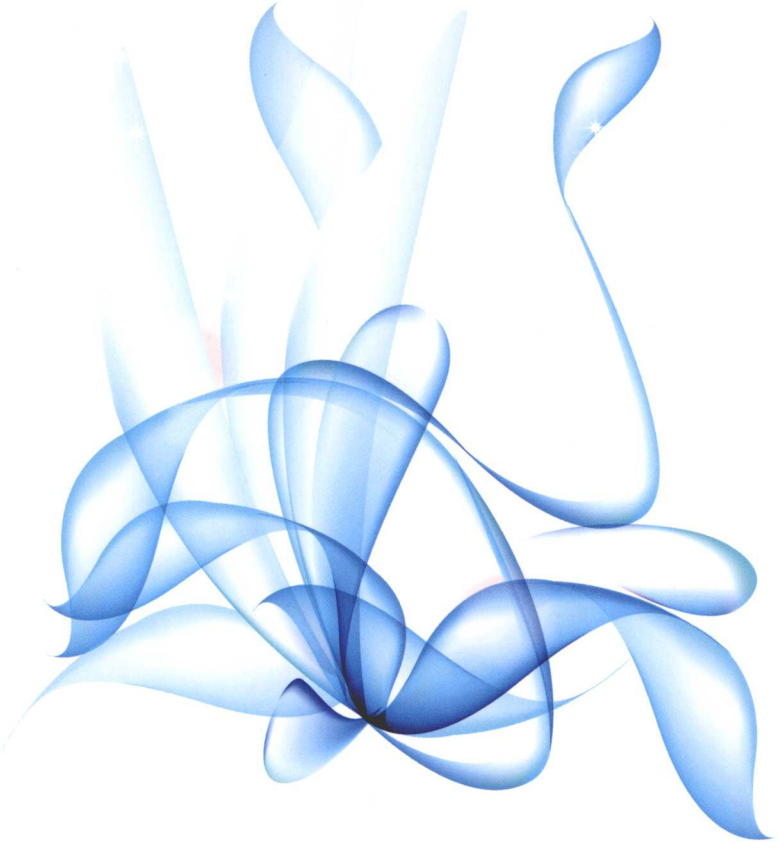
HZ BOOKS
华章科技

腾讯资深Hadoop技术专家撰写，EasyHadoop和51CTO等专业技术社区联袂推荐！

从源代码角度深入分析Common和HDFS的架构设计与实现原理，为Hadoop的优化、定制和扩展提供原理性指导。

从源代码中参透分布式技术精髓与分布式系统设计的优秀思想和方法。

大数据 技术丛书



Hadoop Internals: in-depth study of Common and HDFS

Hadoop技术内幕

深入解析Hadoop Common和HDFS架构设计与实现原理

蔡斌 陈湘萍◎著

机械工业出版社
China Machine Press

技术丛书

Hadoop Internals: in-depth study of Common and HDFS

Hadoop技术内幕

深入解析Hadoop Common和HDFS架构设计与实现原理

蔡斌 陈湘萍◎著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Hadoop 技术内幕：深入解析 Hadoop Common 和 HDFS 架构设计与实现原理 / 蔡斌，陈湘萍著 . —北京：机械工业出版社，2013.3

ISBN 978-7-111-41766-8

I. H… II. ①蔡… ②陈… III. ①数据处理软件 ②分布式文件系统 IV. ① TP274 ② TP316

中国版本图书馆 CIP 数据核字（2013）第 047263 号

“Hadoop 技术内幕”共两册，分别从源代码的角度对“Common+HDFS”和 MapReduce 的架构设计与实现原理进行了极为详细的分析。本书由腾讯数据平台的资深 Hadoop 专家、X-RIME 的作者亲自执笔，对 Common 和 HDFS 的源代码进行了分析，旨在为 Hadoop 的优化、定制和扩展提供原理性的指导。除此之外，本书还从源代码实现中对分布式技术的精髓、分布式系统设计的优秀思想和方法，以及 Java 语言的编码技巧、编程规范和对设计模式的精妙运用进行了总结和分析，对提高读者的分布式技术能力和 Java 编程能力都非常有帮助。本书适合 Hadoop 的二次开发人员、应用开发工程师、运维工程师阅读。

