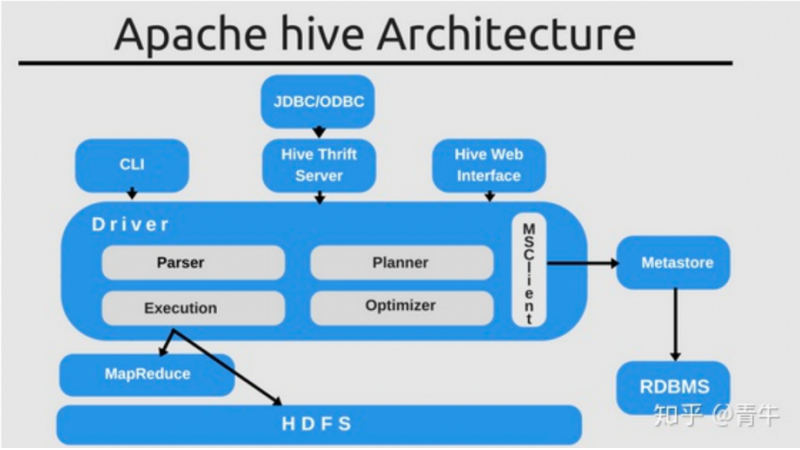
数据仓库构建_hive

（相当于自己对这几天学习的 hive 的基本使用和数据仓库构建的一个笔记整理，同时也包含了一些其他外延的简单知识扩展补充。）

一．Hive 架构



依赖服务：hdfs spark zookeeper yarn hive hue

Hadoop 的基础框架是怎样的？

Hadoop 是一个**开源的分布式计算框架**，由 Apache Software Foundation 开发和维护。它主要用于处理和分析大数据。 Hadoop 的核心设计理念是将计算任务分布到多个节点上，以实现高度可扩展性和容错性。

Hadoop 作为一个分布式框架，对于数据管理系统，无非就是解决两个问题：第一，数据怎么存，第二是数据怎么计算。

对于数据怎么存这个问题，从原来的一台服务器足矣存储所有数据，到现在大数据爆炸式增长，我们不得不用多台服务器去存储数据，所以为了更好地管理这些服务器，就用到了 HDFS；它是一个分布式文件系统，它为这些服务器提供了一个接口，从而在用户使用时，让这些服务器能够像是一个存储空间无限大的服务器。

对于数据怎么算这个问题，Hadoop 提供了 mapreduce。这是一个分布式计算框架（基于磁盘），它提供一系列 API，从而提供一个任务并行的框架。它计算时主要分为两个阶段：首先是 map 阶段，在这个阶段，它会为一个大任务找 100（总之就是很多）个帮手，然后把任务分成 100 份小任务，每个人将会完成其中一份小任务。接着，来到第二个阶段，reduce。当每个人的小任务做完后，小任务的结果都会汇总到你这儿，然后导出一份最终的结果。（类似的有 spark，是基于内存的计算框架。）

另外，在过去单机数据库的时代，一些用户可以使用 sql 语言（它是单机计算，计算太慢），它降低了数据处理的门槛。但到了大数据时代，我们发现不得不写 mapreduce 程序（分布式处理程序），于是我们希望在 hadoop 上写 sql 程序。于是有了 hive，它是一个在 hadoop 上进行结构化数据处理的解决方案。其核心模块是 metastore，可以存储结构化数据（为了让用户用 sql 来处理数据），其执行引擎和 sql 没太大区别，一个区别在于 Hive 可以把一个 sql 语句翻译成 mapreduce 任务，翻译后的结果加工再传给用户。（Hadoop 上除了有 hive 这样的解决方案，还有 impala 和 presto）

除了 mapreduce 和 hive，hadoop 还有 yarn，它是资源管理和任务调度的模块。

二．Hive 的一些基本操作

Mysql 连接方法：

```
1. mysql-uroot-pcloudera
```

使用 hue（Hue 是一个能够与 Apache Hadoop 交互的 Web 应用程序。）连接 hive：

```
1. hive
```

1. 内表 dw 层

```
2. show tables;
3. create database taobao;
4. use taobao;
5.
6. # 创建数据表
7. CREATETABLE`taobao`(`user`STRING,`item`STRING,`category`STRING,
8.  `behavior`STRING,`time`STRING)
9. ROWFORMATDELIMITED
10. FIELDSTERMINATEDBY','
11. STOREDASTEXTFILE;
12. -- 读取 HDFS 目录文件的方式
13.
14. # 加载本地文件到 taobaohive 表中
15. LOADDATALOCALINPATH'/home/cloudera/Desktop/UserBehavior_200w.csv'
16. OVERWRITEINTOTABLEtaobao;
17.
18.  LOADDATALOCALINPATH'/tmp/UserBehavior_200w.csv'
19. OVERWRITEINTOTABLEtaobao;
20. show
21. LOADDATALOCALINPATH'/home/cloudera/Desktop/test.txt'
22. OVERWRITEINTOTABLEtaobao;
23.
24. # 加载 HDFS 文件到 taobao hive 表中
25. LOADDATAINPATH'/tmp/taobao/UserBehavior_200w.csv'
26. OVERWRITEINTOTABLEtaobao;
```

