Hive 数据仓库应用教程

第1讲 任务 1-2 认识Hive数据仓库 —— 讲义原稿

授课教师：李阳源

适用对象：专科大学学生 | 语言：中文（口语化） | 难词标注拼音

幻灯片 1 —— 任务 1-2 认识Hive数据仓库（标题页）

同学们好！

上节课我们学了什么是数据仓库。

这节课我们要认识一个具体的数据仓库工具——Hive。

它是当前大数据领域(大数据 lǐng yù)里面最常用的数据仓库软件之一。

幻灯片 2 —— 1.2.1 Apache Hive概述(gài shù)

好，这节课我们要深入(shēn rù)了解Hive。

Hive是一个基于(jī yú)Hadoop的数据仓库工具。

我们会讲到Hive的架构(jià gòu)、基本数据类型、以及它在大数据生态系统(shēng tài xì tǒng)中的角色。

> **【幻灯片图片说明】***图片显示一个未来感的数据中心建筑，周围有各种数据图标和网络连接线，代表大数据处理和分析的场景。建筑顶部有发光的数据流向天空发射，周围是城市建筑群，象征(xiàng zhēng)着数据仓库是整个数据城市的核心。*

幻灯片 3 —— 什么是Hive

什么是Hive?

Apache Hive是一款(yī kuǎn하나의)建立在Hadoop之上的开源(kāi yuán)数据仓库系统。

它能做什么呢?
它可以把存储在Hadoop文件里的数据，映射(ìng shè)成一张数据库表。
然后提供一种类似(lèi sì)SQL的查询语言，叫HQL——Hive Query Language。
用这个HQL就可以查询和分析存储在Hadoop里的大量数据。

Hive的核心是什么呢?
就是把HQL转换(zhuǎn huàn)成MapReduce程序，然后提交到Hadoop集群(jí qún)去执行(zhí xíng)。

还有一个小知识：Hive是Facebook开发的，后来开源给了大家。

> **【幻灯片图片说明】***图片显示了Hive的Logo——一只黄色的卡通图案，同时具有大象的头和蜜蜂(mì fēng)的身体。大象代表Hadoop（因为Hadoop的Logo就是大象），蜜蜂代表Hive（Hive英文意思是‘蜂巢’）。图案下方有大字‘HIVE’。这个Logo非常形象地表达Hive和Hadoop的关系。*

幻灯片 4 —— 为什么使用Hive

为什么不直接用Hadoop MapReduce?

如果直接用Hadoop MapReduce来处理数据，会有什么问题呢?
第一，学习成本太高了！你得先学会Java编程。
这就好比你只是想查一下“哪个商品卖得最好”，

却要先学好几个月的编程语言，太折腾(zhē teng)了。

第二，用MapReduce写复杂(fù zá)的查询逻辑(ì jí)，开发难度太大。

用Hive有什么好处?

第一，操作接口用的是类似SQL的语法，简单、容易上手(róng yì shàng shǒu)。

如果你以前学过MySQL，那学Hive就很快。

第二，不用直接写MapReduce了，减少(jiǎn shǎo)了学习成本。

第三，支持自定义函数(hán shù)，功能扩展(kuò zhǎn)很方便。

第四，背靠(bèi kào)Hadoop，擅长(shàn cháng)存储和分析海量(hǎi liàng)数据。

幻灯片 5-6 —— Hive和Hadoop的关系

我们说过，数据仓库软件至少需要两种能力：

第一，存储数据的能力。

第二，分析数据的能力。

Hive作为大数据时代的数据仓库，当然也有这两种能力。

但是！Hive并不是自己实现的，而是借助(jiè zhù)Hadoop：

Hive用HDFS存储数据，用MapReduce查询分析数据。

可能有同学会说：那Hive不就是个‘套壳(tào ké)’吗?

