1.什么是hive

1、Hive 由 Facebook 实现并开源

2、是基于 Hadoop 的一个数据仓库工具

3、可以将结构化的数据映射为一张数据库表

4、并提供 HQL(Hive SQL)查询功能

5、底层数据是存储在 HDFS 上

6、Hive的本质是将 SQL 语句转换为 MapReduce 任务运行

7、使不熟悉 MapReduce 的用户很方便地利用 HQL 处理和计算 HDFS 上的结构化的数据，适用于离线的批量数据计算。

数据仓库之父比尔·恩门（Bill Inmon）在 1991 年出版的“Building the Data Warehouse”（《建 立数据仓库》）一书中所提出的定义被广泛接受——数据仓库（Data Warehouse）是一个面 向主题的（Subject Oriented）、集成的（Integrated）、相对稳定的（Non-Volatile）、反映历史 变化（Time Variant）的数据集合，用于支持管理决策(Decision Making Support)。

Hive 依赖于 HDFS 存储数据，Hive 将 HQL 转换成 MapReduce 执行，所以说 Hive 是基于 Hadoop 的一个数据仓库工具，实质就是一款基于 HDFS 的 MapReduce 计算框架，对存储在 HDFS 中的数据进行分析和管理

1.1 Hive 特点

优点：

1、可扩展性,横向扩展，Hive 可以自由的扩展集群的规模，一般情况下不需要重启服务 横向扩展：通过分担压力的方式扩展集群的规模 纵向扩展：一台服务器 cpu i7-6700k 4核心8线程，8核心16线程，内存64G => 128G

2、延展性，Hive 支持自定义函数，用户可以根据自己的需求来实现自己的函数

3、良好的容错性，可以保障即使有节点出现问题，SQL 语句仍可完成执行

缺点：

1、Hive 不支持记录级别的增删改操作，但是用户可以通过查询生成新表或者将查询结 果导入到文件中（当前选择的 hive-2.3.2 的版本支持记录级别的插入操作）

2、Hive 的查询延时很严重，因为 MapReduce Job 的启动过程消耗很长时间，所以不能 用在交互查询系统中。

3、Hive 不支持事务（因为不没有增删改，所以主要用来做 OLAP（联机分析处理），而 不是 OLTP（联机事务处理），这就是数据处理的两大级别）。

2. hive架构

2.1 底层的Driver： 驱动器Driver，编译器Compiler，优化器Optimizer，执行器Executor

Driver 组件完成 HQL 查询语句从词法分析，语法分析，编译，优化，以及生成逻辑执行 计划的生成。生成的逻辑执行计划存储在 HDFS 中，并随后由 MapReduce 调用执行

Hive 的核心是驱动引擎， 驱动引擎由四部分组成：

(1) 解释器：解释器的作用是将 HiveSQL 语句转换为抽象语法树（AST）

(2) 编译器：编译器是将语法树编译为逻辑执行计划

(3) 优化器：优化器是对逻辑执行计划进行优化

(4) 执行器：执行器是调用底层的运行框架执行逻辑执行计划

2.2 元数据存储系统： RDBMS MySQL

元数据 :通俗的讲，就是存储在 Hive 中的数据的描述信息。

Hive 中的元数据通常包括：表的名字，表的列和分区及其属性，表的属性（内部表和 外部表），表的数据所在目录

Metastore 默认存在自带的 Derby 数据库中。缺点就是不适合多用户操作，并且数据存 储目录不固定。数据库跟着 Hive 走，极度不方便管理

解决方案：通常存我们自己创建的 MySQL 库（本地 或 远程）

Hive 和 MySQL 之间通过 MetaStore 服务交互

2.3. 表的分类

Hive 中的表分为 内部表、外部表、分区表和 Bucket 表

内部表和外部表的区别：

删除内部表，删除表元数据和数据

删除外部表，删除元数据，不删除数据

内部表和外部表的使用选择：

大多数情况，他们的区别不明显，如果数据的所有处理都在 Hive 中进行，那么倾向于 选择内部表，但是如果 Hive 和其他工具要针对相同的数据集进行处理，外部表更合适。

使用外部表访问存储在 HDFS 上的初始数据，然后通过 Hive 转换数据并存到内部表中

使用外部表的场景是针对一个数据集有多个不同的 Schema

通过外部表和内部表的区别和使用选择的对比可以看出来，hive 其实仅仅只是对存储在 HDFS 上的数据提供了一种新的抽象。而不是管理存储在 HDFS 上的数据。所以不管创建内部 表还是外部表，都可以对 hive 表的数据存储目录中的数据进行增删操作。

分区表和分桶表的区别：

Hive 数据表可以根据某些字段进行分区操作，细化数据管理，可以让部分查询更快。同 时表和分区也可以进一步被划分为 Buckets，分桶表的原理和 MapReduce 编程中的 HashPartitioner 的原理类似。

分区和分桶都是细化数据管理，但是分区表是手动添加区分，由于 Hive 是读模式，所 以对添加进分区的数据不做模式校验，分桶表中的数据是按照某些分桶字段进行 hash 散列 形成的多个文件，所以数据的准确性也高很多

3. 数据类型

3.1 基本类型

描述	示例	

描述		
boolean	true/false	TRUE
tinyint	1字节的有符号整数	-128~127 1Y
smallint	2个字节的有符号整数，-32768~32767	1S
int	4个字节的带符号整数	1
bigint	8字节带符号整数	1L
float	4字节单精度浮点数	1.0
double	8字节双精度浮点数	1.0
deicimal	任意精度的带符号小数	1.0
String	字符串，变长	“a”,’b’
varchar	变长字符串	“a”,’b’
char	固定长度字符串	“a”,’b’
binary	字节数组	无法表示
timestamp	时间戳，纳秒精度	122327493795
date	日期	‘2018-04-07’

Hive 支持关系型数据中大多数基本数据类型,和其他的SQL语言一样，这些都是保留字。需要注意的是所有的这些数据类型都是对Java中接口的实现，因此这些类型的具体行为细节和Java中对应的类型是完全一致的。例如，string类型实现的是Java中的String，float实现的是Java中的float，等等。

3.2 复杂类型

类型	描述	示例
array	有序的的同类型的集合	array(1,2)
map	key-value,key必须为原始类型，value可以任意类型	map(‘a’,1,’b’,2)
struct	字段集合,类型可以不同	struct(‘1’,1,1.0), named_stract(‘col1’,’1’,’col2’,1,’clo3’,1.0)

4. 存储格式

1. textfile

textfile为默认格式，存储方式为行存储。数据不做压缩，磁盘开销大，数据解析开销大。

2. SequenceFile

SequenceFile是Hadoop API提供的一种二进制文件支持，其具有使用方便、可分割、可压缩的特点。

SequenceFile支持三种压缩选择：NONE, RECORD, BLOCK。 Record压缩率低，一般建议使用BLOCK压缩。

3. RCFile

一种行列存储相结合的存储方式。

4. ORCFile

数据按照行分块，每个块按照列存储，其中每个块都存储有一个索引。hive给出的新格式，属于RCFILE的升级版,性能有大幅度提升,而且数据可以压缩存储,压缩快 快速列存取。

5. Parquet

Parquet也是一种行式存储，同时具有很好的压缩性能；同时可以减少大量的表扫描和反序列化的时间。

> 来自: https://www.cnblogs.com/qingyunzong/p/8733924.html