Bigtable论文主要描述了BigTable的数据模型、设计思想和具体实现。

Bigtable兼具了广泛的可使用性、可拓展性、高性能和高可用性。Bigtable在实现策略上与传统数据库有很多相似的地方，但是Bigtable不支持完整的关系型数据模型；相反，Bigtable提供的是一种非常简单的数据模型，以及为客户端提供接口，可以灵活地操控数据层和格式，可以控制数据在物理层的存储方式。对于Bigtable来说，存储的数据都是未解释的字符串，bigtable对数据的行键和列名建立索引。最后，通过调整Bigtable的表结构参数，客户端还可以控制数据是存在内存还是硬盘上。

Bigtable是一个稀疏的、分布式的、持久化存储的多维度排序Map。Map的索引是行关键字、列关键字以及时间戳；Map中的每个value都是一个未经解析的byte数组。

(row:string, column:string,time:int64)->string

数据的下标是行和列的名字，名字可以是任意的字符串。Bigtable将存储的数据都视为字符串，但是Bigtable本身不去解析这些字符串，客户程序通常会在把各种结构化或者半结构化的数据序列化到这些字符串里。

行：

表中的行关键字可以是任意的字符串（目前支持最大64KB的字符串）。对同一个行关键字的读或者写操作都是原子的（不管读或者写这一行里多少个不同列）（这是通过支持单行上的事务处理实现的）。Bigtable目前还不支持通用的跨行事务处理。

Bigtable通过行关键字的字典顺序来组织数据。

用户可以通过选择合适的行关键字，在数据访问时有效利用数据的位置相关性，从而更好的利用这个特性。举例来说，在Webtable里，通过反转URL中主机名的方式，可以把同一个域名下的网页聚集起来组织成连续的行。具体来说，我们可以把maps.google.com/index.html的数据存放在关键字com.google.maps/index.html下。把相同的域中的网页存储在连续的区域可以让基于主机和域名的分析更加有效。相当于是大类放在一起，然后大类里面不同的小类放在一起；而不是小类放在一起，然后小类与小类之间并没有什么大的共同特性。

BigTable使用Google的分布式文件系统(GFS)存储日志文件和数据文件。

BigTable内部存储数据的文件是Google SSTable格式的。

BigTable还依赖一个高可用的、序列化的分布式锁服务组件，叫做Chubby。

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。

Master服务器用来管理Tablet服务器，同时可以对模式进行相关的修改操作。

Tablet服务器都管理一个Tablet的集合，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。客户程序可以直接和Tablet服务器通信进行读写操作，以避免Master服务器负载过大。

列族：列族数据库将数据存储在列族中，而列族里的行则把许多列数据与本行的“行键”关联起来。存放在同一列族下的所有数据通常都属于同一个类型（我们可以把同一个列族下的数据压缩在一起）。感觉一个列族对应的就是一张表。

时间戳：数据库中自动生成的唯一二进制数字，与时间和日期无关的，通常用作给表行加版本戳的机制。每次修改或插入包含column列的行时，就会在column列中插入增量数据库时间戳值。使用某一行中的column列可以很容易地确定该行中的任何值自上次读取以后是否发生了更改。如果对行进行了更改，就会更新该时间戳值。如果没有对行进行更改，则该时间戳值将与以前读取该行时的时间戳值一致。这将会在并发控制中有广泛应用。

原子操作：一个操作要不然就做完，要不然就不做，不会做一半就结束。

Tablet：表中的每个行都可以动态分区。每个分区叫做一个”Tablet”，Tablet是数据分布和负载均衡调整的最小单位。一个BigTable集群存储了很多表，每个表包含了一个Tablet的集合，而每个Tablet包含了某个范围内的行的所有相关数据。

数据块：SSTable是一系列的数据块组成的。SSTable使用块索引（通常存储在SSTable的最后）来定位数据块。在打开SSTable的时候，索引被加载到内存，每次查找都可以通过一次磁盘搜索完成；也可以选择把整个SSTable都放在内存中，这样就不必访问硬盘了。

Chubby:首先我认为Chubby是一个解决分布式系统一致性问题的组件。利用这个组件，可以实现一个分布式系统的一致性。Chubby类似于一个提供锁的服务器，里面有很多小的文件。假如有一个系统想要实现一致性，获取一个一致性的值，则就会所有的机器同时向Chubby申请创建一个文件，最终仅会有一个机器创建文件成功，那么这个创建成功的机器的值则将会被选为这个分布系统最终的值，并将这个确定的值分发给所有系统中的其他机器，从而实现系统的一致性。<https://blog.csdn.net/historyasamirror/article/details/3870168> （更多细致性的内容可以再看看这个网站）

SSTable与memtable、REDO：SSTable中是Tablet的已保存更新数据（存储在GFS中），memtable中存储的是Tablet较新的数据。当要恢复Tablet时，通过SSTable的数据以及其中的REDO Point，来一步步的重建Tablet的最新数据更新表memtable，从而获得Tablet出问题时到底存有哪些数据。

Bigtable是一个分布式的结构化数据存储系统，能够实现适用性广泛、可扩展、高性能和高可用性的目标。Bigtable面对不同产品的迥异需求，Bigtable成功的提供了一个灵活的、高性能的解决方案。

Bigtable中的数据没有Schema，用户自己去定义Schema。