全书共 9 章，分为三部分：第一部分（第 1 章）主要介绍了 Hadoop 源代码的获取和源代码阅读环境的搭建；第二部分（第 2 ~ 5 章）对 Hadoop 公共工具 Common 的架构设计和实现原理进行了深入分析，包含 Hadoop 的配置信息处理、面向海量数据处理的序列化和压缩机制、Hadoop 的远程过程调用，以及满足 Hadoop 上各类应用访问数据的 Hadoop 抽象文件系统和部分具体文件系统等内容；第三部分（第 6 ~ 9 章）对 Hadoop 的分布式文件系统 HDFS 的架构设计和实现原理进行了详细的分析，这部分内容采用了总分总的结构，第 6 章对 HDFS 的各个实体和实体间接口进行了分析；第 7 章和第 8 章分别详细地研究了数据节点和名字节点的实现原理，并通过第 9 章对客户端的解析，回顾了 HDFS 各节点间的配合，完整地介绍了一个大规模数据存储系统的实现。

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：朱秀英

北京市荣盛彩色印刷有限公司印刷

2013 年 4 月第 1 版第 1 次印刷

186mm × 240 mm · 32.75 印张

标准书号：ISBN 978-7-111-41766-8

定 价：89.00 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 投稿热线：（010）88379604

购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

前　言

为什么写本书

互联网使得信息的采集、传播速度和规模达到空前的水平，实现了全球的信息共享与交互，它已经成为信息社会必不可少的基础设施，同时也带来了多方面的新挑战。2003 年，Google 发表了《Google File System》，介绍了 Google 海量数据处理使用的文件系统，使互联网时代的数据存储发生了革命性的变化。而 Doug Cutting 等人在 Nutch 项目上应用 GFS 和 MapReduce 思想，并演化为 Hadoop 项目，经过多年的发展，最终形成了包含多个相关项目的软件生态系统，开创了海量数据处理的新局面。

Hadoop 正是为了解决互联网时代的海量数据存储和处理而设计、开发的。简单地讲，Hadoop 是一个可以更容易开发和并行处理大规模数据的分布式计算平台，它的主要特点是：扩展能力强、成本低、高效率、可靠。目前，Hadoop 的用户已经从传统的互联网公司，扩展到科学计算、电信行业、电力行业、生物行业以及金融公司，并得到越来越广泛的应用。

Hadoop 作为一个优秀的开源项目，提供了一些文档和所有的源代码，但是，对于很多开发人员，仅仅通过一些简单的例子或教程学习使用 Hadoop 的基本功能是远远不够的。同时，随着云计算和大数据的发展，产业界正在经历一次重大变革，特别是基于云计算的海量数据处理，改变着我们思考的方式和习惯，开发者们越来越有必要去了解 Hadoop 的架构与设计原理。

本书从源代码的层面上对 Hadoop 的公共工具 Common 和 Hadoop 的分布式文件系统 HDFS 进行了介绍，帮助广大开发者从架构与设计原理的角度去理解 Hadoop，从而为更好

地使用和扩展 Hadoop 打下坚实的基础。同时，Hadoop 是一个使用 Java 语言实现的优秀系统，从事 Java 和分布式计算相关技术的开发者们能从它的源码实现中看到许多优秀的设计思想、对各种设计模式的灵活运用、语言的使用技巧以及编程规范等。这些都有助于加深开发者们对 Java 相关技术，尤其是 Hadoop 的理解，从而提高自己的开发水平，拓展自己的技术视野，为工作带来帮助。

读者对象

❑Hadoop 开发人员

对这部分读者来说，本书的内容能够帮助他们加深对 Hadoop 的理解，通过全面了解 Hadoop，特别是 HDFS 的实现原理，为进一步优化、定制和扩展 Hadoop 提供坚实基础。

