Hive实现Zebra

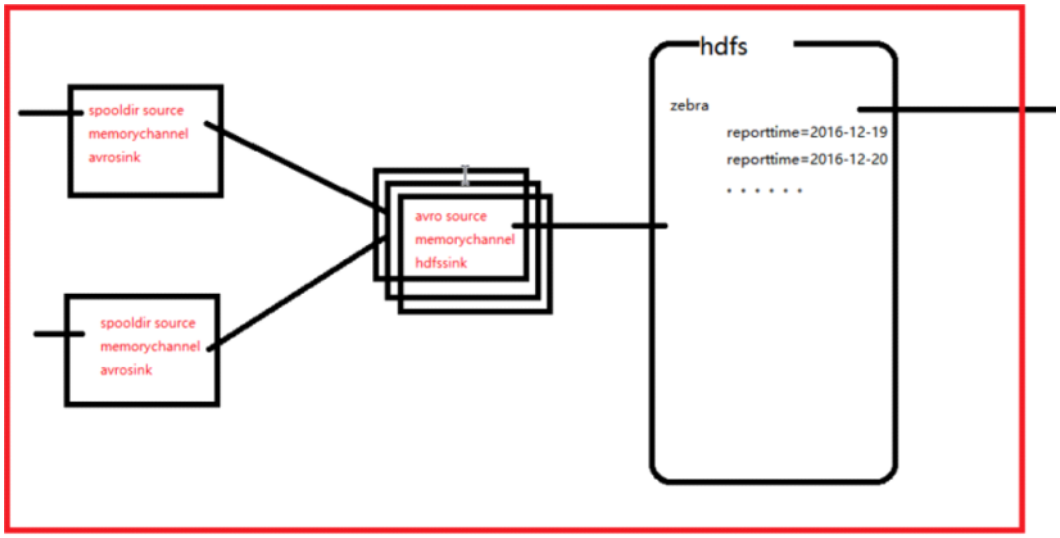
2016年1月20日 13:22

实现流程：

使用flume收集数据 --> 落地到hdfs系统中 --> 创建hive的外部表管理hdfs中收集到的日志 --> 利用hql处理zebra的业务逻辑 --> 使用sqoop技术将hdfs中处理完成的数据导出到mysql中

flume组件工作说明：

flume在收集日志的时候，按天为单位进行收集。hive在处理的时候，按天作为分区条件，继而对每天的日志进行统计分析。最后，hive将统计分析的结果利用sqoop导出到关系型数据库里，然后做数据可视化的相关工作。
对于时间的记录，一种思路是把日志文件名里的日志信息拿出来，第二种思路是flume在收集日志时，将当天的日期记录下来。我们用第二种思路。



环境结构说明

三台虚拟机01,02,03，01安装好flume，hadoop，hive，02，03安装flume
其中两台虚拟机02,03作为日志收集的服务器，然后汇聚给01虚拟机。
01虚拟机将汇总的日志数据落到hdfs上。供后续hive去处理

实现步骤

1.配置02，03的flume，进入flume安装目录下的conf目录
2.vim zebra.conf（文件名可以不固定）
3.配置示例：

```
#配置Agent
a1.sources=r1
a1.channels=c1
a1.sinks=k1

#配置source
a1.sources.r1.type=spooldir
a1.sources.r1.spoolDir=/home/zebra
a1.sources.r1.interceptors=i1
a1.sources.r1.interceptors.i1.type=timestamp

#配置sink
a1.sinks.k1.type=avro
a1.sinks.k1.hostname=192.168.234.21
a1.sinks.k1.port=44444

#配置channel
a1.channels.c1.type=memory
a1.channels.c1.capacity=1000
a1.channels.c1.transactionCapacity=100
```

#绑定
a1.sources.r1.channels=c1
a1.sinks.k1.channel=c1

4.将配置文件通过scp拷贝到01,03机器上

5.因为01虚拟机的flume是扇入操作，和02,03的配置不同，所以需要修改。

01虚拟机flume的zebra.conf配置示例：

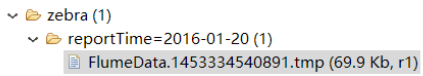
```
#配置Agent
a1.sources=r1
a1.channels=c1
a1.sinks=k1

#配置source
a1.sources.r1.type=avro
a1.sources.r1.bind=0.0.0.0
a1.sources.r1.port=44444

#配置sink
a1.sinks.k1.type=hdfs
a1.sinks.k1.hdfs.path=hdfs://192.168.234.21:9000/zebra/reportTime=%Y-%m-%d
a1.sinks.k1.hdfs.fileType=DataStream
a1.sinks.k1.hdfs.rollInterval=10
a1.sinks.k1.hdfs.rollSize=0
a1.sinks.k1.hdfs.rollCount=0

#配置channel
a1.channels.c1.type=memory
a1.channels.c1.capacity=1000
a1.channels.c1.transactionCapacity=100

#绑定
a1.sources.r1.channels=c1
a1.sinks.k1.channel=c1
```

