任务 1-2 认识Hive数据仓库

1.2.1 Apache Hive概述

在当前的任务中，我们将深入探讨Hive，这是一个基于Hadoop的数据仓库工具。我们将详细介绍Hive的架构、基本数据类型及其如何融入整个大数据生态系统。通过本次任务，学生们将更好地理解Hive的功能、优点以及它在大规模数据处理和分析中的角色。



什么是Hive

- Apache Hive是一款建立在Hadoop之上的开源**数据仓库系统**，可以将存储在Hadoop文件中的结构化、半结构化数据文件映射为一张数据库表，基于表提供了一种类似SQL的查询模型，称为Hive查询语言（HQL），用于访问和分析存储在Hadoop文件中的大型数据集。
- Hive核心是将HQL转换为MapReduce程序，然后将程序提交到Hadoop群集执行。
- Hive由Facebook实现并开源。



大象的头

蜜蜂 (mìfēng)

为什么使用Hive

- 使用Hadoop MapReduce直接处理数据所面临的问题

  人员学习成本太高 需要掌握java语言

  MapReduce实现复杂查询逻辑开发难度太大

- 使用Hive处理数据的好处

  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）

  避免直接写MapReduce，减少开发人员的学习成本

  支持自定义函数，功能扩展很方便

  背靠Hadoop，擅长存储分析海量数据集

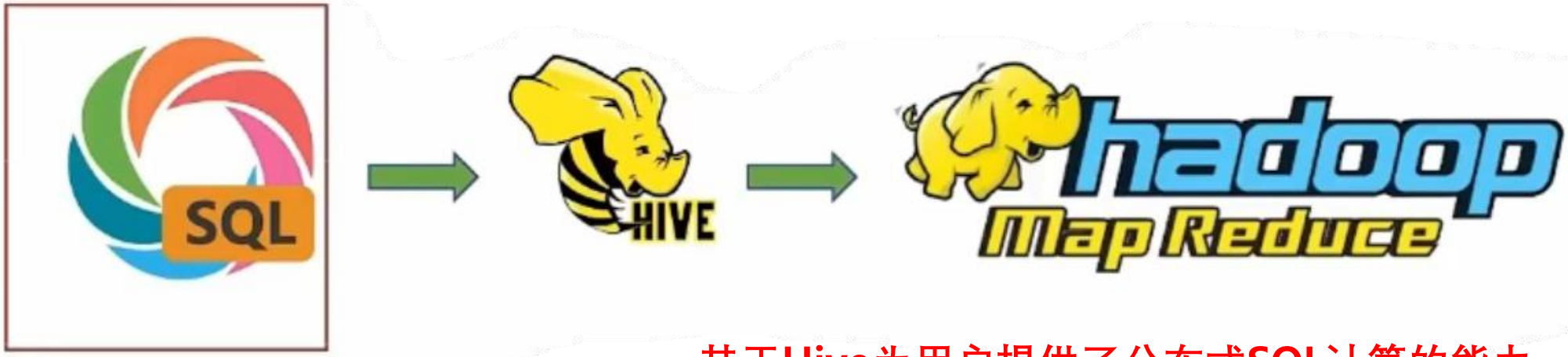
Hive和Hadoop关系

- 从功能来说，数据仓库软件，至少需要具备下述两种能力：
  - 存储数据的能力
  - 分析数据的能力
- Apache Hive作为一款大数据时代的数据仓库软件，当然也具备上述两种能力。只不过Hive并不是自己实现了上述两种能力，而是借助Hadoop。

  Hive利用HDFS存储数据，利用MapReduce查询分析数据。
- 这样突然发现Hive没啥用，不过是套壳Hadoop罢了。其实不然，Hive的最大魅力在于用户专注于编写HQL，Hive帮您转换成为MapReduce程序完成对数据的分析。

Hive和Hadoop关系

- Apache Hive是一款分布SQL计算的工具，其主要功能是：
  - 将SQL语句翻译MapReduce程序运行



其于Hive为用户提供了分布式SQL计算的能力
写的是SQL，执行的是MapReduce

案例：如何模拟实现Apache Hive的功能

如果让您设计Hive这款软件，要求能够实现用户编写SQL语句，Hive自动将SQL转换为MapReduce程序，处理位于HDFS上的结构化数据。如何实现？

在HDFS文件系统上有一个文件，路径为/data/china_user.txt，其内容如下：

```
1,zhangsan,18,beijing
2,lisi,25,shanghai
3,allen,30,shanghai
4,wangwu,15,nanjing
5,sangok,55,gwangju
6,yangweon,68,seoul
```

原本数据：
结构化数据

需求：统计来自于上海且年龄大于25岁的用户有多少个？

场景案例：如何模拟实现Apache Hive的功能

场景目的

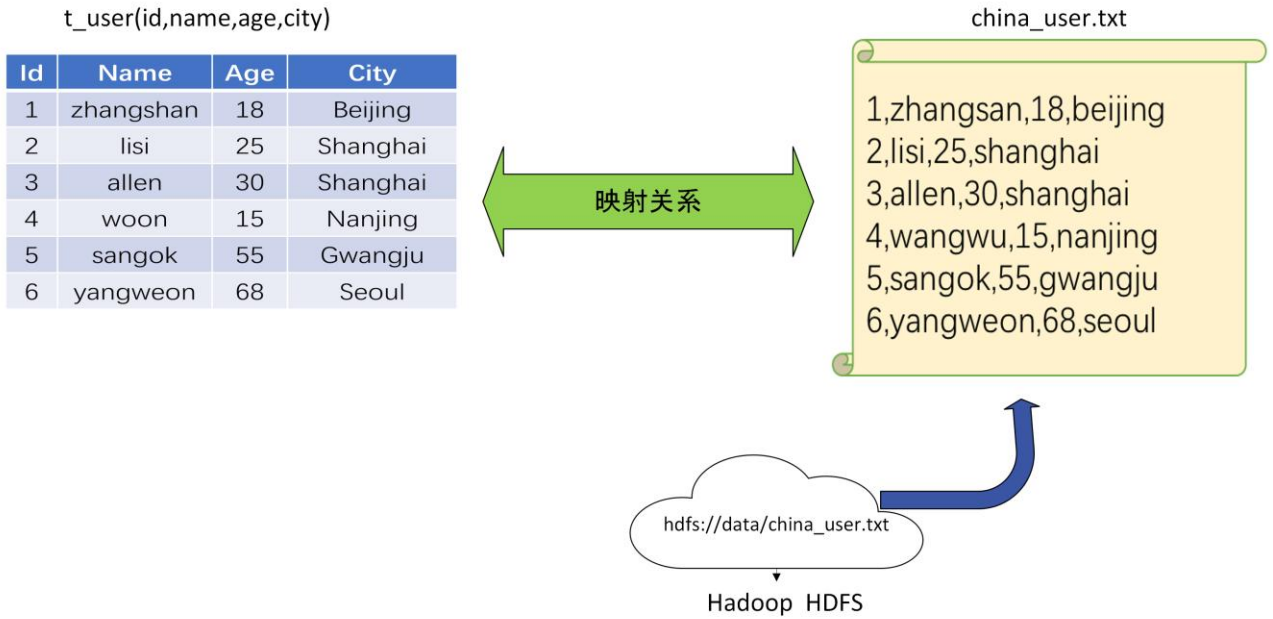
- 重点理解下面两点：
  - Hive能将数据文件映射成为一张表，这个**映射**是指什么？
  - Hive软件本身到底承担了什么**功能职责**？

场景案例：如何模拟实现Apache Hive的功能

映射信息记录

- 映射在数学上称之为一对对应关系，比如 y = x + 1，对于每一个 x 的值都有与之对应的 y 的值。

- 在hive 中能够写 SQL 处理的前提是针对于表，而不是针对于文件，因此需要将 文件和表之间的对应关系 描述记录清楚。
  映射信息专业的叫法称之为 元数据信息元数据是指用来描述数据的数据 metadata）。



场景案例：如何模拟实现Apache Hive的功能

映射信息记录

- 具体来看，要记录的元数据信息包括:

➢ 表对应着哪个文件（位置信息）
➢ 表的列对应着文件哪个字段（顺序信息）
➢ 文件字段之间的分隔符是什么。

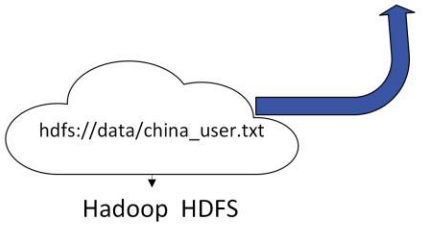
t_user(id,name,age,city)

