## 大型网站日志分析项目

## 概述

随着互联网的发展，在互联网上产生了大量的Web日志或者移动应用日志，日志包含用户的重要信息，通过日志分析，用户可以获取到网站或者应用的访问量，那个页面的访问人数最多，那个网页最有价值，用户特征用户兴趣等。

一般中型网站（10W的pv以上），每天会产生1G以上的web日志文件。大型或者超大型网站，一天的日至数量在500G~1T的数据量。

Web日志由web服务器产生，现在的互联网公司主流的服务器包括nginx、apache、tomcat等。从web日志中，我们可以获取网站每类页面的PV、UV，更复杂一些的，可以计算得出用户所检索的关键词排行榜、用户停留时间最高的页面等。更为复杂的，构建广告点击模型、分析用户行为特征等。

1. 日志格式

1.客户端（用户）IP地址。如：上例中的 201.158.69.116

2.访问时间。如：上例中的 [03/Jan/2013:21:17:20 -0600]

3.访问端口。如：上例中的 127.0.0.1:9000

4.响应时间。如：上例中的 0.007

5.请求时间。如：上例中的 0.007

6.用户地理位置代码（国家代码）。如：上例中的 MX（墨西哥）

7.请求的url地址（目标url地址）的host。如：上例中的 pythontab.com

8.请求方式（GET或者POST等）。如：上例中的 GET

9.请求url地址（去除host部分）。如：上例中的 /html/test.html

10.请求状态（状态码，200表示成功，404表示页面不存在，301表示永久重定向等，具体状态码可以在网上找相关文章，不再赘述）。如：上例中的 "200"

11.请求页面大小，默认为B（byte）。如：上例中的 2426

12.来源页面，即从哪个页面转到本页，专业名称叫做“referer”。如：上例中的 "http://a.com"

13.用户浏览器语言。如：上例中的 "es-ES,es;q=0.8"

14.用户浏览器其他信息，浏览器版本、浏览器类型等。如：上例中的  "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1271.97 Safari/537.11"

1. 日志格式是可以设置的，在nginx的nginx.conf配置文件找到：log\_format 这里就是日志的格式

#access日志格式配置，具体参数不再细说，上面都已经说过了，自己对应一下即可

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

'fwf[$http\_x\_forwarded\_for] tip[$http\_true\_client\_ip] '

'$upstream\_addr $upstream\_response\_time $request\_time '

'$geoip\_country\_code '

'$http\_host $request '

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

'"$http\_accept\_language" "$http\_user\_agent" ';

#配置access log日志的存储位置及文件，注意：access.log文件是可以按日期进行分割的，方便查看及处理

access\_log /home/serversoft/nginx/log/access.log main;

