

Analysis and Performance Evaluation of Web Server Extensible Architecture Implementing Cluster and Load Balance*

Bingwei Liu¹, Qunhui Wu², Junhui Du^{3#}

¹Karamay City Drainage Management Office, Karamay

²Shanghai Bunny Information Technology Co. Ltd., Shanghai

³School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Email: 1445335630@qq.com, #dujunhui110@sina.com

Received: Jul. 2nd, 2013; revised: Jul. 13th, 2013; accepted: Jul. 21st, 2013

Copyright © 2013 Bingwei Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract: Configuring Tomcat clusters based on Apache and Nginx respectively, and comparing the performance including TPS (Transaction Per Second) and ART (Average Response Time) under the same network environment, a more extensible structure combining the effectiveness and resource-saving of Nginx and the flexibility and popularity of Apache is designed. Finally the analysis, design and the performance test of the proposed web server cluster are demonstrated superior on performance and practical operation.

Keywords: Cluster and Load Balance; Apache HTTP Server; Nginx HTTP Server; Tomcat Server; Reverse Proxy

Web 集群负载可扩展性架构分析及性能评价*

刘兵伟¹, 邬群辉², 杜军辉^{3#}

¹克拉玛依市排水管理处, 克拉玛依

²上海浦理信息科技有限公司, 上海

³上海理工大学光电信息与计算机工程学院, 上海

Email: 1445335630@qq.com, #dujunhui110@sina.com

收稿日期: 2013 年 7 月 2 日; 修回日期: 2013 年 7 月 13 日; 录用日期: 2013 年 7 月 21 日

摘要: 本文通过对基于 Apache 的 Tomcat 集群和基于 Nginx 的 Tomcat 集群的配置, 在相同网络环境下进行单位时间内处理请求数、平均响应时间等性能的测试, 提出一种更具扩展性的 Tomcat 集群, 既融合了 Nginx 高效节能特点, 又具备 Apache 的灵活性和普适性优点。通过分析、设计和性能测试, 验证了这种 Web 服务器集群在性能和实际操作中的优越性。

关键词: 集群负载; Apache 服务器; Nginx 服务器; Tomcat 服务器; 反向代理

1. 引言

Web 服务器系统集群负载技术为系统容量及其可扩展性提供了良好的途径, 它通过一定形式可以把

一组服务器组织起来, 对外提供更为强大的服务能力, 也可以在同一台服务器上部署多个相同系统进行集群, 充分挖掘服务器的潜力。随着 F5 BIG-IP、Radware、Citrix NetScaler 等硬件设备的大量涌现, 可以说实现负载均衡并不是难事, 但是这些硬件设备高

*基金项目: 国家自然科学基金项目(No. 61202376)。

#通讯作者。

昂的价格^[1]却往往让很多中小型网络公司或高校等事业单位不得不望而却步,因此负载均衡软件自然而然成为它们青睐的宠儿。开始于 20 世纪 90 年代的 Apache 与诞生于 2002 年的后起之秀 Nginx, 分别以其优秀的高效性、反向代理功能和灵活的负载均衡策略等备受业界的广泛关注和设计研究^[2,3]。然而相比于 Apache, Nginx 相关的可扩展性架构分析并不是很多。本文从架构分析、设计实现和性能评价等方面介绍 Apache、Nginx 与轻量级应用服务器 Tomcat 可扩展性负载均衡的实现。

2. Apache、Nginx 与 Tomcat 服务器

2.1. Apache 服务器

Apache HTTP Server (即 Apache HTTP 服务器, 简称 Apache 服务器)作为 Web 服务器软件,是 Apache Software Foundation (即 Apache 软件基金会)的一个开源项目, 伴随 Internet 的出现而出现, 目前依然是 Internet 中使用最多的 Web 服务器软件, 可以说 Apache 代表了目前网络服务器的先进技术。

