Hadoop学习之HIVE安装

1. **HIVE简介：**

HIVE在hadoop生态圈中属于数据仓库的角色，能够管理hadoop中的数据，同时可以查询hadoop中的数据

本质上，hive是一个sql解析引擎，Hive可以把sql查询转换为MapReduce中的Job类运行

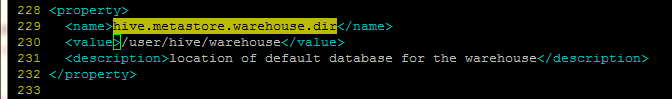
Hive有一套映射工具，可以把sql转换为MapReduce的Job，可以把sql中的表，字段转换为HDFS中的文件及文件中的列

这套映射工具称之为metastore，一般存放在derby、mysq中

Hive在hdfs中默认位置为：/user/hive/warehouse

位置是由配置文件hive-site.xml中的属性hive.metastore.warehouse.dir决定的

在文件原位置的229行



四种数据模型:表(table)、外部表(external table)、分区（partition）和桶(Bucket)!

表和外部表的区别：

1. 创建表有两个步骤：表创建过程、数据加载。（两个过程可以用一个语句完成），数据加载过程中实际数据移动到数据仓库目录下，之后的数据访问将会在数据仓库目录完成。

删除表时，表中的数据和元数据都会删除!!

外部表的创建只有一个步骤：加载数据和创建表同时完成。实际数据存储在创建语句LOCATION指定的HDFS中，不会移到数据仓库目录中。

删除外部表时仅删除元数据！！

Hive的元数据存储:

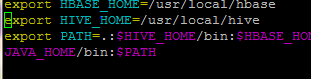
元数据存储在mysql或者derby等RDBMS中，Hive连接Derby有三种模式：

1. Single User Mode：一般用于单元测试
2. Multi User Mode ：网络连接数据库，最常用
3. Remote Server Mode ：用于非java客户端访问元数据库，在服务器启动MetaStoreServer，客户端使用Thrift协议访问。
4. **Hive的安装(hive可理解为客户端工具)**
   1. 使用winscp将hive-0.9.0.tar.gz 复制到linux的/usr/local、目录下
   2. 解压,重命名为hive,并加入到环境变量中

#tar –xzvf hive-0.9.0.tar.gz

#mv hive-0.9.0 hive

#vim /etc/profile



#source /etc/profile #使得配置文件生效

* 1. 切换到hive的配置文件目录

#cd /usr/local/hive/conf

* 1. 复制一份模板并修改配置文件

# cp hive-default.xml.template hive-site.xml

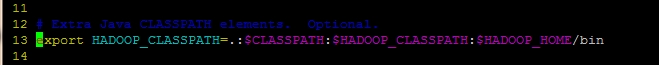
# cp hive-env.sh.template hive-env.sh

修改hadoop的配置文件hadoop-env.sh

#vim /usr/local/hadoop/conf/hadoop-env.sh

添加这一行

export HADOOP\_CLASSPATH=.:$CLASSPATH:$HADOOP\_CLASSPATH:$HADOOP\_HOME/bin



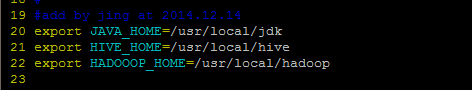
修改$HIVE\_HOME/bin下的hive-config.sh增加以下三行

#vim /usr/local/hive/bin/hive-config.sh

export JAVA\_HOME=/usr/local/jdk

export HIVE\_HOME=/usr/local/hive

export HADOOOP\_HOME=/usr/local/hadoop



* 1. 启动hive

#/usr/local/hive/bin/hive

操作与mysql类似

PS：hive在哪个目录下启动的就会在哪个目录下建立一个derby.log的文件，在其他位置是看不到该处建立的表的，即如果在/home/test/目录下启动的hive，则在其他目录下启动时是看不到在/home/test/处启动的hive的操作的