其实不是！Hive最大的魅力(mèi lì)在于：

你只需要专注于写HQL，Hive帮你转换成MapReduce程序。

用一句话总结：写的是SQL，执行的是MapReduce！

> **【幻灯片图片说明】***图片展示了一个流程：左边是SQL的Logo→中间是Hive的蜜蜂大象Logo→右边是Hadoop MapReduce的大象Logo。三个图标之间用箭头连接，表示用户写SQL，Hive负责翻译(fān yì)，最终由Hadoop执行。下方红色字体写着‘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’。*

幻灯片 7 —— 案例：如何模拟(mó nǐ)实现Hive的功能

好，我们来看一个具体的例子。

假设在HDFS文件系统上有一个文件，路径是 /data/china_user.txt。
文件内容是这样的：

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

大家看，这是一个结构化(jié gòu huà)的数据文件。
每一行有ID、名字、年龄、城市，用逗号分隔(fēn gé)。

需求(xū qiú)是：统计来自上海且年龄大于25岁的用户有多少个？

如果用SQL来写，很简单：

```
SELECT count(*) FROM t_user WHERE city='shanghai' AND age > 25;
```

答案是1个——只有allen（30岁，上海）符合条件。
lisi是25岁，不是大于25，所以不算。

但问题是，这个文件在HDFS上，不是在MySQL数据库里。
我们怎么才能用SQL来查询它呢？这就是Hive要做的事情！

幻灯片 8-9 —— 映射(yìng shè)信息记录

要想用SQL查询HDFS上的文件，首先要解决一个问题：

如何把文件和表对应起来?

在Hive里面，写SQL的前提是针对‘表’，而不是‘文件’。

所以我们需要把文件和表之间的对应关系描述清楚。

这种对应关系就叫做‘映射’。

映射在数学上就是‘一对一对应关系’，比如 y = x + 1。

而记录这些映射信息的数据，专业名词叫做元数据(yuán shù jù)——Metadata。

元数据就是‘描述数据的数据’。

> **【幻灯片图片说明】***图片左侧是一张数据库表t_user，有Id、Name、Age、City四列，包含6条数据（zhangshan、lisi、allen、woon、sangok、yangweon）。右侧是一个文本文件china_user.txt，内容是通号分隔的原始数据。中间用绿色箭头标注‘映射关系’。文件下方指向hdfs://data/china_user.txt→Hadoop HDFS。表示文件存储在HDFS上，通过映射关系可以像表一样被查询。*

幻灯片 10 —— 元数据具体记录什么?

具体来说，元数据要记录哪些信息呢?

第一，表对应的是哪个文件（位置信息）。

比如：t_user表对应的是 /data/china_user.txt 这个文件。

第二，表的每一列对应文件的哪个字段(zì duàn)（顺序信息）。

比如：第一列Id对应文件的第一个字段，Name对应第二个，以此类推。

第三，文件字段之间的分隔符(fēn gé fú)是什么。

比如：这个文件用的是逗号(dòu hào) ',' 来分隔的。

有了这些元数据，Hive就知道怎么把文件当作表来处理了。

幻灯片 11 —— SQL语法解析(jiě xī)、编译(biān yì)

用户写完SQL之后，Hive要做什么呢?

第一步，对SQL进行语法校验(jiào yàn)——检查你写的SQL有没有语法错误。
第二步，根据元数据信息来理解SQL的含义(hán yì)。
第三步，制定执行计划，并把它转换成MapReduce程序。
第四步，执行并把结果返回给用户。

就像翻译官一样：你说中文，翻译官帮你转成英文，然后别人就能听懂了。
Hive就是那个翻译官，把SQL翻译成MapReduce!

幻灯片 12 —— 场景案例总结

好，我们来总结一下刚才的两个问题：

Q1：Hive能把数据文件映射成一张表，这个映射是什么?
答：就是文件和表之间的对应关系，也就是元数据!

Q2：Hive软件本身到底做了什么?
答：SQL语法解析，然后编译(biān yì)成为MapReduce程序!

幻灯片 13 —— 模拟Hive的整体架构图

大家看这张图，这就是Hive的简化(jiǎn huà)架构：

> **【幻灯片图片说明】***图片展示了Hive的工作流程。左上角是用户，步骤如下：1.编写SQL→进入Apache Hive模块（包含UI用户界面、语法解析、SQL语句编辑、执行计划优化）；3.查询元数据→右侧的元数据存储（绿色圆柱体，存储映射信息）；4.编写成MapReduce程序提交YARN执行；5.监督(jiān dū)执行情况；最终从MR→YARN→HDFS读取china_user.txt数据，处理后结果返回给用户。整体分为三大块：Hive本身、元数据存储、Hadoop集群。*

简单说，Hive的工作流程就是：

用户写SQL → Hive解析和编译 → 查询元数据 → 生成MapReduce → 提交Hadoop执行 → 返回结果

幻灯片 14-15 —— 1.2.2 Apache Hive架构图和组件(zǔ jiàn)