2. 外表 ods 层

```
3. # 放置外表数据
4. # 在 hdfs 上创建/tmp/taobao/目录
5. hadoop fs -mkdir /tmp/taobao/
6. # 将本地/home/cloudera/Desktop/UserBehavior_200w.csv 拷贝到 hdfs /tmp/taobao/ 目录下
7. hadoop fs -put /home/cloudera/Desktop/UserBehavior_200w.csv /tmp/taobao/
8.
9. CREATE EXTERNAL TABLE `taobao_external` (`user` STRING, `item` STRING, `category` S
   TRIN
10. `behavior`STRING,`time`STRING)
11. ROWFORMATDELIMITED
```

```
FIELDSTERMINATEDBY','
STOREDASTEXTFILE
LOCATION'/tmp/taobao/';

LOADDATALOCALINPATH'/tmp/UserBehavior_200w.csv'
OVERWRITEINTOTABLEtaobao_external;
```

内外表的一个区别：

内表删除: drop table table_name l 表 数据都会被删除

外表删除: drop table table_name l 表会被删除 数据不会被删除

2. 分区

```
CREATETABLE`taobao_partition`(`user`STRING,`item`STRING,`category`STRING,
`behavior`STRING,`time`STRING)
PARTITIONEDBY(`date`STRING)
ROWFORMATDELIMITED
FIELDSTERMINATEDBY'\t'
STOREDASTEXTFILE;

# 从本地加载数据到 Hive 中
LOADDATALOCALINPATH'/home/cloudera/Desktop/taobao-11-27.csv'
INTOTABLE`taobao_partition`
PARTITION(date='2017-11-27');

LOADDATALOCALINPATH'/home/cloudera/Desktop/taobao-11-28.csv'
INTOTABLE`taobao_partition`
PARTITION(date='2017-11-28');
select * from taobao_partition where date = '2017-11-27'
select*fromtaobaopartitionwheredatelike'2017-11-%'
                G

select from taobao_partition where date like 2017 11 %
面试题:hive 分区表如何提高查询速度?
回答:分区表通过利用分区条件加载对应数据，采用减少数量的方式来加速查询

LOADDATALOCALINPATH'/tmp/taobao-11-27.csv'INTOTABLE`taobao_partition`
PARTITION(date='2017-11-27');
LOADDATALOCALINPATH'/tmp/taobao-11-28.csv'INTOTABLE`taobao_partition`
PARTITION(date='2017-11-28');

# 查看分区值
show partitions {table_name};

```

```
# 将本地数据上传到 HDFS 内
hadoop fs -put /home/cloudera/Desktop/taobao-11-29.csv /home/cloudera/
# 将 HDFS 数据添加到 Hive 分区中
ALTERTABLE`taobao_partition`ADDPARTITION(date='2017-11-29')location'/tmp/taobao-11
```

三．数据仓库构建

1. 工作流：

前端 ->java->Mysql 业务数据库->mysql dump->etl(file)->ods->dwd->dws->dwb

首先，这是一个模拟正常工作流程的过程。首先从 mysql 业务数据库中拉表，然后下载成文件形式，导入到 hive 中，从 ods 往后，就是数据仓库构建的过程。

而在业务数据库之前，是前端到服务端。服务端比如 Java 层对数据进行增删改查。比如从前端下到一个订单，发送到 java，从 java 层校验一个订单，帮助创建一个订单，或者从 mysql 的 order 表中创建一个数据，然后每天都这样跑。然后凌晨的时候，通过 mysqldump 下载成文件然后往数据仓库中走。数据分析主要做的是从业务数据端抽数据，增删改查，然后后面只是读，不会去改。

2. 具体流程

首先是 Mysql 业务数据库->mysql dump->etl(file)

```
# connect to mysql
mysql -uroot -pcloudera

# create
CREATE DATABASE taobao; USE taobao;

# mysql create table
CREATE TABLE `userbehavior` (
`user` varchar(255) DEFAULT NULL,
`item` varchar(255) DEFAULT NULL, `category` varchar(255) DEFAULT NULL, `behavior` varchar(255) DEFAULT NULL, `time` varchar(255) DEFAULT NULL,
KEY `user` (`user`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

# load data
LOAD DATA INFILE '/home/cloudera/Desktop/UserBehavior_200w.csv' INTO TABLE `userbehavior`
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';
select * from userbehavior limit 10;


hadoop: hdfs mr yarn

