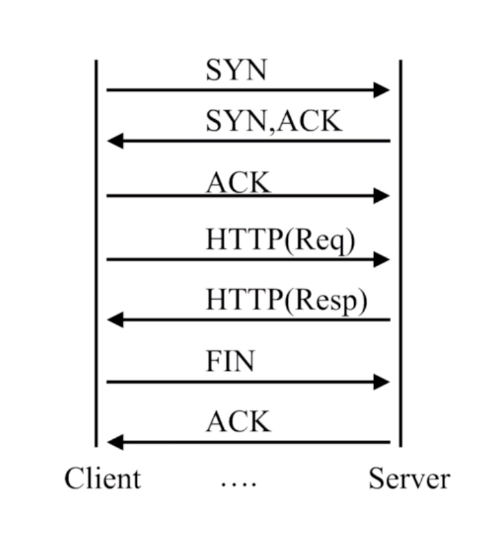
Nginx访问限制

连接频率限制 limit\_conn\_module

请求频率限制 limit\_req\_module

**http协议的连接与请求**

HTTP是建立在TCP，在完成HTTP请求需要先建立TCP三次握手（称为TCP连接），在连接的基础上在HTTP请求。



HTTP协议的连接与请求

|  |  |
| --- | --- |
| **HTTP协议版本** | **连接关系** |
| HTTP1.0 | TCP不能复用 |
| HTTP1.1 | 顺序性TCP复用 |
| HTTP2.0 | 多路复用TCP复用 |

HTTP请求建立在一次TCP连接基础上

一次TCP请求至少产生一次HTTP请求

**Nginx连接限制配置**

//Nginx连接限制语法

Syntax: limit\_conn\_zone key zone= name:size;

Default: —

Context: http

Syntax: limit\_conn zone number;

Default: —

Context: http, server, location

//具体配置如下:

http {

//http段配置连接限制, 同⼀一时刻只允许⼀一个客户端IP连接

limit\_conn\_zone $binary\_remote\_addr zone=conn\_zone:10m;

...

server {

...

location / {

//同⼀一时刻只允许⼀一个客户端IP连接

limit\_conn conn\_zone 1;

}

//压⼒力力测试

yum install -y httpd-tools

ab -n 50 -c 20 http://127.0.0.1/index.html

**Nginx请求限制配置**

//Nginx

请求限制语法

Syntax: limit\_req\_zone key zone=name:size rate=rate;

Default: —

Context: http

Syntax: limit\_conn zone number [burst=number] [nodelay];

Default: —

Context: http, server, location

//具体配置如下:

http {

//http段配置请求限制, rate限制速率，限制⼀一秒钟最多⼀一个IP请求

limit\_req\_zone $binary\_remote\_addr zone=req\_zone:10m rate=1r/s;

...

server {

...

Location / {

//1r/s只接收⼀一个请求,其余请求拒绝处理理并返回错误码给客户端

limit\_req zone=req\_zone;

//请求超过1r/s,剩下的将被延迟处理理,请求数超过burst定义的数量量, 多余的请返回503

#limit\_req zone=req\_zone burst=3 nodelay;

}

//压⼒力力测试

yum install -y httpd-tools

ab -n 50 -c 20 http://127.0.0.1/index.html

**连接限制没有请求限制有效?**

我们前⾯面说过, 多个请求可以建⽴立在⼀一次的TCP连接之上, 那么我们对请求的精度限制，当然⽐比对⼀一个连接的限制会更更加的有效。

因为同⼀一时刻只允许⼀一个连接请求进⼊入。

但是同⼀一时刻多个请求可以通过⼀一个连接进⼊入。

所以请求限制才是⽐比较优的解决⽅方案。