



# 厦门大学研究生课程 《大数据处理技术Spark》

<http://dblab.xmu.edu.cn/post/7659/>

温馨提示：编辑幻灯片母版，可以修改每页PPT的厦大校徽和底部文字

## 第3章 Spark的设计与运行原理

(PPT版本号：2017年春季学期)

林子雨

厦门大学计算机科学系

E-mail: [ziyulin@xmu.edu.cn](mailto:ziyulin@xmu.edu.cn) ▶▶

主页: <http://www.cs.xmu.edu.cn/linziyu>



扫一扫访问班级主页





# 提纲

- 3.1 Spark概述
- 3.2 Spark生态系统
- 3.3 Spark运行架构
- 3.4 Spark的部署和应用方式



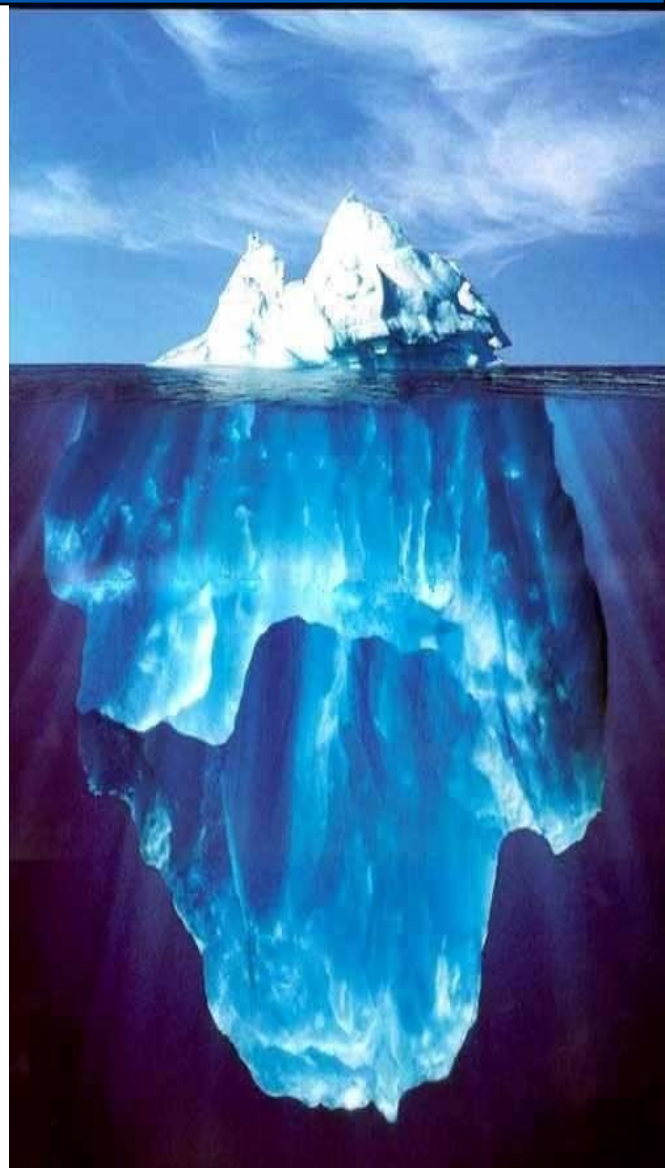
厦门大学林子雨



子雨大数据之Spark入门教程

披荆斩棘，在大数据丛林中开辟学习捷径

免费在线教程：<http://dblab.xmu.edu.cn/blog/spark/>





# 3.1 Spark概述

3.1.1 Spark简介

3.1.2 Scala简介

3.1.3 Spark与Hadoop的比较



## 3.1.1 Spark简介

- Spark最初由美国加州伯克利大学（UC Berkeley）的AMP实验室于2009年开发，是基于内存计算的大数据并行计算框架，可用于构建大型的、低延迟的数据分析应用程序
- 2013年Spark加入Apache孵化器项目后发展迅猛，如今已成为Apache软件基金会最重要的三大分布式计算系统开源项目之一（Hadoop、Spark、Storm）
- Spark在2014年打破了Hadoop保持的基准排序纪录
  - Spark/206个节点/23分钟/100TB数据
  - Hadoop/2000个节点/72分钟/100TB数据
  - Spark用十分之一的计算资源，获得了比Hadoop快3倍的速度



## 3.1.1 Spark简介

Spark具有如下几个主要特点：

- 运行速度快：使用DAG执行引擎以支持循环数据流与内存计算
- 容易使用：支持使用Scala、Java、Python和R语言进行编程，可以通过Spark Shell进行交互式编程
- 通用性：Spark提供了完整而强大的技术栈，包括SQL查询、流式计算、机器学习和图算法组件
- 运行模式多样：可运行于独立的集群模式中，可运行于Hadoop中，也可运行于Amazon EC2等云环境中，并且可以访问HDFS、Cassandra、HBase、Hive等多种数据源



## 3.1.1 Spark简介

Spark如今已吸引了国内外各大公司的注意，如腾讯、淘宝、百度、亚马逊等公司均不同程度地使用了Spark来构建大数据分析应用，并应用到实际的生产环境中

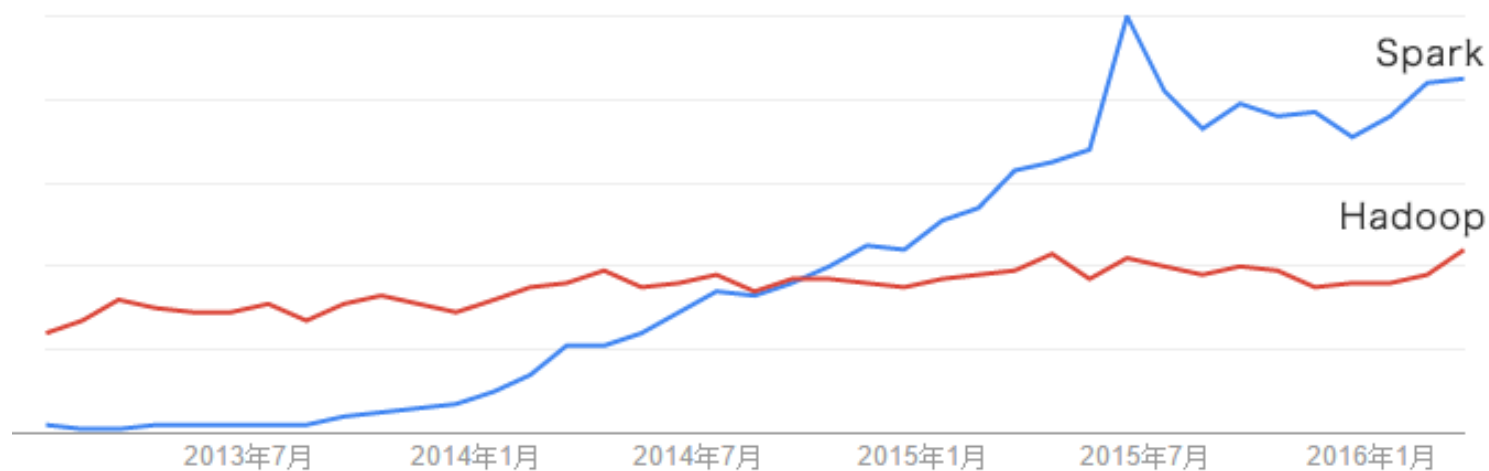


图16-1 谷歌趋势：Spark与Hadoop对比



## 3.1.2 Scala简介

Scala是一门现代的多范式编程语言，运行于Java平台（JVM，Java虚拟机），并兼容现有的Java程序

Scala的特性：

- Scala具备强大的并发性，支持函数式编程，可以更好地支持分布式系统
- Scala语法简洁，能提供优雅的API

Scala兼容Java，运行速度快，且能融合到Hadoop生态圈中

Scala是Spark的主要编程语言，但Spark还支持Java、Python、R作为编程语言

Scala的优势是提供了REPL（Read-Eval-Print Loop，交互式解释器），提高程序开发效率



## 3.1.3 Spark与Hadoop的对比

Hadoop存在如下一些缺点：

- 表达能力有限
- 磁盘IO开销大
- 延迟高
  - 任务之间的衔接涉及IO开销
  - 在前一个任务执行完成之前，其他任务就无法开始，难以胜任复杂、多阶段的计算任务





