nginx 日志分析可视化【批处理】

2020年11月3日 21:59 阅读 2659 评论 0

1.nginx 日志收集

```
# 查看日志配置，不知道配置路径的话，可以执行 nginx -t
less /etc/nginx/nginx.conf
```

```
[root@iZbp16e8sau5mdc1x3ajuuZ ~]# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[root@iZbp16e8sau5mdc1x3ajuuZ ~]# less /etc/nginx/nginx.conf
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
```

```
# 查看日志
cd /var/log/nginx;
ll
```

```
[root@iZbp16e8sau5mdclx3ajuuZ nginx]# ll
total 512
-rw-rw-r-- 1 nginx root  96005 Nov  3 15:08 access.log
-rw-rw-r-- 1 nginx root  17779 Oct 25 03:24 access.log-20201025.gz
-rw-rw-r-- 1 nginx root  18451 Oct 26 03:39 access.log-20201026.gz
-rw-rw-r-- 1 nginx root  19853 Oct 27 03:03 access.log-20201027.gz
-rw-rw-r-- 1 nginx root  21519 Oct 28 03:09 access.log-20201028.gz
-rw-rw-r-- 1 nginx root  23320 Oct 29 03:34 access.log-20201029.gz
-rw-rw-r-- 1 nginx root  21905 Oct 30 02:59 access.log-20201030.gz
-rw-rw-r-- 1 nginx root  19005 Oct 31 03:05 access.log-20201031.gz
-rw-rw-r-- 1 nginx root  14555 Nov  1 02:56 access.log-20201101.gz
-rw-rw-r-- 1 nginx root  11395 Nov  2 03:11 access.log-20201102.gz
-rw-rw-r-- 1 nginx root  13067 Nov  3 03:05 access.log-20201103.gz
-rw-rw-r-- 1 nginx root    720 Nov  3 11:05 error.log
-rw-rw-r-- 1 nginx root    281 Oct 24 01:09 error.log-20201024.gz
-rw-rw-r-- 1 nginx root    533 Oct 25 00:54 error.log-20201025.gz
-rw-rw-r-- 1 nginx root    390 Oct 25 22:11 error.log-20201026.gz
-rw-rw-r-- 1 nginx root    415 Oct 27 22:02 error.log-20201028.gz
-rw-rw-r-- 1 nginx root    249 Oct 28 21:58 error.log-20201029.gz
-rw-rw-r-- 1 nginx root    498 Oct 29 22:13 error.log-20201030.gz
-rw-rw-r-- 1 nginx root    359 Oct 30 21:05 error.log-20201031.gz
-rw-rw-r-- 1 nginx root    815 Oct 31 22:12 error.log-20201101.gz
-rw-rw-r-- 1 nginx root    523 Nov  1 22:06 error.log-20201102.gz
-rw-rw-r-- 1 nginx root    218 Nov  2 22:10 error.log-20201103.gz
-rw-r--r-- 1 root  root 175909 Nov  3 15:06 nginx.log.gz
```

正常使用的话，这边应该使用 flume 直接将日志文件收集，并上传到 hdfs。方便起见，这边直接手动打包，上传。

```
# 合并打包日志
cat access.log > nginx.log;
gunzip -c access.log*gz > nginx.log;
gzip nginx.log;
sz nginx.log.gz;
```

