

数据中国“百校工程”项目系列教材
数据科学与大数据技术专业系列规划教材

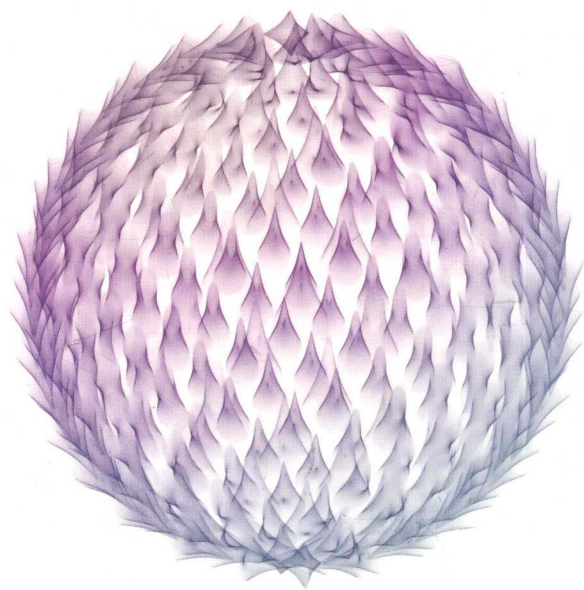
 瑞翼教育

Spark

大数据技术与应用

赵红艳 许桂秋 ● 主编

潘晓洋 张越 李阳 张军 王露露 ● 副主编



BIG DATA

Technology

 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS

数据中国“百校工程”项目系列教材
数据科学与大数据技术专业系列规划教材

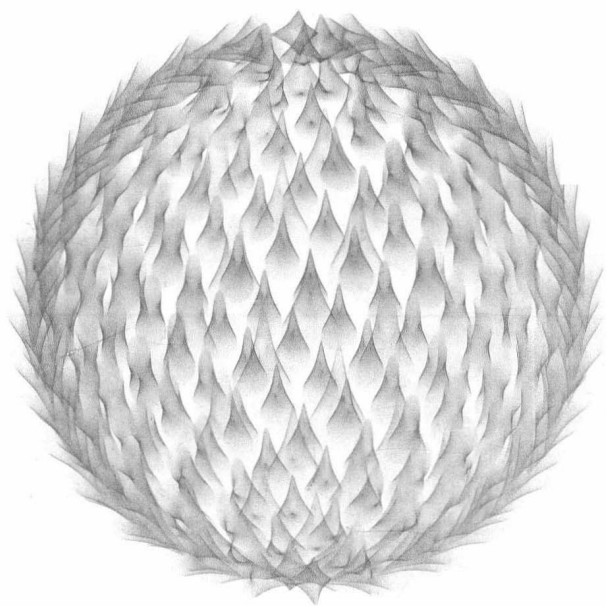
 瑞翼教育

Spark

大数据技术与应用

赵红艳 许桂秋 ● 主编

潘晓洋 张越 李阳 张军 王露露 ● 副主编



BIG DATA
Technology

人民邮电出版社

北京

图书在版编目 (C I P) 数据

Spark大数据技术与应用 / 赵红艳, 许桂秋主编. —
北京: 人民邮电出版社, 2019. 4
数据科学与大数据技术专业系列规划教材
ISBN 978-7-115-50347-3

I. ①S… II. ①赵… ②许… III. ①数据处理软件—
教材 IV. ①TP274

中国版本图书馆CIP数据核字(2019)第029374号

内 容 提 要

本书采用理论与实践相结合的方式, 介绍了 Spark 大数据分析计算框架的基础知识, 培养读者使用 Spark 解决实际问题的能力。本书内容包括: Spark 简介与运行原理、Spark 的环境搭建、使用 Python 开发 Spark 应用、Spark RDD、DataFrame 与 Spark SQL、Spark Streaming、Spark 机器学习库、GraphFrames 图计算, 并给出了两个综合案例: 出租车数据分析、图书推荐系统。