## 3.1.3 Spark与Hadoop的对比

Spark在借鉴Hadoop MapReduce优点的同时，很好地解决了MapReduce所面临的问题

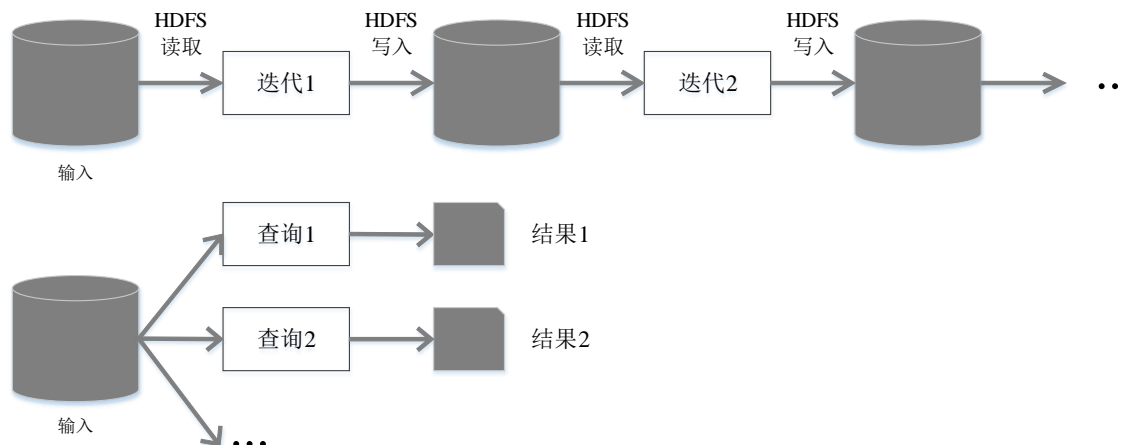
相比于Hadoop MapReduce，Spark主要具有如下优点：

- Spark的计算模式也属于MapReduce，但不局限于Map和Reduce操作，还提供了多种数据集操作类型，编程模型比Hadoop MapReduce更灵活
- Spark提供了内存计算，可将中间结果放到内存中，对于迭代运算效率更高

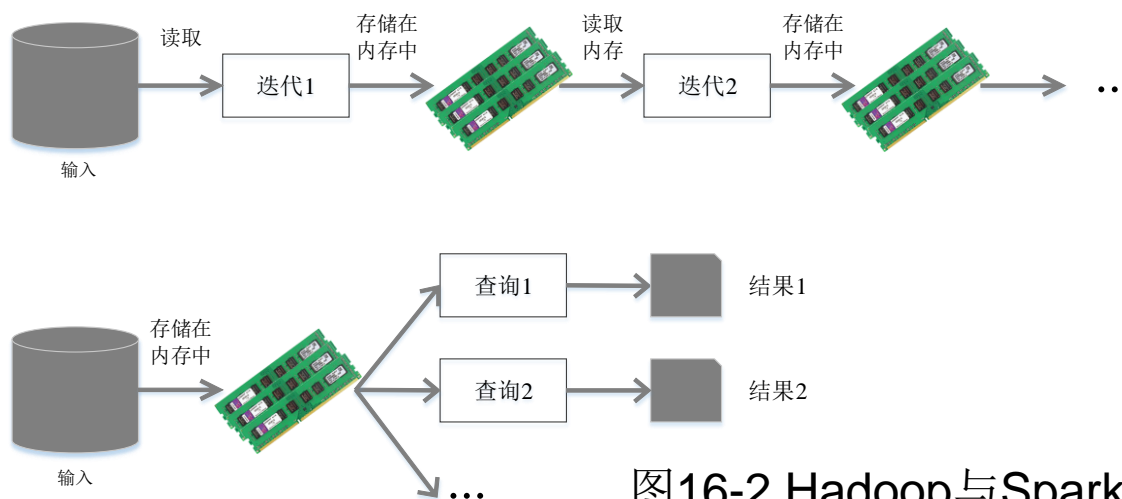
Spark基于DAG的任务调度执行机制，要优于Hadoop MapReduce的迭代执行机制



# 3.1.3 Spark与Hadoop的对比



(a) Hadoop MapReduce执行流程



(b) Spark执行流程

图16-2 Hadoop与Spark的执行流程对比



## 3.1.3 Spark与Hadoop的对比

- 使用Hadoop进行迭代计算非常耗资源
- Spark将数据载入内存后，之后的迭代计算都可以直接使用内存中的中间结果作运算，避免了从磁盘中频繁读取数据

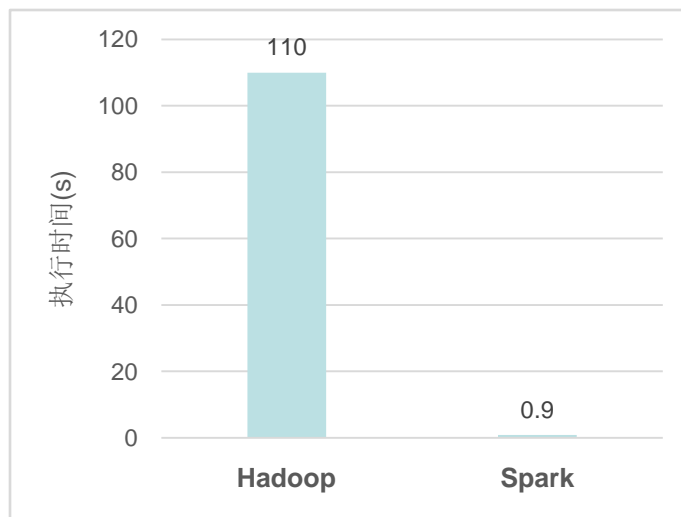


图16-3 Hadoop与Spark执行逻辑回归的时间对比



## 3.1.3 Spark与Hadoop的对比

MapReduce	Spark
数据存储结构：磁盘HDFS文件系统的split	使用内存构建弹性分布式数据集RDD 对数据进行运算和cache
编程范式：Map + Reduce	DAG: Transformation + Action
计算中间结果落到磁盘，IO及序列化、反序列化代价大	计算中间结果在内存中维护 存取速度比磁盘高几个数量级
Task以进程的方式维护，需要数秒时间才能启动任务	Task以线程的方式维护 对于小数据集读取能够达到亚秒级的延迟



## 3.2 Spark生态系统

在实际应用中，大数据处理主要包括以下三个类型：

- 复杂的批量数据处理：通常时间跨度在数十分钟到数小时之间
- 基于历史数据的交互式查询：通常时间跨度在数十秒到数分钟之间
- 基于实时数据流的数据处理：通常时间跨度在数百毫秒到数秒之间

当同时存在以上三种场景时，就需要同时部署三种不同的软件

- 比如: MapReduce / Impala / Storm

这样做难免会带来一些问题：

- 不同场景之间输入输出数据无法做到无缝共享，通常需要进行数据格式的转换
- 不同的软件需要不同的开发和维护团队，带来了较高的使用成本
- 比较难以对同一个集群中的各个系统进行统一的资源协调和分配



## 3.2 Spark生态系统

- Spark的设计遵循“一个软件栈满足不同应用场景”的理念，逐渐形成了一套完整的生态系统
- 既能够提供内存计算框架，也可以支持SQL即席查询、实时流式计算、机器学习和图计算等
- Spark可以部署在资源管理器YARN之上，提供一站式的大数据解决方案
- 因此，Spark所提供的生态系统足以应对上述三种场景，即同时支持批处理、交互式查询和流数据处理



## 3.2 Spark生态系统

Spark生态系统已经成为伯克利数据分析软件栈BDAS（Berkeley Data Analytics Stack）的重要组成部分

Access and Interfaces	Spark Streaming	BlinkDB Spark SQL	GraphX	MLBase MLlib
Processing Engine	Spark Core			
Storage	Tachyon HDFS, S3			
Resource Virtualization	Mesos		Hadoop Yarn	