2.原始记录加载到ODS层

```
# 上传解压日志
rz;
gunzip nginx.log.gz;
```

```
-- 根据日志格式，使用正则序列化解析，一个（）是一个字段，注意转义
drop table if exists spider.nginx_log;
create table spider.nginx_log(
remote_addr STRING,
remote_user STRING,
time_local STRING,
request STRING,
status STRING,
body_bytes_sent STRING,
http_referer STRING,
http_user_agent STRING
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
"input.regex" = '(.*?) - (.*?) \\[(.*?)\\] "(.*?)" (\\d+) (\\d+) "(.*?)" "(.*?)" .*',
"output.format.string" = "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s"
);

-- 加载数据
load data local inpath '/home/getway/nginx.log' into table spider.nginx_log;
```

```
hive> -- 根据日志格式，使用正则序列化解析，一个（）是一个字段，注意转义
    > drop table if exists spider.nginx_log;
OK
Time taken: 1.785 seconds
hive> create table spider.nginx_log(
    > remote_addr STRING,
    > remote_user STRING,
    > time_local STRING,
    > request STRING,
    > status STRING,
    > body_bytes_sent STRING,
    > http_referer STRING,
    > http_user_agent STRING
    > )
    > ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
    > WITH SERDEPROPERTIES (
    > "input.regex" = '(.*?) - (.*?) \\[(.*?)\\] "(.*?)" (\\d+) (\\d+) "(.*?)" "(.*?)" .*',
    > "output.format.string" = "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s"
    > );
OK
Time taken: 0.194 seconds
hive>
    > -- 加载数据
    > load data local inpath '/home/getway/nginx.log' into table spider.nginx_log;
Loading data to table spider.nginx_log
Table spider.nginx_log stats: [numFiles=1, totalSize=2566030]
OK
Time taken: 0.726 seconds
hive>
```

```
-- 查看一下原始记录
select * from spider.nginx_log limit 10;
```

```
select * from spider.nginx_log limit 10;
```

nginx_log

select * from spider.nginx_log limit 10 输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)

	remote_addr	remote_user	time_local	request	status	body_bytes_sent	http_referer	http_user_agent
1	42.224.250.252	-	24/Oct/2020:03:24:53 +0800	POST /GponForm/diag_Form?images/	404	2116	-	Hello, World
2	42.224.250.252	-	24/Oct/2020:03:24:54 +0800	;sh+/tmp/gpon80&ipv=0	400	157	-	-
3	203.208.60.113	-	24/Oct/2020:03:25:34 +0800	GET /tag/linux/ HTTP/1.1	200	296752	-	Mozilla/5.0 (Linux; Android 6.0.1; Nexu
4	183.136.225.56	-	24/Oct/2020:03:54:01 +0800	GET / HTTP/1.1	200	28795	-	Baiduspider+(+http://www.baidu.com
5	80.82.70.187	-	24/Oct/2020:04:04:29 +0800	GET http://www.baidu.com/cache/glo	404	2116	-	Mozilla
6	95.123.41.94	-	24/Oct/2020:04:44:46 +0800	GET / HTTP/1.1	200	28795	-	Mozilla/5.0 zgrab/0.x
7	203.208.60.18	-	24/Oct/2020:04:45:26 +0800	GET /archive/ HTTP/1.1	200	297769	-	Mozilla/5.0 (compatible; Googlebot/2
8	220.132.235.162	-	24/Oct/2020:04:45:59 +0800	GET / HTTP/1.1	200	28795	-	Mozilla/5.0
9	203.208.60.6	-	24/Oct/2020:04:53:14 +0800	GET /robots.txt HTTP/1.1	404	2116	-	Mozilla/5.0 (compatible; Googlebot/2
10	203.208.60.55	-	24/Oct/2020:04:53:15 +0800	GET /comics/101/chapters/81110 HTT	200	5035	-	Mozilla/5.0 (Linux; Android 6.0.1; Nexu

3.数据清洗加工到DWS层

3.1 加工 IP 省份纬度表

```
-- 通过 udf 获取 ip 所在省份，创建IP和省份的映射表
set hive.exec.mode.local.auto=true;
set hive.exec.mode.local.auto.inputbytes.max=52428800;
set hive.exec.mode.local.auto.input.files.max=10;

add file /home/getway/udf_baidu_api.py;

create table spider.dim_ip as
select transform(remote_addr)
  USING 'python3 udf_baidu_api.py' AS (remote_addr, province)
from (
select remote_addr
from spider.nginx_log
group by remote_addr
) as cte;
```

```
-- 查看一下加工好的数据
select * from spider.dim_ip limit 10;
```

```
select * from spider.dim_ip limit 10;
```

dim_ip

select * from spider.dim_ip limit 10 输入一个 SQL 表达

	remote_addr	province
1	1.163.113.135	台湾
2	1.202.114.155	北京
3	1.246.192.246	ERROR
4	1.63.49.31	黑龙江
5	101.132.111.175	上海
6	101.132.113.110	上海
7	101.132.113.224	上海
8	101.132.134.94	上海
9	101.132.162.197	浙江
10	101.132.175.77	上海