本书可作为高等院校计算机、数据科学与大数据技术等相关专业的教材, 也可作为 Spark 开发人员的参考用书。

-
- ◆ 主 编 赵红艳 许桂秋
 - 副 主 编 潘晓洋 张 越 李 阳 张 军 王露露
 - 责任编辑 李 召
 - 责任印制 陈 彝

 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 固安县铭成印刷有限公司印刷

 - ◆ 开本: 787×1092 1/16
 - 印张: 8.75 2019 年 4 月第 1 版
 - 字数: 214 千字 2019 年 4 月河北第 1 次印刷
-

定价: 39.80 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

本课程建议安排 64 课时，教师可根据学生的实际学习情况以及高校的培养方案选择教学内容。本书可以作为高等院校计算机、数据科学与大数据技术等相关专业的教材，也可作为 Spark 开发人员的参考用书。

由于编者水平有限，书中难免出现一些疏漏和不足，恳请广大读者批评指正。

编者

2019 年 1 月

目 录

第 1 章 Spark 简介与运行原理1	2.6 小结.....20
1.1 Spark 是什么.....1	习题.....20
1.1.1 Spark 的版本发展历程.....2	第 3 章 使用 Python 开发 Spark
1.1.2 Spark 与 Hadoop 的区别与联系.....2	应用21
1.1.3 Spark 的应用场景.....3	3.1 Python 编程语言.....21
1.2 Spark 的生态系统.....3	3.1.1 Python 语言介绍.....21
1.3 Spark 的架构与原理.....4	3.1.2 PySpark 是什么.....22
1.3.1 Spark 架构设计.....4	3.2 PySpark 的启动与日志设置.....22
1.3.2 Spark 作业运行流程.....5	3.2.1 PySpark 的启动方式.....22
1.3.3 Spark 分布式计算流程.....6	3.2.2 日志输出内容控制.....24
1.4 Spark 2.X 新特性.....6	3.3 PySpark 开发包的安装.....24
1.4.1 精简的 API.....6	3.3.1 使用 pip 命令安装.....24
1.4.2 Spark 作为编译器.....7	3.3.2 使用离线包安装.....25
1.4.3 智能化程度.....7	3.4 使用 PyCharm 编写 Spark 应用.....25
1.5 小结.....7	3.4.1 PyCharm 的安装与基本配置.....25
习题.....8	3.4.2 编写 Spark 应用.....27
第 2 章 Spark 的环境搭建9	3.5 小结.....29
2.1 环境搭建前的准备.....9	习题.....30
2.2 Spark 相关配置.....13	第 4 章 Spark RDD31
2.2.1 安装 SSH.....13	4.1 弹性分布式数据集.....31
2.2.2 SSH 免密码登录.....14	4.1.1 RDD 的定义.....31
2.2.3 修改访问权限.....15	4.1.2 RDD 的特点.....32
2.2.4 修改 profile 文件.....15	4.1.3 RDD 的创建.....33
2.2.5 修改 Spark 配置文件.....16	4.1.4 RDD 的操作.....34
2.3 Spark 集群启动与关闭.....17	4.2 transform 算子.....34
2.4 Spark 应用提交到集群.....18	4.2.1 map 转换.....34
2.5 Spark Web 监控页面.....19	4.2.2 flatMap 转换.....35

