用Splash做页面抓取时，如果爬取的量非常大，任务非常多，用一个Splash服务来处理的话，未免压力太大了，此时可以考虑搭建一个负载均衡器来把压力分散到各个服务器上。这相当于多台机器多个服务共同参与任务的处理，可以减小单个Splash服务的压力。

## 1. 配置Splash服务

要搭建Splash负载均衡，首先要有多个Splash服务。假如这里在4台远程主机的8050端口上都开启了Splash服务，它们的服务地址分别为41.159.27.223:8050、41.159.27.221:8050、41.159.27.9:8050和41.159.117.119:8050，这4个服务完全一致，都是通过Docker的Splash镜像开启的。访问其中任何一个服务时，都可以使用Splash服务。

## 2. 配置负载均衡

接下来，可以选用任意一台带有公网IP的主机来配置负载均衡。首先，在这台主机上装好Nginx，然后修改Nginx的配置文件nginx.conf，添加如下内容：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | http {      upstream splash {          least\_conn;          server 41.159.27.223:8050;          server 41.159.27.221:8050;          server 41.159.27.9:8050;          server 41.159.117.119:8050;      }      server {          listen 8050;          location / {              proxy\_pass http:*//splash;*          }      }  } |

这样我们通过upstream字段定义了一个名字叫作splash的服务集群配置。其中least\_conn代表最少链接负载均衡，它适合处理请求处理时间长短不一造成服务器过载的情况。

当然，我们也可以不指定配置，具体如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | upstream splash {      server 41.159.27.223:8050;      server 41.159.27.221:8050;      server 41.159.27.9:8050;      server 41.159.117.119:8050;  } |

这样默认以轮询策略实现负载均衡，每个服务器的压力相同。此策略适合服务器配置相当、无状态且短平快的服务使用。

另外，我们还可以指定权重，配置如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | upstream splash {      server 41.159.27.223:8050 weight=4;      server 41.159.27.221:8050 weight=2;      server 41.159.27.9:8050 weight=2;      server 41.159.117.119:8050 weight=1;  } |

这里weight参数指定各个服务的权重，权重越高，分配到处理的请求越多。假如不同的服务器配置差别比较大的话，可以使用此种配置。

最后，还有一种IP散列负载均衡，配置如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | upstream splash {      ip\_hash;      server 41.159.27.223:8050;      server 41.159.27.221:8050;      server 41.159.27.9:8050;      server 41.159.117.119:8050;  } |

服务器根据请求客户端的IP地址进行散列计算，确保使用同一个服务器响应请求，这种策略适合有状态的服务，比如用户登录后访问某个页面的情形。对于Splash来说，不需要应用此设置。

我们可以根据不同的情形选用不同的配置，配置完成后重启一下Nginx服务：

|  |  |
| --- | --- |
| 1 | sudo nginx -s reload |

这样直接访问Nginx所在服务器的8050端口，即可实现负载均衡了。

## 3. 配置认证

现在Splash是可以公开访问的，如果不想让其公开访问，还可以配置认证，这仍然借助于Nginx。可以在server的location字段中添加auth\_basic和auth\_basic\_user\_file字段，具体配置如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | http {      upstream splash {          least\_conn;          server 41.159.27.223:8050;          server 41.159.27.221:8050;          server 41.159.27.9:8050;          server 41.159.117.119:8050;      }      server {          listen 8050;          location / {              proxy\_pass http:*//splash;*              auth\_basic "Restricted";              auth\_basic\_user\_file /etc/nginx/conf.d/.htpasswd;          }      }  } |

这里使用的用户名和密码配置放置在/etc/nginx/conf.d目录下，我们需要使用htpasswd命令创建。例如，创建一个用户名为admin的文件，相关命令如下：

|  |  |
| --- | --- |
| 1 | htpasswd -c .htpasswd admin |

接下来就会提示我们输入密码，输入两次之后，就会生成密码文件，其内容如下：

|  |  |
| --- | --- |
| 1  2 | cat .htpasswd  admin:5ZBxQr0rCqwbc |

配置完成后，重启一下Nginx服务：

|  |  |
| --- | --- |
| 1 | sudo nginx -s reload |

这样访问认证就成功配置好了。

## 4. 测试

最后，我们可以用代码来测试一下负载均衡的配置，看看到底是不是每次请求会切换IP。利用[http://httpbin.org/get](http://httpbin.org/get" \o ")测试即可，实现代码如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | import requests  from urllib.parse import quote  import re    lua = '''  function main(splash, args)    local treat = require("treat")    local response = splash:http\_get("http://httpbin.org/get")    return treat.as\_string(response.body)  end  '''    url = 'http://splash:8050/execute?lua\_source=' + quote(lua)  response = requests.get(url, auth=('admin', 'admin'))  ip = re.search('(\d+\.\d+\.\d+\.\d+)', response.text).group(1)  print(ip) |

这里URL中的splash字符串请自行替换成自己的Nginx服务器IP。这里我修改了Hosts，设置了splash为Nginx服务器IP。

多次运行代码之后，可以发现每次请求的IP都会变化，比如第一次的结果：

|  |  |
| --- | --- |
| 1 | 41.159.27.223 |

第二次的结果：

|  |  |
| --- | --- |
| 1 | 41.159.27.9 |

这就说明负载均衡已经成功实现了。

本节中，我们成功实现了负载均衡的配置。配置负载均衡后，可以多个Splash服务共同合作，减轻单个服务的负载，这还是比较有用的。