Hive数据存储

高校大数据课程系列

ENTER

课程目标

Course objectives

1. 掌握Hive支持数据格式
2. 掌握Hive压缩应用
3. 掌握文件格式与压缩案例
4. 掌握Java调用Hive案例

本章任务

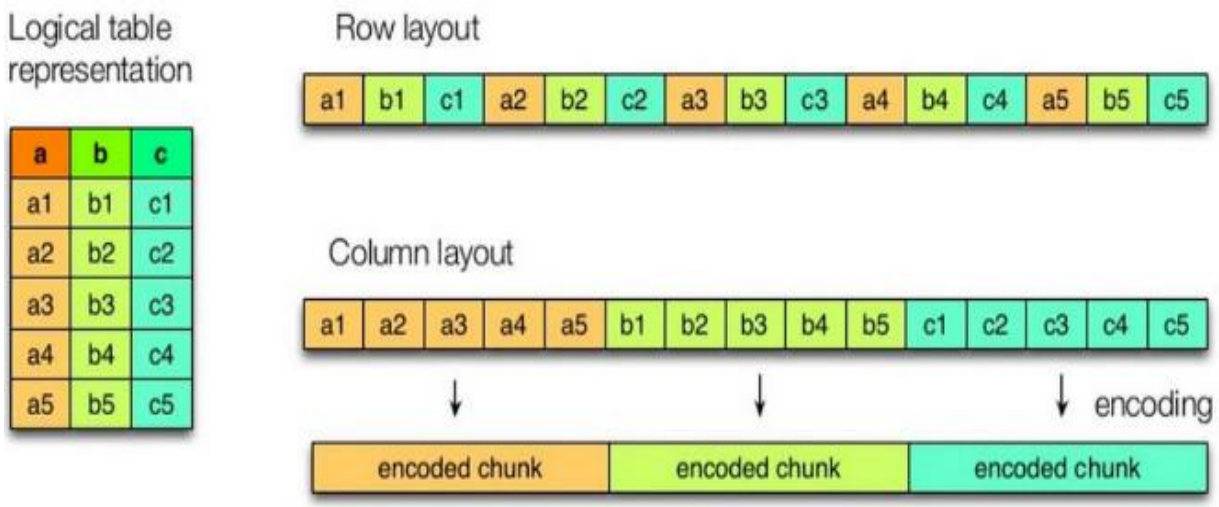
Task of this chapter

1 Hive支持数据格式

2 文件格式与压缩案例

3 Java调用Hive案例

Hive支持数据格式



列式存储与行式存储：

- 左边为逻辑表，右边第一个为行式存储，第二个为列式存储。
- 行存储的特点： 查询满足条件的一整行数据的时候，列存储则需要去每个聚集的字段找到对应的每个列的值，行存储只需要找到其中一个值，其余的值都在相邻地方，所以此时行存储查询的速度更快。
- 列存储的特点： 因为每个字段的数据聚集存储，在查询只需要少数几个字段的时候，能大大减少读取的数据量；每个字段的数据类型一定是相同的，列式存储可以针对性的设计更好的设计压缩算法。
- TEXTFILE和SEQUENCEFILE的存储格式都是基于行存储的；
- ORC和PARQUET是基于列式存储的。

