Hive 是一种底层封装了Hadoop 的数据仓库处理工具，使用类SQL 的HiveQL 语言实现数据查询，所有Hive 的数据都存储在Hadoop 兼容的文件系统（例如，Amazon S3、HDFS）中。Hive 在加载数据过程中不会对数据进行任何的修改，只是将数据移动到HDFS 中Hive 设定的目录下，因此，Hive 不支持对数据的改写和添加，所有的数据都是在加载的时候确定的。Hive 的设计特点如下。

● 支持索引，加快数据查询。

● 不同的存储类型，例如，纯文本文件、HBase 中的文件。

● 将元数据保存在关系数据库中，大大减少了在查询过程中执行语义检查的时间。

● 可以直接使用存储在Hadoop 文件系统中的数据。

● 内置大量用户函数UDF 来操作时间、字符串和其他的数据挖掘工具，支持用户扩展UDF 函数来完成内置函数无法实现的操作。

● 类SQL 的查询方式，将SQL 查询转换为MapReduce 的job 在Hadoop集群上执行。

作者：yuan daisy

链接：https://www.zhihu.com/question/21677041/answer/78289309

来源：知乎

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

1. Hive中的表是纯逻辑表，就只是表的定义等，即表的元数据。Hive本身不存储数据，它完全依赖HDFS和MapReduce。这样就可以将结构化的数据文件映射为为一张数据库表，并提供完整的SQL查询功能，并将SQL语句最终转换为MapReduce任务进行运行。 而HBase表是物理表，适合存放非结构化的数据。

2. Hive是基于MapReduce来处理数据,而MapReduce处理数据是基于行的模式；HBase处理数据是基于列的而不是基于行的模式，适合海量数据的随机访问。

3. HBase的表是疏松的存储的，因此用户可以给行定义各种不同的列；而Hive表是稠密型，即定义多少列，每一行有存储固定列数的数据。

4. Hive使用Hadoop来分析处理数据，而Hadoop系统是批处理系统，因此不能保证处理的低迟延问题；而HBase是近实时系统，支持实时查询。

5. Hive不提供row-level的更新，它适用于大量append-only数据集（如日志）的批任务处理。而基于HBase的查询，支持和row-level的更新。

6. Hive提供完整的SQL实现，通常被用来做一些基于历史数据的挖掘、分析。而HBase不适用与有join，多级索引，表关系复杂的应用场景。hbase是列式存储的数据库，而hive（小蜜蜂）是一个简化操作hbase的接口，类似于sql语句去操作hbase。没有万能，只能说在哪种场合适合用列式的存储（一般是用在数据量巨大，且数据间关系简单，比如K，V对。）或者关系型的数据库（一般用在数据关系严密，数据量相对较小的情况下）

作者：左良

链接：https://www.zhihu.com/question/21677041/answer/79969051

来源：知乎

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

hbase是db， hive 是工具，不是一个次元的

hadoop由大数据存储hdfs和大数据计算Map/Reduce构成， hbase可以理解为 NOSQL存储， 类似与hdfs的改进， hive可以理解为简单化的大数据计算和分析， 类似于对Map/Reduce的封装

**hive shell 命令使用笔记：**

$ $HIVE\_HOME/bin/hive shell #打开hive命令行

hive(default)> show databases #查看已有数据库(default为现在所在的数据库)

OK

database\_name

default

event\_db

Time taken: 2.293 seconds, Fetched: 2 row(s)

hive(default)> show databases like ‘de\*’ #模糊查询部分数据库

hive(default)> show tables #查看当前数据库所有的表

OK

tab\_name

t\_hadoop

t\_hft

Time taken: 0.085 seconds, Fetched: 2 row(s)

hive(default)> use event\_db #更换数据库

OK

Time taken: 0.049 seconds

hive (event\_db)>

hive (default)> create database if not exists movie\_ratings01; #创建数据库

OK

Time taken: 0.637 seconds

hive (default)> create external table movie\_ratings01.users(

> id int,sex string,age int,occupation string ,zipCode string )

> partitioned by (day string) row format delimited fields terminated by '::' ;

OK

Time taken: 1.223 seconds

hive (movie\_ratings01)> desc users ;

