**读后感**

18301009 李卓航

Google在03至06年发表了著名的三大论文——GFS、BigTable、MapReduce，用来实现一个大规模的管理计算系统。2003年，Google发布了Google File System论文，是一个可扩展的分布式文件系统，用于大型的，分布式的，对大量数据进行访问的应用。其运行于廉价的普通硬件下，提供容错功能，是一种将文件分割成许多块，使用荣誉的方式储存在商用机器集群上的一种方式；2004年，其公布了MapReduce论文，讲述了大数据的分布式计算方式，主要思想是将任务分解然后再堕胎处理能力较弱的计算节点中同时处理，并将计算结果合并从而完成大数据处理的一种思想；最后是2006年的Bigtable论文，现如今的很多非关系型数据库都受到了其启发，例如HBase，Cassandra等。距离谷歌发布三篇论文至今已有十六年，在这段时间里，计算机技术和网络技术呈现飞速发展，随着硬件的发展和普及，技术发展呈现井喷趋势，例如大数据，人工智能，云计算等高科技服务，也包括例如饿了么美团这种全国服务平台和近期兴起的共享项目，都离不开谷歌的三大论文的支持，因此谷歌的三篇论文可以说是为这个时代的计算机技术奠定了基础，具有引领时代的作用。

首先是最先发布的Google File System(GFS)，其设计初衷是为了满足Google迅速增长的数据处理需求。GFS技术和思想应用的背景如下，首先，组件失效被认为是常态事件，GFS所包括的成百上千台廉价设备组装的存储机器在任何时间上都有可能发生组件无法工作或无法从失误中恢复的情况因此，持续的监控，错误侦测与处理以及自动恢复机制是需要在GFS中集成实现的。第二，所处理的文件量非常巨大，在大数据时代普遍为GB，TB级别，使得对上亿个小文件的组织方式不再适用。第三，绝大部分文件的修改是在末尾追加而不是覆盖原有数据，因此客户端对数据块缓存是没有意义的，最后，应用程序和文件系统的API协同设计提高了整个系统的灵活性，通过原子性记录追加操作，保证多个客户端的操作同时进行而不会产生二义性，在不需要额外同步操作下保证了数据的一致性。目前已部署多套GFS集群，具有TB级别硬盘空间，支持多用户不间断频繁访问。一个GFS集群包括一个单独的Master节点，多台chunk服务器，大多是运行着用户级别服务进程的Linux机器。GFS储存的文件被分割成了固定大小的chunk，每个chunk有着自己唯一的64位标识，默认下一般备份三份。在读写操作时，chunk服务器通过指定的chunk标识和字节范围来进行读写操作。Master节点管理着所有文件系统元数据，负责发送指令到各个chunk服务器并接受服务器的状态信息。客户端和chunk服务器都不需要缓存文件数据。同时GFS系统维护一个操作日志，包含了关键的元数据变更记录，是唯一的持久化储存记录，可用于判断同步顺序操作的逻辑时间线，因此十分重要。在master服务器产生宕机等灾难时，需通过操作日志恢复，通过重演操作日志把文件系统恢复到最近状态。为了使重演时效率高，需要保证日志较小，因此需要在日志增长到一定大小时做一次checkpoint(检查点)，发生灾难时通过最近的检查点文件与操作日志文件共同快速恢复至最近状态。在删除方面，GFS在文件删除后不会立刻收回物理空间，其采用的惰性回收策略，旨在文件和chunk级的常规垃圾及收集时进行。GFS文件系统的设计保证了在有大量的并发读写操作时能够提供较高的合计吞吐量，实现了我们对存储的需求。

