

11.Nginx性能优化

- 11.Nginx性能优化
 - 性能优化概述
 - 压力测试工具
 - 影响性能指标
 - 系统性能优化
 - Nginx性能优化

徐亮伟, 江湖人称标杆徐。多年互联网运维工作经验, 曾负责过大规模集群架构自动化运维管理工作。擅长Web集群架构与自动化运维, 曾负责国内某大型电商运维工作。

个人博客"[徐亮伟架构师之路](#)"累计受益数万人。

笔者Q:552408925、572891887

架构师群:471443208

性能优化概述

在做性能优化前, 我们需要对如下进行考虑

- 1.当前系统结构瓶颈
 - 观察指标
 - 压力测试
- 2.了解业务模式
 - 接口业务类型
 - 系统层次化结构
- 3.性能与安全
 - 性能好安全弱
 - 安全好性能低

压力测试工具

1.安装压力测试工具 ab

```
[root@nginx-lua ~]# yum install httpd-tools -y
```

2.了解压测工具使用方式

```
[root@nginx-lua ~]# ab -n 200 -c 2 http://127.0.0.1/
```

// -n 总的请求次数

// -c 并发请求数

// -k 是否开启长连接

3. 配置 Nginx 静态网站与 tomcat 动态网站环境

```
[root@nginx-lua conf.d]# cat jsp.conf
```

```
server {  
    server_name localhost;  
    listen 80;  
    location / {  
        root /soft/code;  
        try_files $uri @java_page;  
        index index.jsp index.html;  
    }  
    location @java_page {  
        proxy_pass http://192.168.56.20:8080;  
    }  
}
```

// 分别给 Nginx 准备静态网站

```
[root@nginx-lua ~]# cat /soft/code/bgx.html
```

```
<h1> Ab Load </h1>
```

// 给 Tomcat 准备静态网站文件

```
[root@tomcat-node1-20 ROOT]# cat /soft/tomcat-8080/webapps/ROOT/bgx.html
```

```
<h1> Ab Load </h1>
```

4. 使用 ab 工具进行压力测试

// 进行压力测试

```
[root@Nginx conf.d]# ab -n2000 -c2 http://127.0.0.1/bgx.html
```

...

Server Software: nginx/1.12.2

Server Hostname: 127.0.0.1

Server Port: 80

Document Path: /bgx.html

Document Length: 19 bytes

Concurrency Level: 200

总花费总时长

Time taken for tests: 1.013 seconds

总请求数

```
Complete requests:      2000
# 请求失败数
Failed requests:        0
Write errors:           0
Total transferred:      510000 bytes
HTML transferred:       38000 bytes
# 每秒多少请求/s(总请求出/总共完成的时间)
Requests per second:    9333.23 [#/sec] (mean)
# 客户端访问服务端，单个请求所需花费的时间
Time per request:       101.315 [ms] (mean)
# 服务端处理请求的时间
Time per request:       0.507 [ms] (mean, across all concurrent requests)
# 判断网络传输速率，观察网络是否存在瓶颈
Transfer rate:          491.58 [Kbytes/sec] received
```

5.将 `nginx` 下的 `bgx` 文件移走, 再次压测会由 `tomcat` 进行处理

```
Concurrency Level:      200
Time taken for tests:    1.028 seconds
Complete requests:      2000
Failed requests:         0
Write errors:           0
Total transferred:      510000 bytes
HTML transferred:       38000 bytes
Requests per second:    1945.09 [#/sec] (mean)
Time per request:       102.823 [ms] (mean)
Time per request:       0.514 [ms] (mean, across all concurrent requests)
Transfer rate:          484.37 [Kbytes/sec] received
```

影响性能指标

影响性能方便整体关注

- 1.网络
 - 网络的流量
 - 网络是否丢包
 - 这些会影响http的请求与调用
- 2.系统
 - 硬件有没有磁盘损坏,磁盘速率
 - 系统负载、内存、系统稳定性
- 3.服务
 - 连接优化、请求优化

- 根据业务形态做对应的服务设置
- 4.程序
 - 接口性能
 - 处理速度
 - 程序执行效率
- 5.数据库

每个架构服务与服务之间都或多或少有一些关联, 我们需要将整个架构进行分层, 找到对应系统或服务的短板, 然后进行优化

系统性能优化

- 文件句柄, Linux一切皆文件, 文件句柄可以理解为就是一个索引
 - 文件句柄会随着我们进程的调用频繁增加
 - 系统默认对文件句柄有限制, 不能让一个进程无限的调用
 - 需要限制每个进程和每个服务使用多大的文件句柄
 - 文件句柄是必须要调整的优化参数
- 设置方式
 - 系统全局性修改
 - 用户局部性修改
 - 进程局部性修改

```
vim /etc/security/limits.conf
//针对root用户
root soft nofile 65535
root hard nofile 65535
//所有用户, 全局
* soft nofile 25535
* hard nofile 25535

//对于Nginx进程
worker_rlimit_nofile 45535;

//root用户
//soft提醒
//hard限制
//nofile文件数配置项
//65535最大大小
```

Nginx性能优化

CPU 亲和, 减少进程之间不断频繁迁移, 减少性能损耗

1. 查看当前 CPU 物理状态

```
[root@nginx ~]# lscpu |grep "CPU(s)"
CPU(s):                24
On-line CPU(s) list:   0-23
NUMA node0 CPU(s):    0,2,4,6,8,10,12,14,16,18,20,22
NUMA node1 CPU(s):    1,3,5,7,9,11,13,15,17,19,21,23
```

//2颗物理cpu, 没颗cpu12核心, 总共24核心

2. 将 Nginx worker 进程绑到不同的核心上

```
//启动多少worker进程, 官方建议和cpu核心一致, 第一种绑定组合方式
#worker_processes 24;
#worker_cpu_affinity 000000000001 000000000010 000000000100 000000001000 00000000100
00 000000100000 000001000000 000010000000 000100000000 001000000000 001000000000 010000000000 10
0000000000;

//第二种方式
#worker_processes 2;
#worker_cpu_affinity 101010101010 010101010101;

//最佳方式绑定方式
worker_processes auto;
worker_cpu_affinity auto;
```

3. 查看 nginx worker 进程绑定至对应 cpu

```
ps -eo pid,args,psr|grep [n]ginx
```

4. Nginx 通用优化配置文件

```
[root@nginx ~]# cat nginx.conf
user nginx;
worker_processes auto;
worker_cpu_affinity auto;

error_log /var/log/nginx/error.log warn;
pid /run/nginx.pid;
#调整至1w以上, 负荷较高建议2-3w以上
worker_rlimit_nofile 35535;
```

```
events {
    use epoll;
    #限制每个进程能处理多少个连接请求,10240x16
    worker_connections 10240;
}

http {
    include                /etc/nginx/mime.types;
    default_type           application/octet-stream;
    # 统一使用utf-8字符集
    charset utf-8;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';
    access_log /var/log/nginx/access.log main;

    # Core module
    sendfile on;
    # 静态资源服务器建议打开
    tcp_nopush on;
    # 动态资源服务建议打开,需要打开keepalived
    tcp_nodelay on;
    keepalive_timeout 65;

    # Gzip module
    gzip on;
    gzip_disable "MSIE [1-6]\.";
    gzip_http_version 1.1;

    # Virtual Server
    include /etc/nginx/conf.d/*.conf;
}
```