〔大数据技术丛书〕

Hive

入门与大数据分析实战

迟殿委 著

清華大學出版社
北京

内 容 简 介

Hive 是基于 Hadoop 的一个数据仓库工具，用来进行数据的提取、转换、加载，这是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据的机制。Hive 能将结构化的数据文件映射为一张数据库表，并能提供 SQL 查询分析功能，将 SQL 语句转换成 MapReduce 任务来执行，从而实现对数据进行分析的目的。本书配套示例源码、PPT 课件、教学大纲。

本书共分 11 章，内容包括数据仓库与 Hive、Hive 部署与基本操作、Hive 语法基础、Hive 数据定义、Hive 数据操作、Hive 查询、Hive 函数、Hive 数据压缩、Hive 调优、基于 Hive 的网站流量分析项目实战、旅游酒店评价大数据分析项目实战。最后的两个项目实战（均包括 SQL 和 Java 编程两种解决方法）帮助读者提高 Hive 大数据分析的综合实战能力。

本书可作为Hive数据仓库初学者的入门书，也可作为Hive大数据分析与大数据应用开发工程师的指导手册，还可作为高等院校或者高职高专计算机技术、人工智能、大数据技术及相关专业的教材或教学参考书。

图书在版编目（CIP）数据

Hive 入门与大数据分析实战 / 迟殿委著. —北京：清华大学出版社，2023.5
（大数据技术丛书）
ISBN 978-7-302-63421-8

Ⅰ. ①H… Ⅱ. ①迟… Ⅲ. ①数据库系统－程序设计 Ⅳ. ①TP311.13

中国国家版本馆 CIP 数据核字（2023）第 079534 号

责任编辑：夏毓彦
封面设计：王　翔
责任校对：闫秀华
责任印制：朱雨萌

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-83470000　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：三河市铭诚印务有限公司
经　　销：全国新华书店
开　　本：190mm×260mm　　印　　张：14　　字　　数：377 千字
版　　次：2023 年 5 月第 1 版　　印　　次：2023 年 5 月第 1 次印刷
定　　价：89.00 元

产品编号：102320-01

前　言

如今各个行业都积累了海量的历史数据，并不断产生大量的新数据，数据计量已经发展到PB、EB、ZB、YB，甚至BB、NB、DB级别。由此催生了一门全新的技术——Hive离线计算。Hive是Hadoop生态体系的关键组件之一，它的出现使得海量数据可以继续使用传统的数据分析方法SQL语句来处理，降低了数据分析人员的学习成本。数据分析人员不需要学习新的脚本语言，可以继续使用熟悉的SQL结构化查询语句来分析大规模数据。但是，Hive的SQL语句不再运行在传统的数据库或者数据仓库中，而是运行在大数据分布式并行计算处理平台上。

本书内容

本书内容按照从易到难、理论与实战相结合的思路来组织。俗话说“工欲善其事，必先利其器”，本书在介绍数据仓库和Hive的基本概念之后，马上开始讲解从创建虚拟机、安装Linux操作系统到逐步完成Hive部署的详细过程；然后在部署完成的Hive环境基础上，学习Hive语法基础、Hive数据定义语言、Hive数据操纵语言、Hive数据基本查询等相关操作；接下来深入介绍Hive的其他功能，包括Hive函数、Hive数据压缩、Hive调优等；最后，本书通过网站流量分析项目实战、旅游酒店评价大数据分析项目实战这两个开发案例，帮助读者提升大数据分析的综合实战能力。这两个实战项目都给出了SQL实现和Java编程实现这两种解决方法，为读者做大数据开发起到抛砖引玉的作用。

本书目的

本书目的是带领读者系统掌握Hive大数据分析工具的使用与开发方法，并通过两个综合项目案例帮助读者提高Hive大数据分析的实战能力。

配套示例源码、PPT课件

本书配套示例源码、PPT课件、教学大纲，需要用微信扫描右边二维码获取。如果阅读中发现问题或疑问，请联系booksaga@163.com，邮件主题写“Hive入门与大数据分析实战”。

本书适合的读者

本书可作为 Hive 数据仓库初学者的入门书、Hive 离线大数据分析人员的参考手册，也可作为高校开设大数据平台搭建、数据仓库技术或大数据开发课程的参考教材。

学习本书要求读者有一定的 Java 编程基础并了解 Linux 系统的基础知识。本书每一个章节的实践操作都有详细清晰的步骤讲解，即使读者没有任何大数据基础，也可以对照书中的步骤成功搭建属于自己的大数据分析平台；可以说本书是一本真正能提高读者动手能力、以实操为主的 Hive 入门书。通过本书的学习，结合每章的示例源代码，读者能够迅速理解和掌握 Hive 技术框架，并能熟练使用 Hive 数据仓库进行大数据分析和大数据应用开发。

