大数据科学丛书
BIG DATA SCIENCE

Spark

原理、机制及应用

刘驰 主编　符积高　徐闻春 编著

机械工业出版社
CHINA MACHINE PRESS

大数据科学丛书
BIG DATA SCIENCE

Spark

原理、机制及应用

刘驰 主编 符积高 徐闻春 编著

机械工业出版社
CHINA MACHINE PRESS

本书以 Spark 1.4 为基础，详细介绍了 Spark 技术概况、内部机制和应用情况。作者结合国内外众多资料和项目经验，力求深入浅出地讲解 Spark 技术的生态应用和发展状况，选取了 Spark Summit 中的典型案例进行解析，为读者全国展现 Spark 技术在业界的应用情况。

本书适合 Spark 技术初学者、Spark 技术爱好者、Spark 运维工程师和开源软件爱好者，也可以作为相关培训学校和大专院校相关专业的教学用书。

图书在版编目（CIP）数据

Spark：原理、机制及应用 / 刘驰主编；符积高，徐闻春编著.
—北京：机械工业出版社，2016.2
（大数据科学丛书）
ISBN 978-7-111-52928-6

Ⅰ. ①S… Ⅱ. ①刘… ②符… ③徐… Ⅲ. ①数据处理软件 Ⅳ. ①TP274

中国版本图书馆 CIP 数据核字（2016）第 026520 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：张 恒 责任编辑：张 恒
责任印制：乔 宇 责任校对：张艳霞
保定市中画美凯印刷有限公司印刷
2016 年 3 月第 1 版 • 第 1 次印刷
184mm × 260mm • 17 印张 • 1 插页 • 421 千字
0001 － 3500 册
标准书号：ISBN 978-7-111-52928-6
定价：49.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
服务咨询热线：（010）88361066
读者购书热线：（010）68326294
（010）88379203

网络服务
机 工 官 网：www.cmpbook.com
机 工 官 博：weibo.com/cmp1952
教育服务网：www.cmpedu.com
金 书 网：www.golden-book.com

前　言

随着互联网与移动终端行业的迅猛发展，企业和个体对数据相关服务需求不断提升，以Apache Hadoop 为代表的分布式并行计算技术进一步发展，数据由量变而引发的质变正在全球范围内掀起深刻的技术与商业变革。在产业界，以数据驱动的发展策略也已逐渐被提升到前所未有的高度。在金融、电信、房地产和众多传统领域，沉积的数据价值开始被重视，这些公司逐渐在大数据领域加强资金和研发投入。在学术界，国内外越来越多的高校和研究机构在云计算和大数据领域投入大量的人力研究大数据及其相关技术。不仅如此，我国政府提出的“中国制造 2025”战略规划和“互联网+”的概念也与大数据技术有着密不可分的联系，这更预示了大数据技术未来广阔的发展前景。

大数据的处理主要依靠分布式并行处理技术。本书主要介绍大数据分析平台的后起之秀 Apache Spark。相对于人们近年来熟知的 Apache Hadoop，Apache Spark 具有基于内存计算、适合迭代计算并兼容多应用场景的特点，同时它还能兼容 Hadoop 生态系统中的组件，能吸收 Hadoop 的优点。经过短短 6 年的飞跃式发展，Spark 已经成为业内颇具发展潜力的大数据分析平台之一。近两年召开的 Spark Summit 峰会，年均参会人数近 2000 人，业内对 Spark 的研究热情进一步提升，Spark 的应用领域也在不断扩展，包括医疗、金融、O2O 电商、政府、教育、电信、智慧城市和安全等，且在诸多领域都已经有 Spark 的成功应用案例。

编者基于国内外的研究和企业项目实践的经验，基于截稿时最新的 Spark 1.4 版来介绍 Spark 技术的应用实践和最新动向，让读者更容易地迈上 Spark 学习之路。

本书是国内（包括 Github 社区）较新的基于 Spark 1.4 版本的技术书籍，涵盖 Spark 技术的环境搭建、RDD 实操应用、内部机制、调优和企业应用等内容，具体如下。

