## 概述

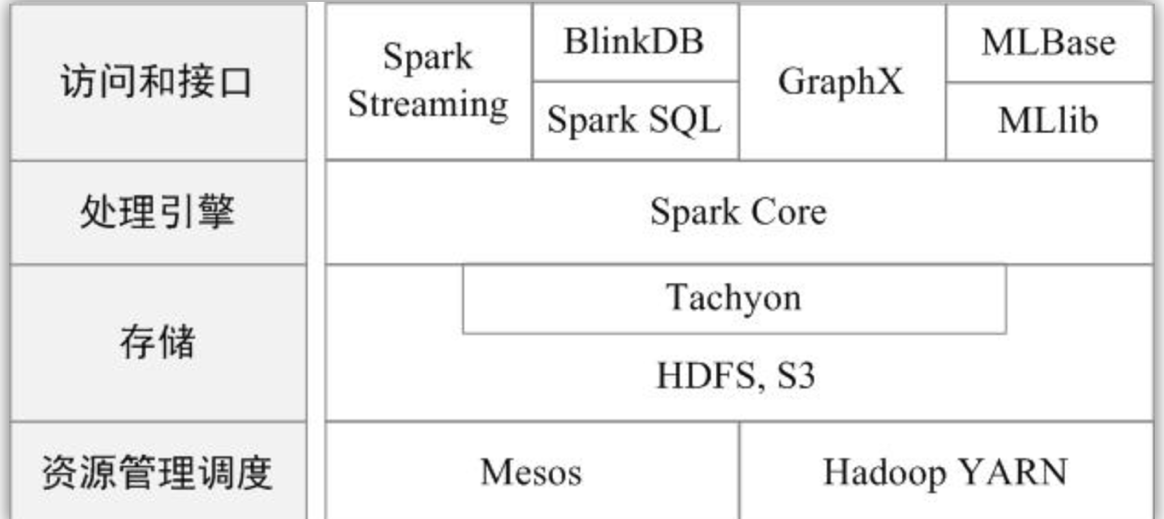
apache软件基金会最重要的三大分布式计算系统开源项目之一（hadoop，spark，storm）。hadoop存在缺点：

* 表达能力有限，计算都必须转换为map reduce
* 磁盘io开销大
* 延迟高

spark借鉴hadoop mapreduce的优点，很好的解决了mapreduce所面临的问题：

* 属于mapreduce，但不局限于mapreduce，还提供了多种数据集操作类型，编程模型更灵活
* 提供内存计算
* 基于DAG任务调度机制，由于mapreduce的迭代执行机制

架构图：



BDAS（berkeley data analytics stack）

主要包括spark core（提供spark基本功能，如内存计算、任务调度、部署模式等），spark sql（开发人员直接处理rdd，同时也可查询hive，hbase），spark streaming（处理流式计算），mllib（机器学习相关），graphx（图计算api）

