**对BIGTABLE的初步了解与认识**

学号：18301166

姓名：马鸿凯

**一、什么是Bigtable**

Bigtable是一种用于管理结构化数据的分布式储存系统，该系统用来满足“大数据量、高吞吐量、快速响应”等不同应用场景下的存储需求。被设计用来处理海量的数据。其目的在于可靠地拓展到PB级数据和数千台服务器。Bigtable被Google的很多应用项目用来储存数据，这些应用对Bigtable提出的无论是在数据量上（从URL到网页到卫 星图像）还是在响应速度上（从后端的批量处理到实时数据服务）的需求差异都十分巨大。但是Bigtable对这些需求都成功的提供了一个灵活的、高性能的解决方案。

Bigtable在很多方面与数据库相似，其本身实现了很多的数据库策略。与已经实现可拓展性与高性能的并行数据库和内存数据库不同，Bigtable相比于它们提供了不同的接口。Bigtable本身不支持全关系型数据库， Bigtable为客户端提供了一种简单的数据模型，客户端可以动态地控制数据的布局和格式，并且利用底层数据存储的局部性特征。Bigtable将数据统统看成无意义的字节串，客户端需要将结构化和非结构化数据串行化再存入Bigtable。

**二、Bigtable的数据模型**

Bigtable是一个稀疏的、分布式的、持久化的存储的多维排序的Map。本质上说，Bigtable是一个键值（key-value）映射。Bigtable的键有三维，分别是行键（row key）、列键（column key）和时间戳（timestamp），行键和列键都是字节串，时间戳是64位整型；而值是一个字节串。可以用 (row:string, column:string, time:int64)→string 来表示一条键值对记录。

**行键（row key）**

行键可以是任意字节串，通常有10-100字节。行的读写都是原子性的。Bigtable按照行键的字典序存储数据。Bigtable的表会根据行键自动划分为片（tablet），片是负载均衡的单元。最初表都只有一个片，但随着表不断增大，片会自动分裂，片的大小控制在100-200MB。行是表的第一级索引，可以把该行的列、时间和值看成一个整体，简化为一维键值映射，类似于：



**列键（column key）**

列是第二级索引，每行拥有的列是不受限制的，可以随时增加减少。为了方便管理，列被分为多个列族（column family，是访问控制的单元），一个列族里的列一般存储相同类型的数据。一行的列族很少变化，但是列族里的列可以随意添加删除。列键按照family:qualifier格式命名的。将列拿出来，将时间和值看成一个整体，简化为二维键值映射。

**时间戳（timestamp）**

时间戳是第三级索引。Bigtable依据时间戳来区分所保存数据的版本。时间戳代表数据储存进Bigtable的准确时间由Bigtable赋值，也可以通过客户端来赋值。由于时间戳是按降序储存，所以在Bigtable中最先读取到的数据版本是最新的。

在Bigtable中进行查询时，如果只给出行列，那么Bigtable返回的是最新版本数据；如果行、列、时间戳都被提供了，那么得到的是时间小于或等于时间戳的版本的数据。

**持久化（ Persistence）、分布式和稀疏**

和其他持久化存储的方式没有区别，当程序结束或者数据入口被关闭后，保存在Bigtable中的数据会被持久化，Bigtable的数据最终会以文件的形式放到Google File System去。GFS即一种分布式文件管理系统，构建在GFS上的Bigtable可以便于底层文件存储可以在独立机器阵列之间传播，数据以类似的方式在多个参与节点中复制，以便数据在基于独立冗余磁盘阵列的系统中的光盘之间进行条带化。稀疏即指的是Bigtable中，一个表里不同的行键的数据，其列键可能完全不同。

**有序和多维度**

有序是指在Bigtable中，不同于其他的大多数Map，Bigtable储存的键值对严格按照字母表顺序排列。即键‘a’的存储行一定在键‘a’的旁边，并远离键’z’的存储行。由于应用Bigtable的系统通常体积庞大，并且数据是分布式存储的，所以通过这种严格的排序方式来建立空间优势确保当一个需要扫描表格的操作发生时，相类似的内容会彼此聚集。

多维度是指Bigtable这种行列时间戳值得多维数据存储结构。

**三、Bigtable的API介绍**

Bigtable提供的API函数包括建立和删除表以及列族。在Bigtable的API中用户还可以做到修改集群、表和列族的元数据等操作。API还介绍了用户可以拿来进行复杂数据处理的Bigtable特性。例如，Bigtable支持单行的事务处理，用户可以据此对存储在一个行关键字下的数据进行原子性的读写更新操作。Bigtable提供了一个允许用户进行跨行批量写入数据的接口，但不支持通用的跨行事务处理。Bigtable的其他特性还包括允许把数据项做成整数计时器，允许用户在服务器地址空间执行脚本程序（使用Google开发的Sawzall数据处理语言）。

**四、Bigtable的组成构件**

Bigtable是基于Google的几个基础构件上的。

**GFS :**Google的分布式文件系统。Bigtable使用它来存储日志文件和数据文件。Bigtable的集群通常运行在一个共享的机器池中，其中还可以运行其他的分布式应用程序，所以Bigtable的进程经常和其他进程共享机器。Bigtable依赖集群管理器来调度任务、管理共享的机器的资源、处理机器的故障和监视机器的状态。