3.2 加工 nginx 日志事实表

```
-- 数据清洗：选择有用的字段，并通过 udf 解析 useragent，将数据写到新表 fact_nginx_log
set hive.exec.mode.local.auto=true;
set hive.exec.mode.local.auto.inputbytes.max=52428800;
set hive.exec.mode.local.auto.input.files.max=10;

add file /home/getway/udf_log_clean.py;

create table spider.fact_nginx_log as
select transform(a.remote_addr,
                 a.time_local,
                 b.province,
                 a.request,
                 a.http_user_agent
                 )
       USING 'python3 udf_log_clean.py' AS (remote_addr,
                                    time_local,
                                    province,
                                    request,
                                    device,
                                    os,
                                    browser)
from spider.nginx_log a
left join spider.dim_ip b on a.remote_addr = b.remote_addr
where a.request not rlike '\\.[css|js|woff|TTF|png|jpg|ico]';
```

```
Automatically selecting local only mode for query
Query ID = getway_20201103155757_29b6c3c4-473f-4182-91a9-49353664c2f9
Total jobs = 3
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=512M; support was removed in 8.0
Execution log at: /tmp/getway/getway_20201103155757_29b6c3c4-473f-4182-91a9-49353664c2f9.log2020-11-03 03:57:42 Starting to laun
to process map join;    maximum memory = 1908932608
2020-11-03 03:57:43     Uploaded 1 File to: file:/tmp/getway/a8b4797f-585a-498e-a489-9ff7b915b3d0/hive_2020-11-03_15-57-40_363_7
5068-1/-local-10004/HashTable-Stage-9/MapJoin-mapfile41--.hashtable (48026 bytes)
2020-11-03 03:57:43     End of local task; Time Taken: 0.555 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2020-11-03 15:57:44,997 Stage-9 map = 0%,  reduce = 0%
2020-11-03 15:57:45,999 Stage-9 map = 100%,  reduce = 0%
Ended Job = job_local933744943_0006
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://nameservice1/tmp/hive/staging/.hive-staging_hive_2020-11-03_15-57-40_363_7616550807227585068-1/-ext-10001
Moving data to: hdfs://nameservice1/user/hive/warehouse/spider.db/fact_nginx_log
Table spider.fact_nginx_log stats: [numFiles=1, numRows=5955, totalSize=682000, rawDataSize=676045]
MapReduce Jobs Launched:
Stage-Stage-9:  HDFS Read: 10526506 HDFS Write: 2617655 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 5.935 seconds
```

```
-- 查看数据示例
select * from spider.fact_nginx_log limit 10;
```

```
select * from spider.fact_nginx_log limit 10;
```

fact_nginx_log

select * from spider.fact_nginx_log limit 10 输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)

	remote_addr	time_local	province	request	device	os	browser
1	42.224.250.252	2020-10-24 03:24:53	河南	POST /GponForm/diag_Form?images/ HTTP/1.1	Other	Other	Other
2	42.224.250.252	2020-10-24 03:24:54	河南	;sh+/tmp/gpon80&ipv=0	Other	Other	Other
3	203.208.60.113	2020-10-24 03:25:34	福建	GET /tag/linux/ HTTP/1.1	Spider	Android 6.0.1	Googlebot 2.1
4	183.136.225.56	2020-10-24 03:54:01	浙江	GET / HTTP/1.1	Spider	Other	bingbot
5	95.123.41.94	2020-10-24 04:44:46	ERROR	GET / HTTP/1.1	Other	Other	Other
6	203.208.60.18	2020-10-24 04:45:26	福建	GET /archive/ HTTP/1.1	Spider	Other	Googlebot 2.1
7	220.132.235.162	2020-10-24 04:45:59	台湾	GET / HTTP/1.1	Other	Other	Other
8	203.208.60.6	2020-10-24 04:53:14	福建	GET /robots.txt HTTP/1.1	Spider	Other	Googlebot 2.1
9	203.208.60.55	2020-10-24 04:53:15	福建	GET /comics/101/chapters/81110 HTTP/1.1	Spider	Android 6.0.1	Googlebot 2.1
10	120.78.225.124	2020-10-24 04:56:48	广东	GET / HTTP/1.0	Other	Other	Other

