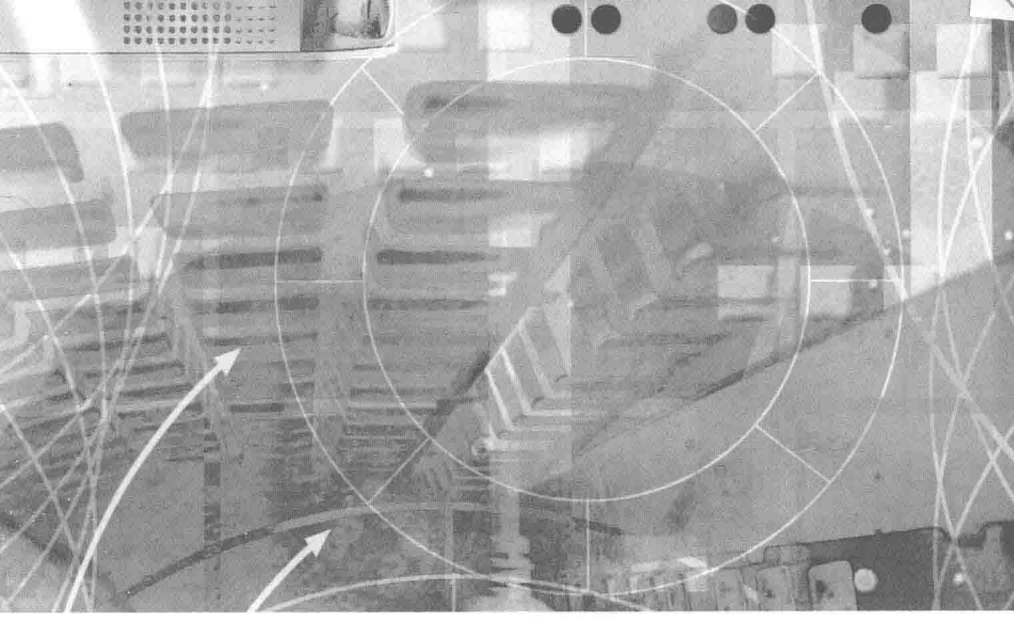
数据馆员的 Spark 简明手册

顾立平　马景源　**编著**

科学技术文献出版社
SCIENTIFIC AND TECHNICAL DOCUMENTATION PRESS



数据馆员的 Spark 简明手册

>>> 顾立平 马景源 编著

科学技术文献出版社
SCIENTIFIC AND TECHNICAL DOCUMENTATION PRESS

·北京·

图书在版编目(CIP)数据

数据馆员的Spark简明手册 / 顾立平，马景源编著. —北京：科学技术文献出版社，2017. 10

ISBN 978-7-5189-3015-9

Ⅰ. ①数… Ⅱ. ①顾… ②马… Ⅲ. ①数据处理软件—技术手册 Ⅳ. ① TP274-62

中国版本图书馆 CIP 数据核字（2017）第 161353 号

数据馆员的Spark简明手册

策划编辑：崔灵菲　责任编辑：崔灵菲　责任校对：张吲哚　责任出版：张志平

出　版　者　科学技术文献出版社
地　　址　北京市复兴路15号　邮编 100038
编　务　部　（010）58882938，58882087（传真）
发　行　部　（010）58882868，58882874（传真）
邮　购　部　（010）58882873
官方网址　www.stdp.com.cn
发　行　者　科学技术文献出版社发行　全国各地新华书店经销
印　刷　者　虎彩印艺股份有限公司
版　　次　2017 年 10 月第 1 版　2017 年 10 月第 1 次印刷
开　　本　850 × 1168　1/32
字　　数　46千
印　　张　2.875
书　　号　ISBN 978-7-5189-3015-9
定　　价　28.00元

Preface 前言

本手册旨在协助初级数据馆员们能够迅速了解 Spark 方面的知识、用途及整体概貌，作为进一步实践操作之前的入门基础读物。

数据馆员是能够充分实现开放科学政策、措施、服务的一群新型信息管理人员，他们熟悉数据处理、数据分析、数据权益、数据政策，且具有知识产权与开放获取的知识和经验。

Spark 是由美国加州大学伯克利分校（UC Berkeley）开源的计算框架，其特点是能够将任务的中间结果保存在内存中，不进行读写磁盘的操作，因而能够实现更快的处理。它在解决复杂线性代数、某些优化问题、迭代计算、机器学习等方面具有较强优势。作为一种适合实时计算的方案，Spark 是进行大数据分析的一种有力工具。

