# WanShun环境

## 测试环境

### 测试阿里云

172.18.19.217 port：20962 用户名和密码是你名字拼音

### Kafka

172.18.5.156

kafka-console-consumer.sh --bootstrap-server 172.18.5.156:9092 --topic test-hive-to-kafka

kafka-console-producer.sh --broker-list 172.18.5.156:9092 --topic test-hive-to-kafka

### mongo

mongo 172.18.5.100:20017

use admin

db.auth("appuser","0IP6k+ERm39lTpjsdBkhohY75WMkyB")

## 其他网站

git账号

用户名和密码为名字拼音

禅道账号：

用户名： xuxiu, 密码：xuxiu@1006

阿里云HDFS 和Hue

用户：xuxiu，密码：xuxiu-0922

腾讯云的HDFS HUE HBASE

账户 xuxiu xuxiu@1016

## 生产环境

~~阿里云的跳板机： 172.18.2.214 端口：20962~~

腾讯云的跳板机： 172.25.0.3 端口：20962

腾讯云mysql-ads

mysql -udriver -P3306 -h 172.21.78.11 -p'YWfD\_nz6jlBJCtQ2tN2q6yZqt6mxYQ'

# Linux 常用命令

#指定目录查询文件

find /home/pubuser/ -name hive.para

#全盘扫描查找文件，将错误日志不显示

find / -name hive.para 2>/dev/null

#任务启动

sh test.sh >/dev/null 2>&1 &

#查看大小

du -h

# 加入环境变量

vi /etc/profile

export PATH=$PATH:/export/project/bin

source /etc/profile

## vim 常用命令

ggVG 全选

set nu 显示行数

num1,num2d 删除从num1到num2的行

# Shell 常用命令

## shell的if判断语句

if [ command ];then

符合该条件执行的语句

elif [ command ];then

符合该条件执行的语句

else

符合该条件执行的语句

fi

#判断执行成功

echo 'hello'

if [ $? -eq 0 ];then

echo 'That is ok'

else

echo 'That is not ok'

fi

## shell的函数

#!/bin/bash

count=0 #全局变量

function getsum(){

local sum=0 #局部变量

for((i=$1; i<=$2; i++)); do

let sum+=i

let count++

done

echo "$sum"

return $?

}

m=1

n=100

#函数的结果没有真正输出终端，$() 只是获取第一个echo的输出结果;

total=$(getsum $m $n)

echo "The sum is $total"

echo "The count is $count"

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>"

#真正执行函数，并且输出

getsum $m $n

echo "The sum2 is $total"

echo "The count2 is $count"

----------------------------------|

The sum is 5050 |

The count is 0 |

>>>>>>>>>>>>>>>>>>>>>>>>>> |

5050 |

The sum2 is 5050 |

The count2 is 100 |

----------------------------------|

# Hadoop 组件常用命令

## Hadoop常用命令

--查看HDFS的NameNode的状态

hdfs haadmin -getServiceState nn1

--查看yarn日志

yarn application -kill application\_Id

yarn logs -applicationId application\_1605488491965\_0291 > application\_1605488491965\_0291.log

## Hive 常用命令

常用参数

#表头显示

set hive.resultset.use.unique.column.names=false;

set hive.cli.print.header=true;

#日志

set hiveconf hive.root.logger=ERROR;

#引擎

set hive.execution.engine=spark;

### 迁移hive数据

Eg：dws\_exporder\_order\_company\_hour

--##1.检查分区数据

hadoop fs -ls /user/hive/warehouse/exporder.db/dws\_exporder\_order\_company\_hour/date\_month\_p=2020-10/date\_day\_p=2020-10-23

--##2.建本地目录存放数据

mkdir -p /home/pubuser/v3copyhive/dws\_exporder\_order\_company\_hour/date\_month\_p=2020-10/date\_day\_p=2020-10-23

