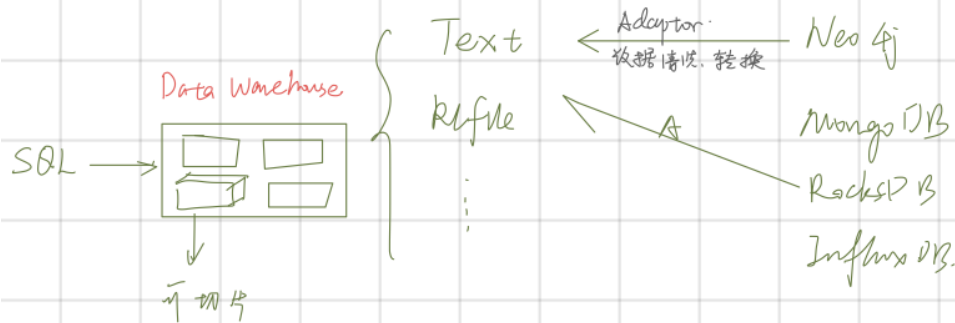


lect30 hive

Hive 背景: Lake & repository



数据库: 先定义好 schema 再写: schema on write.

数据仓库: 先定义好表格式, 数据清洗转换, 校验

↓
problem: 原始数据没3.

↓
数据湖: 把原始数据放入, 要用列, 才写入

数据仓库, everybody, 把 data 放到 Lake.

需求: 用一个 SQL 可对多个 data 调用

① 整理 data → 数据仓库

② 翻译 SQL: 把 SQL 翻译为不同 db 认识的 SQL, → 性能差

另有数据湖

③ 判断 data 的访问频率, 高于一定值, 就从 Lake → repository.

∴ 热 data → repository cold data → lake

湖仓一体: lakehouse.

ETL: 清洗、转换、加载

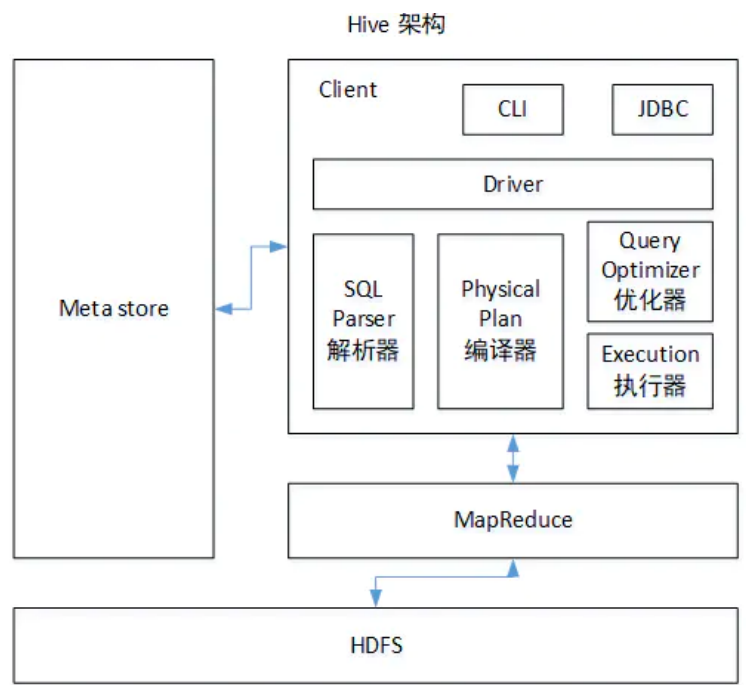
数据仓库是一个面向主题的（Subject Oriented）、集成的（Integrate）、相对稳定的（Non-Volatile）、反映历史变化（Time Variant）的数据集合，用于支持管理决策。

对于数据仓库的概念我们可以从两个层次予以理解，首先，数据仓库用于支持决策，面向**分析型数据处理**，它不同于企业现有的操作型数据库；其次，数据仓库是对**多个异构的数据源有效集成**，集成后按照主题进行了重组，并包含历史数据，而且存放在数据仓库中的数据一般不再修改。

目前数据湖和数据仓库的三种关系：

- 1、数据湖存所有的数据原始文件，作为数据仓库的数据源，在数据仓库做高级操作时直接从数据湖找数据
- 2、数据仓库里的数据是文件格式区别的，数据湖可以翻译不同格式的数据，就是直接把数据湖作为数据仓库，都可以用SQL方式数据湖访问数据湖
- 3、介于前两种之间，当数据的使用高于一定频率时移动到数据仓库里，即热数据存在数据仓库里，可以提高处理这些数据时的效率，这种关系也称湖仓一体（lakehouse），是hive的前身

hive 是基于 Hadoop 的一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供完整的 sql 查询功能，可以将 sql语句转换为MapReduce 任务进行运行。其优点是学习成本低，可以通过类 SQL 语句快速实现简单的 MapReduce 统计。



