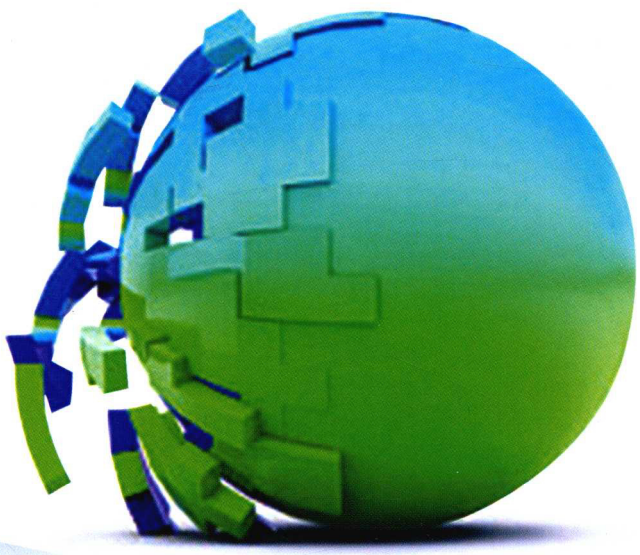
大数据专业应用型人才培养规划教材

湍翼教育

Spark 大数据技术与应用

韦德泉　许桂秋　主编



浙江科学技术出版社

大数据专业应用型人才培养规划教材

瑞翼教育

Spark 大数据技术与应用

韦德泉　许桂秋　主编

浙江科学技术出版社

图书在版编目(CIP)数据

Spark大数据技术与应用/韦德泉，许桂秋主编. —杭州：浙江科学技术出版社，2020.1
大数据专业应用型人才培养规划教材
ISBN 978-7-5341-8892-3

Ⅰ. ①S… Ⅱ. ①韦…②许… Ⅲ. ①数据处理软件—高等学校—教材 Ⅳ. ①TP274

中国版本图书馆CIP数据核字(2020)第002056号

主　　编　韦德泉　许桂秋
副 主 编　李世贵　黄　孝　潘晓洋　张　越
　　李　阳　张　军　王露露

丛 书 名　大数据专业应用型人才培养规划教材
书　　名　Spark大数据技术与应用
主　　编　韦德泉　许桂秋

出版发行　浙江科学技术出版社
杭州市体育场路347号　邮政编码：310006
办公室电话：0571－85176593
销售部电话：0571－85176040
网　址：www.zkpress.com
E-mail：zkpress@zkpress.com
排　　版　杭州大漠照排印刷有限公司
印　　刷　杭州广育多莉印刷有限公司
经　　销　全国各地新华书店

开　　本　787×1092　1/16　　印　张　10
字　　数　231 000
版　　次　2020年1月第1版　　印　次　2020年1月第1次印刷
书　　号　ISBN 978-7-5341-8892-3　　定　价　50.80元

责任编辑　张祝娟　　**文字编辑**　方　晴　　**责任校对**　顾旻波
责任美编　金　晖　　**责任印务**　崔文红

前 言

2015 年 9 月，国务院印发《促进大数据发展行动纲要》，确定了大数据发展的国家顶层设计，大数据与各行的结合已是未来发展的必然趋势。国家规划的大数据发展目标是：到 2020 年，技术先进、应用繁荣、保障有力的大数据产业体系基本形成。大数据相关产品和服务业务收入突破 1 万亿元，年均复合增长率保持在 30% 左右，加快建设数据强国，为实现制造强国和网络强国提供强大的产业支撑。为实现这一宏伟目标，大数据产业需要大量优质的技术人才。国内高校大力发展大数据教学、培训，势在必行！

在 Spark 大数据分析计算框架诞生之前，Hadoop 起着重要的作用，在实际的项目中担任数据分析和存储的角色，但在实时性、处理速度等方面均不具备优势。随着数据规模不断扩大，以及使用场景不断丰富，用户对于大数据处理系统的要求也越来越高。而 Hadoop 生态中的数据处理引擎 MapReduce，则越来越不能满足用户的需求。在这样的背景下，Spark 的出现打破了这种局面，Spark 特别擅长迭代式计算，相对 MapReduce 而言，其性能提升了上百倍。Spark 基于自身的核心 API，发展出适应大数据处理的多种场景生态组件，包括 SparkSQL、Spark Streaming、Spark GraphX、Spark MLlib 等，极大地满足了用户的需求。Spark 生态使构建端到端的大数据应用成为可能，在处理各种场景时，给用户提供统一的编程体验，可极大提高效率。