图16-4 BDAS架构

Spark的生态系统主要包含了Spark Core、Spark SQL、Spark Streaming、MLlib和GraphX 等组件



## 3.2 Spark生态系统

表1 Spark生态系统组件的应用场景

应用场景	时间跨度	其他框架	Spark生态系统中的组件
复杂的批量数据处理	小时级	MapReduce、Hive	Spark
基于历史数据的交互式查询	分钟级、秒级	Impala、Dremel、Drill	Spark SQL
基于实时数据流的数据处理	毫秒、秒级	Storm、S4	Spark Streaming
基于历史数据的数据挖掘	-	Mahout	MLlib
图结构数据的处理	-	Pregel、Hama	GraphX





## 3.3 Spark运行架构

3.3.1 基本概念

3.3.2 架构设计

3.3.3 Spark运行基本流程

3.3.4 Spark运行原理



## 3.3.1 基本概念

- **RDD**: 是Resilient Distributed Dataset（弹性分布式数据集）的简称，是分布式内存的一个抽象概念，提供了一种高度受限的共享内存模型
- **DAG**: 是Directed Acyclic Graph（有向无环图）的简称，反映RDD之间的依赖关系
- **Executor**: 是运行在工作节点（WorkerNode）的一个进程，负责运行Task
- **Application**: 用户编写的Spark应用程序
- **Task**: 运行在Executor上的工作单元
- **Job**: 一个Job包含多个RDD及作用于相应RDD上的各种操作
- **Stage**: 是Job的基本调度单位，一个Job会分为多组Task，每组Task被称为Stage，或者也被称为TaskSet，代表了一组关联的、相互之间没有Shuffle依赖关系的任务组成的任务集



## 3.3.2 架构设计

- Spark运行架构包括集群资源管理器（Cluster Manager）、运行作业任务的工作节点（Worker Node）、每个应用的任务控制节点（Driver）和每个工作节点上负责具体任务的执行进程（Executor）
- 资源管理器可以自带或Mesos或YARN

与Hadoop MapReduce计算框架相比，Spark所采用的Executor有两个优点：

- 一是利用多线程来执行具体的任务，减少任务的启动开销
- 二是Executor中有一个BlockManager存储模块，会将内存和磁盘共同作为存储设备，有效减少IO开销

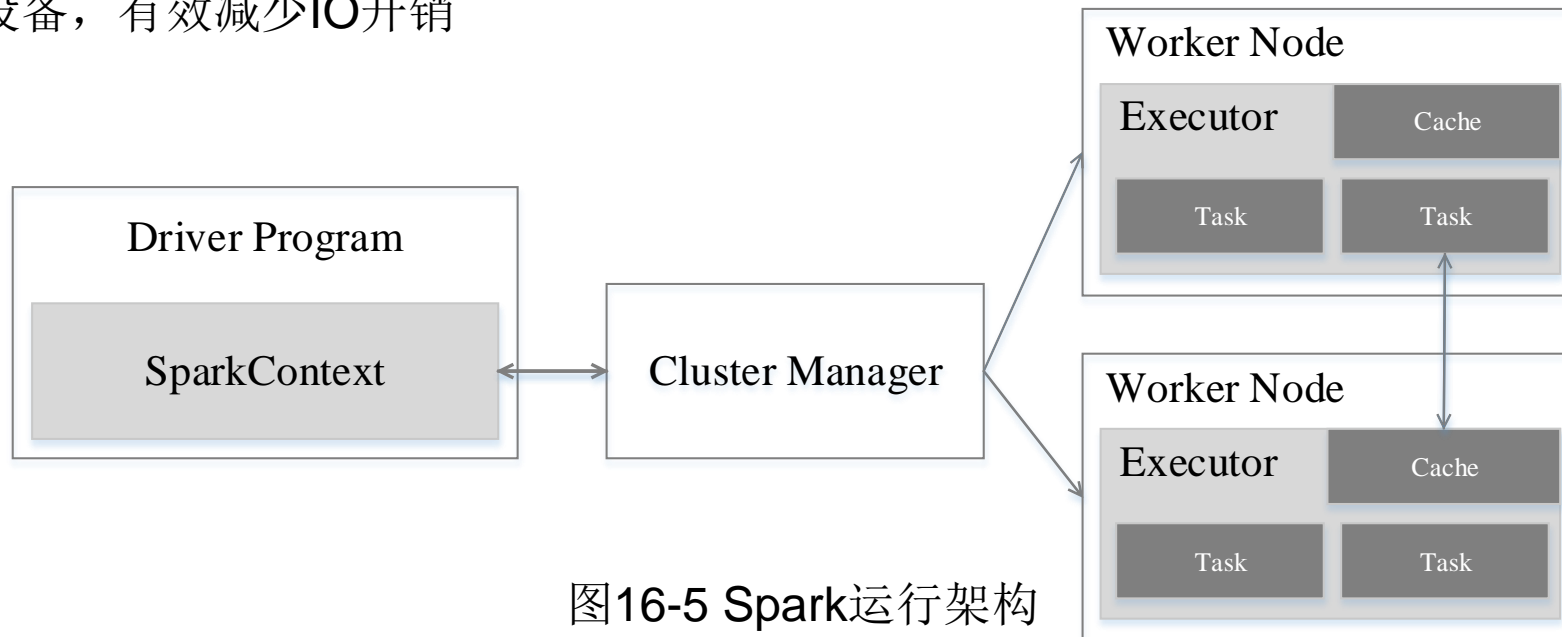


图16-5 Spark运行架构



## 3.3.2 架构设计

- 一个Application由一个Driver和若干个Job构成，一个Job由多个Stage构成，一个Stage由多个没有Shuffle关系的Task组成
- 当执行一个Application时，Driver会向集群管理器申请资源，启动Executor，并向Executor发送应用程序代码和文件，然后在Executor上执行Task，运行结束后，执行结果会返回给Driver，或者写到HDFS或者其他数据库中

