夏天的森林

好记性不如烂笔头

session机制详解以及session的相关应用

session是web开发里一个重要的概念，在大多数web应用里session都是被当做现成的东西，拿来就直接用，但是一些复杂的web应用里能拿来用的session已经满足不了实际的需求，当碰到这样的情况时候我们需要更加深入的理解session的机制，本文将梳理下session的相关知识，为设计可替代web容器自带的session机制打个基础。

1.1 session的概念

在计算机专业术语里：session是指一个终端用户与交互系统进行通信的时间间隔，通常指从注册入系统到注销系统之间所经过的时间以及如果需要的话，可能还有一定操作空间。

具体到web应用里的session，大家都做过web开发，这里我就先不提出web里session的定义，先和大伙讲下和session相关的技术背景。

早期的web应用或者说早期的网站都是一种处理静态资源的网站，功能主要是查看文档，看看图片，而现在的web应用和早期的差别已经很大，互联网的网站更准确的定义应该是互联网软件即网站就是软件，网站所代表的软件和早期软件的定义是不一样的，早期的软件都是在单机环境下运行，而互联网的普及让软件和网络技术融合在一起，这就要求网站所代表的软件应该要有一个对事务处理的记忆功能，事务处理的记忆功能就是我们常说的要有状态。而实现web应用技术的核心http协议是一个无状态的协议，http这种设计也许是历史遗留问题，也许无状态的http是最简单也是最有效的通讯方式，但是当网站成为软件后，状态的保持就是一个很重要的功能。

因此在web应用开发里就出现了保持http链接状态的技术：一个是cookie技术，另一种是session技术。

cookie技术是客户端的解决方案（当然随着html5的出现，比cookie更为强劲和安全的技术出现了，但是鉴于html5的普及度不够，就不做本文讨论的内容了），Cookie就是由服务器发给客户端的特殊信息，而这些信息以文本文件的方式存放在客户端，然后客户端每次向服务器发送请求的时候都会带上这些特殊的信息。让我们说得更具体一些：当用户使用浏览器访问一个支持Cookie的网站的时候，用户会提供包括用户名在内的个人信息并且提交至服务器；接着，服务器在向客户端回传相应的超文本的同时也会发回这些个人信息，当然这些信息并不是存放在HTTP响应体（Response Body）中的，而是存放于HTTP响应头（Response Header）；当客户端浏览器接收到来自服务器的响应之后，浏览器会将这些信息存放在一个统一的位置，对于Windows操作系统而言，我们可[illegible]nd Settings\[用户名]\Cookies目录中找到存储的Cookie；自此，客户[illegible]，都会把相应的Cookie再次发回至服务器。而这次，Cookie信息则存放在H[illegible]了。有了Cookie这样的技术实现，服务器在接收到来自客户端浏览器的请求之后，就能够通过分析存放于请求头的Cookie得到客户端特有的信息，从而动态生成与该客户端相对应的内容。通常，我们可以从很多网站的登录界面中看到“请记住我”这样的选项，如果你勾选了它之后再登录，那么在下一次访问该网站的时候就不需要进行重复而繁琐的登录动作了，而这个功能就是通过Cookie实现的。

session技术则是服务端的解决方案，它是通过服务器来保持状态的。由于Session这个词汇包含的语义很多，因此需要在这里明确一下 Session的含义。首先，我们通常都会把Session翻译成会话，因此我们可以把客户端浏览器与服务器之间一系列交互的动作称为一个 Session。从这个语义出发，我们会提到Session持续的时间，会提到在Session过程中进行了什么操作等等；其次，Session指的是服务器端为客户端所开辟的存储空间，在其中保存的信息就是用于保持状态。从这个语义出发，我们则会提到往Session中存放什么内容，如何根据键值从 Session中获取匹配的内容等。要使用Session，第一步当然是创建Session了。那么Session在何时创建呢？当然还是在服务器端程序运行的过程中创建的，不同语言实现的应用程序有不同创建Session的方法，而在Java中是通过调用HttpServletRequest的getSession方法（使用true作为参数）创建的。在创建了Session的同时，服务器会为该Session生成唯一的Session id，而这个Session id在随后的请求中会被用来重新获得已经创建的Session；在Session被创建之后，就可以调用Session相关的方法往Session中增加内容了，而这些内容只会保存在服务器中，发到客户端的只有Session id；当客户端再次发送请求的时候，会将这个Session id带上，服务器接受到请求之后就会依据Session id找到相应的Session，从而再次使用之。正式这样一个过程，用户的状态也就得以保持了。