瑞典网站监测服务公司 Pingdom 于 2012 年 5 月发布了最近网页软件使用报告。该调查结果显示, 在全球排名最高的 1 万名网站中, 开源软件的使用率高达 75%。Pingdom 的调查显示, 全球使用人数最多的网站服务器依然是 Apache HTTP Server, 占到 39.8%; 其次是俄罗斯公司开发的 Nginx (发音同 engine X), Nginx 由 Igor Sysoev 为俄罗斯最大的门户网站 Rambler.ru 而开发^[2]; 而作为非开源类服务软件, 微软的 IIS (Internet Information Services, 即互联网信息服务)市场占有率已降到 14%。

Apache HTTP 服务器作为现今最流行的 Web 服务器端软件之一, 以其功能完善、性能稳定、完全免费、源码完全开放等原因成为 Web 服务器的最佳选择。

2.2. Nginx 服务器

根据瑞典网站监测服务公司 Pingdom 发布的该网页软件使用报告显示, Nginx 已然成为全球使用人数仅次于 Apache HTTP Server 的第二大 Web 服务器软件, 并且其使用数量呈日益增长的趋势, 直逼 Apache 服务器市场。

Nginx 以其高性能、稳定性、丰富的功能设置、简单的配置和资源消耗低等优点而闻名世界。人们使用 Nginx 服务器主要使用它的两大特点: 一是高并发访问, 二是代理服务器 Proxy, Nginx 可以迅速解析静态文件, 而对于动态语言实现的动态程序(如 Servlet、JSP 等)则传递到后台服务器(如 Tomcat 服务器等), 从而实现了动态和静态网页解析的分离^[3], 提高效率, 节省资源。

Nginx 能够很好地解决服务器的 C10K 问题(网络服务在处理数以万计的客户端连接时, 往往出现效率低下甚至完全瘫痪, 即 C10K 问题), 因为它的设计不像传统服务器使用线程处理请求, 取而代之的是使用了一个更加高级的机制——事件驱动机制, 而且是异步事件驱动结构^[2]。

Nginx Proxy 是 Nginx 的特色功能, 利用 Proxy 实现负载均衡, 具有以下特色: 1) 功能强大, 性能卓越, 运行稳定; 2) 配置简单灵活; 3) 能够自动剔除工作不正常的后端服务器; 4) 上传文件使用异步传输模式; 5) 支持多种分配策略, 可以分配权重, 分配方式灵活^[3]。

2.3. Tomcat 服务器

Apache Tomcat 服务器(简称 Tomcat 服务器), 由 Apache 软件基金会下属的 Jakarta 项目开发, 以其运行时占用的系统资源小、开源性、可扩展性、支持负载均衡与邮件服务等功能, 而且不断的改进和完善更新, 使得很多感兴趣的程序员都可以对其进行更改或者往其中注入新功能, 因而深受广大程序员的喜欢。

Tomcat 以轻量级应用服务器身份, 在中小型系统或者并发性要求不是很高的场合下被普遍使用, 是开发和调试 JSP 程序的首选。Tomcat 部分是 Apache 服务器的扩展, 但它是可以独立运行的, 所以当运行 Tomcat 时, 它实际上是作为一个与 Apache 服务器独立的进程单独运行的。

Tomcat 和 IIS、Apache 等 Web 服务器一样, 具有处理 HTML 页面的功能, 另外 Tomcat 还是一个 Servlet 和 JSP 的容器, Tomcat 的默认模式就是一个独立的 Servlet 容器。但是, Tomcat 处理静态 HTML 的能力不如 Apache 服务器, 这也正是将 Apache HTTP Server 与 Tomcat 整合、集群的重要原因之一。

3. 基于 Apache 的 Tomcat 集群负载

将 Apache 服务器与 Tomcat 集群的原因在于：提升服务器对静态文件(如 html、图片、CSS 等)的处理性能^[4]；利用 Web 服务器做负载均衡以及容错；无缝地对应用程序进行升级^[5]。