derby.log 默认只让一个用户登录，不方面。所以改换为在mysql下存储

* 1. 安装mysql

先将本机已经存在的mysql卸载掉再安装

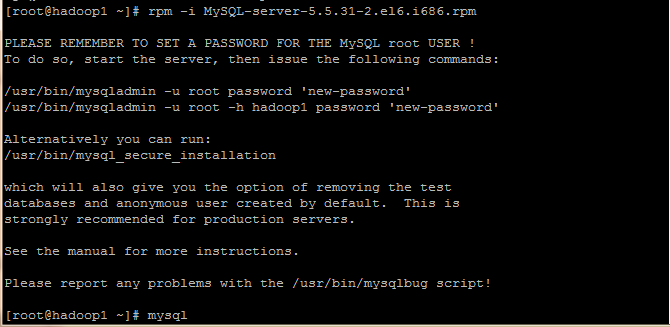
#rpm –qa | grep mysql #检查mysql的安装情况

#yum remove mysql

#rpm –e mysql-lib-5.1.66.xxx --nodeps 删除mysql存在的依赖包

安装mysql服务端

#rpm –I MySql-server-xxxx.rpm



安装mysql客户端

#rpm –i MySql-client-xxx.rpm

启动mysql

#mysql\_safe &

Ctrl+c

# mysql\_secure\_installation #配置密码

配置密码为mysql

客户端登录mysql

#mysql –u root

* 1. 使用mysql作为hive的metastore
     1. 将mysql的jdbc驱动放置到hive的lib目录下
     2. 修改hive-sizte.xml文件

添加一下内容

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://hadoop1:3306/hive?createDatabaseIfNotExist=true</value>

</property>

<property>

<name>javax.jdo.option.ConnectionDriverName</name>

<value>com.mysql.jdbc.Driver</value>

</property>

<property>

<name>javax.jdo.option.ConnectionUserName</name>

<value>root</value>

</property>

<property>

<name>javax.jdo.option.ConnectionPassword</name>

<value>mysql</value>

</property>

1. **Hive QL详解：**

表的操作语句，数据类型不区分大小写,但SerDe和属性名区分大小写，注释是单引号的字符串

DDL操作

1. 创建表

Create [external] table [if not exists] table\_name

[(

Col\_name data\_type [comment col\_comment]

)] [comment table\_comment];

1. 如果需要存储为纯文本文件，使用STORED AS TEXTFILE
2. 如果数据需要压缩,使用STORED AS SEQUENCEFILE
3. INPUTFORMAT和OUTPUTFORMAT定义一个与inputformat和outputformat类相对应的名字作为字符串
4. Hive 支持带有分区(partition)的表，创建的时候使用PARTITIONED BY语句，可有多个分区，每个分区一个目录。
5. 表和分区都能对某列CLUSTERED BY操作，将若干个列放在一个桶中。
6. 可以利用SORT BY列存储数据，提高查询性能

例1：创建普通表

create table page1(

pageId INT,

pageUrl STRING,

userid BIGINT,

referrer\_url STRING,

IP STRING COMMENT "IP address of the user"

) comment 'this is the page1 table';

例2：添加表分区,并用制表符区分同一行的不同字段

create table page1(

pageId INT,

pageUrl STRING,

userid BIGINT,

referrer\_url STRING,

IP STRING COMMENT "IP address of the user")

comment 'this is the page1 table'

PARTITIONED BY(dt STRING,country STRING)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\001'

STORED AS SEQUENCEFILE;

例3：添加聚类存储,将列按照userid进行分区并划分到不同的桶中,按照pageid值得大小进行排序存储。这样允许用户通过userid属性高效的对集群列进行采样。

create table page1(

pageId INT,

pageUrl STRING,

userid BIGINT,

referrer\_url STRING,

IP STRING COMMENT "IP address of the user")

comment 'this is the page1 table'

PARTITIONED BY(dt STRING,country STRING)

CLUSTERED BY(userid) SORTED BY(pageid) INTO 32 BUCKETS

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\001'

COLLECTION ITEMS TERMINATED BY '\002'

MAP KEYS TERMINATED BY '\003'

STORED AS SEQUENCEFILE;

例4：指定存储路径，通过LOCATION

Create external table page\_view(

viewTime INT,

userid BIGINT,

page\_url STRING,

referrer\_url string,

ip string comment 'ip address',

country string )

comment 'this is the staging page view table'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\054'

STORED AS TEXTFILE

LOCATION '<hdfs\_location>';

1. 删除表

Drop table table\_name;

用于删除表的元数据和数据，如果配置trash，数据删除到Trash/Current目录，元数据完全丢失。当删除external定义的表时，表中的数据不会从文件系统中删除

1. 修改表

例1：重命名表

alter table page1 rename to page2;

例2：改变列名、类型、位置、注释

Alter table table\_name change [column] col\_old\_name col\_new\_name column type

[comment col\_comment][FIRST | AFTER column\_name]

将列名userid改为uid，数据类型为string

alter table page\_view change userid uid STRING;

将列名uid改为userid，数据类型为string，并置于ip后

例3：增加\更新列

alter table page2 ADD COLUMNS (name STRING comment 'name ');

alter table page2 replace COLUMNS (name STRING comment 'name ');只有在使用native的SerDE时才可以这样做。