3.3 数据同步到 mysql

将加工好的表 dim_ip 和 fact_nginx_log，通过 DataX 同步到 mysql。

```
python datax/bin/datax.py fact_nginx_log.json
python datax/bin/datax.py dim_ip.json
```

```
2020-11-03 16:31:04.797 [job-0] INFO  JobContainer - PerfTrace not enable!
2020-11-03 16:31:04.797 [job-0] INFO  StandAloneJobContainerCommunicator - Tota
 Error 6 records, 3756 bytes |  All Task WaitWriterTime 6.612s |  All Task Wai
2020-11-03 16:31:04.799 [job-0] INFO  JobContainer -
任务启动时刻                    : 2020-11-03 16:30:52
任务结束时刻                    : 2020-11-03 16:31:04
任务总计耗时                    :                 11s
任务平均流量                    :           56.50KB/s
记录写入速度                    :            595rec/s
读出记录总数                    :                5955
读写失败总数                    :                   6
```

```
2020-11-03 16:33:05.289 [job-0] INFO  JobContainer - PerfTrace not enable!
2020-11-03 16:33:05.290 [job-0] INFO  StandAloneJobContainerCommunicator - Total 1217 records,
rror 0 records, 0 bytes |  All Task WaitWriterTime 0.001s |  All Task WaitReaderTime 0.040s |
2020-11-03 16:33:05.291 [job-0] INFO  JobContainer -
任务启动时刻                    : 2020-11-03 16:32:53
任务结束时刻                    : 2020-11-03 16:33:05
任务总计耗时                    :                 11s
任务平均流量                    :            1.94KB/s
记录写入速度                    :             121rec/s
读出记录总数                    :                1217
读写失败总数                    :                   0
```