Hive支持数据格式

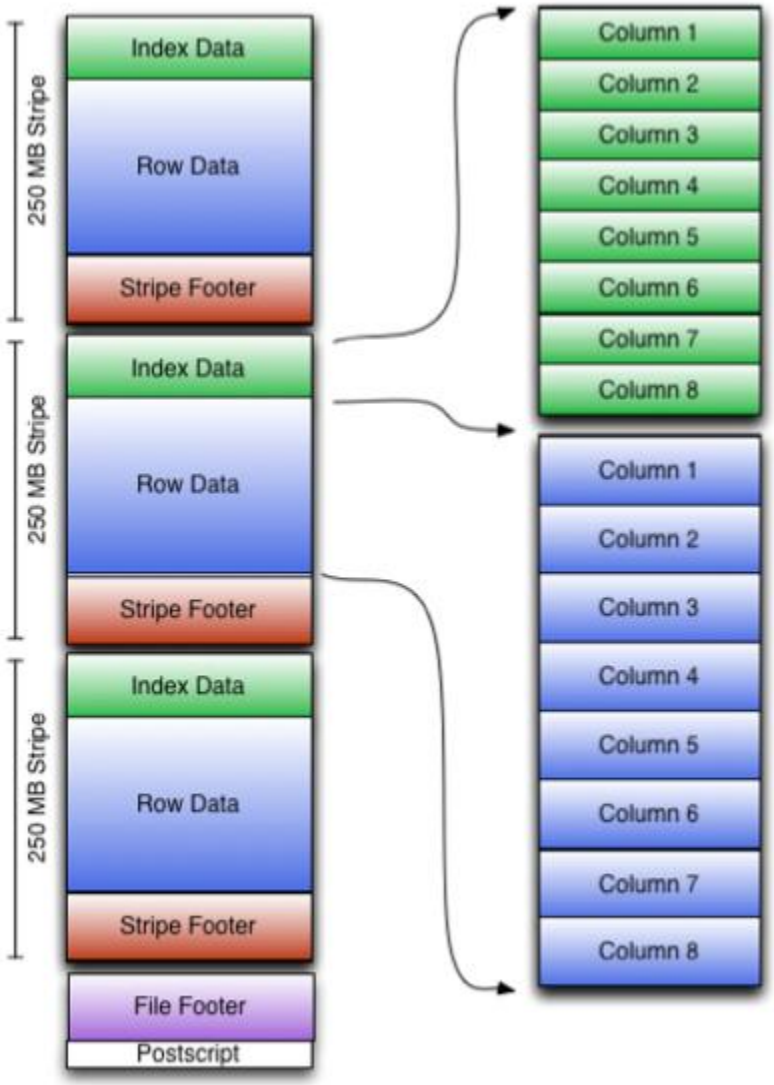
存储格式	描述
TEXTFILE	通过stored as子句存储为纯文本文件。TEXTFILE是通过hive.default.fileformat属性设置的默认文件格式，用户也可通过改变该属性的值设置不同于TEXTFILE的格式。使用DELIMITED子句读取分隔文件。通过使用ESCAPED BY子句（例如ESCAPED BY'\'）为分隔符字符启用转义，如果要处理包含这些分隔符字符的数据，即数据包含分隔符启转义是必须的。也可以使用NULL DEFINED AS子句指定自定义NULL格式（默认为'\N'）。
SEQUENCEFILE	通过stored as子句存储为**可压缩**的。（SequenceFile是Hadoop API提供的一种二进制文件，支持多种压缩格式，如RECORD、BLOCK等，具体看Hive应用的Hadoop版本）
ORC	通过stored as子句存储为ORC文件格式。 支持ACID和基于成本的优化（Cost-Based Optimization，CBO）。存储列级元数据。例：stored as orc TBLPROPERTIES ('transactional'='true');
PARQUET	通过stored as子句存储为Parquet格式（从Hive 0.13.0及更高版本开始）。在使用ROW FORMAT SERDE ... STORED AS INPUTFORMAT ... OUTPUTFORMAT 语法存储。
AVRO	通过stored as子句存储为Avro格式（从Hive 0.14.0及更高版本开始）。有关此选项的更多信息，请参见https://cwiki.apache.org/confluence/display/Hive/AvroSerDe.
RCFILE	通过stored as子句存储为RCFILE格式。有关此选项的更多信息，请参见https://en.wikipedia.org/wiki/RCFile
JSONFILE	过stored as子句存储为JSON文件格式（从Hive 4.0.0及更高版本开始）。
STORED BY	以非Hive本身表格形式存储，可创建或链接到要整合的其它工具的表，例如HBase、Druid或Accumulo支持的表。有关此选项的更多信息，请参见https://cwiki.apache.org/confluence/display/Hive/StorageHandlers
INPUTFORMAT and OUTPUTFORMAT	在设定文件格式中，将相应的InputFormat和OutputFormat类的名称指定为字符串文字。例如：org.apache.hadoop.hive.contrib.fileformat.base64.Base64TextInputFormat，对于LZO压缩，INPUTFORMAT要使用的值是com.hadoop.mapred.DeprecatedLzoTextInputFormat，而OUTPUTFORMAT要使用的值是“org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat”。有关LZO压缩的更多信息，请参见https://cwiki.apache.org/confluence/display/Hive/LanguageManual+LZO

Hive支持数据格式

TEXTFILE格式

默认格式，数据不做压缩，磁盘开销大，数据解析开销大。可结合Gzip、Bzip2使用(系统自动检查，执行查询时自动解压)，但使用这种方式，hive不会对数据进行切分，从而无法对数据进行并行操作。