在apache的配置文件httpd.conf中，有一行是这么配置的：

CustomLog "logs/access.log" common

指定了日志的目录路径./logs/，指定了日志的格式，默认common。

common格式是通过：LogFormat "%h %l %u %t \"%r\" %>s %b" common定义的。

双引号里面的格式字符串代表特定的信息。

apache格式字符串及其含义：

%%    百分号(Apache2.0.44或更高的版本)  
%a    远端IP地址  
%A    本机IP地址  
%B    除HTTP头以外传送的字节数  
%b    以CLF格式显示的除HTTP头以外传送的字节数，也就是当没有字节传送时显示'-'而不是0。  
%{Foobar}C    在请求中传送给服务端的cookieFoobar的内容。  
%D    服务器处理本请求所用时间，以微为单位。  
%{FOOBAR}e    环境变量FOOBAR的值  
%f    文件名  
%h    远端主机  
%H    请求使用的协议  
%{Foobar}i    发送到服务器的请求头Foobar:的内容。  
%l    远端登录名(由identd而来，如果支持的话)，除非IdentityCheck设为"On"，否则将得到一个"-"。  
%m    请求的方法  
%{Foobar}n    来自另一个模块的注解Foobar的内容。  
%{Foobar}o    应答头Foobar:的内容。  
%p    服务器服务于该请求的标准端口。  
%P    为本请求提供服务的子进程的PID。  
%{format}P 服务于该请求的PID或TID(线程ID)，format的取值范围为：pid和tid(2.0.46及以后版本)以及hextid(需要APR1.2.0及以上版本)  
%q    查询字符串(若存在则由一个"?"引导，否则返回空串)  
%r    请求的第一行  
%s    状态。对于内部重定向的请求，这个状态指的是原始请求的状态，---%>s则指的是最后请求的状态。  
%t    时间，用普通日志时间格式(标准英语格式)  
%{format}t    时间，用strftime(3)指定的格式表示的时间。(默认情况下按本地化格式)  
%T    处理完请求所花时间，以秒为单位。  
%u    远程用户名(根据验证信息而来；如果返回status(%s)为401，可能是假的)  
%U    请求的URL路径，不包含查询字符串。  
%v    对该请求提供服务的标准ServerName。  
%V    根据UseCanonicalName指令设定的服务器名称。  
</span><span style="font-size:18px;">

在tomcat的server.xml文件中，host主机配置区域找到类似如下(红色部分)即为访问日志的配置：

 <Host name="localhost"  appBase="webapps"  
            unpackWARs="true" autoDeploy="true">

       <!--...部分内容略..-->  
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"  
               prefix="localhost\_access\_log." suffix=".txt"  
               pattern="%h %l %u %t &quot;%r&quot; %s %b" />  
      </Host>

其中的directory用于指定日志的存放路径，默认位于tomcat的logs目录中，例如我们可以修改成：

directory="c:/wwwlogs" 使日志放到c:\wwwlogs目录中去。

其中的prefix和suffic分别用于指定日志文件的前缀和后缀，不用我多说。

现在我们主要来看一下pattern配置段，它用于指定日志的输出格式。有效的日志格式模式可以参见下面内容，如下字符串，其对应的信息由指定的响应内容取代：  
  
    ％a - 远程IP地址  
    ％A - 本地IP地址  
    ％b - 发送的字节数，不包括HTTP头，或“ - ”如果没有发送字节  
    ％B - 发送的字节数，不包括HTTP头  
    ％h - 远程主机名  
    ％H - 请求协议  
    ％l (小写的L)- 远程逻辑从identd的用户名（总是返回' - '）  
    ％m - 请求方法  
    ％p - 本地端口  
    ％q - 查询字符串（在前面加上一个“？”如果它存在，否则是一个空字符串  
    ％r - 第一行的要求  
    ％s - 响应的HTTP状态代码  
    ％S - 用户会话ID  
    ％t - 日期和时间，在通用日志格式

    ％u - 远程用户身份验证  
    ％U - 请求的URL路径  
    ％v - 本地服务器名  
    ％D - 处理请求的时间（以毫秒为单位）  
    ％T - 处理请求的时间（以秒为单位）  
    ％I （大写的i） - 当前请求的线程名称  
  
此外，您可以指定以下别名来设置为普遍使用的模式之一：

    common - %h %l %u %t "%r" %s %b  
    combined - %h %l %u %t "%r" %s %b "%{Referer}i" "%{User-Agent}i"   
  
另外，还可以将request请求的查询参数、session会话变量值、cookie值或HTTP请求/响应头内容的变量值等内容写入到日志文件。  
它仿照了apache的语法：  
  
    ％{XXX}i xxx代表传入的头(HTTP Request)

    ％{XXX}o xxx代表传出​​的响应头(Http Resonse)  
    ％{XXX}c  xxx代表特定的Cookie名  
    ％{XXX}r  xxx代表ServletRequest属性名

    ％{XXX}s xxx代表HttpSession中的属性名

**1、查看有多少个IP访问：**

**2、查看某一个页面被访问的次数：**

**3、查看每一个IP访问了多少个页面：**

**4、将每个IP访问的页面数进行从小到大排序：**

**5、查看某一个IP访问了哪些页面：**

**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