MapReduce论文提供了一种用于大规模数据集（大于1TB）的并行运算的一种编程模型，通过此架构可以使得程序员充分有效的利用分布式系统的丰富资源。其编程原理是，利用一个输入的K/V pair 集合来产生一个输出的K/V pair 集合，通过Map（映射）和Reduce（归约）两个函数进行表达实现。Map函数接受一个输入的K/V pair 值，然后产生一个中间K/V pair 值的集合，并把所有具有相同中间key值得中间value值集合在一起传递给Reduce函数。Reduce函数接受一个中间key值和相关value值得集合，将其合并，形成一个较小的value集合，用来保证所有映射的键值对中的每一个共享相同的键组。通过Reduce方法迭代大大精简了大容量的存储体积，使得大规模数据集的并行运算成为可能。具体操作流程如下，1.用户调用MapReduce库将输入文件分成M个数据段，大小一般选取16MB-64MB，同时在集群中创建大量的程序副本；2.这些副本中有一个特殊的程序Master负责分配程序，其余worker程序由master分配任务。共有M个Map任务和R个Reduce任务被分配，每一个对应一个空闲的worker；3.分配到Map程序的worker程序读取相关输入数据片段，解析出K/V pair并传递给Map函数，再由Map函数生成并输出中间K/V pair缓存在内存中；4.缓存中的K/V pair在本地磁盘上的存储位置回传给Master，再由master回传给Reduce worker;5.Reduce worker读取到所有中间数据后，通过对key排序使得具有相同key值的数据聚合在一起；6.Reduce worker程序遍历排序后的中间数据，将值回传给用户自动逸的Reduce函数；7.所有Map和Reduce任务完成后,master唤醒用户程序，任务完成。对于错误处理，master会周期性的将数据写入磁盘，也就是GFS中曾出现过的检查点，当master失效时，可以从最近的检查点进行快速恢复和重做。MapReduce编程模型的实现有重大意义，首先，通过将并行处理，容错处理，负载均衡等细节难题进行封装的方式，使得MapReduce库易于使用，及时没有并行开发经验或分布式开发经验的程序员也能很好的应用他，其次，MapReduce模型可用于很多不同问题，例如数据挖掘，机器学习等等，第三，在大规模集群上的成功运行使得计算资源得到了有效地利用。

最后是BigTable分布式数据存储系统，是Google为其内部海量的结构化数据所开发的云存储技术，也是所有云时代分布式存储系统的开发蓝本，在多个产品项目上得到使用。通常BigTable处理的数据为PB级别，分布在数千台普通服务器上。BigTable的主要目标，也是已经实现的目标是：适用性广泛，可扩展，高性能和高可用性。首先介绍了BigTable的数据模型，BigTable是一个稀疏的，分布式的，持久化出巡的多维度排序Map,以行关键字，列关键字，时间戳作为索引，每个Map的value都是未解析的byte数组。行关键字可以是任意字符串，最大支持64kb长度，BigTable通过行关键字的字典序来进行数据的组织，对同一行的关键字进行读写操作均是原子性的，就不会产生数据的二义性。每个行可进行动态分区，每个分区为一个“Tablet”，使得当只读取行中很少几列数据时的效率很高。列关键字组成的集合叫做列族，是访问控制的基本单位。同一列族下的数据一般为相同类型，访问龙之，磁盘和内存的使用统计都是在列族层次进行的。时间戳的功能主要是对同一份数据的不同版本加以区分和索引，使得其在数据恢复或其他需要辨别版本时可以快速定位到所需版本。BigTable时间戳的类型为64为整形，其最精确可达到毫秒级别。系统会时间戳赋值，用户也可以选择手动为其进行赋值操作，一般数据版本按照时间戳的倒序排序，即最新的数据排在最前面。过多数据的冗余会导致资源的浪费，因此可以在每个列族上设置两个参数，通过这两个参数BigTable可以只保留某一时间单位内的数据，例如“仅保留一天前的数据”，那么从现在计算起一天前的数据就会被实时删除，以减轻其负担。BigTable提供和建立了相关的API函数，方便程序员使用操作。BigTable为许多非关系型数据库提供了设计思路，例如Hbase等等，截至论文出版已有60多个项目使用BigTable技术了，且当数据膨胀时，只需要简单的添加机器，就可以扩展系统的数据承载能力。