

# Nginx性能优化

Nginx作为一个非常流行和成熟的Web Server和Reverse Proxy Server，网上有大量的性能优化教程，但是不同的业务场景千差万别，什么配置是最适合自己的，需要大量的测试和实践以及不断的优化改进。最近用户调用量突破百万大关之后，就遇到了一些问题，虽然不算太复杂，但也折腾了挺长时间才搞定，积累了不少经验。

问题：

系统监控是否存在“盲区”？

碰到的这个问题其实已经有一段时间了，有客户给我们反馈调用超时，但是我们自己从系统监控上看都是正常的，只有几十毫秒肯定不会超时，怀疑是不是网络的原因，但是出现几次后，就隐隐感觉这个问题可能不是偶发性的，应该还有深层次的原因。

业务特性：

因为我们服务面向企业客户的，虽然每家客户的调用量可能会非常大，但每家企业客户就那么几个公网IP，即使以后有上千家客户，Nginx也可以轻松支撑这些并发连接。因此，首先先从网络上对Nginx长连接作了优化，将长连接从原来配置的 5秒钟 改成 5分钟，将每次建立连接请求的数目从默认的100调整到1000。

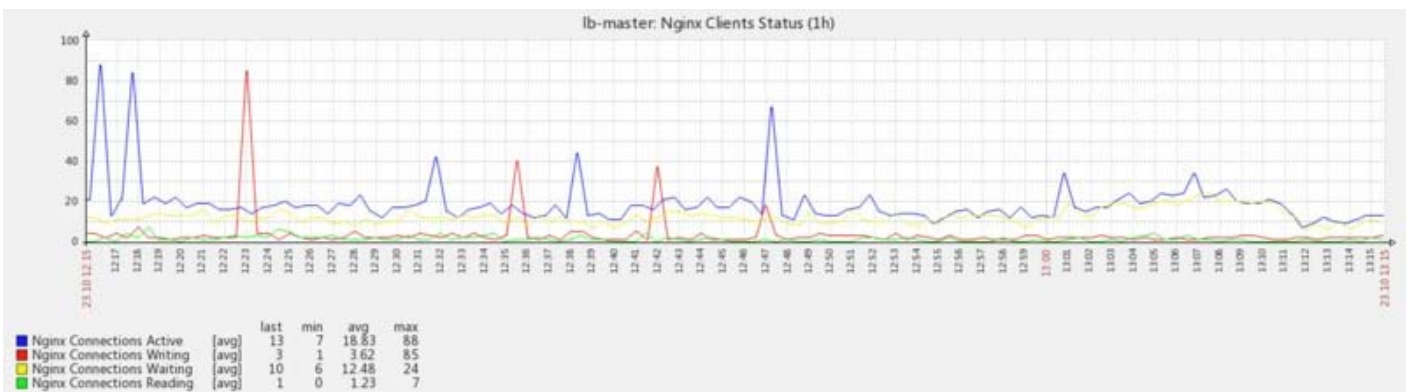
```
keepalive_timeout 300;
keepalive_requests 1000;
```

问题再次发生：

调整完毕后，通过 `netstat -anp` 命令可以看到，新建连接请求会减少，说明长连接已起到作用。但过了一段时间，仍然发现有客户调用超时的情况发生，从Nginx日志中可以看到请求时间还是有超过1s的，甚至有长达20s左右的，如下所示：

```
59.151.111.111 - - [23/Oct/2014:09:19:06 +0800] "POST /riskService HTTP/1.0" 200 1030 "-" "-" "-" "19.114" "https://risk-service.114.com.cn/riskService.do?token_id=Elba40d83f5a0ae8fbb2da0c16cf3c1eb8eeb09205e2ced6"
```