本书作为大数据分析与内存计算的入门教材，希望能够帮助读者打开大数据分析与应用的大门。全书采用理论结合实践的方式，循序渐进地介绍了 Spark 的运行原理，并引入综合性的实践案例，以引导读者运用所学知识解决现实中的问题。

全书共分为 10 章，第 1～8 章阐述了 Spark 核心组件的用法，以及使用相应组件进行大数据分析应用的方法，通过综合性实践案例将理论和实践相结合；第 9～10 章通过两个综合实例让读者从理论阶段上升到实际应用层面，可以帮助读者更深入理解 Spark 的使用方法。本书各章内容具体如下：

第 1 章主要介绍了 Spark 的发展历程、起源和基本的运行原理。

第 2 章介绍了 Spark 的环境搭建，从软件下载、解压到环境变量、相应参数的设置等方面讲解了如何在本地搭建 Spark 环境。

第3章主要介绍了 Python 编程语言、Java 编程语言、R 编程语言、Scala 编程语言、PySpark 的启动与日志的设置、PySpark 开发包的安装以及如何使用开发工具 PyCharm 编写 Spark 应用。

第4章主要介绍了 Spark RDD 弹性分布式数据集、共享变量、依赖关系等。

第5章主要介绍了 DataFrame 与 SparkSQL 的区别与联系，并对常用操作进行了阐释。

第6章介绍了 Spark Streaming 的特点、流数据加载、DStream 输出与转换操作以及 DataFrame 和 SQL 的互操作等。

第7章主要介绍了机器学习的相关概念、机器学习的一般流程、Spark 机器学习库 MLlib 和 ML。

第8章主要介绍了基于 Spark 的第三方开源图计算工具 GraphFrames，内容包括图的概念、图的操作和图算法，并用实际案例进行了应用操作。

第9章主要介绍了 Spark 综合实例 1：出租车数据分析。

第10章主要介绍了 Spark 综合实例 2：图书推荐系统。

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，书中难免存在疏漏和不足之处，恳请广大读者批评指正。

编著者

2019 年 12 月

目 录

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、发展及生态圈等方面对Spark进行介绍。

本章重点

◎Spark的定义。
◎Spark的生态系统。
◎Spark的架构与原理。
◎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)在美国加州大学伯克利分校的AMPLab实验室开发的子项目，经过开源后捐赠给了Apache软件基金会，最后成为现在众所周知的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且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)，应用开发者只要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，开发者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，其性能和生态系统也越来越好，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

时　间	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，增加对结构化流连续处理的支持
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架构，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，由其中的多个节点进行存储，这意味着用户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 的分布式数据存储功能，还提供了 MapReduce的数据处理功能。因此用户可以不使用 Spark，而选择使用 Hadoop 自身的 MapReduce对数据进行处理。

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。一些国内外的大公司，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采集、快速迭代处理，并进行综合分析，用来满足线上系统的分析要求。

(3) 业务推荐系统。Spark 将业务推荐系统按小时和日期级别的模型训练，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，可对上亿的请求量进行处理。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) 用户图计算系统。利用 Spark 图计算可解决许多生产问题，如基于度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。

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。

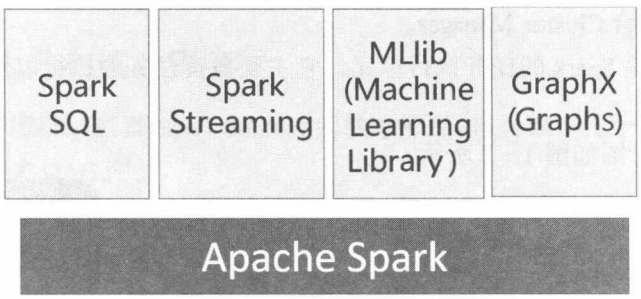


图 1－1　Spark 生态系统图

(1) Spark SQL 是一种结构化的数据处理模块。它提供了一个名为 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)。