## 运行架构

RDD：resilient distributed dataset，弹性分布式数据集，分布式内存的一个抽象概念

DAG：directed acyclic graph，有向无环图

Executor：运行在工作节点的一个进程

应用：spark应用程序

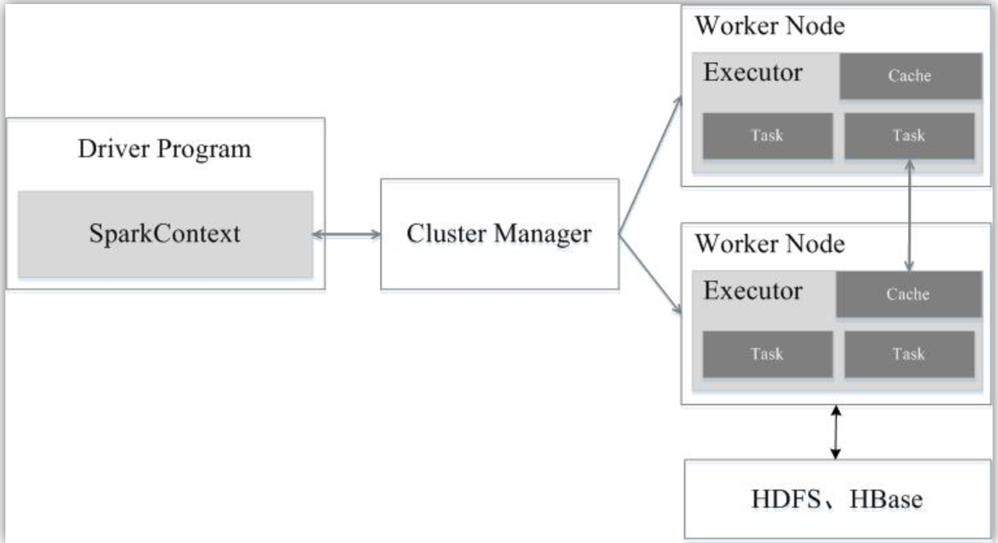
任务：运行在executor上的工作单元

作业：一个作业包含多个RDD及作用于RDD上的各种操作

阶段：作业的基本调度单位

基本运行流程：

1. 为application构建基本运行环境，有driver构建sparkcontext
2. sparkcontext跟资源管理器（cluster manager）通信，申请运行executor运行的资源，executor运行情况随着“心跳”发送到资源管理器
3. sparkcontext根据RDD依赖构建DAG，DAG调度器解析，该部分见下一节任务在executor上运行结束反馈结果

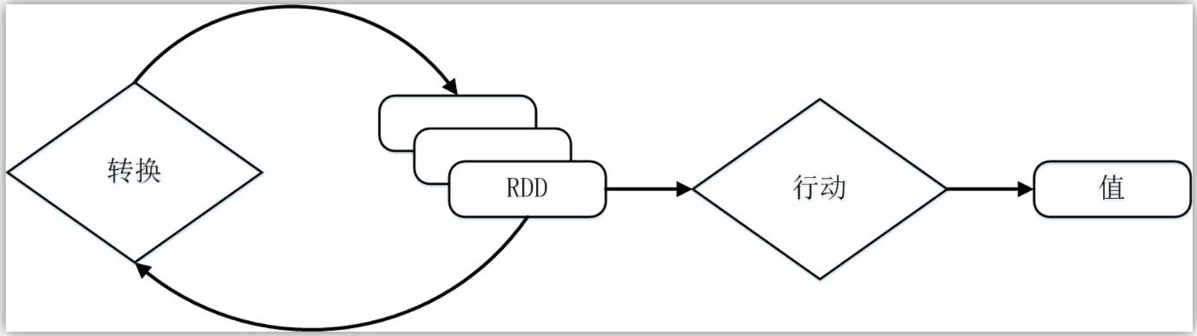


## RDD设计与运行原理

<http://dblab.xmu.edu.cn/blog/985-2/>

### 基本概念

RDD弹性分布式数据集，本质上只读，可执行的操作有transformation和action，transformation指定各个RDD的相互依赖的关系，action执行计算并指定输出形式。两者区别：transfromation接受RDD返回RDD，action接受RDD返回非RDD。

