14 | BigTable的开源实现：HBase

2018-11-29 李智慧

从0开始学大数据 进入课程 >

讲述：李智慧
时长 10:15 大小 4.70M

我们知道，Google 发表 GFS、MapReduce、BigTable 三篇论文，号称“三驾马车”，开启了大数据的时代。那和这“三驾马车”对应的有哪些开源产品呢？我们前面已经讲过了 GFS 对应的 Hadoop 分布式文件系统 HDFS，以及 MapReduce 对应的 Hadoop 分布式计算框架 MapReduce，今天我们就来领略一下BigTable 对应的 NoSQL 系统 HBase，看看它是如何大规模处理海量数据的。

在计算机数据存储领域，一直是关系数据库（RDBMS）的天下，以至于在传统企业的应用领域，许多应用系统设计都是面向数据库设计，也就是**先设计数据库然后设计程序**，从而导致**关系模型绑架对象模型**，并由此引申出旷日持久的业务对象贫血模型与充血模型之争。

业界为了解决关系数据库的不足，提出了诸多方案，比较有名的是对象数据库，但是这些数据库的出现似乎只是进一步证明关系数据库的优越而已。直到人们遇到了关系数据库难以克

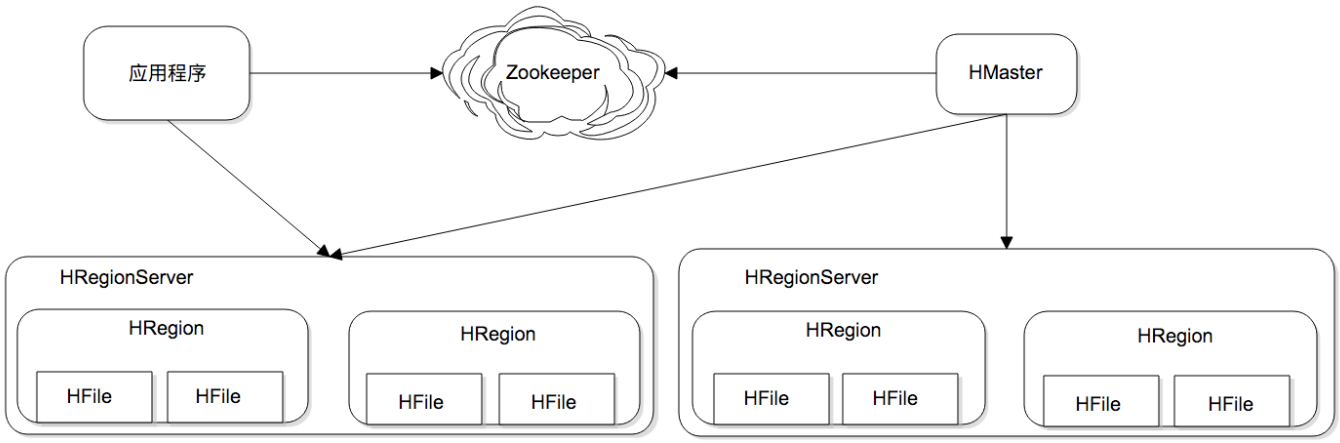
服的缺陷——糟糕的海量数据处理能力及僵硬的设计约束，局面才有所改善。从 Google 的 BigTable 开始，一系列的可以进行海量数据存储与访问的数据库被设计出来，更进一步说，NoSQL 这一概念被提了出来。

NoSQL，主要指非关系的、分布式的、支持海量数据存储的数据库设计模式。也有许多专家将 NoSQL 解读为 Not Only SQL，表示 NoSQL 只是关系数据库的补充，而不是替代方案。其中，HBase 是这一类 NoSQL 系统的杰出代表。

HBase 之所以能够具有海量数据处理能力，其根本在于和传统关系型数据库设计的不同思路。传统关系型数据库对存储在其上的数据有很多约束，学习关系数据库都要学习数据库设计范式，事实上，是在数据存储中包含了一部分业务逻辑。而 NoSQL 数据库则简单暴力地认为，数据库就是存储数据的，业务逻辑应该由应用程序去处理，有时候不得不说，简单暴力也是一种美。

HBase 可伸缩架构

我们先来看看 HBase 的架构设计。HBase 为可伸缩海量数据储存而设计，实现面向在线业务的实时数据访问延迟。HBase 的伸缩性主要依赖其可分裂的 HRegion 及可伸缩的分布式文件系统 HDFS 实现。