Id	Name	Age	City
1	zhangshan	18	Beijing
2	lisi	25	Shanghai
3	allen	30	Shanghai
4	woon	15	Nanjing
5	sangok	55	Gwangju
6	yangweon	68	Seoul



china_user.txt

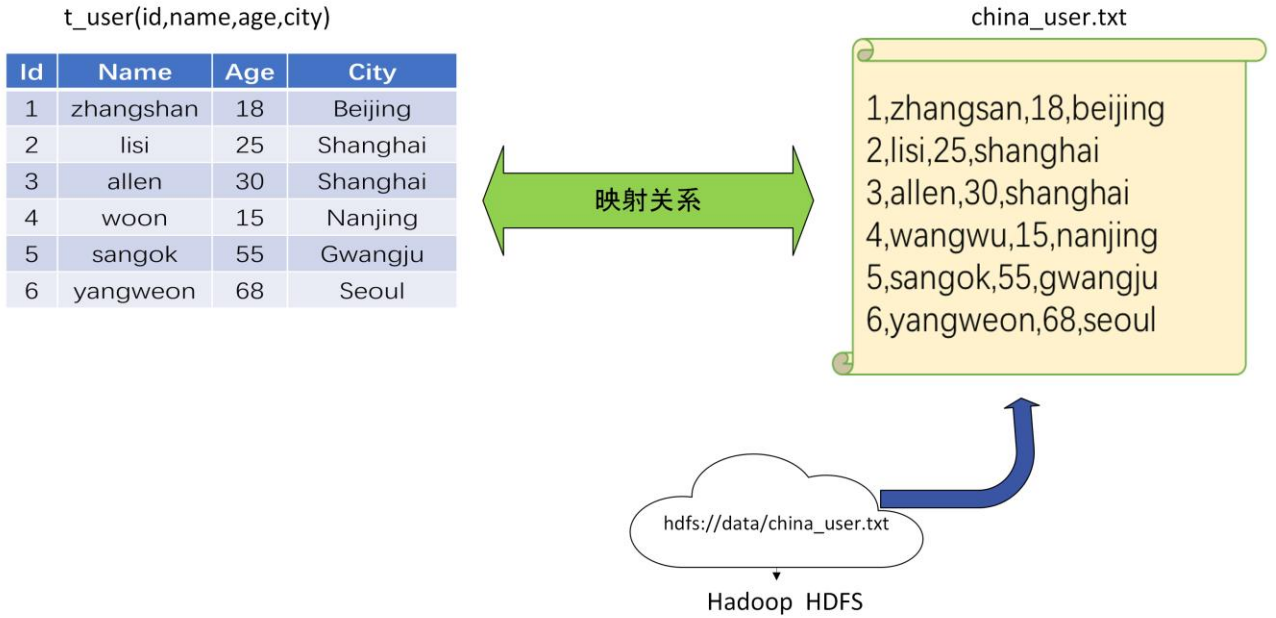
1,zhangsan,18,beijing
2,lisi,25,shanghai
3,allen,30,shanghai
4,wangwu,15,nanjing
5,sangok,55,gwangju
6,yangweon,68,seoul



场景案例：如何模拟实现Apache Hive的功能

SQL语法解析、编译

- 用户写完SQL之后，Hive需要针对SQL进行语法校验，并且根据记录的元数据信息解读SQL背后的含义，制定执行计划。并且把执行计划转换成MapReduce程序来执行，把执行的结果封装返回给用户。



场景案例：如何模拟实现Apache Hive的功能

- Hive能将数据文件映射成为一张表，这个映射是指什么？

文件和表之间的对应关系

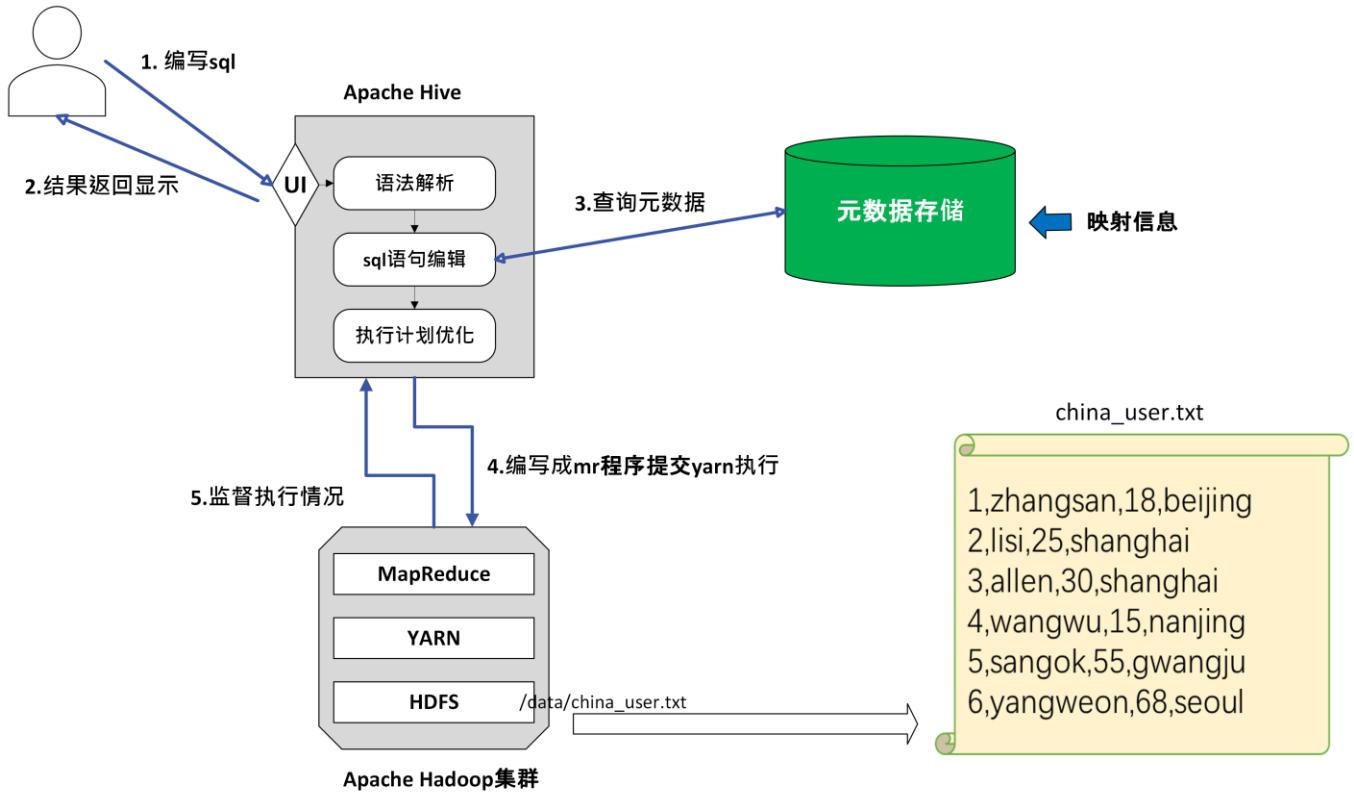
- Hive软件本身到底承担了什么功能职责？

SQL语法解析编译成为MapReduce

场景案例：如何模拟实现Apache Hive的功能

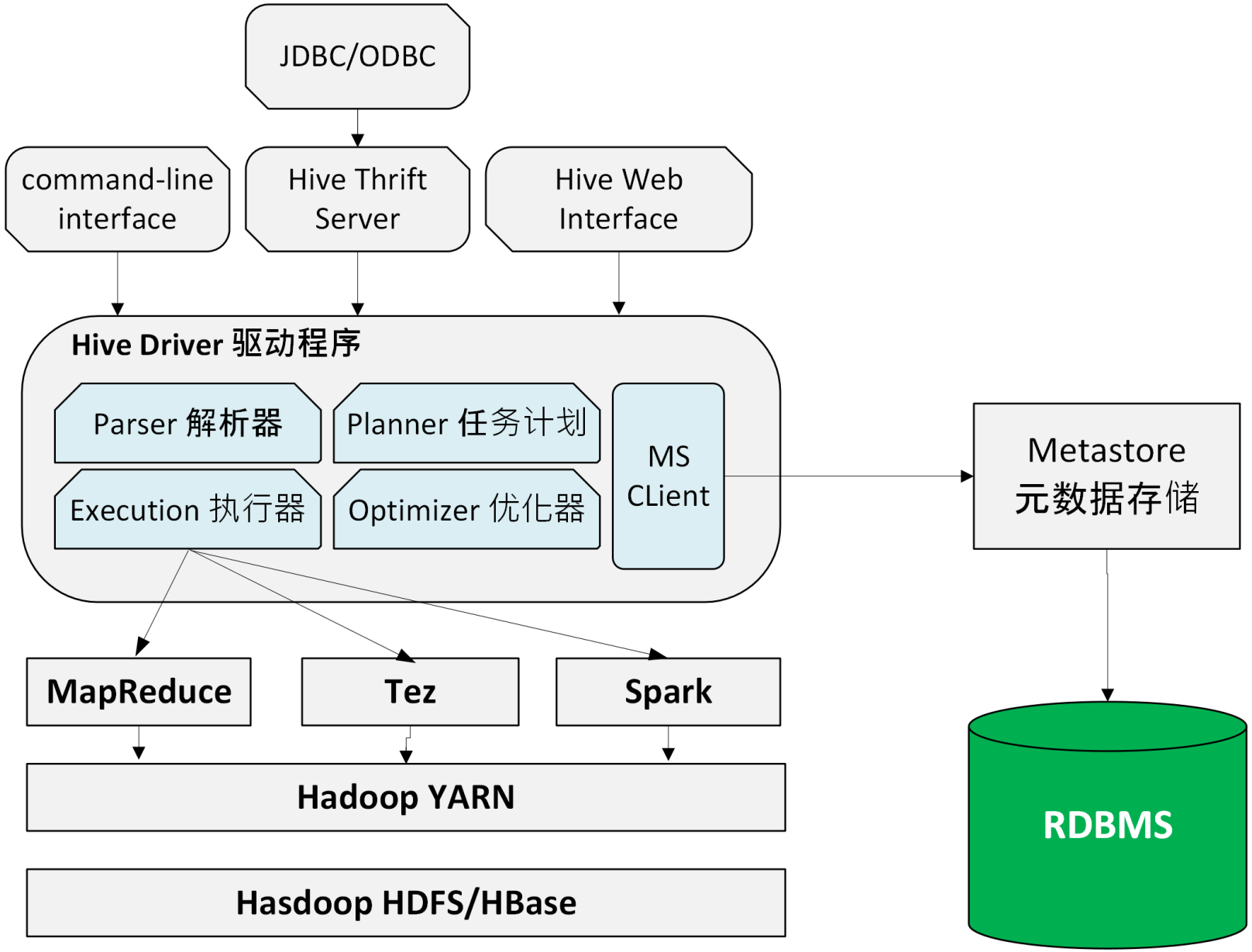
最终效果

- 基于上述分析，最终要想模拟实现的Hive的功能，大致需要下图所示组件参与其中。
- 从中可以感受一下Hive承担了什么职责，当然，也可以把这个理解为Hive的架构图。



