

**罗文强**

**本文通过实际案例结合，使用最简短的文字、图形等资料，把Tomcat、Apache、Nginx、Memcached等技术应用起来，专注如何提高程序的并发性能，以及提高程序的可用性。**

Java分布式应用入门

[前言 i](#_Toc937108590)

[负载均衡 I](#_Toc1371681090)

[基于Apache的负载均衡 I](#_Toc692791680)

[Apache安装和启动 I](#_Toc1252223935)

[Tomcat安装和启动 III](#_Toc1131301844)

[负载均衡配置 III](#_Toc412468244)

[基于Nginx的负载均衡 V](#_Toc1979092472)

[Tomcat的Cluster 一](#_Toc28155995)

[Apache和Cluster的结合使用 ①](#_Toc1909140417)

[Nginx和Cluster的结合使用 ②](#_Toc989230083)

[使用Memcached管理Session - 1 -](#_Toc201017646)

# 前言

随着类似电商的高并发应用越来越多，一台服务器已经远远无法满足应用的需求。本文就从实际应用出发，讲解集群、分布式应用、负载均衡的各个知识点。

本文分为四种方案分别来讲解Tomcat的分布式应用，分别是：

1. 简单负载平衡：基于Apache的负载均衡、基于Nginx的负载均衡
2. 集群管理：自带的Cluster
3. 负载均衡和集群：结合Nginx和Tomcat Cluster的集群负载均衡
4. 集中管理session：使用memcached同步Tomcat的session

但是随着应用越来越复杂，在实际的实施过程中，会有一些问题不断涌现，所以本文也将不断更新。本文将被我放到GITHUB上面共享，基于GPL协议授权，欲获取更新版，请在GITHUB上面找到 luowenqiang 的用户，里面有一个 documents 的应用，专门用于分享各种原创文章。

# 负载均衡

负载均衡，是指如果一台Tomcat服务器不能满足业务需求的时候，同时启用多台Tomcat来处理业务。

但是即使有多台Tomcat服务器，对于互联网来讲，也只能有一个入口，因为太多的入口用户是无法选择的，必须有可靠的负载分配机制，通过一个统一的入口，把业务分配到合适的服务器上面进行处理。

## 基于Apache的负载均衡

Apache是一个软件基金会的名称，这个软件基金会下面，有很多的开源软件。其中的HTTP服务器就是名为Apache httpd，应用非常广，最早是应用在Unix、Linux系统上的，后来移植到了Windows，几乎统治了所有的开源HTTP服务器市场。

官方网站在 <http://httpd.apache.org/> ，在此网站可以得到软件的最新版本和更新信息。

基于Apache httpd，可以非常方便实现反向代理，对Tomcat实现负载均衡。

下面演示的是在Ubuntu下面使用apt-get命令安装的方式，如果需要源代码方式编译，请联系作者。

Windows下的安装方式非常简单，下载exe安装即可，配置和Linux相似。

### Apache安装和启动

安装

|  |
| --- |
| lwq@fkjava:~$ sudo apt-get install apache2 |

安装后，所有的配置信息放到 /etc/apache2 目录下：

|  |  |
| --- | --- |
| apache2.conf | Apache的核心配置文件，所有的配置信息入口都在这里，但是通常不建议修改此文件，避免此文件过度膨大。 |
| conf-available | 所有可用的配置，各个模块的配置都放到此处。 |
| conf-enabled | 所有激活的配置，这里面是可用配置的软链接，如果哪个配置需要启用，则对配置文件建立链接，apache2.conf会把此目录下所有链接对应的文件都包含进系统中。 |
| envvars | 此文件是apache的环境变量配置文件， |
| magic | 此文件是供mod\_mime\_magic使用的配置文件，用于智能识别文档的内容类型。 |
| mods-available | 所有的可用模块配置文件目录。 |
| mods-enabled | 所有激活的模块目录。此目录下依然是一些软连接，把需要启用的模块在此目录建立链接即可。 |
| ports.conf | 端口配置文件，专门用于配置端口的监听，包括80端口、443端口等。 |
| sites-available | 所有可用的站点配置文件目录。 |
| sites-enabled | 所有激活的站点配置文件目录，同样创建软链接。 |