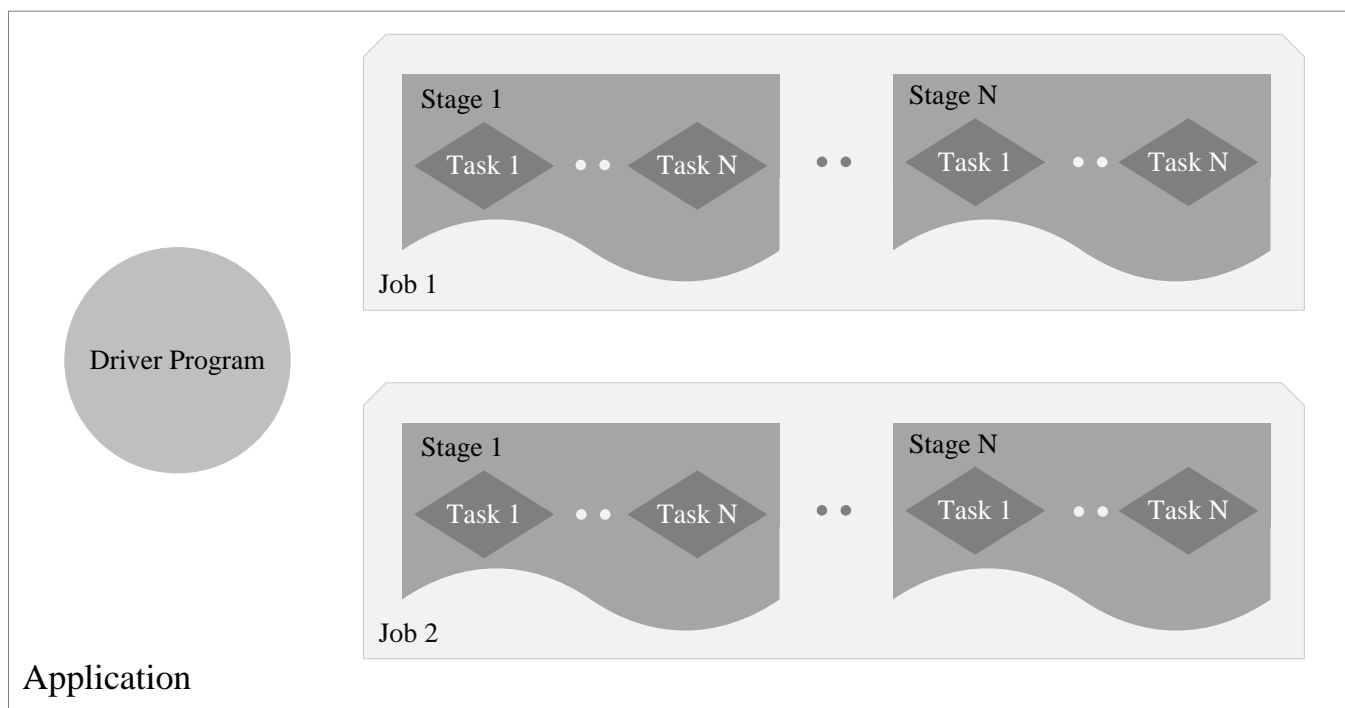


图16-6 Spark中各种概念之间的相互关系



# 3.3.3 Spark运行基本流程

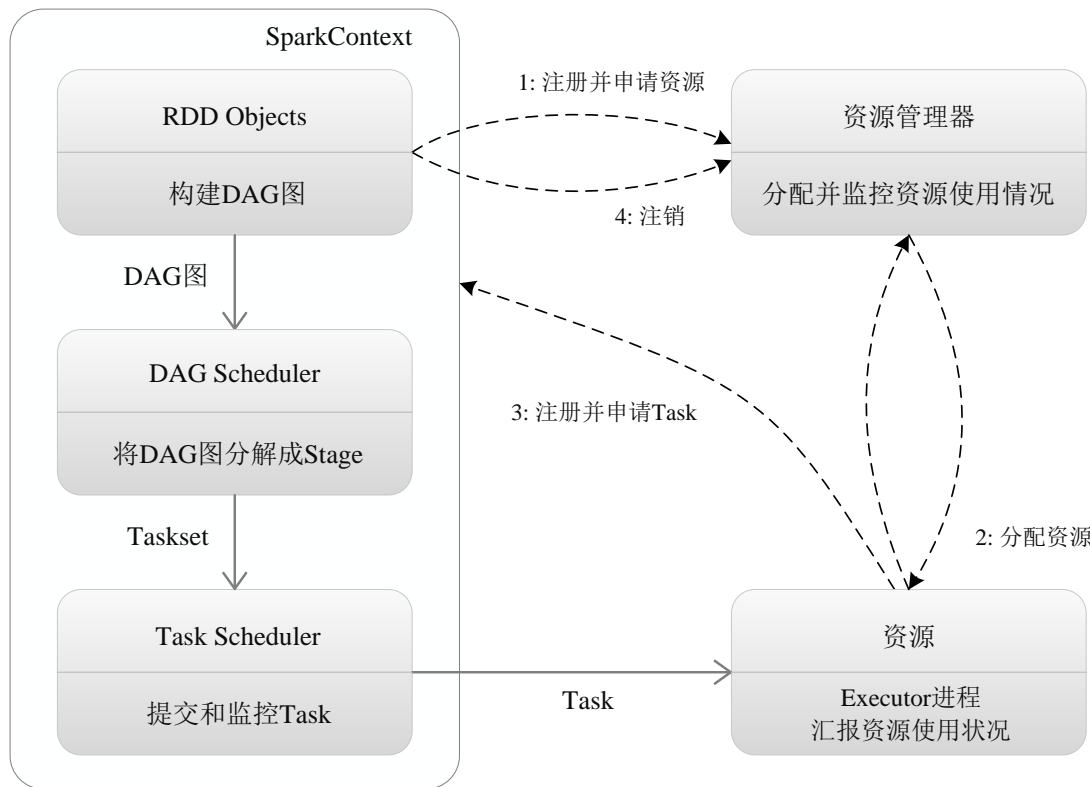


图16-7 Spark运行基本流程图

SparkContext对象代表了和一个集群的连接

(1) 首先为应用构建起基本的运行环境，即由Driver创建一个SparkContext，进行资源的申请、任务的分配和监控

(2) 资源管理器为Executor分配资源，并启动Executor进程

(3) SparkContext根据RDD的依赖关系构建DAG图，DAG图提交给DAGScheduler解析成Stage，然后把一个个TaskSet提交给底层调度器TaskScheduler处理；Executor向SparkContext申请Task，Task Scheduler将Task发放给Executor运行，并提供应用程序代码

(4) Task在Executor上运行，把执行结果反馈给TaskScheduler，然后反馈给DAGScheduler，运行完毕后写入数据并释放所有资源



## 3.3.3 Spark运行基本流程

总体而言，Spark运行架构具有以下特点：

(1) 每个Application都有自己专属的Executor进程，并且该进程在Application运行期间一直驻留。Executor进程以多线程的方式运行Task

(2) Spark运行过程与资源管理器无关，只要能够获取Executor进程并保持通信即可

(3) Task采用了数据本地性和推测执行等优化机制



## 3.3.4 RDD运行原理

- 1.设计背景
- 2.RDD概念
- 3.RDD特性
- 4.RDD之间的依赖关系
- 5.Stage的划分
- 6.RDD运行过程



## 3.3.4 RDD运行原理

### 1.设计背景

- 许多迭代式算法（比如机器学习、图算法等）和交互式数据挖掘工具，共同之处是，不同计算阶段之间会重用中间结果
- 目前的MapReduce框架都是把中间结果写入到HDFS中，带来了大量的数据复制、磁盘IO和序列化开销
- RDD就是为了满足这种需求而出现的，它提供了一个抽象的数据架构，我们不必担心底层数据的分布式特性，只需将具体的应用逻辑表达为一系列转换处理，不同RDD之间的转换操作形成依赖关系，可以实现管道化，避免中间数据存储





## 3.3.4 RDD运行原理

### 2.RDD概念

- 一个**RDD**就是一个分布式对象集合，本质上是一个只读的分  
区记录集合，每个**RDD**可分成多个分区，每个分区就是一个  
数据集片段，并且一个**RDD**的不同分区可以被保存到集群中  
不同的节点上，从而可以在集群中的不同节点上进行并行计  
算
- **RDD**提供了一种高度受限的共享内存模型，即**RDD**是只读  
的记录分区的集合，不能直接修改，只能基于稳定的物理存  
储中的数据集创建**RDD**，或者通过在其他**RDD**上执行确定的  
转换操作（如**map**、**join**和**group by**）而创建得到新的**RDD**



## 3.3.4 RDD运行原理

- RDD提供了一组丰富的操作以支持常见的数据运算，分为“动作”（Action）和“转换”（Transformation）两种类型
- RDD提供的转换接口都非常简单，都是类似map、filter、groupBy、join等粗粒度的数据转换操作，而不是针对某个数据项的细粒度修改（不适合网页爬虫）
- 表面上RDD的功能很受限、不够强大，实际上RDD已经被实践证明可以高效地表达许多框架的编程模型（比如MapReduce、SQL、Pregel）
- Spark用Scala语言实现了RDD的API，程序员可以通过调用API实现对RDD的各种操作



## 3.3.4 RDD运行原理

RDD典型的执行过程如下：

- RDD读入外部数据源进行创建
- RDD经过一系列的转换（Transformation）操作，每一次都会产生不同的RDD，供给下一个转换操作使用
- 最后一个RDD经过“动作”操作进行转换，并输出到外部数据源

这一系列处理称为一个**Lineage**（血缘关系），即**DAG**拓扑排序的结果  
优点：惰性调用、管道化、避免同步等待、不需要保存中间结果、每次操作变得简单