速度快、稳定性高，对一个 Web 网站来说是非常重要的，而这种 Apache + “Tomcats” (Apache 与多个 Tomcat 进行集群实现负载均衡)的结构(见图 1)具有很高的性价比和良好的可靠性，不会因为某个或者某几个 Tomcat 宕机或程序升级导致用户访问不了，而能够完成这种功能的最好的 HTTP 服务器就是 Apache 服务器，因为它跟 Tomcat 的结合是最紧密和可靠的。Apache 负责静态资源的处理，而 Tomcat 完成 JSP 和 Servlet 等动态资源的处理，正如前面所说，原因在于 Tomcat 处理静态资源的能力不如 Apache 服务器。

在 JK (Tomcat Connector，即 Tomcat 连接器)^[6]出现之前，Apache 与 Tomcat 的整合通常采用 mod_jserv (Jserv 模块)作为 Apache 中的 WEB 服务器接口，来将客户端的请求从 Apache 重定向到 Tomcat 以建立 Tomcat 与 Apache 的联系。然而由于使用 mod_jserv 会使 Apache 的配置比较复杂，且不能用于其它 HTTP 服务器上，只能用在 Apache 上，同时它还不能很好地支持 SSL (Secure Socket Layer)，因此采用这种方式来整合 Apache 和 Tomcat 会增加配置文件的复杂性，从而降低了灵活性。

因此 JK 应运而生，JK 可以理解为 Tomcat 提供给 HTTP 服务器的插件，实现 Tomcat 与 Apache 等 HTTP 服务器之间的通信^[7]，其配置比 mod_jserv 要简单方便得多。除了使用 JK 方式实现 Apache 和 Tomcat 的整合外，还有 http_proxy 和 ajp_proxy 连接方式等^[8]。http_proxy 方式利用 Apache 自带的 mod_proxy 模块使用代理技术来连接 Tomcat。http_proxy 模式基于 HTTP 协议的代理，要求 Tomcat 必须提供 HTTP 服务，所以必须启用 Tomcat 的 HTTP Connector。ajp_proxy 方式其实跟 http_proxy 方式一样，均由 mod_proxy 提供连接功能。配置也是一样，只需要在访问时将 http:// 换成 ajp:// 即可，同时连接的是 Tomcat 的 AJP Connector 所在的端口。

相对于 JK 的连接方式，http_proxy 和 ajp_proxy

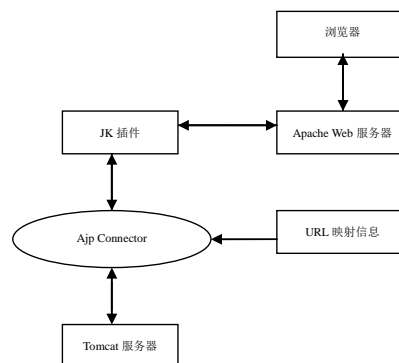


Figure 1. Apache and Tomcat integration principle
图 1. Apache 与 Tomcat 的集成原理

连接方式在配置上是比较简单的，灵活性方面也非常出色。但就稳定性而言则不如 JK 这样久经考验，因此，如果是应用于关键的互联网网站，最好还是采用 JK 的连接方式。

集群配置完成后，建立 test_cluster 测试项目，在项目的 web.xml 中添加 <istributable/>，distributable 元素用来告诉 Servlet 容器(即 Tomcat)，程序将部署在分布式 Web 容器中，从而实现 session 的共享。将动、静态文件分别部署，即将项目的动态文件部署到实现集群的 2 个 Tomcat 服务器的 webapps 目录中，而将静态文件部署到 Apache 的 htdocs 目录中，然后分别启动 Apache 和 2 个 Tomcat 服务器，打开 URL：http://localhost/test_cluster/test.jsp。

页面刷新，则依次会看到 tomcat1、tomcat2 出现，实现了负载均衡，多次刷新页面，看到 session 的 ID 号是相同的一个，说明 session 复制是成功的。通过往输入框中分别输入名称为 number，值为 1 的信息后，显示结果如图 2 所示，即为 session 中存储的属性。页面刷新，看到该属性也得到了复制。

以上的测试说明，集群中的 session 已经共享，每个集群对于同一访问均有相同的 session，而且 session 中存储的属性变量也得到了复制。当关闭 tomcat2，则只访问 tomcat1，说明节点关闭时运行正常。至此基于 Apache HTTP Server 的 Apache Tomcat 集群负载基本完成。