4.2.3	filter 转换	35	4.6	共享变量	45
4.2.4	union 转换	35	4.6.1	累加器	45
4.2.5	intersection 转换	36	4.6.2	广播变量	45
4.2.6	distinct 转换	36	4.7	依赖关系	47
4.2.7	sortBy 转换	36	4.7.1	血统	47
4.2.8	mapPartitions 转换	36	4.7.2	宽依赖与窄依赖	47
4.2.9	mapPartitionsWithIndex 转换	37	4.7.3	shuffle	48
4.2.10	partitionBy 转换	37	4.7.4	DAG 的生成	49
4.3	action 算子	37	4.8	Spark RDD 的持久化	50
4.3.1	reduce(f)动作	37	4.8.1	持久化使用方法	50
4.3.2	collect()动作	38	4.8.2	持久化存储等级	51
4.3.3	count()动作	38	4.8.3	检查点	52
4.3.4	take(num)动作	39	4.9	小结	52
4.3.5	first()动作	39	习题	52	
4.3.6	top(num)动作	39	第 5 章 DataFrame 与 Spark SQL		
4.3.7	saveAsTextFile()动作	39	SQL		
4.3.8	foreach(f)动作	40	5.1	DataFrame	54
4.3.9	foreachPartition(f)动作	40	5.1.1	DataFrame 介绍	54
4.4	RDD Key-Value 转换算子	41	5.1.2	DataFrame 创建	55
4.4.1	mapValues(f)操作	41	5.2	Spark SQL	56
4.4.2	flatMapValues(f)操作	41	5.2.1	Spark SQL 介绍	56
4.4.3	combineByKey 操作	41	5.2.2	Spark SQL 的执行原理	57
4.4.4	reduceByKey 操作	42	5.2.3	Spark SQL 的创建	58
4.4.5	groupByKey 操作	42	5.3	Spark SQL、DataFrame 的常用操作	61
4.4.6	sortByKey 操作	43	5.3.1	字段计算	61
4.4.7	keys()操作	43	5.3.2	条件查询	62
4.4.8	values()操作	43	5.3.3	数据排序	63
4.4.9	join 操作	43	5.3.4	数据去重	63
4.4.10	leftOuterJoin 操作	43	5.3.5	数据分组统计	64
4.4.11	rightOuterJoin 操作	44	5.3.6	数据连接	65
4.5	RDD Key-Value 动作运算	44	5.3.7	数据绘图	67
4.5.1	collectAsMap()操作	44	5.4	小结	68
4.5.2	countByKey()操作	44	习题	69	

第 6 章 Spark Streaming70	7.3.4 预测婴儿生存机会.....92
6.1 Spark Streaming 介绍.....70	7.4 使用 ML 机器学习库.....93
6.1.1 什么是 Spark Streaming.....70	7.4.1 转换器、评估器和管道.....94
6.1.2 Spark Streaming 工作原理.....70	7.4.2 预测婴儿生存率.....95
6.2 流数据加载.....71	7.5 小结.....97
6.2.1 初始化 StreamingContext.....71	习题.....97
6.2.2 Discretized Stream 离散化流.....71	第 8 章 GraphFrames 图计算98
6.2.3 Spark Streaming 数据源.....72	8.1 图.....98
6.3 DStream 输出操作.....73	8.1.1 度.....99
6.4 DStream 转换操作.....75	8.1.2 路径和环.....99
6.4.1 map 转换.....75	8.1.3 二分图.....100
6.4.2 flatMap 转换.....76	8.1.4 多重图和伪图.....100
6.4.3 filter 转换.....76	8.2 GraphFrames 介绍.....101
6.4.4 reduceByKey 转换.....77	8.2.1 应用背景.....101
6.4.5 count 转换.....77	8.2.2 GraphFrames 库.....102
6.4.6 updateStateByKey 转换.....77	8.2.3 使用 GraphFrames 库.....102
6.4.7 其他转换.....78	8.3 GraphFrame 编程模型.....102
6.5 DataFrame 与 SQL 操作.....78	8.3.1 GraphFrame 实例.....103
6.6 实时 WordCount 实验.....79	8.3.2 视图和图操作.....104
6.7 小结.....81	8.3.3 模式发现.....105
习题.....81	8.3.4 图加载和保存.....105
第 7 章 Spark 机器学习库82	8.4 GraphFrames 实现的算法.....106
7.1 Spark 机器学习库.....82	8.4.1 广度优先搜索.....106
7.1.1 机器学习简介.....82	8.4.2 最短路径.....106
7.1.2 Spark 机器学习库的构成.....82	8.4.3 三角形计数.....107
7.2 准备数据.....83	8.4.4 连通分量.....107
7.2.1 获取数据.....83	8.4.5 标签传播算法.....108
7.2.2 数据预处理.....84	8.4.6 PageRank 算法.....109
7.2.3 数据探索.....84	8.5 基于 GraphFrames 的网页排名.....110
7.3 使用 MLlib 机器学习库.....85	8.5.1 准备数据集.....110
7.3.1 搭建环境.....85	8.5.2 创建 GraphFrames.....111
7.3.2 加载数据.....86	8.5.3 使用 PageRank 进行网页排名.....111
7.3.3 探索数据.....89	8.6 小结.....111

