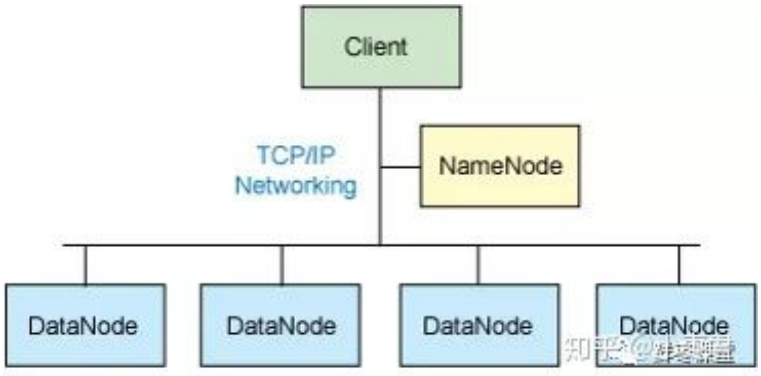
1. Hadoop

Hadoop的核心，说白了，就是**HDFS**和**MapReduce**.
HDFS为海量数据提供了**存储**，**MapReduce**为海量数据提供了**计算框架**。

1）HDFS

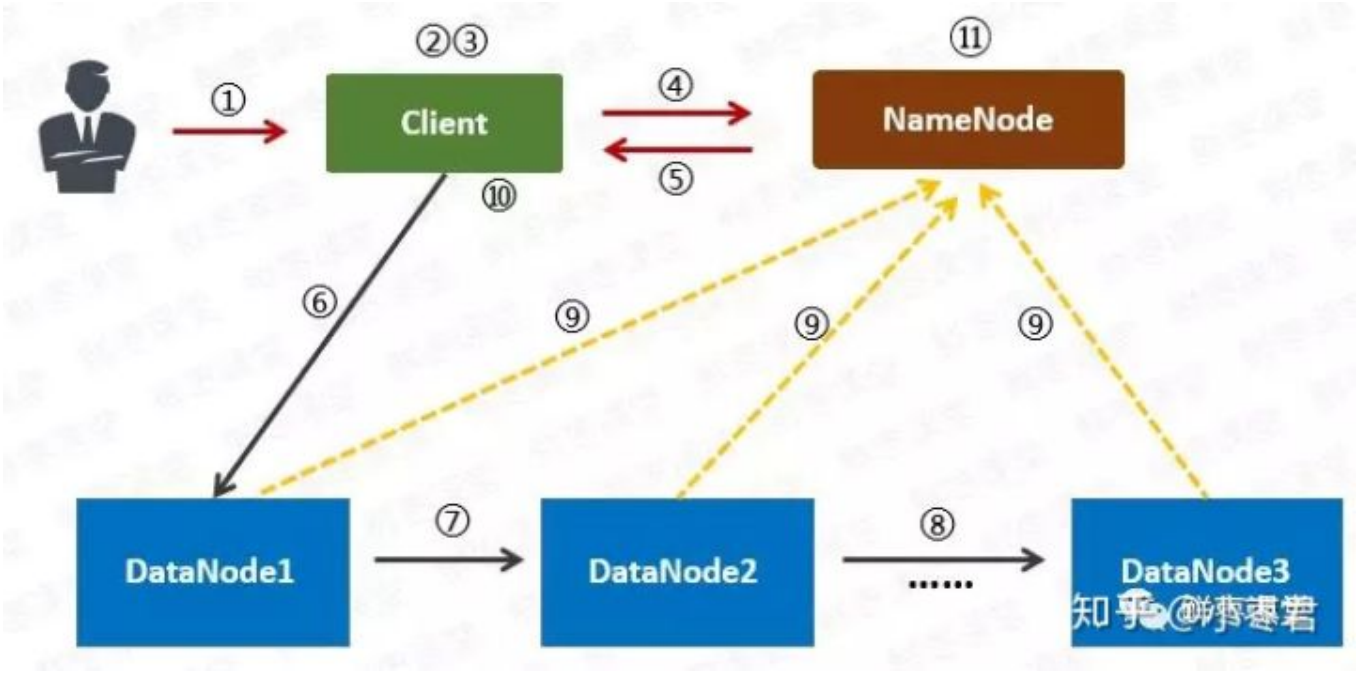
整个HDFS有三个重要角色：**NameNode（名称节点）**、**DataNode（数据节点）**和**Client（客户机）**。