❑学习分布式技术的读者

Hadoop 是一个得到广泛应用的大型分布式系统，开放的源代码中包含了大量分布式系统设计原理和实现，读者可以通过本书，充分学习、体验和实践分布式技术。

❑学习 Java 语言的中高级读者

Hadoop 使用 Java 语言实现，它充分利用了 Java 的语言特性，并使用了大量的标准库和开源工具，很多功能的设计和实现非常优秀，是极佳的学习 Java 技术的参考资料。

本书的主要内容

本书主要分为三个部分。

第一部分（第 1 章）对如何建立 Hadoop 的开发、分析环境做了简单的介绍。对于 Hadoop 这样复杂、庞大的项目，一个好的开发环境可以让读者事半功倍地学习、研究源代码。

第二部分（第 2~5 章）主要对 Hadoop 公共工具 Common 的实现进行研究。分别介绍了 Hadoop 的配置系统、面向海量数据处理的序列化和压缩机制、Hadoop 使用的远程过程调用，以及满足 Hadoop 上各类应用访问数据的 Hadoop 抽象文件系统和部分具体文件系统。

第三部分（第 6~9 章）对 Hadoop 分布式文件系统进行了详细的分析。这部分内容采用总 - 分 - 总的结构，第 6 章介绍了 HDFS 各个实体和实体间接口，第 7 章和第 8 章分别详细地研究了数据节点和名字节点的实现原理，第 9 章通过对客户端的解析，回顾 HDFS 各节点间的配合，完整地介绍了一个大规模数据存储系统的实现。

通过本书，读者不仅能全面了解 Hadoop 的优秀架构和设计思想，而且还能从 Hadoop，特别是 HDFS 的实现源码中一窥 Java 开发的精髓和分布式系统的精要。

勘误和支持

由于作者的水平有限，编写时间跨度较长，同时开源软件的演化较快，书中难免会出现

一些错误或者不准确的地方，恳请读者批评指正。如果大家有和本书相关的内容需要探讨，或有更多的宝贵意见，欢迎通过 caibinbupt@qq.com 和我们联系，希望能结识更多的朋友，大家共同进步。书中的源代码文件可以从华章网站[⊖]下载。

致谢

感谢机械工业出版社华章公司的编辑杨福川和白宇，杨老师的耐心和支持让本书最终得以出版，白老师的很多建议使本书的可读性更强。

感谢腾讯数据平台部的张文郁、赵重庆和徐钊，作为本书的第一批读者和 Hadoop 专家，他们的反馈意见让本书增色不少。

感谢和我们一起工作、研究和应用 Hadoop 的腾讯数据平台部，以及 IBM 中国研究中心和中山大学的领导和同事们，本书的很多内容是对实际项目的总结。

最后，作者向支持本书写作的家人深表谢意，感谢他们的耐心和理解。

⊖ 参见华章网站 www.hzbook.com——编辑注

目　录

前　言

第一部分　环境准备

第 1 章　源代码环境准备 / 2

1.1　什么是 Hadoop / 2

1.1.1　Hadoop 简史 / 2

1.1.2　Hadoop 的优势 / 3

1.1.3　Hadoop 生态系统 / 4

1.2　准备源代码阅读环境 / 8

1.2.1　安装与配置 JDK / 8

1.2.2　安装 Eclipse / 9

1.2.3　安装辅助工具 Ant/ 12

1.2.4　安装类 UNIX Shell 环境 Cygwin / 13

1.3　准备 Hadoop 源代码 / 15

1.3.1　下载 Hadoop / 15

1.3.2　创建 Eclipse 项目 / 16

1.3.3 Hadoop 源代码组织 / 18
1.4 小结 / 19

第二部分 Common 的实现

第 2 章 Hadoop 配置信息处理 / 22
2.1 配置文件简介 / 22
2.1.1 Windows 操作系统的配置文件 / 22
2.1.2 Java 配置文件 / 23
2.2 Hadoop Configuration 详解 / 24
2.2.1 Hadoop 配置文件的格式 / 24
2.2.2 Configuration 的成员变量 / 26
2.2.3 资源加载 / 27
2.2.4 使用 get* 和 set* 访问 / 设置配置项 / 32
2.3 Configurable 接口 / 34
2.4 小结 / 35