1.2.2 Apache Hive架构，组件

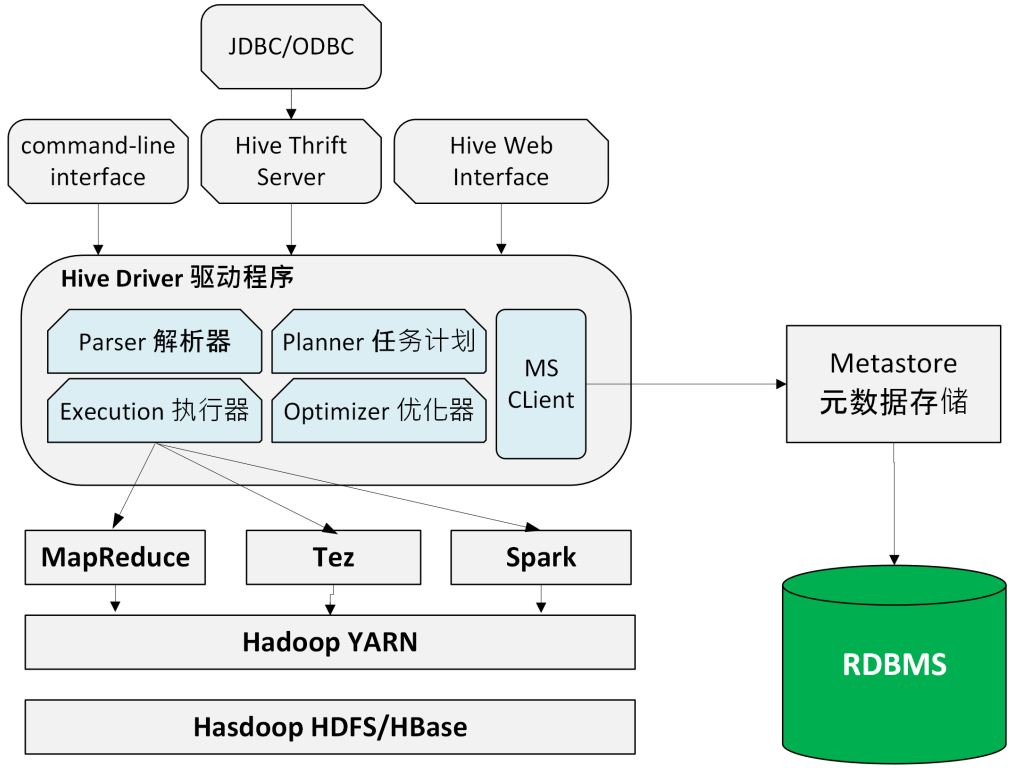
架构图



组件

- 用户接口

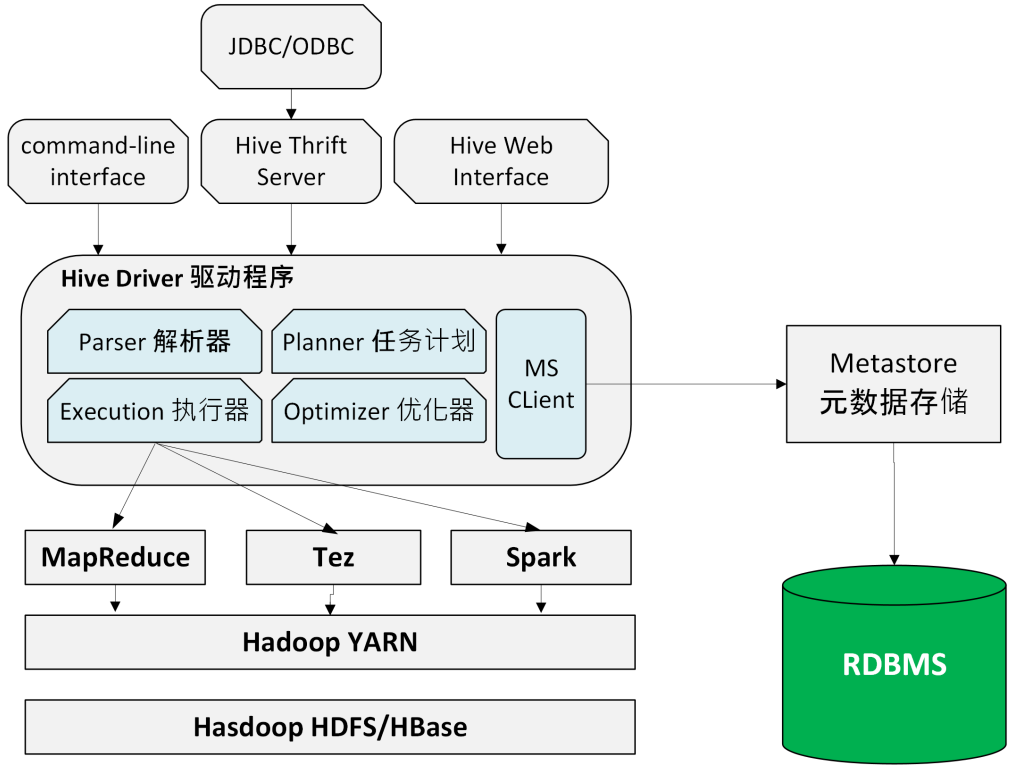
  包括 CLI、JDBC/ODBC、WebGUI。其中，CLI (command line interface) 为shell命令行；Hive中的Thrift服务器允许外部客户端通过网络与Hive进行交互，类似于JDBC或ODBC协议。WebGUI是通过浏览器访问Hive。



组件

●元数据存储

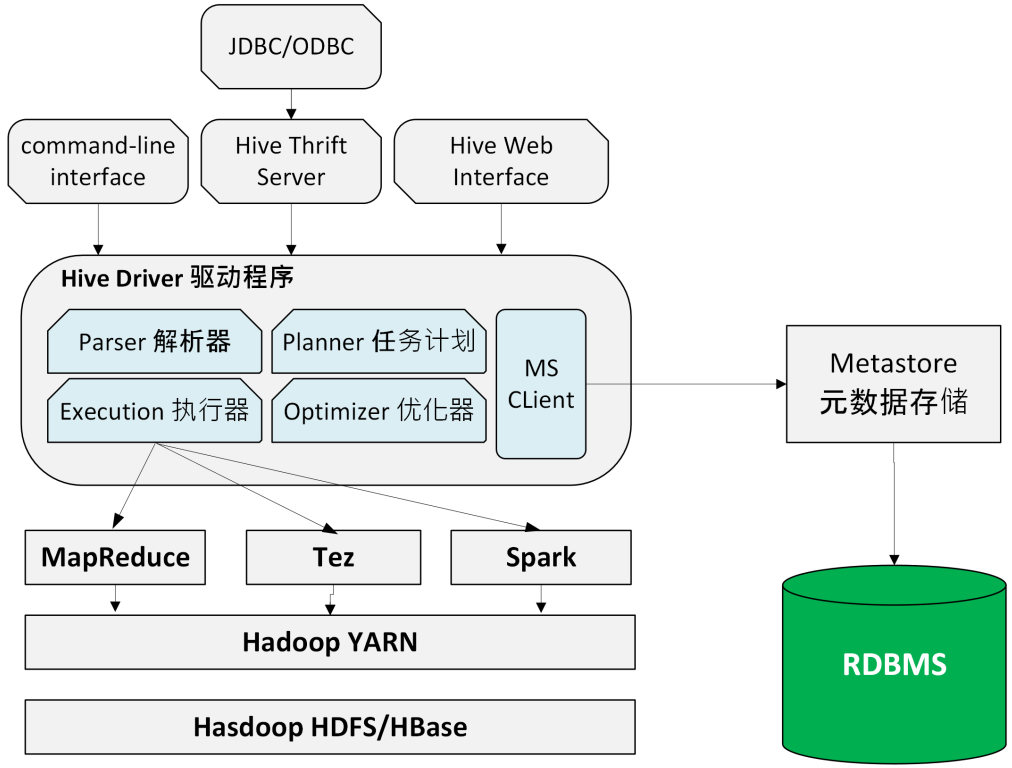
通常是存储在关系数据库如 mysql/derby 中。
Hive 中的元数据包括表的名字、表的列和分区及其属性，
表的属性（是否为外部表等），表的数据所在目录等。