4.数据分析可视化

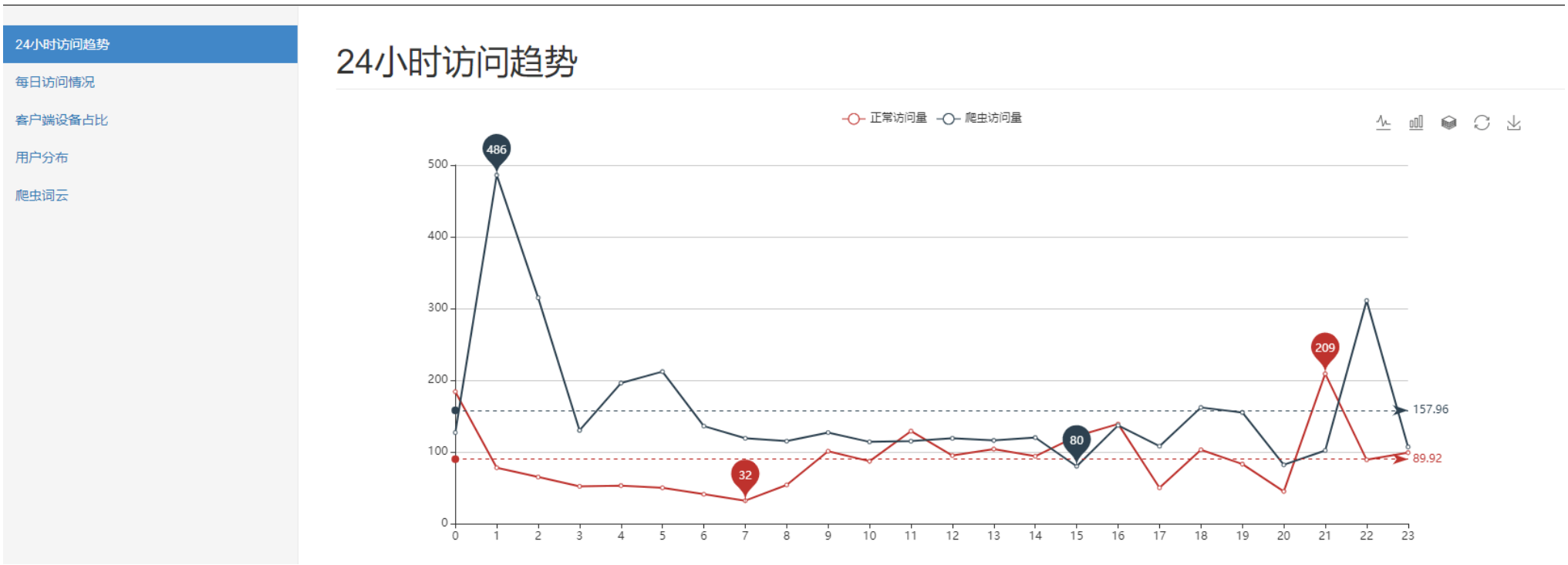
数据都准备好了，分析就很简单，直接通过 sql 就可以查出想要的数据了。

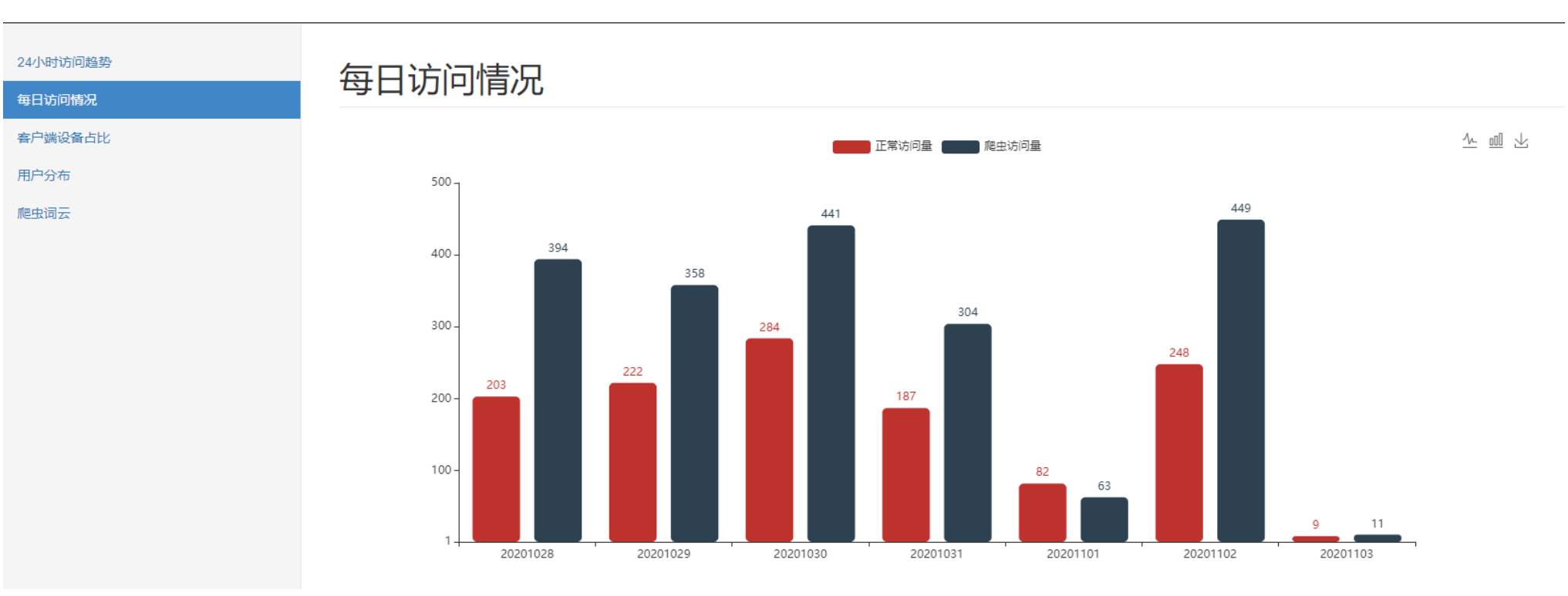
> 数据量比较大时，加上适当的索引可以提高查询效率。

```sql
-- 用户分布
select province, count(distinct remote_addr) from fact_nginx_log group by province;

-- 不同时段访问情况
select case when device='Spider' then 'Spider' else 'Normal' end, hour(time_local), count(1)
from fact_nginx_log
group by case when device='Spider' then 'Spider' else 'Normal' end, hour(time_local);

-- 最近7天访问情况
select case when device='Spider' then 'Spider' else 'Normal' end, DATE_FORMAT(time_local, '%Y%m%d'), count(1)
from fact_nginx_log
where time_local > date_add(CURRENT_DATE, interval - 7 day)
group by case when device='Spider' then 'Spider' else 'Normal' end, DATE_FORMAT(time_local, '%Y%m%d');

-- 用户端前10的设备
select device, count(1)
from fact_nginx_log
where device not in ('Other', 'Spider') -- 过滤掉干扰数据
group by device
order by 2 desc
limit 10

-- 搜索引擎爬虫情况
select browser, count(1) from fact_nginx_log where device = 'Spider' group by browser;
```

