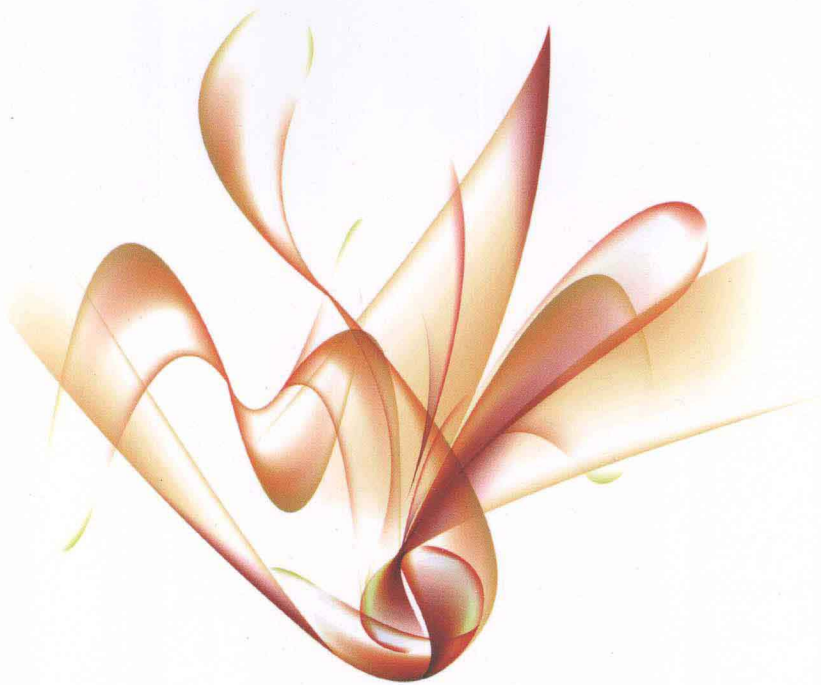




资深Hadoop技术专家撰写，从开发者角度系统深入地讲解了Hadoop分布式文件系统、Hadoop文件I/O、Hive、HBase、Mahout，以及MapReduce的工作原理、编程方法和高级应用
内容细致，包含大量基于实际生产环境的案例，实战性强



技术丛书



Programming Hadoop

Hadoop应用开发 技术详解

刘刚◎著



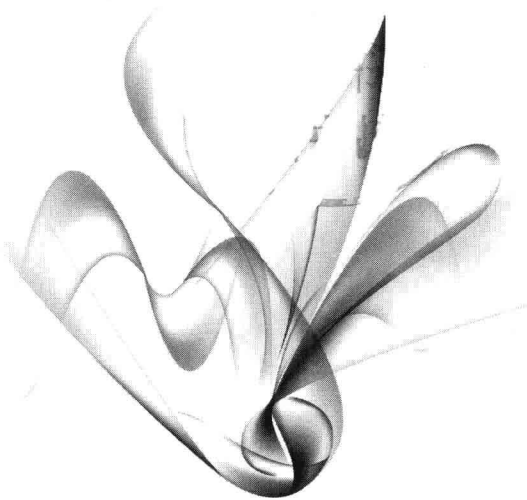
机械工业出版社
China Machine Press

技术丛书

Programming Hadoop

Hadoop应用开发 技术详解

刘 刚◎著



机械工业出版社

图书在版编目 (CIP) 数据

Hadoop 应用开发技术详解 / 刘刚著. — 北京: 机械工业出版社, 2014.1

(大数据技术丛书)

ISBN 978-7-111-45244-7

I. H… II. 刘… III. 数据处理软件—程序设计 IV. TP274

中国版本图书馆 CIP 数据核字 (2013) 第 309446 号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书由资深 Hadoop 技术专家撰写, 系统、全面、深入地讲解了 Hadoop 开发者需要掌握的技术和知识, 包括 HDFS 的原理和应用、Hadoop 文件 I/O 的原理和应用、MapReduce 的原理和高级应用、MapReduce 的编程方法和技巧, 以及 Hive、HBase 和 Mahout 等技术和工具的使用。并且提供大量基于实际生产环境的案例, 实战性非常强。

全书一共 12 章。第 1 ~ 2 章详细地介绍了 Hadoop 的生态系统、关键技术以及安装和配置; 第 3 章是 MapReduce 的使用入门, 让读者了解整个开发过程; 第 4 ~ 5 章详细讲解了分布式文件系统 HDFS 和 Hadoop 的文件 I/O; 第 6 章分析了 MapReduce 的工作原理; 第 7 章讲解了如何利用 Eclipse 来编译 Hadoop 的源代码, 以及如何对 Hadoop 应用进行测试和调试; 第 8 ~ 9 章细致地讲解了 MapReduce 的开发方法和高级应用; 第 10 ~ 12 章系统地讲解了 Hive、HBase 和 Mahout。