笔　者

2023 年 3 月

目　　录

第 1 章　数据仓库与 Hive …… 1

1.1　数据仓库概述 …… 1

1.1.1　数据仓库特征与重要概念 …… 1

1.1.2　数据仓库的数据存储方式 …… 2

1.2　Hive 数据仓库简介 …… 5

1.3　Hive 版本和 MapReduce 版本的 WordCount 比较 …… 6

1.4　Hive 和 Hadoop 的关系 …… 7

1.5　Hive 和关系数据库的异同 …… 8

1.6　Hive 数据存储简介 …… 9

第 2 章　Hive 部署与基本操作 …… 11

2.1　Linux 环境的搭建 …… 11

2.1.1　VirtualBox 虚拟机安装 …… 11

2.1.2　安装 Linux 操作系统 …… 13

2.1.3　SSH 工具与使用 …… 19

2.1.4　Linux 统一设置 …… 21

2.2　Hadoop 伪分布式环境的搭建 …… 23

2.2.1　安装本地模式运行的 Hadoop …… 23

2.2.2　Hadoop 伪分布式环境的准备 …… 25

2.2.3　Hadoop 伪分布式的安装 …… 29

2.3　Hadoop 完全分布式环境的搭建 …… 35

2.3.1　Hadoop 完全分布式集群的搭建 …… 35

2.3.2　ZooKeeper 高可靠集群的搭建 …… 40

2.3.3　Hadoop 高可靠集群的搭建 …… 44

2.4　Hive 的安装与配置 …… 53

2.4.1　Hive 的安装与启动 …… 53

2.4.2　基本的 SQL 操作命令 …… 54

2.5　Hive 的一些命令 …… 56

2.5.1　显示 Hive 的帮助 …… 56

2.5.2　显示 Hive 某个命令的帮助 …… 56

2.5.3　变量与属性 …… 56

2.5.4 指定 SQL 语句或文件 …… 57
2.5.5 显示表头 …… 58
2.6 Hive 元数据库 …… 58
2.6.1 Derby …… 58
2.6.2 MySQL …… 60
2.7 MySQL 的安装 …… 61
2.8 配置 MySQL 保存 Hive 元数据 …… 62
2.9 HiveServer2 与 Beeline 配置 …… 65

第 3 章 Hive 语法基础 …… 68

3.1 数据类型列表 …… 68
3.2 集合类型 …… 69
3.2.1 array 测试 …… 70
3.2.2 map 测试 …… 71
3.2.3 struct 测试 …… 71
3.3 数据类型转换 …… 72
3.4 运算符 …… 73
3.5 Hive 表存储格式 …… 74
3.6 Hive 的其他操作命令 …… 75
3.7 Hive 分析 Tomcat 日志案例 …… 76

第 4 章 Hive 数据定义 …… 79

4.1 数据库的增删改查 …… 79
4.1.1 在默认位置创建数据库 …… 79
4.1.2 指定目录创建数据库 …… 80
4.1.3 显示当前使用的数据库 …… 81
4.1.4 删除数据库 …… 81
4.2 创建内部表 …… 81
4.3 使用关键字 external 创建外部表 …… 83
4.3.1 指定现有目录 …… 84
4.3.2 先创建表，再指定目录 …… 84
4.3.3 显示某个表或某个分区的信息 …… 85
4.4 创建分桶表 …… 86
4.5 分区表 …… 89
4.5.1 创建和显示分区表 …… 89
4.5.2 增加、删除和修改分区 …… 90
4.6 显示某张表的详细信息 …… 92
4.7 指定输入输出都是 SequenceFile 类型 …… 94

4.8 关于视图 94
4.8.1 使用视图降低查询的复杂度 94
4.8.2 查看视图的信息 95
4.8.3 删除视图 95

第 5 章 Hive 数据操作 96

5.1 向表中装载数据 96
5.2 通过 Insert 向表中插入数据 97
5.3 动态分区插入数据 98
5.4 创建表并插入数据 100
5.5 导出数据 100

第 6 章 Hive 查询 103

6.1 Select...From 语句 103
6.2 Select 基本查询 104
6.3 Where 语句 105
6.4 Group By 语句 107
6.5 Join 语句 108
6.6 排序 110
6.6.1 Order By 110
6.6.2 Sort By 112
6.6.3 Distribute By 113
6.6.4 Cluster By 114
6.7 抽样查询 114

第 7 章 Hive 函数 117

7.1 查看系统内置函数 117
7.2 常用内置函数 117
7.3 Hive 的其他函数 121
7.3.1 准备数据 121
7.3.2 其他函数的使用 121
7.3.3 显示某个函数的帮助信息 131
7.4 自定义函数 132
7.4.1 Hive 自定义 UDF 的过程 132
7.4.2 Hive UDTF 函数 135

