HZ BOOKS
华章IT

Apress®

世界级大数据专家亲笔撰写，深入剖析Hadoop系统实现原理及其工程实践应用，包含大量真实案例和数据

从Hadoop使用者角度深入浅出地讲解Hadoop各组件的运行机理，深入到系统实现源码，帮助你从架构、开发、应用和运维等多方面全面掌握Hadoop系统

大数据技术丛书

Pro Apache Hadoop

Second Edition

深入理解Hadoop

（原书第2版）

[美] 萨米尔·瓦德卡（Sameer Wadkar）
马杜·西德林埃（Madhu Siddalingaiah） 著
杰森·文纳（Jason Venner）

于博 冯傲风 译

机械工业出版社
China Machine Press

技术丛书

Pro Apache Hadoop

Second Edition

深入理解Hadoop

（原书第2版）

[美] 萨米尔·瓦德卡（Sameer Wadkar）
马杜·西德林埃（Madhu Siddalingaiah） 著
杰森·文纳（Jason Venner）

于博 冯傲风 译

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

深入理解 Hadoop（原书第 2 版）/（美）瓦德卡（Wadkar, S.），（美）西德林埃（Siddalingaiah, M.），（美）文纳（Venner, J.）著；于博，冯傲风译 . —北京：机械工业出版社，2015.10
（大数据技术丛书）
书名原文：Pro Apache Hadoop, Second Edition

ISBN 978-7-111-51565-4

I. 深… II. ①瓦… ②西… ③文… ④于… ⑤冯… III. 数据处理软件 IV. TP274

中国版本图书馆 CIP 数据核字（2015）第 227389 号

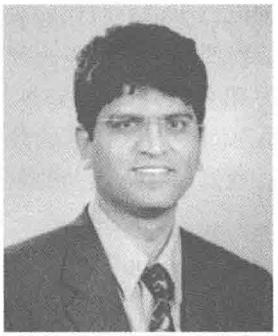
本书版权登记号：图字：01-2015-4154

深入理解 Hadoop（原书第 2 版）

出版发行：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：吴 怡　　责任校对：殷 虹
印　刷：北京诚信伟业印刷有限公司　　版　次：2016 年 1 月第 1 版第 1 次印刷
开　本：186mm × 240mm 1/16　　印　张：25
书　号：ISBN 978-7-111-51565-4　　定　价：79.00 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）88379426 88361066　　投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259　　读者信箱：hzit@hzbook.com

作者简介 *About the Author*



萨米尔·瓦德卡（Sameer Wadkar）在软件架构与开发方面有多达16年的工作经验。他在孟买大学取得了电子工程方向的硕士学位，及金融专业的MBA学位，在National Center of Software Technology（现为Center for Development of Advanced Computing）取得了软件工程专业的硕士学位，在约翰霍普金斯大学获得了应用与计算数学专业的博士学位。他还实现了分布式系统及高访问量的Web网站，从政府机构到银行投资公司多种领域都有客户使用这个Web网站。他从2011年就积极地参与到了Hadoop/HBase的开发实现当中，也是开源软件的贡献者。他的GitHub的主页是：https://github.com/sameerwadkar。

他在开源领域的贡献包括Latent Dirichlett Allocation文本挖掘算法，该算法能够在单台机器上处理数百万级别的文档。他还是一位活跃的国际象棋玩家，经常在网站www.chessclub.com上下国际象棋。

马杜·西德林埃（Madhu Siddalingaiah）是一名拥有25年工作经验的技术顾问，其工作经验涉及多个领域，包括航空航天、医疗保险、金融、能源、科学研究等。多年以来，他专门从事电子工程、互联网技术、大数据等领域。最近，他发布了几个著名的大数据系统及其解决方案。

他在马里兰大学取得了物理学的硕士学位。专业之外，Madhu还是一名私人直升飞机驾驶员，他喜欢旅行，不断参与越来越多的爱好活动。

杰森·文纳（Jason Venner）有20多年的软件开发经验，涉及软件工程、数据挖掘、架构设计等领域，曾在IT企业中担任过副总裁、主管、顾问，近些年来他的钻研方向是Java、Hadoop、云计算等领域。