好，现在我们来看一下Hive的正式架构图。

> **【幻灯片图片说明】**Hive完整架构图从上到下分为几层：最上层是客户端接入方式：JDBC/ODBC、command-line interface（命令行）、Hive Thrift Server、Hive Web Interface（网页界面）。中间是Hive Driver驱动程序，内部包含：Parser解析器、Planner任务计划、Execution执行器、Optimizer优化器、MS Client元数据客户端。Driver右侧连接到Metastore元数据存储，Metastore下方连接到绿色RDBMS圆柱体（存储元数据的关系型数据库）。Driver下方是三个执行引擎：MapReduce、Tez、Spark。再下方是Hadoop YARN资源管理。最底层是Hadoop HDFS/HBase数据存储。

[다음 슬라이드 넘겨서 시작]

组件一：用户接口(jiē kǒu)

用户可以通过三种方式连接Hive：

CLI——命令行(mìng lìng háng)，就是在终端(zhōng duān)输入命令。

JDBC/ODBC——通过程序连接，类似于用Java连接MySQL。

WebGUI——通过浏览器(liú lǎn qì)访问Hive。

幻灯片 16 —— 组件二：元数据存储

元数据通常存储在关系型数据库里面，比如MySQL或者Derby。

Hive的元数据包括什么呢?
——表的名字、表有哪些列、分区(fēn qū)信息、表的属性(shǔ xìng)、
表的数据存放在HDFS的哪个目录(mù lù)等等。

简单说，就是记录“表长什么样子、数据在哪里”这些信息。

幻灯片 17 —— 组件三：Driver驱动(qū dòng)程序

Driver是Hive的核心，里面包含四个重要部分：

Parser解析器——对SQL进行词法分析和语法分析。
就像老师批改作文，先检查你的句子有没有语法错误。

Planner计划编译器——根据解析结果生成执行计划。
Optimizer优化器——对执行计划进行优化(yōu huà)，让它跑得更快。
Execution执行器——把最终的计划转换成MapReduce程序并执行。

幻灯片 18 —— 组件四：执行引擎(yǐn qíng)

Hive本身并不直接处理数据文件。

它是通过执行引擎来处理的。

目前Hive支持三种执行引擎：

MapReduce——最经典的，但速度较慢。

Tez——比MapReduce快很多。

Spark——基于内存计算，速度最快。

就像你可以选择不同的交通工具去上学：

公交车（慢但稳定）、地铁（较快）、出租车（最快）。

幻灯片 19 —— 1.2.3 Apache Hive数据模型(mó xíng)

Data Model 概念

数据模型是用来描述数据、组织(zǔ zhī)数据和操作数据的方式。

Hive的数据模型和MySQL这些关系型数据库很像，

也有库、表的概念。但它还有自己特有的模型。

Hive中的数据可以分为三类：

Table 表 —— 和MySQL的表一样，基本单位。

Partition 分区(fēn qū) —— 把表按某个列分成不同的部分。

Bucket 分桶(fēn tǒng) —— 把数据按hash规则分成小文件。

【幻灯片图片说明】图片展示了Hive数据模型的层次关系：最上层是DB数据库，下面是Table表，表对应一个Directory目录。目录下可以有Partition分区，分区下可以有Bucket分桶(**fēn tǒng**)。

幻灯片 20-21 —— Databases 数据库

Hive和MySQL一样，也有数据库的概念。

每个数据库下面有各自的表。默认(mò rèn)数据库叫 default。

Hive的数据都存储在HDFS上。

默认的根目录(gēn mù lù)是：/user/hive/warehouse

比如你创建了一个叫 itcast 的数据库，

它在HDFS上的路径就是：/user/hive/warehouse/itcast.db

[여기서 다음장 넘겨서 그림 설명→ 실제 실습으로 표시 가능]

> **【幻灯片图片说明】**幻灯片右侧展示了在Linux终端执行hdfs dfs -ls命令的截图，显示了/user/hive/warehouse目录下的多个数据库文件夹，如dual、employees、hive_db、retail_db.db、test1、test2、test3、test_data、test_db_yangweon等。

幻灯片 22 —— Tables 表

Hive表和关系型数据库的表一样。

但有一个重要区别：

表的数据存在HDFS上，表的元数据存在RDBMS（比如MySQL）里。

表在HDFS上的存储路径是：数据库目录/表名

比如 itcast 数据库下的 t_user 表：

/user/hive/warehouse/itcast.db/t_user

> **【幻灯片图片说明】**图片展示了数据库和表的层次关系。上层有两个数据库：default（/user/hive/warehouse/）和itcast（/user/hive/warehouse/itcast.db）。default下有t_order和t_pay两张表；itcast下有t_student、t_teacher、t_address三张表。底部标注了每张表对应的HDFS路径。

幻灯片 23 —— Partitions 分区(fēn qū)

分区是Hive的一种优化(yōu huà)手段。

什么意思呢?

