[**nginx+iis实现负载均衡**](http://www.cnblogs.com/yanweidie/p/4658136.html)

　　http://www.cnblogs.com/yanweidie/p/4658136.html

最近在研究分布式系统架构方面的知识，包括负载均衡，数据库读写分离，分布式缓存redis等。本篇先从负载均衡服务架构入手，关于负载均衡百度百科的定义如下：负载均衡，英文名称为Load Balance，其意思就是分摊到多个操作单元上进行执行，例如Web服务器、FTP服务器、企业关键应用服务器和其它关键任务服务器等，从而共同完成工作任务。

　　我的解释：一项任务交由一个开发人员处理总会有上限处理能力，这时可以考虑增加开发人员来共同处理这项任务，多人处理同一项任务时就会涉及到调度问题，即任务分配，这和多线程理念是一致的。nginx在这里的角色相当于任务分配者。

**阅读目录**

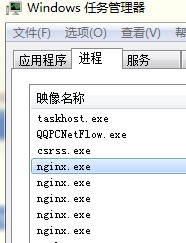
* [nginx安装](http://www.cnblogs.com/yanweidie/p/4658136.html#_label0)
* [站点搭建及配置](http://www.cnblogs.com/yanweidie/p/4658136.html#_label1)
* [问题分析](http://www.cnblogs.com/yanweidie/p/4658136.html#_label2)
* [总结](http://www.cnblogs.com/yanweidie/p/4658136.html#_label3)

[回到顶部](http://www.cnblogs.com/yanweidie/p/4658136.html#_labelTop)

**nginx安装**

**Nginx**是一款轻量级的Web 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。由俄罗斯的程序设计师Igor Sysoev所开发，供俄国大型的入口网站及搜索引擎Rambler（俄文：Рамблер）使用。其特点是占有内存少，并发能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：百度、新浪、网易、腾讯等。

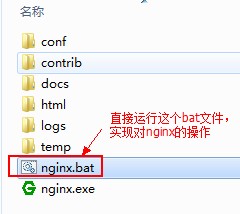
　 最新版本的nginx版本为1.9.3，我这下载的是window版本的，一般实际场景都是安装在linux系统下的，由于linux系统目前正在摸索中这里就不介绍。官方下载地址：[nginx-1.9.3.zip](http://nginx.org/download/nginx-1.9.3.zip) 博客园下载地址：[nginx-1.9.3.zip](http://files.cnblogs.com/files/yanweidie/nginx-1.9.3.zip) 。下载完成之后解压运行nginx.exe就启动了nginx了，启动后会在进程里面看到nginx。



   要实现负载均衡需要修改conf/nginx.conf的配置信息，修改配置信息之后重新启动nginx服务，可以通过nginx -s reload指令实现。这里我们使用 [Ants](http://leleroyn.cnblogs.com/) 提供的一个批处理来操作。



将nginx.bat文件放到nginx.exe相同文件夹下，直接运行就行了。文章结尾会提供本文用到的所有文件。



[回到顶部](http://www.cnblogs.com/yanweidie/p/4658136.html#_labelTop)

**站点搭建及配置**

**1.搭建两个iis站点**

 站点下只有一个简单的index页面，用来输出当前服务器信息。由于我没有两台机器，所以将两个站点都部署到本机了，分别绑定了8082和9000两个端口。

[复制代码](javascript:void(0);)

protected void Page\_Load(object sender, EventArgs e)

{

Label0.Text = "请求开始时间："+DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");

Label1.Text = "服务器名称：" + Server.MachineName;//服务器名称

Label2.Text = "服务器IP地址：" + Request.ServerVariables["LOCAL\_ADDR"];//服务器IP地址

Label3.Text = "HTTP访问端口：" + Request.ServerVariables["SERVER\_PORT"];//HTTP访问端口"

Label4.Text = ".NET解释引擎版本：" + ".NET CLR" + Environment.Version.Major + "." + Environment.Version.Minor + "." + Environment.Version.Build + "." + Environment.Version.Revision;//.NET解释引擎版本