由此我们可以得出，session是解决http协议无状态问题的服务端解决方案，它能让客户端和服务端一

公告

昵称：夏天的森林
园龄：5年4个月
荣誉：推荐博客
粉丝：2871
关注：18
+加关注

< 2016年10月 >

日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

搜索

找找看

谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

随笔分类(204)

android
css(2)
flex&flash
hadoop(11)
java(23)
javascript(42)
php(1)
python
ruby
数据库(9)
系统设计与架构(47)
项目经验(49)
心情(4)
云计算(16)

随笔档案(133)

系列交互动作变成一个完整的事务，能使网站变成一个真正意义上的软件。

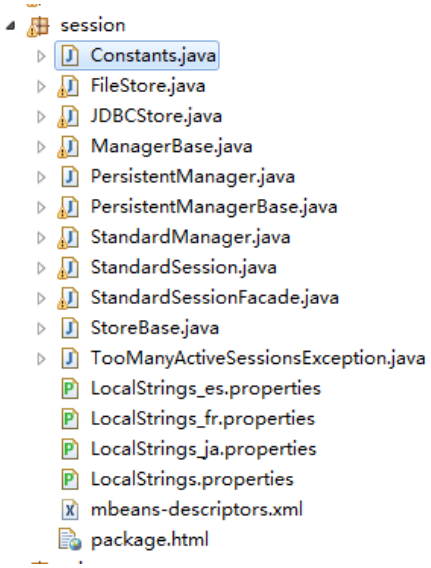
1.2 cookie与session的关系

cookie和session的方案虽然分别属于客户端和服务端，但是服务端的session的实现对客户端的cookie有依赖关系的，上面我讲到服务端执行session机制时候会生成session的id值，这个id值会发送给客户端，客户端每次请求都会把这个id值放到http请求的头部发送给服务端，而这个id值在客户端会保存下来，保存的容器就是cookie，因此当我们完全禁掉浏览器的cookie的时候，服务端的session也会不能正常使用（注意：有些资料说ASP解决这个问题，当浏览器的cookie被禁掉，服务端的session任然可以正常使用，ASP我没试验过，但是对于网络上很多用php和jsp编写的网站，我发现禁掉cookie，网站的session都无法正常的访问）

1.3 session实现的原理

java的web容器都实现了session机制，实现的逻辑思想都是一致的，但是具体方案可能会存在一定差异，这里我以tomcat容器为例，探讨下session实现的机制。

下图是tomcat源码里session实现：



实现包的路径是：org.apache.catalina.session，tomcat对外提供session调用的接口不在这个实现包里，对外接口是在包javax.servlet.http下的HttpSession，而实现包里的StandardSession是tomcat提供的标准实现，当然对外tomcat不希望用户直接操作StandardSession，而是提供了一个StandardSessionFacade类，tomcat容器里具体操作session的组件是servlet，而servlet操作session是通过StandardSessionFacade进行的，这样就可以防止程序员直接操作StandardSession所带来的安全问题。(StandardSessionFacade使用了设计模式里的Façade（外观）模式，外观模式能让不同逻辑层的组件进行解耦)。

