

HZ BOOKS
华章科技

内容全面，涵盖Hadoop技术本身以及与其相关的Hive、HBase、Mahout、Pig、ZooKeeper、Avro、Chukwa等所有子项目

实战性强，为各个知识点精心设计了大量经典的案例，易于理解，可操作性强

實戰



陆嘉恒 著

Hadoop in Action

Hadoop 实战



机械工业出版社
China Machine Press

實戰



Hadoop in Action

Hadoop 实战



陆嘉恒 著



YZLI0890101857



机械工业出版社
China Machine Press

本书是一本系统且极具实践指导意义的 Hadoop 工具书和参考书。内容全面，对 Hadoop 整个技术体系进行了全面的讲解，不仅包括 HDFS 和 MapReduce 这两大核心内容，而且还包括 Hive、HBase、Mahout、Pig、ZooKeeper、Avro、Chukwa 等与 Hadoop 相关的子项目的内容。实战性强，为各个知识点精心设计了大量经典的小案例，易于理解，可操作性强。

全书一共 18 章：第 1 章全面介绍了 Hadoop 的概念、优势、项目结构、体系结构，以及它与分布式计算的关系；第 2 章详细讲解了 Hadoop 集群的安装和配置，以及常用的日志分析技巧；第 3 章分析了 Hadoop 在 Yahoo!、eBay、Facebook 和百度的应用案例，以及 Hadoop 平台上海量数据的排序；第 4～7 章深入地讲解了 MapReduce 计算模型、MapReduce 应用的开发方法、MapReduce 的工作机制，同时还列出了多个 MapReduce 的应用案例，涉及单词计数、数据去重、排序、单表关联和多表关联等内容；第 8～11 章全面地阐述了 Hadoop 的 I/O 操作、HDFS 的原理与基本操作，以及 Hadoop 的各种管理操作，如集群的维护等；第 12～17 章详细而系统地讲解了 Hive、HBase、Mahout、Pig、ZooKeeper、Avro、Chukwa 等所有与 Hadoop 相关的子项目的原理及使用，以及这些子项目与 Hadoop 的整合使用；第 18 章以实例的方式讲解了常用 Hadoop 插件的使用和 Hadoop 插件的开发。

本书既适合没有 Hadoop 基础的初学者系统地学习，又适合有一定 Hadoop 基础但是缺乏实践经验的读者实践和参考。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目 (CIP) 数据

Hadoop 实战 / 陆嘉恒著. —北京：机械工业出版社，2011.9

(云计算技术系列丛书)

ISBN 978-7-111-35944-9

I. H… II. 陆… III. 数据处理—应用软件—网络编程 IV. TP274

中国版本图书馆 CIP 数据核字 (2011) 第 192844 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑：杨绣国 陈佳媛

北京京师印务有限公司印刷

2011 年 10 月第 1 版第 1 次印刷

186mm×240mm·28.75 印张

标准书号：ISBN 978-7-111-35944-9

定价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com



目 录

前 言

第 1 章 Hadoop 简介 /1

- 1.1 什么是 Hadoop/2
 - 1.1.1 Hadoop 概述 /2
 - 1.1.2 Hadoop 的历史 /2
 - 1.1.3 Hadoop 的功能与作用 /2
 - 1.1.4 Hadoop 的优势 /3
 - 1.1.5 Hadoop 的应用现状和发展趋势 /3
- 1.2 Hadoop 项目及其结构 /3
- 1.3 Hadoop 的体系结构 /6
 - 1.3.1 HDFS 的体系结构 /6
 - 1.3.2 MapReduce 的体系结构 /7

- 1.4 Hadoop 与分布式开发 /7
- 1.5 Hadoop 计算模型——MapReduce/10
- 1.6 Hadoop 的数据管理 /10
 - 1.6.1 HDFS 的数据管理 /11
 - 1.6.2 HBase 的数据管理 /12
 - 1.6.3 Hive 的数据管理 /15
- 1.7 小结 /17

