這些Shell 分析服務器日誌命令集錦，收藏好了~

良許Linux 今天

> 作者：Panda
>
> 鏈接：https://segmentfault.com/a/1190000009745139

自己的小網站跑在阿里雲的ECS上面,偶爾也去分析分析自己網站服務器日誌,看看網站的訪問量。看看有沒有黑闊搞破壞！於是收集，整理一些服務器日誌分析命令，大家可以試試！

1、查看有多少個IP訪問：

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

2、查看某一個頁面被訪問的次數：

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

3、查看每一個IP訪問了多少個頁面：

```
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、將每個IP訪問的頁面數進行從小到大排序：

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

5、查看某一個IP訪問了哪些頁面：

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

6、去掉搜索引擎統計的頁面：

```
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時這一個小時內有多少IP訪問:

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

8、查看訪問前十個ip地址

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

uniq -c 相當於分組統計並把統計數放在最前面

```
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、訪問次數最多的10個文件或頁面

```
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
```

訪問量最大的前20個ip

10、通過子域名訪問次數，依據referer來計算，稍有不准

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

11、列出傳輸大小最大的幾個文件

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

12、列出輸出大於200000byte(約200kb)的頁面以及對應頁面發生次數

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

13、如果日誌最後一列記錄的是頁面文件傳輸時間，則有列出到客戶端最耗時的頁面

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

14、列出最最耗時的頁面(超過60秒的)的以及對應頁面發生次數

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

15、列出傳輸時間超過30 秒的文件

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

16、列出當前服務器每一進程運行的數量，倒序排列

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

17、查看apache當前並發訪問數

對比httpd.conf中MaxClients的數字差距多少

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

18、可以使用如下參數查看數據

```
ps -ef|grep httpd|wc -l
1388
```

統計httpd進程數，連個請求會啟動一個進程，使用於Apache服務器。
表示Apache能夠處理1388個並發請求，這個值Apache可根據負載情況自動調整

```
netstat -nat|grep -i "80"|wc -l
4341
```

netstat -an會打印系統當前網絡鏈接狀態，而grep -i "80"是用來提取與80端口有關的連接的，wc -l進行連接數統計。
最終返回的數字就是當前所有80端口的請求總數

```
netstat -na|grep ESTABLISHED|wc -l
376
```

netstat -an會打印系統當前網絡鏈接狀態，而grep ESTABLISHED 提取出已建立連接的信息。然後wc -l統計最終返回的數字就是當前所有80端口的已建立連接的總數。

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

可查看所有建立連接的詳細記錄

19、輸出每個ip的連接數，以及總的各個狀態的連接數

```
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
```

20、其他的收集

分析日誌文件下2012-05-04 訪問頁面最高的前20個URL 並排序

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

查詢受訪問頁面的URL地址中含有www.abc.com 網址的IP 地址

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

獲取訪問最高的10個IP地址同時也可以按時間來查詢

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

時間段查詢日誌時間段的情況

```
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 訪問"/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|uni
```

($7~/.php/) $7裡麵包含.php的就輸出,本句的意思是最耗時的一百個PHP頁面

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

列出最最耗時的頁面(超過60秒的)的以及對應頁面發生次數

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

統計網站流量（G)

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

統計404的連接

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

統計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
```

每秒並發

```
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"
```

帶寬統計

```
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"}'
```

找出某天訪問次數最多的10個IP

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

當天ip連接數最高的ip都在幹些什麼

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

小時單位裡ip連接數最多的10個時段

```
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
```

找出訪問次數最多的幾個分鐘

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

取5分鐘日誌

```
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的鏈接狀態

```
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,"",state[key]}'

netstat -n | awk '/^tcp/ {++arr[$NF]};END {for(k in arr) print k,"",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_codecount_num")}{COUNT[$10]++}END{for (a in COUNT) printf a""COUNT[a]""}'
```

查找請求數前20個IP（常用於查找攻來源）：

```
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端口的訪問看看誰最高

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

查找較多time_wait連接

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

找查較多的SYN連接

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

根據端口列進程

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

查看了連接數和當前的連接數

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

查看IP訪問次數

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

Linux命令分析當前的鏈接狀況

```
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LAS

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；
```

閱讀原文

喜歡此內容的人還喜歡

高並發場景下，到底先更新緩存還是先更新數據庫?

高性能服務器開發

Redis 為什麼默認16 個數據庫?

Java後端

為什麼不建議把數據庫部署在docker容器內?

JavaCat