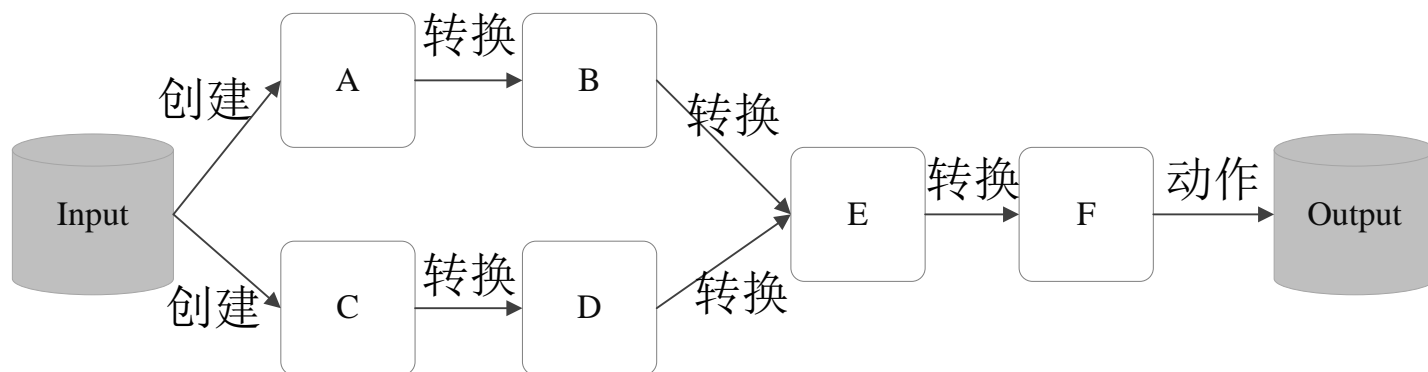


图16-8 RDD执行过程的一个实例



## 3.3.4 RDD运行原理

### 3.RDD特性

Spark采用RDD以后能够实现高效计算的原因主要在于：

(1) 高效的容错性

- 现有容错机制：数据复制或者记录日志
- **RDD**：血缘关系、重新计算丢失分区、无需回滚系统、重算过程在不同节点之间并行、只记录粗粒度的操作

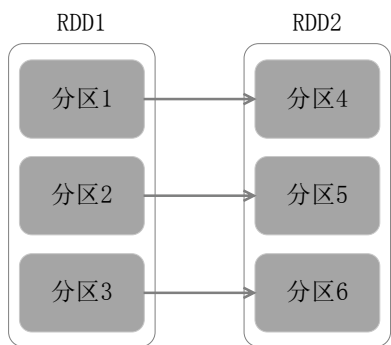
(2) 中间结果持久化到内存，数据在内存中的多个RDD操作之间进行传递，避免了不必要的读写磁盘开销

(3) 存放的数据可以是Java对象，避免了不必要的对象序列化和反序列化

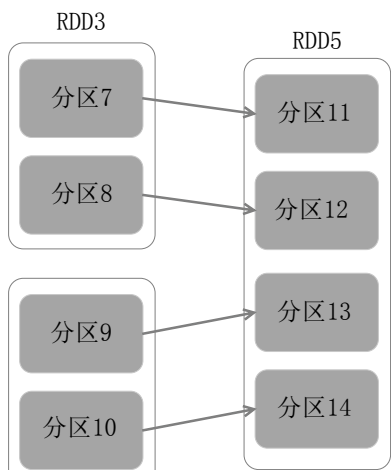


# 3.3.4 RDD运行原理

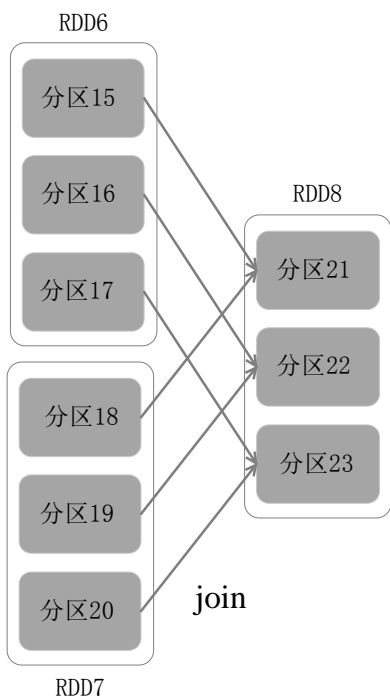
## 4. RDD之间的依赖关系



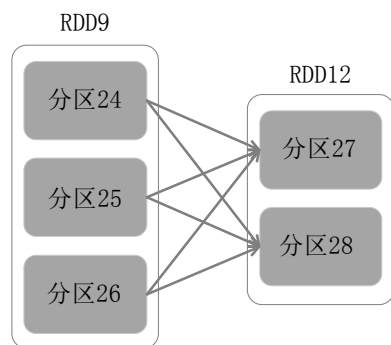
map, filter



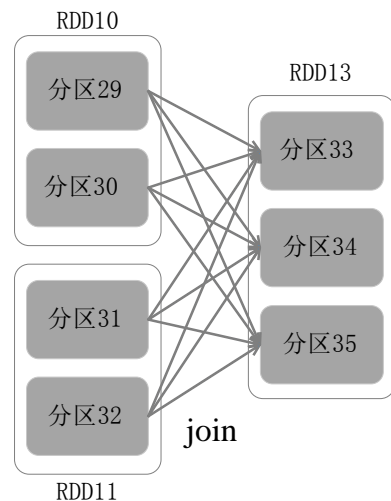
union



join



groupByKey



join

- 窄依赖表现为一个父RDD的分区对应于一个子RDD的分区或多个父RDD的分区对应于一个子RDD的分区
- 宽依赖则表现为存在一个父RDD的一个分区对应一个子RDD的多个分区

(a)窄依赖

(b)宽依赖

图16-9 窄依赖与宽依赖的区别



## 3.3.4 RDD运行原理

### 5.Stage的划分

Spark通过分析各个RDD的依赖关系生成了DAG，再通过分析各个RDD中的分区之间的依赖关系来决定如何划分Stage，具体划分方法是：

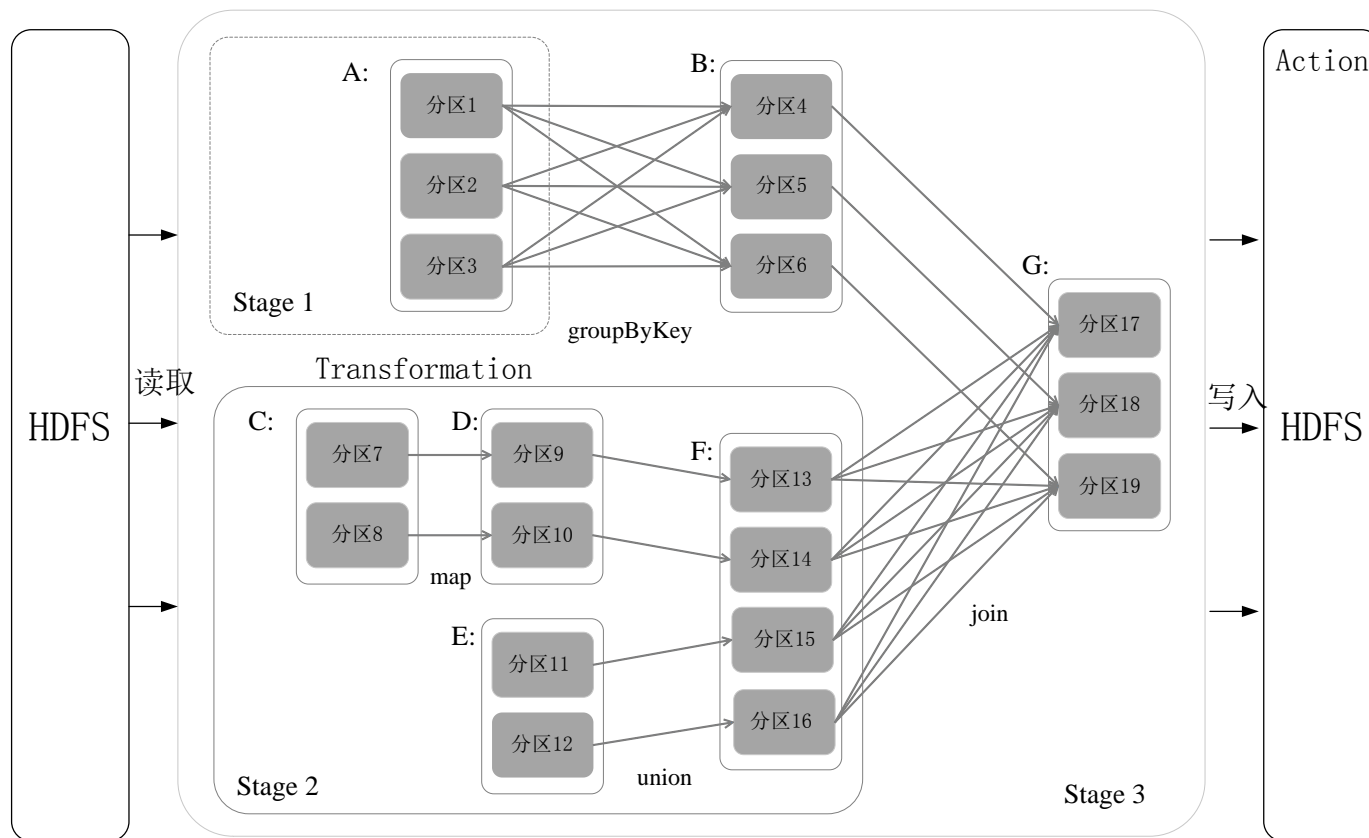
- 在DAG中进行反向解析，遇到宽依赖就断开
- 遇到窄依赖就把当前的RDD加入到Stage中
- 将窄依赖尽量划分在同一个Stage中，可以实现流水线计算



