shell 在手分析服务器日志不愁?

自己的小网站跑在阿里云的 ECS 上面,偶尔也去分析分析自己网站服务器日志,看看网站的访问量。看看有没有黑阔搞破坏！于是收集，整理一些服务器日志分析命令，大家可以试试！

1、查看有多少个 IP 访问：

```
awk '{print $1}' log_file|sort|uniq|wc -l
```

2、查看某一个页面被访问的次数：

```
grep "/index.php" log_file | wc -l
```

3、查看每一个 IP 访问了多少个页面：

```
awk '{++S[$1]} END {for (a in S) print a,S[a]}' log_file > log.txt
```

```
sort -n -t ' ' -k 2 log.txt
```
配合 sort 进一步排序

4、将每个 IP 访问的页面数进行从小到大排序：

```
awk '{++S[$1]} END {for (a in S) print S[a],a}' log_file | sort -n
```

5、查看某一个 IP 访问了哪些页面：

```
grep ^111.111.111.111 log_file| awk '{print $1,$7}'
```

6、去掉搜索引擎统计的页面：

```
awk '{print $12,$1}' log_file | grep ^\"Mozilla | awk '{print $2}' |sort | uniq | wc -l
```

7、查看 2015 年 8 月 16 日 14 时这一个小时内有多少 IP 访问:

```
awk '{print $4,$1}' log_file | grep 16/Aug/2015:14 | awk '{print $2}'| sort | uniq | wc -l
```

8、查看访问前十个 ip 地址

```
awk '{print $1}' |sort|uniq -c|sort -nr |head -10 access_log
```

uniq -c 相当于分组统计并把统计数放在最前面

```
cat access.log|awk '{print $1}'|sort|uniq -c|sort -nr|head -10
```

```
cat access.log|awk '{counts[$(11)]+=1}; END {for(url in counts) print counts[url], url}
```

9、访问次数最多的 10 个文件或页面

```
cat log_file|awk '{print $11}'|sort|uniq -c|sort -nr | head -10
```

```
cat log_file|awk '{print $11}'|sort|uniq -c|sort -nr|head -20
```

```
awk '{print $1}' log_file |sort -n -r |uniq -c | sort -n -r | head -20
```

访问量最大的前 20 个 ip

10、通过子域名访问次数，依据 referer 来计算，稍有不准

```
cat access.log | awk '{print $11}' | sed -e ' s/http:\/\///' -e ' s/\/.*//' | sort | uniq -c | sort -rn | head
-20
```

11、列出传输大小最大的几个文件

```
cat www.access.log |awk '($7~/\.php/){print $10 " " $1 " " $4 " " $7}'|sort -nr|head -100
```

12、列出输出大于 200000byte(约 200kb)的页面以及对应页面发生次数

```
cat www.access.log |awk '($10 > 200000 && $7~/\.php/){print $7}'|sort -n|uniq -c|sort -nr|head -100
```

13、如果日志最后一列记录的是页面文件传输时间，则有列出到客户端最耗时的页面

```
cat www.access.log |awk '($7~/\.php/){print $NF " " $1 " " $4 " " $7}'|sort -nr|head -100
```

14、列出最最耗时的页面(超过 60 秒的)的以及对应页面发生次数

```
cat www.access.log |awk '($NF > 60 && $7~/\.php/){print $7}'|sort -n|uniq -c|sort -nr|head -100
```

15、列出传输时间超过 30 秒的文件

```
cat www.access.log |awk '($NF > 30){print $7}'|sort -n|uniq -c|sort -nr|head -20
```

16、列出当前服务器每一进程运行的数量，倒序排列

```
ps -ef | awk -F ' ' '{print $8 " " $9}' |sort | uniq -c |sort -nr |head -20
```

17、查看 apache 当前并发访问数

对比 httpd.conf 中 MaxClients 的数字差距多少

```
netstat -an | grep ESTABLISHED | wc -l
```