运行

所有的操作都是以mapreduce的方式做的。分布式系统。

需要检查namenode和datanode是否都跑起来了。

配置文件：<install-dir>/conf/hive-default.xml

所有的操作都是基于mapreduce, 但会自己优化，有时候就变成本地了。

分区

partition 列：虚拟的列，专门用来做分区。按hash分区？按range分区？是否合理

不同的列之间的隔开是ctol + A

建表：create table tablename(col1 string) partitioned by(col2 string);

添加分区：alter table tablename add partition(col2='202101');

删除分区：alter table tablename drop partition(col2='202101');

metadata

存储表的结构

Metadata Store

- Metadata is in an **embedded Derby database** whose disk storage location is determined by the Hive configuration variable named **javax.jdo.option.ConnectionURL**.
 - By default this location is **./metastore_db** (see conf/hive-default.xml).
- Right now, in the default configuration, this metadata can only be seen by one user at a time.
- Metastore can be stored in any database that is supported by **JPOX**.
- The location and the type of the RDBMS can be controlled by the two variables **javax.jdo.option.ConnectionURL** and **javax.jdo.option.ConnectionDriverName**.
- In the future, the metastore itself can be a standalone server.
- If you want to run the metastore as a network server so it can be accessed from multiple nodes, see [Hive Using Derby in Server Mode](#).

Data Load

```
hive> LOAD DATA LOCAL INPATH './examples/files/kv2.txt' OVERWRITE INTO TABLE  
invites PARTITION (ds='2008-08-15'); 把kv2.txt 导入invites表的2008-08-15的分区
```

可以用类似SQL的语言进行操作。

数据仓库中是一次写入多次读的，不会有改写，因为不是原始数据改它没有意义。

example

1. 加载数据

```
LOAD DATA LOCAL INPATH '<path>/u.data' OVERWRITE INTO TABLE u_data;
```

2. 操作 **map reduce**

a. `SELECT COUNT(*) FROM u_data;`

map reduce可以的原因是它构造在HDFS之上，而hdfs是构造在hadoop 上面。

b. 可以自己写map reduce 的python 逻辑

Use the mapper script:

```
CREATE TABLE u_data_new (  
  userid INT,  
  movieid INT,  
  rating INT,  
  weekday INT)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t';  
  
add FILE weekday_mapper.py; hive文件系统的路径,  
这里放到了根目录  
  
INSERT OVERWRITE TABLE u_data_new  
SELECT  
  TRANSFORM (userid, movieid, rating, unixtime)  
  USING 'python weekday_mapper.py'  
  AS (userid, movieid, rating, weekday)  
FROM u_data;  
  
SELECT weekday, COUNT(*)  
FROM u_data_new  
GROUP BY weekday;
```

不同的数据库都可以支持中间格式，然后用这个中间格式导入到hive就可以了。eg

- Hive supports several file formats:
 - Text File
 - SequenceFile
 - [RCFile](#)
 - [Avro Files](#)
 - [ORC Files](#)
 - [Parquet](#)
 - Custom INPUTFORMAT and OUTPUTFORMAT
- The [hive.default.fileformat](#) configuration parameter
 - determines the format to use if it is not specified in a [CREATE TABLE](#) or [ALTER TABLE](#) statement.
 - Text file is the parameter's **default** value.