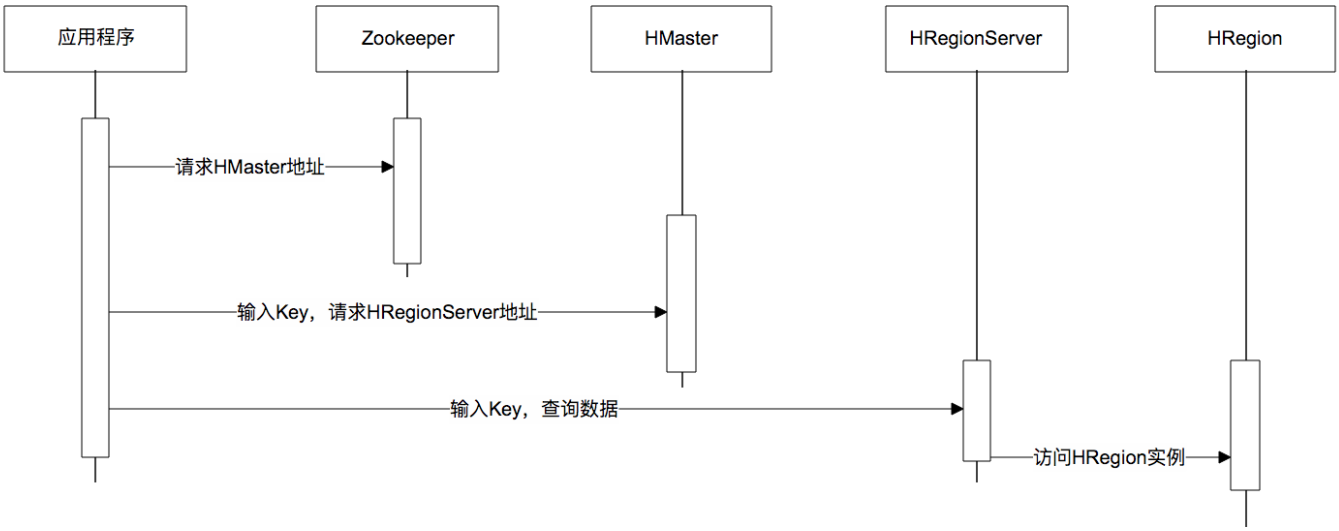


HRegion 是 HBase 负责数据存储的主要进程，应用程序对数据的读写操作都是通过和 HRegion 通信完成。上面是 HBase 架构图，我们可以看到在 HBase 中，数据以 HRegion 为单位进行管理，也就是说应用程序如果想要访问一个数据，必须先找到 HRegion，然后将数据读写操作提交给 HRegion，由 HRegion 完成存储层面的数据操作。

HRegionServer 是物理服务器，每个 HRegionServer 上可以启动多个 HRegion 实例。当一个 HRegion 中写入的数据太多，达到配置的阈值时，一个 HRegion 会分裂成两个 HRegion，并将 HRegion 在整个集群中进行迁移，以使 HRegionServer 的负载均衡。

每个 HRegion 中存储一段 Key 值区间 [key1, key2) 的数据，所有 HRegion 的信息，包括存储的 Key 值区间、所在 HRegionServer 地址、访问端口号等，都记录在 HMaster 服务器上。为了保证 HMaster 的高可用，HBase 会启动多个 HMaster，并通过 ZooKeeper 选举出一个主服务器。

下面是一张调用时序图，应用程序通过 ZooKeeper 获得主 HMaster 的地址，输入 Key 值获得这个 Key 所在的 HRegionServer 地址，然后请求 HRegionServer 上的 HRegion，获得所需要的数据。



数据写入过程也是一样，需要先得到 HRegion 才能继续操作。HRegion 会把数据存储在若干个 HFile 格式的文件中，这些文件使用 HDFS 分布式文件系统存储，在整个集群内分布并高可用。当一个 HRegion 中数据量太多时，这个 HRegion 连同 HFile 会分裂成两个 HRegion，并根据集群中服务器负载进行迁移。如果集群中有新加入的服务器，也就是说有了新的 HRegionServer，由于其负载较低，也会把 HRegion 迁移过去并记录到 HMaster，从而实现 HBase 的线性伸缩。