组件

- **Driver驱动程序，包括语法解析器、计划编译器、优化器、执行器**

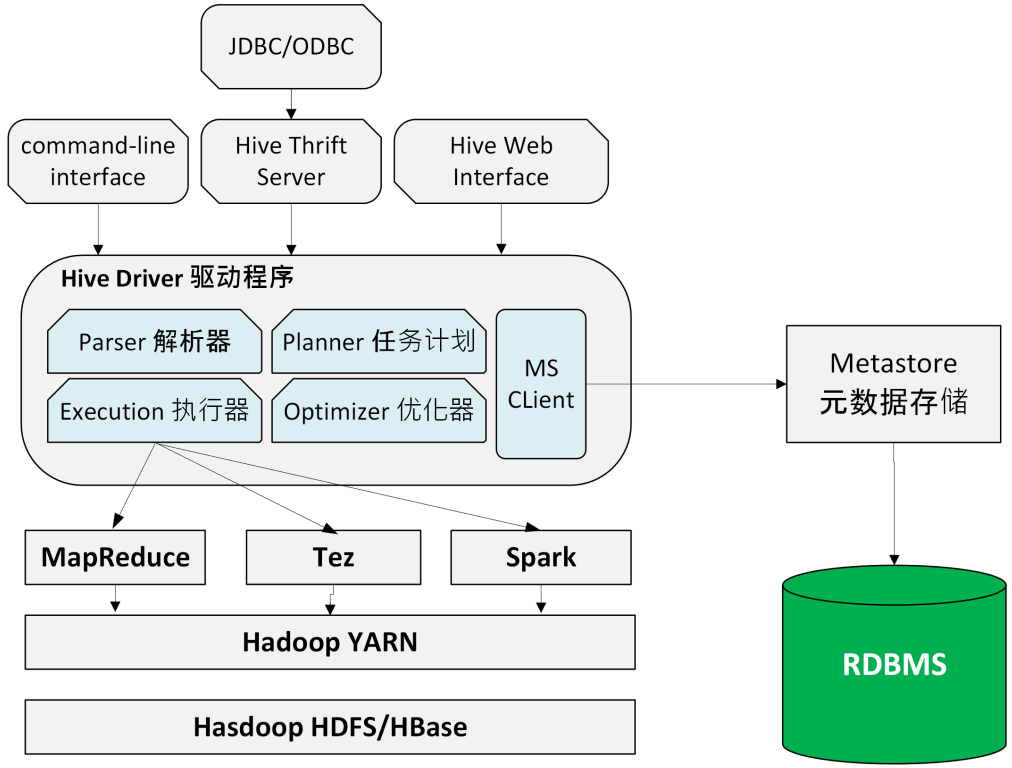
完成 HQL 查询语句从词法分析、语法分析、编译、优化以及查询计划的生成。生成的查询计划存储在 HDFS 中，并在随后由执行引擎调用执行。



组件

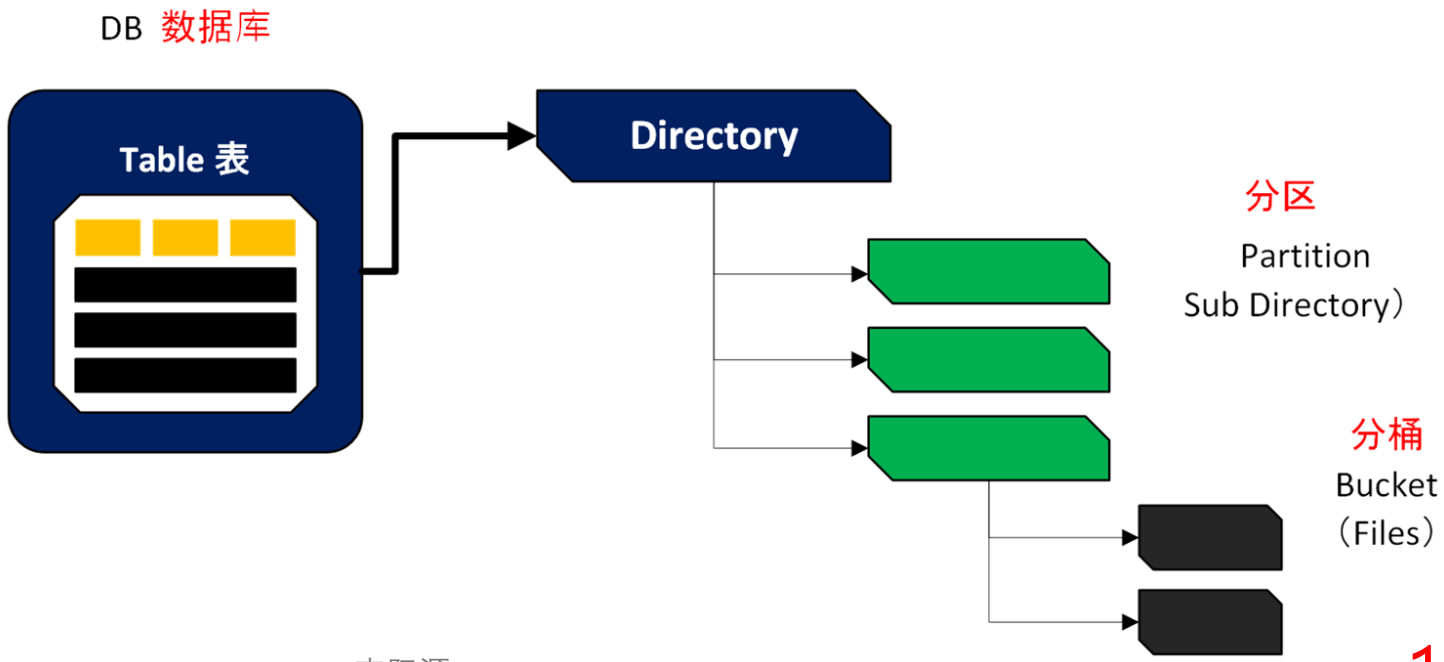
● 执行引擎

Hive本身并不直接处理数据文件。而是通过执行引擎处理。当下Hive支持MapReduce、Tez、Spark3种执行引擎。



Data Model概念

- 数据模型：用来描述数据、组织数据和对数据进行操作，是对现实世界数据特征的描述。
- Hive的数据模型类似于RDBMS库表结构，此外还有自己特有模型。
- Hive中的数据可以在粒度级别上分为三类：
  - Table 表
  - Partition 分区
  - Bucket 分桶



Databases：数据库

- Hive作为一个数据仓库，在结构上积极向传统数据库看齐，也分数据库（Schema），每个数据库下面有各自的表组成。默认数据库default。
- Hive的数据都是存储在HDFS上的，默认有一个根目录，在hive-site.xml中，由参数hive.metastore.warehouse.dir指定。默认值为/user/hive/warehouse。
- 因此，Hive中的数据库在HDFS上的存储路径为：

  $ {hive.metastore.warehouse.dir}/databasename.db
- 比如，名为itcast的数据库存储路径为：

  /user/hive/warehouse/itcast.db

Databases 数据库

- Hive作为一个数据仓库，在结构上积极向传统数据库看齐，也分数据库（Schema），每个数据库下面有各自的表组成。默认数据库default。
- Hive的数据都是存储在HDFS上的，默认有一个根目录，在hive-site.xml中，由参数hive.metastore.warehouse.dir指定。默认值为/user/hive/warehouse。
- 因此，Hive中的数据库在HDFS上的存储路径为：