RCFile

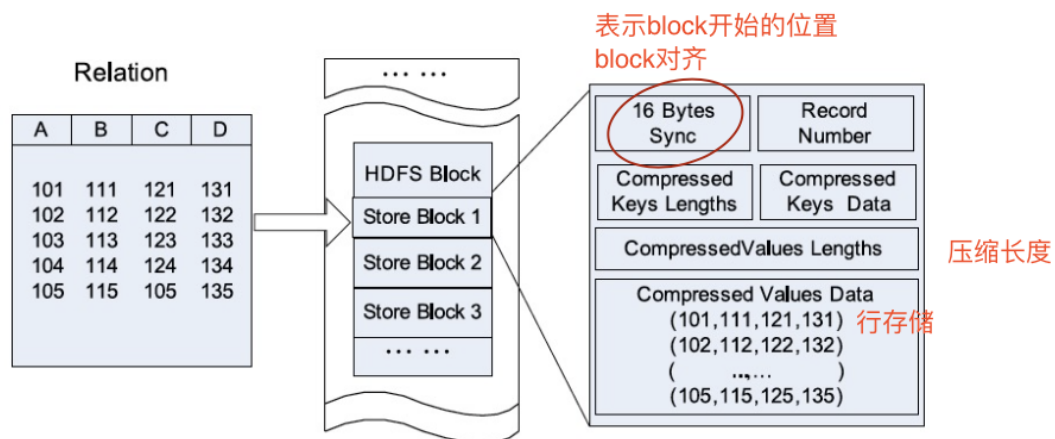


Fig. 1: An example of row-store in an HDFS block.

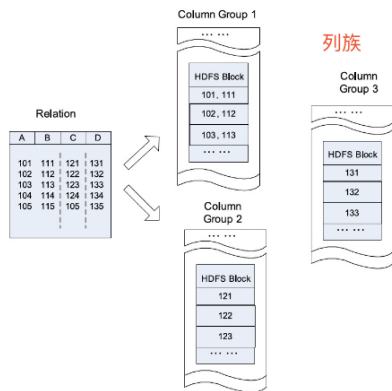


Fig. 2: An example of column-group in an HDFS block. The four columns are stored into three column groups, since column A and B are grouped in the first column group.

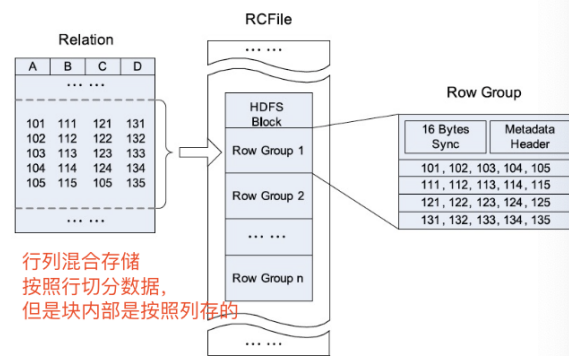


Fig. 3: An example to demonstrate the data layout of RCFile in an HDFS block.

2011 ICDE conference paper "[RCFile: A Fast and Space-efficient Data Placement Structure in MapReduce-based Warehouse Systems](#)"
by Yongqiang He, Rubao Lee, Yin Huai, Zheng Shao, Namit Jain, Xiaodong Zhang, and Zhiwei Xu

Parquet：大家公认的格式，最常用的格式。

file compression：文件很大，需要压缩，这个是hive的优势。load的时候直接load压缩文件也可以。就是一个lazy 的操作，不用的时候不解压，但是这样不解压怎么分开存储呢？就类似关键帧，在某一些地方可以分开。这样就在空间的占用率和时间上权衡

补充问答：

1、Hive与传统数据库的区别？

Hive和数据库除了拥有类型的查询语言外，无其他相似

存储位置：Hive数据存储在HDFS上。数据库保存在块设备或本地文件系统

数据更新：Hive不建议对数据改写。数据库通常需要经常修改

执行引擎：Hive通过MapReduce来实现。数据库用自己的执行引擎

执行速度：Hive执行延迟高，但它数据规模远超过数据库处理能力时，Hive的并行计算能力就体现优势了。数据库执行延迟较低

数据规模：hive大规模的数据计算。数据库能支持的数据规模较小

扩展性：Hive建立在Hadoop上，随Hadoop的扩展性。数据库由于ACID语义[wh1] 的严格限制，扩展有限

2、Hive内部表和外部表的区别

存储：外部表数据由HDFS管理；内部表数据由hive自身管理

存储：外部表数据存储位置由自己指定（没有指定location则在默认地址下新建）；内部表数据存储在hive.metastore.warehouse.dir（默认在/uer/hive/warehouse）

创建：被external修饰的就是外部表；没被修饰是内部表

删除：删除外部表仅仅删除元数据；删除内部表会删除元数据和存储数据

3、Hive优点

操作接口采用类SQL语法，提供快速开发的能力(简单、容易上手)；

避免了去写MapReduce，减少开发人员的学习成本；

统一的元数据管理，可与impala/spark等共享元数据；

易扩展(HDFS+MapReduce：可以扩展集群规模；支持自定义函数)；

4、Hive使用场景

数据的离线处理；比如：日志分析，海量结构化数据离线分析...

Hive的执行延迟比较高，因此hive常用于数据分析的，对实时性要求不高的场合；

Hive优势在于处理大数据，对于处理小数据没有优势，因为Hive的执行延迟比较高。