

## 论文读后感

19301019 武仪

### google mapreduce论文读后感

Google mapreduce是老师要求我们看的三篇论文之一。这篇论文主要介绍了mapreduce编程模型，以及mapreduce架构的程序是如何在普通配置的计算机上实现并行处理。我阅读完论文后总结出以下内容。

mapreduce编程模型所执行的分布式计算会以一组键值对作为输入，输出另一组键值对，用户则通过编写 Map 函数和 Reduce 函数来指定所要进行的计算。由用户编写的Map 函数将被应用在每一个输入键值对上，并输出若干键值对作为中间结果。之后，MapReduce 框架则会将与同一个键 **K** 相关联的值都传递到同一次 Reduce 函数调用中。同样由用户编写的 Reduce 函数以键 **K** 以及与该键相关联的值的集合作为参数，对传入的值进行合并并输出合并后的值的集合。

MapReduce是一个相当简单的计算模型，它尝试将所有的计算任务都拆分成基础的Map和Reduce，以此降低实现的复杂度。但是，这恰恰提高了编程逻辑的复杂度。使用MapReduce实现Join功能的代码，往往十分地巧妙灵活。但是看似巧妙的背后，是模型过于简单而导致复杂度转移到了代码逻辑的层面。另一方面，MapReduce的程序类似于批处理程序，需要完整的输入程序才能开始运算，而且每次运算都要至少写入两次磁盘。这就导致每次运算都要等待很长的时间，完全不能实现需要快速响应的业务场景的需求。

以上两个方面，一个引出了支持类SQL的计算工具，另一个引出了支持流式计算的工具，而这两个特性正是今天流行的计算工具的热点。

### Google\_Bigtable论文读后感

这篇论文涉及到的方面不单单介绍了bigtable的基本定义,还介绍了bigtable的应用以及与bigtable相关的数据模型等一系列内容

首先，我们要先知道什么是Bigtable。Bigtable 是一种压缩的、高性能的、高可扩展性的，基于 Google 文件系统（Google File System，GFS）的数据存储系统，用于存储大规模结构化数据，适用于云端计算。Bigtable 的设计是为了能可靠地处理 PB 级的海量数据，使其能够部署在千台机器上。它借鉴了并行数据库和内存数据库的一些特性，但是却提供了一个完全不同的借口。它实现了四个目标，分别是广泛的适用性，可伸缩性（特指横向），高性能，高可用性。

然后我们又知道了Bigtable的data-model数据模型。Bigtable 是一个稀疏、分布式、持久化存储的多维有序映射表。Bigtable 的数据模型具体来说就是：Bigtable 的每一个键值对的 Key 都为 Row key + Column key + Timestamp 的结构，Value 则是字符串。

然后我们又知道了bigtable的架构。Bigtable 是在 Google 的其他基础设施之上构建的：依赖 WorkQueue 负责故障处理和监控；依赖 GFS 存储日志文件和数据文件；依赖 Chubby 存储元数据和进行主服务器的选择。

Bigtable 集群(cluster)通常在运行其他分布式应用程序的共享机器池(shared pool of machines)中，这是因为 Bigtable 本质是一个进程，Bigtable 进程通常与其他应用程序的进程共享同一台机器。Bigtable 依赖于集群管理系统来调度作业、管理共享机器上的资源、处理机器故障和监视机器状态。

总结一下我在这篇文章中学到的内容，bigtable既有它的优势也有它的劣势。它提供自定义的数据读写接口而非用户已习惯的 SQL 接口，会带来额外的学习成本。Bigtable 本是为结构化数据存储而设计的，却采用了 Schema Less 的设计。该设计的优势在于能够支持 Schema 的灵活变更，不需要预先定义 Schema 信息。但该设计的缺点也非常明显：Key-Value 中携带了充分的自我描述信息，导致数据有大量的膨胀。同时在数据压缩粒度上比较受限。在range分区方面优点：Range 分区能够很好的保证数据在底层存储上与 Row Key 的顺序是一致的，对 Scan

类型查询比较友好。缺点：对用户 Row Key 的设计提出了非常高的要求。容易导致数据不均匀。另外从计算与存储分离角度来看，Tablet Server 中仅仅提供了数据读写服务入口，但并不存储任何数据，数据文件交由底层的 GFS/Colossus 来存储，可以说，这是一个典型的计算与存储分离的架构。该架构优点：分层设计，每一层更专注于自身的业务能力。Tablet 在 Tablet Server 之间迁移时，不需要移动底层的数据文件。Tablet Server 故障不会导致数据丢失问题。可更充分的利用每一层的资源，降低整体的成本。更适合云上服务架构模型。缺点：更多的网络开销。

## google filesystem读后感

Google filesystem,这篇也是老师要求我们阅读的三篇论文之一。在阅读完之后，我主要有以下总结和心得体会。

google filesystem是分布式存储系统的老祖宗，它最开始是google针对其数据访问模式而设计的分布式存储系统。它主要具备以下特点，高度扩展性，单个GFS集群文件系统可扩展至几千个节点，数PB的容量；高可靠性，GFS运行在普通x86服务器上，但GFS通过多副本等机制保证可靠性；满足POSIX语义，应用程序无需任何修改即可使用GFS。GFS系统主要存储文件系统两类数据：文件元数据和文件数据。

GFS的主要设计目标是能够存放大量的数据集，可以自动分片，适用于各种不同的应用，自动容错。其GFS集群包括：一个master和多个chunkserver，并且若干client会与之交互。GFS是宽松的一致性模型，可以理解是弱一致性的，它并不保证一个chunk的所有副本是相同的。GFS并不是强一致性的，如果要转变成强一致性的设计，几乎要重新设计系统。

虽然这篇论文读的比较仓促，但是我还是有所收获，我准备在假期自己继续去查找相关资料学习。