习题	111	10.1.4 View 视图	120
第 9 章 出租车数据分析	112	10.2 Django 项目搭建	121
9.1 数据处理	112	10.2.1 创建项目	121
9.2 数据分析	113	10.2.2 创建应用	122
9.2.1 创建 DataFrame	113	10.2.3 创建模型	122
9.2.2 KMeans 聚类分析	114	10.3 推荐引擎设计	124
9.3 百度地图可视化	115	10.3.1 导入数据	124
9.3.1 申请地图 key	115	10.3.2 训练模型	126
9.3.2 聚类结果可视化	116	10.3.3 图书推荐	127
9.4 小结	117	10.4 系统设计与实现	128
第 10 章 图书推荐系统	118	10.4.1 Bootstrap 介绍与使用	128
10.1 Django 简介	118	10.4.2 Redis 数据库安装与使用	129
10.1.1 Django 是什么	118	10.4.3 视图与路由设计	130
10.1.2 ORM 模型	119	10.5 小结	132
10.1.3 Django 模板	119		

第 1 章

Spark 简介与运行原理

Spark 是现在流行的大数据分析计算框架，在大数据应用中起着不可或缺的作用。本章从 Spark 的产生、发展及其生态圈等方面对 Spark 进行介绍。

本章主要内容如下。

- (1) Spark 是什么。
- (2) Spark 的生态系统。
- (3) Spark 架构与原理。
- (4) Spark 2.X 新特性。

1.1 Spark 是什么

Spark 是 2009 年由马泰·扎哈里亚 (Matei Zaharia) 在加州大学伯克利分校的 AMPLab 实验室开发的子项目，经过开源后捐赠给 Apache 软件基金会，最后成为我们现在众所周知的 Apache Spark。它是由 Scala 语言实现的专门为大规模数据处理而设计的快速通用的计算引擎。经过多年的发展，现已形成了一个高速发展、应用广泛的生态系统。

Spark 主要有以下 3 个特点。

- (1) Spark 提供了高级应用程序编程接口 (Application Programming Interface, API)，应用开发者只用专注于应用计算本身即可，而不用关注集群。
- (2) Spark 计算速度快，支持交互式计算和复杂算法。
- (3) Spark 是一个通用引擎，可用它来完成各种运算，包括 SQL 查询、文本处理、机器学习、实时流处理等。在 Spark 出现之前，我们一般需要学习使用各种各样的大数据分析引擎来分别实现这些需求。

1.1.1 Spark 的版本发展历程

Spark 从诞生至今迭代了很多个版本，其性能和生态也是越来越好，目前已经升级到 2.3.2 版本。其主要发展历程如表 1-1 所示。

表 1-1 Spark 的版本发展历程

年代	说明
2009	Spark 由 Matei Zaharia 在加州大学伯克利分校的 AMPLab 实验室开发
2010	通过 BSD 授权条款发布开放源码
2013	Spark 项目被捐赠给 Apache 软件基金会
2014/2	Spark 成为 Apache 的顶级项目
2014/11	Databricks 团队使用 Spark 刷新数据排序的世界纪录
2015/3	Spark 1.3.0 版本发布，开始加入 DataFrame 与 SparkML
2016/7	Spark 2.0.0 版本发布，提升执行性能，更容易被使用
2017/7	Spark 2.2.0 版本发布，从结构化流中删除实验标签
2018/2	Spark 2.3.0 版本发布，增加对结构流连续处理的支持
2018/9	Spark 2.3.2 版本发布

1.1.2 Spark 与 Hadoop 的区别与联系

Spark 与 Hadoop 处理的许多任务相同，但是在以下两个方面不相同。

(1) 解决问题的方式不一样

Hadoop 和 Spark 两者都是大数据框架，但是各自的属性和性能却不完全相同。Hadoop 是一个分布式数据基础设施，它将巨大的数据集分派到一个由普通计算机组成的集群中的多个节点进行存储，这意味着我们不需要购买和维护昂贵的服务器硬件。同时，Hadoop 还会对这些数据进行排序和追踪，这使得大数据处理和分析更加迅速高效。