# 3.3.4 RDD运行原理

## 5.Stage的划分

被分成三个Stage，在Stage2中，从map到union都是窄依赖，这两步操作可以形成一个流水线操作



**流水线操作实例**  
分区7通过map操作生成的分区9，可以不用等待分区8到分区10这个map操作的计算结束，而是继续进行union操作，得到分区13，这样流水线执行大大提高了计算的效率

图16-10根据RDD分区的依赖关系划分Stage



# 3.3.4 RDD运行原理

## 6.RDD运行过程

通过上述对RDD概念、依赖关系和Stage划分的介绍，结合之前介绍的Spark运行基本流程，再总结一下RDD在Spark架构中的运行过程：

- (1) 创建RDD对象；
- (2) SparkContext负责计算RDD之间的依赖关系，构建DAG；
- (3) DAGScheduler负责把DAG图分解成多个Stage，每个Stage中包含了多个Task，每个Task会被TaskScheduler分发给各个WorkerNode上的Executor去执行。

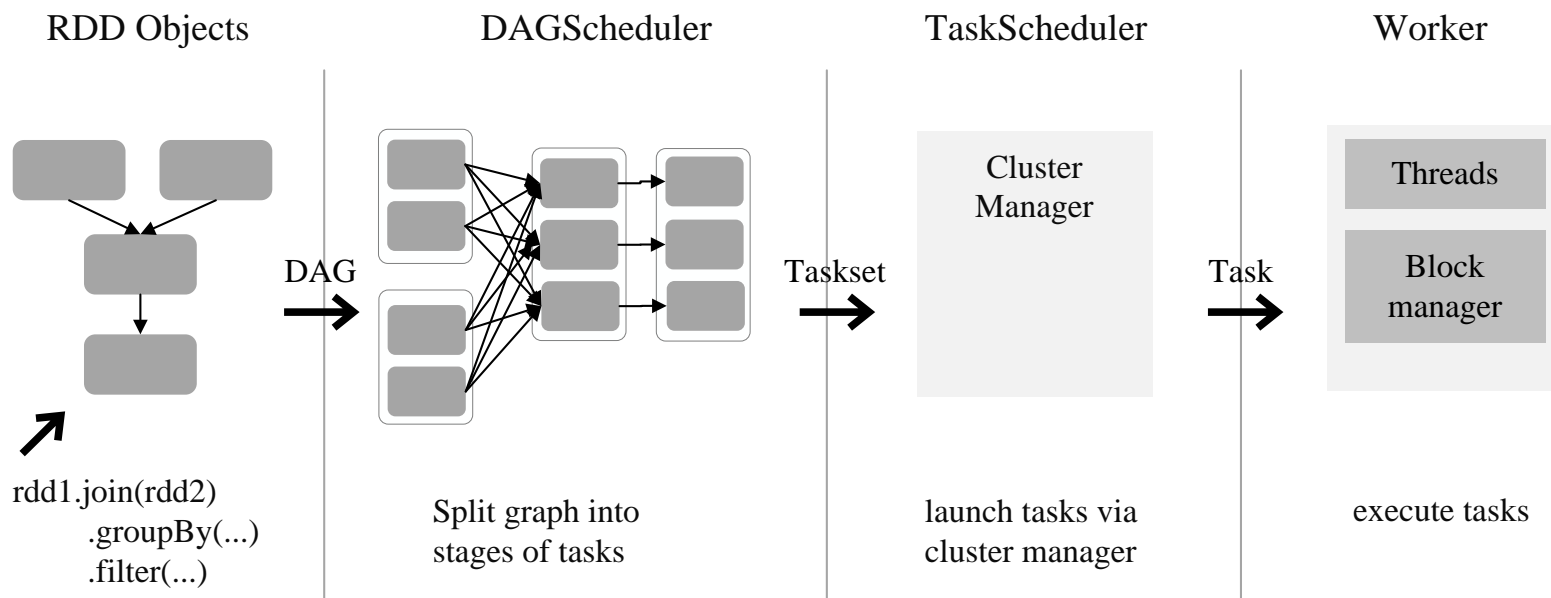


图16-11 RDD在Spark中的运行过程





## 3.4 Spark的部署和应用方式

3.4.1 Spark三种部署方式

3.4.2 从Hadoop+Storm架构转向Spark架构

3.4.3 Hadoop和Spark的统一部署



## 3.4.1 Spark三种部署方式

Spark支持三种不同类型的部署方式，包括：

- Standalone（类似于MapReduce1.0，slot为资源分配单位）
- Spark on Mesos（和Spark有血缘关系，更好支持Mesos）
- Spark on YARN

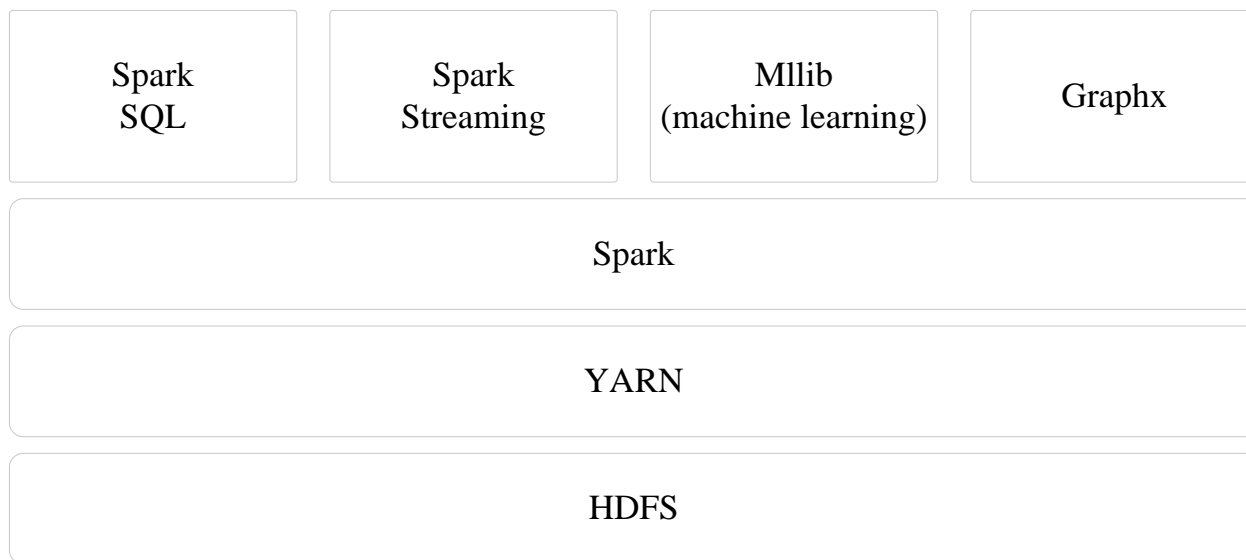
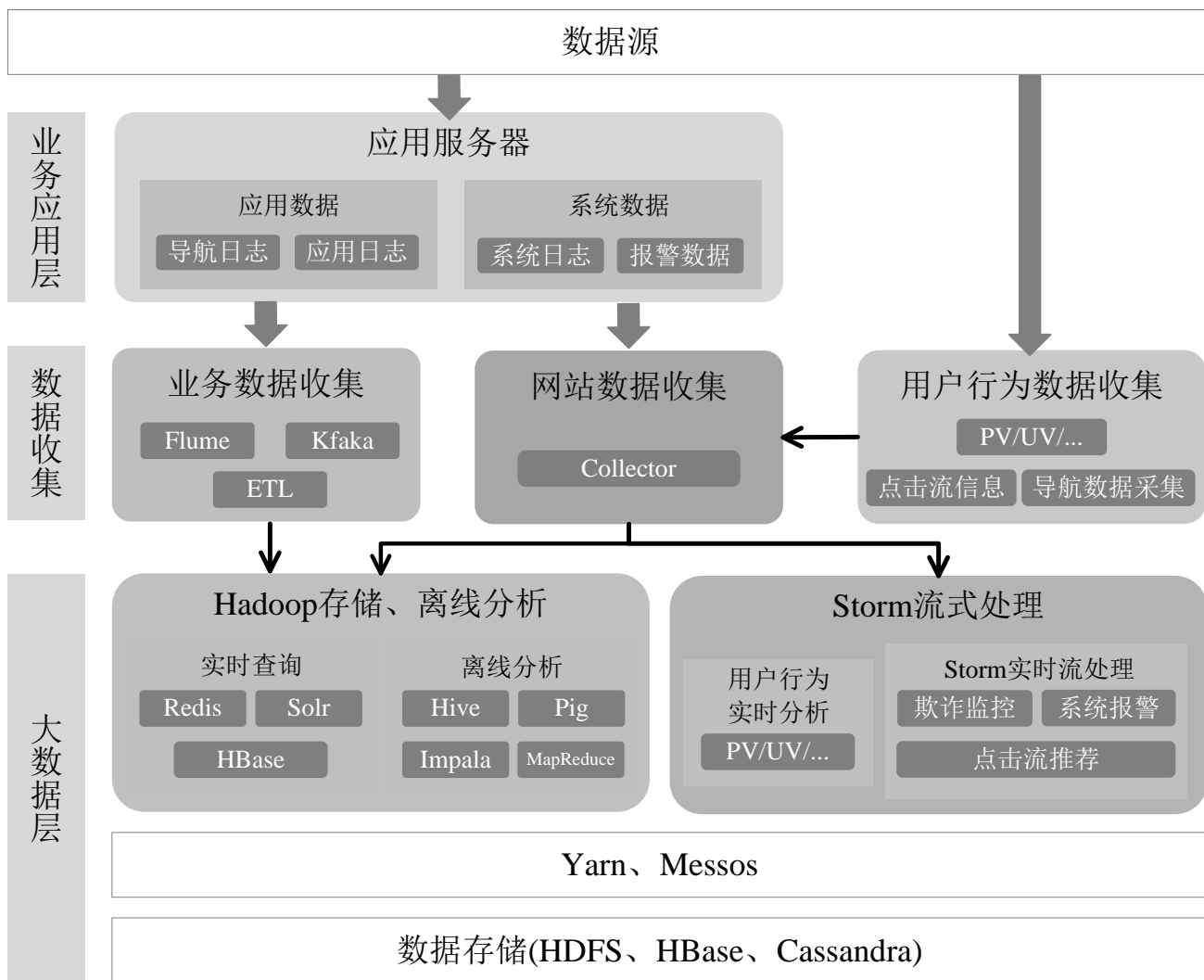