第 8 章 Hive 数据压缩 138

8.1 数据压缩格式 138

8.2 数据压缩配置……139
8.2.1 Snappy 压缩方式配置……139
8.2.2 MapReduce 支持的压缩编码……141
8.2.3 MapReduce 压缩参数配置……142
8.3 开启 Map 端和 Reduce 端的输出压缩……142
8.4 常用 Hive 表存储格式比较……144
8.5 存储与压缩相结合……148

第 9 章 Hive 调优……151

9.1 Hadoop 计算框架特性……151
9.2 Hive 优化的常用手段……151
9.3 Hive 优化要点……152
9.3.1 全排序……152
9.3.2 怎样做笛卡儿积……156
9.3.3 怎样写 exist/in 子句……156
9.3.4 怎样决定 Reducer 个数……156
9.3.5 合并 MapReduce 操作……157
9.3.6 Bucket 与 Sampling……157
9.3.7 Partition……158
9.3.8 Join……158
9.3.9 数据倾斜……160
9.3.10 合并小文件……161
9.3.11 Group By……163

第 10 章 基于 Hive 的网站流量分析项目实战……164

10.1 项目需求及分析……164
10.1.1 数据集及数据说明……164
10.1.2 功能需求……165
10.2 利用 Java 实现数据清洗……165
10.2.1 数据上传到 HDFS……166
10.2.2 http.log 数据清洗……166
10.2.3 phone.txt 数据清洗……170
10.3 利用 MySQL 实现数据清洗……173
10.3.1 http.log 数据清洗……173
10.3.2 phone.txt 数据清洗……175
10.4 数据分析的实现……176
10.4.1 创建 Hive 库和表……176
10.4.2 使用 SQL 进行数据分析……176

第 11 章 旅游酒店评价大数据分析项目实战 …… 180
11.1 项目介绍 …… 180
11.2 项目需求及分析 …… 181
11.2.1 数据集及数据说明 …… 181
11.2.2 功能需求 …… 183
11.3 利用 Java 实现数据清洗 …… 184
11.3.1 本地 Hadoop 运行环境搭建 …… 184
11.3.2 数据上传到 HDFS …… 186
11.3.3 Hadoop 数据清洗 …… 189
11.4 利用 MySQL 实现数据清洗 …… 192
10.4.1 hotelbasic.csv 数据清洗 …… 192
10.4.2 hoteldata.csv 数据清洗 …… 193
11.5 数据分析的实现 …… 194
11.5.1 构建 Hive 数据仓库表 …… 194
11.5.2 导出结果数据到 MySQL …… 197
11.6 分析结果数据可视化 …… 200
11.6.1 数据可视化开发 …… 200
11.6.2 数据可视化部署 …… 208

第1章

数据仓库与 Hive

1.1 数据仓库概述

数据仓库是在数据库的基础上建立起来的，但与传统的数据库又有较大的不同，它将分布在不同数据库中的数据集成起来，将转换后的关系型数据及其他复杂类型数据存储成为一种面向分析的数据集合。

1.1.1 数据仓库特征与重要概念

1. 数据仓库一般具有的特征

1）数据仓库的数据是面向主题的

主题是一个抽象的概念，是在较高层次上综合、归纳企业信息系统中的数据并进行分析利用的抽象。在逻辑意义上，它对应着企业中某一宏观分析领域所涉及的分析对象。

2）数据仓库的数据是集成的

在数据进入数据仓库之前，必然要经过加工与集成，对不同的数据来源统一数据结构和编码，将原始数据由面向应用转向面向主题。

3）数据仓库的数据是可更新的

数据仓库的数据主要供企业决策分析之用，所涉及的数据操作主要是数据查询，一般情况下并不进行修改操作，因而数据经集成进入数据仓库后是极少或根本不更新的。

4）数据仓库的数据是随时间变化的

数据仓库中的数据不可更新是针对应用来说的，也就是说，数据仓库的用户在进行分析处理时是不进行数据更新操作的，但并不是说，在从数据集成输入数据仓库开始到最终被删除的整个数据生存周期中，所有的数据仓库都是永远不变的。数据仓库内的数据时限一般在 5～10 年，故数据的

编码包含时间项。数据仓库要周期性地收集和整理数据，以适应决策支持系统。

2. 数据仓库中的几个重要概念

下面再介绍一下数据仓库中的几个重要概念：粒度、分割、维、元数据。

1）粒度

粒度是指数据仓库中数据单元的详细程度和级别。一般操作型系统中处理的数据都是详细数据，其粒度是最低的。但在分析型处理中需要通过数据的概括和聚集形成较高粒度的数据。粒度越小，细节程度越高，级别就越低；反之，数据的综合程度越高，粒度越大，级别就越高。数据的粒度越高，所需要存储的数据量越少，但对决策者的重要性却随之增加，且能够为用户提供快速方便的查询。数据仓库一般提供多种粒度的数据，不同粒度的数据用于不同类型的处理。比如销售产品，其粒度可以是每天的数据，也可以是每周、每月、每季度，甚至每年记录统计的数据。通常的数据粒度有详细数据、轻度综合、高度综合三级。

