Hadoop大数据技术原理与应用的概述图（1张）

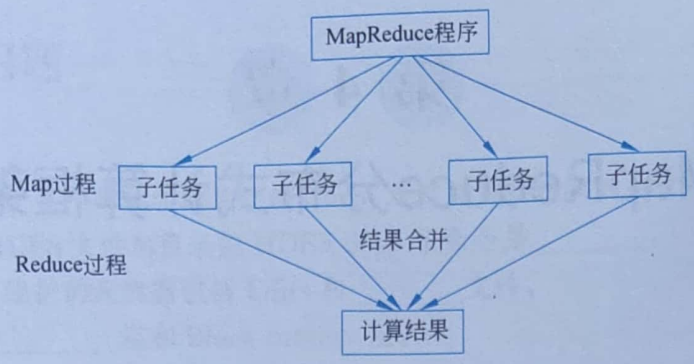

图 4-1 MapReduce 程序的执行过程

从图 4-1 可以看出,MapReduce 其实就是“任务的分解和结果的汇总”,即使用户不懂分布式计算框架的内部运行机制,但是只要能用 MapReduce 计算海量数据的思想清楚描述要处理的问题,就能轻松地在 Hadoop 集群上实现分布式计算功能。

4.2 MapReduce 编程模型

MapReduce 是一种编程模型,用于处理大规模数据集的并行计算。使用 MapReduce 执行大规模数据集计算任务时,计算任务主要经历两个过程,分别是 Map 过程和 Reduce 过程,其中,Map 过程负责对原始数据进行处理;Reduce 过程负责对 Map 过程处理后的数据进行汇总,并得到最终结果。这两个过程的简易模型如图 4-2 所示。

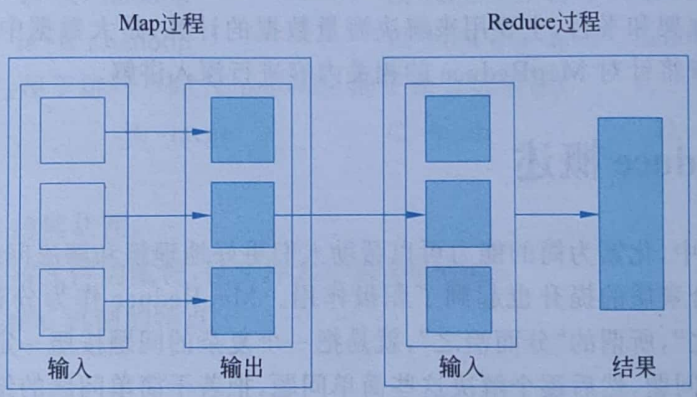

图 4-2 MapReduce 简易模型

从图 4-2 可以看出,Map 过程负责处理原始数据,Reduce 过程负责汇总 Map 过程的输出结果,从而得到最终结果。

MapReduce 的编程模型借鉴了计算机程序设计语言 LISt Processing(LISP)的设计思想,提供了 map()和 reduce()这两个方法,分别用于 Map 过程和 Reduce 过程,其中,map()方法的接收格式为键值对(<Key,Value>)的数据,其中,键(Key)是指每行数据的起始偏移量,也就是每行数据开头的字符所在的位置,值(Value)是指文本文件中的每行数据。使用 map()方法处理后的数据会被映射为新的键值对,作为 reduce()方法的输入,reduce()方法默认会将每个键值对中键相同的值进行合并,也可以根据实际需求调整合并规则。

MapReduce 简易模型的数据处理过程如图 4-3 所示。

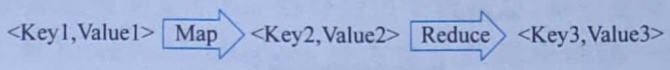

图 4-3　MapReduce 简易模型的数据处理过程

对于图 4-3 描述的 MapReduce 简易模型的数据处理过程，具体介绍如下。

(1) MapReduce 通过特定的规则将原始数据解析成键值对<Key1，Value1>的形式。

(2) 解析后的键值对<Key1，Value1>会作为 map()方法的输入，map()方法根据映射规则将<Key1，Value1>映射为新的键值对<Key2，Value2>。

(3) 新的键值对<Key2，Value2>作为 reduce()方法的输入，reduce()方法将具有相同键的值合并，生成最终的键值对<Key3，Value3>。

在 MapReduce 中，一些数据的计算可能不需要 Reduce 过程，也就是说，MapReduce 简易模型的数据处理过程可能只有 Map 过程，由 Map 过程处理后的数据直接输出到目标文件。但是，大多数数据的计算都是需要 Reduce 过程的，并且由于数据计算烦琐，因此需要设定多个 Reduce 过程。具有多个 Map 过程和 Reduce 过程的 MapReduce 数据处理模型如图 4-4 所示。

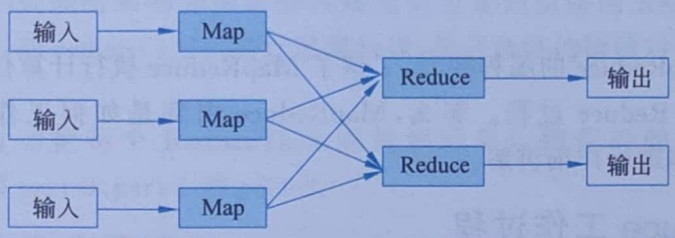

图 4-4　具有多个 Map 过程和 Reduce 过程的 MapReduce 模型

图 4-4 展示的是含有 3 个 Map 过程和 2 个 Reduce 过程的 MapReduce 模型，其中，由 3 个 Map 过程处理后的键值对会根据分区规则输出到不同的 Reduce 过程进行处理，默认情况下，分区规则是根据 Map 过程输出的键值对中的键的哈希值决定的。Reduce 过程是最后的处理过程，其输出结果不会进行第二次合并，也就是说，不同的 Reduce 过程都会将处理结果输出到单独的目标文件。