第 3 章 序列化与压缩 / 36
3.1 序列化 / 36
3.1.1 Java 内建序列化机制 / 36
3.1.2 Hadoop 序列化机制 / 38
3.1.3 Hadoop 序列化机制的特征 / 39
3.1.4 Hadoop Writable 机制 / 39
3.1.5 典型的 Writable 类详解 / 41
3.1.6 Hadoop 序列化框架 / 48
3.2 压缩 / 49
3.2.1 Hadoop 压缩简介 / 50
3.2.2 Hadoop 压缩 API 应用实例 / 51
3.2.3 Hadoop 压缩框架 / 52
3.2.4 Java 本地方法 / 61
3.2.5 支持 Snappy 压缩 / 65
3.3 小结 / 69

第 4 章　Hadoop 远程过程调用 / 70

4.1　远程过程调用基础知识 / 70

4.1.1　RPC 原理 / 70

4.1.2　RPC 机制的实现 / 72

4.1.3　Java 远程方法调用 / 73

4.2　Java 动态代理 / 78

4.2.1　创建代理接口 / 78

4.2.2　调用转发 / 80

4.2.3　动态代理实例 / 81

4.3　Java NIO/ 84

4.3.1　Java 基本套接字 / 84

4.3.2　Java NIO 基础 / 86

4.3.3　Java NIO 实例：回显服务器 / 93

4.4　Hadoop 中的远程过程调用 / 96

4.4.1　利用 Hadoop IPC 构建简单的分布式系统 / 96

4.4.2　Hadoop IPC 的代码结构 / 100

4.5　Hadoop IPC 连接相关过程 / 104

4.5.1　IPC 连接成员变量 / 104

4.5.2　建立 IPC 连接 / 106

4.5.3　数据分帧和读写 / 111

4.5.4　维护 IPC 连接 / 114

4.5.5　关闭 IPC 连接 / 116

4.6　Hadoop IPC 方法调用相关过程 / 118

4.6.1　Java 接口与接口体 / 119

4.6.2　IPC 方法调用成员变量 / 121

4.6.3　客户端方法调用过程 / 123

4.6.4　服务器端方法调用过程 / 126

4.7　Hadoop IPC 上的其他辅助过程 / 135

4.7.1　RPC.getProxy() 和 RPC.stopProxy() / 136

4.7.2　RPC.getServer() 和 Server 的启停 / 138

4.8　小结 / 141

第 5 章　Hadoop 文件系统 / 142

5.1　文件系统 / 142

5.1.1 文件系统的用户界面 / 142
5.1.2 文件系统的实现 / 145
5.1.3 文件系统的保护控制 / 147
5.2 Linux 文件系统 / 150
5.2.1 Linux 本地文件系统 / 150
5.2.2 虚拟文件系统 / 153
5.2.3 Linux 文件保护机制 / 154
5.2.4 Linux 文件系统 API/ 155
5.3 分布式文件系统 / 159
5.3.1 分布式文件系统的特性 / 159
5.3.2 基本 NFS 体系结构 / 160
5.3.3 NFS 支持的文件操作 / 160
5.4 Java 文件系统 / 162
5.4.1 Java 文件系统 API / 162
5.4.2 URI 和 URL / 164
5.4.3 Java 输入 / 输出流 / 166
5.4.4 随机存取文件 / 169
5.5 Hadoop 抽象文件系统 / 170
5.5.1 Hadoop 文件系统 API / 170
5.5.2 Hadoop 输入 / 输出流 / 175
5.5.3 Hadoop 文件系统中的权限 / 179
5.5.4 抽象文件系统中的静态方法 / 180
5.5.5 Hadoop 文件系统中的协议处理器 / 184
5.6 Hadoop 具体文件系统 / 188
5.6.1 FileSystem 层次结构 / 189
5.6.2 RawLocalFileSystem 的实现 / 191
5.6.3 ChecksumFileSystem 的实现 / 196
5.6.4 RawInMemoryFileSystem 的实现 / 210
5.7 小结 / 213