NameNode：是Master节点（主节点）,可以看作是分布式文件系统中的管理者，主要负责管理文件系统的命名空间、集群配置信息和存储块的复制等。NameNode会将文件系统的Meta-data存储在内存中，这些信息主要包括了文件信息、每一个文件对应的文件块的信息和每一个文件块在DataNode的信息等。
DataNode：是Slave节点（从节点），是文件存储的基本单元，它将Block存储在本地文件系统中，保存了Block的Meta-data，同时周期性地将所有存在的Block信息发送给NameNode。
Client：切分文件；访问HDFS；与NameNode交互，获得文件位置信息；与DataNode交互，读取和写入数据。

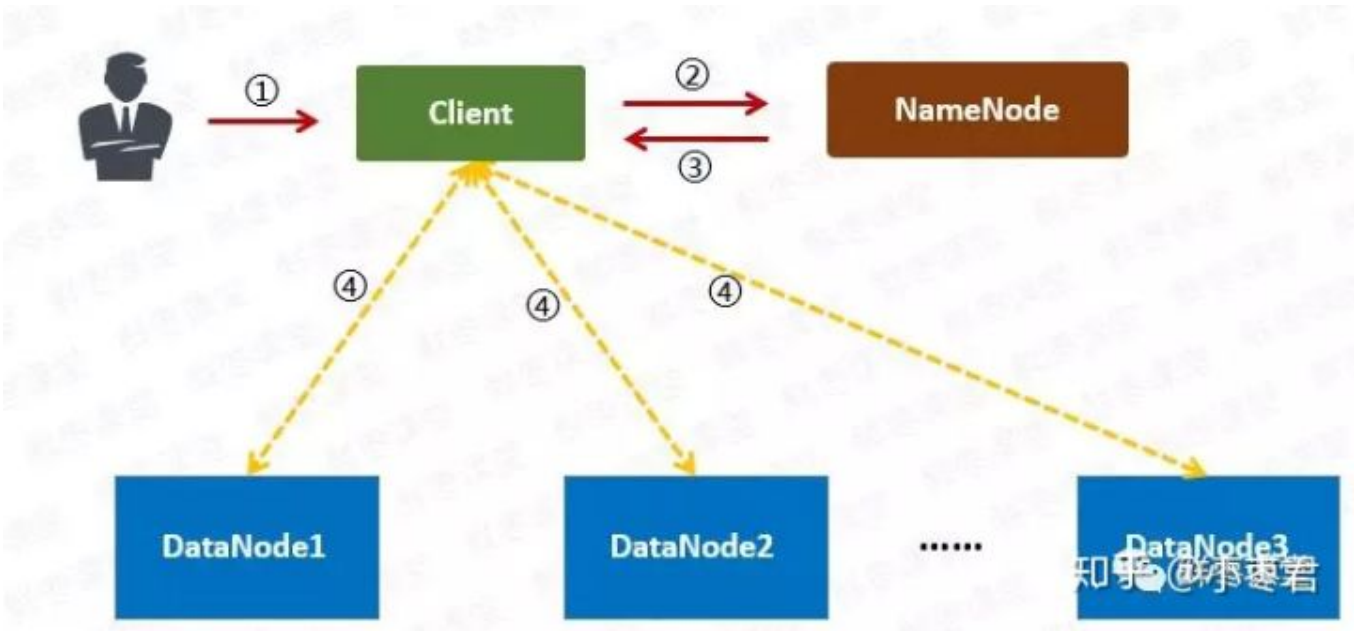
Block（块）的概念：**Block**是HDFS中的**基本读写单元**；HDFS中的文件都是被切割为**block（块）**进行存储的；这些块被复制到多个DataNode中；块的大小（通常为64MB）和复制的块数量在创建文件时由Client决定。