1）基于 IntelliJ IDEA 的运行、开发和编译环境的详细搭建过程。

2）详细介绍 Spark 技术基础概念和应用实践。

3）基于 Spark 1.4 官方文档对 Spark 四大应用框架进行解读。

4）基于最新源码深入剖析 Spark 的资源调度、任务调度和 shuffle 过程。

5）深入解读近两年 Spark 峰会和国内企业分享的典型应用案例。

本书的编写系统完整，力争以通俗易懂的语言全方位精细解读 Spark 技术，本书主要针对大数据技术初学者，包括但不限于大学生、研究生和工程师。此外，Spark 应用开发人员、运维工程师和开源软件爱好者也可以将本书作为参考用书。

本书共分为概念、开发、机制和应用四篇，概念篇介绍 Spark 的背景概念和环境配置方法，开发篇介绍了 Spark 核心开发、四大应用框架和调优策略，机制篇则对 Spark 的 RDD、调度和 shuffle 等机制进行解读，应用篇针对 Spark 在业界的典型应用进行阐述。

对于初学者，建议先学习 Scala 语言的基本语法，并从第 1 章起顺序阅读，搭建好开发环境，边学边进行代码实践。

对于已经有一定基础的读者可以跳过概念篇直接从第 3 章开始阅读，学习完第二篇开发篇，即 Spark 的应用操作后可以通过接着学习第三篇机制篇来加深理解。第四篇比较独立，在学习完概念篇之后就可以进行学习。

本书由刘驰主编，参与编写人员有符积高、徐闻春。在本书的编写过程中，始终本着科学、严谨的态度，力求精益求精，但错误、疏漏之处在所难免，敬请广大读者批评指正。

编　著
lincbit@gmail.com

目　录

第三篇 机 制 篇

第四篇 应 用 篇

第一篇 概 念 篇

概念篇主要介绍 Spark 相关背景知识，包括 Spark 概述和运行开发环境。其中 Spark 概述部分介绍了 Spark 平台的发展背景及运行现状，带领读者认识大数据分析领域中最活跃的技术平台 Spark，随后详细讲解了 Spark 运行环境、开发环境和源码编译及阅读环境的搭建过程，让读者对 Spark 有一个初步了解。本篇目的是带领读者了解大数据及 Spark 平台的背景，搭建自己的开发环境，为后续进一步的学习打好基础。

第1章 Spark概述

随着以互联网为代表的信息技术的深度发展，积累产生了TB、PB甚至EB级别的数据量，由于传统机器的软硬件不足以支撑如此庞大数据的存储、管理及分析能力，因而大数据的分布式处理技术应运而生。如今大数据处理的主流平台为Hadoop和Spark，本书主要介绍大数据平台Spark。

本章主要介绍Spark的基础概念、发展历程、特点、与现有主流分布式应用框架的区别及其生态系统中的重要组成部分（如Spark SQL、Spark Streaming、GraphX和MLlib等子项目），目的在于让读者对分布式框架的背景及主流应用有一个宏观而全面的了解。

1.1 Spark初见

Spark是一种基于内存的开源计算框架，2009年诞生于美国加州大学伯克利分校AMPLab，它最初属于伯克利大学的研究性项目，后来在2010年正式开源，并于2013年成为了Apache基金项目，到2014年便成为Apache基金的顶级项目，该项目只经过了短短几年的发展时间，但其发展速度非常惊人。

正由于Spark来自于大学，其整个发展过程都充满了学术研究的标记，是学术带动了Spark核心架构的发展，如弹性分布式数据集（Resilient Distributed Datasets，RDD）、流处理（Spark streaming）、机器学习（MLlib）、SQL分析（Spark SQL）和图计算（GraphX），本节将主要介绍Spark的发展历程和特点。

1.1.1 Spark的发展史及近况

Spark从创立至今，成为大数据领域风尖浪口的热门项目只花了六年左右的时间，其具体发展大事记如下。

- 2009年，Spark诞生于伯克利分校AMPLab。
- 2010年，项目开源，很多早期关于Spark系统思想的论文发表。
- 项目开源之后，在GitHub上成立了Spark开发社区并在2013年成为Apache孵化项目。
- 该项目在2014年2月成为Apache顶级项目。
- 2014年5月30日，Spark 1.0.0版正式上线。

Spark项目组核心成员在2013年创建了Databricks公司，到目前为止已经在San Francisco连续举办了从2013年到2015年的Spark Summit峰会。会议得到大数据主流厂商Hortonworks、IBM、cloudera、MAPR和Pivotal等公司的支持和大数据方案解决商Amazon、DATASTAX和SAP等公司的合作。Spark的用户和应用情况如图1-1[⊖]所示。