为了帮助读者更好地理解 MapReduce 编程模型，接下来通过一个经典案例——词频统计帮助读者加深对 MapReduce 的理解。

假设有两个文本文件 test1.txt 和文件 test2.txt，具体内容如文件 4-1 和 4-2 所示。

文件 4-1　test1.txt

```
Hello World
Hello Hadoop
Hello itcast
```

文件 4-2　test2.txt

```
Hadoop MapReduce
MapReduce Spark
```

使用 MapReduce 程序统计文件 test1.txt 和 test2.txt 中每个单词的出现次数，其实现流程如图 4-5 所示。

在图 4-5 中，MapReduce 读取文件 test1.txt 和 test2.txt 的每行数据，将每行数据映射为键值对的形式并输入 Map 过程。Map 过程将每行数据按单词拆分，并映射为新的键值对，如

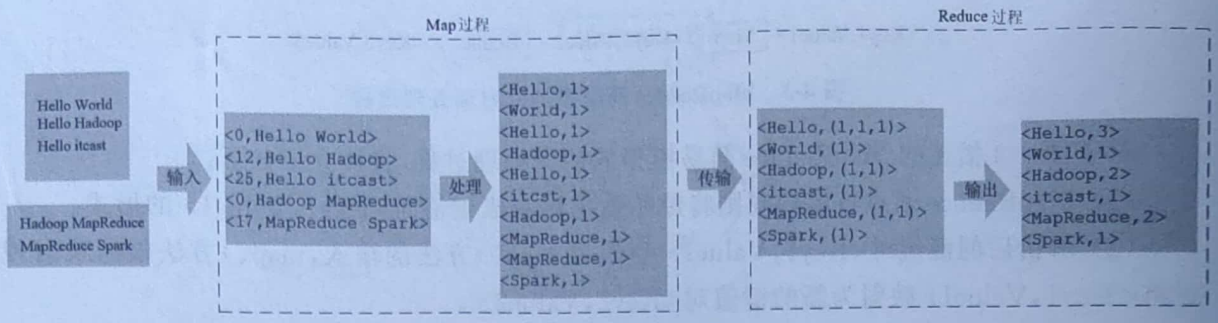

图 4-5 词频统计流程

<0,Hello World>映射为<Hello,1>和<World,1>,其中,键是单词的名称,值是单词出现的次数,标识为 1,以便后续统计单词的个数,新映射的键值对传输到 Reduce 过程后,按照相同键的值进行累加计算,最终输出每个单词的出现次数,如<Hello,3>表示单词 Hello 出现了 3 次。

4.3 MapReduce 工作原理

前面介绍了 MapReduce 的编程模型,了解了 MapReduce 执行计算任务时,计算任务主要经历了 Map 过程和 Reduce 过程。那么,MapReduce 内部是如何工作的呢?本节将针对 MapReduce 工作原理进行详细讲解。

4.3.1 MapReduce 工作过程

MapReduce 编程模型的开发简单且功能强大,专门为并行处理大规模数据量而设计,MapReduce 的工作过程如图 4-6 所示。

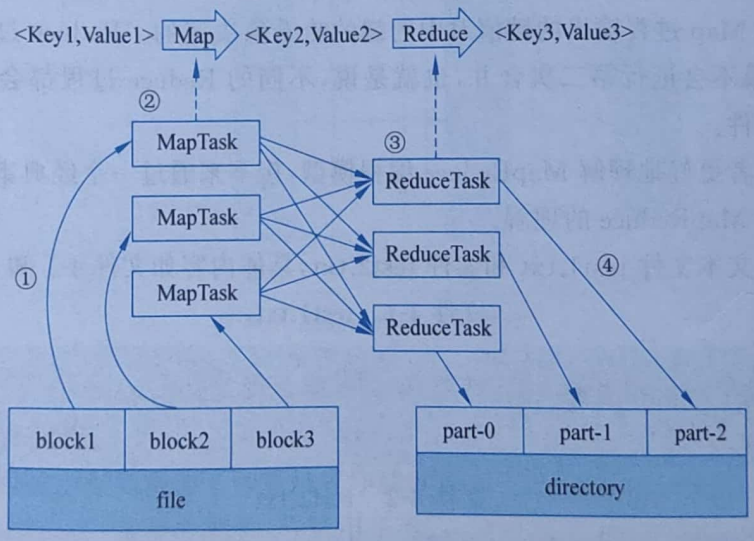

图 4-6 MapReduce 的工作过程

从图 4-6 可以看出,在 MapReduce 中的 MapTask 是实现 Map 过程的任务,ReduceTask 是实现 Reduce 过程的任务。MapReduce 的工作过程大致可以分为以下 4 步。

1. 分片(Split)和解析原始数据

输入 MapTask 的原始数据必须经过分片和解析操作。关于分片和解析操作的说明如下。