先小结一下上面的内容，HBase 的核心设计目标是解决海量数据的分布式存储，和 Memcached 这类分布式缓存的路由算法不同，HBase 的做法是按 Key 的区域进行分片，这个分片也就是 HRegion。应用程序通过 HMaster 查找分片，得到 HRegion 所在的服务器 HRegionServer，然后和该服务器通信，就得到了需要访问的数据。

HBase 可扩展数据模型

传统的关系数据库为了保证关系运算（通过 SQL 语句）的正确性，在设计数据库表结构的时候，需要指定表的 schema 也就是字段名称、数据类型等，并要遵循特定的设计范式。这些规范带来了一个问题，就是僵硬的数据结构难以面对需求变更带来的挑战，有些应用系统设计者通过预先设计一些冗余字段来应对，但显然这种设计也很糟糕。

那有没有办法能够做到可扩展的数据结构设计呢？不用修改表结构就可以新增字段呢？当然有的，许多 NoSQL 数据库使用的列族（ColumnFamily）设计就是其中一个解决方案。列族最早在 Google 的 BigTable 中使用，这是一种面向列族的稀疏矩阵存储格式，如下图所示。

Key	联系方式（Column Family）			课程成绩（Column Family）		
001	Weibo：li_zhihui	分机：233		历史：85		地理：77
002		分机：809	QQ：523		英语：78	地理：87
003		分机：523	QQ：908	历史：91	英语：88	

这是一个学生的基本信息表，表中不同学生的联系方式各不相同，选修的课程也不同， 而且将来还会有更多联系方式和课程加入到这张表里，如果按照传统的关系数据库设计，无论提前预设多少冗余字段都会捉襟见肘、疲于应付。

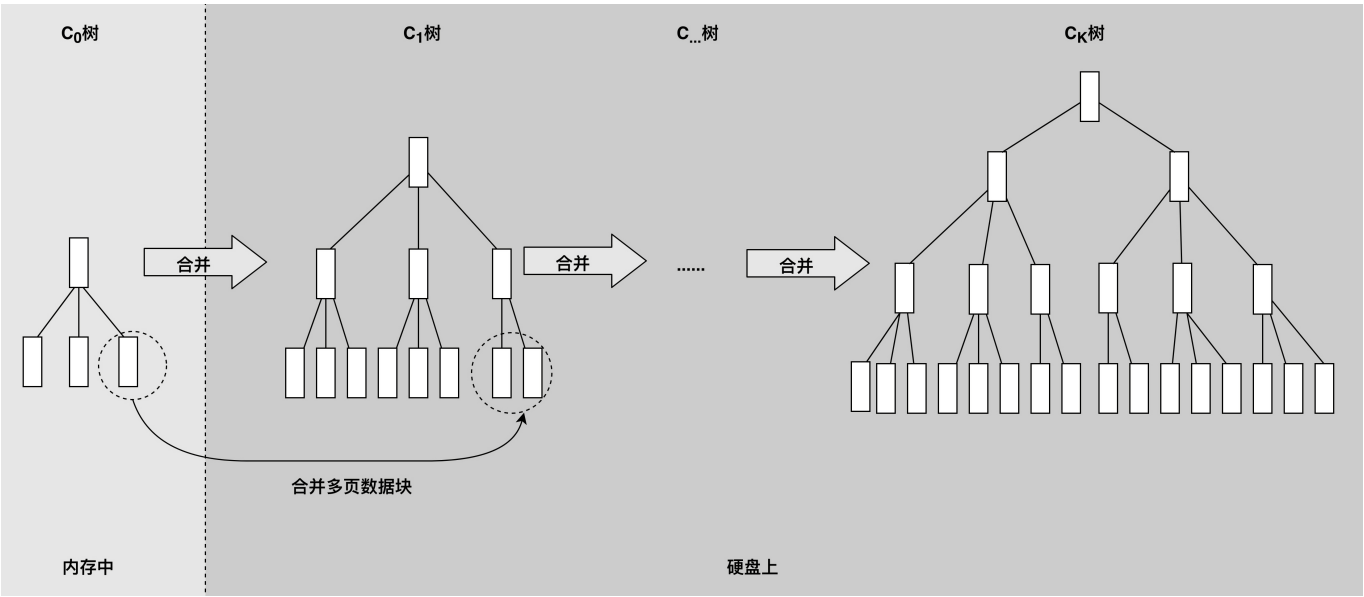
而使用支持列族结构的 NoSQL 数据库，在创建表的时候，只需要指定列族的名字，无需指定字段（Column）。那什么时候指定字段呢？可以在数据写入时再指定。通过这种方式，数据表可以包含数百万的字段，这样就可以随意扩展应用程序的数据结构了。并且这种数据库在查询时也很方便，可以通过指定任意字段名称和值进行查询。