第 2 章 Hadoop 的安装与配置 /18

- 2.1 在 Linux 上安装与配置 Hadoop/19
 - 2.1.1 安装 JDK 1.6/19
 - 2.1.2 配置 SSH 免密码登录 /20
 - 2.1.3 安装并运行 Hadoop/21
- 2.2 在 Windows 上安装与配置 Hadoop/23
 - 2.2.1 安装 Cygwin/24
 - 2.2.2 配置环境变量 /24
 - 2.2.3 安装和启动 sshd 服务 /24
 - 2.2.4 配置 SSH 免密码登录 /24
- 2.3 安装和配置 Hadoop 集群 /25
 - 2.3.1 网络拓扑 /25
 - 2.3.2 定义集群拓扑 /25
 - 2.3.3 建立和安装 Cluster /26
- 2.4 日志分析及几个小技巧 /32
- 2.5 小结 /33

第 3 章 Hadoop 应用案例分析 /35

- 3.1 Hadoop 在 Yahoo! 的应用 /36
- 3.2 Hadoop 在 eBay 的应用 /38
- 3.3 Hadoop 在百度的应用 /40
- 3.4 Hadoop 在 Facebook 的应用 /43
- 3.5 Hadoop 平台上的海量数据排序 /46
- 3.6 小结 /53

第 4 章 MapReduce 计算模型 /54

- 4.1 为什么要用 MapReduce/55

VIII

- 4.2 MapReduce 计算模型 /56
 - 4.2.1 MapReduce Job/56
 - 4.2.2 Hadoop 中的 Hello World 程序 /56
 - 4.2.3 MapReduce 的数据流和控制流 /64
- 4.3 MapReduce 任务的优化 /65
- 4.4 Hadoop 流 /67
 - 4.4.1 Hadoop 流的工作原理 /68
 - 4.4.2 Hadoop 流的命令 /69
 - 4.4.3 实战案例：添加 Bash 程序和 Python 程序到 Hadoop 流中 /70
- 4.5 Hadoop Pipes/72
- 4.6 小结 /74

第 5 章 开发 MapReduce 应用程序 /75

- 5.1 系统参数的配置 /76
- 5.2 配置开发环境 /78
- 5.3 编写 MapReduce 程序 /79
 - 5.3.1 Map 处理 /79
 - 5.3.2 Reduce 处理 /80
- 5.4 本地测试 /81
- 5.5 运行 MapReduce 程序 /83
 - 5.5.1 打包 /84
 - 5.5.2 在本地模式下运行 /85
 - 5.5.3 在集群上运行 /86
- 5.6 网络用户界面 /87
 - 5.6.1 JobTracker 页面 /87
 - 5.6.2 工作页面 /88
 - 5.6.3 返回结果 /90
 - 5.6.4 任务页面 /93
 - 5.6.5 任务细节页面 /93
- 5.7 性能调优 /94
- 5.8 MapReduce 工作流 /96
 - 5.8.1 将问题分解成 MapReduce 工作 /97
 - 5.8.2 运行相互依赖的工作 /97
- 5.9 小结 /98

第 6 章 MapReduce 应用案例 /99

- 6.1 单词计数 /100
 - 6.1.1 实例描述 /100
 - 6.1.2 设计思路 /100
 - 6.1.3 程序代码 /101
 - 6.1.4 代码解读 /102
 - 6.1.5 程序执行 /103
 - 6.1.6 代码结果 /103
- 6.2 数据去重 /104
 - 6.2.1 实例描述 /104
 - 6.2.2 设计思路 /105
 - 6.2.3 程序代码 /105
- 6.3 排序 /106
 - 6.3.1 实例描述 /106
 - 6.3.2 设计思路 /107
 - 6.3.3 程序代码 /107
- 6.4 单表关联 /109
 - 6.4.1 实例描述 /109
 - 6.4.2 设计思路 /110
 - 6.4.3 程序代码 /110
- 6.5 多表关联 /113
 - 6.5.1 实例描述 /113
 - 6.5.2 设计思路 /114
 - 6.5.3 程序代码 /114
- 6.6 小结 /116

第 7 章 MapReduce 工作机制 /117

- 7.1 MapReduce 作业的执行流程 /118
 - 7.1.1 MapReduce 任务的执行总流程 /118
 - 7.1.2 提交作业 /119
 - 7.1.3 初始化作业 /121
 - 7.1.4 分配任务 /123
 - 7.1.5 执行任务 /125
 - 7.1.6 更新任务执行进度和状态 /126
 - 7.1.7 完成作业 /127

