用Hive分析nginx日志

这里用到的nginx日志是网站的访问日志，比如：

180.173.250.74 - - [08/Jan/2015:12:38:08 +0800] "GET /avatar/xxx.png HTTP/1.1" 200 968 "/archives/994"
"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/34.0.1847.131 Safari/537.36"

这条日志里面有九列（为了展示的美观，我在里面加入了换行符），每列之间是用空格分割的，每列的含义分别是客户端访问IP、用户标识、用户、访问时间、请求页面、请求状态、返回文件的大小、跳转来源、浏览器UA。如果想用一般的方法来解析这条日志的话，有点困难。但是我们可以如果我们用正则表达式去匹配这九列的话还是很简单的：

([^]*) ([^]*) ([^]*) (₩[.*₩]) (₩".*?₩") (-|[0-9]*) (-|[0-9]*) (₩".*?₩") (₩".*?₩")

这样就可以匹配出每一列的值。而在Hive中我们是可以指定输入文件的解析器(SerDe)的，并且在Hive中内置了一个org.apache.hadoop.hive.contrib.serde2.RegexSerDe正则解析器，我们可以直接使用它。所以整个的建表语句可以这么写

```
CREATE TABLE logs(
  host STRING,
  identity STRING,
  user STRING,
  time STRING,
  request STRING,
  status STRING,
  size STRING,
  referer STRING,
  agent STRING)
ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  "input.regex" = "([^ ]*) ([^ ]*) ([^ ]*) (₩₩[.*₩₩]) (₩".*?₩") (-|[0-9]*)
            (-|[0-9]*) (₩".*?₩") (₩".*?₩")",
  "output.format.string" = "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s %9$s"
)
STORED AS TEXTFILE;
```

将日志放置到这个表的目录下，gz格式的和未压缩的格式都可以直接被Hive解析。所以我可以用下面已经查询每小时的访问量超过20的IP：

```
hive> select substring(time, 2, 14) date ,host, count(*) as count
from logs
group by substring(time, 2, 14), host
having count > 20
sort by date, count;
```

```
29/Dec/2014:00 47.18.236.106 24
29/Dec/2014:02 81.215.34.45 70
29/Dec/2014:04 66.249.64.18 23
29/Dec/2014:04 66.249.64.22 24
29/Dec/2014:09 119.145.14.213 44
29/Dec/2014:09 113.90.78.63 52
29/Dec/2014:10 106.39.255.133 26
29/Dec/2014:10 211.99.9.68 30
29/Dec/2014:10 60.10.71.97 33
29/Dec/2014:10 222.128.29.21 76
29/Dec/2014:11 91.237.69.17 56
29/Dec/2014:11 211.151.238.52 144
29/Dec/2014:12 222.92.189.35 26
29/Dec/2014:12 218.85.130.110 31
29/Dec/2014:12 218.4.189.13 77
29/Dec/2014:13 61.57.231.254 30
29/Dec/2014:13 124.207.11.123 33
29/Dec/2014:14 134.134.139.76 22
29/Dec/2014:14 218.15.33.28 27
29/Dec/2014:14 218.247.17.100 67
29/Dec/2014:15 116.235.244.139 31
29/Dec/2014:15 101.231.119.202 52
29/Dec/2014:15 183.11.249.158 64
29/Dec/2014:16 116.235.244.139 22
29/Dec/2014:16 211.151.238.52 30
29/Dec/2014:16 123.138.184.84 53
29/Dec/2014:17 219.159.77.110 55
29/Dec/2014:17 87.204.102.195 57
29/Dec/2014:17 111.203.3.1 77
29/Dec/2014:18 125.41.147.243 21
29/Dec/2014:18 66.249.64.18 23
29/Dec/2014:18 101.251.230.3 39
29/Dec/2014:18 110.249.70.182 40
```

```
29/Dec/2014:18 91.200.12.26 44
29/Dec/2014:18 218.64.17.230 93
29/Dec/2014:19 66.249.64.22 27
29/Dec/2014:21 222.129.35.102 25
```

或者其他的一些操作。

如果你对Bash比较熟悉的话，你完全可以不用到Hive，可以用到awk、sort等函数来实现，比如我想知道今天访问量比较多的IP并对它进行排序，取前10条的语句可以在这么写：

```
[root@iteblog ]# awk '{print $1}' www.iteblog.com.access.log | sort | uniq -c |
> sort -nr | head -n 10
   241 46.119.121.149
   224 66.249.65.51
   220 66.249.65.49
   219 66.249.65.47
   210 211.151.238.52
   184 207.46.13.96
   183 157.55.39.44
   182 112.247.104.147
   173 157.55.39.239
   169 157.55.39.106
```

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（过往记忆）所有，未经许可不得转载。
本文链接: 【】（）