4. 基于 Nginx 的 Tomcat 集群负载

作为轻量级的 HTTP 服务器，Nginx 与 Tomcat 集群负载的配置相比于 Apache 要简单得多，只需修改 Nginx 的主配置文件 nginx.conf，在 http 节点里添

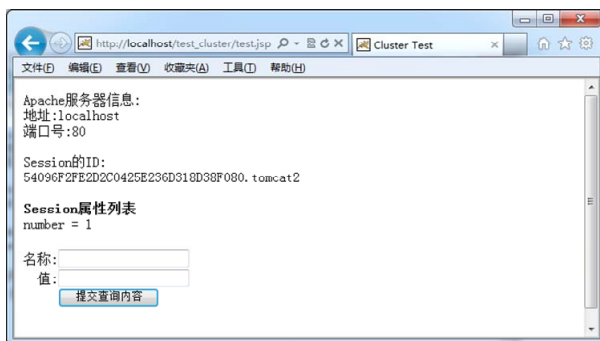


Figure 2. Test project explorer
图 2. 测试项目浏览效果

加:

```
upstream localhost {
```

#定义反向代理及后台服务器的负载均衡设备 IP 及状态, 其中 weight 为分配权重, weight 越大, 负载的权重就越大。除了 weight 参数外, 还有其它, 如 down 表示当前的 server 暂时不参与负载; backup 表示其它所有的非 backup 的机器 down 或者忙的时候, 请求 backup 机器, 所以这台机器压力会最轻。Nginx 还支持多组的负载均衡, 可以配置多个 upstream 来服务于不同的 Server。

```
server localhost:11080 weight=1;
server localhost:12080 weight=1;
}
```

```
location/{
```

```
index index.html index.htm;
```

if (!-e \$request_filename)#在缓存中找不到的文件就会通过下面的 proxy_pass 去后台服务器查找, 并且还会保留在本地的缓存中

```
{
proxy_pass http://localhost;
break;}
}
```

对网站的动态程序设置参数

```
location~\.(\.jsp|do|action|servlet) $ {
```

```
proxy_set_header Host $host;
proxy_set_header X-Forwarded-For $remote_addr;
proxy_pass http://localhost;}
```

#其中 proxy_pass http://localhost 用于指定反向代理的服务器池; proxy_set_header Host \$host 表示当后端 Web 服务器上也配置有多个虚拟主机时, 需要用该 Header 来区分反向代理的主机名; proxy_set_header

X-Forwarded-For\$remote_addr 表示如果后端 Web 服务器上的程序需要获取用户 IP, 就从该 Header 头获取。

#对网站的图片、Flash、JavaScript、CSS、静态 HTML 等静态文件进行 web 缓存

```
location~\.*(gif|jpg|jpeg|png|bmp|swf|js|css|html|shtml){
expires 30d;}
```

为实现动态文件与静态文件分开处理, 我们依然将项目的动态文件(JSP、Servlet 等)部署到这两个 Tomcat 服务器的 webapps 目录中, 而将静态文件部署到 Nginx 的 html 目录中, 以此来实现动静分离, 提高效率, 节省资源。

测试项目启动后和 Apache 下的 Tomcat 集群负载相类似, session 依然得到复制, 运行如图 3 所示。

至此 Window 下 Nginx + “Tomcats”的简单集群负载均衡配置结束。在 Nginx 下, 关于 Tomcat 的 Session 问题通常是采用 Memcached^[3,9], 或者采用 nginx_upstream_jvm_route 模块^[9,10](它是一个 Nginx 的扩展模块, 用来实现基于 Cookie 的 Session Sticky 的功能), 如此则涉及到 Nginx 的优化^[11]、Tomcat 的优化、Apache 的优化^[12]等, 本文只是实现基于 Apache、Nginx 的 Tomcat 可扩展性集群负载分析及其性能比较, 故此不涉及优化方面的研究。

