[**纯手工玩转 Nginx 日志**](https://blog.eood.cn/nginx_logs)

Posted under [其他](https://blog.eood.cn/category/other) by [Bruce Dou](http://blog.eood.cn/about) | Posted 09 Jul 2014

[](http://blog.eood.cn/nginx_logs)**Nginx 日志对于大部分人来说是个未被发掘的宝藏，总结之前做某日志分析系统的经验，和大家分享一下 Nginx 日志的纯手工分析方式。**

Nginx 日志相关配置有 2 个地方：access\_log 和 log\_format 。

**默认的格式：**

access\_log /data/logs/nginx-access.log;

log\_format old ***'$remote\_addr [$time\_local] $status $request\_time $body\_bytes\_sent '***

***'"$request" "$http\_referer" "$http\_user\_agent"'***;

相信大部分用过 Nginx 的人对默认 Nginx 日志格式配置都很熟悉，对日志的内容也很熟悉。但是默认配置和格式虽然可读，但是难以计算。

**Nginx 日志刷盘相关策略可配置：**

比如，设置 buffer，buffer 满 32k 才刷盘；假如 buffer 不满 5s 钟强制刷盘的配置如下：

access\_log /data/logs/nginx-access.log buffer=***32k*** flush=***5s***;

这决定了是否实时看到日志以及日志对磁盘 IO 的影响。

**Nginx 日志能够记录的变量还有很多没出现在默认配置中：**

比如：

请求数据大小：$request\_length  
返回数据大小：$bytes\_sent  
请求耗时：$request\_time  
所用连接序号：$connection  
当前连接发生请求数：$connection\_requests

Nginx 的默认格式不可计算，需要想办法转换成可计算格式，比如用控制字符 ^A (Mac 下 ctrl+v ctrl+a 打出)分割每个字段。

**log\_format 的格式可以变成这样：**

log\_format new ***'$remote\_addr^A$http\_x\_forwarded\_for^A$host^A$time\_local^A$status^A'***

***'$request\_time^A$request\_length^A$bytes\_sent^A$http\_referer^A$request^A$http\_user\_agent'***;

**这样之后就通过常见的 Linux 命令行工具进行分析了：**

* 查找访问频率最高的 URL 和次数：

cat access.log | awk -F ‘^A’ ‘{print $10}’ | sort | uniq -c

* 查找当前日志文件 500 错误的访问：

cat access.log | awk -F ‘^A’ ‘{if($5 == 500) print $0}’

* 查找当前日志文件 500 错误的数量：

cat access.log | awk -F ‘^A’ ‘{if($5 == 500) print $0}’ | wc -l

* 查找某一分钟内 500 错误访问的数量:

cat access.log | awk -F ‘^A’ ‘{if($5 == 500) print $0}’ | grep ’09:00’ | wc-l

* 查找耗时超过 1s 的慢请求：

tail -f access.log | awk -F ‘^A’ ‘{if($6>1) print $0}’

* 假如只想查看某些位：

tail -f access.log | awk -F ‘^A’ ‘{if($6>1) print $3″|”$4}’

* 查找 502 错误最多的 URL：

cat access.log | awk -F ‘^A’ ‘{if($5==502) print $11}’ | sort | uniq -c

* 查找 200 空白页

cat access.log | awk -F ‘^A’ ‘{if($5==200 && $8 < 100) print $3″|”$4″|”$11″|”$6}’

* 查看实时日志数据流

tail -f access.log | cat -e

或者

tail -f access.log | tr ‘^A’ ‘|’

**总结**

照着这个思路可以做很多其他分析，比如 UA 最多的访问；访问频率最高的 IP；请求耗时分析；请求返回包大小分析；等等。

这就是一个大型 Web 日志分析系统的原型，这样的格式也是非常方便进行后续大规模 batching 和 streaming 计算。