本手册力求简单、通俗、易懂，以读者能够快速把握

重点为主，从而开展项目、课题、实验和研究。本手册旨在知识模块化，有了整体概述，可以方便读者与其他解决方案进行比较，在实践中遇到问题可以尽快发现需要深入钻研的部分。

本手册包括 8 章。第 1 章概述 Spark 的发展背景、计算框架及机器学习等。第 2 章描述 Spark 的安装与运行。第 3 章概述 Scala 编程实现的方式。第 4 章概述 Spark 编程模型和解析。第 5 章进入到 Spark 数据挖掘的应用。第 6 章考虑大数据实时计算的问题，进行方案比较，突出 Spark 的特点。第 7 章阐明进一步优化 Spark 的方式。第 8 章概述 Spark SQL 来阐明如何在 Spark 上使用人们比较熟悉的 SQL 数据库语言的方式。

在掌握全部知识点的基础上，通过搭建、测试、运行、试验之后，读者可以逐步参照其他已有的案例经验和 Spark 深入源码的著作进行进一步的探索应用。

编著者

2017 年初春于中关村

Contents 目录

第 1 章 Spark 生态介绍 1

1.1 MapReduce、Storm 和 Spark 模型比较 1

1.2 Spark 产生背景 3

1.3 Spark 的内存计算框架 5

1.4 Spark Streaming：流式计算框架 6

1.5 Spark SQL 7

1.6 Spark MLlib：机器学习 8

1.7 Spark GraphX 和取代 Bagel 的理由 8

1.8 BlinkDB 9

1.9 SparkR 9

第 2 章 Spark 的安装与运行 10

2.1 Spark 的安装 10

2.1.1 Spark 的源码编译方式 10

2.1.2 Spark Standalone 安装 12

2.1.3 Spark 应用程序部署工具 spark-submit 14

2.1.4 Spark 的高可用性部署 15

2.2 Spark 的运行架构 16

2.2.1 基本术语 16

2.2.2 运行架构 17
2.2.3 Spark on Standalone 的运行过程 19
2.2.4 Spark on YARN 的运行过程 20
2.3 Spark 的运行 22
2.3.1 Spark on Standalone 22
2.3.2 Spark on YARN 22
2.3.3 Standalone 与 YARN 模式优缺点比较 23

第 3 章 Spark 的 Scala 编程 25
3.1 Scala 开发环境搭建 25
3.2 Scala 开发 Spark 应用程序 25
3.3 编程实现 26
3.3.1 使用 Java 编程 26
3.3.2 使用 Python 编程 27

第 4 章 Spark 的编程模型和解析 28
4.1 Spark 的编程模型 28
4.2 RDD 的特点、操作、依赖关系 28
4.3 Spark 应用程序的配置 31
4.4 Spark 的架构 31
4.5 Spark 的容错机制 32
4.6 数据的本地性 32
4.7 缓存策略介绍 33
4.8 宽依赖和窄依赖 35

第 5 章 Spark 数据挖掘 38
5.1 MLlib 38
5.2 GraphX 39
5.2.1 GraphX 原理 39
5.2.2 Table Operator 和 Graph Operator 的区别 40
5.2.3 Vertices、Edges 和 Triplets 介绍 42
5.2.4 GraphX 图构造者 43
5.3 SparkR 45
5.3.1 SparkR 原理 45
5.3.2 如何运行 SparkR 46

第 6 章 Spark Streaming 48
6.1 Spark Streaming 与 Storm 的区别 48
6.2 Kafka 的部署 49
6.3 Kafka 与 Spark Streaming 的整合 50
6.4 Spark Streaming 原理 52
6.4.1 Spark 流式处理架构 52
6.4.2 DStream 的特点 53
6.4.3 Dstream 的操作和 RDD 的区别 54
6.4.4 无状态转换操作与有状态转换操作 54
6.4.5 优化 Spark Streaming 55
6.5 Streaming 的容错机制 56
6.6 Streaming 在 YARN 模式下的注意事项 57