HBase 这种列族的数据结构设计，实际上是把字段的名称和字段的值，以 Key-Value 的方式一起存储在 HBase 中。实际写入的时候，可以随意指定字段名称，即使有几百万个字段也能轻松应对。

HBase 的高性能存储

还记得专栏第 5 期讲 RAID 时我留给你的思考题吗？当时很多同学答得都很棒。传统的机械式磁盘的访问特性是**连续读写很快，随机读写很慢**。这是因为机械磁盘靠电机驱动访问磁盘上的数据，电机要将磁头落到数据所在的磁道上，这个过程需要较长的寻址时间。如果数据不连续存储，磁头就要不停的移动，浪费了大量的时间。

为了提高数据写入速度，HBase 使用了一种叫作**LSM 树**的数据结构进行数据存储。LSM 树的全名是 Log Structed Merge Tree，翻译过来就是 Log 结构合并树。数据写入的时候以 Log 方式连续写入，然后异步对磁盘上的多个 LSM 树进行合并。



LSM 树可以看作是一个 N 阶合并树。数据写操作（包括插入、修改、删除）都在内存中进行，并且都会创建一个新记录（修改会记录新的数据值，而删除会记录一个删除标志）。这些数据在内存中仍然还是一棵排序树，当数据量超过设定的内存阈值后，会将这棵排序树和磁盘上最新的排序树合并。当这棵排序树的数据量也超过设定阈值后，会和磁盘上下一级的排序树合并。合并过程中，会用最新更新的数据覆盖旧的数据（或者记录为不同版本）。

在需要进行读操作时，总是从内存中的排序树开始搜索，如果没有找到，就从磁盘 上的排序树顺序查找。

在 LSM 树上进行一次数据更新不需要磁盘访问，在内存即可完成。当数据访问以写操作为主，而读操作则集中在最近写入的数据上时，使用 LSM 树可以极大程度地减少磁盘的访问次数，加快访问速度。

小结

最后，总结一下我们今天讲的内容。HBase 作为 Google BigTable 的开源实现，完整地继承了 BigTable 的优良设计。架构上通过数据分片的设计配合 HDFS，实现了数据的分布式海量存储；数据结构上通过列族的设计，实现了数据表结构可以在运行期自定义；存储上通过 LSM 树的方式，使数据可以通过连续写磁盘的方式保存数据，极大地提高了数据写入性能。

这些优良的设计结合 Apache 开源社区的高质量开发，使得 HBase 在 NoSQL 众多竞争产品中保持领先优势，逐步成为 NoSQL 领域最具影响力的产品。

思考题

HBase 的列族数据结构虽然有灵活的优势，但是也有缺点。请你思考一下，列族结构的缺点有哪些？如何在应用开发的时候克服这些缺点？哪些场景最好还是使用 MySQL 这类关系数据库呢？

欢迎你写下自己的思考或疑问，与我和其他同学一起讨论。如果你学完今天的内容有所收获的话，也欢迎你点击“请朋友读”，把今天的文章分享给你的朋友。

上一篇　13 | 同样的本质，为何Spark可以更高效?

下一篇　15 | 流式计算的代表：Storm、Flink、Spark Streaming

精选留言 (48)

大马猴
2018-11-29

33

提一个建议，大家尽量发一些跟技术主题相关的评论，这才是对作者劳动成果的尊重，也请作者少放那些吹捧和自说自话的评论，提高阅读体验。

Kaer
2018-11-29

25

1:列族不好查询，没有传统sql那样按照不同字段方便，只能根据rowkey查询，范围查询scan性能低。2:查询也没有mysql一样的走索引优化，因为列不固定 3:列族因为不固定，所以很难做一些业务约束，比如uk等等。4:做不了事务控制