```

```
# 源头表 -> data 数据文件(7 天) -> ods(外表) -> dwd -> dws

# dump data  #  下载数据
mkdir/tmp/userbehavior/
chmod777/tmp/userbehavior/

# mysql 数据导出工具

mysqldump -uroot -pcloudera \
--databases taobao --tables userbehavior \
--fields-terminated-by=',' --no-create-info \ 4 -T/tmp/userbehavior/

# upload to hdfs  #导入到 hdfs
hadoop fs -mkdir /tmp/userbehavior/
hadoop fs -put /tmp/userbehavior/userbehavior.txt /tmp/userbehavior/
```

接着是 hive 的各层建立过程：

ods->dwd->dws->dwb

```
# hive operation
# create external table for ods CREATE DATABASE dw;
USE dw;
CREATE EXTERNAL TABLE `ods_userbehavior_ext` (`user` STRING, `item` STRING, `category` STRING, `behavior` STRING, `time` STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',' STORED AS TEXTFILE
LOCATION '/tmp/userbehavior/';

# create dwd table
CREATE TABLE `dwd_userbehavior`
(`user` STRING, `item` STRING, `category` STRING, `behavior` STRING, `date` STRING, `time` STRING) ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

 # ods -> dwd
INSERT INTO TABLE dwd_userbehavior
SELECT `user`,`item`,`category`,`behavior`, from_unixtime(cast(time as BIGINT), 'yyyy-MM-dd') `date`, from_unixtime(cast(time as BIGINT), 'HH') `time`
FROM ods_userbehavior_ext;
SELECT
from_unixtime(time, 'yyyy-MM-dd') `date` FROM ods_userbehavior;

# create dws table
CREATE TABLE dws_day_summary
```

```
(`date` STRING, `active_user` INT,`active_item` INT) ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

# dwd -> dws
INSERT INTO TABLE dws_day_summary SELECT
'2017-11-27' as date,
count(distinct `user`) as active_user, count(distinct `item`) as active_item FROM dwd_userbehavior
WHERE `date` = '2017-11-27';

INSERT INTO TABLE dws_day_summary SELECT
'2017-11-28' as date,
count(distinct `user`) as active_user, count(distinct `item`) as active_item FROM dwd_userbehavior
WHERE `date` = '2017-11-28';



# create dwb table(宽表)
CREATE TABLE dwb_day_user_summary_inc (`user` STRING,`pv_count` INT, `buy_count` INT) PARTITIONED BY (`date` STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

# dwd -> dwb
INSERT INTO TABLE dwb_day_user_summary_inc partition(date='2017-11-27') SELECT
`user`,
sum(case when `behavior`='pv' then 1 else 0 end) as pv_count,
sum(case when `behavior`='buy' then 1 else 0 end) as buy_count
FROM dwd_userbehavior
WHERE `date` = '2017-11-27'
GROUP BY `user`;
```

ods 层结果展示：

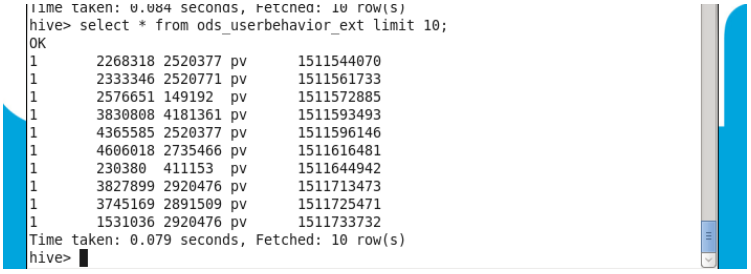


表 dwd_userbehavior:

```
hive> select * from dwd_userbehavior limit 10;
OK
1       2268318 2520377 pv      2017-11-24      09
1       2333346 2520771 pv      2017-11-24      14
1       2576651 149192  pv      2017-11-24      17
1       3830808 4181361 pv      2017-11-24      23
1       4365585 2520377 pv      2017-11-24      23
1       4606018 2735466 pv      2017-11-25      05
1       230380  411153  pv      2017-11-25      13
1       3827899 2920476 pv      2017-11-26      08
1       3745169 2891509 pv      2017-11-26      11
1       1531036 2920476 pv      2017-11-26      14
Time taken: 0.725 seconds, Fetched: 10 row(s)
hive>
```

表 dws_day_summary:

```
Time taken: 0.725 seconds, Fetched: 10 row(s)
hive> select * from dws_day_summary limit 10;
OK
2017-11-27      14003   122488
Time taken: 0.085 seconds, Fetched: 1 row(s)
```