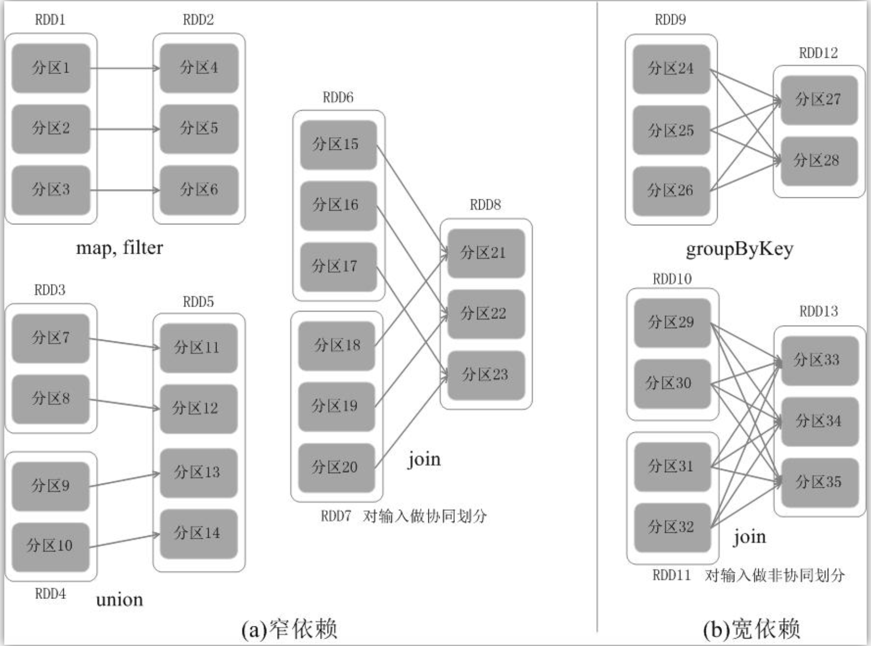


transformation不会出发真正的计算，action才会触发。

窄依赖和宽依赖：

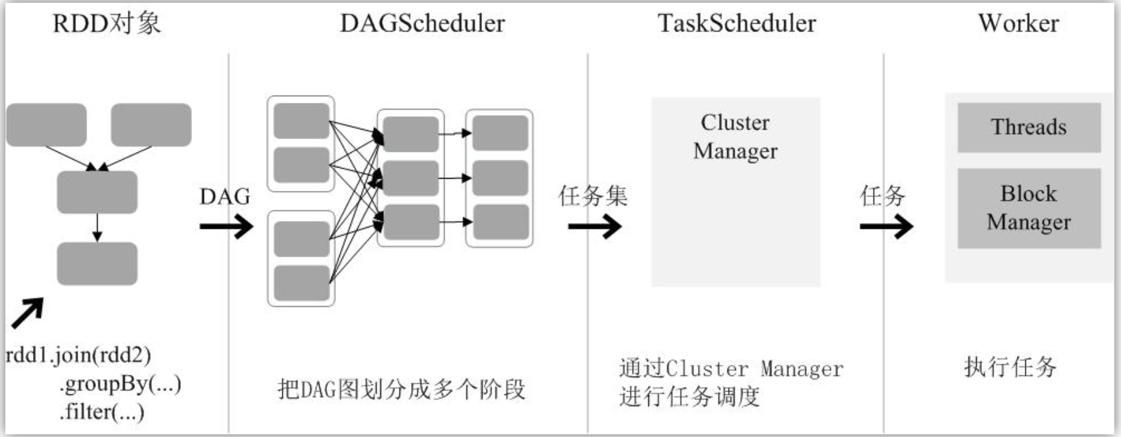
窄依赖：父RDD的每条记录只有一条出路。

宽依赖：父RDD中存在记录有多余一条的出路。



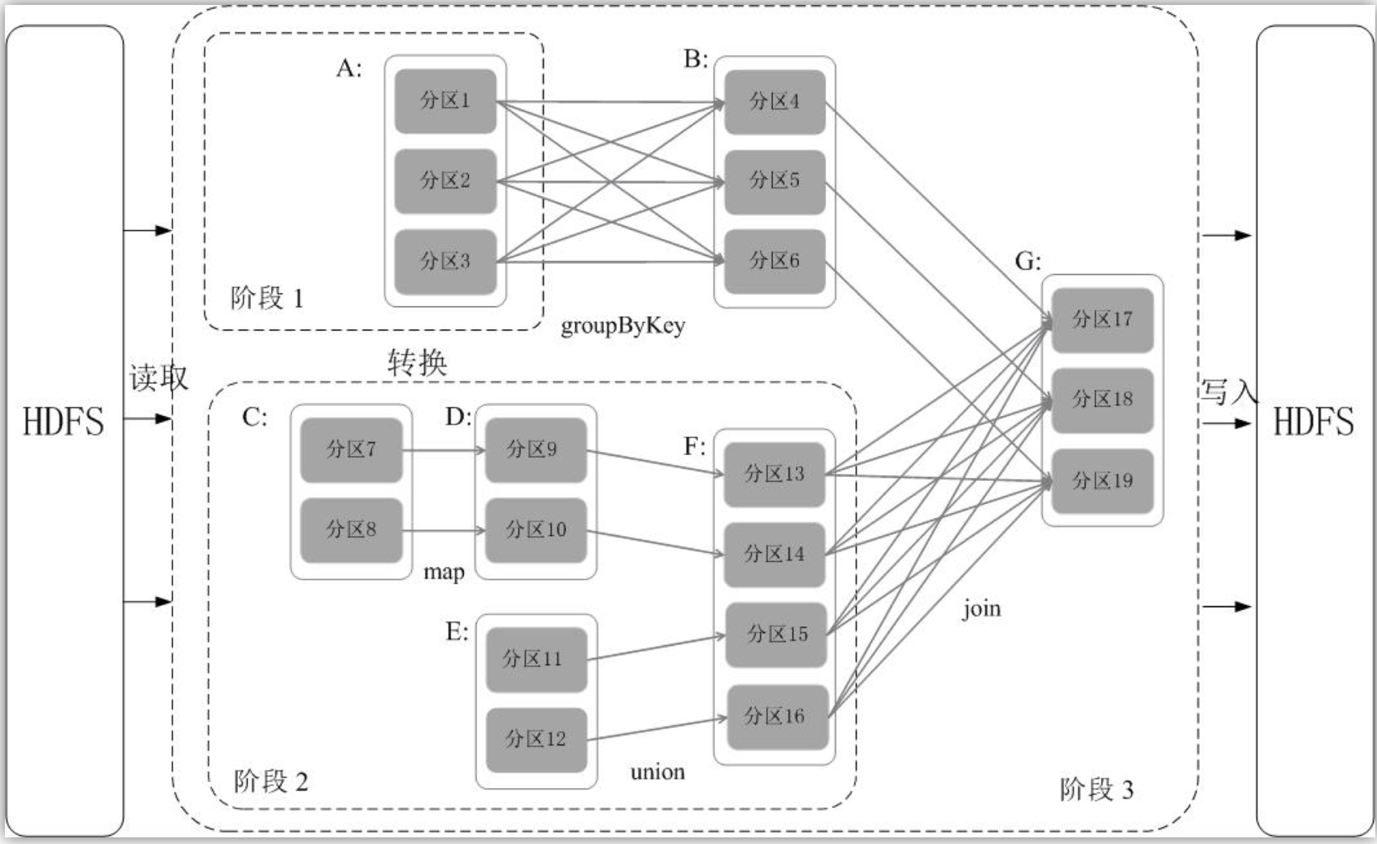
### 运行过程

1. 创建RDD
2. sparkcontext负责计算RDD之间的依赖，构建DAG
3. DAGScheduler 负责分解DAG，然后cluster manager分发任务给worker node上的executor执行



### DAGScheduler工作原理

原则：在DAG中反向分解，遇到宽依赖就断开，遇到窄依赖就把当前的RDD加入到当前阶段。



如图，从G开始，A到B、F到G时宽依赖，断开。把一个DAG图划分为多阶段之后，每个阶段代表一组关联的、相互之间没有shuffle依赖关系的任务组成的任务集合，每个任务集合被cluster manager分配给executor处理。

## spark安装和使用

### Hadoop安装

<http://dblab.xmu.edu.cn/blog/install-hadoop/>

* ssh安装和无密码登录：ssh-keygen -t rsa;cat ~/.ssh/id\_rsa.pub >> authorized.keys;ssh localhost
* jdk安装，jdk8安装，jdk8以上的版本spark-shell报错
* 下载： <http://mirror.bit.edu.cn/apache/hadoop/common/> 解压/usr/local
* 伪分布式配置：

|  |
| --- |
