nginx 日志分析及性能排查

[收藏](javascript:void(0);)

[-外星人-](https://my.oschina.net/362228416/home)

* 发表于 3个月前

* 阅读 3046

* 收藏 486

* 点赞 15

* [评论 18](https://my.oschina.net/362228416/blog/844713#comment-list)

[**“粉丝福利”\_IT在线，精品课程讲解！>>>**](http://web.tanzhouedu.com/index/osc.html)  https://my.oschina.net/img/hot3.png

*摘要: nginx 日志分析及性能排查*

最近一直在做性能排查，思路就是根据分析nginx日志，得到响应耗时的url、以及请求时间，再得到这段时间的请求量，并发量，分析是并发的原因，还是本身就比较慢，如果是应用本身的原因，只需要找到对应的代码，然后进行优化就好了

我找到的几个原因，基本就是后端sql运行的比较多，单次访问看不出来，但是人比较多的时候就比较慢了，人少的时候20-200毫秒，人多的时候，200-6000毫秒，优化之后基本保持在几十毫秒，优化策略就是减少不必要的sql，加上缓存，基本解决了卡顿的问题，顺便把这次用的一系列命令记录下来，当个总结吧

如果需要得到请求处理的时间，需要在nginx log 里面加上$request\_time，下面是我的log\_format

nginx.conf

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent $request\_body "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for" "$request\_time"';

修改之后重启nginx，查看nginx log的时候，就能看到nginx处理请求所花的时间了，这个时间基本就是后端所花的时间，所以可以根据这个字段来得到响应慢的请求

以下是就是我用到的一些命令了

获取pv数

$ cat /usr/local/nginx/logs/access.log | wc -l

获取ip数

$ cat /usr/local/nginx/logs/access.log | awk '{print $1}' | sort -k1 -r | uniq | wc -l

获取最耗时的请求时间、url、耗时，前10名, 可以修改后面的数字获取更多，不加则获取全部

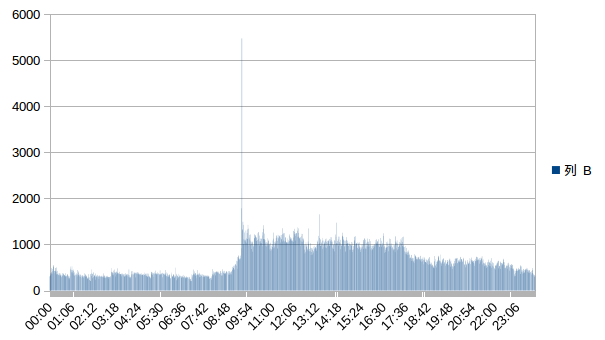
$ cat /usr/local/class/logs/access.log | awk '{print $4,$7,$NF}' | awk -F '"' '{print $1,$2,$3}' | sort -k3 -rn | head -10

获取某一时刻的请求数量，可以把秒去掉得到分钟的数据，把分钟去掉得到小时的数据，以此类推

$ cat /usr/local/class/logs/access.log | grep 2017:13:28:55 | wc -l

获取每分钟的请求数量，输出成csv文件，然后用excel打开，可以生成柱状图

$ cat /usr/local/class/logs/access.log | awk '{print substr($4,14,5)}' | uniq -c | awk '{print $2","$1}' > access.csv

 上面的图是用excel生成的，也可以用命令行工具gnuplot生成png，我也试了一下，没什么问题，直接以编程的形式得到报表，去掉人工操作部分，很方便，但是有一点就是x轴数据比较多的时候，不能像excel一样自动稀释数据，所以我还是喜欢用excel来生成

其实用来用去也就是那么几个命令:

cat：输入文件内容

grep：过滤文本

'sort'：排序

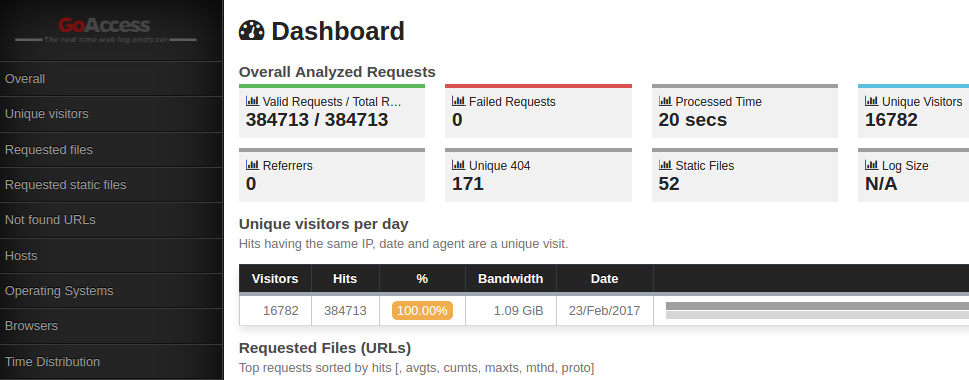
'uniq'：去重

'awk'：文本处理

命令组合使用，单个命令可以使用多次，来达到多重过滤的效果，前面一个命令的输出就是后一个命令的输入，流式处理，只要学会这个命令，有多看似复杂的东西，都变得异常简单。

上面介绍的都是命令，下面再介绍一个直接输出html的，其实就是利用go-access来分析nginx日志

cat /usr/local/nginx/logs/access.log | docker run --rm -i diyan/goaccess --time-format='%H:%M:%S' --date-format='%d/%b/%Y' --log-format='%h %^[%d:%t %^] "%r" %s %b "%R" "%u"' > index.html



go-access是以docker容器的形式运行的，只要你安装了docker，就能直接运行，免安装很方便

以上脚本，配合日志每天的日志分割，然后在crontab里面配置一下自动运行脚本，可生成每一天的nginx报表，网站情况一幕了然，当然这里也有缺点，因为不实时

想要统计实时数据，可以使用ngxtop 来查看，安装起来也很简单

$ pip install ngxtop

运行的话，先进到nginx目录，然后再运行，-c 指定配置文件，-t 刷新频率，单位为秒

$ cd /usr/local/nginx

$ ngxtop -c conf/nginx.conf -t 1

但是这种实时的方式，还需要ssh远程登录，不太方便，还可以使用lua来进行实时统计，然后写一个界面把数据展示出来，通过[lua-nginx-module](https://github.com/openresty/lua-nginx-module)，nginx/tengine 都可以用，如果直接安装openresty的话，就方便了，内嵌了lua，不需要重新编译nginx了，openresty的入门可以看我写的一些[文章](https://github.com/362228416/openresty-web-dev)

以上～～～