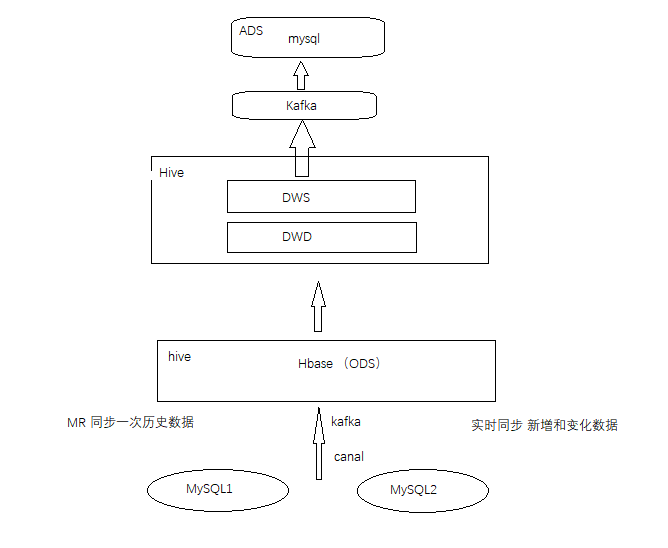
# 累计乘客需求

1. 先将历史数据从mysql数据同步到Hbase ， 使用MR；
2. Canal监控新增和变化数据，更新到Hbase
3. 报表数据同步mysql，读取hive数据后发送kafka，然后消费kafka更新mysql数据；

## 逻辑架构图



## 实时步骤：

同步数仓方案：

历史数据用MR程序一次性同步Hbase;

新增和变化数据使用canal 推送kafka，每天实时更新Hbase;

ADS同步方案

旧方案：取消sqoop同步：先删后写的缺陷；

新方案：读取Hive(DWS)表，写入kafka，再消费kafka更新Mysql(ADS)数据;

ODS：使用Hbase存储，配置Hive映射，数据每天实时更新;

DWD：hive清洗;

DWS：汇总统计;

ADS：存储Mysql,

## 测试环境操作

### 数据源离线同步：mysql -> Hbase(MR同步)

Hbase操作

删表

disable 'usercenter:ods\_usercenter\_user\_info'

drop 'usercenter:ods\_usercenter\_user\_info'

建表

create 'usercenter:ods\_usercenter\_user\_info' \

,{NAME => 'usercenter',VERSIONS=>1,BLOCKCACHE=>true,BLOOMFILTER=>'ROW',DATA\_BLOCK\_ENCODING => 'DIFF', TTL => '43200', COMPRESSION => 'LZO'} \

,{SPLITS => ['0','1','2','3','4','5','6','7','8','9']}

数据量统计

hbase org.apache.hadoop.hbase.mapreduce.RowCounter "usercenter:ods\_usercenter\_user\_info"

同步命令

java -jar usercenterEtl-1.0-SNAPSHOT-jar-with-dependencies.jar 172.20.17.10 3306 pubuser Z883eD21eo21q0E7HAkftKXOZ "172.18.239.190,172.18.239.188,172.18.239.189" "usercenter:ods\_usercenter\_user\_info"

MysqToHbaseMr代码

略

### Sql计算

--更新dwd数据

insert overwrite table dwd\_usercenter\_user\_info\_day

select

'2020-11-12' as date\_day

, T1.rowkey

, T1.uid

, T1.dataNodeId

, T1.nickName

, T1.sex

, T1.age

, T1.phone

, T1.status

, T1.headUrl

, T1.avatarName

, T1.bindEnterprise

, T1.forbidHelpCallCar

, T1.dataSource

, T1.createTime

, T1.updateTime

from ods\_usercenter\_user\_info T1

where NVL(uid,'') <> '' ;

--更新dws数据

insert into dws\_usercenter\_user\_count\_day

select

date\_day

, count(\*) AS user\_ct

from usercenter.dwd\_usercenter\_user\_info\_day

where date\_day='2020-11-12'

and status = '1'

group by date\_day

;

