《MapReduce：Simplied Data Processing on Large Clusters》读后感

文章由Jeffrey Dean 和 Sanjay Ghemawat一同撰写，这两位是谷歌唯二的11级工程师，而且两位结交了深厚的友谊。桑杰和杰夫合作多次重构谷歌搜索引擎核心算法，开发了MapReduce也就是这篇paper详细介绍的技术。这种技术能够把全球各处的谷歌服务器整合成“一台行星那么大的计算机”，在科技领域实现了很大突破，阅读此篇文章后，有不解，有敬佩，还有很多感悟。

通过文章，我明白了MapReduce的本质就是一个编程的模型，主要分为Map和Reduce两个部分。Map就是把庞大的任务细分成为大量的小型任务群，之后把这些任务分配到不同的服务器上运行。而Reduce就是负责把已经细分的任务结果重新合并成为想要的完整结果。在五年的时间里，作者和Google的许多人已经实现了数以百计的专门目的而写的计算来处理大量的原始数据，例如，爬行的文档，web请求日志等等。为了计算各种类型的派生数据，比如倒排索引，web文档的图结构的各种表示，每个主机上爬行的页面数量的概要，每天被请求数量最多的集合等等，所以说MapReduce由此而生。文章中又说以这种方式写的程序能自动的在大规模的普通机器上实现并行化，运行时系统关心这些细节包括分割输入数据，在机群上的调度，机器的错误处理，管理机器之间必要的通信，这样就可以让那些没有并行分布式处理系统经验的程序员利用大量分布式系统的资源。

对于整个领域的技术，我并没有了解很多，所以paper中的一些专业术语很多都是一知半解，但我知道这种技术的优势确实很明显，对科技领域的贡献也很大，致敬伟大的科学家！

《The Google File System》读后感

Google File System 简称GFS，是谷歌设计并实现了的文件系统。为大量客户提供了高性能的服务。

GFS最开始的设计思路是：

1.组件失效被认为是常态事件，而不是意外事件；

2.以通常的标准衡量，我们的文件非常巨大；

3.绝大部分文件的修改是采用在文件尾部追加数据，而不是覆盖原有数据的方式；

4.应用程序和文件系统 API 的协同设计提高了整个系统的灵活性。

由此，Google在设计中做出了如下的假设：

1.系统由许多廉价的普通组件组成，组件失效是一种常态。系统必须持续监控自身的状态，它必须将组件

失效作为一种常态，能够迅速地侦测、冗余并恢复失效的组件。

2.系统存储一定数量的大文件。我们预期会有几百万文件，文件的大小通常在 100MB 或者以上。系统也必须支持小文件，但是不需要针对小文件做专门的优化。

3.负载通常包含两种读：大型的流式读（顺序读），和小型的随机读。前者通常一次读数百KB以上，后者通常在随机位置读几个KB。

4.系统的工作负载还包括许多大规模的、顺序的、数据追加方式的写操作，系统支持小规模的随机位置写入操作，但是可能效率不彰。

5.系统必须高效的、行为定义明确的实现多客户端并行追加数据到同一个文件里的语意。

6.高性能的稳定网络带宽远比低延迟重要。

GFS系统的节点可以分为三种角色

1.GFS Master（主控服务器）

2.GFS ChunkServer（CS，数据块服务器）

3.GFS客户端

GFS主控服务器是整个文件系统架构的核心，他保证了chunk与文件之间的映射和文件的命名空间等等。GFS存储的文件被分成了固定大小的chunk，由GFS Master也就是主控服务器分配一个全球唯一的chunk句柄，并储存三份。客户端是GFS提供给应用程序的访问接口，它是一组专用接口，以库文件的形式提供。客户端访问GFS时，首先访问主控服务器节点，获取与之进行交互的ChunkServer信息，然后直接访问ChunkServer，完成数据的存取工作。

1.客户端向主控服务器请求chunk每个副本所在的ChunkServer。

2.主控服务器返回客户端主副本和各副本所在的ChunkServer的位置信息，客户端将缓存这些信息供以后使用。

3.客户端将要追加的记录发送到每一个副本。GFS中采用数据流和控制流分离的方法，从而能够基于网络拓扑结构很好地调度数据流的传输。

4.当所有副本都确认收到了数据，客户端发起一个写请求控制命令给主副本

5.主副本把写请求提交给所有的备副本。

6.各备副本成功完成后应答主副本。

7.主副本应答客户端，如果有副本发生错误，将出现主副本写成功但某些备副本不成功的情况，客户端将重试。

Google文件系统构建在廉价服务器之上。它将服务器故障视为正常现象，通过软件的方式自动容错，在保证系统可靠性和可用性的同时，大大降低系统的成本。GFS 成功的实现了我们对存储的需求，无论是作为研究和开发的存储平台，还是作为生产系统的数据处理平台，都得到了广泛的应用。

# 《FastSGG: ESGG Using a Degree Distribution Generation Model》读后感

# 文章提到在如今大数据时代，随着社交网络的普及，需要大规模的社交图来评估各种社交网络分析任务的算法，在本文中，提出了一种高效且应用广泛的社交图生成器FastSGG。FastSGG根据描述目标社交图特征的用户定义配置生成社交图，这是在各种应用中生成图的灵活方式。生成方法包括两个主要步骤：确定源顶点的出度和确定目标顶点以构建边。为了加速图生成过程，提出了度分布生成（D 2 G）模型。只要给定概率密度函数或概率质量函数，D 2 G 模型就可以生成不同度分布图的通用模型。大量的实验结果表明，FastSGG 可以生成具有小世界属性、幂律度分布和社区结构的高质量社交图。此外，FastSGG 生成图的速度至少比最先进的图生成器快四倍。此外，FastSGG 的峰值内存使用量不到最先进方法的七分之一。

# 为什么说FastSGG是优秀的，是因为它克服了大规模社交图谱生成存在的三大技术挑战。首先，生成模型和算法应该足够高效以生成万亿级图。其次，生成算法应该具有较低的时间和空间复杂度。最后，图生成和社区生成易于集成以生成具有社区结构的社交图。这三点，FastSGG都能通过其独特的技术优势实现。度分布生成模型为使用节点的度概率生成图提供了原生支持。它是一个通用模型，这意味着如果给定 PDF 或 PMF，我们可以使用该模型生成遵循不同度分布的图形。使用此模型，确定源顶点的出度或目标顶点只需要 O(1) 时间。

# 文章通过很多数据证实了FastSGG模型的高效性，实用性与创新性，让我真切地感受到创新思维的重要性，模型开发对于社会发展的重要性。