2）分割

分割是指将逻辑上统一的数据分割成较小的、可以独立管理的物理单元进行存储，以便于提高数据处理效率，数据分割后的单元称为分片。数据分割的标准是按照实际情况确定的，通常按日期、地理分布、业务范围等进行分割。数据分割后较小单元的数据处理相对独立，使得数据更易于重构、索引、恢复和监控，处理起来更快。比如产品销售数据可以按照不同地域（如北京地区、东北地区、华北地区等）进行分割。

3）维

维是人们观察数据的特定角度，是数据的视图。比如可以从销售时间、销售地区分布等不同角度来观察产品销售数据。维可以有细节程度的不同描述方面，这些不同描述方面称为维的不同维层次。最常用的维是时间维，时间维的维层次可以有日、周、月、季、年等。数据仓库中的数据按照不同的维组织起来形成一个多维立方体。

4）元数据

所谓元数据就是关于数据的数据，它描述了数据的结构、内容、码、索引等项内容。传统数据库中的数据字典就是一种元数据，但在数据仓库中，元数据的内容比数据库中的数据字典更丰富、更复杂。

1.1.2 数据仓库的数据存储方式

数据仓库的数据存储方式一般说来有两种，即基于关系表的存储方式和基于多维数据库的存储方式。

1. 基于关系表的存储方式

基于关系表的存储方式是将数据仓库的数据存储在关系数据库的表结构中，在元数据的管理下完成数据仓库的功能。这种组织方式在建库时有两个主要过程用以完成数据的抽取：首先要提供一种图形化的点击操作界面，使分析员能对源数据的内容进行选择、定义多维数据模型；然后再编制程序，把数据库中的数据抽取到数据仓库的数据库中。基于关系表的数据存储方式主要有星型模型

和雪花模型两种。

1）星型模型

大多数数据仓库都采用“星型模型”来表示多维概念模型。数据库中包括一张“事实表”，对于每一维都有一张“维表”。“事实表”中的每条元组都包含指向各个“维表”的外键和一些相应的测量数据。“维表”中记录的是有关这一维的属性。销售数据仓库的星型模型图如图 1-1 所示。

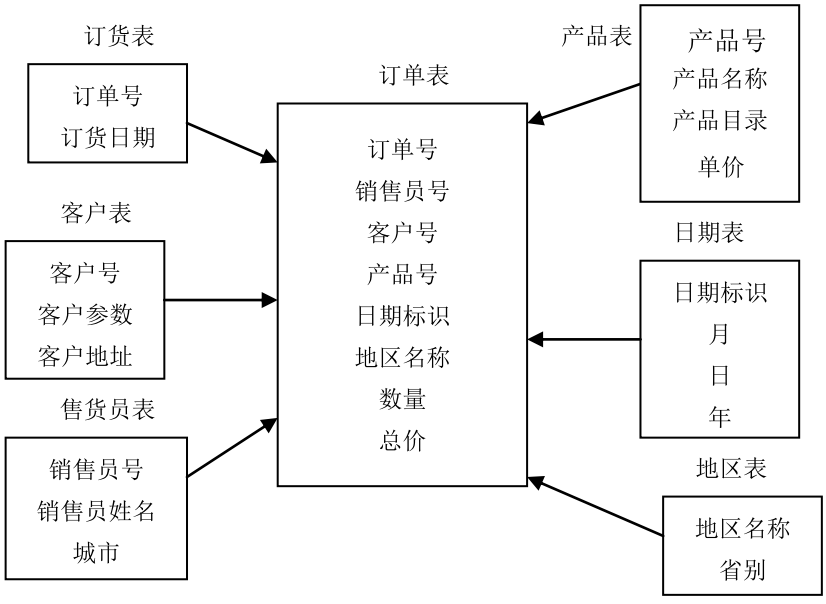

图 1-1　销售数据仓库的星型模型

事实表中的每一个元组包含一些指针（是外键，主键在其他表中），每个指针指向一张维表，这就构成了数据库的多维联系。相应每个元组中多维外键限定数据测量值。在每张维表中，除包含每一维的主键外，还有说明该维的一些其他属性。维表记录了维的层次关系。在数据仓库模型中执行查询的分析过程，需要花费大量时间在相关各表中寻找数据。而星型模型使数据仓库的复杂查询可以直接通过各维的层次比较、上钻、下钻等操作完成。

星型模型的数据组织方式存在数据冗余、多维操作速度慢的缺点，但这种方式是主流方案，大多数数据仓库集成方案都采用这种方式。

2）雪花模型

“雪花模型”是对星型模型的扩展，它进一步层次化了星型模型的维表，原有各维表可能被扩展为小的事实表，形成一些局部的“层次”区域。对应的雪花模型图如图 1-2 所示。