18、可以使用如下参数查看数据

```
ps -ef|grep httpd|wc -l
```

```
1388
```

统计 httpd 进程数，连个请求会启动一个进程，使用于 Apache 服务器。

表示 Apache 能够处理 1388 个并发请求，这个值 Apache 可根据负载情况自动调整

```
netstat -nat|grep -i "80"|wc -l
```

```
4341
```

netstat -an 会打印系统当前网络链接状态，而 grep -i "80"是用来提取与 80 端口有关的连接的，wc -l 进行连接数统计。

最终返回的数字就是当前所有 80 端口的请求总数

```
netstat -na|grep ESTABLISHED|wc -l
```

```
376
```

netstat -an 会打印系统当前网络链接状态，而 grep ESTABLISHED 提取出已建立连接的信息。 然后 wc -l 统计

最终返回的数字就是当前所有 80 端口的已建立连接的总数。

```
netstat -nat||grep ESTABLISHED|wc
```

可查看所有建立连接的详细记录

19、输出每个 ip 的连接数，以及总的各个状态的连接数

```
netstat -n | awk '/^tcp/ {n=split($(NF-1),array,":");if(n<=2)++S[array[(1)]];else++S[array[(4)]];++s[$NF];++N}
END {for(a in S){printf("%-20s %s\n", a, S[a]);++I}printf("%-20s %s\n","TOTAL_IP",I);for(a in s)
printf("%-20s %s\n",a, s[a]);printf("%-20s %s\n","TOTAL_LINK",N);}'
```

20、其他的收集

分析日志文件下 2012-05-04 访问页面最高 的前 20 个 URL 并排序

```
cat access.log |grep '04/May/2012'| awk '{print $11}'|sort|uniq -c|sort -nr|head -20
```

查询受访问页面的 URL 地址中 含有 www.abc.com 网址的 IP 地址

```
cat access_log | awk '($11~/\www.abc.com/){print $1}'|sort|uniq -c|sort -nr
```

获取访问最高的 10 个 IP 地址 同时也可以按时间来查询

```
cat linewow-access.log|awk '{print $1}'|sort|uniq -c|sort -nr|head -10
```

时间段查询日志时间段的情况

```
cat log_file | egrep '15/Aug/2015|16/Aug/2015' |awk '{print $1}'|sort|uniq -c|sort -nr|head -10
```

分析 2015/8/15 到 2015/8/16 访问"/index.php?g=Member&m=Public&a=sendValidCode"的 IP 倒序排列

```
cat log_file | egrep '15/Aug/2015|16/Aug/2015' | awk '{if($7 == "/index.php?g=Member&m=Public&a=sendValidCode")
print $1,$7}'|sort|uniq -c|sort -nr
```

($7~/.php/) $7 里面包含.php 的就输出,本句的意思是最耗时的一百个 PHP 页面

```
cat log_file |awk '($7~/\.php/){print $NF " " $1 " " $4 " " $7}'|sort -nr|head -100
```

列出最最耗时的页面(超过 60 秒的)的以及对应页面发生次数

```
cat access.log |awk '($NF > 60 && $7~/\.php/){print $7}'|sort -n|uniq -c|sort -nr|head -100
```

统计网站流量 (G)

```
cat access.log |awk '{sum+=$10} END {print sum/1024/1024/1024}'
```

统计 404 的连接

```
awk '($9 ~/404/)' access.log | awk '{print $9,$7}' | sort
```

统计 http status

```
cat access.log |awk '{counts[$(9)]+=1}; END {for(code in counts) print code, counts[code]}'
cat access.log |awk '{print $9}'|sort|uniq -c|sort -rn
```

每秒并发

```
watch "awk '{if($9~/200|30|404/)COUNT[$4]++}END{for( a in COUNT) print a,COUNT[a]}' log_file|sort -k 2 -nr|head
-n10"
```

带宽统计

```
cat apache.log |awk '{if($7~/GET/) count++}END{print "client_request="count}'
cat apache.log |awk '{BYTE+=$11}END{print "client_kbyte_out="BYTE/1024"KB"}'
```

找出某天访问次数最多的 10 个 IP

```
cat /tmp/access.log | grep "20/Mar/2011" |awk '{print $3}'|sort |uniq -c|sort -nr|head
```

当天 ip 连接数最高的 ip 都在干些什么

```
cat access.log | grep "10.0.21.17" | awk '{print $8}' | sort | uniq -c | sort -nr | head -n 10
```

小时单位里 ip 连接数最多的 10 个时段

```
awk -vFS="[:]" '{gsub("-.*","",$1);num[$2" "$1]++}END{for(i in num)print i,num[i]}' log_file | sort -n -k 3 -r
| head -10
```