Spark 则是一个专门用来对分布式存储的大数据进行处理的工具，但它并不会进行分布式数据的存储。

(2) 两者可合可分

Hadoop 不仅提供了 HDFS 分布式数据存储功能，还提供了 MapReduce 的数据处理功能。因此我们可以不使用 Spark，而选择使用 Hadoop 自身的 MapReduce 对数据进行处理。

试读结束，需要全本请在线购买：www.ertongbook.com

同样，Spark 也不一定需要依附在 Hadoop 系统中。但如上所述，因为 Spark 没有提供文件管理系统，所以它需要和其他的分布式文件系统先进行集成然后才能运作。

1.1.3 Spark 的应用场景

Spark 使用了内存分布式数据集技术，除了能够提供交互式查询外，它还提升了迭代工作负载的性能。在互联网领域，Spark 有快速查询、实时日志采集处理、业务推荐、定制广告、用户图计算等强大功能。国内外的一些大公司，比如谷歌（Google）、阿里巴巴、英特尔（Intel）、网易、科大讯飞等都有实际业务运行在 Spark 平台上。

下面简单介绍一下 Spark 在各个领域中的用途。

（1）快速查询系统。基于日志数据的快速查询系统业务构建于 Spark 之上，利用其快速查询和内存表等优势，Spark 能够承担大多数日志数据的即时查询工作，在性能方面普遍比 Hive 快 2~10 倍。如果借助内存表的功能，性能将会比 Hive 快百倍。

（2）实时日志采集处理系统。Spark 流处理模块对业务日志进行实时快速迭代处理，并进行综合分析，用来满足线上系统分析要求。

（3）业务推荐系统。Spark 将业务推荐系统的小时和天级别的模型训练，转变为分钟级别的模型训练；能有效地优化相关排名、个性化推荐以及热点分析等。

（4）定制广告系统。定制广告业务需要大数据做应用分析、效果分析、定向优化等，借助 Spark 快速迭代的优势，可以实现在“数据实时采集、算法实时训练、系统实时预测”的全流程实时并行高维算法，支持上亿的请求量处理。模拟广告投放计算延迟小、效率高，同 MapReduce 相比，延迟至少降低一个数量级。

（5）用户图计算。利用 Spark 图计算解决了许多生产问题，如基于分布的中枢节点发现、基于最大连通图的社区发现、基于三角形计数的关系衡量、基于随机游走的用户属性传播等。

1.2 Spark 的生态系统

Spark 生态系统以 Spark Core 为核心，利用 Standalone、YARN 和 Mesos 等进行资源调度管理，完成应用程序分析与处理。这些应用程序来自 Spark 的不同组件，如 Spark Shell、Spark Submit 交互式批处理、Spark Streaming 实时流处理、Spark SQL 快速查询、MLlib 机器学习、GraphX 图处理等，如图 1-1 所示。

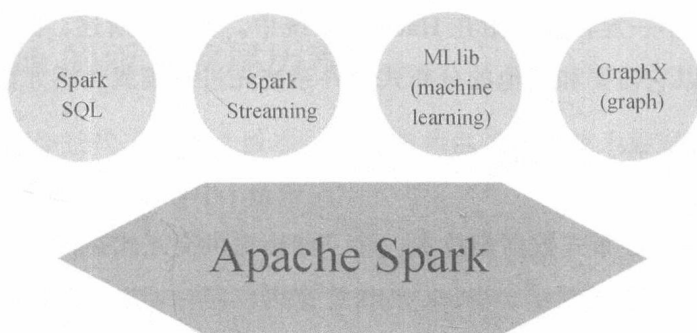


图 1-1 Spark 生态系统图

(1) Spark Core 提供 Spark 最基础与最核心的功能，它的子框架包括 Spark SQL、Spark Streaming、MLlib 和 GraphX。