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: 白 宇

三河市杨庄长鸣印刷装订厂印刷

2014 年 1 月第 1 版第 1 次印刷

186mm × 240 mm · 26.25 印张

标准书号: ISBN 978-7-111-45244-7

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) 88378991 88361066

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 88379604

读者信箱: hzsj@hzbook.com

前 言

为什么要写这本书

三年前接触 Hadoop 的时候，Hadoop 在国内还没有普及，学习的资料也是少之又少，只能看英文文档一步一步操作，出现错误只能自己解决或者查看英文资料，有时候一个问题好几天才解决，学习起来非常的麻烦。

古时候，人们用牛来拉重物，当一头牛拉不动一根圆木的时候，他们不曾想过培育更强壮的牛。同样，我们也不需要尝试更大的计算机，而是应该开发更多的计算系统。简单说，数据量越来越大，我们的大型机负担不了，因此集群就产生了，而 Hadoop 就是为分布式计算和分布式存储而诞生的。最近两年，随着互联网的快速发展，公司的日志量成数量级增长，从而导致使用 Hadoop 的公司也越来越多，Hadoop 人才也就越来越供不应求，尤其是有丰富工作经验的人才。

在开源的 Hadoop 社区里，很多人问我怎么学习 Hadoop，学习 Hadoop 需要什么样的基础，如何才能成为一个 Hadoop 的技术大牛。为了帮助更多的人，我决定写本书，将我对 Hadoop 的理解和实战经验与大家分享。希望帮助读者在学习 Hadoop 的道路上少走弯路，避免一些不必要的时间付出。

读者对象

本书适合以下读者阅读。

- ☐ 大数据爱好者
- ☐ Hadoop 入门者
- ☐ 海量数据处理的从业人员
- ☐ 有 Java 基础，想学习 Hadoop 的开发者
- ☐ Hive、HBase 和 Mahout 的爱好者
- ☐ 具有 1 ~ 3 年使用 Hadoop 经验，想进一步提高能力的使用者
- ☐ 开设相关课程的大专院校

如何阅读本书

本书共 12 章。

第1章是Hadoop概述，互联网时代什么最重要——数据！大数据时代什么工具最流行——Hadoop！本章主要从大体上介绍Hadoop，让读者对Hadoop有个大体的印象。

第2章主要是对Hadoop安装进行一步一步地详解，包括Hadoop的单机模式、伪分布式模式以及分布式模式，让读者了解Hadoop的安装过程。

第3章是MapReduce快速入门，通过一个实例介绍MapReduce开发的整个过程。包括环境的准备、Mapper和Reducer代码的编写、打包、部署和运行的一整套流程。这套流程对以后MapReduce的开发奠定了基础。此外，介绍了MapReduce开发遇到的常见错误，以及解决方法。

第4章本章详细讲解了Hadoop分布式文件系统（HDFS），让读者对HDFS有一个全面的了解。HDFS是Hadoop里面的一个核心组件，任何基于Hadoop应用的工具都要用到HDFS，学习本章对Hadoop系统的维护有很大的帮助。

第5章本章主要讲Hadoop I/O 底层的知识，让读者了解Hadoop底层的一些原理，比如数据在Hadoop上是怎么传输的。学完这章对Hadoop的输入和输出、序列化和反序列化、Writable类型等有进一步的理解。

第6章主要介绍MapReduce的工作原理，讲解了JobTracker、TaskTracker、任务的调度、任务工作的原理以及MapReduce的Shuffle和Sort机制等，此外还介绍第二代MapReduce（YARN）。通过本章内容的学习，将会对日后MapReduce编程开发奠定基础。

第7章主要介绍三部分内容，Eclipse的MapReduce插件安装、利用插件调试MapReduce代码，以及MapReduce单元测试。目的是提高读者编写MapReduce的调试、测试的效率。选择和使用一个好的工具是项目成功的一半。

第8章详细讲解了MapReduce编程开发用到的知识点，如Combiner、Partitioner、DistributedCache等。全面介绍了Mapper和Reducer两个抽象类，最后介绍了如何使用Hadoop Streaming接口开发MapReduce程序。

第9章介绍了MapReduce的一些高级应用、算法等知识点，算是对第8章学的知识进行应用，本章内容是MapReduce开发和Hadoop系统调优常用的知识点，希望读者能够掌握，为MapReduce的算法开发和调优打下基础。

第10章对Hive的知识点和使用做了介绍。Hive是基于Hadoop的一个数据仓库，支持类似SQL语句来替代MapReduce，这使不会写代码的数据分析师也能利用Hadoop平台对海量数据进行分析，如DBA。Hive工具现在已经非常成熟，使用的人也越来越多。

第11章介绍了HBase的安装和Shell操作、MapReduce操作HBase等。HBase是一个基于列存储的NoSQL数据库，不同于一般的NoSQL数据库，HBase是一个适合于非结构化数据存储的数据库。

第12章介绍了Mahout算法框架的安装和使用。本章介绍的常用知识是入门的最佳实战。Mahout是非常优秀的基于MapReduce的算法框架。Mahout工具在商品推荐、电影推荐、广告推荐等领域应用非常广泛。