第三部分 Hadoop 分布式文件系统

第 6 章 HDFS 概述 / 216

6.1 初识 HDFS / 216

6.1.1 HDFS 主要特性 / 216
6.1.2 HDFS 体系结构 / 217
6.1.3 HDFS 源代码结构 / 221
6.2 基于远程过程调用的接口 / 223
6.2.1 与客户端相关的接口 / 224
6.2.2 HDFS 各服务器间的接口 / 236
6.3 非远程过程调用接口 / 244
6.3.1 数据节点上的非 IPC 接口 / 245
6.3.2 名字节点和第二名字节点上的非 IPC 接口 / 252
6.4 HDFS 主要流程 / 254
6.4.1 客户端到名字节点的文件与目录操作 / 254
6.4.2 客户端读文件 / 256
6.4.3 客户端写文件 / 257
6.4.4 数据节点的启动和心跳 / 258
6.4.5 第二名字节点合并元数据 / 259
6.5 小结 / 261

第 7 章 数据节点实现 / 263

7.1 数据块存储 / 263
7.1.1 数据节点的磁盘目录文件结构 / 263
7.1.2 数据节点存储的实现 / 266
7.1.3 数据节点升级 / 269
7.1.4 文件系统数据集的工作机制 / 276
7.2 流式接口的实现 / 285
7.2.1 DataXceiverServer 和 DataXceiver / 286
7.2.2 读数据 / 289
7.2.3 写数据 / 298
7.2.4 数据块替换、数据块拷贝和读数据块检验信息 / 313
7.3 作为整体的数据节点 / 314
7.3.1 数据节点和名字节点的交互 / 314
7.3.2 数据块扫描器 / 319
7.3.3 数据节点的启停 / 321
7.4 小结 / 326

第 8 章　名字节点实现 / 327

8.1　文件系统的目录树 / 327

8.1.1　从 i-node 到 INode/ 327

8.1.2　命名空间镜像和编辑日志 / 333

8.1.3　第二名字节点 / 351

8.1.4　FSDirectory 的实现 / 361

8.2　数据块和数据节点管理 / 365

8.2.1　数据结构 / 366

8.2.2　数据节点管理 / 378

8.2.3　数据块管理 / 392

8.3　远程接口 ClientProtocol 的实现 / 412

8.3.1　文件和目录相关事务 / 412

8.3.2　读数据使用的方法 / 415

8.3.3　写数据使用的方法 / 419

8.3.4　工具 dfsadmin 依赖的方法 / 443

8.4　名字节点的启动和停止 / 444

8.4.1　安全模式 / 444

8.4.2　名字节点的启动 / 449

8.4.3　名字节点的停止 / 454

8.5　小结 / 454

第 9 章　HDFS 客户端 / 455

9.1　认识 DFSClient / 455

9.1.1　DFSClient 的构造和关闭 / 455

9.1.2　文件和目录、系统管理相关事务 / 457

9.1.3　删除 HDFS 文件 / 目录的流程 / 459

9.2　输入流 / 461

9.2.1　读数据前的准备：打开文件 / 463

9.2.2　读数据 / 465

9.2.3　关闭输入流 / 475

9.2.4　读取 HDFS 文件数据的流程 / 475

9.3　输出流 / 478

9.3.1　写数据前的准备：创建文件 / 481

9.3.2　写数据：数据流管道的建立 / 482

9.3.3 写数据：数据包的发送 / 486
9.3.4 写数据：数据流管道出错处理 / 493
9.3.5 写数据：租约更新 / 496
9.3.6 写数据：DFSOutputStream.sync() 的作用 / 497
9.3.7 关闭输出流 / 499
9.3.8 向 HDFS 文件写入数据的流程 / 500
9.4 DistributedFileSystem 的实现 / 506
9.5 HDFS 常用工具 / 508
9.5.1 FsShell / 508
9.5.2 DFSAdmin / 510
9.6 小结 / 511