(2) Spark Streaming 是 Spark API 核心的一个存在可达到超高通量的扩展，可以处理实时数据流的数据并进行容错。它可以从 Kafka、Flume、Twitter、ZeroMQ、Kinesis、TCP sockets 等数据源获取数据，并且可以使用复杂的算法和高级功能对数据进行处理。处理后的数据可以被推送到文件系统或数据库。

(3) Spark SQL 是一种结构化的数据处理模块。它提供了一个称为 Data Frame 的编程抽象，也可以作为分布式 SQL 查询引擎。

一个 DataFrame 相当于一个列数据的分布式采集组织，类似于一个关系型数据库中的一个表。它可以从多种方式构建，如结构化数据文件、Hive、外部数据库或分布式动态数据集 (RDD)。

(4) GraphX 在 Graphs 和 Graph-parallel 并行计算中是一个新的部分，GraphX 是 Spark 上的分布式图形处理架构，可用于图表计算。

1.3 Spark 的架构与原理

1.3.1 Spark 架构设计

Spark 架构主要包括客户端驱动程序 Driver App、集群管理器 Cluster Manager、工作节点 Worker 以及基本任务执行单元 Executor。

(1) Driver App 是客户端驱动程序，也可以理解为客户端应用程序，用于将任务程序转换为 RDD 和 DAG，并与 Cluster Manager 进行通信与调度。

(2) Cluster Manager 是 Spark 的集群管理器。它主要负责资源的分配与管理。集群管

理器分配的资源属于一级分配，它将各个 Worker 上的内存、CPU 等资源分配给应用程序，但是并不分配 Executor 的资源。目前，Standalone、YARN、Mesos、EC2 等都可以作为 Spark 的集群管理器。

(3) Worker 是 Spark 的工作节点。对 Spark 应用程序来说，由集群管理器分配得到资源的 Worker 节点主要负责以下工作：创建 Executor，将资源和任务进一步分配给 Executor，然后同步资源信息给 Cluster Manager。