2）HDFS写入流程



1 用户向Client（客户机）提出请求。例如，需要写入200MB的数据。
2 Client制定计划：将数据按照64MB为块，进行切割；所有的块都保存三份。
3 Client将大文件切分成块（block）。
4 针对第一个块，Client告诉NameNode（主控节点），请帮助我，将64MB的块复制三份。
5 NameNode告诉Client三个DataNode（数据节点）的地址，并且将它们根据到Client的距离，进行了排序。
6 Client把数据和清单发给第一个DataNode。
7 第一个DataNode将数据复制给第二个DataNode。
8 第二个DataNode将数据复制给第三个DataNode。
9 如果某一个块的所有数据都已写入，就会向NameNode反馈已完成。
10 对第二个Block，也进行相同的操作。
11 所有Block都完成后，关闭文件。NameNode会将数据持久化到磁盘上。

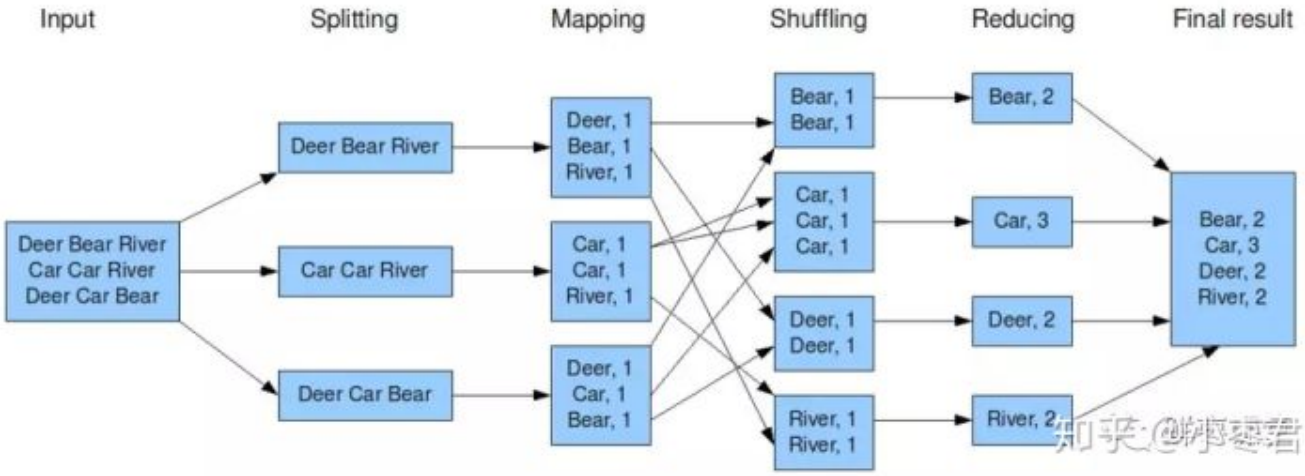
3）HDFS读取流程



1 用户向**Client**提出**读取请求**。
2 **Client**向**NameNode**请求这个**文件的所有信息**。
3 **NameNode**将给**Client**这个文件的**块列表**，以及存储各个块的数据节点清单（按照和客户端的距离排序）。
4 **Client**从**距离最近**的数据节点下载所需的块。