(2) Spark Streaming 是 Spark API 核心的一个超高通量的扩展,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) MLlib(Machine Learning Library)是 Spark 提供的可扩展的机器学习库。MLlib 中已经包含了一些通用的学习算法和工具,如分类、回归、聚类、协同过滤、降维,以及底层的优化原语等算法和工具。

(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。集群管理器对资源的分配属于一级分配,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,所得资源的 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 的任务执行单元。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。

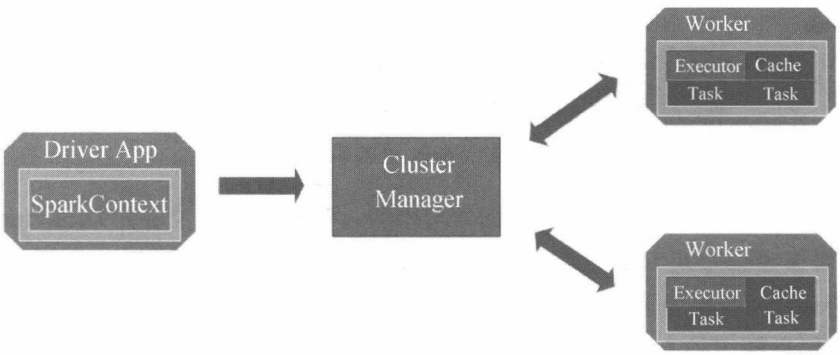


图 1-2 Spark 架构设计图

1.3.2　Spark 作业运行流程

Spark 作业流程图如图 1－3 所示。

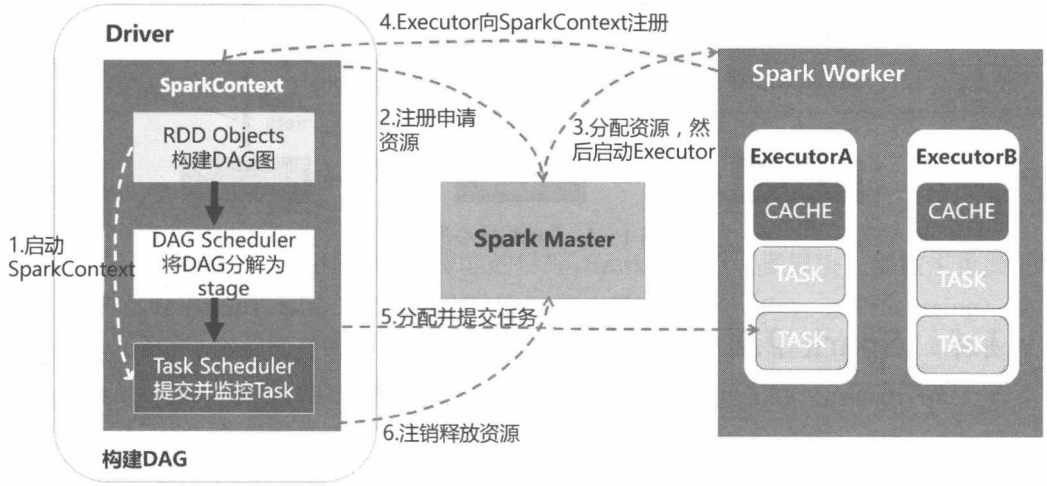


图 1－3　Spark 作业流程图

（1）构建 Spark Application 的运行环境，启动 SparkContext。

（2）SparkContext 向资源管理器 Spark Master（可以是 Standalone、Mesos、YARN）注册申请运行 Executor 资源。

（3）资源管理器分配 Executor 资源，并启动 Executor。并且 Executor 发送心跳给资源管理器。

（4）Executor 向 SparkContext 申请 Task。

（5）SparkContext 将应用程序分发给 Executor。具体包括构建 DAG 图，将 DAG 图分解成 Stage，将 Taskset 发送给 Task Scheduler，以及由 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，即 SparkContext 的核心原理，包含以下几个步骤。