OK

col\_name data\_type comment

id int

sex string

age int

occupation string

zipcode string

day string

hive (movie\_ratings01)> load data local inpath '/data/hive-1.1.0-cdh5.5.0/testinput/users.dat' overwrite into table users partition (day='20160602');

Loading data to table movie\_ratings01.users partition (day=20160602)

Partition movie\_ratings01.users{day=20160602} stats: [numFiles=1, numRows=0, totalSize=134368, rawDataSize=0]

OK

Time taken: 2.461 seconds

hive (movie\_ratings01)> alter table users change zipcode zipcode int ; #修改列

hive (movie\_ratings01)> alert table users set serdeproperties('field.delim'=':+'); #修改分割符

(

修改数据库

修改数据库属性：

ALTER (DATABASE|SCHEMA) database\_name

SET DBPROPERTIES (property\_name=property\_value, …);

修改数据库属主：

ALTER (DATABASE|SCHEMA) database\_name

SET OWNER [USER|ROLE] user\_or\_role;

删除数据库

DROP (DATABASE|SCHEMA) [IF EXISTS] database\_name

[RESTRICT|CASCADE];

默认情况下，Hive不允许删除一个里面有表存在的数据库，如果想删除数据库，要么先将数据库中的表全部删除，要么可以使用CASCADE关键字，使用该关键字后，Hive会自己将数据库下的表全部删除。RESTRICT关键字就是默认情况，即如果有表存在，则不允许删除数据库。

)

<http://lxw1234.com/archives/2015/06/265.htm> 更多使用方法

查询平均评分大于4的：电影id，评价人数，平均评分，电影名称

select avg\_t.movieid,avg\_t.rating\_num,avg\_t.avg\_r,m.title from (select movieid,count(userid) as rating\_num,avg(rating) as avg\_r from ratings group by movieid)as avg\_t join movies as m on m.movieid=avg\_t.movieid where avg\_t.avg\_r > 4 ;

Time taken: 51.427 seconds, Fetched: 370 row(s) #直接在hive中执行的时间

平均评分最高的10条（需要有50人评论以上的）：

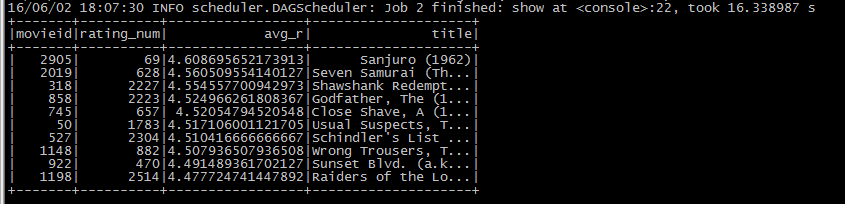
select avg\_t.movieid,avg\_t.rating\_num,avg\_t.avg\_r,m.title from (select movieid,count(userid) as rating\_num,avg(rating) as avg\_r from ratings group by movieid)as avg\_t join movies as m on m.movieid=avg\_t.movieid where avg\_t.avg\_r > 4 and avg\_t.rating\_num > 50 order by avg\_t.avg\_r DESC limit 10 ;

Time taken: 55.05 seconds, Fetched: 10 row(s) #直接在hive中执行的时间

下面是使用spark-shell运行hive查询的结果:

[hadoop@master spark-1.5.0-bin-hadoop2.6]$ spark-shell --jars ../hive-1.1.0-cdh5.5.0/lib/mysql-connector-java-5.1.33.jar --master spark://master:7077

select avg\_t.movieid,avg\_t.rating\_num,avg\_t.avg\_r,m.title from (select movieid,count(userid) as rating\_num,avg(rating) as avg\_r from movie\_ratings01.ratings group by movieid)as avg\_t join movie\_ratings01.movies as m on m.movieid=avg\_t.movieid where avg\_t.avg\_r > 4 and avg\_t.rating\_num > 50 order by avg\_t.avg\_r DESC limit 10



spark-submit 提交方式：

spark-submit --master spark://master:7077 ./work/TomcatLogRealtimeAnalysis-0.0.1.jar spark://master:7077 10 --class com.ai.demo.MovieRatingSparkHiveDemo --driver-class-path ../hive-1.1.0-cdh5.5.0/lib/mysql-connector-java-5.1.33.jar