找出访问次数最多的几个分钟

```
awk '{print $1}' access.log | grep "20/Mar/2011" |cut -c 14-18|sort|uniq -c|sort -nr|head
```

取 5 分钟日志

```
if [ $DATE_MINUTE != $DATE_END_MINUTE ] ;then #则判断开始时间戳与结束时间戳是否相等
START_LINE=sed -n "/$DATE_MINUTE/=" $APACHE_LOG|head -n1 #如果不相等，则取出开始时间戳的行号，与结束时间戳的行
号
```

查看 tcp 的链接状态

```
netstat -nat |awk '{print $6}'|sort|uniq -c|sort -rn

netstat -n | awk '/^tcp/ {++S[$NF]};END {for(a in S) print a, S[a]}'

netstat -n | awk '/^tcp/ {++state[$NF]}; END {for(key in state) print key,"\t",state[key]}'

netstat -n | awk '/^tcp/ {++arr[$NF]};END {for(k in arr) print k,"\t",arr[k]}'

netstat -n |awk '/^tcp/ {print $NF}'|sort|uniq -c|sort -rn

netstat -ant | awk '{print $NF}' | grep -v '[a-z]' | sort | uniq -c

netstat -ant|awk '/ip:80/{split($5,ip,":");++S[ip[1]]}END{for (a in S) print S[a],a}' |sort -n

netstat -ant|awk '/:80/{split($5,ip,":");++S[ip[1]]}END{for (a in S) print S[a],a}' |sort -rn|head -n 10
```

```
awk 'BEGIN{printf ("http_code\tcount_num\n")}{COUNT[$10]++}END{for (a in COUNT) printf a"\t\t"COUNT[a]"\n"}'
```

查找请求数前 20 个 IP（常用于查找攻来源）：

```
netstat -anlp|grep 80|grep tcp|awk '{print $5}'|awk -F: '{print $1}'|sort|uniq -c|sort -nr|head -n20
netstat -ant |awk '/:80/{split($5,ip,":");++A[ip[1]]}END{for(i in A) print A[i],i}' |sort -rn|head -n20
```

用 tcpdump 嗅探 80 端口的访问看看谁最高

```
tcpdump -i eth0 -tnn dst port 80 -c 1000 | awk -F"." '{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -nr
|head -20
```

查找较多 time_wait 连接

```
netstat -n|grep TIME_WAIT|awk '{print $5}'|sort|uniq -c|sort -rn|head -n20
```

找查较多的 SYN 连接

```
netstat -an | grep SYN | awk '{print $5}' | awk -F: '{print $1}' | sort | uniq -c | sort -nr | more
```

根据端口列进程

```
netstat -ntlp | grep 80 | awk '{print $7}' | cut -d/ -f1
```

查看了连接数和当前的连接数

```
netstat -ant | grep $ip:80 | wc -l
netstat -ant | grep $ip:80 | grep EST | wc -l
```

查看 IP 访问次数

```
netstat -nat|grep ":80"|awk '{print $5}' |awk -F: '{print $1}' | sort| uniq -c|sort -n
```

Linux 命令分析当前的链接状况

```
netstat -n | awk '/^tcp/ {++S[$NF]} END {for(a in S) print a, S[a]}'
```

```
watch "netstat -n | awk '/^tcp/ {++S[\$NF]} END {for(a in S) print a, S[a]}'" # 通过 watch 可以一直监控
```

```
LAST_ACK 5 #关闭一个 TCP 连接需要从两个方向上分别进行关闭，双方都是通过发送 FIN 来表示单方向数据的关闭，当通信双方发送了最后一个 FIN 的时候，发送方此时处于 LAST_ACK 状态，当发送方收到对方的确认（Fin 的 Ack 确认）后才真正关闭整个 TCP 连接；
```

```
SYN_RECV 30  # 表示正在等待处理的请求数；
```

```
ESTABLISHED 1597 # 表示正常数据传输状态；
```

```
FIN_WAIT1 51 # 表示 server 端主动要求关闭 tcp 连接；

FIN_WAIT2 504 # 表示客户端中断连接；

TIME_WAIT 1057  # 表示处理完毕，等待超时结束的请求数；
```