- 7.2 错误处理机制 /127
 - 7.2.1 硬件故障 /127
 - 7.2.2 任务失败 /128
- 7.3 作业调度机制 /128
- 7.4 shuffle 和排序 /129
 - 7.4.1 map 端 /130
 - 7.4.2 reduce 端 /131
 - 7.4.3 shuffle 过程的优化 /132
- 7.5 任务执行 /133
 - 7.5.1 推测式执行 /133
 - 7.5.2 任务 JVM 重用 /134
 - 7.5.3 跳过坏记录 /134
 - 7.5.4 任务执行环境 /135
- 7.6 小结 /136

第 8 章 Hadoop I/O 操作 /137

- 8.1 I/O 操作中的数据检查 /138
- 8.2 数据的压缩 /142
 - 8.2.1 Hadoop 对压缩工具的选择 /142
 - 8.2.2 压缩分割和输入分割 /143
 - 8.2.3 在 MapReduce 程序中使用压缩 /143
- 8.3 数据的 I/O 中序列化操作 /144
 - 8.3.1 Writable 类 /144
 - 8.3.2 实现自己的 Hadoop 数据类型 /152
- 8.4 针对 MapReduce 的文件类 /153
 - 8.4.1 SequenceFile 类 /154
 - 8.4.2 MapFile 类 /159
- 8.5 小结 /161

第 9 章 HDFS 详解 /162

- 9.1 Hadoop 的文件系统 /163
- 9.2 HDFS 简介 /165
- 9.3 HDFS 体系结构 /166
 - 9.3.1 HDFS 的相关概念 /166
 - 9.3.2 HDFS 的体系结构 /167

- 9.4 HDFS 的基本操作 /169
 - 9.4.1 HDFS 的命令行操作 /169
 - 9.4.2 HDFS 的 Web 界面 /171
- 9.5 HDFS 常用 Java API 详解 /173
 - 9.5.1 使用 Hadoop URL 读取数据 /173
 - 9.5.2 使用 FileSystem API 读取数据 /174
 - 9.5.3 创建目录 /176
 - 9.5.4 写数据 /177
 - 9.5.5 删除数据 /178
 - 9.5.6 文件系统查询 /178
- 9.6 HDFS 中的读写数据流 /182
 - 9.6.1 文件的读取 /182
 - 9.6.2 文件的写入 /184
 - 9.6.3 一致性模型 /185
- 9.7 HDFS 命令详解 /186
 - 9.7.1 通过 distcp 进行并行复制 /186
 - 9.7.2 HDFS 的平衡 /187
 - 9.7.3 使用 Hadoop 归档文件 /188
 - 9.7.4 其他命令 /190
- 9.8 小结 /194

第 10 章 Hadoop 的管理 /195

- 10.1 HDFS 文件结构 /196
- 10.2 Hadoop 的状态监视和管理工具 /200
 - 10.2.1 审计日志 /200
 - 10.2.2 监控日志 /200
 - 10.2.3 Metrics/201
 - 10.2.4 Java 管理扩展 /203
 - 10.2.5 Ganglia/204
 - 10.2.6 Hadoop 管理命令 /206
- 10.3 Hadoop 集群的维护 /210
 - 10.3.1 安全模式 /210
 - 10.3.2 Hadoop 的备份 /211
 - 10.3.3 Hadoop 的节点管理 /212
 - 10.3.4 系统升级 /214

10.4 小结 /216

第 11 章 Hive 详解 /217

11.1 Hive 简介 /218

11.1.1 Hive 的数据存储 /218

11.1.2 Hive 的元数据存储 /220

11.2 Hive 的基本操作 /220

11.2.1 在集群上安装 Hive/220

11.2.2 配置 Hive/222

11.3 Hive QL 详解 /224

11.3.1 数据定义 (DDL) 操作 /224

11.3.2 数据操作 (DML) /231

11.3.3 SQL 操作 /233

11.3.4 Hive QL 的使用实例 /235

11.4 Hive 的网络 (WebUI) 接口 /237

11.5 Hive 的 JDBC 接口 /238

11.6 Hive 的优化 /241

11.7 小结 /243

第 12 章 HBase 详解 /244

12.1 HBase 简介 /245

12.2 HBase 的基本操作 /245

12.2.1 HBase 的安装 /245

12.2.2 运行 HBase /249

12.2.3 HBase Shell/250

12.2.4 HBase 配置 /254

12.3 HBase 体系结构 /255

12.4 HBase 数据模型 /259

12.4.1 数据模型 /259

12.4.2 概念视图 /260

12.4.3 物理视图 /260

12.5 HBase 与 RDBMS/261

12.6 HBase 与 HDFS/262

12.7 HBase 客户端 /262

12.8 Java API /263

- 12.9 HBase 编程实例之 MapReduce /270
- 12.10 模式设计 /273
 - 12.10.1 学生表 /273
 - 12.10.2 事件表 /274
- 12.11 小结 /275