然后，自己也在 hadoop 中，针对各时间行为、商品、用户对象做了相关的一些 dws 表。但是用 hive 时的代码并未及时保存下来，只有相关的表、结果截图和 mysql 实现的代码。

```
hive> show databases;
OK
business
business1
default
taobao1
taobao_dw
test
Time taken: 1.391 seconds, Fetched: 6 row(s)
hive> use taobao_dw;
OK
Time taken: 0.029 seconds
hive> show tables;
OK
dwd_userbehavior
dws_day_summary
dws_day_user
dws_goods
dws_hour_count
ods_userbehavior_ext
Time taken: 0.055 seconds, Fetched: 6 row(s)
```

在 database taobao_dw 下存在的表。

在表中 dws_hour_count 有三个指标，分别是：2022-11-27 日日期、当天各时段、按时段分组计算的总商品行为数。

Mysql 实现：

```sql
select
  '2022-11-27' as date,
  distinct (time) as hor,
  count(behavior) over(partition by hor order by hor) as time_bhvr_count
  from taobao
  where date = '2022-11-27'
```

Hive 中表结果展示：

```
hive> select * from dws_hour_count limit 10;
OK
2017-11-27      6       16039
2017-11-27      20      9318
2017-11-27      19      9146
2017-11-27      15      4740
2017-11-27      0       9904
2017-11-27      2       9339
2017-11-27      4       14877
2017-11-27      22      10193
2017-11-27      17      8514
2017-11-27      13      1236
Time taken: 0.09 seconds, Fetched: 10 row(s)
```

接下来是表 dws_goods，该表中存在三个指标：商品类别编号、商品行为、各商品类型中各行为的计数。

Mysql 实现：

```sql
select
  category,
  behavior,
  count(item) as ctg_bhvr_count
  from taobao where date = '2022-11-25'
  group by category, behavior
```

Hive 中表结果展示：

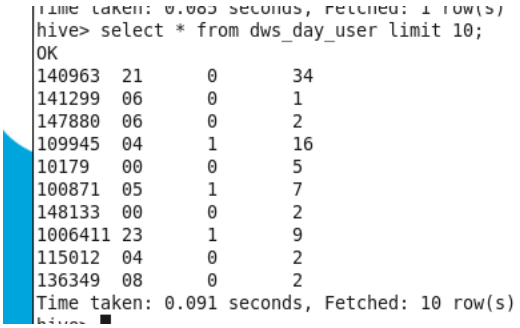
```
hive> select * from dws_goods limit 10;
OK
466400  cart    5
2031969 buy     1
4107781 cart    1
3314621 pv      6
3645362 fav     9
2642752 buy     1
1154246 fav     3
1249923 pv      20
2253970 cart    1
1617627 pv      1
Time taken: 0.087 seconds, Fetched: 10 row(s)
```

表 dws_day_user 中包含四个指标：用户编号、时段、各用户在该时段购买的次数、各用户在该时段收藏/加购的次数。

Mysql 实现：

```sql
select
  user,
  time,
  sum(case when behavior = 'buy' then 1 else 0 end) as user_buy_count,
  sum(case when behavior = 'pv' or 'fav' then 1 else 0 end) as user_favpv_count
  from taobao
  where date = '2022-11-27'
  group by user, time
```

Hive 中表结果展示：



四．几点补充

1. 对于 dws 中，设置 where date='2022-11-27'的操作。

   数据处理的方式一般有两种比较经典的：批量数据处理和实时数据处理。

   在批量数据处理中是按天把前一天的数据跑全，很多业务系统是按天的跑出来并下载下载下来，按天的原因是等待其他的系统一起跑完后，比如抖音和今日头条，再在凌晨下载下来。在实际操作中一般用 date_format('YYYY-mm-dd',now())-1 自动设为前一天。

2. 宽表和 dws 一些区别：

   首先无论是宽表还是 dws 表，都是对 dwd 表的一个汇总。

   然后，在来源上，其实也可以都当成 dws 表，但在业务中经常会做一个抽象；比如，在对象层面上构建一些数据指标，有很多维度。比如用户有很多维度，查询次数、购买次数、点击次数等，这种对象我们一般不把它放在很多个 dws 表中，而是放在一个宽表中，但并不是为了 dwb 为创建，而只是因为这个对象的指标太多了，比如有几十个指标，在未来我们都把指标往里面塞。

   其好处是，在这一个宽表里我可以做很多分析，因为这个宽表里维度很多，任意的搭配起来分析。

   总之，它是一个特殊的 dws 表，特殊在它针对于主要的业务对象创建，第二是它的指标比 dws 表中的指标多很多。