The Translator's Words 译 者 序

我们正处在一个数据量爆炸增长的时代，也是一个越来越依靠数据进行决策的时代，对海量数据的挖掘分析也从抽样数据分析变成了全量数据分析。为此，业界涌现出了许多新技术，Apache Hadoop是其中当之无愧的王者。当前Hadoop已然发展为拥有HDFS、YARN、MapReduce、Pig、HCatalog、HBase等系统的较为完整的大数据系统生态圈，为大数据的开发利用提供了比较全面的解决方案。毫无疑问，Hadoop已成为大数据行业发展背后强劲的驱动力。

本书详细地讲述了Hadoop生态圈中最为重要的几个组件。不仅介绍了Hadoop涉及的分布式理论基础知识，还着重讲解Hadoop系统的工程实践应用。为了深入浅出地讲述Hadoop各个组件的运行机理，作者使用了贴切的实战用例贯穿全书，同时，又深入到系统实现源码，使读者做到知其然，更知其所以然。通过对本书的学习，读者可以从架构、开发、应用和运维等多方面全面地掌握Hadoop系统，从而成为名副其实的Hadoop专家。

我曾经拜读过本书英文版的第1版，机缘巧合，华章给了我翻译该书第2版的机会，感谢我的小伙伴百度公司的冯傲风同学与我并肩奋战，完成了本书的翻译；感谢吴怡编辑对本书认真负责的审校和中肯的修改建议，使得本书能够顺利地出版，她的认真和敬业令人钦佩！

最后还要感谢家人，只有他们的理解和支持，我才能每天抽出那么多时间来完成本书的翻译工作。

书中概念和术语颇多，有一些目前尚无固定中文译法，加上译者水平有限，译文中的不当之处在所难免，我们真诚地希望同行和读者们不吝赐教。

于博

2015年9月于北京

技术评审人简介

威米什·奥姆·米塔尔（Vimlesh Om Mittal）具有多达 14 年的技术实现经验，是 Apache Hadoop CDH4 的 Cloudera 认证开发者（Cloudera Certified Developer）。他具有非常广博的技术背景，他的经验主要包括关于现代商业处理的相关项目、客户应用开发、ETL 开发、数字应用开发、商业智能、数据库设计和数据转换策略。

他拥有构建健壮的信息管理系统、解决方案架构设计和基于多种开发框架进行软件开发的综合经验。这些开发框架包括 Hadoop 生态系统中的大数据框架、数据获取、ETL、报告 / 分析、.NET、J2EE 和不同技术供应商提供的脚本语言，这些技术供应商包括 Cloudera、SAS、IBM、Oracle、Microsfot、iOS 以及 BEA。

Vimlesh 还专注于金融服务数据的解决方案、移动应用设计及开发。他对软件开发的整个生命周期中的各个方面都十分熟悉，在项目管理科学方面有宝贵经验。

前　言 Preface

Hadoop 已经进入 Apache 社区发展五年多了，使用 Hadoop 系统进行开发的工作仍然富于挑战但收获丰厚。本书第 1 版在若干年前就已经出版了，在这期间，Hadoop 系统已经被越来越多的企业使用，自身也得到了飞速发展。Hadoop2.0 基于 YARN 框架做了全新升级，重写了 Hadoop 系统的底层平台。本书从 Hadoop 使用者的角度出发讲解 Hadoop 的实现原理，浓缩了 Hadoop 软件系统的精华。作为作者，我们希望可以深入到源代码级别来理解 Hadoop 的运行原理及其背后的设计目标，渴望与你分享 Hadoop 带给我们的启迪。通过本书，你不仅能够深入学习 Hadoop 系统，还可以对 Java 语言在大数据处理过程中的运用有更深入的认识。