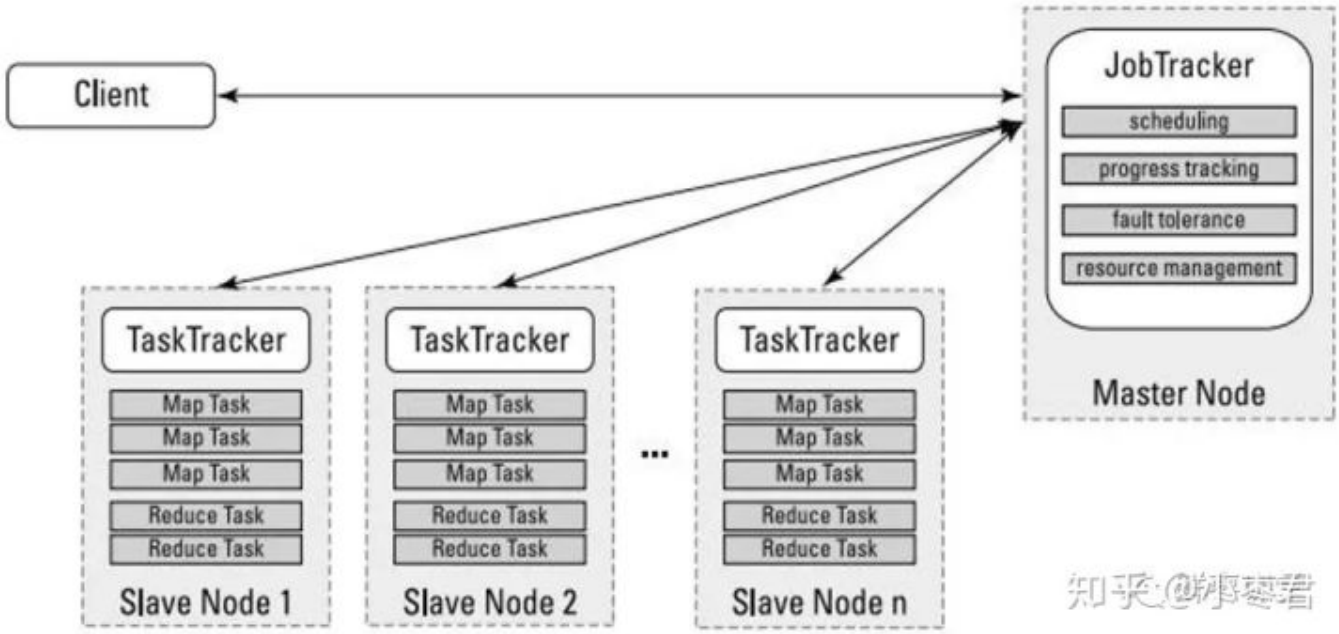
4）MapReduce

MapReduce其实是一种**编程模型**。这个模型的核心步骤主要分两部分：**Map（映射）**和**Reduce（归约）**。



1 Hadoop将输入数据**切成若干个分片**，并将每个**split（分割）**交给一个**map task（Map任务）**处理。

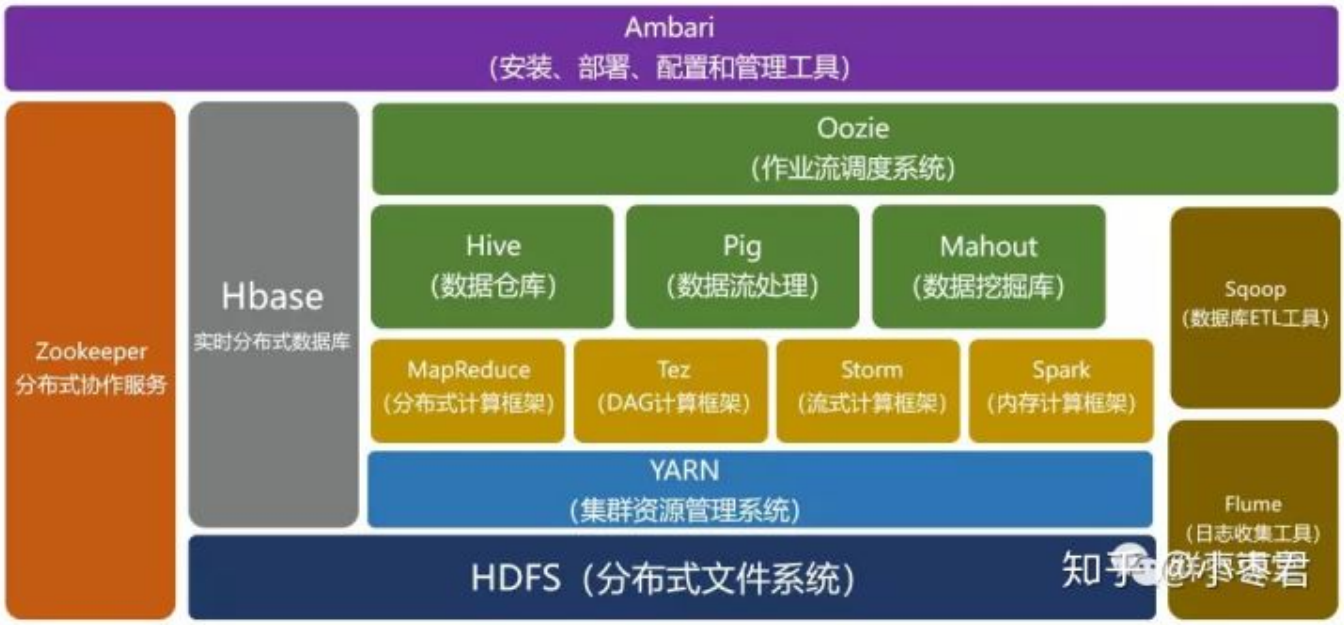
2 **Mapping**之后，相当于得出这个**task**里面，每个词以及它出现的次数。
3 **shuffle（拖移）**将相同的词放在一起，并对它们进行**排序**，分成若干个分片。
4 根据这些分片，进行**reduce（归约）。**
5 统计出**reduce task**的结果，输出到文件。
在MapReduce里，为了完成上面这些过程，需要两个角色：**JobTracker**和**TaskTracker。**