Label5.Text = "服务器操作系统版本：" + Environment.OSVersion.ToString();//服务器操作系统版本

Label6.Text = "服务器IIS版本：" + Request.ServerVariables["SERVER\_SOFTWARE"];//服务器IIS版本

Label7.Text = "服务器域名：" + Request.ServerVariables["SERVER\_NAME"];//服务器域名

Label8.Text = "虚拟目录的绝对路径：" + Request.ServerVariables["APPL\_RHYSICAL\_PATH"];//虚拟目录的绝对路径

Label9.Text = "执行文件的绝对路径：" + Request.ServerVariables["PATH\_TRANSLATED"];//执行文件的绝对路径

Label10.Text = "虚拟目录Session总数：" + Session.Contents.Count.ToString();//虚拟目录Session总数

Label11.Text = "虚拟目录Application总数：" + Application.Contents.Count.ToString();//虚拟目录Application总数

Label12.Text = "域名主机：" + Request.ServerVariables["HTTP\_HOST"];//域名主机

Label13.Text = "服务器区域语言：" + Request.ServerVariables["HTTP\_ACCEPT\_LANGUAGE"];//服务器区域语言

Label14.Text = "用户信息：" + Request.ServerVariables["HTTP\_USER\_AGENT"];

Label14.Text = "CPU个数：" + Environment.GetEnvironmentVariable("NUMBER\_OF\_PROCESSORS");//CPU个数

Label15.Text = "CPU类型：" + Environment.GetEnvironmentVariable("PROCESSOR\_IDENTIFIER");//CPU类型

Label16.Text = "请求来源地址：" + Request.Headers["X-Real-IP"];

}

[复制代码](javascript:void(0);)

**2.修改nginx配置信息**

修改nginx监听端口，修改http server下的listen节点值，由于本机80端口已经被占用，我改为监听8083端口。

   listen       8083;

  在http节点下添加upstream（服务器集群），server设置的是集群服务器的信息，我这里搭建了两个站点，配置了两条信息。

    #服务器集群名称为Jq\_one  
    upstream Jq\_one {  
 　　server  127.0.0.1:9000;   
 　　server  127.0.0.1:8082;   
    }

 在http节点下找到location节点修改

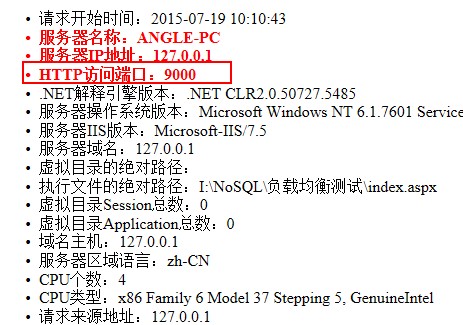
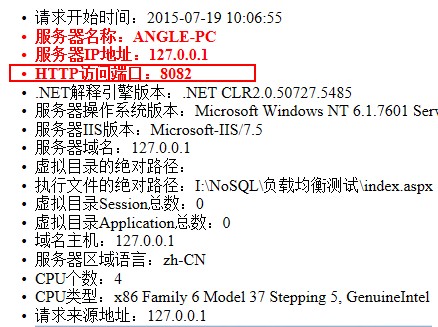
  location / {  
            root   html;  
            index  index.aspx index.html index.htm; #修改主页为index.aspx  
     #其中jq\_one 对应着upstream设置的集群名称  
     proxy\_pass         http://Jq\_one;   
     #设置主机头和客户端真实地址，以便服务器获取客户端真实IP  
     proxy\_set\_header   Host             $host;   
     proxy\_set\_header   X-Real-IP        $remote\_addr;   
     proxy\_set\_header   X-Forwarded-For  $proxy\_add\_x\_forwarded\_for;  
 }

