

SPARK 工作原理及 RDD 特性

Spark 基本工作原理

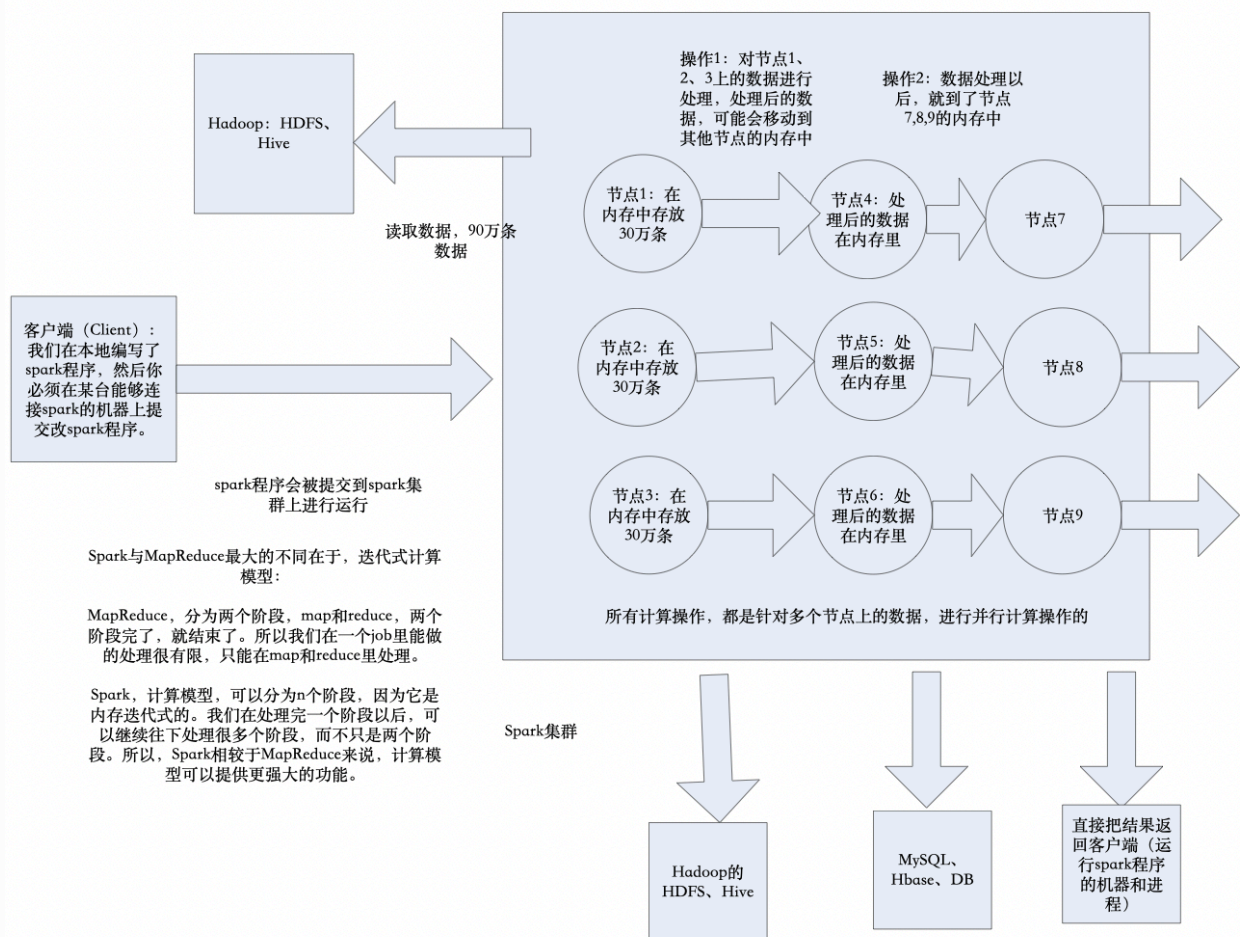
分布式

主要基于内存（少数情况基于磁盘）

迭代式计算

Spark 开发

1. 核心开发:离线批处理 / 延迟性的交互式数据处理
2. SQL查询:底层都是RDD和计算操作
3. 实时计算:底层都是RDD和计算操作



Spark的核心编程是什么? 其实, 就是:

首先, 第一, 定义初始的RDD, 就是说, 你要定义第一个RDD是从哪里, 读取数据, hdfs、linux本地文件、程序中的集合。

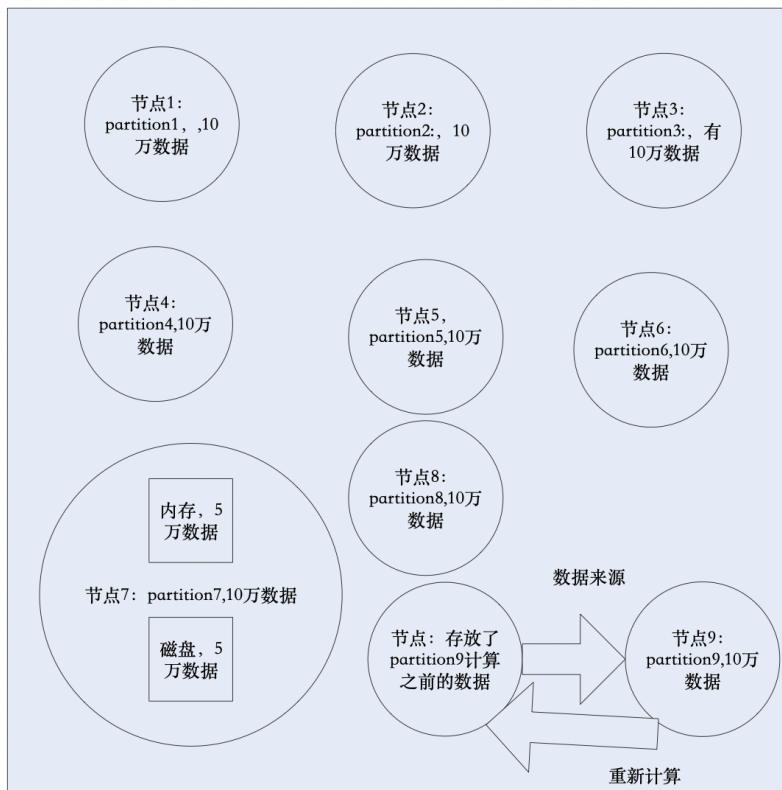
第二, 定义对RDD的计算操作, 这个在spark里称之为算子, map、reduce、flatMap、groupByKey, 比mapreduce提供的map和reduce强大的太多太多了。

第三, 其实就是循环往复的过程, 第一个计算完了以后, 数据可能就会到了新的一批节点上, 也就是变成一个新的RDD。然后再次反复, 针对新的RDD定义计算操作。。。。

第四, 最后, 就是获得最终的数据, 将数据保存起来。

每一批节点上的每一批数据, 实际上就是一个RDD!!! 一个RDD是分布式的, 所以数据都散落在一批节点上了, 每个节点都存储了RDD的部分partition。

1. RDD是Spark提供的核心抽象, 全称为Resilient 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RDD partition, 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)



Spark集群

一个RDD，在逻辑上，抽象地代表了一个HDFS文件。

但是，它实际上是被分区得。分为多个分区。多个分区散落在Spark集群中，不同的节点上。

比如说，RDD有90万数据。分为9个partition，9个分区。

现在，节点9出了些故障，导致partition9的数据丢失了。那么此时Spark会脆弱到直接报错，直接挂掉吗？不可能！！

RDD是有很强的容错性的，当它发现自己的数据丢失了以后，会自动从自己来源的数据进行重计算，重新获取自己这份数据，这一切对用户，都是完全透明的。

RDD的每个partition，在spark节点上存储时，默认都是放在内存中的。但是如果说内存放不下这么多数据时，比如每个节点最多放5万数据，结果你每个partition是10万数据。那么就会把partition中的部分数据写入磁盘上，进行保存。

而上述这一切，对于用户来说，都是完全透明的。也就是说，你不用去管RDD的数据存储在哪里，内存，还是磁盘。只要关注，你针对RDD来进行计算，和处理，等等操作即可。

所以说，RDD的这种自动进行内存和磁盘之间权衡和切换的机制，就是RDD的弹性的特点所在。