MapReduce 读后感

1. 概念

MapReduce是一个处理和生成超大数据集的算法模型，一个抽象模型，用户创建一个Map函数来处理一个基于key/value pair的数据集合，然后再创建一个Reduce函数合并所有的具有相同中间key值的value值。MapReduce 架构的程序能够在大量的普通配置的计算机上实现并行化处理。采用 MapReduce 架构可以使那些没有并行计算和分布式处理系统开发经验的程序员有效利用分布式系统的丰富资源。

用户定义的 Map 和 Reduce 函数都有相关联的类型。

1. 如何实现

MapReduce 模型可以有多种不同的实现方式，如何正确选择取决于具体的环境。

* 用户程序首先调用的 MapReduce 库将输入文件分成 M 个数据片度，。然后用户程序在机群中创建大量的程序副本。这些程序副本中的有一个特殊的程序--master。副本中其它的程序都是 worker 程序，由 master 分配任务。有 M个 Map 任务和 R 个 Reduce 任务将被分配，master 将一个 Map 任务或 Reduce 任务分配给一个空闲的 worker。
* 被分配了 map 任务的 worker 程序读取相关的输入数据片段，从输入的数据片段中解析出 key/value pair，然后把 key/value pair 传递给用户自定义的 Map 函数，由 Map 函数生成并输出的中间 key/value pair，并缓存在内存中。缓存中的 key/value pair 通过分区函数分成 R 个区域，之后周期性的写入到本地磁盘上。缓存的 key/value pair 在 本地磁盘上的存储位置将被回传给 master，由 master 负责把这些存储位置再传送给 Reduce worker。.当 Reduce worker 程序接收到 master 程序发来的数据存储位置信息后，使用 RPC 从 Map worker 所在主机的磁盘 上读取这些缓存数据。当 Reduce worker 读取了所有的中间数据后，通过对 key 进行排序后使得具有相同 key 值的数据 聚合在一起。由于许多不同的 key 值会映射到相同的 Reduce 任务上，因此必须进行排序。如果中间数据太大无法在内 存中完成排序，那么就要在外部进行排序。
* .Reduce worker 程序遍历排序后的中间数据，对于每一个唯一的中间 key 值，Reduce worker 程序将这个 key 值和 它相关的中间 value 值的集合传递给用户自定义的 Reduce 函数。Reduce 函数的输出被追加到所属分区的输出文件。
* 当所有的 Map 和 Reduce 任务都完成之后，master 唤醒用户程序。在这个时候，在用户程序里的对 MapReduce 调用才返回。

1. MapReduce的一些技巧
   1. 分区函数

MapReduce 的使用者通常会指定 Reduce 任务和 Reduce 任务输出文件的数量（R）。我们在中间 key 上使用分区函数来对数据进行分区，之后再输入到后续任务执行进程。

* 1. 顺序保证

我们确保在给定的分区中，中间 key/value pair 数据的处理顺序是按照 key 值增量顺序处理的。这样的顺序保证对每个分成生成一个有序的输出文件。

* 1. Combiner函数

我们允许用户指定一个可选的 combiner 函数，combiner 函数首先在本地将这些记录进行一次合并，然后将合并的结果再通过网络发送出去。

* 1. 输入和输出的类型

MapReduce 库支持几种不同的格式的输入数据。Reader 并非一定要从文件中读取数据，我们提供了一些预定义的输出数据的类型，通过这些预定义类型能够产生不同格式的数据。用户采用类似添加新的输入数据类型的方式增加新的输出类型。

* 1. 计数器

MapReduce 库使用计数器统计不同事件发生次数。这些计数器的值周期性的从各个单独的 worker 机器上传递给 master（附加在 ping 的应答包中传递）。master 把执 行成功的 Map 和 Reduce 任务的计数器值进行累计，当 MapReduce 操作完成之后，返回给用户代码。

1. 性能分析

排序程序输入数据读取速度小于分布式grep程序。：输入数据的读取速度比排序速度和输出数据写入磁盘速度要高不少，这是因为我们的 输入数据本地化优化策略起了作用 -- 绝大部分数据都是从本地硬盘读取的，从而节省了网络带宽。排序速度比输出数 据写入到磁盘的速度快，这是因为输出数据写了两份。

具有高效的backup任务。

1. 经验

MapReduce 的成功取决于采用 MapReduce 库能够在 不到半个小时时间内写出一个简单的程序，这个简单的程序能够在上千台机器的组成的集群上做大规模并发处理，这 极大的加快了开发和原形设计的周期。另外，采用 MapReduce 库，可以让完全没有分布式和/或并行系统开发经验的程 序员很容易的利用大量的资源，开发出分布式和/或并行处理的应用。MapReduce 重写了 Google 网络搜索服务所使用到的 index 系统。