修改完成配置文件以后记得重启nginx服务，最终完整配置文件信息如下



**3.运行结果**

访问http://127.0.0.1:8083/index.aspx ，多访问几次，着重关注标红部分。



可以看到，我们的请求被分发到了8082站点和9000站点，并且第一次是8082站点第二次9000。出现这样的结果证明我们负载均衡搭建成功了。 尝试关闭其中的9000站点，然后刷新页面发现输出的http端口一直是8082，也就是说其中一个站点挂了，只要还有一个站点是好的，我们的还是可以服务。

[回到顶部](http://www.cnblogs.com/yanweidie/p/4658136.html#_labelTop)

**问题分析**

  虽然我们搭建好了负载均衡站点，但是还存在以下问题。

**1.如果站点使用了session，请求平均分配到两个站点，那么必然存在session共享问题，该如何解决？**

* 使用数据库保存session信息
* 使用nginx将同一ip的请求分配到固定服务器，修改如下。ip\_hash会计算ip对应hash值，然后分配到固定服务器

　　upstream Jq\_one{  
  　　server 127.0.0.1:8082 ;  
 　　 server 127.0.0.1:9000 ;  
   　　ip\_hash;  
　　}

* 搭建一台Redis服务器，对session的读取都从该Redis服务器上读取。后面的文章将介绍分布式缓存Redis的使用

**2.管理员更新站点文件，该怎么操作，现在还只有两台服务器，可以手工将文件更新到两台服务器，如果是10台呢，那么手工操作必然是不可行的**

* 多服务器站点更新可以使用GoodSync 文件同步程序，会自动检测文件的修改新增，然后同步到其它服务器上。在linux下可以使用rsync

**3.站点中的文件上传功能会将文件分配到不同的服务器，文件共享问题如何解决。**

* 使用文件服务器将所有文件存储到该服务器上，文件操作读取写入都在该服务器上。这里同样会存在一个问题，文件服务器存在读写上限。

**4.负载的服务器配置不一样，有的高有的低可不可以让配置高的服务器处理请求多一些**

* 这里讲一下，负载均衡有好几种算法  轮转法，散列法， 最少连接法，最低缺失法，最快响应法，加权法。我们这里可以使用加权法来分配请求。

     upstream Jq\_one{  
  　　server 127.0.0.1:8082 weight=4;  
 　　 server 127.0.0.1:9000 weight=1;  
　　}

     通过weight设置每台服务器分配请求站的权重，值越高分配的越多。

**5.由于请求是经过nginx转发过来的，可以在代码里面获取到用户请求的实际ip地址吗？**

* 答案是肯定的，在localtion节点设置如下请求头信息

    #设置主机头和客户端真实地址，以便服务器获取客户端真实IP  
     proxy\_set\_header   Host             $host;   
     proxy\_set\_header   X-Real-IP        $remote\_addr;   
     proxy\_set\_header   X-Forwarded-For  $proxy\_add\_x\_forwarded\_for;

    代码里面通过Request.Headers["X-Real-IP"]，就能获取到真实ip

**6.nginx实现静态文件(image,js,css)缓存**

* 在server节点下添加新的localtion
* #静态资源缓存设置  
   location ~ \.(jpg|png|jpeg|bmp|gif|swf|css)$  
          {   
              expires 30d;  
              root /nginx-1.9.3/html;#root:  #静态文件存在地址，这里设置在/nginx-1.9.3/html下  
              break;  
          }

    这是index页面的代码 <li><img src="/images/1.jpg"/></li>



[回到顶部](http://www.cnblogs.com/yanweidie/p/4658136.html#_labelTop)

**总结**

　　通过nginx我们实现了一个简单的负载均衡，实际情况比这复杂很多。比如nginx服务器挂了，那我们的站点就直接挂了，正确的通过keepalived组件来搭建多台nginx服务提供服务。本篇只做为分布式系统的开篇，后续会陆续推出Redis缓存，数据库实现分布式架构的文章，敬请期待！希望能够得到博客园分布式大牛的指导。

      本篇文章用到的资源打包下载地址：[nginx\_iis](http://files.cnblogs.com/files/yanweidie/nginx_iis.rar)