5. 基于 Apache 与 Nginx 的 Tomcat 集群

鉴于 Nginx 的优秀性能, 特别是在处理静态文件和反向代理所表现出来的良好性, 使它结合功能完善、模块众多、可扩展性更强的 Apache, 来实现基于 Nginx + Apache 的 Tomcat 集群负载均衡也不失一条实际的解决方案。具体的方法还是使用 Nginx 的反向代理, 让 Apache 运行在后台, 而 Nginx 运行在前台。为了避免端口冲突, 需要将 Apache 端口号从原有的 80 修改为其它端口, 如 9090 即可。

6. 性能比较及分析

OpenWebLoad 是一款自由的开源系统, 在 Windows 和 Linux 操作系统上均可使用。它可以在服务器上抛出大量请求, 查看服务器处理请求的准确率, 使用数量可变的连接, 发送尽可能多的请求, 每一秒发送报告, 见图 4 操作显示界面。

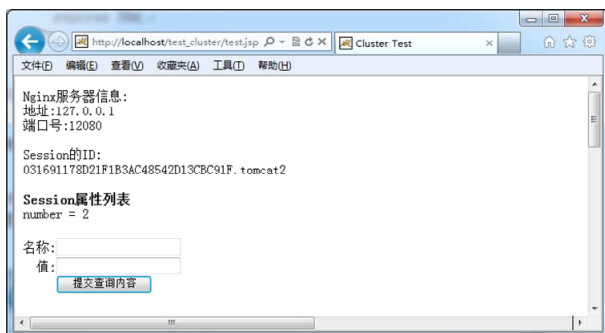


Figure 3. Implements replication of session attribute
图 3. Session 属性实现复制

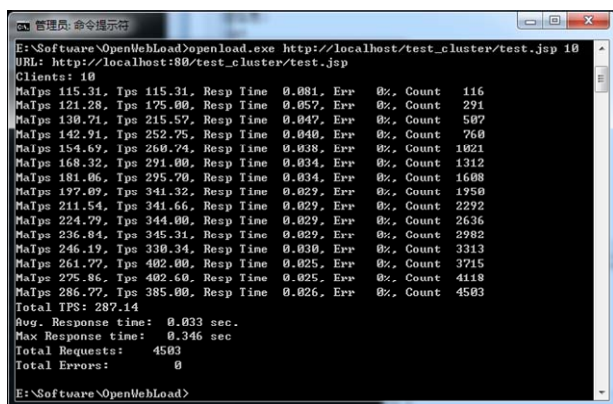


Figure 4. Performance testing using OpenWebLoad
图 4. 使用 OpenWebLoad 进行性能测试

从表 1 的数据看出,无论是基于 Apache 的 Tomcat 集群,还是基于 Nginx 的 Tomcat 集群,当并发连接数很少时,每秒处理的完整请求数(TPS)较低,然而响应时间较短;当并发连接数太多时,每秒处理的完整请求数(TPS)较高,但响应时间较长。对于两种集群来说,各自的并发连接数可以取得一个合适的中间值,使得每秒处理完整请求数和平均响应时间均比较合适。但是对比基于 Apache 的 Tomcat 集群和基于 Nginx 的 Tomcat 集群,我们可以看到随着并发连接数的增加,前者性能急剧下降,当并发连接数达到 400 时已经受到了瓶颈限制,TPS 值已经为 0,即用户已经无法访问了。而后者并没有受到此限制,经测试其并发连接数可以达到 1000 多。

从表 2 的数据可以看出,基于 Apache 与 Nginx 的 Tomcat 集群,随着并发连接数的增长,每秒处理的完整请求数(TPS)在短暂的增长之后下降,而平均响应时间持续增长。整体性能较之基于 Apache 的 Tomcat 集群和基于 Nginx 的 Tomcat 集群,优于前者,略逊与后者。