### 同步ADS

### 接口开发

## 正式shell脚本

#!/bin/bash

#######################load variable##################################

source /data/wanshun/config/apollo/apolloUtil.sh

hbaseConfig=`getApollo usercenterSyncConfig "dataMpl.hbase"`

mysqlConfig=`getApollo usercenterSyncConfig "dataMpl.mysql"`

echo ---------------"hbaseConfig----------------"

echo $hbaseConfig

echo ---------------"mysqlConfig----------------"

echo $mysqlConfig

zk=`echo "$hbaseConfig" | jq '."hbase.zkList"'`

echo $zk

zkList=`getJsonValue "$hbaseConfig" "hbase.zkList"`

echo ---------------"zkList----------------"

echo $zkList

# username=`getJsonValue "$configJson" "druid.exporder.username"`

#获取节点1配置

#key=druid.bmpl.usercenter1.url

#url=`echo "$mysqlConfig" | jq '."'$key'"'`

url=`getJsonValue "$mysqlConfig" "druid.bmpl.usercenter1.url"`

ipportdb=`echo $url | awk -F '//' '{print $2}' | awk -F '?' '{print $1}'`

node=`echo $ipportdb | awk -F ':' '{print $1}'`

port=`echo $ipportdb | awk -F ':' '{print $2}' | awk -F '/' '{print $1}'`

echo "url: $url"

echo "node: $node"

echo "port: $port"

#key=druid.bpml.usercenter1.username

#username=`echo "$mysqlConfig" | jq '."'$key'"'`

#key=druid.bpml.usercenter1.password

#password=`echo "$mysqlConfig" | jq '."'$key'"'`

username=`getJsonValue "$mysqlConfig" "druid.bpml.usercenter1.username"`

username=`decode64 "$username"`

password=`getJsonValue "$mysqlConfig" "druid.bmpl.usercenter1.password"`

password=`decode64 "$password"`

echo "username: $username"

echo "password: $password"

#获取节点2配置

#key=druid.bmpl.usercenter2.url

#url2=`echo "$mysqlConfig" | jq '."'$key'"'`

url2=`getJsonValue "$mysqlConfig" "druid.bmpl.usercenter2.url"`

ipportdb2=`echo $url2 | awk -F '//' '{print $2}' | awk -F '?' '{print $1}'`

node2=`echo $ipportdb2 | awk -F ':' '{print $1}'`

port2=`echo $ipportdb2 | awk -F ':' '{print $2}' | awk -F '/' '{print $1}'`

echo "url2: $url2"

echo "node2: $node2"

echo "port2: $port2"

#key=druid.bpml.usercenter2.username

#username2=`echo "$mysqlConfig" | jq '."'$key'"'`

#key=druid.bmpl.usercenter2.password

#password2=`echo "$mysqlConfig" | jq '."'$key'"'`

username2=`getJsonValue "$mysqlConfig" "druid.bpml.usercenter2.username"`

username2=`decode64 "$username2"`

password2=`getJsonValue "$mysqlConfig" "druid.bmpl.usercenter2.password"`

password2=`decode64 "$password2"`

echo "username2: $username2"

echo "password2: $password2"

######################execute script#####################################

#jar\_path="/data/wanshun/hadoop/packages/dataPlatform"

jar\_path="/home/pubuser/xxu"

hbase\_table="usercenter:ods\_usercenter\_user\_info"

echo "---------节点1同步开始---------------"

java -jar ${jar\_path}/userToHbase-1.0-SNAPSHOT-jar-with-dependencies.jar ${node} ${port} ${username} ${password} ${zkList} ${hbase\_table}

if [ $? -eq 0 ];then

echo 'No1 node sync data success'

else

echo 'No1 node sync data error'

fi

echo "---------节点1同步开始---------------"

java -jar ${jar\_path}/userToHbase-1.0-SNAPSHOT-jar-with-dependencies.jar ${node2} ${port2} ${username2} ${password2} ${zkList} ${hbase\_table}

if [ $? -eq 0 ];then

echo 'No2 node sync data success'

else

echo 'No2 node sync data error'