JobTracker调度和管理TaskTracker
JobTracker集群中任一台计算机上DataNode

5）新版本及生态圈

2.0版本中，在HDFS之上，增加了**YARN（资源管理框架）层。**它是一个资源管理模块，为各类**应用程序提供资源管理和调度。**



HBase：来源于Google的BigTable；是一个高可靠性、高性能、面向列、可伸缩的分布式数据库。

Hive：是一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，通过类SQL语句快速实现简单的MapReduce统计，不必开发专门的MapReduce应用，十分适合数据仓库的统计分析。

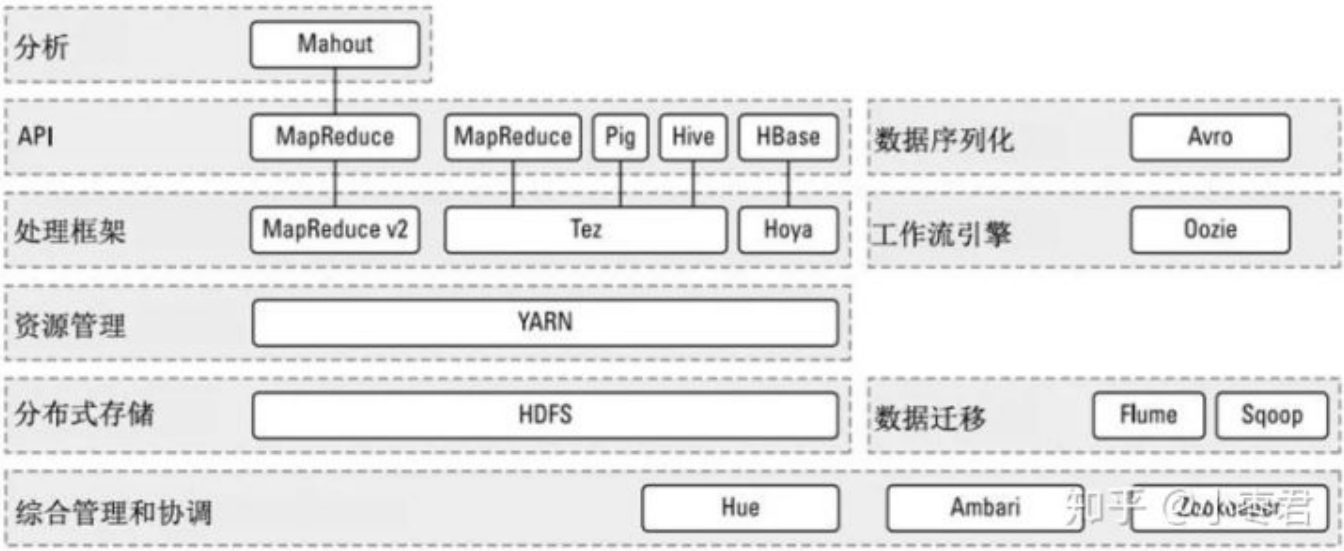
Pig：是一个基于Hadoop的大规模数据分析工具，它提供的SQL-LIKE语言叫Pig Latin，该语言的编译器会把类SQL的数据分析请求转换为一系列经过优化处理的MapReduce运算。