Table 1. Nginx + Tomcat with Apache + Tomcat clustering performance test

表 1. Nginx + Tomcat 与 Apache + Tomcat 集群性能测试

| 性能指标 | 测试 1 | | 测试 2 | |
|----------------|--------|--------|--------|--------|
| | Apache | Nginx | Apache | Nginx |
| 并发连接数 | 10 | | 100 | |
| 每秒处理完整请求数(TPS) | 287.14 | 283.94 | 281.69 | 339.30 |
| 平均响应时间 | 33 ms | 34 ms | 338 ms | 280 ms |

| 性能指标 | 测试 3 | | 测试 4 | |
|----------------|--------|--------|--------|--------|
| | Apache | Nginx | Apache | Nginx |
| 并发连接数 | 300 | | 400 | |
| 每秒处理完整请求数(TPS) | 124.25 | 316.18 | 0 | 301.67 |
| 平均响应时间 | 981 ms | 384 ms | 0 | 393 ms |

Table 2. Nginx + Apache Tomcat clustering to achieve performance test

表 2. Nginx + Apache 实现 Tomcat 集群性能测试

| 性能指标 | 测试 1 | 测试 2 | 测试 3 | 测试 4 |
|----------------|--------|--------|--------|--------|
| | 并发连接数 | 10 | 100 | 300 |
| 每秒处理完整请求数(TPS) | 282.72 | 315.38 | 293.65 | 290.00 |
| 平均响应时间 | 35 ms | 302 ms | 402 ms | 417 ms |

7. 结论

Nginx 对系统资源消耗少、允许高并发访问、配置简单,使它相对于 Apache 大有取代之势。从本文的实验性能对比来看,基于 Nginx 的 Tomcat 集群实现负载均衡的效果确实要远远好于基于 Apache 的 Tomcat 集群。但是 Apache 现如今依然是全球使用人数最多的网站服务器,凭借多年的发展,其功能很完善,模块众多,因而想一夜之间将 Apache 切换到 Nginx 不切合实际。所以我们可以使用 Nginx、Apache 的混合环境,一方面可以根据实际的应用需求来逐步使用 Nginx 替换 Apache,另一方面可以在现有基础上实现 Nginx + Apache 的集群负载,从性能测试结果看到这种基于 Nginx 与 Apache 的 Tomcat 可扩展性集群负载效果也很不错。

参考文献 (References)

- [1] 刘振宇. 利用 Nginx 实现网站负载均衡[J]. 中国管理信息化, 2012, 15(16): 96.
- [2] 张宴. 实战 Nginx: 取代 Apache 的高性能 Web 服务器[M]. 北

- 京: 电子工业出版社, 2010: 5-6.
- [3] 陶利军. 决战 Nginx: 高性能 Web 服务器部署与运维[M]. 北京: 清华大学出版社, 2012: 1,159-190.
- [4] 刘薇. 基于 J2EE 架构的分布式静态页面生成分析与设计[J]. 福建工程学院学报, 2010, 8(1): 90-93.
- [5] 孙仁鹏. Tomcat 与 Apache 集成的研究[J]. 电脑编程技巧与维护, 2011, 14: 6-8.
- [6] 刘尚旺, 何东健, 闫艳. Tomcat 与 IIS 或 Apache 服务器集成的应用研究[J]. 计算机工程与设计, 2009, 30(10): 2543-2546.
- [7] 郝海峰, 徐海峰, 叶俊. 基于 Windows 平台的 JSP Web 服务器架设[J]. 计算机与数字工程, 2007, 35(8): 68-70.
- [8] 王江伟. Apache 服务器配置与使用工作笔记[M]. 北京: 电子工业出版社, 2012: 1-5.
- [9] 吴翔毅. Nginx 在校园网络中的应用研究[J]. 福建电脑, 2009, (7): 103-104.
- [10] C. Nedelcu 著, 陶利军 译. 学习 Nginx HTTP Server(中文版)[M]. 北京: 清华大学出版社, 2012: 103-105.
- [11] X. N. Chi, B. C. Liu, Q. Niu and Q. X. Wu. Web load balance and cache optimization design based Nginx under high concurrency environment. IEEE, 2012: 1029-1032.
- [12] J. Zhao, K. S. Trivedi. Performance modeling of apache web server affected by aging. IEEE, 2011: 56-61.