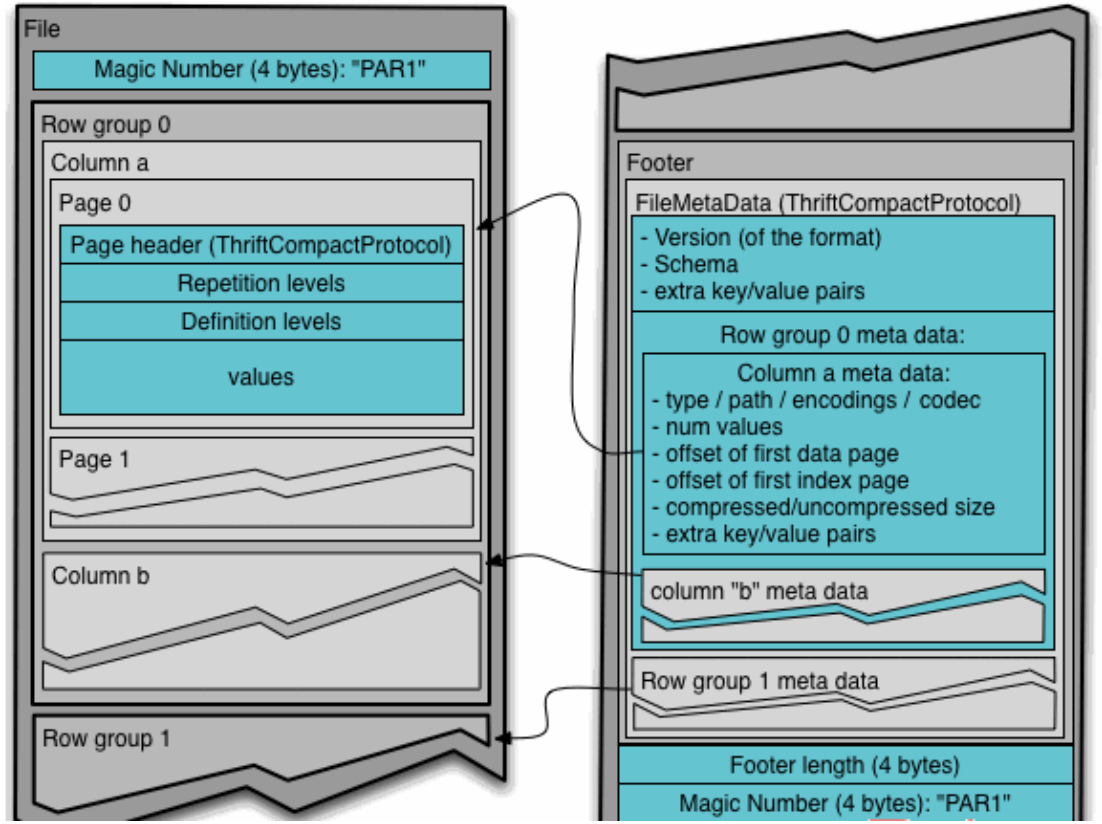
Hive支持数据格式



ORC格式：

- Orc (Optimized Row Columnar)是hive 0.11版里引入的新的存储格式。
- 可以看到每个Orc文件由1个或多个stripe组成，每个stripe250MB大小，这个Stripe实际相当于RowGroup概念，不过大小由4MB->250MB，这样能提升顺序读的吞吐率。每个Stripe里有三部分组成，分别是Index Data,Row Data,Stripe Footer：
- 一个orc文件可以分为若干个Stripe
- 一个stripe可以分为三个部分
- indexData：某些列的索引数据；rowData ：真正的数据存储；StripFooter：stripe的元数据信息。
- 每个文件有一个File Footer，这里面存的是每个Stripe的行数，每个Column的数据类型信息等；每个文件的尾部是一个PostScript，这里面记录了整个文件的压缩类型以及FileFooter的长度信息等。在读取文件时，会seek到文件尾部读PostScript，从里面解析到File Footer长度，再读FileFooter，从里面解析到各个Stripe信息，再读各个Stripe，即从后往前读。

Hive支持数据格式



PARQUET格式：

- Parquet文件是以二进制方式存储的，所以是不可以直接读取的，文件中包括该文件的数据和元数据，因此Parquet格式文件是自解析的。
- 通常情况下，在存储Parquet数据的时候会按照Block大小设置行组的大小，由于一般情况下每一个Mapper任务处理数据的最小单位是一个Block，这样可以把每一个行组由一个Mapper任务处理，增大任务执行并行度。Parquet文件的格式如左图所示。
- 左图展示了一个Parquet文件的内容，一个文件中可以存储多个行组，文件的首位都是该文件的Magic Code，用于校验它是否是一个Parquet文件，Footer length记录了文件元数据的大小，通过该值和文件长度可以计算出元数据的偏移量，文件的元数据中包括每一个行组的元数据信息和该文件存储数据的Schema信息。除了文件中每一个行组的元数据，每一页的开始都会存储该页的元数据，在Parquet中，有三种类型的页：数据页、字典页和索引页。数据页用于存储当前行组中该列的值，字典页存储该列值的编码字典，每一个列块中最多包含一个字典页，索引页用来存储当前行组下该列的索引，目前Parquet中还不支持索引页。

本章任务

Task of this chapter

1 Hive支持数据格式

2 文件格式与压缩案例

3 Java调用Hive案例

任务1 文件格式与压缩案例

任务背景

如何提高Hive数据存储的性能？Hive中提供了各种数据格式的存储文件。各存储文件存储格式，压缩性能各有侧重。掌握Hive中的存储文件的使用对利用存储空间，提高查询性能有很大的裨益。本实验在创建表时，利用不同的存储格式存储，导入数据后比较占用空间和查询时间，来对各种存储格式进行对比，说明不同存储格式的使用方法。