本书总体上介绍了大数据的范畴概念，然后对 Hadoop 进行了详细讲解。因为如果对大数据没有一个总体上的认知，是不可能真正理解 Hadoop 系统的。本书的读者对象具备中级 Java 开发能力的 Hadoop 开发人员。本书可以让你成为企业中的 Hadoop 专家。通过本书的学习，你可以获得大量实用技巧，这些技巧都是我们在从事基于 Hadoop 的分布式数据处理系统实践过程中总结出来的。

本书循序渐进地讲解了大量的实例，会帮助你从 Hadoop 集群初级使用者成长为能胜任运行任何复杂应用程序的专家。下面是本书的一个内容纲要：

- 第 1 章，讲解大数据系统的作用，介绍各种各样的大数据系统。
- 第 2 章，高屋建瓴地介绍 Hadoop2.0 和 YARN，讲解 Hadoop 平台的关键概念。
- 第 3 章，开始 Hadoop 学习，构建你的第一个 MapReduce 程序。
- 第 4 章，讲解 Hadoop 平台管理的关键概念要素。
- 第 5 章、第 6 章和第 7 章是本书的重点，深入分析讲解了 MapReduce 框架。你将会学习 MapReduce 框架所有知识点。我们以当前广泛使用的语言 SQL 作为对照，来学习理解 MapReduce 框架的使用。利用 MapReduce 框架来实现 SQL 语言中 SELECT、WHERE、GROUP BY 和 JOIN 这些关键字的功能。Hadoop 系统一个常用用途就是来做数据 ETL。这几章会让你掌握如何使用 MapReduce 框架支持通用的数据处理功能。我们不仅仅学习 MapReduce 框架的 API，还会学习 MapReduce 框架中更复杂的概念

及其设计理念。

- 第 8 章，介绍一种测试框架，这种测试框架支持 MapReduce 程序的单元测试和集成测试。
- 第 9 章，讲解 Hadoop 框架的监控和日志。
- 第 10 章，讲解 Hive 框架，这是一个基于 MapReduce 的数据仓库框架。
- 第 11 章，讲解 Pig 和 Crunch 框架。这些框架可以帮助使用者在 Hadoop 平台上构建数据处理管道。
- 第 12 章，讲解 HCatalog 框架。该框架帮助企业用户可以像访问数据库中的数据表一样，来访问存放在 Hadoop 文件系统上的海量数据。
- 第 13 章，讲解如何使用 Hadoop 来分析处理日志流。
- 第 14 章，介绍 HBase，这是一个基于 Hadoop 系统的 NoSQL 数据库。你会学习许多 HBase 用法范例。
- 第 15 章，对数据科学做简要介绍。介绍了 MapReduce 框架在数学科学上的局限和不足之处。同时引入介绍新的框架，比如 Spark、Hama，这些框架突破了 MapReduce 框架中的某些局限。
- 第 16 章，简要介绍了 Hadoop 系统在云计算中的应用。讲解如何在一个真实的生产环境中的 Hadoop 集群上工作。
- 第 17 章，粗略地介绍 Hadoop2.0 的一个关键附加功能：在 Hadoop 上开发一个自己的类似于 MapReduce 框架的分布式数据处理框架。讲解了如何基于 Hadoop2.0 开发一个简单的分布式下载服务。

致谢

Hadoop 在过去的 10 年发展过程中，有许多厂商和独立开发者为之贡献代码。Hadoop 系统中也使用了大量的开源函数库。我们要感谢为 Hadoop 系统贡献源码的所有开源作者所付出的辛勤劳动。我们也要感谢那些在论坛上回答了 Hadoop 系统相关问题的博客作者及专家学者。他们的积极参与，使得 Hadoop 这样复杂的系统变得易于使用并成为应用主流。