之后使用的时候，首先会需要在mods-available里面创建加载模块，然后把配置文件链接到mods-enabled目录。接着还要在sites-available里面创建一个站点，用于表示tomcat，并且把文件链接到sites-enabled目录。

现在需要安装好多个Tomcat，在这个示例里面，只考虑一台服务器里面的多个Tomcat实例。这和多台服务器上面的情况配置是一样的。

### Tomcat安装和启动

Tomcat解压以后就能够使用，不需要做任何的配置，但是前提是系统要先配置好JAVA\_HOME环境变量。这里把Tomcat放到 /usr/local/tomat 目录下解压缩。

Tomcat直接复制了三个，分别是tomcat1、tomcat2、tomcat3。默认情况下，Tomcat需要三个端口，分别是8005、8009、8080，那么此时就通过修改server.xml的方式把这三个端口分别改掉。

我这里改的比较有规律，tomcat1的端口全部是81开头、tomcat2的全部82开头，同样tomcat3的则是83开头。

同时在这三个tomcat所在的目录，创建了一个 start-tomcat.sh 文件，用于同时启动三个Tomcat。

目录结构

|  |
| --- |
| lwq@fkjava:/usr/local/tomcat$ ls -l  total 20  -rwxrw-r-- 1 lwq lwq 96 Aug 7 22:53 start-tomcat.sh  -rwxrw-r-- 1 lwq lwq 93 Aug 7 23:09 stop-tomcat.sh  drwxrwxr-x 9 lwq lwq 4096 Aug 7 22:47 tomcat1  drwxrwxr-x 9 lwq lwq 4096 Aug 7 22:47 tomcat2  drwxrwxr-x 9 lwq lwq 4096 Aug 7 22:47 tomcat3  lwq@fkjava:/usr/local/tomcat$ |

启动脚本

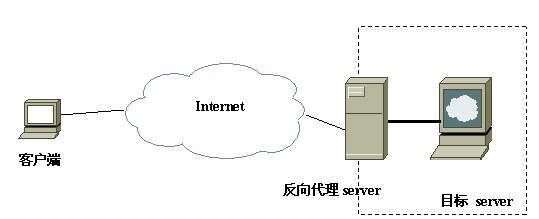
|  |
| --- |
| lwq@iZ94do94mv6Z:~$ cat /usr/local/tomcat/start-tomcat.sh  ./tomcat1/bin/catalina.sh start  ./tomcat2/bin/catalina.sh start  ./tomcat3/bin/catalina.sh start |

### 负载均衡配置

通过apache来对tomcat进行负载均衡，有两种方式，分别是JK、反向代理。而其中JK协议是早期的、并且配置比较复杂，后来基本上都只是使用反向代理。

反向代理：当Tomcat不能被浏览器直接访问的时候，需要通过一个代理服务器来访问服务器。具体过程是：先访问到前置服务器，然后由前置服务器把请求转发给实际的Tomcat。那么这个前置服务器就是我们在这里使用的apache服务器。

具体的反向代理过程通过下图示意。



Apache支持两种方式和Tomcat进行反向代理，一种是基于HTTP的、另外一种则是基于AJP协议的。其中AJP协议的方式性能会比较好。这里两种方式分开讲。

**基于HTTP的反向代理**

* 加载模块

|  |
| --- |
| #Ubuntu 里面使用apt-get方式安装的apache，所有的模块都已经有了  #现在只需要在 /etc/apache2/mods-enabled 目录里面，创建对应配置的链接即可  #下面的命令在 /etc/apache2/modes-enabled 目录下面执行  sudo ln -s ../mods-available/proxy.load proxy.load  sudo ln -s ../mods-available/proxy\_http.load proxy\_http.load |