图16-17 Spark on Yarn架构



# 3.4.2 从Hadoop+Storm架构转向Spark架构



这种架构部署较为繁琐

图16-18 采用Hadoop+Storm部署方式的一个案例



## 3.4.2 从Hadoop+Storm架构转向Spark架构

用Spark架构具有如下优点：

- 实现一键式安装和配置、线程级别的任务监控和告警
- 降低硬件集群、软件维护、任务监控和应用开发的难度
- 便于做成统一的硬件、计算平台资源池

需要说明的是，Spark Streaming无法实现毫秒级的流计算，因此，对于需要毫秒级实时响应的企业应用而言，仍然需要采用流计算框架（如Storm）

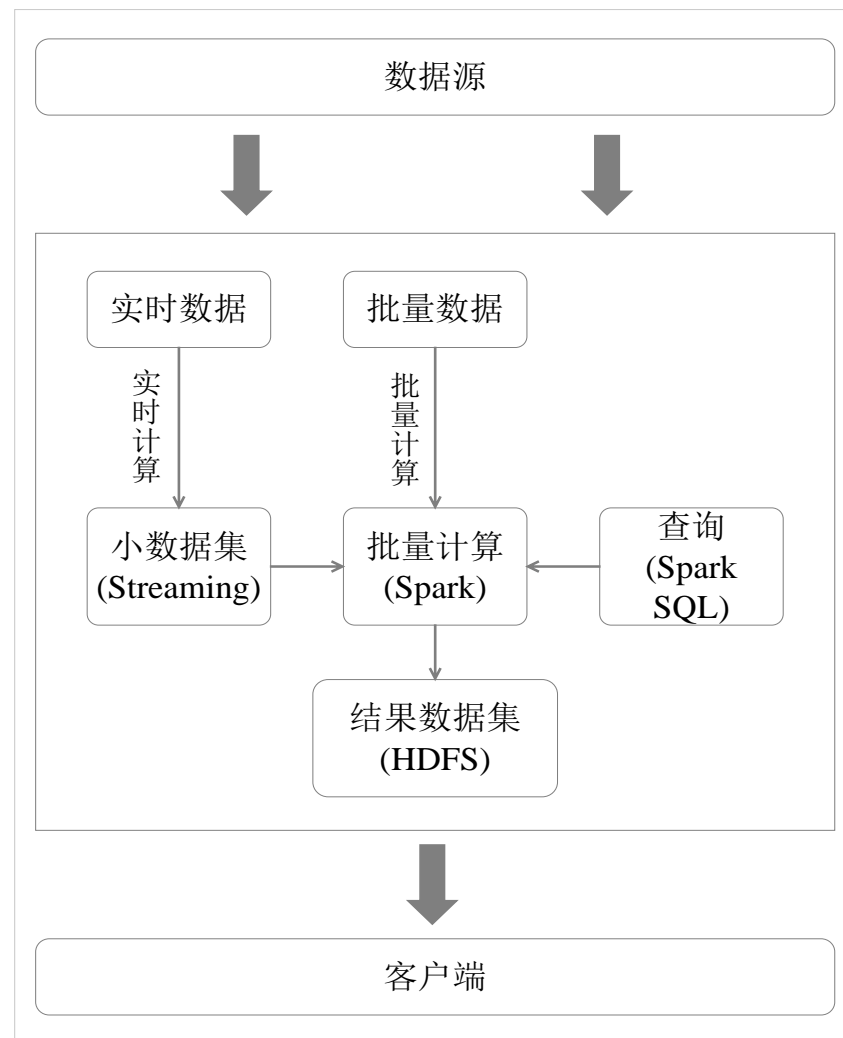


图16-19 用Spark架构满足批处理和流处理需求



## 3.4.3 Hadoop和Spark的统一部署

- 由于Hadoop生态系统中的一些组件所实现的功能，目前还是无法由Spark取代的，比如，Storm
- 现有的Hadoop组件开发的应用，完全转移到Spark上需要一定的成本