我们还想感谢 Apress 的全体员工，他们的努力付出使得本书内容清晰易读。

我们最后要感谢我们的家庭、朋友及同事，本书在编写过程中得到他们的一贯支持。

目　录 *Contents*

第 1 章 Chapter 1

为什么会有大数据

随着近 20 多年来计算技术的不断革新，企业积累了大量数据。数字传感器的进步使得通信系统越来越广泛的使用，尤其是移动平台和移动终端的飞速增长；系统运行产生的大量日志以及越来越多的企业采用无纸化办公的工作方式，这些情况都使得企业积攒起了海量数据资源。并且随着人们对现代科技越来越多的依赖，数据将会以更快的速度增长下去。

摩尔定律告诉我们，大约每隔两年，计算机的性能将历史性地提升一倍。最初，计算资源的提升速度满足了飞速增长的数据的处理需求。好景不长，到 2005 年前后，数据处理需求增长的速度已经快于计算资源处理能力的提升速度。

计算机工业界想到了另外一个更加经济有效的解决办法，就是数据的并行处理。既然单台计算机无法满足大数据量的计算需求，那就用多台计算机来并行处理这些海量数据。Hadoop 就是利用互相联网的多台计算机使用 MapReduce（一种改进的单指令多数据流 [SIMD] 计算技术）来并行地处理计算大量数据。

像亚马逊、谷歌、微软这样的公司为我们提供了较为廉价的云计算服务，使得分布式并行计算的概念更加深入人心，相比于购买计算资源，我们只要花少量的费用就可以租用相应的计算资源。

本书是使用 Hadoop 平台来开发和运行软件的实用指南。Hadoop 项目起初由 Apache Software Foundation 来负责管理，目前像 Cloudera、MapR 和 Hortonworks 这样的公司也加入到维护发展它的阵营中。本章会从整体上介绍大数据的由来以及 Hadoop 项目。

1.1 什么是大数据

就本书而言，大数据（Big Data）姑且定义为无法被符合服务等级协议（service level agreement，

SLA）的单台计算机处理或（有些时候）存储的任何数据集。后续内容会有更精确的解释。理论上讲，单台计算机可以处理任意规模的数据。对于超过单台计算机存储量的海量数据，可以存放到类似网络附属存储（network attached storage，NAS）这样的共享存储设备中，然后输入到单台计算机去计算处理。但是，这样处理数据所花费的时间往往会大大超过允许的数据处理时间。

举个简单的例子。某个业务单元每次平均需要处理 200GB 的数据，数据的读取速率是每秒钟 50MB。如果按照每秒钟 50MB 的数据读取速率来计算，顺序读取硬盘上的 100MB 数据，需要两秒钟，读取全部的 200GB 数据就大约需要一个小时了。然而这 200GB 的数据要求在 5 分钟之内处理完毕。如果这个业务处理的 200GB 数据可以被平均分布到 100 个计算节点，每个节点处理分配的数据（举一个简单的用例，假设用 SALES_YEAR>2001 这样一条简单的准则来划分数据集），去掉这 100 个计算节点的 CPU 处理时间和数据结果整合的时间，整个数据处理过程会在 1 分钟之内完成。

这个简单的例子说明，大数据是一个相对的概念，是相对于我们的业务需求而言的。

注意 Jeff Dean 博士有一篇关于并行计算的论文，详见 http://www.cs.cornell.edu/projects/ladis2009/talks/dean-keynote-ladis2009pdf。从本地磁盘顺序读取 1MB 数据需要 2 千万纳秒。从速度为 1 Gbps 的网络中读取 1MB 的数据需要 2 亿 5 千万纳秒（假定传输 2KB 的数据需要 25 万纳秒，还要返回传输确认，这样每传输 2KB 数据需要网络往返共 50 万纳秒的时间）。虽然文中的描述稍显过时，数据的传输读取绝对速度已经有了提高，但是这不妨碍我们在本章中拿来做比照，因为这些数据之间的相对值变化不大。

1.2 大数据技术背后的核心思想

上文中的例子我们作了诸多假设，要表明的核心问题是虽然我们可以很快地处理数据，但是从持久性的存储设备中读取的速度受到限制，这是整个数据处理流程上的关键瓶颈所在。相对于读写本地节点存储设备上的数据，通过网络来传输数据会更慢。