纯齐
2018-11-29

12

文中提到hbase数据的修改在内存中处理，就是说如果机器断电的话数据会丢失，请问hbase有没有措施来保证数据不丢失?

special
2018-12-21

10

看了十多篇文章了，大都是从大数据领域相关技术的特点，原理及应用场景等方面来阐述，讲得很不错，不过有不少内容需要具备一定的大数据实践及理论基础才能很好的吸收，文章没有针对大数据相关框架或工具的实践介绍，比如环境搭建，操作使用等。这也可以理解，这些放在文章中进行介绍也不大合适。
我最近学习了大数据快一年了，对于大数据领域的常用工具，如...
展开

夏一Sunny
2018-11-29

9

对于LSM树的合并和高效，还是不太理解。
展开

伊森
2018-11-29

7

李老师，hbase对olap的分析场景支持不行吧？这也是我正想问的问题，一般都咋么解决，那种可变化的数据的实时统计分析场景的?

Jowin
2018-12-01

5

列族数据组织方式的缺点：
1）在需要读取整条记录的时候，需要访问多个列族组合数据，效率会降低，可以通过字段冗余来解决一些问题。
2）只能提供Key值和全表扫描两种访问方式，很多情况下需要自己建耳机索引。
3）数据是非结构化，或者说是半结构化的，应用在处理数据时要费点心，不像关系数据...
展开

shangyu
2018-11-30

5

内存写操作 如何保证突然掉电的话不丢数据呢
展开

作者回复: 还有一个写操作日志记录数据，所以数据不会丢，但是宕机恢复需要时间，就是根据日志恢复数据，这段时间部分数据更新访问不到。

程序员小灰
2018-11-29

3

外行人强答一下。
我记得parquet文件的大小和列数有很大关系。如果可以随便增加列，文件会变得很大，增加的列可能包含的数据很少。
个人感觉这种可拓展性强的数据库更适合类似于电商的情况多变的业务。

往事随风，...
2018-11-29

3

LSM树的合并过程是咋样的，还有分区是怎么分的，存储到不同节点上
展开

暴风雪
2018-12-02

2

根据老师的说法，LSM数据先是存储在内存中，当到达阈值的时候才会和磁盘合并（个人理解为序列化），当时如何保证断电的时候，内存中的数据会丢失的问题？

展开

足迹
2018-11-29

2

hbase不支持二级索引，只能通过类似es这样的组件来实现。还是就是事务处理。所以一般oltp还是选择关系型数据库。

Zach_
2018-11-29

2

老师、那我代表同学们也问一个宽泛的问题可以吗：
　　请问您觉得，是哪一本书、或者哪一件事、或者哪一句话，对你的人生产生过很大的影响？ 或者说、您觉得您人生的转折点在哪里。

谢谢老师！ 蜗牛给您致敬！

展开

作者回复: 王小波《我的精神家园》，努力做一个有趣的人，努力超越现实的蝇营狗苟。

谢谢，一起努力。

风中有个肉...
2018-12-04

1

传统业务还是使用关系数据库，列结构没啥使用经验，想想聚合计算不方便

展开

杰之7
2018-12-01

1

通过本节学习了代表NoSQL的Hbase,Hbase可通过HDFS分区对海量数据进行分布式存储，具有可伸缩的特性。同时，具有MySQL等关系数据库不具备的可拓展模型。Hbase通过LSM树结构满足高性能储存。最后，我认为关系数据库发展到今天依然是数据库的主流，具备了其他类型的数据库不具有的特征，所以NoSQL数据库是关系数据库的补充，而不是替代。

展开

Q_x H!
2018-11-29

1

我也想知道lsm树是怎么合并的

展开

纯洁的憎恶
2018-11-29

1

列族的思路和LMS树没看懂

展开

Albert
2018-11-29

1

HBase只能通过key操作数据，不像关系数据库那么灵活，所以需要根据应用程序怎样去操作数据来设计表；字段是以key-value形式存储的，key应该会存在大量冗余

展开

胡家鹏
2018-11-29

1

“HRegion 是 HBase 负责数据存储的主要进程，应用程序对数据的读写操作都是通过和 HRetion 通信完成。” 没有看到HRetion

作者回复: 拼写错误，尽快改正，谢谢纠正

Li Shundu...
2018-11-29

1

HBase的一个缺点是非key列的查找没有索引支持，非常慢。

展开