第 13 章 Mahout 详解 /276

- 13.1 Mahout 简介 /277
- 13.2 Mahout 的安装和配置 /277
- 13.3 Mahout API 简介 /278
- 13.4 Mahout 中的聚类和分类 /280
 - 13.4.1 什么是聚类和分类 /280
 - 13.4.2 Mahout 中的数据表示 /281
 - 13.4.3 将文本转化成向量 /282
 - 13.4.4 Mahout 中的聚类、分类算法 /283
 - 13.4.5 算法应用实例 /288
- 13.5 Mahout 应用：建立一个推荐引擎 /292
 - 13.5.1 推荐引擎简介 /292
 - 13.5.2 使用 Taste 构建一个简单的推荐引擎 /292
 - 13.5.3 简单分布式系统下基于产品的推荐系统简介 /294
- 13.6 小结 /297

第 14 章 Pig 详解 /299

- 14.1 Pig 简介 /300
- 14.2 Pig 的安装和配置 /300
 - 14.2.1 Pig 的安装条件 /300
 - 14.2.2 Pig 的下载、安装和配置 /301
 - 14.2.3 Pig 运行模式 /301
- 14.3 Pig Latin 语言 /304
 - 14.3.1 Pig Latin 语言简介 /304
 - 14.3.2 Pig Latin 的使用 /305
 - 14.3.3 Pig Latin 的数据类型 /307
 - 14.3.4 Pig Latin 关键字 /308
- 14.4 用户定义函数 /313
 - 14.4.1 编写用户定义函数 /313

- 14.4.2 使用用户定义函数 /315
- 14.5 Pig 实例 /315
 - 14.5.1 Local 模式 /316
 - 14.5.2 MapReduce 模式 /318
- 14.6 Pig 进阶 /319
 - 14.6.1 数据实例 /319
 - 14.6.2 Pig 数据分析 /320
- 14.7 小结 /324

第 15 章 ZooKeeper 详解 /326

- 15.1 ZooKeeper 简介 /327
 - 15.1.1 ZooKeeper 的设计目标 /327
 - 15.1.2 数据模型和层次命名空间 /328
 - 15.1.3 ZooKeeper 中的节点和临时节点 /328
 - 15.1.4 ZooKeeper 的应用 /329
- 15.2 ZooKeeper 的安装和配置 /329
 - 15.2.1 在集群上安装 ZooKeeper /329
 - 15.2.2 配置 ZooKeeper /334
 - 15.2.3 运行 ZooKeeper /336
- 15.3 ZooKeeper 的简单操作 /339
 - 15.3.1 使用 ZooKeeper 命令的简单操作步骤 /339
 - 15.3.2 ZooKeeper API 的简单使用 /340
- 15.4 ZooKeeper 的特性 /343
 - 15.4.1 ZooKeeper 的数据模型 /343
 - 15.4.2 ZooKeeper 会话及状态 /345
 - 15.4.3 ZooKeeper Watches /346
 - 15.4.4 ZooKeeper ACL /346
 - 15.4.5 ZooKeeper 的一致性保证 /347
- 15.5 ZooKeeper 的 Leader 选举 /348
- 15.6 ZooKeeper 锁服务 /348
 - 15.6.1 ZooKeeper 中的锁机制 /349
 - 15.6.2 ZooKeeper 提供的一个写锁的实现 /350
- 15.7 使用 ZooKeeper 创建应用程序 /351
- 15.8 小结 /355