附录A为Hive内置操作符与函数。附录B为HBase默认配置解释。附录C为Hadoop

三个配置文件的参数含义说明。

其中第 8 章和第 9 章侧重于 Java 开发 MapReduce 的用户，如果你是一名有经验的 Hadoop 用户，并且想学习 MapReduce 开发，可以直接阅读这两章的内容。但是如果你是一名初学者，请一定从第 1 章的基础理论知识开始学习。

勘误和支持

除封面署名外，参加本书编写工作的还有：向磊、艾男、曾泽、湖畔、郭洁、童小军、王虎、李晋博、林斌、闫越等。由于作者的水平有限，编写时间仓促，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。书中的全部源文件可以从华章网站^①下载，我也会定期将相应的功能更新及时更正出来。可以将书中的错误和遇到的任何问题，发邮件到 jaylg2010@gmail.com。如果你有更多的宝贵意见，也欢迎发送邮件，期待能够得到你们的真挚反馈。

致谢

首先要感谢 Doug Cutting，他开创了一款影响整个互联网领域的开源平台——Hadoop。

感谢机械工业出版社华章公司的杨福川老师，在这一年多的时间始终支持我的写作，你的鼓励和帮助引导我顺利完成全部书稿。感谢编辑白宇对稿件的认真修改，你提出的一些写书注意事项，让我受益匪浅。

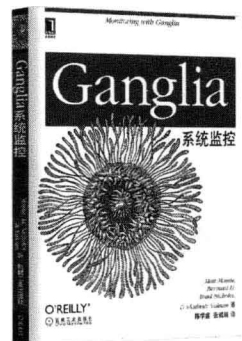
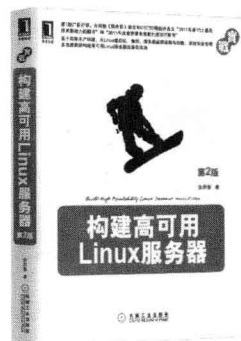
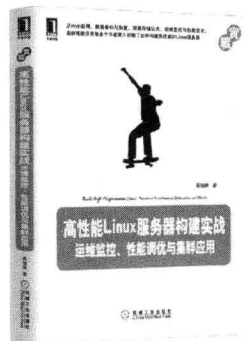
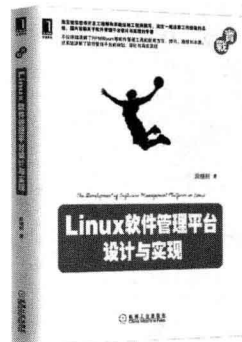
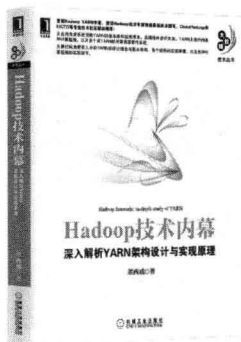
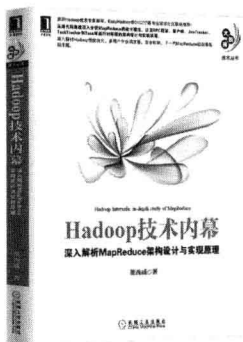
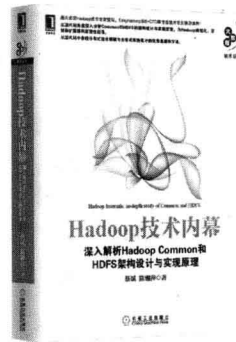
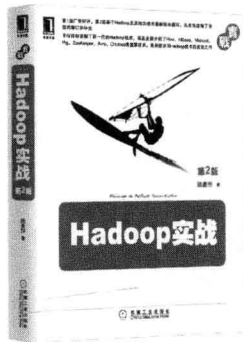
感谢湖畔老师对我的支持和鼓励。

最后感谢我的爸爸、妈妈，感谢你们将我培养成人，并时时刻刻为我灌输着信心和力量！

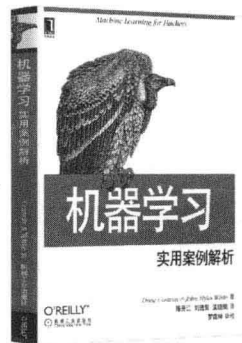
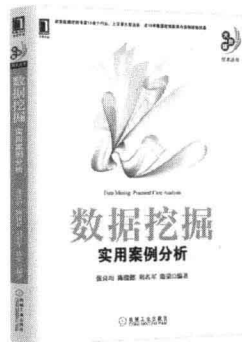
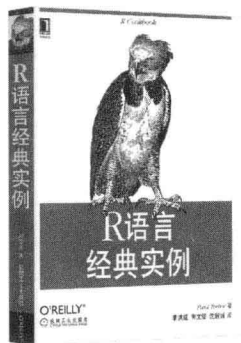
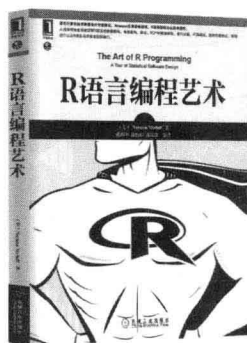
谨以此书献给我最亲爱的家人，以及众多热爱 Hadoop 的朋友们！