（1）从代码构建 DAG 图。

（2）将 DAG 拆分为 Stage。

（3）Stage 生成作业。

（4）FinalStage 提交任务集。

（5）TaskSets 提交任务。

（6）Tasks 执行任务。

（7）Results 跟踪结果。

Spark 分布式计算流程如图 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 版本相比有了较大的改变。

（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 数据库的理念而构建的，它将这些理念用于数据处理，其主要思想就是在运行时，使用优化后的字节码，将整体查询转换为单个函数，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 分别在单核上处理函数的操作时间对比。

表 1-2 Spark 1.6 与 Spark 2.0 操作时间对比 （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

续表

原生的函数	Spark 1.6	Spark 2.0
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，如在 Structured Streaming 中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 种分别是自带的模式 Standalone、YARN、Mesos）等。

本章小结

本章主要介绍了 Spark 的定义、生态系统、架构原理和新特性等内容，从原理到应用由浅入深地介绍了 Spark，让读者从宏观到微观上对 Spark 有了更深的认识和了解。

本章习题

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？

第2章 Spark的环境搭建

上一章主要对 Spark 的定义、生态系统、架构原理和新特性等方面进行了介绍，让读者对 Spark 有了一个整体的认识。本章主要对 Spark 的环境搭建过程进行介绍，让读者了解基本的配置过程和操作命令。

本章重点

◎Spark 相关依赖软件的下载。
◎Spark 环境配置。
◎Spark 集群的启动与关闭。
◎Spark 应用的提交和 Web 页面的监控。

2.1 环境搭建前的准备

Spark 使用 Scala 语言进行开发，Scala 运行在 Java 平台之上，因此需要下载并安装 JDK 和 Scala。值得注意的是，Scala、Java 和 Spark 三者之间是有版本搭配限制的，可以根据官方文档提供的组合进行下载，否则会启动异常。具体的版本对应关系可在官网相关文档中看到，如图 2-1 所示，本书使用的环境组合是 Spark 2.3.0+Java 8+Scala 2.11。

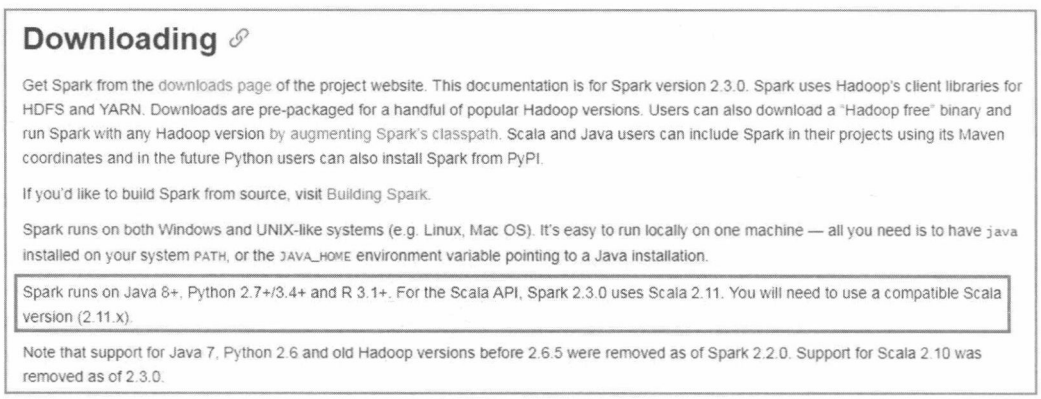


图 2-1 Spark、Java、Scala 的版本对应关系

Spark 运行在 Linux 操作系统下，因此在进行环境搭建之前需要一个 Linux 环境，可以是物理机，也可以是虚拟机。具体的操作系统安装此处不做重点讲解，此处采用 Linux 的一个发行版 Ubuntu 作为演示系统。Ubuntu 可以在其官网上直接下载。