  $ {hive.metastore.warehouse.dir}/databasename.db

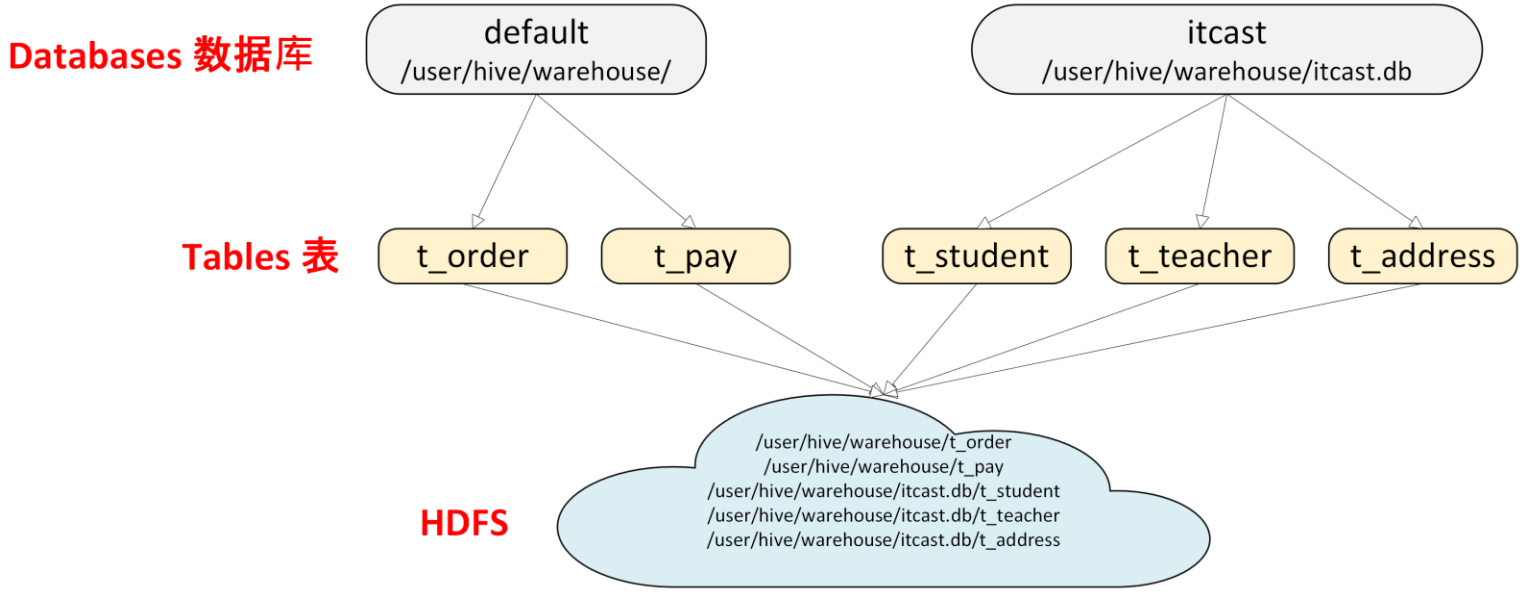
- 比如，名为itcast的数据库存储路径为：

  /user/hive/warehouse/itcast.db

```
hadoop@hadoop01:~$ hdfs dfs -ls /user/hive
Found 1 items
drwxrwxrwx   - hadoop hadoop          0 2026-01-31 20:26 /user/hive/warehouse
hadoop@hadoop01:~$ hdfs dfs -ls /user/hive/warehouse
Found 10 items
drwxr-xr-x   - hadoop hadoop          0 2026-01-29 15:25 /user/hive/warehouse/dual
drwxr-xr-x   - hadoop hadoop          0 2026-01-30 11:30 /user/hive/warehouse/employees
drwxr-xr-x   - hadoop hadoop          0 2026-01-31 20:36 /user/hive/warehouse/hive_db
drwxrwxrwx   - hadoop hadoop          0 2026-01-28 23:14 /user/hive/warehouse/retail_db.db
drwxr-xr-x   - hadoop hadoop          0 2026-01-30 14:42 /user/hive/warehouse/test1
drwxr-xr-x   - hadoop hadoop          0 2026-01-30 15:40 /user/hive/warehouse/test2
drwxr-xr-x   - hadoop hadoop          0 2026-01-30 16:05 /user/hive/warehouse/test3
drwxr-xr-x   - hadoop hadoop          0 2026-01-30 16:05 /user/hive/warehouse/test_data
drwxrwxrwx   - hadoop hadoop          0 2026-01-27 15:32 /user/hive/warehouse/test_db_yangweon202601.db
drwxrwxrwx   - hadoop hadoop          0 2026-01-27 17:12 /user/hive/warehouse/test_pk_ywlee.db
```

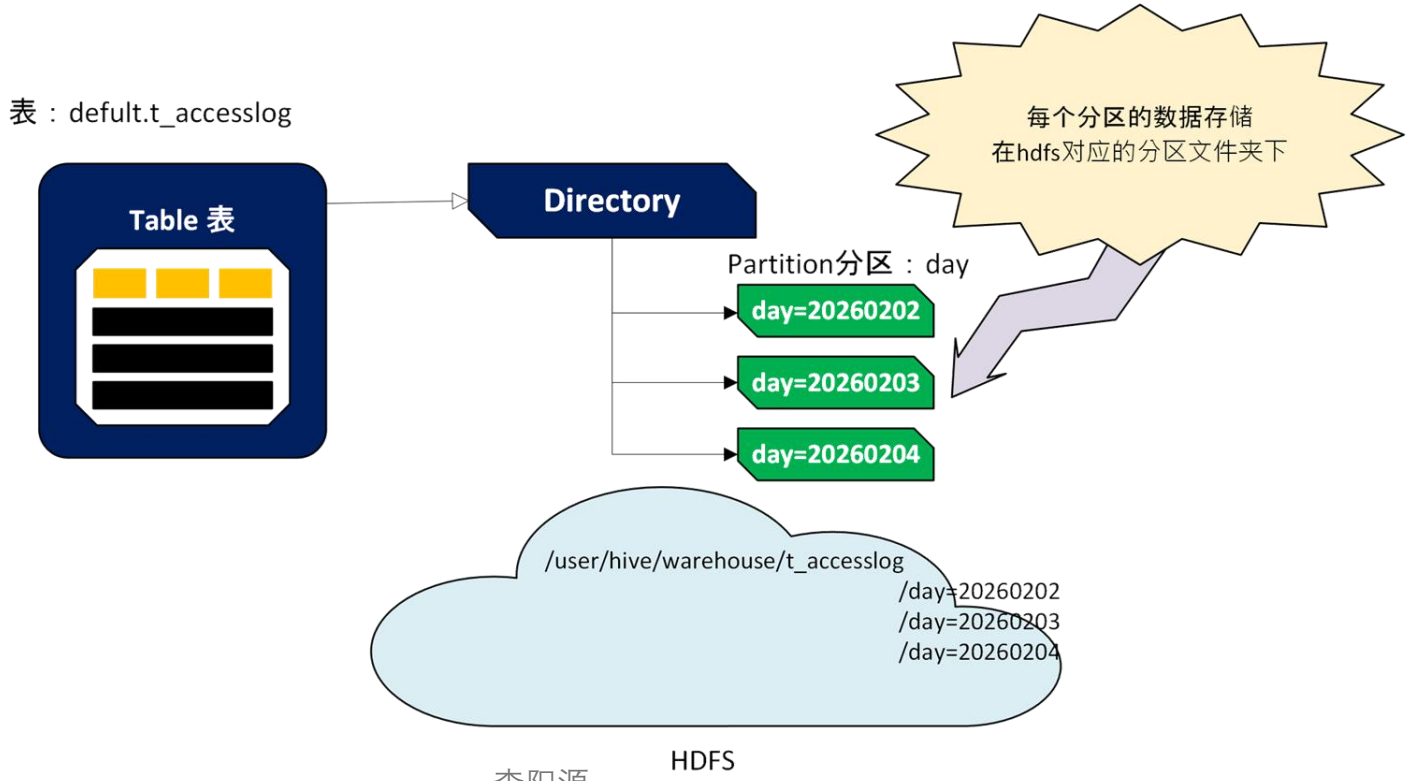
Tables：表

- Hive表与关系数据库中的表相同。Hive中的表所对应的数据通常是存储在HDFS中，而表相关的元数据是存储在RDBMS中。
- Hive中的表的数据在HDFS上的存储路径为：
  ${hive.metastore.warehouse.dir}/databasename.db/tablename
- 比如，itcast的数据库下t_user表存储路径为：
  /user/hive/warehouse/itcast.db/t_user



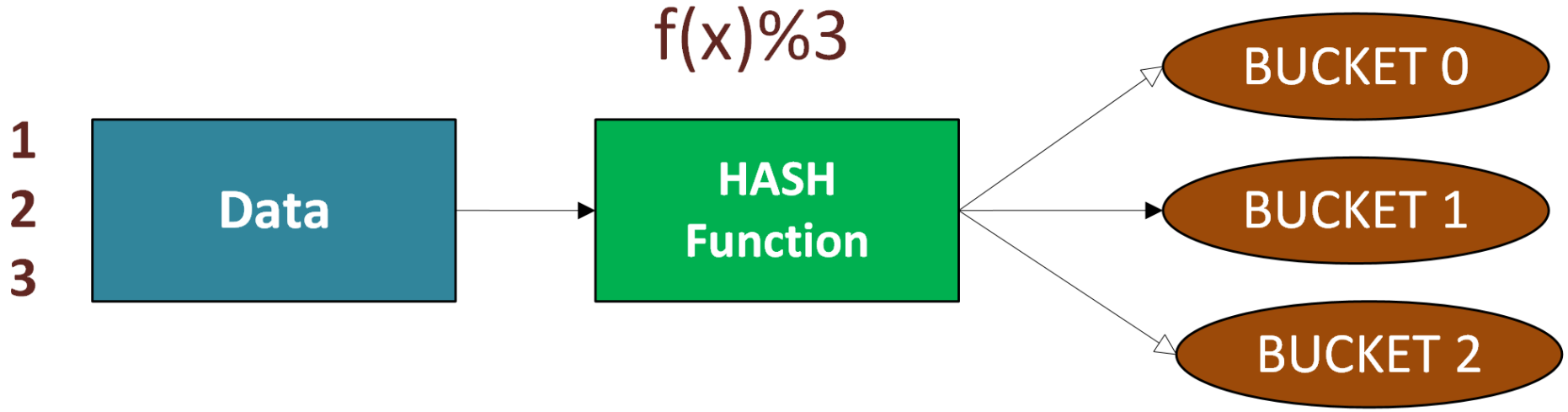
Partitions：分区

- Partition分区是hive的一种优化手段表。分区是指**根据分区列（例如“日期day”）的值将表划分为不同分区**。这样可以更快地对指定分区数据进行查询。
- 分区在存储层面上的表现是：table表目录下以子文件夹形式存在。
- **一个文件夹表示一个分区**。子文件命名标准：分区列=分区值
- Hive还支持分区下继续创建分区，所谓的多重分区。关于分区表的使用和详细介绍，后面模块会单独展开。