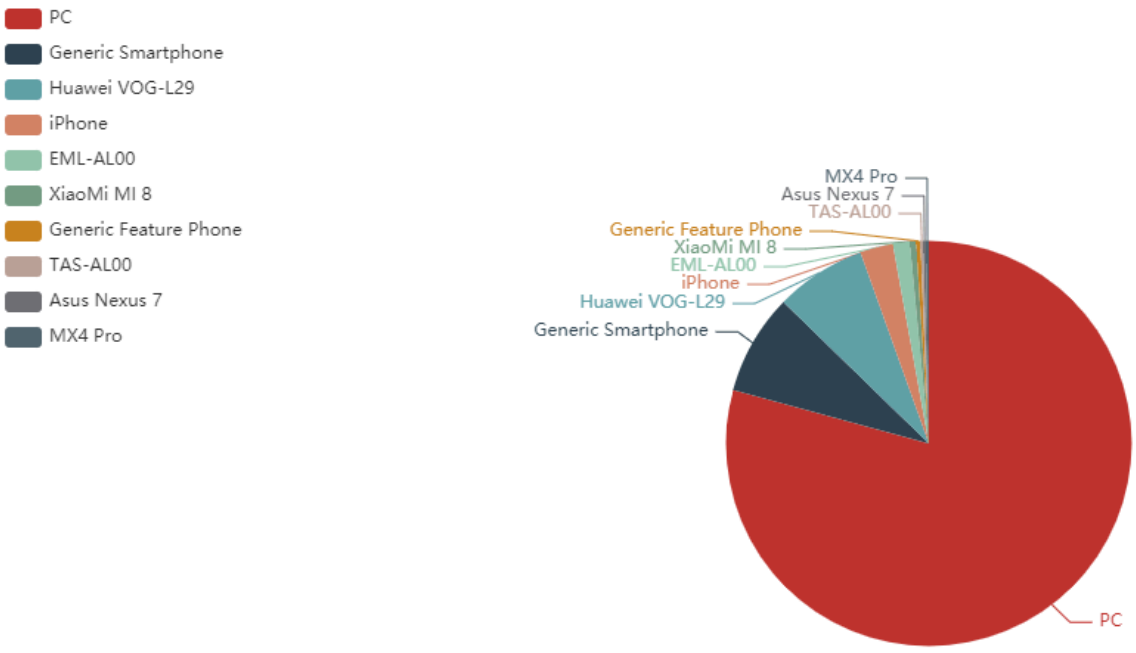
最后，通过 pandas 读取 mysql，经 ironman 进行可视化展示。