① 参见华章网站 www.hzbook.com。——编辑注

推荐阅读



推荐阅读



目 录

前 言

第 1 章 Hadoop 概述 / 1

- 1.1 Hadoop 起源 / 1
 - 1.1.1 Google 与 Hadoop 模块 / 1
 - 1.1.2 为什么会有 Hadoop / 1
 - 1.1.3 Hadoop 版本介绍 / 2
- 1.2 Hadoop 生态系统 / 3
- 1.3 Hadoop 常用项目介绍 / 4
- 1.4 Hadoop 在国内的应用 / 6
- 1.5 本章小结 / 7

第 2 章 Hadoop 安装 / 8

- 2.1 Hadoop 环境安装配置 / 8
 - 2.1.1 安装 VMware / 8
 - 2.1.2 安装 Ubuntu / 8
 - 2.1.3 安装 VMware Tools / 15
 - 2.1.4 安装 JDK / 15
- 2.2 Hadoop 安装模式 / 16
 - 2.2.1 单机安装 / 17
 - 2.2.2 伪分布式安装 / 18
 - 2.2.3 分布式安装 / 20
- 2.3 如何使用 Hadoop / 27
 - 2.3.1 Hadoop 的启动与停止 / 27
 - 2.3.2 Hadoop 配置文件 / 28
- 2.4 本章小结 / 28

第 3 章 MapReduce 快速入门 / 30

- 3.1 WordCount 实例准备开发环境 / 30

- 3.1.1 使用 Eclipse 创建一个 Java 工程 / 30
- 3.1.2 导入 Hadoop 的 JAR 文件 / 31
- 3.2 MapReduce 代码的实现 / 32
 - 3.2.1 编写 WordMapper 类 / 32
 - 3.2.2 编写 WordReducer 类 / 33
 - 3.2.3 编写 WordMain 驱动类 / 34
- 3.3 打包、部署和运行 / 35
 - 3.3.1 打包成 JAR 文件 / 35
 - 3.3.2 部署和运行 / 36
 - 3.3.3 测试结果 / 38
- 3.4 本章小结 / 39

第 4 章 Hadoop 分布式文件系统 详解 / 40

- 4.1 认识 HDFS / 40
 - 4.1.1 HDFS 的特点 / 40
 - 4.1.2 Hadoop 文件系统的接口 / 45
 - 4.1.3 HDFS 的 Web 服务 / 46
- 4.2 HDFS 架构 / 46
 - 4.2.1 机架 / 47
 - 4.2.2 数据块 / 47
 - 4.2.3 元数据节点 / 48
 - 4.2.4 数据节点 / 50
 - 4.2.5 辅助元数据节点 / 50
 - 4.2.6 名字空间 / 52
 - 4.2.7 数据复制 / 53
 - 4.2.8 块备份原理 / 53
 - 4.2.9 机架感知 / 54

