Bigtable: A Distributed Storage System for Structured Data读后感

Bigtable是一个分布式存储系统，用于管理结构化数据，这些数据被设计成可以扩展到非常大的规模:跨越数千台商品服务器的pb级数据。Bigtable已经成功地为谷歌的许多项目提供了灵活、高性能的解决方案。本文描述了Bigtable提供的一个简单的数据模型，它使客户端能够动态地控制数据的布局和格式，并描述了Bigtable的设计和实现。

在两年半时间里，名为Bigtable的分布式存储系统被设计出来并实现部署，用于管理谷歌的结构化数据。Bigtable设计用于可靠地伸缩到pb级的数据和数千台机器。Bigtable已经实现了几个目标:广泛的适用性、可伸缩性、高性能和高可用性。客户端可以通过在模式中谨慎选择来控制数据的位置。最后，Bigtable模式参数允许客户端动态控制是从内存中还是从磁盘中提供数据。

在Bigtable的数据模型上使用这个datamodelafterexamina类bigtable系统。每个数据的读或写在一行关键是原子的不同的列数(不管读或写在一行),一个设计决定,使它更容易为客户原因aboutthesystem behaviorinthepresenceofconcurrent更新相同的行。Bigtable按行键按字典顺序维护数据。Therowrangeforatable dynamicallypartitioned。每行范围称为一个平板，它是分配和负载平衡的单位。因此，读取较短的行范围是有效的，并且通常只需要与少量的机器进行通信。客户端可以通过选择其行键来利用此属性，以便为其数据访问获得良好的位置。

Bigtable 提供了用于创建和删除表和列族的函数。它还提供了更改集群、表和列族元数据的功能。客户端应用程序可以在Bigtable中写入或删除值，从各个行中查找值，或者遍历表中的数据子集。客户机可以遍历多个列族，并且有几种机制可以限制扫描产生的行、列和时间戳。Bigtable还支持其他几个特性，允许用户以更复杂的方式操作数据。首先，Bigtable支持单行事务，可用于在单个rowkey下对datastoredon执行原子的读-修改-写序列。Bigtable目前不支持跨行键的一般事务，尽管它提供了在客户端跨行键批处理写操作的接口。其次，Bigtable允许将单元格用作整数计数器。最后，Bigtable支持在服务器的地址空间中执行客户机提供的脚本。脚本是用谷歌开发的一种语言编写的，该语言用于处理称为Sawzall[28]的数据。目前，我们基于sawzall的API不允许客户端脚本写回Bigtable，但是它允许各种形式的数据转换、基于任意表达式的过滤和通过各种操作符的摘要。Bigtable可以与MapReduce[12]一起使用，后者是谷歌开发的用于运行大规模并行计算的框架。我们编写了一组包装器，允许将Bigtable用作MapReduce作业的输入源和输出目标。

Bigtable构建在谷歌基础设施的其他几个部分上。Bigtable使用分布式谷歌文件系统(GFS)来存储日志和数据文件。Bigtable集群通常在运行各种其他分布式应用程序的共享机器池中操作，而Bigtable进程通常与来自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。在内部使用谷歌SSTable文件格式存储Bigtable数据。SSTable提供了一个持久的、有序的、从键到值的不可变映射，其中键和值都是任意字节字符串。提供了查找与OSDI 2006 3键关联的值的操作，并在指定的键范围内遍历所有键/值对。Bigtable依赖于一个名为Chubby的高可用性和持久性分布式锁服务。Chubby服务由五个活动副本组成，其中一个被选为主副本并主动为请求提供服务。当大部分副本都在运行时，服务是活动的，并且可以彼此通信。当客户端的会话到期时，它会丢失所有锁并打开句柄。Chubby客户端还可以注册对Chubby文件和目录的回调，以通知更改或会话过期。

Bigtable的核心数据模型是一个稀疏的多维Map数据结构，以 (行-row, 列-column, 时间戳-timestamp)为索引，在每个被索引的单元中存储对Bigtable不透明的用户数据。

使用GFS存储文件内容底部的平板,因为只读文件内容,任何更新操作实现通过创建一个新版本的线,为了加快反应,减少IO操作,写数据,首先将数据写入内存Memtable结构(类)Rowkey Memtable大小超过某一阈值后,Memtable作为新的文件写入文件系统。为了防止发生故障时内存中的数据丢失，在磁盘上维护一个提交日志，数据的操作记录在插入到Memtable之前被写入日志。此外，由于读取数据时，同一行的不同列或版本可能分布在不同的物理文件和memtable中，因此需要同时扫描这些位置，以交付合并的结果。为了降低这些操作的成本，BigTable定期合并这些文件的内容，并将它们写入新文件(以及实际删除一些逻辑删除的数据)

参考文献：Bigtable: A Distributed Storage System for Structured Data