雪花模型的优点：通过最大限度地减少数据存储量及联合较小的维表来改善查询性能。

雪花模型增加了用户必须处理的表数量，增加了某些查询的复杂性，降低了系统的通用程度，但同时这种方式可以使系统进一步专业化和实用化。前端工具仍然要用户在雪花的逻辑概念模式上操作，然后将用户的操作转化为具体的物理模式，从而完成对数据的查询。

从功能结构的划分来看，数据仓库系统至少应包含数据获取（Data Acquisition）、数据存储（Data Storage）、数据访问（Data Access）三个核心部分。

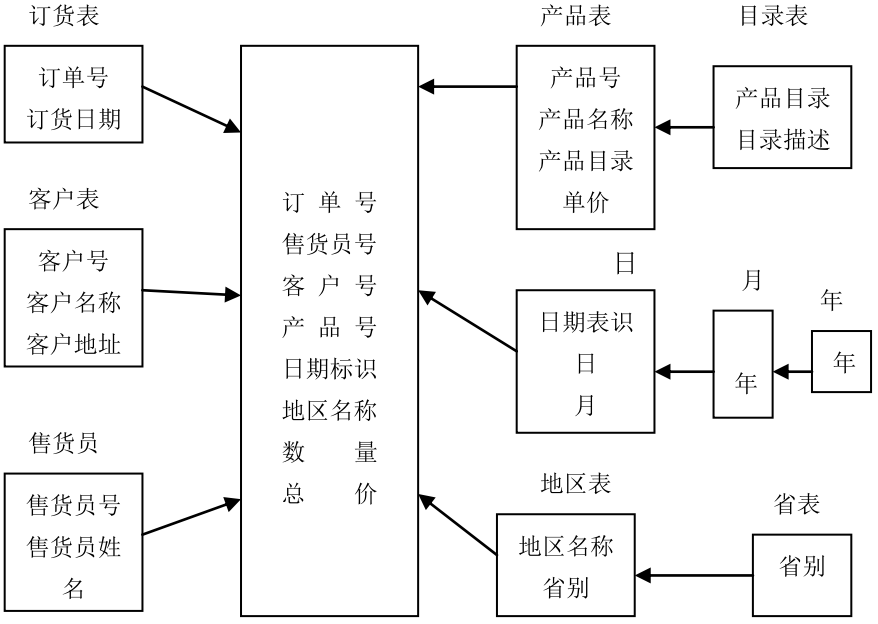

图 1-2　雪花模型

数据源是数据仓库系统的基础，是整个数据仓库系统的数据源泉。数据通常存储在关系数据库中，比如 Oracle 或者 MySQL。数据也可能来自文档资料，比如 CSV 文件或者 TXT 文件。数据还可能来自一些其他的文件系统。数据库是整个数据仓库系统的核心，是数据存放的地方，并能提供对数据检索的支持。

对不同的数据进行抽取（Extract）、转换（Transform）和装载（Load）的过程，也就是通常所说的 ETL 过程。

抽取是指把数据源的数据按照一定的方式从各种各样的存储方式中读取出来。对各种不同数据存储方式的访问能力是数据抽取工具的关键。因为不同数据源的数据格式可能会有所不同，不一定能满足业务的需求，所以还要按照一定的规则进行转换。数据转换包括：删除对决策应用没有意义的数据，将数据转换为统一的数据名称、定义及格式，计算统计和衍生数据，为缺失值赋予默认值，把不同的数据定义方式进行统一。只有转换后符合要求的数据才能进行装载。装载就是将满足格式要求的数据存储到数据仓库中。

2. 基于多维数据库的存储方式

多维数据的组织包括维数据组织和度量数据组织两个方面。维数据组织主要是组织多维数组数据结构和存储维的结构信息，度量数据则是以提高聚集查询的效率来进行组织的。

1）维数据组织

组织维数据首先要对维成员层次进行组织，然后将维成员映射为多维数组的坐标值。

2）度量数据组织

OLAP 操作通常需要处理整个多维数组，由于数据仓库的数据是海量数据，多维数组中的记录数一般很大，经常会超出系统内存，因此需要将多维数组进行分块（Chunk）。OLAP 操作以 I/O 块

大小为单位进行存取，这样可显著提高数据访问的性能。

1.2　Hive 数据仓库简介

Hive 是基于 Hadoop 的一个数据仓库工具，用来进行数据的提取、转化、加载，这是一种可以查询和分析存储在 Hadoop 中的大规模数据的机制。Hive 数据仓库工具能将结构化的数据文件映射为一张数据库表，并提供 SQL 查询功能，能将 SQL 语句转换成 MapReduce（简称 MR）任务来执行。

关于 Hive 的描述可以归结为以下几点：

- Hive 是工具。
- Hive 可以用来构建数据仓库。
- Hive 具有类似 SQL 的操作语句 HQL。
- Hive 是用来开发 SQL 类型脚本，用于开发 MapReduce 操作的平台。

