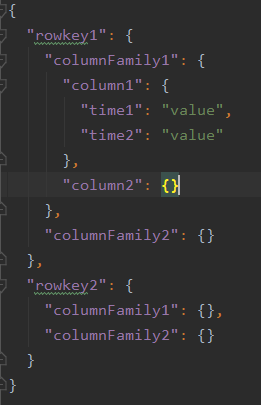
读 google 三驾马车论文有感 18301181逄思默

因为还是把时间拖到了最后才去看了这三篇文章，所以可以说这三篇文章看的都是似懂非懂非懂非懂，所以我想结合着上课所学过的hbase，多讲一讲我对于稍微熟悉点的bigtable的理解和问题。

刚看bigtable这篇文的时候我感觉这不就是学过的hbase么，行索引、列族、列，连分布式存储单元都差不多，一个叫region，一个叫tablet ，最后一查才发现Hbase是Bigtable的一个开源实现，网上也查到一些两者的区别比较，因为区别更看不太懂，我就贴个感觉还很专业的链接，后面就先当他俩是一回事了。

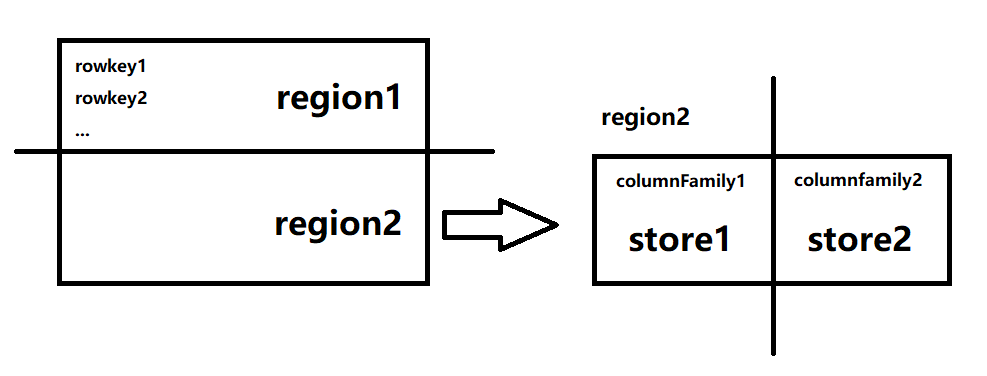
<https://blog.csdn.net/shenxiaoming77/article/details/18267337>

 首先hbase的数据存储是按照多级的索引存的，我觉得把它理解成嵌套的字典是最舒服的方式，比如一个hbase的索引结构可能是这样的：

它的第一索引依然是rowkey，定位到了表中每一个不同的行，之后的索引是columnFamily也就是列族，其实我理解中，到这hbase和一般的关系型数据库存储是很像的，一个行索引和一个列族索引定位到了表中的一个单元格内，只是hbase的每一个单元格内依然可以像redis一样存储k,v的数据结构，而且这个k,v对是可以无限增加的，同一列族的不同行内也可以有完全不同的key，这应该是它特别“非关系型”的部分，这个key就反应成列族中不同的列，value则是值，其实我感觉叫它“列”就不太对头，因为它并没有贯穿一个表。当然列和值之间应该还有一层时间戳的索引，用来记录不同版本，如果不用时间戳索引访问value则默认调用最晚的时间戳索引，返回最新的版本。

明白了hbase基本的表结构，我又从论文和课上回忆了一下它的分布式的数据存储，hbase和bigtable都有着自己的基本数据存储单元，分别叫region和tablet。它们都会把rowkey按照字母顺序排序，然后把整个表横向切开分片，一组行分成一个region，这个region会随着内部行数的变多进行分裂，但每个region的行还是按照字母顺序分的，在外部会有个表存着每个region的行区间，以方便对rowkey进行定位。

了解了一些hbase的存储方式后，我突然发现之前自己理解的一个误区，所谓列族型数据库在文件存取时，它首先还是一个行存储数据库，因为当我想定位到一个列族中的k，v对时，第一件要做的事还是把行拿出来，设想一个部署在分布式系统上的一张很大的表，有很多行，如果我想遍历一个列族中的所有k，v对，那我仍然要去每一个region遍历它的每一行，定位到列族，再遍历其中所有数据，这应该是很难的。

那么列族数据库在文件存取中和行数据库差别非常大的是什么呢，我感受最深的是空间存储性能，也就是数据的压缩，我在阅读资料后得知在hbase中，数据先是按行被分到了不同的region上面，而每一个region中的数据则是按照不同的列族被分成了不同的store，画张示意图是这样的

这样就意味着其实在每一个region的行区域内，一个列族中的k,v对是存在一起的，而一个列族通常被设计成存放数据类型相似的数据，虽然存放到hbase中的数据应该都是bytes[]的格式，但相似类型的数据应该是会体现在重复的字节段的增多来让压缩算法效率升高？（猜）总之类型相似，肯定是要好压的，而反观行数据库的存储方式，每一行中每个单元格都可能是不同的数据结构，这样的行堆在一起，压缩的效率一定就低了许多。

总之，读这三篇的时候的第一感觉就是和hbase那两节课所讲的东西很像，包括hbase的底层文件系统HDFS和Bigtable的底层文件系统GFS，而两者组成的架构都能适用于Mapreduce的计算方法，至于mapreduce我也贴个很好理解的例子了，看了感觉明白不少

<https://www.zhihu.com/question/23345991/answer/223113502>

随着数据量的越来越大，我们对数据的存储，处理，dbms的实现确实都需要一些特殊的办法，平常拿着笔记本跑跑小程序的我可能很难体会到这些精密设计的系统的每一步都有什么样的意义，但这些设计哪怕错了一点都可能是致命的，所以才有了这些看着麻烦的架构，理解他们也确实不能嫌麻烦。