Google 和 Hadoop MapReduce

MapReduce 基本模型

Map 和 Reduce 两个抽象函数

上升到构架后：combiner，partitioner

Google MapReduce

基本流程

1. 输入数据被划分为大小相同的数据块
2. 有主节点（Master）和负责 map 和 reduce 的工作节点（worker）
3. 用户的作业被提交给主节点
4. 主节点为作业程序配备 map 节点，将程序和数据传送给 map 节点
5. 主节点为作业程序配备 reduce 节点，将程序传送给 reduce 节点
6. 主节点启动 map
7. map 处理读取的数据并进行整理工作（combining，sorting），然后将中间结果存放在本地，通知主节点
8. 主节点等待所有 map 计算完成，启动 reduce，reduce 远程读取数据
9. reduce 节点汇总计算结果

通过 combiner 进行带宽优化

使用冗余 map 进行计算优化

partitioner 进行分区

MapReduce 分布式文件系统 GFS

Google GFS：基于分布式集群的大型分布式文件系统

多数据备份

GFS 基本构架有 GFS master，chunk server 和 client 等

GFS master 保存了一些元数据：命名空间，chunk 与文件的映射关系，chunk 副本的位置信息

GFS chunk server 保存实际数据，每个文件被划分为 chunk，在各个节点有 3 个备份

访问过程为

1. 应用程序将要获取的文件信息发送给 GFS master
2. GFS master 返回具体的 chunk server 的位置
3. 程序直接访问对应 chunk server

由于获取数据不经过 master 节点，避免节点成为访问瓶颈

分布式结构化数据表 BigTable

GFS 是文件系统，难以提供对结构化数据的访存，故在 GFS 上又设计了 BigTable 用于结构化数据的访存

BigTable 的数据通过行关键字，列关键字，时间戳定位数据

表中数据一律视为字符串

BigTable 的主服务器用于管理子表，以及负载均衡

数据存放在子表服务器中，客户端直接与子表服务器通信，子表由 SSTable 组成，一个 SSTable 就对应 GFS 中一个 chunk

Hadoop MapReduce 基本工作原理

JobTracker 与 TaskTracker 对应于 Google MapReduce 的 master 与 server

namenode 对应 GFS 的 master，datanode 对应 GFS 的 chunk server

主要组件：

- InputFormat：将输入分割成 InputSplit，每个 InputSplit 对应一个 map 任务
- RecordReader：将 InputSplit 读取成 kv pair
- Mapper：实现 map 任务
- Combiner：合并相同 Key 的键值对，减少通信开销
- Partitioner：决定给的的 kv pair 传到哪个 reduce 节点，传入的键值对会被排序
- Reducer：实现 reduce 任务
- OutputFormat：将输出写入到 HDFS

Hadoop 分布式文件系统 HDFS

模仿 GFS，是对 MapReduce 的底层支持

支持快速的顺序读取，支持一次写入多次读取，但不支持已写入数据的更新（可以 append）

基本构架是一个 namenode 存储元数据，管理 datanode，datanode 存储实际数据

每个 chunk 有 3 个备份