MapReduce的思想核心是“分而治之”，适用于大量复杂的任务处理场景（大规模数据处理场景）。

* Map负责“分”，即把复杂的任务分解为若干个“简单的任务”来并行处理。可以进行拆分的前提是这些小任务可以并行计算，彼此间几乎没有依赖关系。
* Reduce负责“合”，即对map阶段的结果进行全局汇总。

MapReduce处理的数据类型是<key, value>

Map: (k1, v1) -> (k2, v2)

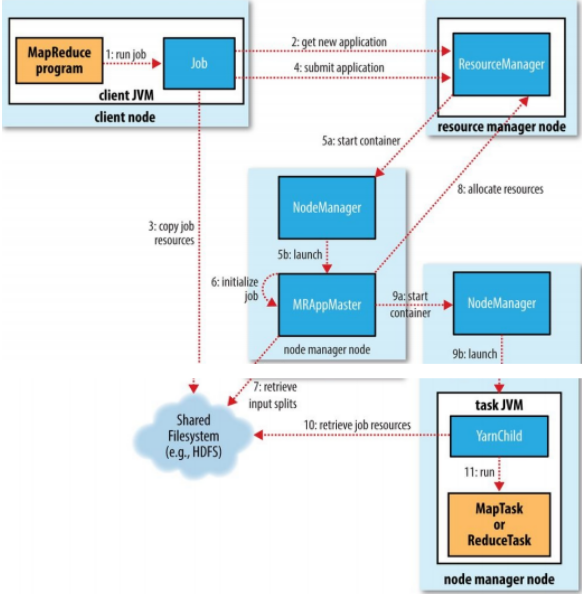
Reduce(k2, [v2,v2...v2]) -> (k3, v3)

参考：

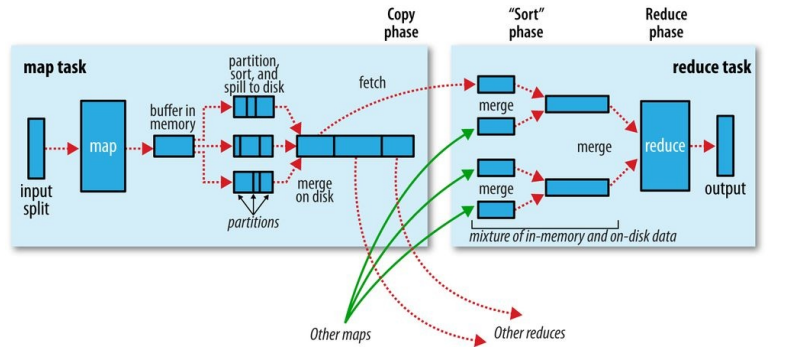
<https://www.cnblogs.com/luengmingbiao/p/11333618.html>

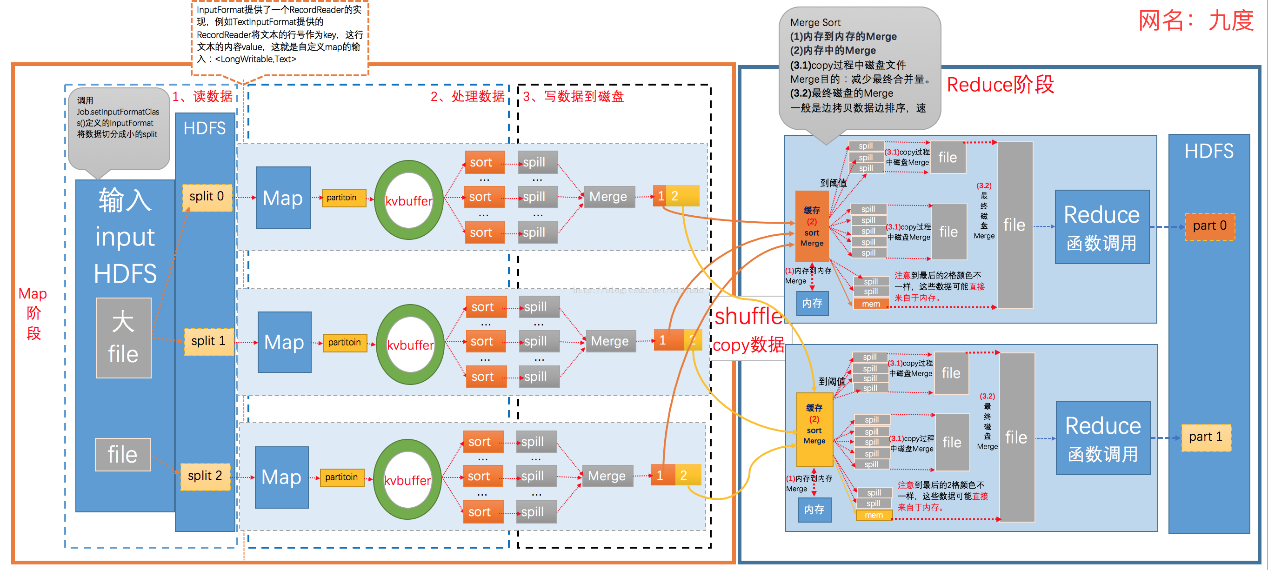
<https://www.jianshu.com/p/9e4d01b74600>

MapReduce在Yarn上运行机制



MapReduce执行过程





**MapReduce的八大步骤:**

Map阶段:

* 第一步: 通过FileInputFormat读取文件, 解析文件成为key, value对, 输出到第二步.
* 第二步: 自定义Map逻辑, 处理key1, value1, 将其转换为key2, value2, 输出到第三步.

Shuffle阶段:

* 第三步: 对key2, value2进行分区.
* 第四步: 对不同分区内的数据按照相同的key进行排序.
* 第五步: 分组后的数据进行规约(combine操作)，降低数据的网络拷贝（可选步骤）
* 第六步: 对排序后的数据, 将相同的key的value数据放入一个集合中, 作为value2.

Reduce阶段:

* 第七步: 对多个map的任务进行合并, 排序. 自定义reduce逻辑, 处理key2, value2, 将其转换为key3, value3, 进行输出.
* 第八步: 通过FileOutputFormat输出处理后的数据, 保存到文件.

1. FileInputFormat

默认是TextInputFormat

自定义InputFormat

public class MyInputFormat extends FileInputFormat

实现RecordReader方法

自定义RecordReader

public class MyRecordReader extends RecordReader

8. FileOutputFormat

默认是TextOutputFormat

自定义OutputFormat

public class MyOutputFormat extends FileOutputFormat

实现RecordWriter方法

自定义RecordWriter

public class MyRecordWriter extends RecordWriter