* 创建站点

|  |
| --- |
| #接着要在 site-available目录下复制创建一个 fkjava.com.conf 文件，内容如下：  <VirtualHost \*:80>  #表示如果http头里面的Host为www.fkjava.com的时候，访问此虚拟主机  ServerName www.fkjava.com  ServerAdmin luo\_wenqiang@qq.com  ErrorLog ${APACHE\_LOG\_DIR}/error.log  CustomLog ${APACHE\_LOG\_DIR}/access.log combined  #直接配置ajp代理  # ProxyPass 路径 协议://主机:端口/URI  ProxyPass / ajp://127.0.0.1:8109/  # 所有访问 <http://www.fkjava.org> 的请求，都被自动代理转发到 /  # 而这个 / 则是前面 ProxyPass 里面的 /  ProxyPassReverse / http://www.fkjava.com  </VirtualHost> |

* 激活站点

|  |
| --- |
| #和其他的配置一样，站点也是通过创建链接来激活的  #这样如果要把站点删除，只需要把链接删除掉就可以了。  #下面的命令在 /etc/apache2/site-enabled 下执行  sudo ln -s ../sites-available/fkjava.com.conf fkjava.com.conf |

以上配置是模块化编译的apache里面有效，如果是Windows系统，需要根据不同的发行版来确定配置方式，但是一般可以把内容全部加入到 httpd.conf 文件中。这个是通用的。

完成以上动作以后，需要修改一下本地的域名解析，用于测试 VirtualHost 能否访问成功。在非Windows系统，通常修改 /etc/hosts 文件，在Windows则修改C:\Windows\System32\devices\etc\hosts文件。增加的内容大约如下：

|  |
| --- |
| #格式固定，前面IP，后面域名  120.76.xx.xx www.fkjava.com |

然后重启apache，并且在浏览器输入 [http://www.fkjava.com](http://www.fkjava.org) 即可访问到tomcat了。

**基于AJP的反向代理**

* 加载模块

|  |
| --- |
| #Ubuntu 里面使用apt-get方式安装的apache，所有的模块都已经有了  #现在只需要在 /etc/apache2/mods-enabled 目录里面，创建对应配置的链接即可  #下面的命令在 /etc/apache2/modes-enabled 目录下面执行  sudo ln -s ../mods-available/proxy.load proxy.load  sudo ln -s ../mods-available/proxy\_ajp.load proxy\_ajp.load |

* 创建站点

|  |
| --- |
| #接着要在 site-available目录下复制创建一个 fkjava.net.conf 文件，内容如下：  <VirtualHost \*:80>  #表示如果http头里面的Host为www.fkjava.net的时候，访问此虚拟主机  ServerName www.fkjava.net  ServerAdmin luo\_wenqiang@qq.com  ErrorLog ${APACHE\_LOG\_DIR}/error.log  CustomLog ${APACHE\_LOG\_DIR}/access.log combined  #直接配置ajp代理  # ProxyPass 路径 协议://主机:端口/URI  ProxyPass / ajp://127.0.0.1:8209/  # 所有访问 <http://www.fkjava.org> 的请求，都被自动代理转发到 /  # 而这个 / 则是前面 ProxyPass 里面的 /  ProxyPassReverse / http://www.fkjava.net  </VirtualHost> |

* 激活站点

|  |
| --- |
| #和其他的配置一样，站点也是通过创建链接来激活的  #这样如果要把站点删除，只需要把链接删除掉就可以了。  #下面的命令在 /etc/apache2/site-enabled 下执行  sudo ln -s ../sites-available/fkjava.net.conf fkjava.net.conf |

以上配置是模块化编译的apache里面有效，如果是Windows系统，需要根据不同的发行版来确定配置方式，但是一般可以把内容全部加入到 httpd.conf 文件中。这个是通用的。

完成以上动作以后，需要修改一下本地的域名解析，用于测试 VirtualHost 能否访问成功。在非Windows系统，通常修改 /etc/hosts 文件，在Windows则修改C:\Windows\System32\devices\etc\hosts文件。增加的内容大约如下：

|  |
| --- |
| #格式固定，前面IP，后面域名  120.76.xx.xx www.fkjava.net |

然后重启apache，并且在浏览器输入 [http://www.fkjava.net](http://www.fkjava.org) 即可访问到tomcat了。

## 基于Nginx的负载均衡

# Tomcat的Cluster

# Apache和Cluster的结合使用

# Nginx和Cluster的结合使用

# 使用Memcached管理Session