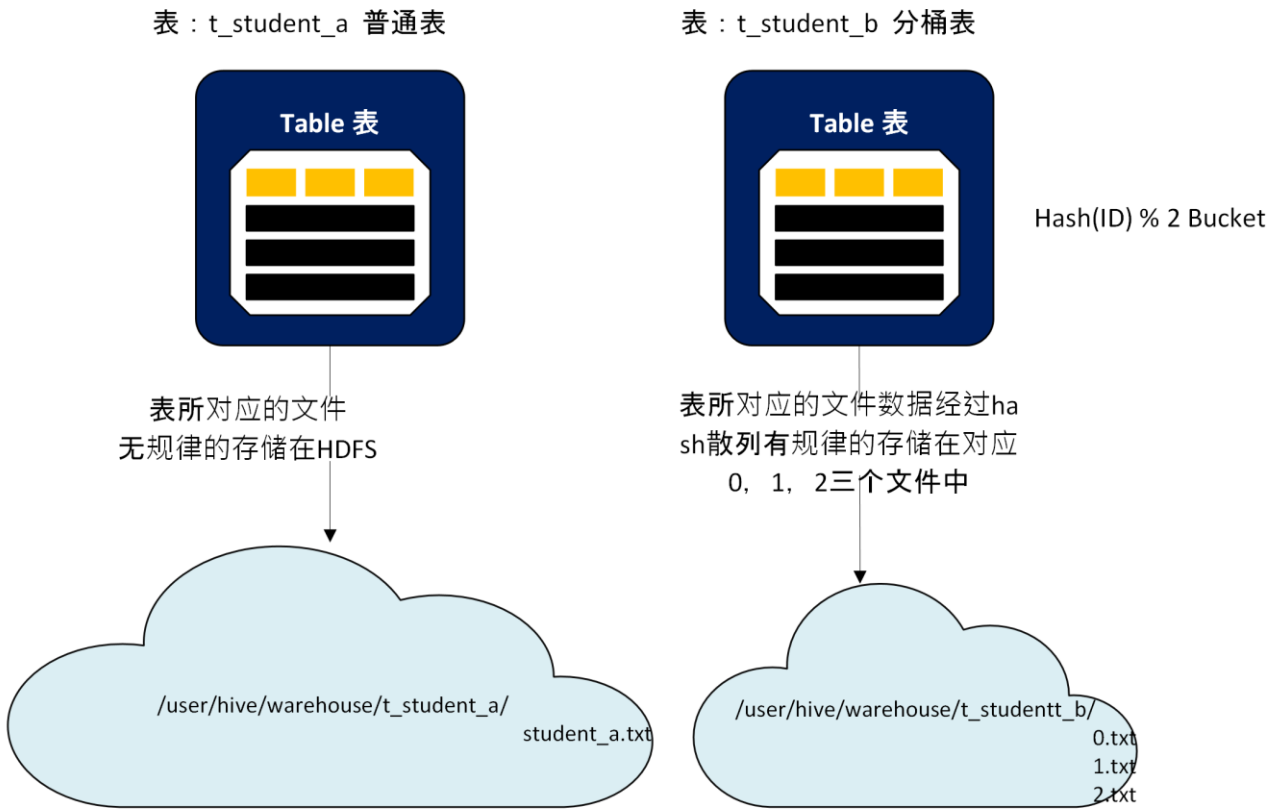
Buckets：分桶

- **Bucket分桶表是hive的一种优化手段表。分桶是指根据表中字段（例如“编号ID”）的值，经过hash计算规则将数据文件划分成指定的若干个小文件。**
- 分桶规则：hashfunc(字段) % 桶个数，余数相同的分到同一个文件。

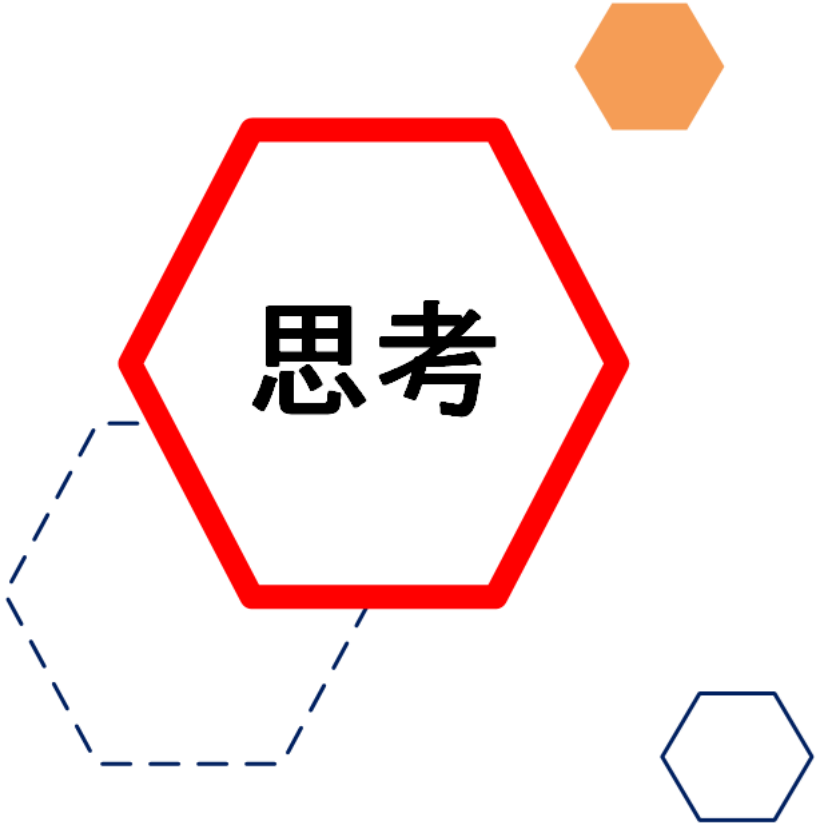


Buckets：分桶

- **分桶的好处是可以优化join查询和方便抽样查询。**
- Bucket分桶表在HDFS中表现为同一个表目录下数据根据hash(散列)之后变成多个文件。关于桶表以及分桶操作，后面模块会单独展开详细讲解。



Apache Hive是要取代MySQL吗?



1. 从数据模型上来看Hive和MySQL非常相似，库、表、字段之类的，难道Hive也是数据库?
2. Hive依赖Hadoop，理论上可以无限存储，难道是大型数据库？取代MySQL的?

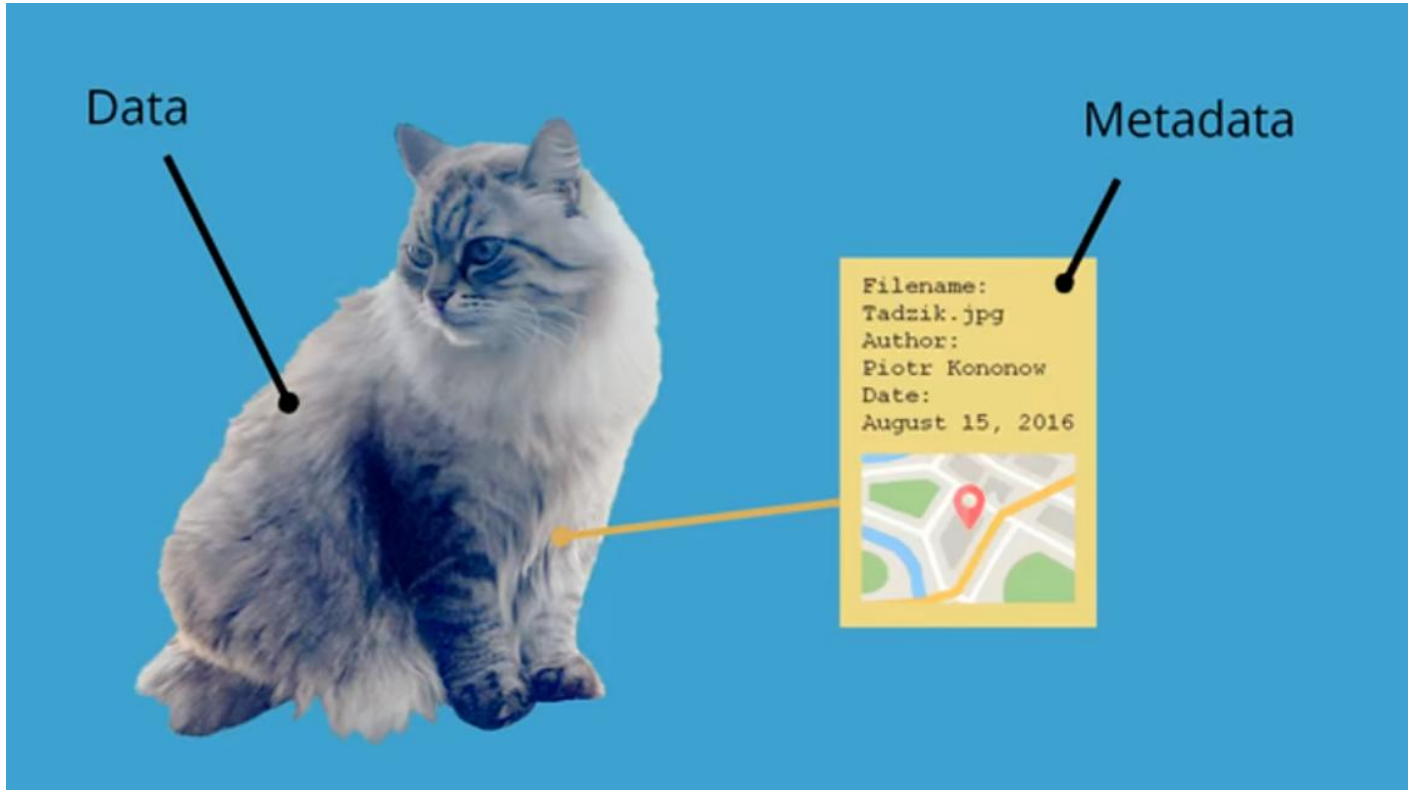
Apache Hive是要取代MySQL吗?