- 4.3 Hadoop 的 RPC 机制 / 55
 - 4.3.1 RPC 的实现流程 / 56
 - 4.3.2 RPC 的实体模型 / 56
 - 4.3.3 文件的读取 / 57
 - 4.3.4 文件的写入 / 58
 - 4.3.5 文件的一致模型 / 59
 - 4.4 HDFS 的 HA 机制 / 59
 - 4.4.1 HA 集群 / 59
 - 4.4.2 HA 架构 / 60
 - 4.4.3 为什么会有 HA 机制 / 61
 - 4.5 HDFS 的 Federation 机制 / 62
 - 4.5.1 单个 NameNode 的 HDFS 架构的局限性 / 62
 - 4.5.2 为什么引入 Federation 机制 / 63
 - 4.5.3 Federation 架构 / 64
 - 4.5.4 多个名字空间的管理问题 / 65
 - 4.6 Hadoop 文件系统的访问 / 66
 - 4.6.1 安全模式 / 66
 - 4.6.2 HDFS 的 Shell 访问 / 67
 - 4.6.3 HDFS 处理文件的命令 / 67
 - 4.7 Java API 接口 / 72
 - 4.7.1 Hadoop URL 读取数据 / 73
 - 4.7.2 FileSystem 类 / 73
 - 4.7.3 FileStatus 类 / 75
 - 4.7.4 FSDataInputStream 类 / 77
 - 4.7.5 FSDataOutputStream 类 / 81
 - 4.7.6 列出 HDFS 下所有的文件 / 83
 - 4.7.7 文件的匹配 / 84
 - 4.7.8 PathFilter 对象 / 84
 - 4.8 维护 HDFS / 86
 - 4.8.1 追加数据 / 86
 - 4.8.2 并行复制 / 88
 - 4.8.3 升级与回滚 / 88
 - 4.8.4 添加节点 / 90
 - 4.8.5 删除节点 / 91
 - 4.9 HDFS 权限管理 / 92
 - 4.9.1 用户身份 / 92
 - 4.9.2 权限管理的原理 / 93
 - 4.9.3 设置权限的 Shell 命令 / 93
 - 4.9.4 超级用户 / 93
 - 4.9.5 HDFS 权限配置参数 / 94
 - 4.10 本章小结 / 94
- 第 5 章 Hadoop 文件 I/O 详解 / 95**
- 5.1 Hadoop 文件的数据结构 / 95
 - 5.1.1 SequenceFile 存储 / 95
 - 5.1.2 MapFile 存储 / 99
 - 5.1.3 SequenceFile 转换为 MapFile / 101
 - 5.2 HDFS 数据完整性 / 103
 - 5.2.1 校验和 / 103
 - 5.2.2 数据块检测程序 / 104
 - 5.3 文件序列化 / 106
 - 5.3.1 进程间通信对序列化的要求 / 106
 - 5.3.2 Hadoop 文件的序列化 / 107
 - 5.3.3 Writable 接口 / 107
 - 5.3.4 WritableComparable 接口 / 108
 - 5.3.5 自定义 Writable 接口 / 109
 - 5.3.6 序列化框架 / 113
 - 5.3.7 数据序列化系统 Avro / 114
 - 5.4 Hadoop 的 Writable 类型 / 115
 - 5.4.1 Writable 类的层次结构 / 115
 - 5.4.2 Text 类型 / 116
 - 5.4.3 NullWritable 类型 / 117
 - 5.4.4 ObjectWritable 类型 / 117
 - 5.4.5 GenericWritable 类型 / 117
 - 5.5 文件压缩 / 117
 - 5.5.1 Hadoop 支持的压缩格式 / 118

- 5.5.2 Hadoop 中的编码器和
解码器 / 118
- 5.5.3 本地库 / 121
- 5.5.4 可分割压缩 LZO / 122
- 5.5.5 压缩文件性能比较 / 122
- 5.5.6 Snappy 压缩 / 124
- 5.5.7 gzip、LZO 和 Snappy 比较 / 124
- 5.6 本章小结 / 125

第 6 章 MapReduce 工作原理 / 126

- 6.1 MapReduce 的函数式编程概念 / 126
 - 6.1.1 列表处理 / 126
 - 6.1.2 Mapping 数据列表 / 127
 - 6.1.3 Reducing 数据列表 / 127
 - 6.1.4 Mapper 和 Reducer 如何
工作 / 128
 - 6.1.5 应用实例：词频统计 / 129
- 6.2 MapReduce 框架结构 / 129
 - 6.2.1 MapReduce 模型 / 130
 - 6.2.2 MapReduce 框架组成 / 130
- 6.3 MapReduce 运行原理 / 132
 - 6.3.1 作业的提交 / 132
 - 6.3.2 作业初始化 / 134
 - 6.3.3 任务的分配 / 136
 - 6.3.4 任务的执行 / 136
 - 6.3.5 进度和状态的更新 / 136
 - 6.3.6 MapReduce 的进度组成 / 137
 - 6.3.7 任务完成 / 137
- 6.4 MapReduce 容错 / 137
 - 6.4.1 任务失败 / 138
 - 6.4.2 TaskTracker 失败 / 138
 - 6.4.3 JobTracker 失败 / 138
 - 6.4.4 子任务失败 / 138
 - 6.4.5 任务失败反复次数的处理
方法 / 139

- 6.5 Shuffle 阶段和 Sort 阶段 / 139
 - 6.5.1 Map 端的 Shuffle / 140
 - 6.5.2 Reduce 端的 Shuffle / 142
 - 6.5.3 Shuffle 过程参数调优 / 143
- 6.6 任务的执行 / 144
 - 6.6.1 推测执行 / 144
 - 6.6.2 任务 JVM 重用 / 145
 - 6.6.3 跳过坏的记录 / 145
 - 6.6.4 任务执行的环境 / 146
- 6.7 作业调度器 / 146
 - 6.7.1 先进先出调度器 / 146
 - 6.7.2 容量调度器 / 146
 - 6.7.3 公平调度器 / 149
- 6.8 自定义 Hadoop 调度器 / 153
 - 6.8.1 Hadoop 调度器框架 / 153
 - 6.8.2 编写 Hadoop 调度器 / 155
- 6.9 YARN 介绍 / 157
 - 6.9.1 异步编程模型 / 157
 - 6.9.2 YARN 支持的计算框架 / 158
 - 6.9.3 YARN 架构 / 158
 - 6.9.4 YARN 工作流程 / 159
- 6.10 本章小结 / 160