| ./etc/Hadoop/core-site.xml:  <configuration>  <property>  <name>hadoop.tmp.dir</name>  <value>file:/usr/local/hadoop/tmp</value>  <description>Abase for other temporary directories.</description>  </property>  <property>  <name>fs.defaultFS</name>  <value>hdfs://localhost:9000</value>  </property>  </configuration> |
| hdfs-site.xml:  <configuration>  <property>  <name>dfs.replication</name>  <value>1</value>  </property>  <property>  <name>dfs.namenode.name.dir</name>  <value>file:/usr/local/hadoop/tmp/dfs/name</value>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value>file:/usr/local/hadoop/tmp/dfs/data</value>  </property>  </configuration> |

* 格式化namenode：bin/hdfs namenode -format
* 开启hdfs：./sbin/start-dfs.sh
* 验证：jps或者http://ip:50070/

### spark安装

* 下载：<http://spark.apache.org/downloads.html>

选择pre-build with user-provided Hadoop[…..]

* 解压到/usr/local
* 配置：cp ./conf/spark-env.sh.template ./conf/spark-env.sh

添加：export SPARK\_DIST\_CLASSPATH=$(/usr/local/hadoop/bin/hadoop classpath)

* 验证：bin/run-example SparkPi|grep ‘Pi is’或者bin/spark-shell