- **Hive虽然具有RDBMS数据库的外表，包括数据模型、SQL语法都十分相似，但应用场景却完全不同。**
- **Hive只适合用来做海量数据的离线分析。**Hive的定位是数据仓库，面向分析的OLAP系统。
- 因此时刻告诉自己，**Hive不是大型数据库，也不是要取代MySQL承担业务数据处理。**

对比项	Apache Hive	MySQL
定位	数据仓库	数据库
使用场景	离线数据分析	业务数据事务处理
查询语言	HQL（Hive 查询语言）	SQL
数据存储	HDFS	本地文件系统（Local FS）
执行引擎	MR、Tez、Spark	Executor（MySQL内部执行引擎）
执行延迟	高	低
处理数据规模	大	小
常见操作	导入数据、查询分析	增删改查（CRUD）

什么是元数据

- 元数据（Metadata），又称中介数据、中继数据，为 描述数据的数据（data about data），主要是描述数据属性（property）的信息，用来支持如指示存储位置、历史数据、资源查找、文件记录等功能。



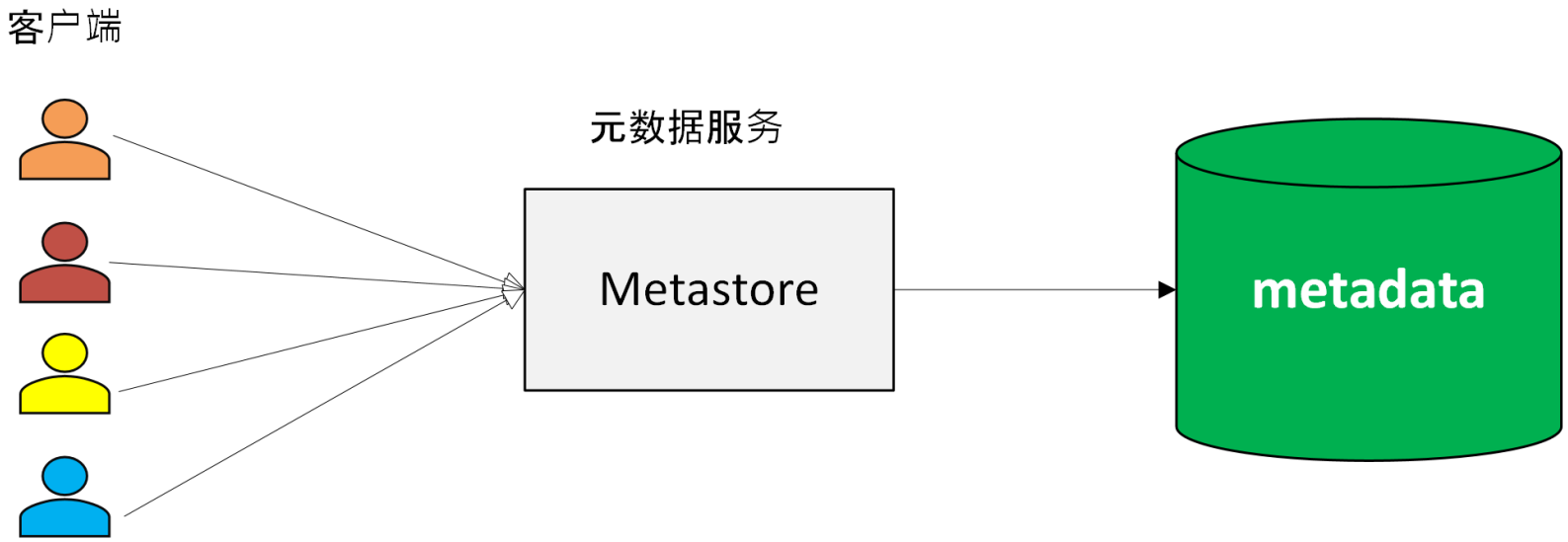

화산_00_등산시작.jpg 속성
일반
디지털 서명
보안
자세히
이전 버전
화산_00_등산시작.jpg
파일 형식:
JPG 파일(.jpg)
연결 프로그램:
사진
변경(C)...
위치:
C:₩Users₩ebi₩Pictures
크기:
1.36MB (1,430,506 바이트)
디스크 할당 크기:
1.36MB (1,433,600 바이트)
만든 날짜:
2025년 11월 17일 월요일, 오후 12:37:25
수정한 날짜:
2025년 11월 17일 월요일, 오후 12:37:39
액세스한 날짜:
2025년 11월 17일 월요일, 오후 12:37:39
특성:
읽기 전용(R)
숨김(H)
고급(D)...
확인
취소
적용(A)

Hive Metadata

- **Hive Metadata即Hive的元数据。**
- 包含用Hive创建的database、table、表的位置、类型、属性，字段顺序类型等元信息。
- 元数据存储在关系型数据库中。如hive内置的Derby、或者第三方如MySQL等。

Hive Metastore

- **Metastore即元数据服务。Metastore服务的作用是管理metadata元数据，对外暴露服务地址，让各种客户端通过连接metastore服务，由metastore再去连接MySQL数据库来存取元数据。**
- 有了metastore服务，就可以有多个客户端同时连接，而且这些客户端不需要知道MySQL数据库的用户名和密码，只需要连接metastore服务即可。某种程度上也保证了hive元数据的安全。



1.2.6 Metastore配置方式

概述

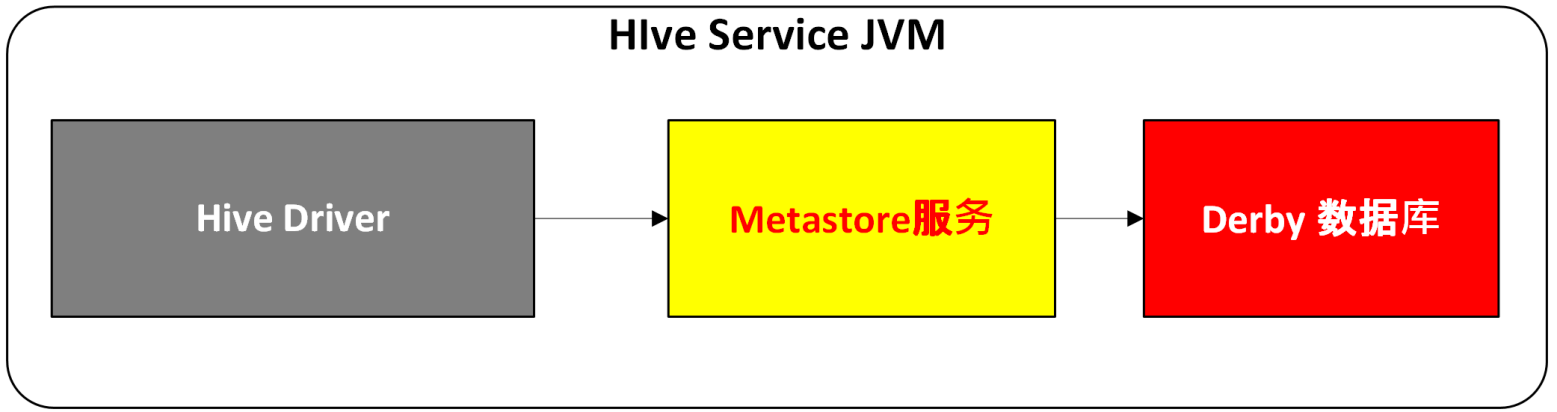
- Hive Metastore服务有三种配置模式：**内嵌模式、本地模式、远程模式。**
- 区分这三种配置方式的关键是弄清楚两个问题：
  - Metastore服务是否需要单独配置、单独启动？
  - Metadata是存储在内置的Derby中，还是第三方RDBMS（比如MySQL）中。
- 本系列课程中使用企业推荐模式——**本地模式部署**。

	内嵌模式	本地模式	远程模式
Metastore单独配置，启动	否	否	是
Metadata存储介质	Derby	MySQL	MySQL

内嵌模式

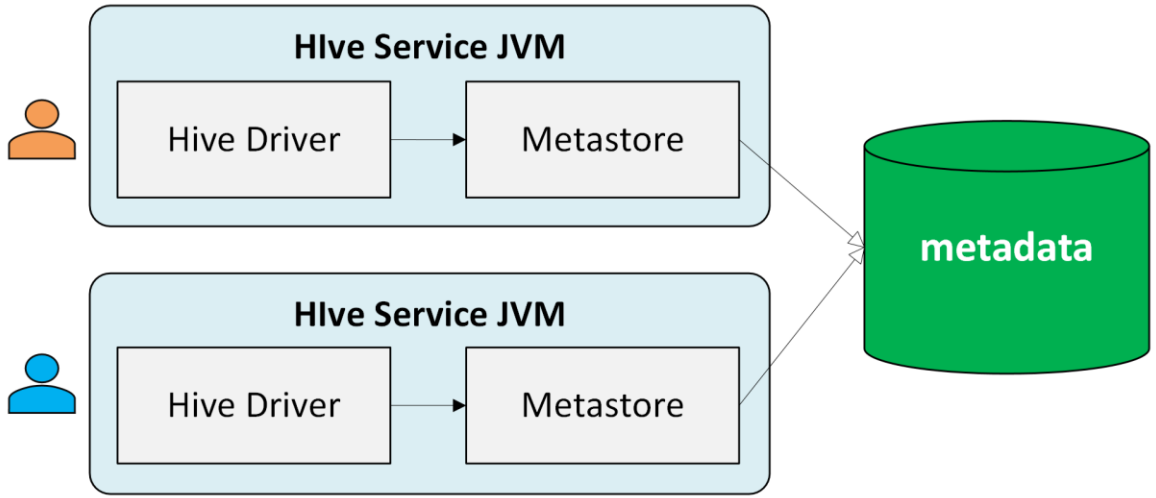
- 内嵌模式（Embedded Metastore）是metastore默认部署模式。
- 此种模式下，元数据存储在内置的Derby数据库，并且Derby数据库和metastore服务都嵌入在主HiveServer进程中，当启动HiveServer进程时，Derby和metastore都会启动。不需要额外起Metastore服务。
- 但是一次只能支持一个活动用户，适用于测试体验，不适用于生产环境。