实现类里有Manager的类是用来管理session的工具类，它负责创建和销毁session对象，其中ManagerBase是所有session管理工具类的基类，它是一个抽象类，所有具体实现session管理功能的类都要继承这个类，该类有一个受保护的方法，该方法就是创建sessionId值的方法（tomcat的session的id值生成的机制是一个随机数加时间加上jvm的id值，jvm的id值会根据服务器的硬件信息计算得来，因此不同jvm的id值都是唯一的），StandardManager类是tomcat容器里默认的session管理实现类，它会将session的信息存储到web容器所在服务器的内存里。PersistentManagerBase也是继承ManagerBase类，它是所有持久化存储session信息的基类，PersistentManager继承了PersistentManagerBase，但是这个类只是多了一个静态变量和一个getName方法，目前看来意义不大，对于持久化存储session，tomcat还提供了StoreBase的抽象类，它是所有持久化存储session的基类，另外tomcat还给出了文件存储FileStore和数据存储JDBCStore两个实现。

1.4 在实际运用中session所带来的问题

由上面所描述的session实现机制，我们会发现，为了弥补http协议的无状态的特点，服务端会占用一定的内存和cpu用来存储和处理session计算的开销，这也就是tomcat这个的web容器的并发连接那么低（tomcat官方文档里默认的连接数是200）原因之一。因此很多java语言编写的网站，在生产环境里web容器之前会加一个静态资源服务器，例如：apache服务器或nginx服务器，静态资源服务器没有解决http无状态问题的功能，因此部署静态资源的服务器也就不会让出内存或cpu计算资源专门去处理像session这样的功能，这些内存和cpu资源可以更有效的处理每个http请求，因此静态资源服务器的并发连接数更高，所以我们可以让那些没有状态保持要求的请求直接在静态服务器里处理，而要进行状态保持的请求则在java的web容器里进行处理，这样能更好的提升网站的效率。

当下的互联网网站为了提高网站安全性和并发量，服务端的部署的服务器的数量往往是大于或等于两

2016年6月 (3)
2016年5月 (4)
2016年2月 (1)
2016年1月 (1)
2015年12月 (1)
2015年11月 (4)
2015年3月 (6)
2015年2月 (12)
2015年1月 (4)
2014年12月 (3)
2014年11月 (7)
2014年9月 (4)
2014年8月 (1)
2014年7月 (1)
2014年1月 (2)
2013年12月 (1)
2013年11月 (1)
2013年10月 (1)
2013年9月 (4)
2013年8月 (4)
2013年6月 (7)
2013年5月 (3)
2013年1月 (1)
2012年8月 (3)
2012年7月 (1)
2012年6月 (2)
2012年5月 (5)
2012年4月 (4)
2012年3月 (1)
2012年1月 (2)
2011年12月 (9)
2011年11月 (3)
2011年10月 (12)
2011年9月 (15)

友情链接

【Aaron】
豪情

积分与排名

积分 - 333220
排名 - 375

最新评论

1. Re:java笔记：SpringSecurity应用（二）
简单易懂，正好是我需要的
--有一个年轻的老人
2. Re:重学hadoop技术
博主在武汉哪里？
--carrding
3. Re:HTML5笔记：跨域通讯、多线程、本地存储和多图片上传技术学习了。
--程序仲小仲
4. Re:HBase笔记：对HBase原理的简单理解
不错，博主有没有github
--一只尼玛
5. Re:用户行为分析笔记（二）：系统的整体架构
这套系统现在运行的怎么样，如何现在重新来设计这套系统，你会有

台，多台服务器对外提供的服务是等价的，但是不同的服务器上面肯定会有不同的web容器，由上面的讲述我们知道session的实现机制都是web容器里内部机制，这就导致一个web容器里所生成的session的id值是不同的，因此当一个请求到了A服务器，浏览器得到响应后，客户端存下的是A服务器上所生成的session的id，当在另一个请求分发到了B服务器，B服务器上的web容器是不能识别这个session的id值，更不会有这个sessionID所对应记录下来的信息，这个时候就需要两个不同web容器之间进行session的同步。Tomcat容器有一个官方的解决方案就是使用apache+tomcat+mod_jk方案，当一个web容器里session的信息发生变化后，该web容器会向另一个web容器进行广播，另一个web收到广播后将session信息同步到自己的容器里，这个过程是十分消耗系统资源，当访问量增加会严重影响到网站的效率和稳定性。