就是根据某一列的值，把表划分(huà fēn)成不同的部分。

举个例子:

假设有一张访问日志(rì zhì)表 t_accesslog，数据量特别大。

我们可以按照日期(rì qī)来分区:

day=20260202 → 一个文件夹

day=20260203 → 一个文件夹

day=20260204 → 一个文件夹

分区的好处是什么?

查询的时候可以只扫描(sǎo miáo)需要的分区，不用扫描全部数据，速度快很多!

> **【幻灯片图片说明】**图片展示了表default.t_accesslog的分区结构。表对应一个Directory目录，下面按day列分成三个子目录：day=20260202、day=20260203、day=20260204。HDFS上对应路径为/user/hive/warehouse/t_accesslog/day=20260202等。

幻灯片 24-25 —— Buckets 分桶(fēn tǒng)

分桶也是Hive的一种优化手段。

它的原理(yuán lǐ)是：
根据表中某个字段的值，用hash函数计算，然后按照余数分到不同的文件里。
分桶规则：hash(字段) % 桶个数，余数相同的分到同一个文件。

举个例子：假设有三个桶(3个文件)，数据经过hash计算后：
余数为0的→ 放入BUCKET 0
余数为1的→ 放入BUCKET 1
余数为2的→ 放入BUCKET 2

[다음 장으로 넘기면서 설명]

分桶的好处：可以优化join查询，也方便抽样(chōu yàng)查询。

> **【幻灯片图片说明】**图片左侧是普通表t_student_a，数据无规律存储在HDFS的一个文件student_a.txt中。右侧是分桶表t_student_b，通过Hash(ID)%2的规则，数据被有规律地分布到三个文件：0.txt、1.txt、2.txt中。

幻灯片 26-27 —— 1.2.4 Hive和MySQL对比

Hive是要取代(qǔ dài)MySQL吗?

有同学可能会想：Hive和MySQL长得这么像，也有库、表、字段。
Hive难道是大型数据库吗？要取代MySQL吗?

答案是：绝对不是!

[다음 슬라이드 넘기면서 설명]

虽然看起来很像，但它们的应用场景(chǎng jǐng)完全不同：

Hive——定位是数据仓库，用于离线(ì xiàn)数据分析。
MySQL——定位是数据库，用于业务数据事务处理。

我们来对比一下：
定位：Hive是数据仓库 vs MySQL是数据库
场景：Hive用于离线分析 vs MySQL用于业务事务
查询语言：HQL vs SQL
数据存储：HDFS vs 本地文件系统
执行引擎：MR/Tez/Spark vs MySQL内部引擎
执行延迟(yán chí)：高 vs 低
数据规模：大 vs 小
常见操作：导入和查询 vs 增删改查(CRUD)

所以记住：Hive不是大型数据库，也不是要取代MySQL!

幻灯片 28-29 —— 1.2.5 什么是元数据(Metadata)

我们前面已经提到过元数据，现在详细说一下。

元数据（Metadata）就是‘描述数据的数据’。

举个生活中的例子：
你手机里有一张照片，这张照片本身就是‘数据’。
而照片的文件名、大小、拍摄(pāi shè)日期、存储位置——这些就是‘元数据’。

> **【幻灯片图片说明】**图片展示了一只猫的照片，猫本身被标注为‘Data’（数据），旁边有一个信息卡被标注为‘Metadata’（元数据），卡片上写着Filename: Tadzik.jpg、Author: Piotr Kononow、Date: August 15, 2016。非常直观地说明了数据和元数据的关系。

[다음 슬라이드 넘기면서 설명]
另外，PPT上还展示了一个韩语(hán yǔ)的Windows文件属性窗口截图，
显示了一张图片文件的属性信息：文件名、格式、位置、大小、日期等。
这些都是‘描述这张图片的数据’，也就是元数据。

幻灯片 30 —— Hive Metadata 元数据内容

Hive的元数据具体包含哪些信息呢?