第一部分

环境准备

本部分内容

- 源代码环境准备

第 1 章　源代码环境准备

数据！数据！数据！

今天，我们正被数据包围。全球 43 亿部电话、20 亿位互联网用户每秒都在不断地产生大量数据，人们发送短信给朋友、上传视频、用手机拍照、更新社交网站的信息、转发微博、点击广告等，使得机器产生和保留了越来越多的数据。数据的指数级增长对处于市场领导地位的互联网公司，如 Facebook、谷歌、雅虎、亚马逊、腾讯等提出了挑战。它们需要对 TB 级别和 PB 级别的数据进行分析处理，以发现哪些网站更受欢迎，哪些商品更具有吸引力，哪些广告更吸引用户。传统的工具对于处理如此规模的数据集越来越无能为力。

现在，Hadoop 应运而生，庞大的信息流有了新的处理平台。

1.1　什么是 Hadoop

Hadoop 是 Apache 基金会下的一个开源分布式计算平台，以 Hadoop 分布式文件系统（Hadoop Distributed File System，HDFS）和 MapReduce 分布式计算框架为核心，为用户提供了底层细节透明的分布式基础设施。HDFS 的高容错性、高伸缩性等优点，允许用户将 Hadoop 部署在廉价的硬件上，构建分布式系统；MapReduce 分布式计算计算框架则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、分布的应用程序，充分利用大规模的计算资源，解决传统高性能单机无法解决的大数据处理问题。

Apache Hadoop 是目前分析海量数据的首选工具。

1.1.1　Hadoop 简史

谈到 Hadoop 的历史，就不得不提到 Lucene 和 Nutch。Hadoop 开始时是 Nutch 的一个子项目，而 Nutch 又是 Apache Lucene 的子项目。这 3 个项目都是由 Doug Cutting 创立，每个项目在逻辑上都是前一个项目的演进。

Lucene 是引擎开发工具包，提供了一个纯 Java 的高性能全文索引，它可以方便地嵌入各种实际应用中实现全文搜索 / 索引功能。Nutch 项目开始于 2002 年，是以 Lucene 为基础实现的搜索引擎应用。Lucene 为 Nutch 提供了文本搜索和索引的 API，Nutch 不光有搜索功能，还有数据抓取的功能。

但很快，Doug Cutting 和 Mike Calarella（Hadoop 和 Nutch 的另一位创始人）就意识到，他们的架构无法扩展以支持拥有数十亿网页的网络。这个时候，Google 的研究人员在 2003 年的 ACM SOSP（Symposium on Operating Systems Principles）会议上发表的描述 Google 分布式文件系统（简称 GFS）的论文及时地为他们提供了帮助。GFS 或类似的系统可以解决他

们在网络抓取和索引过程中产生的大量文件存储需求。于是，在 2004 年，他们开始写 GFS 的一个开源实现，即 Nutch 分布式文件系统（NDFS）。

2004 年，在 OSDI（Operating Systems Design and Implementation）会议上，Google 发表了论文，向全世界介绍了 MapReduce。2005 年初，Nutch 的开发者在 Nutch 上有了一个可工作的 MapReduce 应用，到当年的年中，所有主要的 Nutch 算法被迁移到 MapReduce 和 NDFS 上。

在 Nutch0.8.0 版本之前，Hadoop 还属于 Nutch 的一部分，而从 Nutch0.8.0 开始，Doug Cutting 等人将其中实现的 NDFS 和 MapReduce 剥离出来成立了一个新的开源项目，这就是 Hadoop。同时，对比以前的 Nutch 版本，Nutch0.8.0 在架构上有了根本性的变化，它完全构建在 Hadoop 的基础之上了。这个时候，已经是 2006 年 2 月，大约在同一时间，Doug Cutting 加入雅虎，Yahoo 投入了专门的团队和资源将 Hadoop 发展成一个可在网络上运行的系统。

值得一提的是 Hadoop 名字的来源。

为软件项目命名时，Doug Cutting 似乎总会得到家人的启发。Lucene 是他妻子的中间名，也是她外祖母的名字。他的儿子在咿呀学语时，总把所有用于吃饭的词叫成 Nutch。Doug Cutting 如此解释 Hadoop 的得名："这是我的孩子给一头吃饱了的棕黄色大象起的名字。我的命名标准就是简短，容易发音和拼写，没有太多的意义，并且不会被用于别处。小孩子是这方面的高手，Googol 就是由小孩命名的。"