1. 下载 Spark

打开浏览器进入 Spark 的官网，如图 2－2 所示。本书所用的 Spark 版本为 2.3.0。

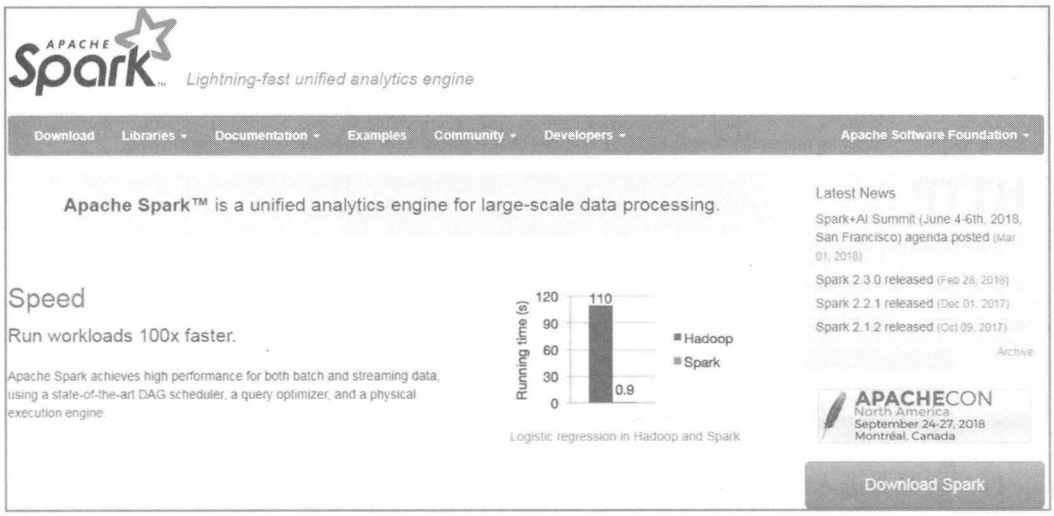


图 2－2　Spark 官网

单击“Download Spark”按钮进入下载页面，如图 2－3 所示。

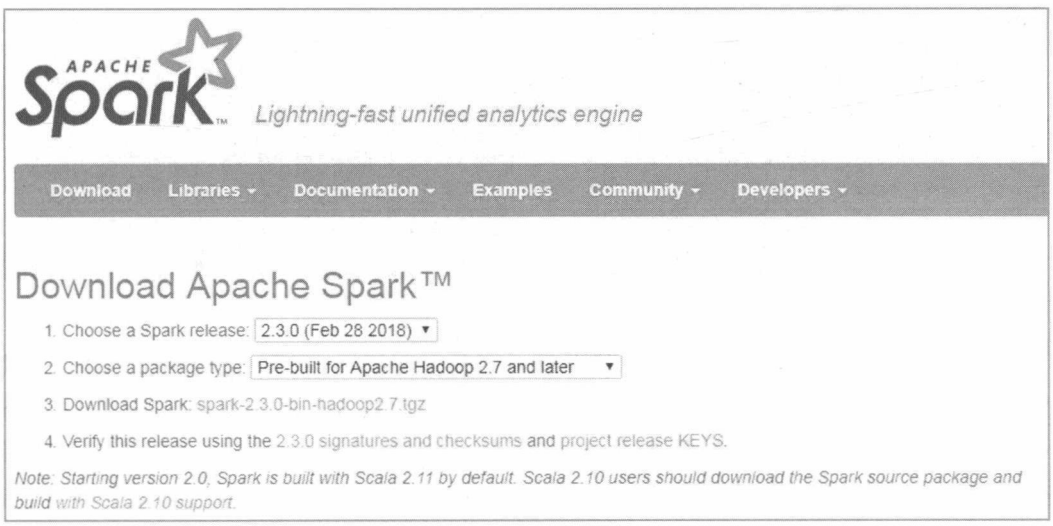


图 2－3　选择下载版本

在“Choose a Spark release”的下拉框中可以选择历史版本。“Choose a package type”的下拉框中可以选择集成 Hadoop 的版本，也可以选择源码进行编译。选择后单击“Download Spark”后面的文件名称即可进入下载页面，如图 2－4 所示。

可供选择的镜像地址有很多，选择网站推荐的镜像地址下载即可。