--##3.从生产拉取指定分区

hadoop fs -get /user/hive/warehouse/exporder.db/dws\_exporder\_order\_company\_hour/date\_month\_p=2020-10/date\_day\_p=2020-10-23/\* /home/pubuser/v3copyhive/dws\_exporder\_order\_company\_hour/date\_month\_p=2020-10/date\_day\_p=2020-10-23

\*/ ---

--##4.压缩拷贝到测试

zip -r dws\_exporder\_order\_company\_hour.zip dws\_exporder\_order\_company\_hour/

--##5.先建好分区，再向分区分录加载数据

-- 或者使用修复分区命令： msck repair table TABLE\_NAME;

alter table exporder.dws\_exporder\_order\_company\_hour add partition(date\_month\_p='2020-10',date\_day\_p='2020-10-23')

--##6.将数据上传至 分区目录

hadoop fs -put /home/pubuser/xxu/data/dws\_exporder\_order\_company\_hour/date\_month\_p=2020-10/date\_day\_p=2020-10-23/\* /user/hive/warehouse/exporder.db/dws\_exporder\_order\_company\_hour/date\_month\_p=2020-10/date\_day\_p=2020-10-23

\*/

--##7.检查，count() 失效，select 是正常

select \* from exporder.dws\_exporder\_order\_company\_hour where date\_day\_p='2020-10-23' limit 10;

### 自定义UDF函数

--# 编写，打包

--# 启动hive添加jar包，创建临时方法，测试

add jar /home/pubuser/xxu/testUFF01.jar

create temporary function testUFF01 as 'com.wsjc.app.hivefunc.TestUDF01';

select testUFF01('1583776223469|{"cm":{"ln":"-48.5","mid":"0","nw":"4G"},"ap":"app","et":[{"ett":"1583705574227","en":"display"},{"ett":"1583760986259","en":"loading"}]}','st')

## Hbase 常用命令

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

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

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

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

--hbase建命名空间

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

--hbase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']}

--hbase统计数据量

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

--创建hive映射表

drop table if exists usercenter.ods\_usercenter\_user\_info;

create external table usercenter.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');

## Mongo 常用命令

--常用查询

show dbs

use carorder

show collections

db.OrderExpCarInstant.find().pretty().limit(1)

--插入数据自动建表

db.goods.recommendPassengerRecord({\_id:1,passengerId:'111',createTime:'2018-04-19 06:51:32.000','2019-04-19 06:51:32.000'});

--列出重复的id数据 = select passengerId where count()>1

db.recommendPassengerRecord.aggregate( [

{

$group: {

\_id: "$passengerId",

count: { $sum: 1 }

}

},

{ $match: { count: { $gt: 1 } } }

] )

db.recommendPassengerRecord.find({passengerId:1021009300338}).pretty()

## Sqoop测试

参考案列

sqoop import

"-Dorg.apache.sqoop.splitter.allow\_text\_splitter=true"

--connect jdbc:mysql://172.18.225.100:3306/driver

--driver com.mysql.jdbc.Driver

--username hadoopuser

--password 'ji8crOgOyQjwweaRyONmAKB6rrKcsa'

-m 12

--split-by driverId

--hbase-table 'exporder:dim\_expdrivervip'

--hbase-row-key 'rowkey'

--column-family 'exporder'

-query 'select driverId as rowkey,driverId,cityCode,areaCode,status,smsStatus,startDate,endDate,updateTime,createTime from expdrivervip where $CONDITIONS';

### sqoop联通测试

#成功

sqoop list-databases --connect jdbc:mysql://192.168.10.15:3306/ -username root -password 666666

### sqoop import：Mysql=>Hive

#成功

sqoop list-databases --connect jdbc:mysql://172.20.17.10:3306/ -username pubuser -password 'Z883eD21eo21q0E7HAkftKXOZ'

sqoop list-databases --connect jdbc:mysql://172.20.17.11:3306/ -username pubuser -password 'Z883eD21eo21q0E7HAkftKXOZ'

#成功

sqoop list-databases --connect jdbc:mysql://172.18.239.193:3306/ -username pubuser -password 'Ko8euGIJJkt>KDPEeCBQCn]Cs1wKjp'