(4) Executor 是 Spark 任务的执行单元。它主要负责任务的执行以及与 Worker、Driver App 的信息同步。

Spark 架构设计图如图 1-2 所示。

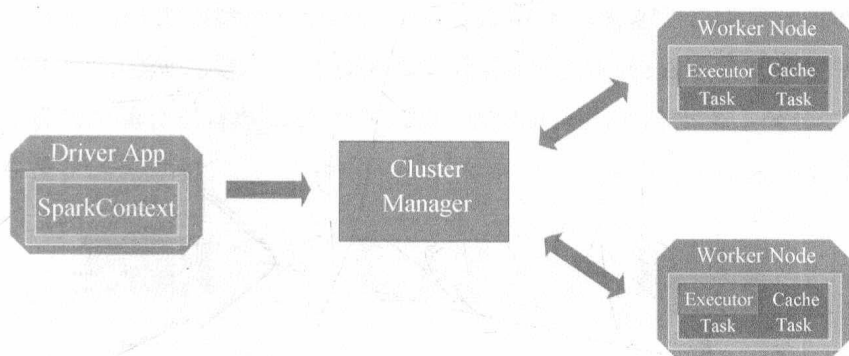


图 1-2 Spark 架构设计图

1.3.2 Spark 作业运行流程

Spark 作业流程图如图 1-3 所示。

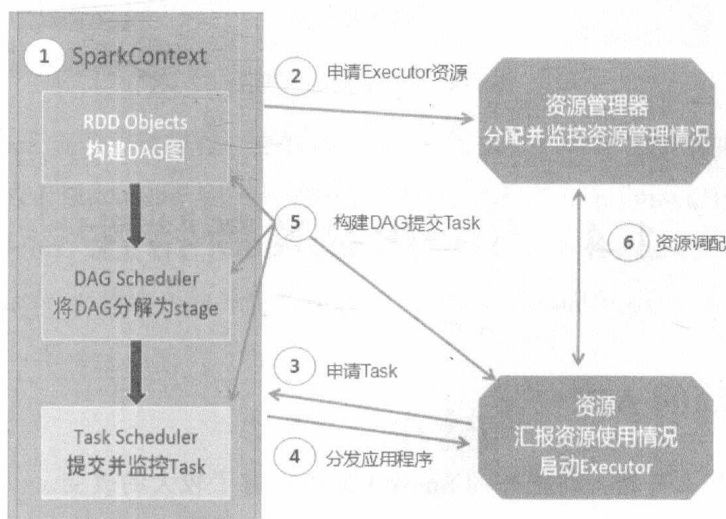


图 1-3 Spark 作业流程图

- (1) 构建 Spark Application 的运行环境, 启动 SparkContext。
- (2) SparkContext 向资源管理器申请运行 Executor 资源, 并启动 StandaloneExecutorbackend。
- (3) Executor 向 SparkContext 申请 Task。
- (4) SparkContext 将应用程序分发给 Executor。
- (5) SparkContext 构建 DAG 图, 将 DAG 图分解成 Stage, 将 Taskset 发送给 Task Scheduler, 由 Task Scheduler 将 Task 发送给 Executor 运行。
- (6) Task 在 Executor 上运行, 运行完释放所有资源。

1.3.3 Spark 分布式计算流程

Spark 分布式计算流程包含以下几个步骤: 首先分析应用的代码创建有向无环图, 然后将有向无环图划分为 Stage, 然后 Stage 生成作业 (job), 生成作业后由 FinalStage 提交任务集, 提交任务的工作交给 TaskSets 完成, 然后每个 Task 执行所分配的任务, 最终 Results 跟踪结果。流程图如图 1-4 所示。



图 1-4 Spark 核心原理图

1.4 Spark 2.X 新特性

1.4.1 精简的 API

从 Spark 2.0 版本开始, 与之前的 Spark 1.X 版本有了较大的改变。

- (1) 统一的 DataFrame 和 Dataset 接口。统一了 Scala 和 Java 的 DataFrame、Dataset

接口，在 R 和 Python 中由于缺乏安全类型，DataFrame 成为主要的程序接口。

(2) 新增 SparkSession 入口。SparkSession 替代原来的 SQLContext 和 HiveContext 作为 DataFrame 和 Dataset 的入口函数。SQLContext 和 HiveContext 保持向后兼容。

(3) 为 SparkSession 提供全新的工作流式配置。

(4) 更易用、更高效的计算接口。

(5) Dataset 中的聚合操作有全新的、改进的聚合接口。

1.4.2 Spark 作为编译器

Spark 2.0 搭载了第二代 Tungsten 引擎，该引擎是根据现代编译器与 MPP 数据库的理念来构建的，它将这些理念用于数据处理中，其主要思想就是在运行时使用优化后的字节码，将整体查询合成为单个函数，不再使用虚拟函数调用，而是利用 CPU 来注册中间数据。

为了有直观的感受，表 1-2 显示了 Spark 1.6 与 Spark 2.0 分别在一个核上处理一行的操作时间（单位：ns）。

表 1-2

Spark 1.6 与 Spark 2.0 操作时间对比图

单位：ns

原生的函数	Spark 1.6	Spark 2.0
filter	15	1.1
sum w/o group	14	0.9
sum w/ group	79	10.7
hash join	115	4.0
sort (8-bit entropy)	620	5.3
sort (64-bit entropy)	620	40
sort-merge join	750	700

1.4.3 智能化程度

为了实现 Spark 更快、更轻松、更智能的目标，Spark 2.X 在许多模块上都做了重要的更新，如 Structured Streaming 引入了低延迟的连续处理（Continuous Processing）、支持 Stream-to-stream Joins、通过改善 Pandas UDFs 的性能来提升 PySpark、支持第 4 种调度引擎 Kubernetes Clusters（其他 3 种分别是自带的独立模式 Standalone、YARN、Mesos）等。

1.5 小结

本章主要介绍了 Spark 定义、生态系统、架构原理和新特性等内容，从原理到应用由

深入浅出地介绍了 Spark，让读者从宏观和微观上对 Spark 有了认识 and 了解。

习 题

简答题

- (1) Spark 与 Hadoop 的区别是什么？
- (2) Spark 的应用场景有哪些？
- (3) 简述 Spark 的作业运行流程。
- (4) Spark 2.X 与 Spark 1.X 有什么不同点？