Hive 最初由 Facebook 开源，用于解决海量结构化日志的数据统计分析，是建立在 Hadoop 集群的 HDFS 上的数据仓库基础框架，其本质是将类 SQL 语句转换为 MapReduce 任务来运行。可以通过类 SQL 语句快速实现简单的 MapReduce 统计计算，十分适合数据仓库的统计分析。

所有 Hive 处理的数据都存储在 HDFS 中，Hive 在加载数据过程中不会对数据进行任何修改，只是将数据移动或复制到 HDFS 中 Hive 设定的目录下，因此 Hive 不支持对数据的改写和添加，所有数据都是在加载时确定的。

Hive 总体来说具有以下特点：

（1）Hive 是一个构建在 Hadoop 上的数据仓库框架。

（2）Hive 设计的目的是让精通 SQL 技能、但 Java 编程技能相对较弱的数据分析师能够快速进行大数据分析项目的开发与应用。

非结构化数据分析步骤如图 1-3 所示，其中 Hive 的能力在于直接分析通过 ETL 清洗过后的半结构化数据。

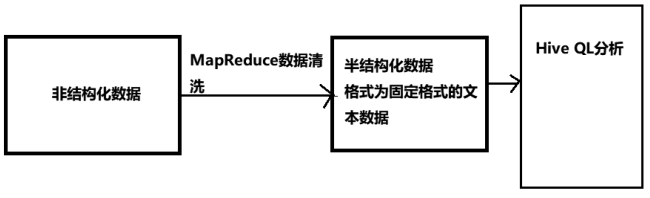

图 1-3　非结构化数据分析步骤

1.3 Hive 版本和 MapReduce 版本的 WordCount 比较

1. MapReduce 版本的 WordCount

读者之前应该都学习过 Hadoop 的 MapReduce 框架应用的相关知识，因此这里就不写完整的 MapReduce 应用的代码了。纯使用 MapReduce 方式的整个流程比较复杂，如果需要修改某个部分，那么首先需要修改代码中的逻辑，然后把代码打包上传到某个可访问路径上（一般就是 HDFS），再在调度平台内执行。如果是改动较大的情况，则可能还会需要在测试环境中多次调试。总之，就是会花比较多的时间在非业务逻辑改动的工作上。本节用于说明 Hive 和 MapReduce 二者的主要区别，具体的操作验证可在第 6 章之后进行。

2. Hive 版本的 WordCount

使用 Hive 来开发一个 WordCount 程序的基本流程如下：

步骤 01 创建表：

```
hive> create table docs(line string);
OK
Time taken: 0.232 seconds
```

步骤 02 导入数据（首先在/home/hadoop 路径下创建一个文本文件 derby.log，hadoop 是笔者的 CentOS 系统用户名，这个用户名在安装 Linux 时创建的）：

```
hive> load data local inpath '/home/hadoop/derby.log' into table docs;
Loading data to table default.docs
Table default.docs stats: [numFiles=1]
OK
Time taken: 1.119 seconds
```

步骤 03 编写 Hive 版本的 WordCount，开发极其简单：

```
hive> create table word_count as
    > select word,count(1) as count from
    > (select explode(split(line,'\s+')) as word from docs) w group by word
    > order by word;
```

上面语句直接将查询运算的结果保存到 word_count 表中去。

由此可以看出 Hive 和 MapReduce 二者的主要区别在于下面两点。

1）运算资源消耗

从时间、数据量、计算量上来看，一般情况下 MapReduce 都是优于或者等于 Hive 的。MapReduce 的灵活性毋庸置疑。在转换到 Hive 的过程中，会有一些为了实现某些场景的需求而不得不用多步 Hive 来实现的情形。

2）开发成本/维护成本

毫无疑问，Hive 的开发成本远低于 MapReduce。后面会介绍，如果能熟练地运用 udf 和 transform，那么 Hive 的开发效率会更高。另外，由于使用了 SQL 语法对数据进行操作，因此处理起来非常直观，也让 Hive 开发更加容易上手。