#### #全量覆盖: mysql导入hive 成功,

sqoop import \

--connect jdbc:mysql://192.168.10.15:3306/test \

--username root \

--password 666666 \

--table user \

--input-null-string '\\N' \

--input-null-non-string '\\N' \

--fields-terminated-by '\t' \

--lines-terminated-by '\n' \

--delete-target-dir \

--num-mappers 1 \

--hive-import \

--hive-overwrite \

--hive-database wsjc \

--hive-table usercenter\_user\_info

##--sqoop 联通测试 ： 成功

sqoop list-databases --connect jdbc:mysql://172.18.5.102:3306/ -username pubuser -password 'nh7lTVh5f9HFgjczHChvmqQcSQjE62'

#### #全量导入测试 ： 成功

sqoop import \

--connect jdbc:mysql://172.18.5.102:3306/test \

--username pubuser \

--password 'nh7lTVh5f9HFgjczHChvmqQcSQjE62' \

--table user \

--input-null-string '\\N' \

--input-null-non-string '\\N' \

--fields-terminated-by '\t' \

--lines-terminated-by '\n' \

--delete-target-dir \

--num-mappers 1 \

--hive-import \

--hive-overwrite \

--hive-database usercenter \

--hive-table ods\_usercenter\_user\_info

#### #增量导入一天 ： 成功

sqoop import \

--connect jdbc:mysql://172.18.5.102:3306/test \

--username pubuser \

--password 'nh7lTVh5f9HFgjczHChvmqQcSQjE62' \

--table user \

--columns "uid,dataNodeId,nickName,sex,age,phone,status,headUrl,avatarName,bindEnterprise,forbidHelpCallCar,dataSource,createTime,updateTime" \

--where "createTime='2020-11-02 00:30:00'" \

--input-null-string '\\N' \

--input-null-non-string '\\N' \

--fields-terminated-by '\t' \

--lines-terminated-by '\n' \

--num-mappers 1 \

--hive-import \

--hive-overwrite \

--hive-database usercenter \

--hive-table ods\_usercenter\_user\_info \

#### #限定条件1：增量导入 ： 成功

sqoop import \

--connect jdbc:mysql://172.18.5.102:3306/test \

--username pubuser \

--password 'nh7lTVh5f9HFgjczHChvmqQcSQjE62' \

--table user \

--columns "uid,dataNodeId,nickName,sex,age,phone,status,headUrl,avatarName,bindEnterprise,forbidHelpCallCar,dataSource,createTime,updateTime" \

--where "createTime='2020-11-02 00:30:00'" \

--input-null-string '\\N' \

--input-null-non-string '\\N' \

--fields-terminated-by '\t' \

--lines-terminated-by '\n' \

--num-mappers 1 \

--hive-import \

--hive-overwrite \

--hive-database usercenter \

--hive-table ods\_usercenter\_user\_info \

#### #限定条件导入：成功 substring(createTime,1,10)='2020-11-03'

sqoop import \

--connect jdbc:mysql://172.18.5.102:3306/test \

--username pubuser \

--password 'nh7lTVh5f9HFgjczHChvmqQcSQjE62' \

--table user \

--columns "uid,dataNodeId,nickName,sex,age,phone,status,headUrl,avatarName,bindEnterprise,forbidHelpCallCar,dataSource,createTime,updateTime" \

--where "substring(createTime,1,10)='2020-11-03' OR substring(updateTime,1,10)='2020-11-03'" \

--input-null-string '\\N' \

--input-null-non-string '\\N' \

--fields-terminated-by '\t' \

--lines-terminated-by '\n' \

--num-mappers 1 \

--hive-import \

--hive-overwrite \

--hive-database usercenter \

--hive-table ods\_usercenter\_user\_info

# IDEA常用命令

# 其他开发工具

## Notpad++

### #去重

^(.\*?)$\s+?^(?=.\*^\1$)

### #删除每行--remove之前所有内容

(.\*)--remove

--remove

## sublime\_text