6.在02,03虚拟机上，创建spoolDir指定的文件路径
/home/zebra

7.启动01,02,03的flume，进行测试。

8.将待处理的日志文件上传到/home/zebra下，最终，这个文件会被01虚拟机收集到，最后落到hdfs上。
注意：在上传日志文件的时候，不要在/root/work/data/flumedata 目录下通过rz 上传，因为rz是连续传输文件，这样会使得flume在处理时报错，错误为正在处理的日志文件大小被修改，所以最好是先把日志上传到linux的其他目录下，然后通过mv 指令移动到/root/work/data/flumedata 目录下

```
{ spoolDir: /root/work/data/flumedata }: Uncaught exception in SpoolDirectorySource thread. Restart or reconfigure Flume to continue processing
java.lang.IllegalStateException: File has been modified since being read: /root/work/data/flumedata/103_20150615143630_00_00_000.csv
```



Hive组件工作流程（建表语句不用写，重在了解整个ETL过程，这个过程很重要）

9.使用hive，创建zebra数据库
执行：create database zebra;
执行：use zebra;
然后建立分区，再建立表

详细建表语句：

```
create EXTERNAL table zebra (a1 string,a2 string,a3 string,a4 string,a5 string,a6 string,a7 string,a8 string,a9
string,a10 string,a11 string,a12 string,a13 string,a14 string,a15 string,a16 string,a17 string,a18 string,a19 string,a20
string,a21 string,a22 string,a23 string,a24 string,a25 string,a26 string,a27 string,a28 string,a29 string,a30 string,a31
string,a32 string,a33 string,a34 string,a35 string,a36 string,a37 string,a38 string,a39 string,a40 string,a41 string,a42
string,a43 string,a44 string,a45 string,a46 string,a47 string,a48 string,a49 string,a50 string,a51 string,a52 string,a53
string,a54 string,a55 string,a56 string,a57 string,a58 string,a59 string,a60 string,a61 string,a62 string,a63 string,a64
string,a65 string,a66 string,a67 string,a68 string,a69 string,a70 string,a71 string,a72 string,a73 string,a74 string,a75
string,a76 string,a77 string) partitioned by (reportTime string) row format delimited fields terminated by '|' stored as
textfile location '/zebra';
```

10.增加分区操作
执行：ALTER TABLE zebra add PARTITION (reportTime='2016-01-20') location '/zebra/reportTime=2016-01-20';

11.执行查询，看是否能查出数据

执行：select * from zebra;

```
Mobile Safari/534.30   text/html       http://detail.m.tmall.com/item.htm?id=40302977
78&pos=1       cookie2=1f79b50a70a5b5c623c7904336806420; t=15d42b44dee4ae0a412405c059
Fr4zjaEkT98e1%2Fg9rN1nBMQUUuR%2FpAxDovO7f5ednNQjnqIxEN0WI1YD   0       0       0
88             11      93287887015639365       6                               1
09649427375    1       15      155             0       10.83.106.110           55697
       48      14600   1400    1       0       1       3       6       200     51
d526334abae83a1368e200e7e7290be%26ttid%3D600000%2540taobao_android_4.9.0&ttid=&token=
roid 4.3; zh-cn; vivo X3L Build/JLS36C) AppleWebKit/534.30 (KHTML, like Gecko) Versio
SIONID=2BE2F826770EA5C0E2B298A581D72AF5; isg=28E892BDBBD711DD1988D030D50D0554; uc1=co
CLz3%2F65A%    0       0       0       0                       3       0       217
016            11      93287887015639366       6                               1
09649435581    1       15      3               0       10.83.106.110           55714
       49      14600   1400    1       0       1       3       6       200     56
       login.m.taobao.com      Mozilla/5.0 (Linux; U; Android 4.3; zh-cn; vivo X3L Bu
xt/html        http://login.taobao.com/member/logout.jhtml?loginURL=http%3A%2F%2Flogi
/RXTsCAd9oCsBM0mh0; _m_unitapi_v_=1409121915930; ALIPAYJSESSIONID=101f32dc35e44f9f92a
1409644839533; _m_h5_tk_enc=5bf6a44    0       0       0       0
34             11      93287887015639367       6                               1
09649434007    1       15      3               0       10.98.67.79             55087
       51      65535   1400    1       0       1       3       6       200     67
```