例4：增加表属性

alter table page2 set tblproperties table\_properties

table\_properties:

: (property\_name=property\_value, property\_name=property\_value, property\_name=property\_value)

用户可以用这个命令向表中增加元数据，目前last\_modified\_user、last\_modified\_time属性由hive自动管理的。用户也可以向列表增加自己的属性，可以使用DESCRIBE EXTRENDED TABLE获取信息。

例5：增加SerDe属性

Alter table table\_name

Set SERDE serde\_class\_name

[with serdeproperties serde\_properties]

Alter table table\_name

Set serdeproperties serde\_properties

serde\_properties:

: (property\_name=property\_value, property\_name=property\_value, property\_name=property\_value)

例6：改变表文件格式和组织(只修改表的物理存储属性)

Alter table table\_name set FILEFORMAT file\_format

Alter table table\_name CLUSTERED BY (col\_name,col\_name,…)

[SORTED BY (col\_name,…)] INTO num\_buckets BUCKETS

这些命令只能修改hvie的元数据，不能重组或格式化现有的数据，用户应该实际数据的分布符合元数据的定义。

1. 表分区操作语句

查询时会对整个表扫描，为了只对关心的部分数据进行扫描，hive引入分区、相当于简单的索引功能

创建时指定模式、字段指定虚拟列，表中不存在、可指定多级结构，相当于对目录嵌套、在创建完成后使用之前还需通过alter table 语句添加具体的分区目录才能使用

Hive表分区的命令主要包括创建分区、增加分区和删除分区

例1：增加分区

Alter table table\_name add partition\_spec [LOCATION ' location1' ] partition\_spec [LOCATION ' location1' ]

Partition\_spec:

: PARTITION (partition\_col=partition\_col\_value, partition\_col=partition\_col\_value,…)

分区名是字符串时加引号。

Alter table page\_view add PATITION (de='2015-2-4',country='us')

Location '/path/to/us/part080808';

例2：删除分区

Alter table page\_view drop PATITION (de='2015-2-4',country='us')

PS：1.当没有声明表模式的时候不能为表指定具体的分区

2.分区名不能与表属性名重复

1. 创建、删除视图

Create view [if not exists] view\_name [(col\_name [comment col\_comment],…)]

As select

视图是只读的，不能用于LOAD\INSERT\ALTER 的目标

视图可能包含order by或limit子句，如果一个引用了视图的查询也包含了这些子句，那么在执行这些子句时，首先查看视图子句，然后返回结果按视图中语句执行

删除视图：

Drop view view\_name;

1. 创建、删除函数

Create temporary function function\_name;

该语句创建了一个由类名实现的函数，在hive中可以持续使用该函数查询，也可以使用Hive类路径中的任何类。用户可以执行add files 语句将函数类添加到类路径中

Drop temporary function function\_name;

1. 展示描述语句
2. 显示表

Show tables identifier\_with\_wildcards

1. 显示分区

Show partitions table\_name;

1. 显示表、分区扩展

Show table external [in | from databases\_name] like identifier\_with\_wildcards [partition(partition\_desc)]

1. 显示函数

Show functions;

1. 描述表、列

Describe [extended] table\_name [dot col\_name]

Describe [extended] table\_name [dot col\_name([DOT field\_name] | [DOT ‘$elem$’] | [DOT $key$’]|[DOT ‘$value$’])\*]

通常只用于调试，不用于平时使用中。

[DOT ‘$elem$’] 🡪数组元素

[DOT $key$’] 🡪 图的主键

[DOT ‘$value$’]) 🡪 图的属性

1. 描述分区

Describe [extended] table\_name partition\_spec;

数据操作（DML）

1. 向数据表中加载文件

不会对数据进行任何转换，Load只是将数据复制或移动到Hive表对应的位置

语法：

Load data [LOCAL] INPATH 'filepath' [OVERWRITE]

INTO TABLE table\_name

[PARTITION (partcol1=val1,part2=val2,…)]

PS:1.指定LOCAL,load命令会查找本地文件系统的filepath,

2.filepath:支持相对路径、绝对路径、完整URI，目标可以是表或分区;

如果表包含分区，必须指定每个分区名;

Filepath可以引用文件或目录。

3.没有指定LOCAL，filepath 指向URI，Hive会直接使用URI，没有schema或authority，Hive会使用hadoop配置文件中定义的schema或authority，fs.default.name 🡪 指定nameNode的URI；