任务1 文件格式与压缩案例

任务需求

1）创建表txt_stocks，存储数据格式为TEXTFILE。导入数据查看占用存储空间。

2）创建表orc_stocks，存储数据格式为ORC。导入数据查看占用存储空间。

3）创建表parquet_stocks，存储数据格式为Parquet。导入数据查看占用存储空间。

4）创建表sequence_stocks，存储数据格式为Sequence。导入数据查看占用存储空间。

5）查询以上各表，对比查询时间。

任务1 文件格式与压缩案例

任务分析

启动Hadoop服务，查看安全模式的状态。启动Hive服务，进入Hive命令行客户端。创建不同格式的表stocks（包含股票代码、股票交易日期、股票开盘价、股票最高价、股票最低价、股票收盘价、股票交易量和股票成交价），并导入数据。对比各种格式表的存储空间占用大小，对比各表执行查询操作所需时间的多少。退出Hive环境，停止Hadoop服务。

任务1 文件格式与压缩案例

任务步骤

1、启动Hadoop服务,启动Hive服务

2、创建不同格式的表stocks（包含股票代码、股票交易日期、股票开盘价、股票最高价、股票最低价、股票收盘价、股票交易量和股票成交价）并导入数据

3、对比各种格式表的存储空间占用大小，对比各表执行查询操作所需时间的多少

4、停止Hadoop服务

5、退出Hive环境

任务1 文件格式与压缩案例

任务结果

Hive命令行客户端正常启动，成功执行相应Hive语句。

本章任务

Task of this chapter

1 Hive支持数据格式

2 文件格式与压缩案例

3 Java调用Hive案例

任务2 Java调用Hive案例

任务背景

访问Hive的方式除了可以用Cli访问方式外，还可以通过ODBC或JDBC访问。JDBC restful方式来访问Hive，可以通过Beeline也可以通过Java编写的客户端程序来访问。下面的实验通过在Java程序中编写JDBC来访问Hive。完成Hive表的创建、描述、导入数据和查询数据。

任务2 Java调用Hive案例

任务需求

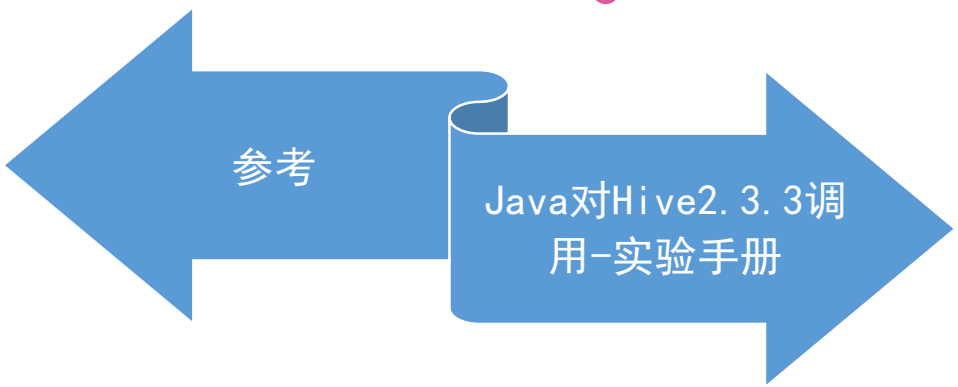
编写Java客户端程序，通过JDBC操作访问Hive 完成以下功能：

1）创建表jdbc_userinfo（id int ,name String）

2）打印表jdbc_userinfo结构信息

3）数据导入到表jdbc_userinfo。

4）打印表jdbc_userinfo数据内容



任务2 Java调用Hive案例

任务分析

创建Java项目，导入Hive操作需要的相关Jar包。利用JDBC访问Hive。执行JDBC操作Hive，完成表创建、打印表结构、数据导入、打印表数据等功能。

任务2 Java调用Hive案例

任务步骤

1、建立新项目HiveTest

2、项目HiveTest中导入Hive操作需要的相关Jar包

3、编写Java程序，操作Hive库表

4、运行编写的Java操作Hive库表的程序

5、退出实验环境，停止平台服务

任务2 Java调用Hive案例

任务结果

可以看到控制台中输出了“表创建成功”，表的结构以及数据导入成功的提示内容。最后把导入表中的数据打印出来。

```
jdbc_userinfo表创建成功。
执行"DESCRIBE table"运行结果：
id  int
name    string
age int
jdbc_userinfo表数据导入成功。
执行表查询运行结果：
1   lixiaosan   20
2   wangyanli   23
3   zhangxiao   32
```

谢谢观看

THANKS FOR WATCHING