第 7 章 Eclipse 插件的应用 / 161

- 7.1 编译 Hadoop 源码 / 161
 - 7.1.1 下载 Hadoop 源码 / 161
 - 7.1.2 准备编译环境 / 161
 - 7.1.3 编译 common 组件 / 162
- 7.2 Eclipse 安装 MapReduce 插件 / 166
 - 7.2.1 查找 MapReduce 插件 / 166
 - 7.2.2 新建一个 Hadoop location / 167
 - 7.2.3 Hadoop 插件操作 HDFS / 168
 - 7.2.4 运行 MapReduce 的
驱动类 / 170
- 7.3 MapReduce 的 Debug 调试 / 171

- 7.3.1 进入 Debug 运行模式 / 171
- 7.3.2 Debug 调试具体操作 / 172
- 7.4 单元测试框架 MRUnit / 174
 - 7.4.1 认识 MRUnit 框架 / 174
 - 7.4.2 准备测试案例 / 174
 - 7.4.3 Mapper 单元测试 / 176
 - 7.4.4 Reducer 单元测试 / 177
 - 7.4.5 MapReduce 单元测试 / 178
- 7.5 本章小结 / 179

第 8 章 MapReduce 编程开发 / 180

- 8.1 WordCount 案例分析 / 180
 - 8.1.1 MapReduce 工作流程 / 180
 - 8.1.2 WordCount 的 Map 过程 / 181
 - 8.1.3 WordCount 的 Reduce 过程 / 182
 - 8.1.4 每个过程产生的结果 / 182
 - 8.1.5 Mapper 抽象类 / 184
 - 8.1.6 Reducer 抽象类 / 186
 - 8.1.7 MapReduce 驱动 / 188
 - 8.1.8 MapReduce 最小驱动 / 189
- 8.2 输入格式 / 193
 - 8.2.1 InputFormat 接口 / 193
 - 8.2.2 InputSplit 类 / 195
 - 8.2.3 RecordReader 类 / 197
 - 8.2.4 应用实例：随机生成 100 个小数并求最大值 / 198
- 8.3 输出格式 / 205
 - 8.3.1 OutputFormat 接口 / 205
 - 8.3.2 RecordWriter 类 / 206
 - 8.3.3 应用实例：把首字母相同的单词放到一个文件里 / 206
- 8.4 压缩格式 / 211
 - 8.4.1 如何在 MapReduce 中使用压缩 / 211

- 8.4.2 Map 作业输出结果的压缩 / 212
- 8.5 MapReduce 优化 / 212
 - 8.5.1 Combiner 类 / 212
 - 8.5.2 Partitioner 类 / 213
 - 8.5.3 分布式缓存 / 217
- 8.6 辅助类 / 218
 - 8.6.1 读取 Hadoop 配置文件 / 218
 - 8.6.2 设置 Hadoop 的配置
文件属性 / 219
 - 8.6.3 GenericOptionsParser 选项 / 220
- 8.7 Streaming 接口 / 221
 - 8.7.1 Streaming 工作原理 / 221
 - 8.7.2 Streaming 编程接口参数 / 221
 - 8.7.3 作业配置属性 / 222
 - 8.7.4 应用实例：抓取网页的
标题 / 223
- 8.8 本章小结 / 225

第 9 章 MapReduce 高级应用 / 226

- 9.1 计数器 / 226
 - 9.1.1 默认计数器 / 226
 - 9.1.2 自定义计数器 / 229
 - 9.1.3 获取计数器 / 231
- 9.2 MapReduce 二次排序 / 232
 - 9.2.1 二次排序原理 / 232
 - 9.2.2 二次排序的算法流程 / 233
 - 9.2.3 代码实现 / 235
- 9.3 MapReduce 中的 Join 算法 / 240
 - 9.3.1 Reduce 端 Join / 240
 - 9.3.2 Map 端 Join / 242
 - 9.3.3 半连接 Semi Join / 244
- 9.4 MapReduce 从 MySQL 读写
数据 / 244
 - 9.4.1 读数据 / 245

- 9.4.2 写数据 / 248
- 9.5 Hadoop 系统调优 / 248
 - 9.5.1 小文件优化 / 249
 - 9.5.2 Map 和 Reduce 个数设置 / 249
- 9.6 本章小结 / 250

第 10 章 数据仓库工具 Hive / 251

- 10.1 认识 Hive / 251
 - 10.1.1 Hive 工作原理 / 251
 - 10.1.2 Hive 数据类型 / 252
 - 10.1.3 Hive 的特点 / 253
 - 10.1.4 Hive 下载与安装 / 255
- 10.2 Hive 架构 / 256
 - 10.2.1 Hive 用户接口 / 257
 - 10.2.2 Hive 元数据库 / 259
 - 10.2.3 Hive 的数据存储 / 262
 - 10.2.4 Hive 解释器 / 263
- 10.3 Hive 文件格式 / 264
 - 10.3.1 TextFile 格式 / 265
 - 10.3.2 SequenceFile 格式 / 265
 - 10.3.3 RCFile 文件格式 / 265
 - 10.3.4 自定义文件格式 / 269
- 10.4 Hive 操作 / 270
 - 10.4.1 表操作 / 270
 - 10.4.2 视图操作 / 278
 - 10.4.3 索引操作 / 280
 - 10.4.4 分区操作 / 283
 - 10.4.5 桶操作 / 289
- 10.5 Hive 复合类型 / 290
 - 10.5.1 Struct 类型 / 291
 - 10.5.2 Array 类型 / 292
 - 10.5.3 Map 类型 / 293
- 10.6 Hive 的 JOIN 详解 / 294
 - 10.6.1 JOIN 操作语法 / 294
 - 10.6.2 JOIN 原理 / 294

