**谷歌三篇重要论文读后感**

花了部分时间大概将三篇论文看了一遍，感觉处理问题需要系统性，全面性的考虑，了解了作为互联网公司的龙头企业对整个存储系统的优化和各种独创性，深深的感觉到自己现在都是小打小闹，对存储的各种操作头停留在婴儿阶段，应该走出去了解更多的存储技术，为以后的工作打下良好基础。

MapReduce，一个处理和生成超大数据集的算法模型。

有关MapReduce 任务的分配与调度，感觉就是map读取输出，reduce收集合并。在理解掌握的过程中，看到不管是用做拉沙还是纸牌的例子，它的一种完成任务的思路，它的思想是一致的。他们之间重合的共性，就是结构，就是灵魂。

另外，MapReduce的减少“落伍者”机制，让我想起c中的free。Free在指针中，释放内存。我以为它是手动的一种控制机制，减少缓存导致运算变慢的情况，而MapReduce中的backup，则是高阶版，调优机制，通过那仅仅多几个百分点的计算资源，就可以减少总处理时间，而且是效果显著

同时，MapReduce中的拓展功能，分区，顺序，输入输出，从我目前接触到的语言c中，也大概能理解那么一丢丢意思，有那么一个概念在脑子里。总之，我觉得MapReduce就是通过map，reduce两个函数构成的体系来对任何接收到的过大数据的任务进行处理。分而治之，逐步瓦解。其实最开始，在没有查询，专业的了解Goole-MapReduce的时候，我就觉得map和reduce就是两个函数。它们所构成了一个体系，一个结构。其中，我看到的标志性词语就是“库”。

尽管作为一个对云计算一窍不通的人，但是我还是从论文中学到了并尽可能的理解了gfs大体的构建。所以下面主要从gfs的构成，如何有效读取和写入，master服务器稳定和数据恢复等方面浅谈自己的读后感想。

gfs的构建，一个gfs集群包含一个单独的master节点，多台chunk服务器，并且同时被多个客户端访问。在gfs中储存的文件都会被分割成固定大小的chunk，并且为了方便识别储存的数据，会在创建每一个chunk时，分配一个不变的，全球唯一的64位chunk标识，并且以linux文件的形式保存在本地硬盘上。单一的master节点会大大简化gfs的设计，因为客户端通过master找到需要询问的chunk服务器，客户端将这些元数据缓存一段时间后，后续会直接与chunk服务器进行数据读取。对于元数据都储存在master的内存中，有三种主要类型的元数据，文件和chunk的命名空间，文件和chunk的对应关系，每个chunk副本的存放地点。关于我自己的理解是这些类型的元数据会方便实现读取，保持master服务器与chunk同步状态，不会导致master服务器崩溃。

文件的读取和写入，对于读取和写入效率，当读取和写入的客户机增加时，多个客户机同时读取和写入的几率也增加，导致整体的读取和写入效率降低。感觉就像我们平时用网线，当多个人用同一条网线时，你的网速就会变得很慢，放视频打游戏动不动就卡机。gfs系的读取有两种操作，一种是大规模的流式读取，大规模的流式读取通常一次读取数百KB的数据，甚至更多。而另外一种是小规模的随机读取，并且在gfs中对于记录追加通过chunk和chunk的副本进行操作，保证至少有一次原子的写入操作成功执行。

master服务器的稳定和数据恢复备份。在gfs中为了保持master服务器的可靠性，master服务器的状态也要复制，master服务器的所有操作日志和checkpoint文件都被复制到多台机器上，其中操作日志是元数据的唯一持久化存储记录，在master服务器在灾难恢复时，通过重演操作日志把文件系统恢复到最近的状态，而checkpoint是对数据库做一次快照的行为，读取checkpoint文件以及重演checkpoint之后的有限个日志文件就能恢复系统。而且在论文中可以了解到快照是对一个文件或者目录树做一个拷贝，可以瞬间完成，并且不会对同时进行的其他操作造成干扰，所以通过快照，用户可以创建一个巨大的数据集的分支拷贝，就如同备份一样，可以直接提交或者回到拷贝当前状态。

Google-Bigtable是一个分布式的结构化存储系统，被Google用来存储很多项目的数据，这些项目对Bigtable需求差异很大，Bigtable还是能提供很灵活的服务，提现了其功能的强大。论文对Bigtable的主要组件及功能做了详细介绍：“Bigtable 包括了三个主要的组件：链接到客户程序中的库、一个 Master 服务器和多个 Tablet 服务器。针对系统工作负载的变化情况，BigTable 可以动态的向集群中添加（或者删除）Tablet服务器。 Master 服务器主要负责以下工作：为 Tablet 服务器分配 Tablets、检测新加入的或者过期失效的 Table 服务器、对Tablet服务器进行负载均衡、以及对保存在 GFS上的文件进行垃圾收集。除此之外，它还处理对模式的相关修改操作，例如建立表和列族。每个Tablet服务器都管理一个Tablet的集合（通常每个服务器有大约数十个至上千个 Tablet）。每个Tablet服务器负责处理它所加载的Tablet的读写操作，以及在Tablets过大时，对其进行分割。”，从这些可以大致了解到Bigtable的工作原理。