第 16 章 Avro 详解 /356

- 16.1 Avro 简介 /357
 - 16.1.1 模式声明 /358
 - 16.1.2 数据序列化 /362
 - 16.1.3 数据排列顺序 /364
 - 16.1.4 对象容器文件 /365
 - 16.1.5 协议声明 /367
 - 16.1.6 协议传输格式 /368
 - 16.1.7 模式解析 /370
- 16.2 Avro 的 C/C++ 实现 /371
- 16.3 Avro 的 Java 实现 /382
- 16.4 GenAvro (Avro IDL) 语言 /385
- 16.5 Avro SASL 概述 /390
- 16.6 小结 /392

第 17 章 Chukwa 详解 /393

- 17.1 Chukwa 简介 /394
- 17.2 Chukwa 架构 /395
 - 17.2.1 客户端 (Agent) 及其数据模型 /395
 - 17.2.2 收集器 (Collector) 和分离解析器 (Demux) /396
 - 17.2.3 HICC/398
- 17.3 Chukwa 的可靠性 /399
- 17.4 Chukwa 集群搭建 /400
 - 17.4.1 基本配置要求 /400
 - 17.4.2 安装 Chukwa/400
- 17.5 Chukwa 数据流的处理 /407
- 17.6 Chukwa 与其他监控系统比较 /408
- 17.7 小结 /409

第 18 章 Hadoop 的常用插件与开发 /411

- 18.1 Hadoop Studio 简介和使用 /412
 - 18.1.1 Hadoop Studio 的安装和配置 /412
 - 18.1.2 Hadoop Studio 的使用举例 /413
- 18.2 Hadoop Eclipse 简介和使用 /419

- 18.2.1 Hadoop Eclipse 安装和配置 /420
- 18.2.2 Hadoop Eclipse 的使用举例 /420
- 18.2.3 Hadoop Eclipse 插件开发 /421
- 18.3 Hadoop Streaming 简介和使用 /422
 - 18.3.1 Hadoop Streaming 的使用举例 /426
 - 18.3.2 使用 Hadoop Streaming 时常见的问题 /428
- 18.4 Hadoop Libhdfs 简介和使用 /430
 - 18.4.1 Hadoop Libhdfs 安装和配置 /430
 - 18.4.2 Hadoop Libhdfs API 简介 /430
 - 18.4.3 Hadoop Libhdfs 的使用举例 /431
- 18.5 小结 /432

附录 A 云计算在线检测平台 /434

- A.1 平台介绍 /435
- A.2 结构和功能 /435
 - A.2.1 前台用户接口的结构和功能 /435
 - A.2.2 后台程序运行的结构和功能 /437
- A.3 检测流程 /437
- A.4 使用 /438
 - A.4.1 功能使用 /438
 - A.4.2 返回结果介绍 /439
 - A.4.3 使用注意事项 /440
- A.5 小结 /441



第 1 章 Hadoop 简介

本章内容

- 什么是 Hadoop
- Hadoop 项目及其结构
- Hadoop 的体系结构
- Hadoop 与分布式开发
- Hadoop 计算模型——MapReduce
- Hadoop 的数据管理
- 小结

1.1 什么是 Hadoop

1.1.1 Hadoop 概述

Hadoop 是 Apache 软件基金会旗下的一个开源分布式计算平台。以 Hadoop 分布式文件系统（HDFS, Hadoop Distributed Filesystem）和 MapReduce（Google MapReduce 的开源实现）为核心的 Hadoop 为用户提供了系统底层细节透明的分布式基础架构。HDFS 的高容错性、高伸缩性等优点允许用户将 Hadoop 部署在低廉的硬件上，形成分布式系统；MapReduce 分布式编程模型允许用户在不了解分布式系统底层细节的情况下开发并行应用程序。所以用户可以利用 Hadoop 轻松地组织计算机资源，从而搭建自己的分布式计算平台，并且可以充分利用集群的计算和存储能力，完成海量数据的处理。

1.1.2 Hadoop 的历史

Hadoop 的源头是 Apache Nutch，该项目开始于 2002 年，是 Apache Lucene 的子项目之一。2004 年，Google 在“操作系统设计与实现”（OSDI, Operating System Design and Implementation）会议上公开发表了题为“MapReduce: Simplified Data Processing on Large Clusters”（MapReduce: 简化大规模集群上的数据处理）的论文，之后受到启发的 Doug Cutting 等人开始尝试实现 MapReduce 计算框架，并将它与 NDFS（Nutch Distributed File System）结合，以支持 Nutch 引擎的主要算法。由于 NDFS 和 MapReduce 在 Nutch 引擎中有着良好的应用，所以它们于 2006 年 2 月被分离出来，成为了一套完整而独立的软件，起名为 Hadoop。到了 2008 年年初，Hadoop 已成为 Apache 的顶级项目，它被包括 Yahoo! 在内的很多互联网公司所采用。现在，Hadoop 已经发展成为包含 HDFS、MapReduce、Pig、ZooKeeper 等子项目的集合，用于分布式计算。