> 基于 flask 和 echarts 的数据可视化工具 ironman

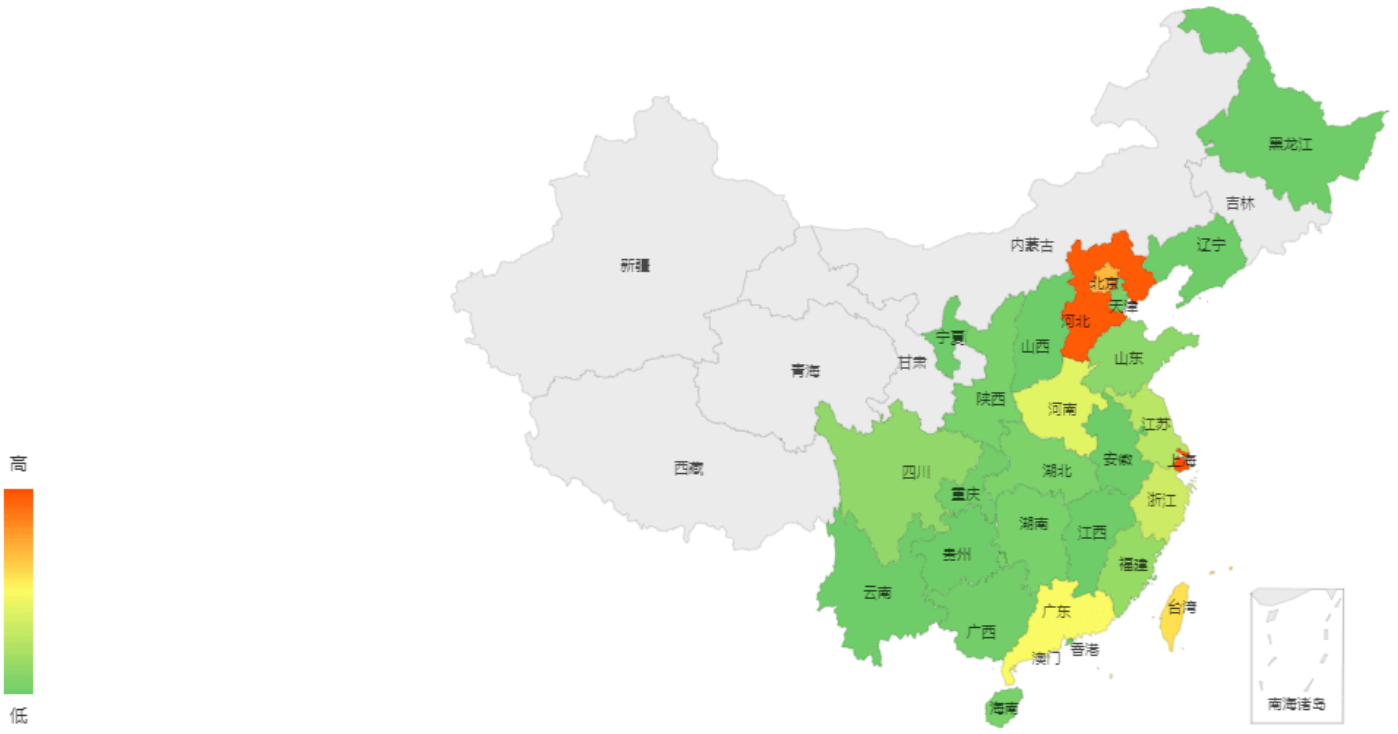




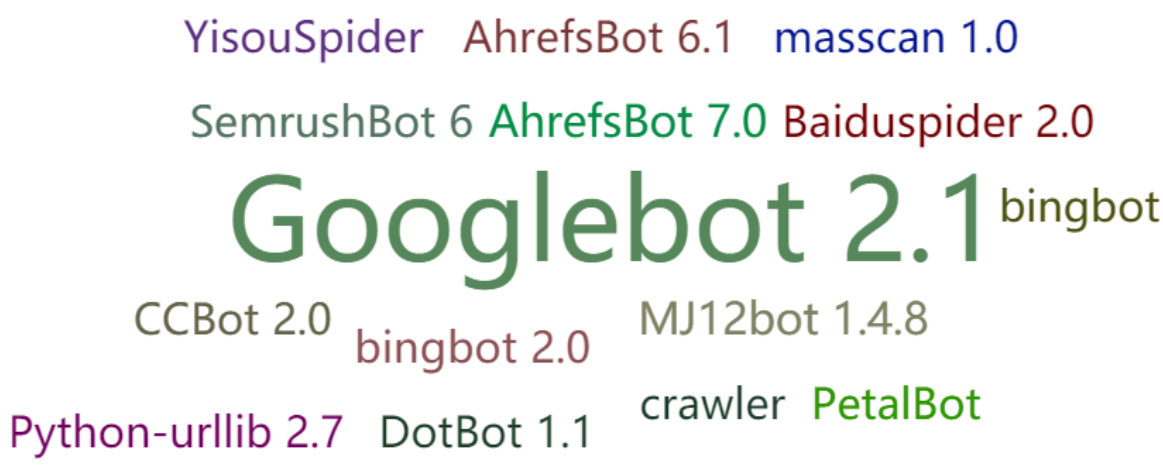
客户端设备占比



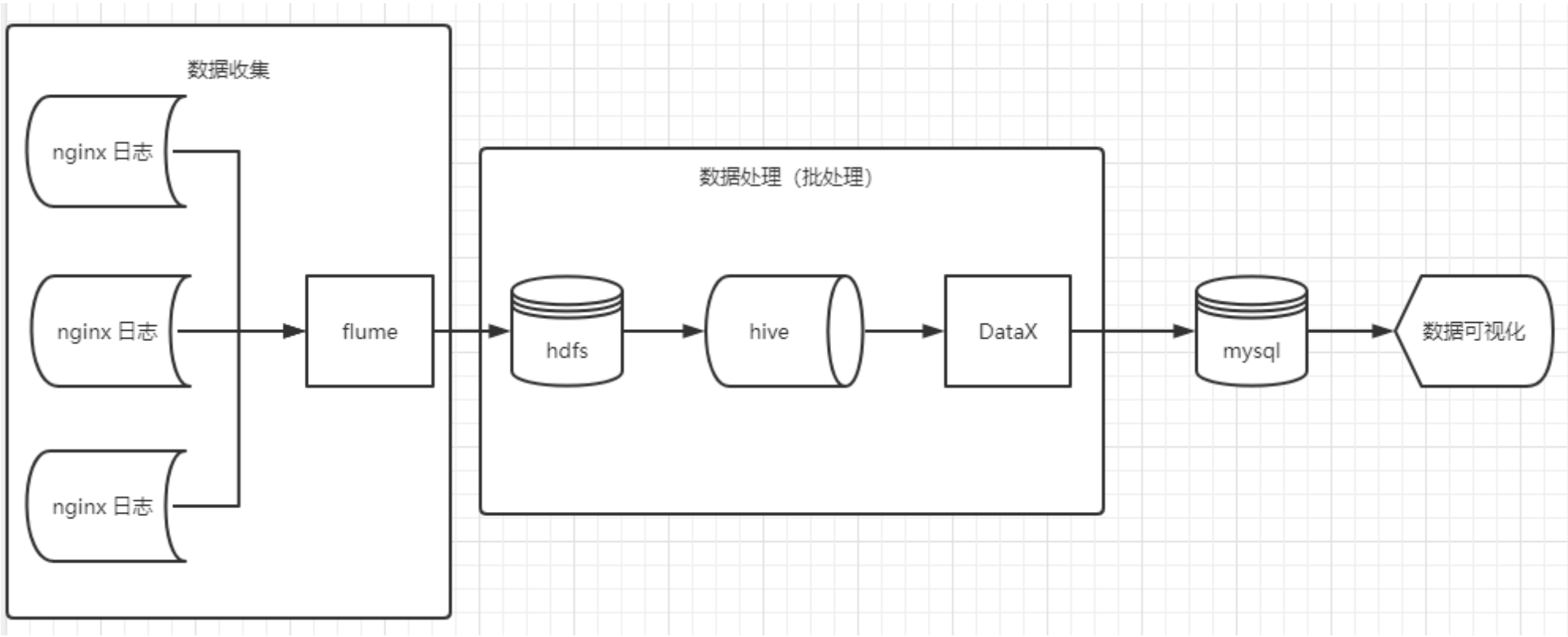
用户分布



爬虫词云



5.总结和思考



整体来看，这个批处理过程的优势有以下几点：

存储：原始记录集中收集到了 hdfs 集群，变成了分布式存储，自动冗余容错

处理：使用 hive 进行分布式处理，理论上数据多大都不怕，真正实现大数据处理

缺点则在于：

实时性：由于是离线的批处理，所以实时性比较低，一般部署为 T+1

可能的瓶颈和优化方向：

瓶颈：加工好的事实表直接落在了 mysql，报表的分析计算也都是在 mysql 完成的，随着数据的增长，mysql 的性能上限会变成瓶颈，导致报表查询越来越慢。

优化方向：临时处理，可以在 mysql 表上加上合适的索引，甚至还可以通过分库分表等方式进行数据库的设计优化；一劳永逸的办法，应该是把 事实表的存储和分析计算都迁移到 hive，只将最终计算好的报表数据，存到 redis，然后报表查询时直接读 redis。

6.开源地址

https://github.com/TurboWay/bigdata_practice

hive datax

上一篇 下一篇