说到Hadoop的起源，我们不得不提到谷歌的三巨头:谷歌FS、MapReduce和 BigTable。谷歌虽然没有发布这三款产品的源代码，但发布了这三款产品的详细设计论文，为全球流行的大数据算法奠定了基础。

2003年，谷歌发布了GFS论文，这是一个可扩展的分布式文件系统，适用于访问大量数据的大型分布式应用程序。它运行在廉价、通用的硬件上，并提供容错功能。基本上，文件被分割成块，并冗余地存储在业务机器集群上。

集群由一个主服务器和多个chunkserver组成。每个节点都是运行用户级服务器进程的普通Linux机器。在GFS下，每个文件被分割成固定大小的块。对于每个块，主机生成一个全局唯一的64位块句柄标志，该标志将根据块创建的时间在以后不会更改。

对于chunkserver，这些块存储在本地磁盘的Linux文件系统中，根据块句柄和字节间隔对Linux文件系统进行读写操作。为了提高可靠性，每个块都备份在不同的ChunkServer 上。默认情况下，保存三个备份。为了避免单master单点故障，谷歌允许在一个GFS集群中存在多个备份主节点。当主服务器出现故障时，通过Chubby lock 服务选择一个Backup master作为主服务器。

GFS由一个主服务器和大量的chunkserver组成。谷歌设置了一个主服务器来保存目录和索引信息，以简化系统结果并提高性能，但这可能会给主服务器带来单点故障或瓶颈。为了消除主要单点故障，谷歌将每个chunk 设置为大块(64M)。因此，由于代码访问数据的本地特性，应用程序端和主端之间的交互减少了，主要的数据传输是在应用程序端和 Chunkserver之间。

主服务器管理元数据，数据直接在客户端/ChunkServers 之间传输。文件被分成块(通常为64MB)。为了安全起见，这些块在三台机器上复制了三次。

此外，所有主信息都存储在内存中，并在启动时从 ChunkServer中检索。这提高了主机的性能和吞吐量，还使得在主机停机时更容易将备份]机器切换到主机。客户端和ChunkServer都不单独缓存文件数据，只是Linux文件系统自己的缓存。

MapReduce和 BigTable都是基于GFS的。三种基本的核心技术构建了完整的分布式计算体系结构。

2004年发表的 MapReduce描述了大数据的分布式计算方法。其主要思想是将任务分解并同时处理到多个处理能力较弱的计算节点上，然后将结果组合起来完成大数据处理。

传说中，谷歌用它来计算他们的搜索索引。Mikio L. Braun认为它应该这样工作：谷歌将所有爬行的页面放到它们的集群中，然后每天使用MapReduce重新计算它们。

MapReduce是一种用于处理和生成大型数据集的编程模型。用户指定一个映射函数来处理键/值对以生成键/值对的中间集，以及一个扩展函数来组合与同一中间键相关联的所有中间值。许多现实世界的任务都在这个模型中表示。用这种函数式风格编写的程序可以自动并行化，并在一集群机器上执行。运行时系统负责划分输入数据的细节，通过执行一系列机器调度程序来处理机器故障，以及管理所需的机器内部通信。这使得没有任何并行和分布式系统经验的程序员可以轻松地利用大型分布式系统的资源。

MapReduce由三种不同类型的服务器组成:Master、MapServers和 Reduceservers。master将用户任务分配给MapServers和 Reduceservers。它还跟踪任务的状态。Mapsevers 接受用户输入，对其执行映射操作，并将结果写入一个中间文件。Reduceservers服务器获取映射服务器生成的中间文件，并对它们进行简化。

Mapreduce 由 Map和 Reduce两部分组成，它们是由Lisp派生而来的。Map是将指令分发给多个worker 的投影，reduce是将Map worker计算的结果合并的协议。Mapreduce 使用GFS来存储数据。

Bigtable于 2006年发布，激发了无数 NoSQL 数据库的灵感，如 Cassandra、HBase等。Cassandra的架构有一半是根据 Bigtable 建模的，包括数据模型、sstable和预写日志(另一半是根据Amazon的 Dynamo 数据库建模的，使用点对点集群模型)。

谷歌在 BigTable中存储结构化数据。BigTable是一个具有容错和自治的大型系统，包括tb的内存和 PB的磁盘存储空间。它每秒可以处理数百万次读写。BigTable是一个建立在GFS之上的分布式哈希系统。它不是关系数据库，不支持连接或sql类型的查询。提供按键值访问结构化数据的查找机制。每个数据都存储在一个单元格中，可以使用行键、列键或时间戳访问该单元格。每行存储在一个或多个片剂中。一个Tablet是64KB，由一系列称为sstable 的数据结构组成。

BigTable有三种不同类型的服务器:主服务器将tablet分配给tablet服务器,跟踪位置，并在需要时重新分配服务。Tablet服务器处理来自Tablet 的读写请求。当tablet超过大小限制(通常是100Nb-200MB)时，他们就会将其分开。当主服务器故障时，Chubby锁服务也会选择BackupMaster 作为主服务器。

BigTable是建立在GFS 和 MapReduce之上的。每个Table都是一个多维稀疏图

为了管理大型表，这些表被按行分解，它们获得的数据被称为tablet。每台tablet的内存大小在100-200 MB左右，每台机器可以存储大约100个内存。底层的体系结构是GFS。由于GFS是一个分布式文件系统，所以可以通过 tablet的机制来实现负载均衡。例如，一个频繁响应的表可以移动到另一个空闲的机器并快速重建。

当这三篇文章发表时，谷歌几乎都集中在上面的搜索引擎上，谷歌搜索引擎的核心是PageRank算法，而url之间的引用数是PageRank的重要组成部分。所有MapReduce是直接计算参考页面的数量,所以他们只是喂爬数据映射文件作为输入,MapReduce得到结果时,他们会参考相关页面的数量,他们进入网页排名作为一个参数来得到他们想要的答案。

另外，由于夜间流量较低，可以在夜间运行MapReduce而不影响用户体验。这解释了为什么谷歌能够忍受如此长的计算时间。

简而言之，MapReduce很简单是因为它的计算很简单，但它计算量很大，所以它使用了分布式架构。MapReduce之所以慢是因为谷歌确实不需要它快。因此 MapReduce可能只适用于谷歌和业务场景适合的公司。

数据库不仅提供数据存储，还提供数据读取。这里的存储和读取，不仅完整一致，而且快速。如果我们直接使用GFS，我们将无法获得数据的完整视图，并且需要像查找文件一样查找数据，这样指定文件路径将效率低下，而且在编程上不灵活。另一方面，GFS根本不会优化数据写入。