并且从Zabbix上的监控发现一个现象，当connection writing或active数突然增高时，请求时间就相应的出现较多超时：



查看应用的日志，发现执行时间并不长：

```
client --> Nginx --> Tomcat --> App
```

2015/8/18 13:17



```

[23/Oct/2014:09:19:07 +0800] 200 1523 13 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:07 +0800] 200 976 10 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:08 +0800] 200 410 4 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:08 +0800] 200 410 3 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:09 +0800] 200 426 9 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:09 +0800] 200 410 4 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:10 +0800] 200 1274 10 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:11 +0800] 200 411 18 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:11 +0800] 200 1625 17 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:12 +0800] 200 410 6 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:12 +0800] 200 426 10 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:13 +0800] 200 410 7 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:14 +0800] 200 410 4 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:15 +0800] 200 426 10 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:15 +0800] 200 410 4 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:16 +0800] 200 1030 22 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:16 +0800] 200 687 16 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:16 +0800] 200 410 5 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:17 +0800] 200 976 11 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:17 +0800] 200 752 9 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:18 +0800] 200 428 10 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:19 +0800] 200 640 19 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:19 +0800] 200 410 4 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:20 +0800] 200 426 9 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:21 +0800] 200 1394 13 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:21 +0800] 200 1001 11 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:21 +0800] 200 1029 10 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:21 +0800] 200 4 2 "GET //ok.htm HTTP/1.1"
[23/Oct/2014:09:19:21 +0800] 200 410 3 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:22 +0800] 200 976 11 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:22 +0800] 200 426 10 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:23 +0800] 200 410 3 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:23 +0800] 200 410 7 "POST /riskService HTTP/1.1"
[23/Oct/2014:09:19:23 +0800] 200 913 17 "POST /riskService HTTP/1.1"

```

从请求路径上分析，肯定是Nginx到Tomcat这层存在一些问题。正在排查这个问题的时候，突然发现有大量30s左右的超时，从Zabbix上也观察到 `connection writing` 非常高，如下所示：



现象：

同时，发现 `TIME_WAIT` 的连接特别多，从现象及抓包分析结果来看，应该是有客户没有开启长连接，而我们在服务端又设置了 `keepalive timeout` 为5分钟，导致大量使用过的连接等待超时，当时有接近2000个，编辑 `/etc/sysctl.conf` 文

件，增加如下两个参数重用连接：

```
net.ipv4.tcp_tw_reuse = 1 #表示开启重用。允许将TIME-WAIT sockets重新用于新的TCP连接，默认为0，表示关闭
net.ipv4.tcp_tw_recycle = 1 #表示开启TCP连接中TIME-WAIT sockets的快速回收，默认为0，表示关闭。
```

调整后效果：

生效之后很快下降到200以下，从Zabbix监控上也看到，`connection writing` 和 `connection active` 都有明显下降，但并没有完全解决问题，还得找其它方面的原因。



Nginx的 `request time` 指的是从客户端接收到第一个字节算起，到调用后端的upstream server完成业务逻辑处理，并将返回结果全部写回到客户端为止的时间，那么调用upstream server的时间如果能够打印出来的话，就更容易将问题范围缩小，幸运的是Nginx有两个参数可以打印后端服务器请求的时间和IP地址，在nginx.conf文件中修改日志的格式如下：

```
log_format main '$remote_addr - [$time_local] "$request" '
                '$status $body_bytes_sent '
                '"$request_time" "$upstream_response_time" "$upstream_addr" "$request_body"';
```

# \$upstream\_response\_time 后端应用服务器响应时间

# \$upstream\_addr 后端服务器IP和端口号

```
log_format main '$remote_addr - [$time_local] "$request" '
                '$status $body_bytes_sent '
                '"$request_time" "$upstream_response_time" "$upstream_addr" "$request_body"';
```

Nginx日志有“盲区”

再观察日志，非常明显地发现，大部分特别长的调用都来自同一台机器：



查看这台机器发现，虽然Java进程还在，但应用实际上已经当掉了，没有真实的请求进来，将之从负载均衡中摘掉，问题马上得到缓解：





另一个“血案”：

这台机器其实已经挂掉了，但为何Nginx没有识别到呢？进一步研究发现，Nginx在调用upstream server时，超时时间默认是60s，我们这些应用对响应时间要求非常高，超过1s已没有意义，因此在nginx.conf文件中修改默认的超时时间，超过1s就返回：

```
# time out settings
proxy_connect_timeout 1s;
proxy_send_timeout 1s;
proxy_read_timeout 1s;
```

效果：

运行一段时间后，问题已基本得到解决，不过还是会发现 request time 超过1s达到5s的，但 upstream response time 都没有超时了，说明上面的参数已起作用，根据我的理解，request time 比较长的原因可能跟客户那边接收慢有关系，不过这个问题最终还需要下周等客户改为长连接才能确认。

yikebocai / 2014-10-25

Published under (CC) BY-NC-SA in categories tech tagged with nginx