1.1.3 Hadoop 的功能与作用

我们为什么需要 Hadoop 呢？众所周知，现代社会的信息量增长速度极快，这些信息里又积累着大量的数据，其中包括个人数据和工业数据。预计到 2020 年，每年产生的数字信息将会有超过 1/3 的内容驻留在云平台中或借助云平台处理。我们需要对这些数据进行分析和处理，以获取更多有价值的信息。那么我们如何高效地存储和管理这些数据，如何分析这些数据呢？这时可以选用 Hadoop 系统，它在处理这类问题时，采用了分布式存储方式，提高了读写速度，并扩大了存储容量。采用 MapReduce 来整合分布式文件系统上的数据，可保证分析和处理数据的高效。与此同时，Hadoop 还采用存储冗余数据的方式保证了数据的安全性。

Hadoop 中 HDFS 的高容错特性，以及它是基于 Java 语言开发的，这使得 Hadoop 可以部署在低廉的计算机集群中，同时不限于某个操作系统。Hadoop 中 HDFS 的数据管理能力，MapReduce 处理任务时的高效率，以及它的开源特性，使其在同类的分布式系统中大放异

彩，并在众多行业和科研领域中被广泛采用。

1.1.4 Hadoop 的优势

Hadoop 是一个能够让用户轻松架构和使用的分布式计算平台。用户可以轻松地在 Hadoop 上开发和运行处理海量数据的应用程序。它主要有以下几个优点：

- 高可靠性。Hadoop 按位存储和处理数据的能力值得人们信赖。
- 高扩展性。Hadoop 是在可用的计算机集簇间分配数据并完成计算任务的，这些集簇可以方便地扩展到数以千计的节点中。
- 高效性。Hadoop 能够在节点之间动态地移动数据，并保证各个节点的动态平衡，因此其处理速度非常快。
- 高容错性。Hadoop 能够自动保存数据的多个副本，并且能够自动将失败的任务重新分配。

1.1.5 Hadoop 的应用现状和发展趋势

由于 Hadoop 优势突出，基于 Hadoop 的应用已经遍地开花，尤其是在互联网领域。Yahoo! 通过集群运行 Hadoop，以支持广告系统和 Web 搜索的研究；Facebook 借助集群运行 Hadoop，以支持其数据分析和机器学习；百度则使用 Hadoop 进行搜索日志的分析和网页数据的挖掘工作；淘宝的 Hadoop 系统用于存储并处理电子商务交易的相关数据；中国移动研究院基于 Hadoop 的“大云”（BigCloud）系统用于对数据进行分析并对外提供服务。

2008 年 2 月，Hadoop 最大贡献者的 Yahoo! 构建了当时规模最大的 Hadoop 应用，它们在 2000 个节点上面执行了超过 1 万个 Hadoop 虚拟机器来处理超过 5PB 的网页内容，分析大约 1 兆个网络连接之间的网页索引资料。这些网页索引资料压缩后超过 300TB。Yahoo! 正是基于这些为用户提供了高质量的搜索服务。

Hadoop 目前已经取得了非常突出的成绩。随着互联网的发展，新的业务模式还将不断涌现，Hadoop 的应用也会从互联网领域向电信、电子商务、银行、生物制药等领域拓展。相信在未来，Hadoop 将会在更多的领域中扮演幕后英雄，为我们提供更加快捷优质的服务。

1.2 Hadoop 项目及其结构

现在 Hadoop 已经发展成为包含多个子项目的集合。虽然其核心内容是 MapReduce 和 Hadoop 分布式文件系统（HDFS），但 Hadoop 下的 Common、Avro、Chukwa、Hive、HBase 等子项目也是不可或缺的。它们提供了互补性服务或在核心层上提供了更高层的服务。图 1-1 展现了 Hadoop 的项目结构图。

下面将对 Hadoop 的各个子项目进行更详细的介绍。