# 第一章

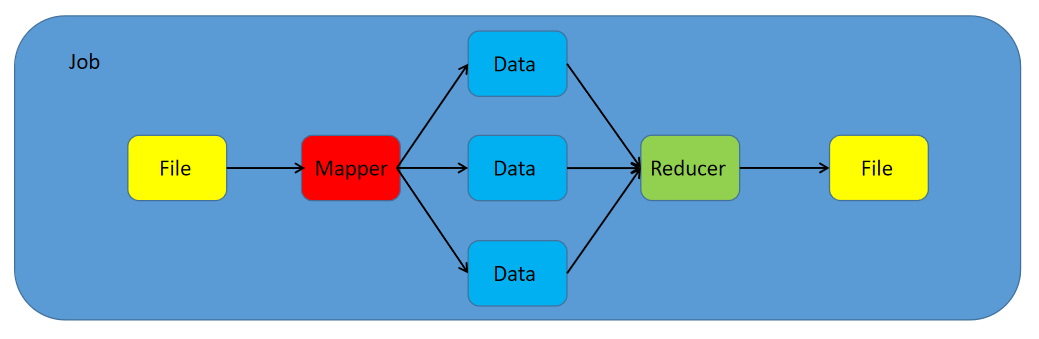
## Spark是什么？

Spark是一种基于内存的快速、通用、可扩展的大数据分析计算引擎。

## Spark VS MapReduce

### MapReduce处理机制

MapReduce设计的初衷主要是用于一次性数据计算，就是框架在处理数据的时候，会从存储设备中读取数据，进行逻辑操作，然后将处理的结果重新存储到介质中。

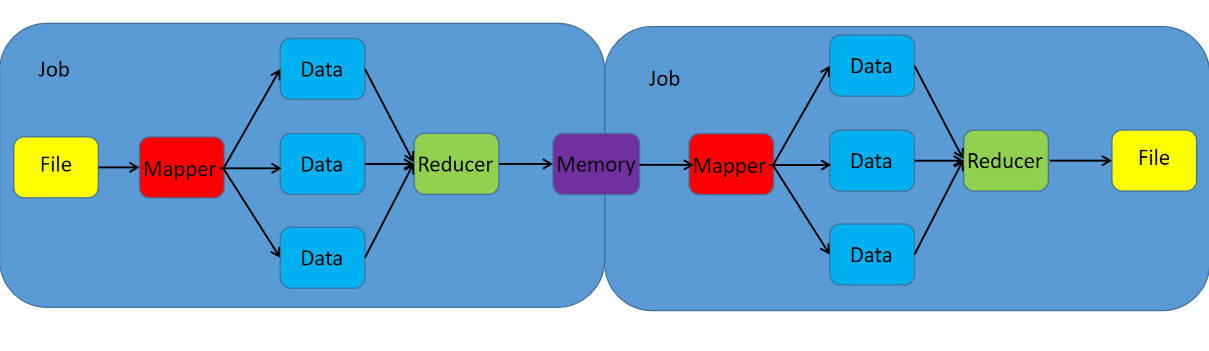


这种计算框架的缺陷在于当我们处理复杂逻辑时的性能是非常低的，因为MapReduce计算引擎中计算模型比较简单，它里面只有Mapper和Reducer，那么这样的话对于我们的上层应用来讲的话，就不得不想方设法去拆分我们的算法，甚至于在上层应用实现多个Job的串联。而且我们可以发现多个Job的之间的关联靠的是磁盘的交互，所以这种我们的磁盘IO会非常的影响性能。

### Spark处理机制

Spark相对于MapReduce提供了更加丰富的数据处理模型，且它可以基于内存来做数据集的多次迭代。如果先不考虑业务复杂度的问题，只考虑数据处理的过程，我们可以发现Spark把计算的结果放到了内存当中，为下次计算提供了更便利的处理方式，这种方式效率非常的高。这也是我们选择Spark而非MapReduce的原因。在内存计算策略和先进的调度机制帮助下，Spark可以更加快速的处理相同的数据集。

而由此带来的问题就是如果Spark部署在共享的集群当中，会遇到资源不足的问题，而且同时会对其他任务产生影响。所以从本质上来看，Spark并不适合和Hadoop堆栈的其他一些组件共同使用。



### Spark和MapReduce的差异

Spark和MapReduce的根本差异是多个作业之间的通信问题：Spark多个作业之间数据通信是基于内存，而MapReduce是基于磁盘。

## Spark核心模块