⊖ 图1-1引用自https://spark-summit.org/2015/。

图 1-1　截止 2015 年 Spark 的主要用户和应用

从图 1-1 中可以看出，Spark 的影响力在 2014 年（可参考 2014 年 Spark 峰会资料）的基础上不断扩大，已经有越来越多的 Spark 用户使用该平台，其中包括传统工业厂商 TOYOTA 和著名 O2O 公司 Uber 与 Airbnb，说明 Spark 的用户领域不断深化到传统工业界和互联网与传统行业交叉的领域。不仅如此，越来越多的大数据商业版发行商，例如 Cloudera 以及 Hortonworks 也开始将 Spark 纳入其部署范围，这无疑对 Spark 的商业应用起到巨大的推广推动作用，也显示了 Spark 平台技术的先进性。

从 Spark 的版本演化速度看，说明了这个平台具有旺盛的生命力以及高的社区活跃度。尤其从 2013 年以来，Spark 进入了一个高速发展期，代码库提交量与社区活跃度都有显著增长。以活跃度来说，Spark 在所有 Apache 基金会开源项目中位列前三。相较于其他大数据平台或框架而言，Spark 的代码库最为活跃，表现出强劲的发展势头，从图 1-2[⊖] 中可以看到。

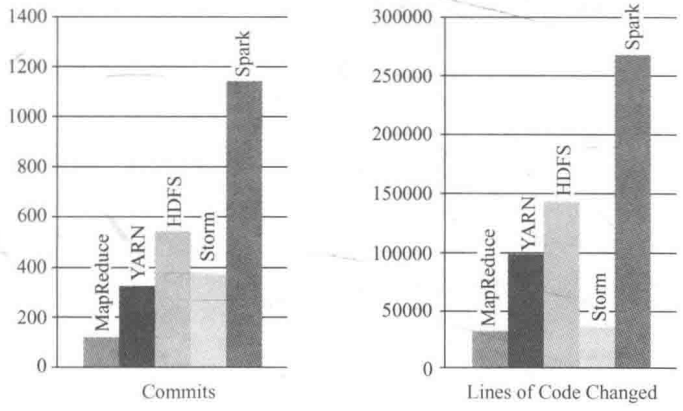
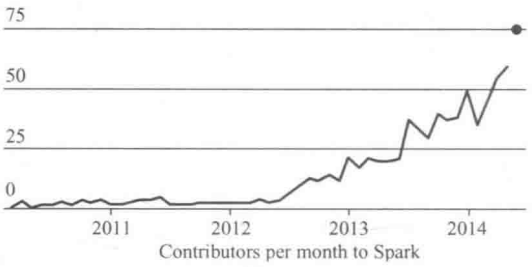


图 1-2　2014 年 6 月 30 日到 2014 年 12 月 30 日，6 个月中 Spark 代码活跃情况

从 2013 年 6 月到 2014 年 6 月，参与 Spark 代码贡献的开发人员从原来的 68 位增长到

⊖ 图 1-2 引用自 https://spark-summit.org/2014/。

255 位，截至 2015 年 6 月参与开发的人员已经达到 730 位[一]，参与贡献的公司逐渐有来自中国的阿里巴巴、百度、网易、腾讯和搜狐等公司。代码库的代码行也从 2014 年的 17 万行增长到 2015 年的 40 万行。图 1-3[二]为截至 2014 年 Spark 代码贡献者每月的增长曲线。



从

图 1-3　截至 2014 年 Spark 代码贡献者每月的增长曲线

从图 1-3 可以看出，Spark 从 2010 年到 2014 年间代码贡献者的数量不断增长，而且增长的速度越来越快。到 2015 年，每月的代码贡献者增长到现在的 135 位，在这些代码贡献者中出现了很多中国公司和开发者的身影。例如目前世界上最大的 Spark 集群在腾讯，拥有高达 8000 个节点；最大的单任务处理数据量达到 1PB，这项记录是由阿里巴巴公司和 databricks 公司共同持有。中国之所以能在这方面发展迅速，是因为中国市场巨大，信息产业的发展积累了更多数据，进而产生更为迫切的大数据处理需求，最后通过市场需求来推动技术发展。