**SSTable:** Google的一种文件格式，被Bigtable用来内部存储数据。SSTable提供了持久的、排序不变的key-value映射。Key和value都可以是任意字符串，与此同时还有根据一个指定的key查找value的操作，以及遍历指定key的一个范围内所有的key-value对。内部地，每个SSTable包含了一系列的块（一般情况下，每个块的大小为64KB，但是这是可配置的）。一个块索引（存储在SSTable的末尾）用来定位块，这个索引会在SSTable打开时载入到内存中。一个查询可以执行一次独立的磁盘查询；样地，一个SSTable能够完整的被映射到内存中，这样可以让我们在不接触磁盘的情况下来执行查询和浏览。

**Chubby:** Google的一个高可用、序列化的分布式锁服务组件。一个Chubby服务包括五个副本，其中的一个副本会被选为Master副本来处理请求。Bigtable使用Chubby来完成多个任务：确保在任何给定的时间内最多只有一个活动的Master副本；存储Bigtable数据的自引导指令的位置；查找Tablet服务器，以及在Tablet服务器失效时进行善后；存储Bigtable的模式信息（每张表的列族信息）；以及存储访问控制列表。如果Chubby长时间无法访问，Bigtable就会失效。

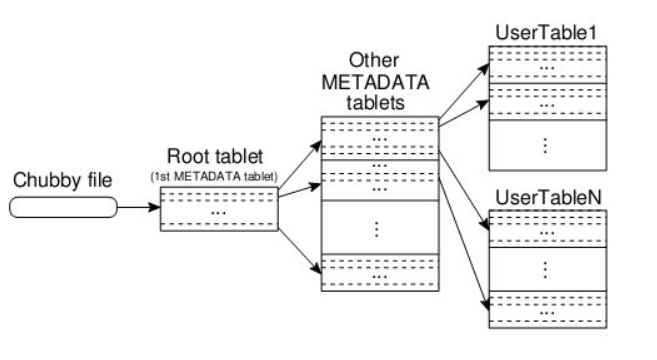
**五、工作原理**

Bigtable有三个主要的组件：一个Master服务器、多个Tablet服务器和链接到客户程序的库。参照工作负载变化情况，Bigtable可以动态的向集群中添加或者删除Table服务器。

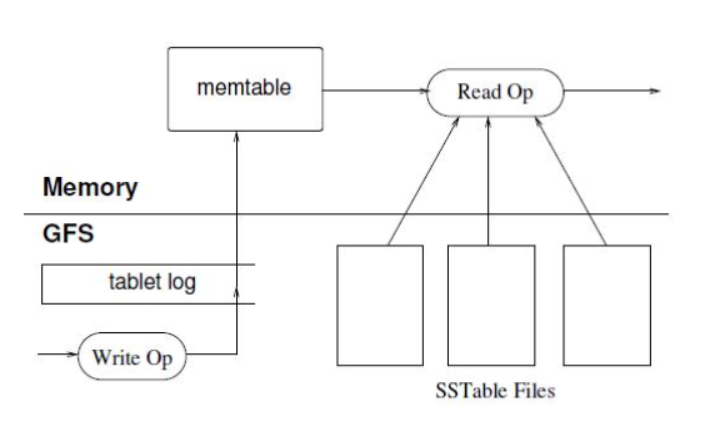
Master服务器：为Table服务器分配Tablets、检测新加入的或者已经过期失效的Tablet服务器、对Tablet服务器进行负载均衡、对保存在GFS的文件进行垃圾处理以及对模式的相关修改。

Tablet服务器：每个Tablet服务器管理一个Tablet集合。每个Tablet服务器负责处理它所加载的Tablet的读写操作，以及在Tablet过大时进行分割。另外Tablet服务器和客户程序直接通信进行Tablet的读写操作，客户程序不必通过Master服务器来获取Tablet的位置信息。

Bigtable集群：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。初始状态下，一个表只有一个Tablet。随着表中数据的增长，它被自动分割成多个Tablet。

Bigtable使用的用来存储位置信息的数据结构存储片类似B+树。在第一层是一个Chubby文件，属于Chubby服务的一部分。当Chubby不可用时，Bigtable将丢失root tablet的位置，整个Bigtable也就不可用；第二层是root table是元数据表的第一个分片，用来保存元数据表的其他分片的位置。为保证树的深度不变，root table不能分裂；第三层是其他的元数据片，和第二层的root tablet组成完整的元数据表。每个元数据片包含许多用户数据片的位置信息。

Tablet服务如图下所示，它的持久化信息保存至GFS里。更新操作提交到REDO日志中



，最近提交的更新操作被存放在一个排序的缓存中；较早的更新操作放在一系列的SSTable中，在恢复Tablet时，Tablet服务器首先从之前提及的那个排序的缓存中读取元数据（其中包含了要被恢复的Tablet的SSTable的列表、一系列指向可能含有该Tablet数据的已提交的日志记录的Redo Point·），Tablet服务器把SSTable的索引读进内存，再重复Redo Point之后提交的更新来重建Tablet。对Tablet服务器进行写操作时，Tablet首先检测操作格式是否正确，再通过Chubby文件来验证操作的权限。操作成功的记录会被记录在提交日志中，允许采用批量提交的方式。在进行读操作时，Tablet服务器也会做类似的完整性和权限检测。在进行Tablet的合并和分割时，正在进行的读写操作能够继续进行。

以上就是对Bigtable的组成部件和主要功能的初步了解。Bigtable在拥有如此多的优势的同时，也有着缺点，比如内存和网络损坏、机器悬挂和时钟偏大等问题。Bigtable在计算机领域算是为人们打开大数据时代的敲门砖。在Bigtable的基础上，更多的更优秀的分布式存储系统得以发展完善，为数据存储与使用提供更加高效方便的安全途径。