第 7 章 Spark 优化 59
7.1 序列化优化——Kryo 59
7.2 Spark 参数优化 60
7.3 Spark 任务的均匀分布策略 61
7.4 Partition key 倾斜的解决方案 63
7.5 Spark 任务的监控 63
7.6 GC 的优化 65
7.7 Spark Streaming 吞吐量优化 69
7.8 Spark RDD 使用内存的优化策略 70

第 8 章 SQL on Spark 72
8.1 BDAS 数据分析软件栈 72
8.2 Spark SQL 工具 74
8.3 Spark SQL 原理 76
8.4 Spark SQL 编程 78

>>>>>> 第 1 章

Spark 生态介绍

1.1 MapReduce、Storm 和 Spark 模型比较

MapReduce 是 Google 于 2004 年提出的能并发处理海量数据的并行编程模型。对大规模数据进行处理时，单个计算机的计算能力显得越来越吃力，而分布式计算对于程序开发人员又过于复杂，此时 MapReduce 应运而生，它的出现使得并不了解分布式计算底层细节的开发人员能够开发分布式程序。

MapReduce 模型将处理过程分为两步，即“Map”和“Reduce”两部分。简单地说，Map 部分将任务分块分布执行，再由 Reduce 部分将各个分块的结果进行归约、汇总。总体的框架如图 1–1 所示。

Storm 是于 2011 年由 BackType 开发并被 Twitter 开源的分布式实时流数据计算系统，它能够使用简单的方式可靠地处理无界持续的流数据，可用来实时处理新数据和更新数据库，兼具容错性和扩展性。Storm 集群的输入流由名

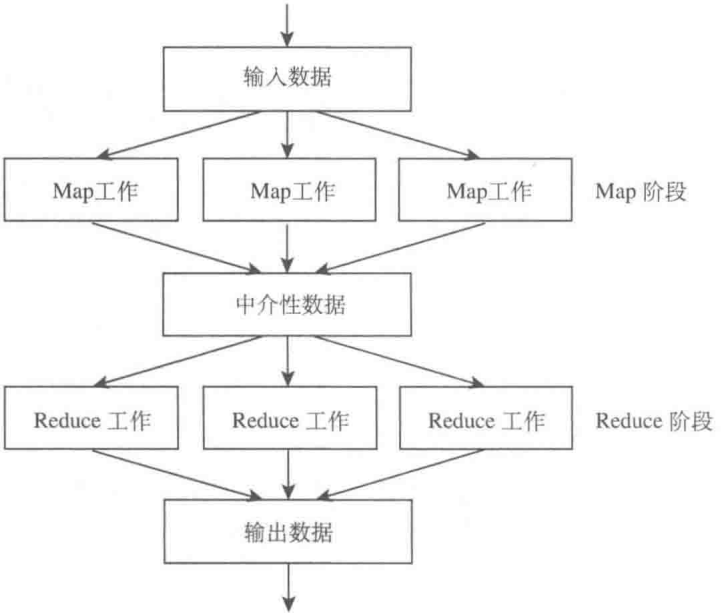


图 1-1 MapReduce 模型工作原理

为 Spout 的组件负责。Spout 将数据传递给名为 Bolt 的组件，后者将以指定的方式处理这些数据，如持久化或者处理并转发给另外的 Bolt。由这些组成一个 Topology，通过这个 Topology 可以实现大多数的需求。

Spark 是美国加州大学伯克利分校 AMP 实验室开发的集群计算平台，是一个基于内存计算的可扩展的开源集群计算系统。针对 MapReduce 的不足，即大量的网络传输和磁盘 I/O 使得效率低下，Spark 使用内存进行数据计算以便快速处理查询，实时返回分析结果。

三者比较如下。

MapReduce 属于批量处理系统。这种系统处理的数据通常以静态的形式存储在硬盘中，很少进行更新，存储时间长，且这些数据精确度较高，但是价值密度较低。以上原因导致系统相对适应更加稳定的工作，如电子商务领域对客户购买记录、浏览记录等数据进行分析，预测客户需求，再如对患者病历、生活方式等进行分析，建立更加准确的预测体系等。