除了影响力巨大的 Spark Summit 之外，Spark 社区还不定期地在全球各地召开小型的 Meetup[三]活动。其中在中国的北京、上海和深圳都有相应的 Spark 技术分享的 Meetup 活动。Spark Meetup Group 已经遍布北美、欧洲、亚洲和大洋洲。图 1-4[四]为 Spark Meetup Groups 在全球的分布图。

图 1-4　全球 Meetup Groups 分布情况

㊀ 引用自https://spark-summit.org/2015/。
㊁ 图 1-3 引用自 https://spark-summit.org/2014/。
㊂ Meetup 是一家知名的在线活动组织平台。
㊃ 图 1-4 引用自https://spark-summit.org/2014/。

从以上情况可以看出 Spark 的良好发展前景，国内外工业界和学术界都对其抱有极大兴趣。

1.1.2 Spark 的特点

Spark 之所以受关注，是因为其有与其他大数据平台不同的特点，主要如下。

1．轻量级快速处理

大数据处理中，速度往往被置于第一位，所以迫切需要能尽快处理数据的工具。Spark 允许传统 Hadoop 集群中的应用程序在内存中以 100 倍的速度运行，即使在磁盘上运行也能加快 10 倍。Spark 通过减少磁盘 I/O 来达到性能的提升，它们将中间的处理数据全部放到了内存中。Spark 使用 RDD 进行数据抽象，以允许在内存中存储数据，只在需要时才持久化到磁盘。这种做法大大地减少了数据处理过程中磁盘的读写，大幅度地降低了运行时间。

2．易于使用

Spark 支持多种语言，包括 Java、Scala、Python 及 R（Spark 1.4 版支持），这允许更多的开发者在自己熟悉的语言环境下进行工作，扩大了 Spark 的应用范围。它自带 80 多个高级操作符，允许在 shell 中进行交互式查询。多种使用模式的特点使应用更灵活。

3．支持复杂查询

除了简单的 map 及 reduce 操作之外，Spark 还支持 filter、foreach、reduceByKey、aggregate 以及 SQL 查询、流式查询等复杂查询。Spark 更为强大之处是用户可以在同一个工作流中无缝的搭配这些功能，例如 Spark 可以通过 Spark Streaming（1.2.2 小节有详细介绍）获取流数据，然后对流数据进行实时 SQL 查询或使用 MLlib 库进行系统推荐，而且这些业务的集成并不复杂，因为它们都基于 RDD 这一抽象数据集在不同业务过程中进行转换，转换代价小，体现了统一引擎解决不同类型工作场景的特点。

有关 Streaming 技术以及 MLlib 库和 RDD 将会在之后的章节详述。

4．实时的流处理（Spark Streaming）

对比 MapReduce 只能处理离线数据，Spark 还能支持实时流计算。Spark Streaming 主要用来对数据进行实时处理，当然在 YARN[⊖]之后 Hadoop 也可以借助其他的工具进行流式计算。对于 Spark Streaming，著名的大数据产品开发公司 Cloudera 的评价如下。

1）简单、轻量且具备功能强大的 API，Spark Streaming 允许开发者快速开发流应用程序。

2）容错能力强，对比其他的流解决方案，比如使用 Storm 需要额外的配置，而 Spark 无须额外的代码和配置，直接使用其上层应用框架 Spark Streaming 就可以做大量的恢复和交付工作，让 Spark 的流计算更加适应不同的需求。

3）集成性好，为流处理和批处理重用了同样的代码，甚至可以将流数据保存到历史数据中（如 HDFS）。

5．与已存 Hadoop 数据整合

Spark 不仅可以独立的运行（使用 Standalone 模式），还可以运行在当下的 YARN 管理集群中。Spark 可以读取已有的 Hadoop 数据，这是一个非常大的优势，也可以运行在任何

⊖ YARN：是一种新的 Hadoop 资源管理器，它是一个通用资源管理系统。

Hadoop 数据源上，比如 HBase、HDFS 等。这个特性可以让用户可以轻易地迁移已有的 Hadoop 应用。

6．活跃的社区

Spark 起源于 2009 年，当下已有超过 50 个机构 730 个工程师贡献过代码。与 2014 年 6 月相比，2015 年 Spark 代码行数增加了近三倍[⊖]，这是个惊人的增长。