ZooKeeper：来源于Google的Chubby；它主要是用来解决分布式应用中经常遇到的一些数据管理问题，简化分布式应用协调及其管理的难度。

Ambari：Hadoop管理工具，可以快捷地监控、部署、管理集群。

Sqoop：用于在Hadoop与传统的数据库间进行数据的传递。

Mahout：一个可扩展的机器学习和数据挖掘库。



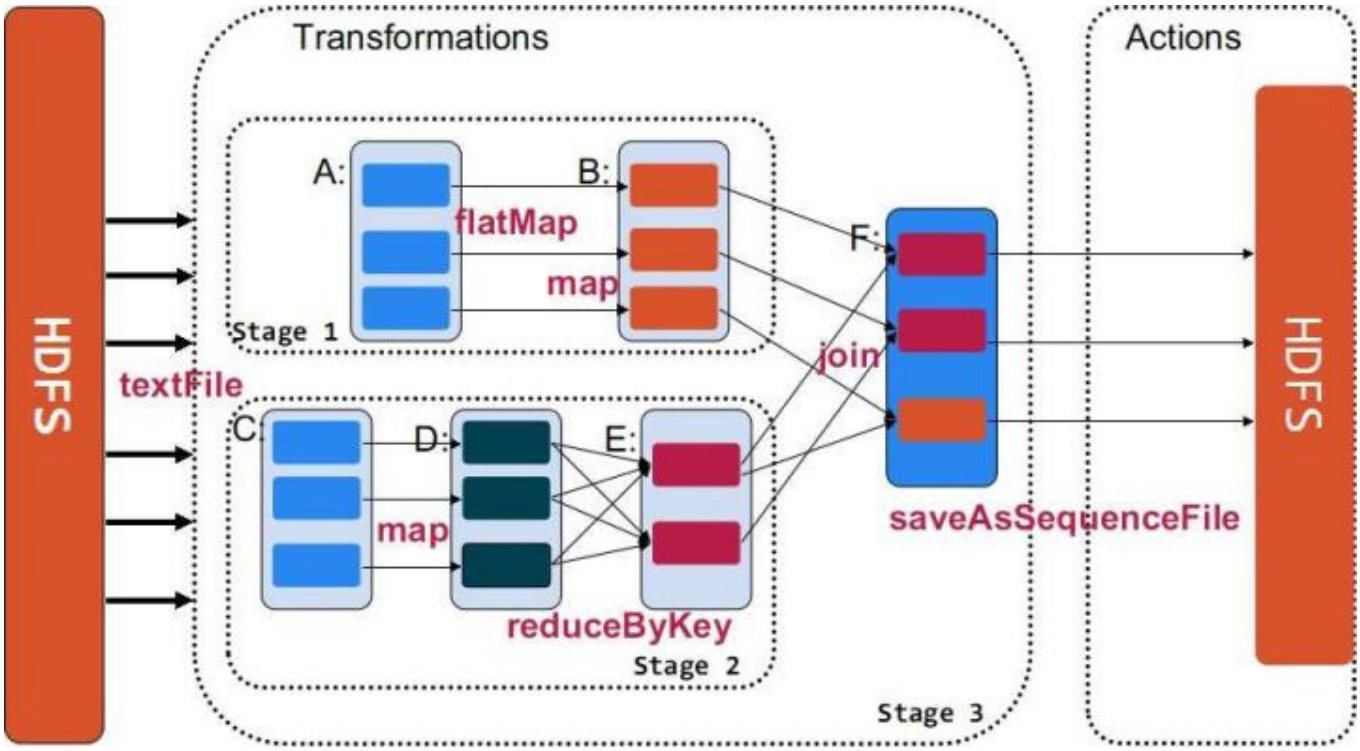
6）Hadoop的优点和应用

高可靠性：这个是由它的基因决定的。
高扩展性：Hadoop是在可用的计算机集群间分配数据并完成计算任务的，这些**集群**可以方便地进行**扩展**。说白了，想变大很容易。
高效性：Hadoop能够在节点之间动态地移动数据，并保证各个节点的**动态平衡**，因此处理速度非常快。
高容错性：Hadoop能够自动保存**数据的多个副本**，并且能够自动将失败的任务重新分配。这个其实也算是高可靠性。
低成本：Hadoop是**开源**的，依赖于**社区服务**，使用成本比较低。

2. Spark

2.1 Spark计算

Spark将数据处理过程抽象为对**内存中RDD（弹性分布式数据集）**的操作，**RDD**的可以通过从**数据源直接读取**和**集合数据类型封装**两种方式创建。针对RDD的操作，根据其结果主要分为如map、flatMap、mapPartition、filter等生成新的RDD的transformation（转换）操作和collect、reduce、foreach等生成集合数据类型或结果写入的action（行为）操作两大类。



action操作RDDtransformation操作DAG（有向无环图）执行策略进行优化partition（分区）partitionaction操作transformation生成的结果是否为RDD

2.2 RDD及其特点

1、**RDD**是Spark的**核心数据模型**，但是个**抽象类**，全称为Resillient Distributed Dataset，即**弹性分布式数据集。**

2、**RDD**在抽象上来说是一种**元素集合**，包含了数据。它是被分区的，分为多个分区，每个分区分布在**集群中的不同节点上**，从而让RDD中的数据可以被**并行操作。**（分布式数据集）