fi

## 正式hive脚本

#!/bin/bash

#- 任务名： <ScriptFile> dws\_usercenter\_user\_count\_day </ScriptFile>

#- 目标表: <TargetTable>

#- dwd\_usercenter\_user\_info\_day

#- dws\_usercenter\_user\_count\_day

#- </TargetTable>

#- 源表: <SourceTable>

#- ods\_usercenter\_user\_info

#- </SourceTable>

#- 运行方式： 每日全量

#- 运行时间： 每天凌晨30点

#- 任务功能说明：<Remark>数据大屏项目v2</Remark>

#- 创建时间： <CreateDate> 2020-11-12 </CreateDate>

#- 修改记录：

#--------------------------------------------------------

#- 修改人 修改日期 修改内容

#---------------------------------------------------------

#加载变量

input=`expr length "$1"`

case "${input}" in

0){

DATE\_DAY=`date -d "yesterday" +%Y-%m-%d`

echo "无日期输入，默认日期取系统时间前一天:"$DATE\_DAY" "

};;

10){

DATE\_DAY=$1

echo “输入跑数日期为: "$DATE\_DAY" ”

};;

\*){

echo "please input 'yyyy-MM-hh' "

exit 0;

};;

esac

#执行Hive

hive shell << EOF

use usercenter;

insert overwrite table dwd\_usercenter\_user\_info\_day

select

'${DATE\_DAY}' as date\_day

, T1.rowkey

, T1.uid

, T1.dataNodeId

, T1.nickName

, T1.sex

, T1.age

, T1.phone

, T1.status

, T1.headUrl

, T1.avatarName

, T1.bindEnterprise

, T1.forbidHelpCallCar

, T1.dataSource

, T1.createTime

, T1.updateTime

from ods\_usercenter\_user\_info T1

where NVL(uid,'') <> '' ;

drop table if exists tmp\_dwd\_usercenter\_user\_info\_day\_01;

create table tmp\_dwd\_usercenter\_user\_info\_day\_01

as

select date\_day,user\_ct

from dws\_usercenter\_user\_count\_day

where date\_day > '${DATE\_DAY}'

OR (date\_day < '${DATE\_DAY}' and date\_day>= date\_add('${DATE\_DAY}',-7));

insert overwrite table dws\_usercenter\_user\_count\_day select date\_day,user\_ct from tmp\_dwd\_usercenter\_user\_info\_day\_01;

insert into dws\_usercenter\_user\_count\_day

select

date\_day

, count(\*) AS user\_ct

from usercenter.dwd\_usercenter\_user\_info\_day

where date\_day='${DATE\_DAY}'

and status = '1'

group by date\_day

;

EOF

## 正式DDL建表语句

HBASE建表

指在帮助用户快速搭建工具框架,利用一个小demo迅速熟悉工具功能.

create 'usercenter:ods\_usercenter\_user\_info' \

,{NAME => 'usercenter',VERSIONS=>1,BLOCKCACHE=>true,BLOOMFILTER=>'ROW',DATA\_BLOCK\_ENCODING => 'DIFF', TTL => '43200', COMPRESSION => 'LZO'} \

,{SPLITS => ['0','1','2','3','4','5','6','7','8','9']}

Hive建表

ODS建表，映射Hbase建表

use usercenter;

drop table if exists `ods\_usercenter\_user\_info`;

create external table `ods\_usercenter\_user\_info` (

rowkey string COMMENT 'uid反转',

`uid` string COMMENT '用户id',

`dataNodeId` string COMMENT '对应的存储数据库id',

`nickName` string COMMENT '用户名',

`sex` string COMMENT '0女 1男 2未知',

`age` string COMMENT '出生日期，年龄',

`phone` string COMMENT '手机号',

`status` string COMMENT '状态：-2带锁定，-1锁定,0删除,1正常 , 2\_注销',

`headUrl` string COMMENT '头像信息',

`avatarName` string COMMENT '头像名称',

`bindEnterprise` string COMMENT '是否绑定企业：0未绑定，1已绑定',

`forbidHelpCallCar` string COMMENT '是否禁止他人为自己叫车：0禁止，1不禁止',

`dataSource` string COMMENT '数据来源:1现网',

`createTime` string COMMENT '创建时间',

`updateTime` string COMMENT '更新时间'

)

ROW FORMAT SERDE

'org.apache.hadoop.hive.hbase.HBaseSerDe'

