尽管不能完全理解，但通篇阅读后，还是有所体会，三篇论文在我的脑海里留下了一些关键字，比如，海量、TB级、PB级、并行化、分布式……

谷歌的三篇论文介绍了Google Bigtable, Google MapReduce和Google File-System三个重要工具。它们有一个共同点——分布式系统。而这个分布式就是将一个业务分拆多个子业务，化整为零，分别放在在不同的服务器上。经过输入、分块、整合、输出四个部分，好比一个大的任务，分配给多个小组共同分工、合作，最后汇总完成。

论文中频繁出现词语海量、大规模数据、超大数据集、TB级、PB级……这些表示数量的词汇都可以很明显地向我们转达出：Google MapReduce ,Google File System,Google Bigtable，三个系统或是模型均是为了处理“海量”的数据而出现和设计的。而这个“海量”之所以打上引号，就是为了强调数据的庞大，一般的大、巨大、庞大，不足以描述其数据量大的程度，应该只有用数量级来计量才足够。

在我的理解中，MapReduce 的设计主要体现出了一种“化整为零”再“化零为整”的思想。比如，论文中提到，“用户程序首先调用的MapReduce库输入文件分成M个数据片度”。基本上都是采用了分割和拆分的方法去处理大量的数据，把问题分解成为大量的“小”任务，很好起到化繁为简的作用，可以使很多复杂的问题的得到简单的解决。MapReduce可以对很多复杂的问题进行简单的解决，这对处理一些现实问题非常有帮助，如空气污染指数、城市外来人口统计等。而它是如何做到的呢?这也要归功于分布式原理。对于一个复杂的问题，并不是直接丢给一台高性能的电脑去做而是分配给许多台低性能的电脑，不仅提高了效率也提高了容错率。可以说到目前为止，MapReduce最成功的应用就是重写了Google网络搜索服务所使用到的index系统。因此减少了针对网络输入量，也解决了由于机器失效导致的数据丢失问题。

对于Google File System ，可以在廉价的普通硬件设备上运行，为大量客户机提供高性能服务。它满足了我们用户对储存的需要，同时它是又许多便宜的普通的组件组成，而且即使在数个G大小的文件也是可以有效的管理。最后还要提到GFS的接口和架构，在GFS的接口中，它提供了一套类似传统文件系统的API接口，可以按照以最低的成本创建一个文件或目录。在GFS的架构中，它以库的形式被连接到用户程序中，这实现了对数据进行简单的操作。GFS与传统文件系统明显的差异是它不需要很强大的硬件支撑就可以进行大规模的数据处理。

而Google Bigtable ，则是一个分布式的结构化数据存储系统，同样是被设计用来处理海量数据（通常是PB级）。它使用了很多数据库的实现策略。当储存数据时，Bigtable把已经储存的数据都看作字符串，但是他不会去解析这些字符串，而是把这种结构化的字符串串行化到字符串中。因此用户可以准确了解到这些数据的位置，给用户省去了找这些数据很大的麻烦。最令我感兴趣的是他的构件，他是在建立在几个Google基本构件的基础上的。通过这几个构件当你每次需要查找数据时，只需要通过一次硬盘搜索就可以在硬盘读取相应的数据块。它不同于Boxwood直接新建文件系统，数据库等高级服务的基础构件，而是直接为客服的需求提供服务，因此收到了广大用户的喜爱。但是他也有一定的缺点，比如所，他的读取可能不太稳定，实时性也不太好，可能这就是他存在的美中不足的地方。

Google的三篇技术论文意味着世界进入大数据时代，数据计算速度的极大提升，也为各个领域提供了新思路。通过与各个领域的结合与应用，相信未来，大数据与人工智能一定有更好的利用和发展。