12.清洗数据，从原来的77个字段变为23个字段

建表语句：

```
create table dataclear(reporttime string,appType bigint,appSubtype bigint,userIp string,userPort bigint,appServerIP
string,appServerPort bigint,host string,cellid string,appTypeCode bigint,interruptType String,transStatus
bigint,trafficUL bigint,trafficDL bigint,retranUL bigint,retranDL bigint,procdureStartTime bigint,procdureEndTime bigint)
row format delimited fields terminated by '|';
```

13.从zebra表里导出数据到dataclear表里（23个字段的值）

建表语句：

```
insert overwrite table dataclear select reportTime,a23,a24,a27,a29,a31,a33,a59,a17,a19,a68,a55,a34,a35,a40,a41,a20,a21
from zebra;
```

14.处理业务逻辑，得到dataproc表

建表语句：

```
create table dataproc (reporttime string,appType bigint,appSubtype bigint,userIp string,userPort bigint,appServerIP
string,appServerPort bigint,host string,cellid string,attempts bigint,accepts bigint,trafficUL bigint,trafficDL
bigint,retranUL bigint,retranDL bigint,failCount bigint,transDelay bigint)row format delimited fields terminated by '|';
```

15.根据业务规则，做字段处理

建表语句：

```
insert overwrite table dataproc select
reporttime,appType,appSubtype,userIp,userPort,appServerIP,appServerPort,host,if(cellid ==
'',"000000000",cellid),if(appTypeCode == 103,1,0),if(appTypeCode == 103 and
find_in_set(transStatus,"10,11,12,13,14,15,32,33,34,35,36,37,38,48,49,50,51,52,53,54,55,199,200,201,202,203,204,205,206,3
02,304,306")!=0 and interruptType == 0,1,0),if(apptypeCode == 103,trafficUL,0), if(apptypeCode == 103,trafficDL,0),
if(apptypeCode == 103,retranUL,0), if(apptypeCode == 103,retranDL,0), if(appTypeCode == 103 and transStatus == 1 and
interruptType == 0,1,0),if(appTypeCode == 103, procdureEndTime - procdureStartTime,0) from dataclear;
```

16.查询关心的信息，以应用受欢迎程度表为例：

建表语句：

```
create table D_H_HTTP_APPTYPE(hourid string,appType bigint,appSubtype bigint,attempts bigint,accepts bigint,succRatio
double,trafficUL bigint,trafficDL bigint,totalTraffic bigint,retranUL bigint,retranDL bigint,retranTraffic
bigint,failCount bigint,transDelay bigint) row format delimited fields terminated by '|';
```

17.根据总表dataproc,按条件做聚合以及字段的累加

建表语句：

```
insert overwrite table D_H_HTTP_APPTYPE select
reporttime,apptype,appsubtype,sum(attempts),sum(accepts),round(sum(accepts)/sum(attempts),2),sum(trafficUL),sum(trafficD
L),sum(trafficUL)+sum(trafficDL),sum(retranUL),sum(retranDL),sum(retranUL)+sum(retranDL),sum(failCount),sum(transDelay)
from dataproc group by reporttime,apptype,appsubtype;
```

18.查询前5名受欢迎app

```
select hourid,apptype,sum(totalTraffic) as tt from D_H_HTTP_APPTYPE group by hourid,apptype sort by tt desc limit 5;
```

Sqoop组件工作流程：

将Hive表导出到Mysql数据库中，然后通过web应用程序+echarts做数据可视化工作。

实现步骤（讲完sqoop后，作为课后作业）：

1.在mysql建立对应的表

2.利用sqoop导出d_h_http_apptype表

导出语句：

```
sqoop export --connect jdbc:mysql://192.168.242.1:3306/zebra --username root --password root --export-dir
'/user/hive/warehouse/zebra.db/d_h_http_apptype/000000_0' --table d_h_http_apptype -m 1 --fields-terminated-by '|'
```