STORED BY

'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

WITH SERDEPROPERTIES (

'hbase.columns.mapping'=':key,usercenter:uid,usercenter:dataNodeId,usercenter:nickName,usercenter:sex,usercenter:age,usercenter:phone,usercenter:status,usercenter:headUrl,usercenter:avatarName,usercenter:bindEnterprise,usercenter:forbidHelpCallCar,usercenter:dataSource,usercenter:createTime,usercenter:updateTime',

'serialization.format'='1')

TBLPROPERTIES (

'hbase.table.name'='usercenter:ods\_usercenter\_user\_info');

DWD建表

drop table if exists usercenter.dwd\_usercenter\_user\_info\_day;

create table usercenter.dwd\_usercenter\_user\_info\_day(

date\_day date COMMENT '统计日期',

rowkey string COMMENT 'uid反转',

`uid` string COMMENT '用户id',

`dataNodeId` string COMMENT '对应的存储数据库id',

`nickName` string COMMENT '用户名',

`sex` string COMMENT '0女 1男 2未知',

`age` string COMMENT '出生日期，年龄',

`phone` string COMMENT '手机号',

`status` string COMMENT '状态：-2带锁定，-1锁定,0删除,1正常 , 2\_注销',

`headUrl` string COMMENT '头像信息',

`avatarName` string COMMENT '头像名称',

`bindEnterprise` string COMMENT '是否绑定企业：0未绑定，1已绑定',

`forbidHelpCallCar` string COMMENT '是否禁止他人为自己叫车：0禁止，1不禁止',

`dataSource` string COMMENT '数据来源:1现网',

`createTime` string COMMENT '创建时间',

`updateTime` string COMMENT '更新时间'

)

row format delimited fields terminated by '\t' lines terminated by '\n' ;

DWS建表

drop table if exists `dws\_usercenter\_user\_count\_day`;

create table `dws\_usercenter\_user\_count\_day` (

date\_day date

, user\_ct bigint --累计乘客数

)

row format delimited fields terminated by '\t' lines terminated by '\n' ;

ADS建表

CREATE TABLE usercenter.ads\_usercenter\_user\_day (

`id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT 'id自增主键',

`dateDay` date DEFAULT NULL COMMENT '统计日期',

`userCnt` bigint DEFAULT NULL COMMENT '累计乘客数',

PRIMARY KEY (`id`),

KEY `idx\_dateDay` (`dateDay`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 COMMENT='累计乘客数日统计数';

## 接口开发代码

# 推荐乘客同步需求，临时

Recommend.recommendPassengerRecord

Mongo 数据同步到Hbase，使用MR

## DDL

### Hbase建表

create 'recommend:ods\_recommend\_recommend\_passenger\_record' \

,{NAME => 'recommend',VERSIONS=>1,BLOCKCACHE=>true,BLOOMFILTER=>'ROW',DATA\_BLOCK\_ENCODING => 'DIFF', TTL => '43200', COMPRESSION => 'LZO'} \

,{SPLITS => ['0','1','2','3','4','5','6','7','8','9']}

### Hive建表

use recommend;

drop table if exists `recommend.ods\_recommend\_passenger\_record\_info`;

create external table `recommend.ods\_recommend\_passenger\_record\_info` (

rowkey string

, id string

, recomUserId string

, recomName string

, recomPhone string

, recomType string

, refereedType string

, passengerPhone string

, result string

, createTime string

, updateTime string

, remark string

, dataSource string

, organizationName string

, agencyNumber string

, refereeStatus string

, registerCityCode string

, passengerId string

)

ROW FORMAT SERDE

'org.apache.hadoop.hive.hbase.HBaseSerDe'

STORED BY

'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

WITH SERDEPROPERTIES (

'hbase.columns.mapping'=':key,recommend:\_id,recommend:recomUserId,recommend:recomName,recommend:recomPhone,recommend:recomType,recommend:refereedType,recommend:passengerPhone,recommend:result,recommend:createTime,recommend:updateTime,recommend:remark,recommend:dataSource,recommend:organizationName,recommend:agencyNumber,recommend:refereeStatus,recommend:registerCityCode,recommend:passengerId',

'serialization.format'='1')

TBLPROPERTIES (

'hbase.table.name'='recommend:ods\_recommend\_recommend\_passenger\_record');