1.1.3 Spark 的作用

为什么现阶段 Spark 被如此众多的公司应用呢？从需求角度来看，随着信息行业数据量的不断积累，传统单机因其本身软硬件限制而无法处理，所以很需要能对大量数据进行存储和分析处理的系统，另一方面，大型互联网公司因为业务数据量增长非常快，对大数据处理技术的高效实时性要求越来越高，迫切的需求促进了数据存储和计算分析系统技术的发展。Spark 就是在这样一个需求导向的背景下产生，其设计的目的就是能快速处理多种场景下的大数据问题，高效挖掘大数据中的价值，从而为业务发展提供决策支持。

目前 Spark 已经在电商、电信、视频娱乐、零售、商业分析和金融等领域有广泛应用，在本书第四部分的应用篇能看到 Spark 在这些领域的应用。

1.1.4 Spark 的体系结构

Spark 的体系结构如图 1-5 所示，不同于 Hadoop 的 MapReduce 和 HDFS，Spark 主要包括 Spark Core 和在 Spark Core 基础之上建立的应用框架 Spark SQL、Spark Streaming、MLlib 和 GraphX。

Core 库中主要包括上下文 Spark Context、抽象数据集 RDD、调度器 Scheduler、洗牌 Shuffle 和序列化器 Serializer 等。Spark 系统中的计算、I/O、调度和 shuffle 等系统基本功能都在其中。

在 Core 库之上，根据业务需求分为用于交互式查询的 SQL、实时流处理 Streaming、机器学习 Mllib 和图计算 GraphX 四大框架，除此之外还有一些其他实验性项目，如 Tachyon、BlinkDB 和 Tungsten 等，这些项目共同组成 Spark 体系结构。当然 Hadoop 中的存储系统 HDFS 迄今仍是不可被替代，一直被各分布式系统所使用，它也是 Spark 应用主要的持久化存储系统。在 1.3 节和第 4 章可以更全面地学习这四大应用框架的内容。

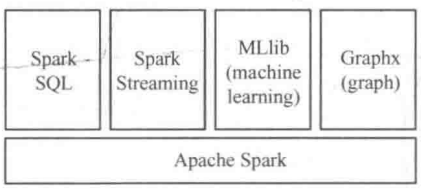


图 1-5 Spark 体系结构

1.1.5 Spark 的发展趋势

不论国内外，信息技术都不断地被企业和政府所重视，从德国的“工业 4.0”到美国的

⊖ 引用自https://spark-summit.org/2015/。

“工业互联网”战略规划，再到中国的“中国制造2025”和“互联网+”，这其中无不体现政府对云计算、物联网和大数据技术与传统工业深度融合、协同发展的期待，而中国本身是制造业大国，更需要先进的信息技术对接来提升工业制造水平以满足客户越来越多的个性化需求。

Spark 平台技术本身也正被医疗、金融、电信、电商和政府等越来越多的领域所使用，相信在未来以大数据技术为代表的 Spark 平台以其优良的设计理念加上其社区蓬勃的发展态势，极有可能在未来 5 到 10 年内成为大数据处理平台的事实标准。

1.2 Spark 框架

在讲述了关于 Spark 的背景、特点等内容后，还有一点值得提出的就是引起大数据处理技术迅猛发展的技术平台应该是始于 2004 年的 Hadoop，然而经过十多年的发展，以 Hadoop 为代表的大数据处理技术因其当初设计的缺陷，已经不能满足当前对实时性及迭代运算的需求，一批新处理框架如雨后春笋般出现，如流处理框架 Storm、Samza、Spark Streaming 和即席查询框架 Spark SQL。本节将对它们进行对比，让读者对新老框架的设计目的、应用及趋势有一定了解。

1.2.1 批处理框架

这里主要对比 Hadoop 与 Spark 在批处理方面的区别，它们在设计目的、计算模型和适用场景等方面进行对比，如表 1-1 所示。

表 1-1 Hadoop 与 Spark 的区别

项　目	Hadoop	Spark
起源时间	2004 年	2009 年
设计目的	使用分布式算法能处理大规模数据	克服 MapReduce 模型缺陷，能在多场景处理大规模数据
计算模型	MapReduce 计算模型	基于内存的抽象数据类型 RDD
主要支持的语言	Java	Scala
处理效率	低	是 Hadoop 处理速度的几十倍
缺点	计算模型单一、效率低	对内存容量要求高，成本较高
适用场景	非迭代批处理	批处理、迭代计算模型
稳定性	高	一般
通用性	较弱，需要与不同框架结合适用	兼容 Spark SQL、Spark Streaming、MLlib 和 GraphX