如果是相对路径，Hive会相对/user/解释。

4.使用OVERWRITE,目标表中的内容会被删除，并将filepath指向的文件\目录中的内容添加到表(或分区)中，如果目标表(或分区)中有文件，且与filepath中文件名冲突，现有文件会被替换。

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

1. 将查询结果插入Hive表中

语法：

INSERT OVERWRITE TABLE table\_name [PARTITION (partcol1=val1,partcol2=val2,…)]

Select\_statement1 from from\_statement

Hive extension (multiple inserts)

FROM from\_statement

INSERT OVERWRITE TABLE table\_name1 [PARTITION (partcol1=val1,partcol2=val2,…)]

select\_statement1

[INSERT OVERWRITE TABLE table\_name2 [PARTITION (partcol1=val1,partcol2=val2,…)]]

Hive extension (dynamic partition inserts):

INSERT OVERWRITE TABLE tableName PARTITION (partcol1=[val1], partcol2=[val2],…)

select\_statement1 from from\_statement

插入可以针对一个表或分区操作，如果对表进行了划分，在插入时要指定划分列的属性值确定分区。每个select 语句的结果会被写入选择的表或分区中，OVERWRITE 关键字会强制将输出结果写入，输出格式和序列化方式由表的元数据决定。在Hive进行多表插入可以减少数据扫描的次数。

1. 将查询的结果写入到文件系统

INSERT OVERWRITE [LOCAL] DIRECTORY directory1 SELECT … FROM ….

Hive extension (multiple inserts)

FROM from\_statement

INSERT OVERWRITE [LOCAL] DIRECTORY directory1 select\_statement1

[INSERT OVERWRITE [LOCAL] DIRECTORY directory1 select\_statement2]…

目录可以是完整的URI,如果使用LOCAL，Hive会将数据写入到本地文件系统中。

PS：数据写入文件系统时会进行文本序列化，并且每列^A区分，换行表示一行数据结束。如果任一列不是原始类型，这些列会被序列化为json格式

Sql 操作(标准的select语句)

语法：

SELECT [ALL | DISTINCT] select\_expr,select\_expr,…

FROM table\_reference

[WHERE where\_condition]

[GROUP BY col\_list]

[ CLUSTER BY col\_list

| [DISTRIBUTE BY col\_list] [SORT BY col\_list]

]

[LIMIT number]

PS:

1. table\_reference 指明查询的输入，可以是表、视图或子查询

select \* from t1

1. WHERE

Where\_condition 是一个布尔表达式。跟sql中的格式一样

1. ALL和DISTINCT

ALL 和DISTINCT 可以定义重复的行是否返回，默认为ALL。

1. LIMIT

LIMIT 可以控制输出的记录数，随机选取检索结果中相应数目输出：

Select \* from t1 limit 5 🡪输出最多5条记录

1. 使用正则表达式

Select ‘(ds|hr)?+.+’ from sales

1. 基于区分的查询

如果一个表是使用PARTITIONED BY 语句生成的，查询可以对输入进行’剪枝’，只对表的相关部分扫描。Hive现只where中指定分区剪枝。

例1：如果表page\_view按照date列的值进行分区，查询检索日期 2010-03-01 --- 2010-03-31。

Select page\_view.\* from page\_view where page\_view.date >= ‘2010-03-01’ and page\_view.date <= ‘2010-03-31’

1. HAVING

Hive 不支持having 语句，但可以使用子查询实现

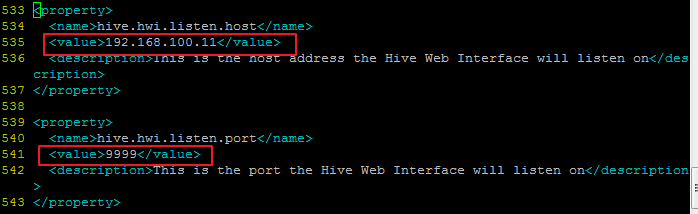
1. Group by和JOIN的作用与SQL相同。

Hive网络(WEB UI)接口

1. 分离查询的执行

命令行下，要执行多个查询就要打开多个终端，通过网络接口，可以同时执行多个查询，网络接口可以在网络服务器上管理会话(session)

1. 远程执行Hive WEB UI，需要在Hive服务端配置文件中配置IP地址和端口号



1. 启动hive网络接口服务

#hive --service hwi

1. 通过浏览器访问：<http://masterIP:port/hwi> 本例中<HTTP://192.168.100.11:9999/hwi>

单击Browse Schema 可以查看当前hive的数据库

Hive查询：首先创建一个session会话，可通过list session连接查看所有的session。

当hive重启后，session信息将全部丢失，会话与认证关联。

在list session处点击action项进行查询操作。