不同的计算框架统一运行在YARN中，可以带来如下好处：

- 计算资源按需伸缩
- 不同负载应用混搭，集群利用率高
- 共享底层存储，避免数据跨集群迁移

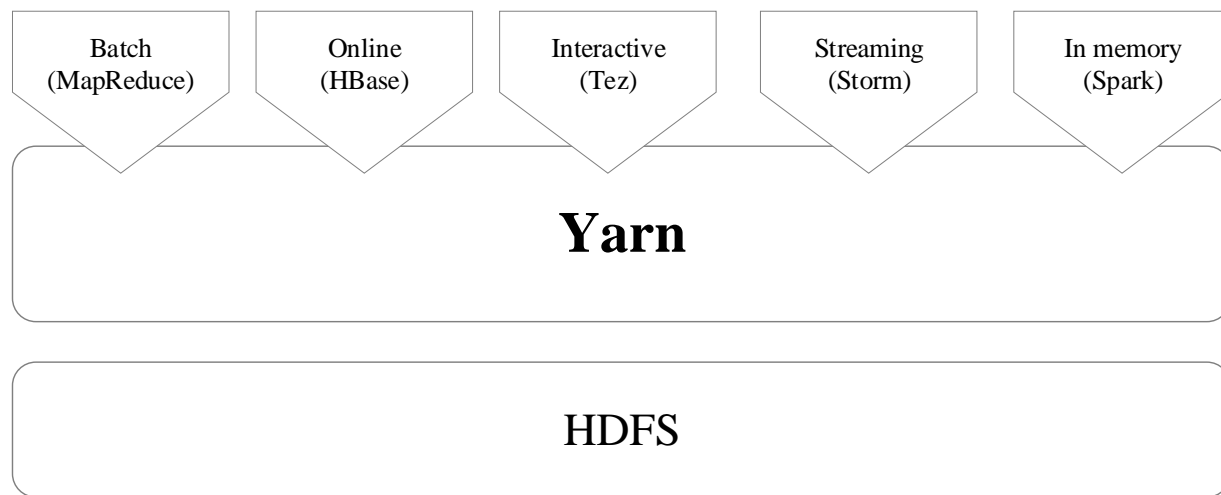


图16-20 Hadoop和Spark的统一部署



# 讨论：Spark和Hadoop

- 虽然Spark很快，但现在在生产环境中仍然不尽人意，无论扩展性、稳定性、管理性等方面都需要进一步增强
- 同时，Spark在流处理领域能力有限，如果要实现亚秒级或大容量的数据获取或处理需要其他流处理产品。Cloudera宣布旨在让Spark流数据技术适用于80%的使用场合，就考虑到了这一缺陷。我们确实看到实时分析（而非简单数据过滤或分发）场景中，很多以前使用S4或Storm等流式处理引擎的实现已经逐渐被Kafka+Spark Streaming代替
- Spark的流行将逐渐让MapReduce、Tez走进博物馆
- Hadoop现在分三块HDFS/MR/YARN，Spark比Hadoop性能好，只是Spark作为一个计算引擎，比MR的性能要好。但它的存储和调度框架还是依赖于HDFS/YARN，Spark也有自己的调度框架，但仍然非常不成熟，基本不可商用



# 附录：主讲教师林子雨简介



## 主讲教师：林子雨

单位：厦门大学计算机科学系

E-mail: ziyulin@xmu.edu.cn

个人网页: <http://www.cs.xmu.edu.cn/linziyu>

数据库实验室网站: <http://dblab.xmu.edu.cn>



扫一扫访问个人主页

林子雨，男，1978年出生，博士（毕业于北京大学），现为厦门大学计算机科学系助理教授（讲师），曾任厦门大学信息科学与技术学院院长助理、晋江市发展和改革局副局长。中国计算机学会数据库专业委员会委员，中国计算机学会信息系统专业委员会委员，荣获“2016中国大数据创新百人”称号。中国高校首个“数字教师”提出者和建设者，厦门大学数据库实验室负责人，厦门大学云计算与大数据研究中心主要建设者和骨干成员，2013年度厦门大学奖教金获得者。主要研究方向为数据库、数据仓库、数据挖掘、大数据、云计算和物联网，并以第一作者身份在《软件学报》《计算机学报》和《计算机研究与发展》等国家重点期刊以及国际学术会议上发表多篇学术论文。作为项目负责人主持的科研项目包括1项国家自然科学基金青年基金项目(No.61303004)、1项福建省自然科学基金项目(No.2013J05099)和1项中央高校基本科研业务费项目(No.2011121049)，同时，作为课题负责人完成了国家发改委城市信息化重大课题、国家物联网重大应用示范工程区域试点泉州市工作方案、2015泉州市互联网经济调研等课题。中国高校首个“数字教师”提出者和建设者，2009年至今，“数字教师”大平台累计向网络免费发布超过100万字高价值的研究和教学资料，累计网络访问量超过100万次。打造了中国高校大数据教学知名品牌，编著出版了中国高校第一本系统介绍大数据知识的专业教材《大数据技术原理与应用》，并成为京东、当当网等网店畅销书籍；建设了国内高校首个大数据课程公共服务平台，为教师教学和学生学习大数据课程提供全方位、一站式服务，年访问量超过50万次。具有丰富的政府和企业信息化培训经验，厦门大学管理学院EDP中心、浙江大学管理学院EDP中心、厦门大学继续教育学院、泉州市科技培训中心特邀培训讲师，曾给中国移动通信集团公司、福州马尾区政府、福建龙岩卷烟厂、福建省物联网科学研究院、石狮市物流协会、厦门市物流协会、浙江省中小企业家、四川泸州企业家、江苏沛县企业家等开展信息化培训，累计培训人数达3000人以上。



# 附录：林子雨编著《Spark入门教程》

厦门大学林子雨编著《Spark入门教程》  
教程内容包括Scala语言、Spark简介、安装、运行架构、RDD的设计与运行原理、部署模式、RDD编程、键值对RDD、数据读写、Spark SQL、Spark Streaming、MLlib等



厦门大学林子雨



子雨大数据之Spark入门教程

披荆斩棘，在大数据丛林中开辟学习捷径



免费在线教程：<http://dbllab.xmu.edu.cn/blog/spark/>





# 附录：《大数据技术原理与应用》教材



扫一扫访问教材官网

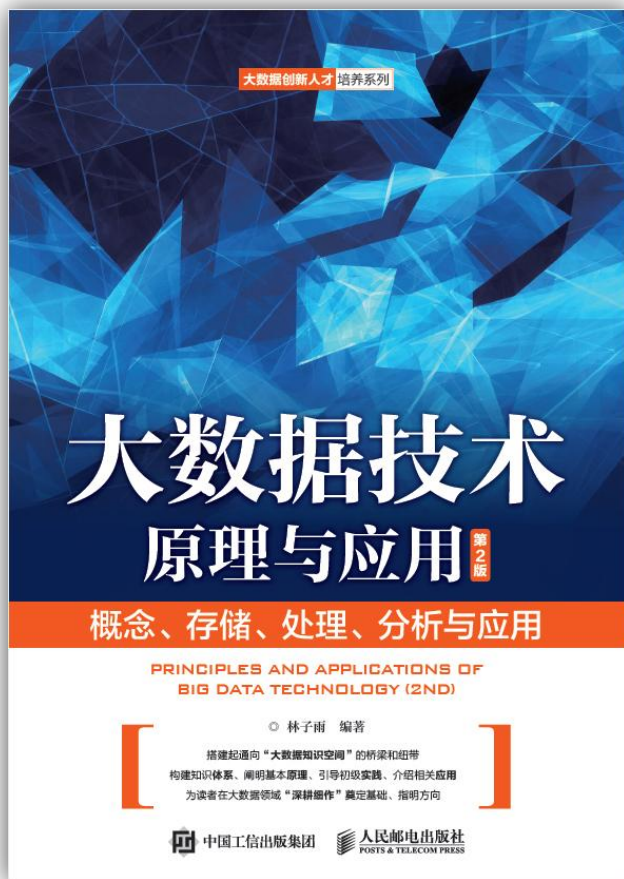
《大数据技术原理与应用——概念、存储、处理、分析与应用（第2版）》，由厦门大学计算机科学系林子雨老师编著，是中国高校第一本系统介绍大数据知识的专业教材。

全书共有15章，系统地论述了大数据的基本概念、大数据处理架构Hadoop、分布式文件系统HDFS、分布式数据库HBase、NoSQL数据库、云数据库、分布式并行编程模型MapReduce、Spark、流计算、图计算、数据可视化以及大数据在互联网、生物医学和物流等各个领域的应用。在Hadoop、HDFS、HBase、MapReduce和Spark等重要章节，安排了入门级的实践操作，让读者更好地学习和掌握大数据关键技术。

本书可以作为高等院校计算机专业、信息管理等相关专业的大数据课程教材，也可供相关技术人员参考、学习、培训之用。

欢迎访问《大数据技术原理与应用——概念、存储、处理、分析与应用（第2版）》教材官方网站：

<http://dblab.xmu.edu.cn/post/bigdata>





# 附录：中国高校大数据课程公共服务平台



## 中国高校大数据课程 公共服务平台

<http://dblab.xmu.edu.cn/post/bigdata-teaching-platform/>



扫一扫访问平台主页



扫一扫观看3分钟FLASH动画宣传片

The background is a solid blue color with faint, light-blue silhouettes of people. At the top, there are two groups of people holding hands. On the right side, there is a silhouette of a person standing with their hand to their face. At the bottom left, there are silhouettes of two people sitting at a table, one with their hand to their face.

**Thank You!**

Department of Computer Science, Xiamen University, 2017