Storm 属于流式数据处理系统。流式数据是一个无穷的数据序列，序列中的每一个元素来源各异，格式复杂，序列往往包含时序特性。这类处理系统一般应用于互联网数据采集器、银行系统交易监测器等。

Spark 则属于交互式数据处理系统。在这个系统下，系统与操作人员以人机对话的形式进行数据处理。采用这种方式，存储在系统中的数据文件能够被及时处理，同时处理结果可以立刻被使用。交互式数据处理具备的这些特征能够保证输入的信息得到及时处理，使交互方式继续进行下去。目前分布式数据仓库就属于这类系统的应用。

1.2　Spark 产生背景

21 世纪初，随着计算机和互联网的发展，数据从类型

到量上都呈爆炸式增长。人们很快产生了对这些数据进行分析的需求。2004 年 Google 发表了一篇关于 MapReduce 的论文，开创了大数据领域的大量研究。随后 Hadoop 的出现几乎成了数据处理的事实标准。Hadoop 中包含了 MR（MapReduce）的实现，然而还是存在一些即便通过它仍难以解决的问题。

Hadoop 对于数据的基础分析，如计算均值、方差、中值等是可以胜任的，但是对于复杂的分析任务如主成分分析、核回归、SVM、图论计算、积分、优化等分析问题则显得较为吃力。另外，由于 Hadoop 在处理过程中需要在磁盘上进行大量的 I/O 操作，无疑大大降低了处理的效率。同样是由于效率的原因，使得 Hadoop 难以在面对实时数据处理、交互式处理的问题上做出令人满意的表现。

Spark 及其生态圈的出现，能从很大程度上解决上述问题。Spark 是由加州大学伯克利分校开源的计算框架。特点是能够将任务的中间结果保存在内存中，不进行读写磁盘的操作，因而能够实现更快的处理。它在解决复杂线性代数、某些优化问题、迭代计算、机器学习等方面能够完胜 Hadoop。对于图论计算等问题，Spark 也提出了 GraphX 等一系列项目进行解决。另外，Spark 在实时处理方面也表现出强大的功能。

1.3　Spark 的内存计算框架

Spark 的核心概念是 RDD（Resilient Distributed Dataset，弹性分布式数据集）。在 Spark 中，所有的数据集都被包装成 RDD 进行操作。每次 RDD 操作结束以后都可以存储至内存，下一个操作可以直接从内存中输入。Spark 数据流如图 1–2 所示。

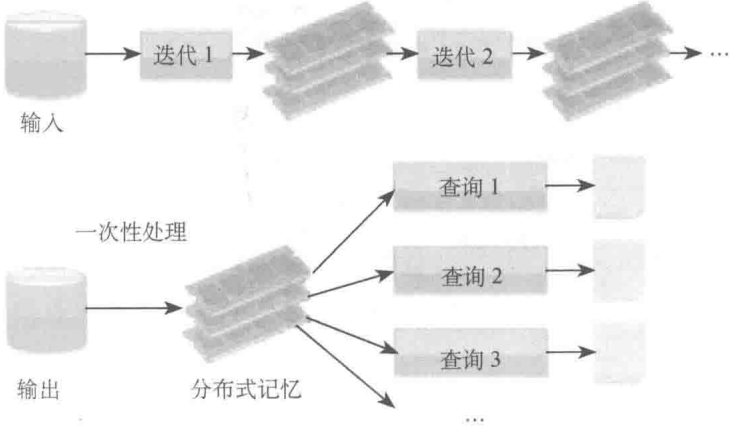


图 1–2　Spark 数据流

由于在内存中进行操作的速度远远大于从磁盘上进行读写的速度，Spark 这种将中间结果保存在内存中的方式与传统的 Hadoop MR 相比，效率上有了巨大的提升。

1.4 Spark Streaming：流式计算框架

Spark Streaming 是建立在 Spark 上的应用框架，属于 Spark 的核心 API，支持高吞吐量、支持容错的实时流数据处理。基于 Spark on YARN 的 Spark Streaming 架构如图 1–3 所示。