## 测试操作

disable 'recommend:ods\_recommend\_recommend\_passenger\_record'

drop 'recommend:ods\_recommend\_recommend\_passenger\_record'

truncate 'recommend:ods\_recommend\_recommend\_passenger\_record'

生产： /usr/local/service/hadoop/bin/Hadoop

测试： /usr/lib/hadoop-current/bin/hadoop

hadoop jar recommendPassengerToHbase-1.0-SNAPSHOT.jar com.wanshun.mongotobbasemr.BatchMain "172.18.5.100:20017//recommend.recommendPassengerRecord" "172.18.239.188:2181" "recommend:ods\_recommend\_recommend\_passenger\_record" "pubuser" "KAov0XvKeFPVrF+4+rz-6P49B9U6Rz"

hbase org.apache.hadoop.hbase.mapreduce.RowCounter 'recommend:ods\_recommend\_recommend\_passenger\_record'

# MysqlToHbae同步：driver.companyexpaudithistory

Mysql表

driver.companyexpaudithistory

hbase建表

create\_namespace 'driver'

create 'driver:ods\_driver\_companyexpaudithistory\_info' \

,{NAME => 'driver',VERSIONS=>1,BLOCKCACHE=>true,BLOOMFILTER=>'ROW',DATA\_BLOCK\_ENCODING => 'DIFF', TTL => '43200', COMPRESSION => 'LZO'} \

,{SPLITS => ['0','1','2','3','4','5','6','7','8','9']}

同步hbase -- ROWS=165515

java -jar companyExpAuditHistoryToHbase-1.0-SNAPSHOT-jar-with-dependencies.jar 172.18.5.101 3306 pubuser nh7lTVh5f9HFgjczHChvmqQcSQjE62 "172.18.239.190,172.18.239.188,172.18.239.189" "driver:ods\_driver\_companyexpaudithistory\_info"

hbase同步验证 --165515

hbase org.apache.hadoop.hbase.mapreduce.RowCounter "driver:ods\_driver\_companyexpaudithistory\_info"

hive映射表

drop table if exists driver.ods\_driver\_companyexpaudithistory\_info;

create external table driver.ods\_driver\_companyexpaudithistory\_info (

rowkey String

, id String

, driverId String

, carId String

, brandId String

, brandModelId String

, seatNum String

, loginPassword String

, name String

, phone String

, drivingAge String

, carNum String

, carColorId String

, idNumber String

, sex String

, auditTime String

, audit String

, auditRemark String

, commitAuditTimes String

, relationId String

, orgType String

, registerSim String

, registerDeviceId String

, processId String

, auditType String

, registerCityCode String

, agencyNumber String

, bindSp String

, createTime String

, updateTime String

, registerPlatform String

, registerSource String

, rejectReason String

, registerVersion String

, encrypt String

, nameEncrypt String

, phoneEncrypt String

, idNumberEncrypt String

, carNumEncrypt String

)

ROW FORMAT SERDE

'org.apache.hadoop.hive.hbase.HBaseSerDe'

STORED BY

'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

WITH SERDEPROPERTIES (

'hbase.columns.mapping'=':key,driver:id ,driver:driverId ,driver:carId ,driver:brandId ,driver:brandModelId ,driver:seatNum ,driver:loginPassword ,driver:name ,driver:phone ,driver:drivingAge ,driver:carNum ,driver:carColorId ,driver:idNumber ,driver:sex ,driver:auditTime ,driver:audit ,driver:auditRemark ,driver:commitAuditTimes ,driver:relationId ,driver:orgType ,driver:registerSim ,driver:registerDeviceId ,driver:processId ,driver:auditType ,driver:registerCityCode ,driver:agencyNumber ,driver:bindSp ,driver:createTime ,driver:updateTime ,driver:registerPlatform ,driver:registerSource ,driver:rejectReason ,driver:registerVersion ,driver:encrypt ,driver:nameEncrypt ,driver:phoneEncrypt ,driver:idNumberEncrypt ,driver:carNumEncrypt',

'serialization.format'='1')

TBLPROPERTIES (

'hbase.table.name'='driver:ods\_driver\_companyexpaudithistory\_info');