18. protocol : XMemcached 客户端与 Memcached 服务器之间的数据传输协议，可以是“text”也可以是“binary”， binary 的话它可以使用 XMemcached 的touch方法等， 但是需要 1.4 以上版本的XMemcached。 我没使用
19.
20. maxInactiveInterval : HttpSession过期时间， 分钟为单位
21.
22. -->
```

在测试中，我胡乱关闭Web服务器和Memcached服务器， 每次请求的 HttpSession 都没有遇见丢失的情况，嘿嘿，感觉挺不错的。性能上也是毫无影响。但是这是没有经受过任何生产考研的！😏

有想法的人可以试试，源码就在jar包中。如果有什么优化想法咱欢迎交流！

感觉俺写的不错的、顶一个啊。点个精华、良好神马的安慰一下俺嘛😊😊

=====

关于这个 failover 解决方案的设计思想：

我是感觉没必要为了 HttpSession failover再另外添置一个 Memcached 服务器集群。在现有的 Tomcat 集群上的每个节点上都绑定一个 Memcached 节点这样构成一个附加的 Memcached 服务器集群是不是更好？既不花钱、对性能基本上没什么影响（Memcached 效率很高的，有人测试了上亿数据量也仅仅占用了10%的CPU）。

关于 **attribute** 可能被修改而没有触发缓存更新的问题：

这个问题就是这样 `session.getAttribute("key") = newObject`; 这样的代码， 我没有深看所以还不太清楚此 `getAttribute` 方法是获取"引用"还是"复制"。

如果是“复制”的话就好说了，如果是引用的话。。。嘿嘿，我目前是没有好办法处理了，只有靠开发人员自己注意编程规范了（一般设置**session**属性的话都是调用**setAttribute**吧？）。 希望有哪位朋友想到了解决办法更俺交流一下。

mem-tomcat-ext-1.3.jar (16.9 KB) 下载次数: 202
Java集群与负载均衡.zip (828.8 KB) 下载次数: 572
memcached-1.2.6-win32-bin.zip (36 KB) 下载次数: 201

声明：ITeye 文章版权属于作者，受法律保护。没有作者书面许可不得转载。

推荐链接

- 3G培训就业月薪平均7K+，不3K就业不花一分钱！

[八百里数据恢复十年专业经验](#)
上海权威数据恢复中心，诚信、专业 快速，数据恢复就选八百里4006208010
www.cn800li.com

Google 提供的广告

- 见证又一个准百万富翁的诞生！
- 20-30万急聘多名天才Java/MTA软件工程师



返回顶楼

[主页](#)[资料](#)[短信](#)[留言](#)[关注](#)

com0606
等级: 初级会员



性别:
文章: 20
积分: 40
来自: 开封
[我现在离线](#)

发表时间：2011-08-21

不错，期待楼主更多的作品

返回顶楼

[主页](#)[资料](#)[短信](#)[留言](#)[关注](#)

回帖地址

0 0 请登录投票

newypei
等级: 初级会员

发表时间：2011-08-21

八错，学习了



性别:

文章: 12

积分: 30

来自: 开封

我现在离线

返回顶楼

主页 资料 短信 留言 关注

回帖地址

0 0 请登录后投票

freish

等级: 初级会员



性别:

文章: 619

积分: 40

来自: 少林寺厨房

我在看问题

发表时间: 2011-08-21

灾难恢复 这个词多好啊，失败转移。。。

返回顶楼

主页 资料 短信 留言 关注

回帖地址

0 0 请登录后投票

s929498110

等级:



性别:

文章: 122

积分: 140

来自: 开封

我现在离线

发表时间: 2011-08-22

freish 写道

灾难恢复 这个词多好啊，失败转移。。。

。。。。。。英语不咋地，见笑了。

理解意思就好

返回顶楼

主页 资料 短信 留言 关注

回帖地址

0 0 请登录后投票

Cindy_Lee

等级:



性别:

发表时间: 2011-08-22

期待有更完备的测试数据

文章: 183

积分: 130

来自: 武汉人在北京

我现在离线

返回顶楼

主页资料短信留言关注

回帖地址

00

请登录后投票

xchao

等级: 初级会员



性别:

文章: 15

积分: 30

来自: 北京

我现在离线

发表时间: 2011-08-22

不错的方案，值得一试！

返回顶楼

主页资料短信留言关注

回帖地址

00

请登录后投票

梦秋雨

等级: 初级会员



性别:

文章: 55

积分: 60

我现在离线

发表时间: 2011-08-22

如果只处理setAttribute/removeAttribute的话，如下情况是否漏掉：

request 1 :

```
session.setAttribute("key1",new String[3]);  
request 结束后同步
```

request 2 :

```
((String[])session.getAttribute("key1"))[0] = "newString";  
没有set/remove，于是不同步？但是Session内容实际上已经变了。
```

返回顶楼

主页资料短信留言关注

回帖地址

00

请登录后投票

buptchenliang

等级: 初级会员



性别:

文章: 1

积分: 30

来自: 北京

我现在离线

发表时间: 2011-08-22

真不错，值得借鉴

返回顶楼

主页

资料

短信

留言

关注

回帖地址

0

0

请登录投票

michael.softtech

等级: 初级会员

ITeye

性别:

文章: 59

积分: 30

来自: 上海

我现在离线

发表时间: 2011-08-22

想法不错，不过有些地方还需商榷。

就我个人来看，存在如下问题：

1. 就像楼主说的，tomcat多的时候session 复制存在问题。假设现在有1000台Tomcast Server,楼主觉得应该用多少memcached？如果少的话,每个memcached需要维护大量的session信息，而sessin本身是频繁变化的(通常情况下session过期时间不会太久),那么memcached的工作就十分沉重. 如果memcached多的话，从楼主的思路来看,因为负载层面可能是在fail-over的情况下面随机选择另一个tomcat,那么为了保证肯定能取到session,就需要每个memcached-1都备份一份,这个备份效率能接受吗？

2. 此外，多备份情况下，如何保证多备份数据在偶然情况下的session一致性? 比如某台memcached偶然网络出问题了，恰好在这时进行了session invalidate操作，于是就产生了一致性。这种不一致某些情况下会带来意想不到的后果。

所以我觉得这种思路研究研究还不错，如果真要使用的话，尚需验证。

发表回复

« 上一页

1

2

3

4

5

下一页 »

论坛首页 → [Java编程和Java企业应用版](#) → [企业应用](#)

跳转论坛: [Java](#)[Java](#)[Java](#)[Java](#)[Java](#)

北京: Coplogic诚聘 JAVA高级工程师 (15万-30万)

浙江: CSST研究院杭州分院诚聘Java开发工程师/部门

上海: 为为网诚聘JAVA软件开发工程师:

浙江: 网新恩普诚聘高级java软件开发工程师

上海: 为为网诚聘JAVA高级开发工程师

浙江: 阿里巴巴诚聘高级 Java 工程师

广告服务 | [ITeye黑板报](#) | [联系我们](#) | [友情链接](#)

© 2003-2011 ITeye.com. [[京ICP证110151号](#)]

http://www.iteye.com/topic/1113748[2011/8/25 11:59:12]