我现在所做的网站里有一个解决方案，当用户请求网站的时候会先将请求发送给硬件的负载均衡设备，该设备可以截获客户端发送过来的session的id值，然后我们根据这个id值找到产生这个session的服务器，将请求直接发送给这台服务器。这种解决方案看起来解决了session共享问题，其实结果是将集群系统最终变回了单点系统，如果处理请求的web容器挂掉了，那么用户的相关会话操作也就废掉了。此外，这种做法也干扰了负载均衡服务器的负载均衡的计算，让请求的分发并不是公平的。

一般大型互联公司的网站都是有一个个独立的频道所组成的，例如我们常用的百度，会有百度搜索，百度音乐，百度百科等等，我相信他们不会把这些不同频道都给一个开发团队完成，应该每个频道都是一个独立开发团队，因为每个频道的应用的都是独立的web应用，那么就存在一个跨站点的session同步的问题，跨站点的登录可以使用单点登录的（SSO）的解决方案，但是不管什么解决方案，跨站点的session共享任然是逃避不了的问题。

1.5 解决session相关问题的技术方案

由上所述，session一共有两个问题需要解决：

1) session的存储应该独立于web容器，也要独立于部署web容器的服务器；

2）如何进行高效的session同步。

在讲到解决这些问题之前，我们首先要考虑下session如何存储才是高效，是存在内存、文件还是数据库了？文件和数据库的存储方式都是将session的数据固化到硬盘上，操作硬盘的方式就是IO，IO操作的效率是远远低于操作内存的数据，因此文件和数据库存储方式是不可取的，所以将session数据存储到内存是最佳的选择。因此最好的解决方案就是使用分布式缓存技术，例如：memcached和redis，将session信息的存储独立出来也是解决session同步问题的方法。

Tomcat的session同步也有使用memcache的解决方案，大家可以参加下面的文章：

http://blog.sina.com.cn/s/blog_5376c71901017bgx.html

但是该方案只是解决了同步问题，session机制任然和web容器紧耦合，我们需要一个高效、可扩展的解决方案，那么我们就应该不是简单的把session独立出来存储而是设计一个完全独立的session机制，它既能给每个web应用提供session的功能又可以实现session同步，下面是一篇用zookeeper实现的分布式session方案：

http://www.open-open.com/lib/view/open1378556537303.html

好了写完了，今天只是简单剖析下session机制，以后有机会我拿出一套最好的独立session设计机制方案来的。

分类： java， 系统设计与架构， 项目经验， 云计算

夏天的森林
关注 - 18
粉丝 - 2871
荣誉：推荐博客
+加关注

« 上一篇：javascript笔记：javascript的关键所在---作用域链
» 下一篇：系统架构：前端监控系统草案（技术路线和用户行为分析相似）

posted @ 2013-10-29 23:22 夏天的森林 阅读(21971) 评论(8) 编辑 收藏

评论列表

#1楼 2013-10-30 10:24 守护晴天

关于楼主说的"（注意：有些资料说ASP解决这个问题，当浏览器的cookie被禁掉，服务端的session任然可以正常使

什么建议呢？

--xiezhengcai

阅读排行榜

1. Web前端学习笔记：Bootstrap框架(62825)
2. 大数据时代的技术hive：hive介绍(59732)
3. 为什么做java的web开发我们会使用struts2，springMVC和spring这样的框架?(40067)
4. 关于大型网站技术演进的思考（一）--存储的瓶颈（1）(35189)
5. hadoop 学习笔记：mapreduce框架详解(34558)
6. 分布式网站架构后续：zookeeper技术浅析(33932)
7. 我设计的网站的分布式架构(31446)
8. 架构设计：前后端分离之Web前端架构设计(30142)
9. java笔记：SpringSecurity应用（一）(25348)
10. javascript笔记:深入理解javascript的function(24840)

评论排行榜

1. 关于大型网站技术演进的思考（一）--存储的瓶颈（1）(97)
2. 我设计的网站的分布式架构(82)
3. 为什么做java的web开发我们会使用struts2，springMVC和spring这样的框架?(62)
4. web前端工程师在移动互联网时代里的地位问题(54)
5. Web前端学习笔记：Bootstrap框架(50)
6. 关于如何提高Web服务端并发效率的异步编程技术(43)
7. 与国内某知名互联网公司交流后的心得(42)
8. 谈谈javascript语法里一些难点问题（一）(40)
9. 把自己开发的网站前端开发框架和大家分享下(36)
10. 关于编写性能高效的javascript事件的技术(35)