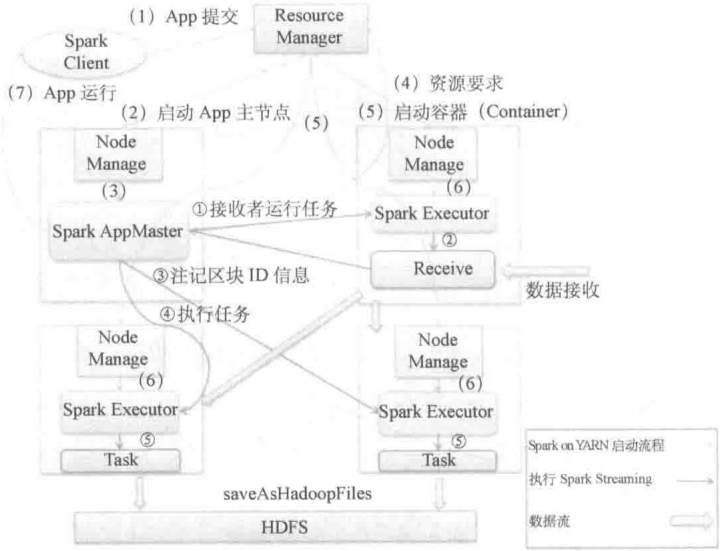


图 1–3 Spark Streaming 架构

Spark Streaming 的计算流程如下：以时间片为单位将数据流进行拆分，然后以类似批处理的方式处理每个时间片的数据，每一块数据都转换成 Spark 的 RDD，然后使用

RDD 处理每一块数据，每一块都会生成一个 Spark Job 处理，最终结果也返回多块。处理流程如图 1-4 所示。

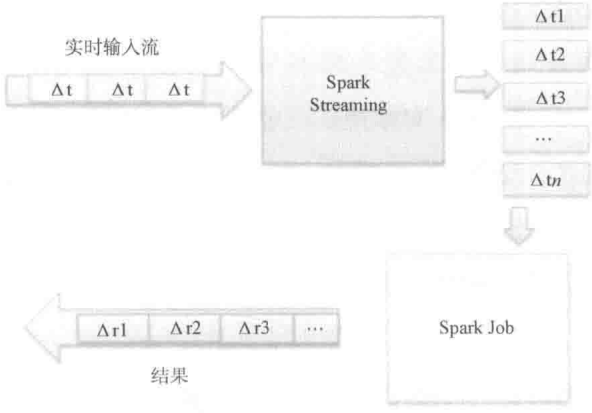


图 1-4　Spark Streaming 处理流程

这种处理方式有较强的容错性，且实时性较好，最小时间片流为 0.5 ～ 2 s，基本能够满足所有准实时性计算场景的要求。

1.5　Spark SQL

Spark SQL 是支持在 Spark 中使用 SQL、HiveSQL、Scala 的关系型查询表达式。其核心组件是一个新增的 RDD 类型的 SchemaRDD，它用一个 Schema 来描述行里所有列的数据类型，类似于关系型数据库的一张表。

Spark SQL 的核心是把已有的 RDD 带上 Schema 信息，然后注册成类似 SQL 里的 Table，对其进行 SQL 查询。

1.6 Spark MLlib：机器学习

MLlib 是 Spark 对常用的机器学习算法的实现库，也包括相应的测试和数据生成器。

目前支持的算法有：基本统计、分类与回归、*K*-means 聚类、主成分分析、奇异值分解、特征值提取及转换等。

1.7 Spark GraphX 和取代 Bagel 的理由

早在 0.5 版本，Spark 就带了一个小型的 Bagel 模块，是 Pregel 在 Spark 上轻量级的实现。当然，这个版本非常原始，性能和功能都比较弱，属于实验型产品。

到 0.8 版本时，鉴于业界对分布式图计算的需求日益见涨，Spark 开始独立一个分支 GraphX-Branch，作为独立的图计算模块，借鉴了 GraphLab，开始设计开发 GraphX。

到 0.9 版本时，官方开始鼓励开发人员使用 GraphX 取代 Bagel。

与 Bagel 相比，GraphX 提供了更加丰富的图计算 API。GraphX 继承了 Spark 中的 RDD，并进行了系统的优