1.4　Hive 和 Hadoop 的关系

Hive 构建在 Hadoop 之上，二者的关系示意图如图 1-4 所示。

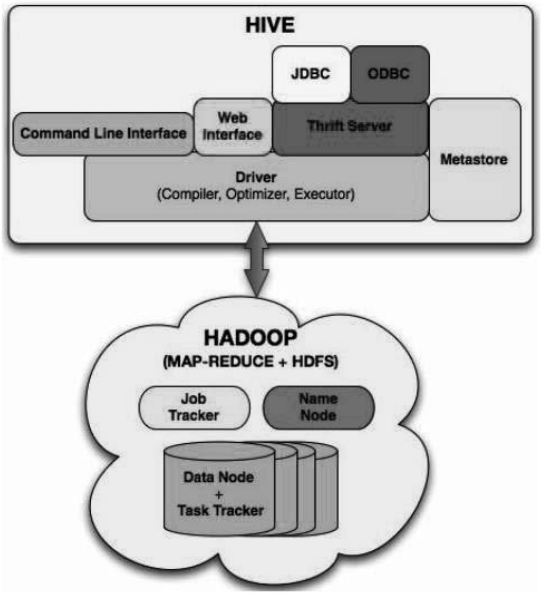

图 1-4　Hive 与 Hadoop 关系

它们关系解释如下：

- Hive 对外提供 CLI、Web Interface（Web 接口）、JDBC、ODBC 等访问接口，Hadoop 提供后台存储和计算服务。
- HQL 中对查询语句的解释、优化、生成查询计划都是由 Hive Diver 完成的。
- 所有的数据都存储在 Hadoop 的 HDFS 中。
- 查询计划被转换为 MapReduce 任务，在 Hadoop 中执行（但要注意有些查询也可能没有 MapReduce 任务，如 select * from table）。
- Hadoop 和 Hive 都是用 UTF-8 编码的。

总之，Hive 是 Hadoop 的延伸，是一个提供了查询功能的数据仓库核心组件，Hadoop 底层的 HDFS 为 Hive 提供了数据存储，MapReduce 为 Hive 提供了分布式运算。HDFS 上存储着海量的数据，如果要对这些数据进行计算和分析，那么需要使用 Java 编写 MapReduce 程序来实现，但 Java 编程门槛较高，且一个 MapReduce 程序写起来要几十、上百行。而 Hive 可以直接通过 SQL 操作 Hadoop，SQL 简单易写、可读性强，Hive 将用户提交的 SQL 解析成 MapReduce 任务供 Hadoop 直接运行。Hive 某种程度来说也不进行数据计算，只是个解释器，只负责将用户对数据处理的逻辑通过 SQL 编程提交后解释成 MapReduce 程序，然后将这个 MapReduce 程序提交给 YARN 进行调度执行，因此，实际进行分布式运算的是 MapReduce 程序。

1.5 Hive 和关系数据库的异同

Hive 数据仓库与传统意义上的数据库是有区别的。一般来说，基于传统方式，可以用 Oracle 数据库或 MySQL 数据库来搭建数据仓库，数据仓库中的数据保存在 Oracle 或 MySQL 数据库中。Hive 数据仓库和它们不同的是，Hive 数据仓库建立在 Hadoop 集群的 HDFS 之上，也就是说，Hive 数据仓库中的数据是保存在 HDFS 上的。Hive 数据仓库可以通过 ETL 的形式来抽取、转换和加载数据。Hive 提供了类似 SQL 的查询语句 HQL，可以用“select*from 表名;”来查询 Hive 数据仓库中的数据，这与关系数据库的操作是一样的。

关系数据库都是为实时查询的业务而设计的，而 Hive 则是为对海量数据进行数据挖掘而设计的，Hive 实时性差，但它很容易扩展自己的存储能力和计算能力。Hive 与关系数据库的对比如表 1-1 所示。

表1-1 Hive与关系数据库的对比

对 比 项	Hive	RDBMS
查询语言	HQL	SQL
数据存储	HDFS	Raw Device 或 Local FS
数据格式	没有定义专门的数据格式	有专门的数据格式
数据更新	不支持对数据的改写和添加	允许添加、修改数据
索引	无	有
执行	MapReduce	Executor
执行延迟	高	低
可扩展性	可扩展性与 Hadoop 一致	受 ACID 语义的严格限制、扩展性非常有限
处理数据规模	大	小

（1）查询语言。由于 SQL 被广泛地应用在数据仓库中，因此，专门针对 Hive 的特性设计了类 SQL 的查询语言 HQL。熟悉 SQL 开发的开发者可以很方便地使用 Hive 进行开发。

（2）数据存储位置。Hive 是建立在 Hadoop 之上的，所有 Hive 的数据都存储在 HDFS 中。数据库则可以将数据保存在块设备或者本地文件系统中。

（3）数据格式。Hive 中没有定义专门的数据格式，数据格式可以由用户指定，用户定义数据格式需要指定三个属性：列分隔符（通常为空格、\t、\x001）、行分隔符（\n）以及读取文件数据的方法（Hive 中默认有三个文件格式，即 TextFile、SequenceFile 以及 RCFile）。由于在加载数据的过程中，不需要进行从用户数据格式到 Hive 定义的数据格式的转换，因此，Hive 在加载的过程中不会对数据本身进行任何修改，而只是将数据内容复制或者移动到相应的 HDFS 目录中。在数据库中，不同的数据库有不同的存储引擎，都定义了自己的数据格式，并且所有数据都会按照一定的组织形式进行存储，因此，数据库加载数据的过程会比较耗时。

（4）数据更新。由于 Hive 是针对数据仓库应用设计的，而数据仓库的内容读多写少，因此，

Hive 不支持对数据的改写和添加，所有的数据都是在加载的时候就确定好的。数据库中的数据通常需要进行修改，因此可以使用 INSERT INTO...VALUES 添加数据，使用 UPDATE...SET 修改数据。