推荐排行榜

1. 关于大型网站技术演进的思考（一）--存储的瓶颈（1）(208)
2. javascript技术难点（三）之this、new、apply和call详解(105)
3. 探真无阻塞加载javascript脚本技术，我们会发现很多意想不到的秘密(102)
4. 为什么做java的web开发我们会使用struts2，springMVC和spring这样的框架?(99)
5. 关于大型网站技术演进的思考（二）--存储的瓶颈（2）(84)
6. 我设计的网站的分布式架构(84)
7. Web前端学习笔记：Bootstrap

用，ASP我没试验过，但是对于网络上很多用php和jsp编写的网站，我发现禁掉cookie，网站的session都无法正常的访问）”，我想说说我的看法。

我是做ASP.NET开发的，对于SessionId如何在客户端存储，我有点模糊的印象，看到上面这句话以后我又问了一下度娘，得到了算是比较靠谱的答案。

首先我觉得Session在客户端不一定非要依赖Cookie，还有两个也是可以尝试，但是都有缺点，一个是请求的URL，一个是表单隐藏域。

URL存在的问题，一个是信息直接暴露出来，安全性有一定隐患（当然sessionid都是加密的），另一个是用户可能会人为修改这个值（但是Cookie也可能被用户伪造）；而表单隐藏域是随表单提交的，如果页面首次加载或者没有表单，实现起来可能就麻烦了。

asp.net就是用url处理禁用Cookie后保持会话状态的问题的，请看我找的别人的博客（原文参照web.config中SessionState的配置）：
“ASP.NET会话状态模块在Web.config文件中像下面这样配置：
<sessionState mode="InProc" cookieless="false" timeout="20" />，不用Cookie来传递会话ID。需要采用下面这样的URL：
http://my.website.com/(12mfju55vgblubjlwsi4dgjq)/education.aspx，圆括号中长长的字母、数字字符串就是会话ID。ASP.NET引擎从查询字符中提取会话ID，并将用户请求与特定会话联系起来。采取这种方式，不管Cookie还是隐藏表单字段都用不着了。”

最后说一句，以前只是会用Session和Cookie，但是对它们哥俩了解得不是很深入，看了楼主的文章后，对它们的了解又深入了一层。感谢楼主分享！

支持(0) 反对(0)

#2楼 2013-10-30 12:32 noahsky

session id在客户端一般使用会话Cookie存储，应该在内存中；不会存在
[系统盘]:\Documents and Settings\[用户名]\Cookies目录下

支持(1) 反对(0)

#3楼 2013-10-30 12:34 codezyc

好文，谢谢楼主的分享。

支持(0) 反对(0)

#4楼 2014-06-28 16:32 Ryan_1715575332

我们公司用cookie来保持会话，其实现在很多主流互联网公司，基本也这么做，你可以试试禁用cookie之后，再访问就会发现不同。
当然一般这个cookie值不是由业务系统写入的，一般会有一个统一系统专门做这个，XXX通行证。

支持(0) 反对(0)

#5楼 2015-07-28 15:38 圣灵的充满

温故而知新

支持(0) 反对(0)

#6楼 2016-07-29 12:20 小丑鱼1

分析得不错， 温故知新。。。

支持(0) 反对(0)

#7楼 2016-08-05 14:16 杰西卡芳

深入地了解了session和cookie这块的知识，谢谢楼主分享

支持(0) 反对(0)

#8楼 2016-09-02 19:14 LeoBron

感谢楼主的分享, 期待后文.

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。

框架(74)
8. 关于编写性能高效的javascript事件的技术(68)
9. 关于大型网站技术演进的思考（三）--存储的瓶颈（3）(61)
10. web前端工程师在移动互联网时代里的地位问题(59)