下面列出了所有大数据处理方法中的一些共同特征：

- 数据分布在多个节点（网络 I/O 速度 << 本地磁盘 I/O 速度）。
- 计算程序离数据更近（集群上的节点），而不是相反。
- 数据的处理尽量在本地完成（网络 I/O 速度 << 本地磁盘 I/O 速度）。
- 使用可顺序读取磁盘 I/O 代替随机读取磁盘 I/O（数据交换速度 << 数据寻道时间）。

所有大数据计算处理模式都有一个目的，就是使输入 / 输出（I/O）并行化，从而提高数据处理性能。

1.2.1 把数据分发到多个节点

根据定义，大数据是无法仅靠单台计算机资源来处理的。大数据的一个特点就是使用商

用服务器。一台典型的商用服务器拥有 2TB 至 4TB 的磁盘。因为大数据是指远远超出这个容量的数据集，所以还是要将数据分布在多个节点上。

需要注意的是，我们把要处理的数据分布存放在多个计算节点，并不是因为这些数据的数据量有数十 T 之巨。你会发现大数据处理系统会有条不紊地在各个计算节点上计算处理所分配的数据。把数据分布到各个服务器节点的最终目的是为了让大量计算节点同时参与到数据的处理计算过程中来。哪怕仅有 500GB 的数据集，这么大的数据量，集群中的单台计算节点完全可以存储，也会把数据分发到多台计算节点。这样的数据分发有两点好处：

- 每个数据块会在多个节点上有多份拷贝（Hadoop 默认是一个数据块有 3 份拷贝），这使得系统具备容错性，当一个节点发生故障，其他节点还备份有故障节点上的数据。
- 为了达到数据并行处理的目的，多个节点可以同时参与数据处理过程。比如，50GB 的数据被分配到 10 台计算节点，每个计算节点仅处理分配给自己的数据集，可以达到数据处理性能 5 ～ 10 倍的提升。读者可能会问，为什么不把这些数据存放到网络文件系统（NFS）中，这样每个节点可以去读取它要处理的部分。答案是本地存储磁盘的数据读取速度要远远高于通过网络来读取数据的速度。当数据分析任务启动的时候，分析任务程序运行所需的函数库被拷贝到各个计算节点，大数据处理系统会尽量在本地处理计算数据。我们会在后面章节中详谈。

1.2.2　把计算逻辑移动到数据附近

对于我们这些精通 J2EE 编程的人来说，三层架构思想深植脑海。在三层编程模型中，所有的数据会通过网络集中到一起，交由应用层来处理。我们由此形成了固有的观念，就是数据应该是分散的，而程序应该是集中的。

大数据系统无法处理网络过载的问题。传输动辄数 T 的数据量给应用层，使得网络带宽耗尽，网络拥挤不堪，导致传输效率大幅下降，甚至有可能导致系统故障。从大数据的观念来看，应该把数据分布存放到各个计算节点，程序也要移动到数据附近。要做到这一点，是一件很不容易的事情。除了程序要移动到存放数据的节点，程序运行所依赖的函数库也要移动到数据处理节点才行。如果大数据系统的集群拥有数百个计算节点，显然那将是程序维护 / 部署人员的噩梦。所以，大数据系统可以让我们集中式地部署程序代码，大数据系统后台会在计算任务启动之前把这些程序移动到各个数据处理节点。

1.2.3　计算节点进行本地数据处理

前文提到的大数据系统的两个特点，决定了分配到计算节点的数据要在计算节点本地处理。所有的大数据编程模型都是基于分布式和并行处理的。网络 I/O 比本地磁盘 I/O 慢了好几个数量级。数据被分发到各个计算节点，程序运行依赖库也移动到了数据所在节点，计算节点就地计算处理数据的条件完备了。

虽然典型的大数据处理系统都希望把数据处理过程放在拥有数据的节点本地完成，但并