从表 1-1 中可以看出，发展十余年的 Hadoop 解决了处理大数据的问题，但因其设计之初没有考虑到效率，导致在面对迭代计算问题时效率很低，主要原因归结于其 MapReduce 计算模型太单一且计算过程中的 Shuffle 过程对本地硬盘的 I/O 消耗太大，不能适应复杂需求。不仅如此，当 Hadoop 要面对 SQL 交互式查询场景、实时流处理场景以及机器学习场景时力不从心，不得不与第三方应用框架相结合，从而导致不同类型业务（如流处理和 SQL 交互查询）在衔接过程中因涉及不同的数据格式，故而在共享和转换数据过程中消耗大量资源。

众所周知，内存计算速度比硬盘要快几个数量级，Spark 作为基于内存计算大数据处理

平台以其高速、多场景适用的特点逐渐脱颖而出，体现了“一个堆栈来解决各种场景（One stack to rule all）”的宗旨。

1.2.2 流处理框架

Storm 是一个分布式的、容错的实时计算系统，它由 Twitter 公司开源，专门用于数据实时处理的平台。它与 Spark Streaming 功能相似，都是将流数据分成一个个小块的批（batch）数据进行数据处理，但也有很多不同，下面主要从处理延迟和容错两个方面进行分析。

1．处理模型，延迟

Storm 和 Spark Streaming 这两个框架都提供可扩展性和容错性，它们根本的区别在于处理模型。Storm 处理的是每次传入的一个事件，而 Spark Streaming 处理的是某个时间段窗口内的事件流。因此，Storm 处理一个事件可以达到秒内的延迟，而 Spark Streaming 则有秒级延迟。

2．容错、数据保证

在容错数据保证方面，Spark Streaming 提供了更好的支持容错状态计算。在 Storm 中，每个单独的记录当它通过系统时必须被跟踪，所以 Storm 能够至少保证每个记录将被处理一次，但是在从错误中恢复过来时允许出现重复记录。这意味着可变状态可能被错误地更新两次。

另一方面，Spark Streaming 只需要在批处理级别进行跟踪处理，因此即便一个节点发生故障，也可以有效地保证每个批处理过程的数据将完全被处理一次。实际上 Storm 的 Trident library 也提供了一次完全处理，但它依赖于事务更新状态，比较慢，通常必须由用户来实现。

对于本书所讲版本的 Spark Streaming 而言，最小选取的批数据量（Batch Size）的选取在 0.5～2s（Storm 最小的延迟是 100ms 左右），故 Spark Streaming 能够满足除对实时性要求非常高（如高频实时交易）之外的流式准实时（“准实时”表示数据延迟在秒级）计算场景。

简而言之，如果限于毫秒级的延迟，Storm 是一个不错的选择，而且没有数据丢失。如果需要有状态的计算，时延秒级，Spark Streaming 更好。Spark Streaming 的编程逻辑也可能更容易，因为它类似于批处理模型 MapReduce，此外因为 Spark 融合了多种业务模型，所以如果面对融合不同业务的场景，考虑到人工维护成本和数据转换成本，则 Spark 平台更能符合需求。

1.3 Spark 的生态系统

Spark 的设计目的是全栈式地解决批处理、结构化数据查询、流计算、图计算和机器学习业务场景，此外其通用性还体现在对存储层（如 HDFS、cassandra[㊀]）和资源管理层（Mesos[㊁]、YARN）的支持。Spark 生态系统如图 1-6 所示，在 Spark Core 的上层有支持 SQL 查询的子项目 Spark SQL、支持机器学习 MLlib 库、支持图计算的 GraphX 以及支持流计算

㊀ cassandra：最初由 Facebook 开发的分布式 NoSQL 数据库，于 2008 年开源。

㊁ Mesos：AMPLab 最初开发的一个集群管理器，提供了有效的、跨分布式应用或框架的资源隔离和共享。