- 10.6.3 外部 JOIN / 295
- 10.6.4 Map 端 JOIN / 296
- 10.6.5 JOIN 中处理 NULL 值的语义区别 / 296
- 10.7 Hive 优化策略 / 297
 - 10.7.1 列裁剪 / 297
 - 10.7.2 Map Join 操作 / 297
 - 10.7.3 Group By 操作 / 298
 - 10.7.4 合并小文件 / 298
- 10.8 Hive 内置操作符与函数 / 298
 - 10.8.1 字符串函数 / 299
 - 10.8.2 集合统计函数 / 299
 - 10.8.3 复合类型操作 / 301
- 10.9 Hive 用户自定义函数接口 / 302
 - 10.9.1 用户自定义函数 UDF / 302
 - 10.9.2 用户自定义聚合函数 UDAF / 304
- 10.10 Hive 的权限控制 / 306
 - 10.10.1 角色的创建和删除 / 307
 - 10.10.2 角色的授权和撤销 / 307
 - 10.10.3 超级管理员权限 / 309
- 10.11 应用实例：使用 JDBC 开发 Hive 程序 / 311
 - 10.11.1 准备测试数据 / 311
 - 10.11.2 代码实现 / 311
- 10.12 本章小结 / 313

第 11 章 开源数据库 HBase / 314

- 11.1 认识 HBase / 314
 - 11.1.1 HBase 的特点 / 314
 - 11.1.2 HBase 访问接口 / 314
 - 11.1.3 HBase 存储结构 / 315
 - 11.1.4 HBase 存储格式 / 317
- 11.2 HBase 设计 / 319
 - 11.2.1 逻辑视图 / 320

- 11.2.2 框架结构及流程 / 321
- 11.2.3 Table 和 Region 的关系 / 323
- 11.2.4 -ROOT- 表和 .META. 表 / 323
- 11.3 关键算法和流程 / 324
 - 11.3.1 Region 定位 / 324
 - 11.3.2 读写过程 / 325
 - 11.3.3 Region 分配 / 327
 - 11.3.4 Region Server 上线和下线 / 327
 - 11.3.5 Master 上线和下线 / 327
- 11.4 HBase 安装 / 328
 - 11.4.1 HBase 单机安装 / 328
 - 11.4.2 HBase 分布式安装 / 330
- 11.5 HBase 的 Shell 操作 / 334
 - 11.5.1 一般操作 / 334
 - 11.5.2 DDL 操作 / 335
 - 11.5.3 DML 操作 / 337
 - 11.5.4 HBase Shell 脚本 / 339
- 11.6 HBase 客户端 / 340
 - 11.6.1 Java API 交互 / 340
 - 11.6.2 MapReduce 操作 HBase / 344
 - 11.6.3 向 HBase 中写入数据 / 348
 - 11.6.4 读取 HBase 中的数据 / 350
 - 11.6.5 Avro、REST 和 Thrift 接口 / 352
- 11.7 本章小结 / 353
- 第 12 章 Mahout 算法 / 354**
 - 12.1 Mahout 的使用 / 354
 - 12.1.1 安装 Mahout / 354
 - 12.1.2 运行一个 Mahout 案例 / 354
 - 12.2 Mahout 数据表示 / 356
 - 12.2.1 偏好 Preference 类 / 356
 - 12.2.2 数据模型 DataModel 类 / 357
 - 12.2.3 Mahout 链接 MySQL 数据库 / 358
 - 12.3 认识 Taste 框架 / 360
 - 12.4 Mahout 推荐器 / 361
 - 12.4.1 基于用户的推荐器 / 361
 - 12.4.2 基于项目的推荐器 / 362
 - 12.4.3 Slope One 推荐策略 / 363
 - 12.5 推荐系统 / 365
 - 12.5.1 个性化推荐 / 365
 - 12.5.2 商品推荐系统案例 / 366
 - 12.6 本章小结 / 370
- 附录 A Hive 内置操作符与函数 / 371**
- 附录 B HBase 默认配置解释 / 392**
- 附录 C Hadoop 三个配置文件的参数含义说明 / 398**

第1章 Hadoop 概述

Hadoop 是一个开发和运行处理大规模数据的软件平台，是 Apache 的一个用 Java 语言实现开源软件框架，实现在大量计算机组成的集群中对海量数据进行分布式计算。