内嵌模式（Embedded Metastore）



本地模式

- 本地模式（Local Metastore）下，Metastore服务与主HiveServer进程在同一进程中运行，但是存储元数据的数据库在单独的进程中运行，并且可以在单独的主机上。metastore服务将通过JDBC与metastore数据库进行通信。
- 本地模式采用外部数据库来存储元数据，推荐使用MySQL。
- hive根据hive.metastore.uris参数值来判断，如果为空，则为本地模式。
- 缺点是：每启动一次hive服务，都内置启动了一个metastore。



远程模式

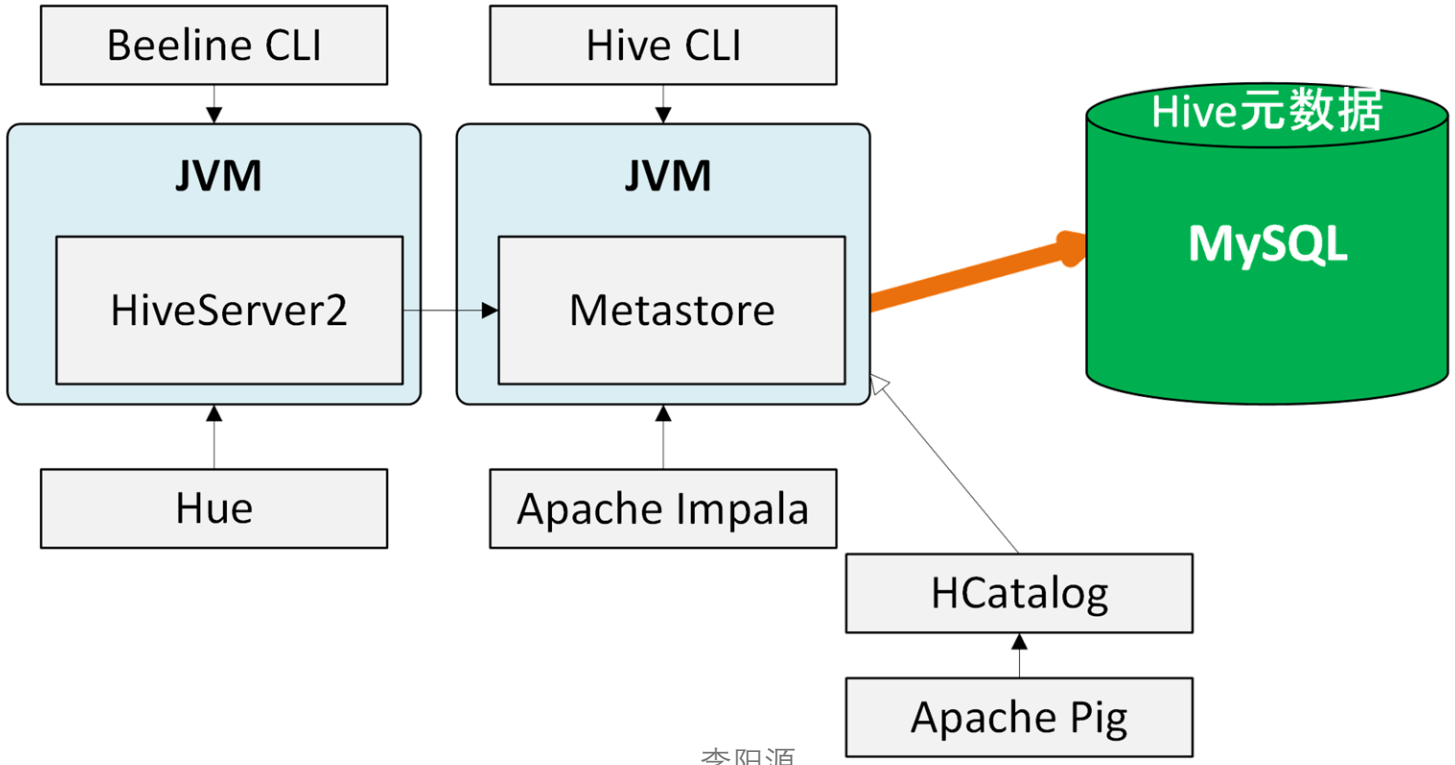
- 远程模式（Remote Metastore）下， Metastore服务在其自己的单独JVM上运行，而不在HiveServer的JVM中运行。

  如果其他进程希望与Metastore服务器通信，则可以使用Thrift Network API进行通信。

- 远程模式下，需要配置hive.metastore.uris 参数来指定metastore服务运行的机器ip和端口，并且需要单独手动启动metastore服务。元数据也采用外部数据库来存储元数据，推荐使用MySQL。

远程模式

- 在生产环境中，建议用远程模式来配置Hive Metastore。在这种情况下，其他依赖hive的软件都可以通过Metastore访问hive。由于还可以完全屏蔽数据库层，因此这也带来了更好的可管理性/安全性。



1.2.7 HiveHive理解度评估选择题（4选1）

问题01:

Hive的主要目的是什么？

A) 实时事务处理
B) 大规模数据的批处理和分析
C) 替代数据库管理系统
D) Web应用程序开发

1.2.7 HiveHive理解度评估选择题（4选1）

问题02:

Hive的查询语言是什么?

A) SQL
B) HQL (Hive Query Language)
C) MapReduce
D) Java

1.2.7 HiveHive理解度评估选择题（4选1）

问题03:

Hive将HQL查询转换为什么?

A) Spark Job
B) Tez DAG
C) MapReduce程序
D) SQL语句

1.2.7 HiveHive理解度评估选择题（4选1）

问题04:

Hive的元数据主要存储在哪里？

A) HDFS
B) 本地文件系统
C) RDBMS (如MySQL, Derby等)
D) NoSQL数据库

1.2.7 HiveHive理解度评估选择题（4选1）

问题05:

Hive的默认数据库名称是什么？

A) master
B) default
C) hive
D) warehouse

1.2.7 HiveHive理解度评估选择题（4选1）

问题06:

Hive中数据物理上存储在何处?

A) 本地磁盘
B) HDFS
C) RDBMS
D) 内存

1.2.7 HiveHive理解度评估选择题（4选1）

问题07:

以下哪项不是Hive支持的执行引擎?

A) MapReduce
B) Tez
C) Spark
D) Flink

1.2.7 HiveHive理解度评估选择题（4选1）

问题08:

Hive中“分区”功能的主要目的是?

A) 数据加密
B) 查询性能优化
C) 数据压缩
D) 自动备份

1.2.7 HiveHive理解度评估选择题（4选1）

问题09:

Hive中“分桶”的主要好处是?

A) 数据加密
B) 优化JOIN操作
C) 加快数据删除速度
D) 节省存储空间

1.2.7 HiveHive理解度评估选择题（4选1）

问题10:

Hive和MySQL之间最大的区别在于?

A) 查询语言
B) 数据存储方式
C) 使用目的 (OLAP vs OLTP)
D) 操作系统兼容性

1.2.7 HiveHive理解度评估选择题（4选1）

问题11:

以下哪项不是Hive Metastore的三种配置模式？

A) 内嵌模式
B) 本地模式
C) 远程模式
D) 分布式模式

1.2.7 HiveHive理解度评估选择题（4选1）

问题12:

在内嵌模式中，元数据存储在哪里？

A) MySQL
B) PostgreSQL
C) Derby
D) HDFS

1.2.7 HiveHive理解度评估选择题（4选1）

问题13:

在本地模式中，hive.metastore.uris参数的值通常是？

A) thrift://localhost:9083
B) jdbc:mysql://localhost:3306
C) 空值
D) hdfs://namenode:9000

1.2.7 HiveHive理解度评估选择题（4选1）

问题14:

在远程模式下，客户端与Metastore服务器之间的通信使用什么协议？

A) HTTP
B) JDBC
C) Thrift
D) FTP

1.2.7 HiveHive理解度评估选择题（4选1）

问题15:

生产环境中推荐的Metastore配置模式是？

A) 内嵌模式
B) 本地模式
C) 远程模式
D) 都一样

1.2.7 HiveHive理解度评估选择题（4选1）

问题16:

以下哪项不是Hive的用户接口？

A) CLI (命令行界面)
B) JDBC/ODBC
C) WebGUI
D) SSH

1.2.7 HiveHive理解度评估选择题（4选1）

问题17:

Hive的Driver（驱动程序）不负责以下哪项工作?

A) HQL语法解析
B) 执行计划生成
C) 数据加密
D) 查询优化

1.2.7 HiveHive理解度评估选择题（4选1）

问题18:

Hive数据模型的基本组成单元不包括以下哪项?

A) 表
B) 分区
C) 分桶
D) 索引

1.2.7 HiveHive理解度评估选择题（4选1）

问题19:

Hive中数据库在HDFS上的默认存储根路径是?

A) /user/hive/warehouse
B) /tmp/hive
C) /var/lib/hive
D) /home/hive

1.2.7 HiveHive理解度评估选择题（4选1）

问题20:

Hive最大的优点是什么?

A) 实时处理
B) 支持复杂事务
C) 可以利用已有的SQL知识
D) 硬件要求低

项目小结

项目小结

我们在项目中首先学习了数据库、数据仓库的概念，其中重点介绍了数据仓库的特点和数据仓库的数据模型以及数据仓库的体系结构，其次进一步学习了Hive数据仓库的基本概念，详细介绍了Hive的架构和Hive的数据类型。在本项目中，不仅理解和掌握数据仓库的基本概念和特点，同时也认识了Hive的基本概念和架构，是后续应用Hive解决实际问题的重要基础。

课后练习

1.简述数据仓库特点?

2.简述Hive架构主要组件和功能?

感谢您的观看



《Hive数据仓库应用教程》