——用Hive创建的database、table的名称。
——表在HDFS上的位置。
——表的类型和属性。
——字段的顺序、类型等。

元数据存储在关系型数据库中。
比如Hive内置(nèi zhì)的Derby，或者第三方的MySQL。

幻灯片 31 —— Hive Metastore 元数据服务

Metastore是什么？就是元数据服务。

它的作用是管理元数据，对外提供服务地址。

各种客户端连接Metastore服务，由Metastore去连接MySQL存取元数据。

举个例子：

Metastore就像一个前台(qián tái)接待员。

你不需要直接去后台找数据库，只要告诉前台你要什么就行了。

这样多个客户端可以同时连接，也更安全。

> 【幻灯片图片说明】图片左侧是四个不同颜色的人物图标，代表多个客户端用户。中间是一个灰色方块标注Metastore（元数据服务）。右侧是一个绿色圆柱体标注metadata（元数据存储在MySQL中）。用户先连接Metastore，Metastore再连接metadata数据库。

幻灯片 32-36 —— 1.2.6 Metastore配置(pèi zhì)方式

Metastore服务有三种配置模式：**内嵌模式、本地模式、远程模式**

[다음 슬라이드]

第一种：内嵌模式(nèi qiàn mó shì) Embedded Metastore

这是默认模式。

元数据存在内置的Derby数据库里。

Derby和Metastore都嵌入在Hive主进程里面。

缺点：一次只能支持一个用户，只适合测试(cè shì)，不适合生产环境。

> 【幻灯片图片说明】*内嵌模式图：一个大框标注Hive Service JVM，里面包含Hive Driver（灰色）*

→Metastore服务（黄色）→Derby数据库（红色），三个组件都在同一个JVM进程内。

[다음 슬라이드]

第二种：本地模式(běn dì mó shì) Local Metastore

Metastore服务和Hive在同一个进程里运行。

但是元数据存在外部的MySQL里，不再用Derby了。

这个模式比较实用，也是我们课程中使用的模式。

缺点：每次启动Hive都会内置启动一个Metastore。

【幻灯片图片说明】*本地模式图：两个Hive Service JVM框，每个里面都有Hive Driver和Metastore。两个JVM都连接到同一个外部的绿色metadata圆柱体（MySQL数据库）。*

[다음 슬라이드]

第三种：远程模式(yuǎn chéng mó shì) Remote Metastore

Metastore服务在单独(dān dú)的JVM上运行，不和Hive在一起。

需要单独配置和启动Metastore服务。

元数据也是存在MySQL里。

这是生产环境(shēng chǎn huán jìng)中推荐(tuī jiàn)的模式。

因为其他软件也可以通过Metastore来访问Hive，更安全、更灵活。

【幻灯片图片说明】*远程模式图：左侧有Beeline CLI和Hive CLI两个客户端，分别连接到各自的JVM。HiveServer2和Metastore分开在不同的JVM中。Metastore连接到右侧的MySQL数据库（绿色圆柱，标注Hive元数据）。下方还有Hue、Apache Impala、HCatalog、Apache Pig等其他软件也可以连接Metastore来访问元数据。*

我们来总结一下三种模式的区别：

内嵌模式：Metastore不用单独配置，元数据存Derby，只能一个用户。

本地模式：Metastore不用单独配置，元数据存MySQL，多用户。

远程模式：Metastore要单独配置启动，元数据存MySQL，最安全。

课堂小结

好，同学们，今天‘认识Hive数据仓库’的内容就讲到这里。

我们来回顾(huí gù)一下今天的重点：

1. Hive是基于Hadoop的开源数据仓库，由Facebook开发。
2. Hive的核心是把HQL转换成MapReduce，写的是SQL，执行的是MR。
3. Hive借助Hadoop：HDFS存储数据，MapReduce分析数据。
4. Hive架构四大组件：用户接口、元数据存储、Driver驱动、执行引擎。
5. 数据模型：Database、Table、Partition分区、Bucket分桶。
6. Hive不是数据库，不是要取代MySQL！它是数据仓库，用于离线分析。
7. 元数据是‘描述数据的数据’，Metastore是元数据服务。
8. Metastore三种模式：内嵌、本地、远程。生产环境用远程模式。

下节课我们将进入项目小结和实践(shí jiàn)环节。

谢谢大家！