1.1 Hadoop 起源

Hadoop 框架中最核心设计就是：HDFS 和 MapReduce。HDFS 提供了海量数据的存储，MapReduce 提供了对数据的计算。

1.1.1 Google 与 Hadoop 模块

Google 的数据中心使用廉价的 Linux PC 机组成集群，在上面运行各种应用。即使是分布式开发的新手也可以迅速使用 Google 的基础设施。Hadoop 核心组件与 Google 对应的组件对应关系如表 1-1 所示。

表 1-1 Google 与 Hadoop 对应模块

Google	功能描述	对应 Hadoop 模块
GFS	分布式文件系统（Google File System），隐藏下层负载均衡、冗余复制等细节，对上层程序提供一个统一的文件系统 API 接口。Google 根据自己的需求对它进行了特别优化，包括超大文件的访问、读操作比例远超过写操作、PC 机极易发生故障造成节点失效等。GFS 把文件分成 64MB 的块，分布在集群的机器上，使用 Linux 的文件系统存放，同时每块文件至少有 3 份以上的冗余。中心是一个 Master 节点，根据文件索引找寻文件块	HDFS
MapReduce	Google 发现大多数分布式运算可以抽象为 MapReduce 操作。Map 是把输入 Input 分解成中间的 Key-Value 对，Reduce 把 Key-Value 合成最终输出 Output。这两个函数由程序员提供给系统，下层设施把 Map 和 Reduce 操作分布在集群上运行，并把结果存储在 GFS 上	MapReduce
BigTable	一个大型的分布式数据库，这个数据库不是关系式的数据库。像它的名字一样，就是一个巨大的表格，用来存储结构化的数据	HBase

1.1.2 为什么会有 Hadoop

随着互联网快速的发展，产生的日志数量级的增加，大量的日志给公司带来了很大的挑战。如日志存储问题、海量日志分析的效率问题、成本问题等。

下面我们来分析个问题。

一般网站把用户的访问行为记录以 Apache 日志的形式记录下来，这些日志中包含了用

户访问网站的所有信息，下面列举关键的信息，关键字段如下。

客户端 IP	用户标示	访问时间	访问 URL	关联的 URL	状态	流量	代理
client_ip	user_ip	access_time	url	refrence	status	page_size	agent

因为需要统一对数据进行离线计算，所以常常把它们全部移到同一个地方，如 Oracle 数据库等，每天产生的日志量大概计算一下，如下所示。

□ 网站请求数：1000 万条 / 天

□ 每天日志大小：450 字节 / 行 × 1000 万条 = 4.2 GB

□ 日志存储周期：2 年

一天产生 4.5 GB 日志，2 年需要 $4.2 \text{ GB} \times 2 \times 365 = 3.0 \text{ TB}$ 。

怎么来解决 3.0 TB 的数据备份和容错的问题？解决方案如下

1) 为了方便系统命令查看日志，不压缩，总共需要 3.0 TB 空间，刚好有一些 2U 的服务器，每台 1 TB 的磁盘空间。

2) 为了避免系统盘坏掉影响服务器使用，对系统盘做 Raid1。

3) 为了避免其他存放数据的盘坏掉导致数据无法恢复，对剩下的盘做 Raid5。

所有的数据都汇聚到这几台 LogBackUp 服务器上了。有了 LogBackUp 服务器，离线统计就可以全部在这些服务器上进行了。在这套架构上，用 wc、grep、sort、uniq、awk、sed 等系统命令，完成了很多的统计需求，比如统计访问频率较高的 client_ip，某个新上线的页面的 reference 主要是哪些网站。

当公司业务迅猛发展，网站流量爆发增长，日志量也就成指数级增长，日志对于一个互联网公司来说是非常重要的，互联网公司可以通过分析日志来获取用户的行为，如推荐系统、淘宝的用户买卖行为分析等。假如现在的日志量计算如下所示。

□ 日志总行数：10 亿 / 天

□ 每天日志大小：450 字节 / 行 × 10 亿 = 420 GB

□ 日志种类：5 种

1 天产生 420 GB 日志，2 年需要 $420 \text{ GB} \times 2 \times 365 = 300 \text{ TB}$ ，这么大的数据量怎么来存储和分析？

1.1.3 Hadoop 版本介绍

最新发布的 Apache Hadoop 版本如图 1-1 所示。一些 Hadoop 入门者会问：这个版本的功能有哪些？基于哪个版本？后续的版本是什么？要解释这一点，我们应该从 Apache 项目发布的一些基本知识开始分析。一般来说，Apache 项目的新功能在主干（trunk）代码上开发。有时候，很大的特性也会有自己的开发分支（branch），它们期望后续会并入 trunk。新功能通常是在 trunk 发布之前就有的，一般质量或稳定性没有太大保证。候选的分支会定期从主干分支上分离出来发布。一旦一个候选分支发布，它通常停止获得新的功能。如果有 bug 修复，经过投票后，会针对这个特定的分支再发布一个新版本。社区的任何成员可以创建一个版本分支，并可随意命名。