3、**RDD通常通过Hadoop上的文件**，即**HDFS文件或者Hive表**，来进行创建；有时也可以通过应用程序中的集合来创建。

4、RDD最重要的特性就是，提供了**容错性**，可以**自动从节点失败中恢复过来。**即如果某个节点上的**RDDpartition**，因为**节点故障**，导致数据丢了，那么**RDD**会自动通过自己的数据来源**重新计算该partition。**这一切对使用者是**透明**的。

5、RDD的**数据默认**情况下**存放在内存中**的，但是在**内存资源不足**时，Spark会自动将**RDD数据写入磁盘。**（弹性）

2.2 Spark 和 Hadoop对比

1. 编程模型：SparkCore vs MapReduce

1）基于**内存**和基于**磁盘**，MR耗时都花在**shuffle**上了
2）**编程模型死板**， 必须填进去**MapReduce**中， spark一段就ok，**mr要写几段。**
3）一个是**迭代式运算**，MR两个阶段完了就是忘了

2. 搜索引擎：SparkSQLvsHive

Hive是数据仓库，SparkSQL替代的并非Hive，而是Hive中**搜索引擎部分**
Hive搜索引擎是基于**MR shuffle**，所以是**非常耗时**；而**Spark**是基于**内存。**

3. 流处理：SparkStreaming vs Storm

1）**SparkStreaming**并非真正意义的**实时流处理**，只能说是**准实时。**因为它使用了**缓存**（linux内核中LRU链表中提到过，如果每条记录都**实时处理**，中间**设计加锁解锁**等处理很耗时），每个一段时间处理一次。
2）所以说对对于银行这种要求准实时最好使用**storm**，对于处理时间段不变化且不要求精确实时的使用**sparkStreaming**.

4. 场景要求

1) **Hadoop**：**离线批处理。**（面对SQL交互式查询、实时处理及机器学习等需要和第三方框架结合。多种数据格式转换，导致消耗大量资源）
2) **Spark**：**批处理、实时处理**