（5）索引。之前已经说过，Hive 在加载数据的过程中不会对数据进行任何处理，甚至不会对数据进行扫描，因此也没有对数据中的某些 key 建立索引。Hive 要访问数据中满足条件的特定值时，需要暴力扫描整个数据，因此访问延迟较高。由于 MapReduce 的引入，Hive 可以并行访问数据，因此即使没有索引，对于大数据量的访问，Hive 仍然可以体现出优势。数据库中，通常会针对一个或者几个列建立索引，因此对于少量的特定条件的数据的访问，数据库可以有很高的效率、较低的延迟。由于数据的访问延迟较高，因此决定了 Hive 不适合在线数据查询。

（6）执行。Hive 中大多数查询的执行是通过 Hadoop 提供的 MapReduce 来实现的（类似 select * from tbl 的查询不需要 MapReduce）。而数据库通常有自己的执行引擎。

（7）执行延迟。之前提到，Hive 在查询数据的时候，由于没有索引，需要扫描整个表，因此延迟较高。另外一个导致 Hive 执行延迟高的因素是 MapReduce 框架。由于 MapReduce 本身具有较高的延迟，因此在利用 MapReduce 执行 Hive 查询时，也会有较高的延迟。相对地，数据库的执行延迟较低。当然，这个低是有条件的，即数据规模较小，当数据规模大到超过数据库的处理能力的时候，Hive 的并行计算显然更有优势。

（8）可扩展性。由于 Hive 建立在 Hadoop 之上，因此 Hive 的可扩展性和 Hadoop 的可扩展性是一致的（国内规模较大的 Hadoop 集群平台提供者有百度和阿里巴巴等大型互联网企业。截至目前，百度 Hadoop 集群规模达到近十个，单集群超过 2800 台机器节点，Hadoop 机器总数有上万台）。而数据库由于 ACID 语义的严格限制，扩展性非常有限。目前最先进的并行数据库 Oracle 在理论上的扩展能力也只有 100 台左右。

（9）数据规模。由于 Hive 建立在集群上并可以利用 MapReduce 进行并行计算，因此可以支持很大规模的数据。数据库可以支持的数据规模较小。

1.6　Hive 数据存储简介

首先，Hive 没有专门的数据存储格式，也没有为数据建立索引，用户可以非常自由地组织 Hive 中的表，只需要在创建表的时候告诉 Hive 数据中的列分隔符和行分隔符，Hive 就可以解析数据。

其次，Hive 中所有的数据都存储在 HDFS 中。Hive 中包含以下数据模型：Table、External Table、Partition、Bucket。

1）Hive

Hive 中的 Table 和数据库中的 Table 在概念上是类似的，每一个 Table 在 Hive 中都有一个相应的目录存储数据。例如，一张表 htduan，它在 HDFS 中的路径为/warehouse/htduan，其中，warehouse 是在 hive-site.xml 中由${hive.metastore.warehouse.dir}指定的数据仓库的目录，所有的 Table 数据（不包括 External Table）都保存在这个目录中。

2）External Table

External Table 指向已经存储在 HDFS 中的数据，可以创建 Partition。它和 Table 在元数据的组

织上是相同的，而实际数据的存储则有较大的差异。

对于 Table 的创建过程和数据加载过程（这两个过程可以在同一个语句中完成），在加载数据的过程中，实际数据会被移动到数据仓库目录中，之后对数据的访问将会直接在数据仓库目录中完成；删除表时，表中的数据和元数据将会被同时删除。External Table 只有一个过程，加载数据和创建表同时完成（CREATE EXTERNAL TABLE…LOCATION），实际数据是存储在 LOCATION 后面指定的 HDFS 路径中，并不会移动到数据仓库目录中；当删除一个 External Table 时，仅删除了 Hive 的元数据。

3）Partition

Partition 对应于数据库中的 Partition 列的密集索引，但是 Hive 中 Partition 的组织方式和数据库中的有很大不同。在 Hive 中，表中的一个 Partition 对应于表下的一个目录，所有的 Partition 的数据都存储在对应的目录中。例如，htduan 表中包含 dt 和 ctry 两个 Partition，则对应于 dt=20100801、ctry=US 的 HDFS 子目录为/warehouse/htduan/dt=20100801/ctry=US，对应于 dt=20100801、ctry=CA 的 HDFS 子目录为/warehouse/htduan/dt=20100801/ctry=CA。

4）Bucket

Bucket 对指定列计算 hash，根据 hash 值切分数据，目的是并行。每一个 Bucket 对应一个文件。例如，将 user 列分散至 32 个 Bucket，对 user 列的值计算 hash，对应 hash 值为 0 的 HDFS 目录为/warehouse/htduan/dt=20100801/ctry=US/part-00000，hash 值为 20 的 HDFS 目录为/warehouse/htduan/dt=20100801/ctry=US/part-00020。