2008 年 1 月，Hadoop 已成为 Apache 顶级项目，证明它是成功的。通过这次机会，Hadoop 成功地被雅虎之外的很多公司应用，如 Facebook、纽约时报等。特别是纽约时报，它使用运行在亚马逊的 EC2 云计算上的 Hadoop，将 4TB 的报纸扫描文档压缩，转换为用于 Web 的 PDF 文档，这个过程历时不到 24 小时，使用 100 台机器运行，这成为 Hadoop 一个良好的宣传范例。

2008 年 2 月，雅虎宣布其索引网页的生产系统采用了在 10 000 多个核的 Linux 集群上运行的 Hadoop。Hadoop 真正达到了万维网的规模。2008 年 4 月，在一个 900 节点的 Hadoop 集群上，雅虎的研究人员运行 1TB 的 Jim Gray 基准排序，只用了 209 秒，而到了 2009 年 4 月，在一个 1400 节点的集群上对 500GB 数据进行排序，只用了 59 秒，这显示了 Hadoop 强大的计算能力。

2008 年开始，Hadoop 迈向主流，开始了它的爆发式发展，出现了大量的相关项目，如 2008 年的 HBase、ZooKeeper 和 Mahout，2009 年的 Pig、Hive 等。同时，还出现了像 Cloudera（成立于 2008 年）和 Hortonworks（以雅虎的 Hadoop 业务部门为基础成立的公司）这样的专注于 Hadoop 的公司。

经过多年的发展，Hadoop 已经从初出茅庐的小象变身为行业巨人。

1.1.2　Hadoop 的优势

将 Hadoop 运用于海量数据处理，主要有如下几个优势：

❑方便：Hadoop 可以运行在一般商业机器构成的大型集群上，或者是亚马逊弹性计算云（Amazon EC2）等云计算服务上。

❑弹性：Hadoop 通过增加集群节点，可以线性地扩展以处理更大的数据集。同时，在集群负载下降时，也可以减少节点，以高效使用计算资源。

❑健壮：Hadoop 在设计之初，就将故障检测和自动恢复作为一个设计目标，它可以从容处理通用计算平台上出现的硬件失效的情况。

❑简单：Hadoop 允许用户快速编写出高效的并行分布代码。

由于 Hadoop 具有上述优势，使得 Hadoop 在学术界和工业界都大受欢迎。今天，Hadoop 已经成为许多公司和大学基础计算平台的一部分。学术界如内布拉斯加大学通过使用 Hadoop，支持紧凑型 μ 子螺旋形磁谱仪实验数据的保存和计算；加州大学伯克利分校则对 Hadoop 进行研究，以提高其整体性能；在国内，中国科学院计算技术研究所在 Hadoop 上开展了数据挖掘和地理信息处理等的研究。在工业界，Hadoop 已经成为很多互联网公司基础计算平台的一个核心部分，如雅虎、Facebook、腾讯等；传统行业，如传媒、电信、金融，也在使用这个系统，进行数据存储与处理。

如今，Hadoop 分布式计算基础架构这把“大伞”下，已经包含了多个子项目。而海量数据处理也迅速成为许多程序员需要掌握的一项重要技能。

1.1.3 Hadoop 生态系统

经过几年的快速发展，Hadoop 现在已经发展成为包含多个相关项目的软件生态系统。狭义的 Hadoop 核心只包括 Hadoop Common、Hadoop HDFS 和 Hadoop MapReduce 三个子项目，但和 Hadoop 核心密切相关的，还包括 Avro、ZooKeeper、Hive、Pig 和 HBase 等项目，构建在这些项目之上的，面向具体领域、应用的 Mahout、X-Rime、Crossbow 和 Ivory 等项目，以及 Chukwa、Flume、Sqoop、Oozie 和 Karmasphere 等数据交换、工作流和开发环境这样的外围支撑系统。它们提供了互补性的服务，共同提供了一个海量数据处理的软件生态系统，Hadoop 生态系统如图 1-1 所示。

图 1-1 Hadoop 生态系统