HBase 性能优化

HDFS 设计的初衷是为了存储大文件(例如日志文件)，面向批处理、顺序 I/O 的。然而架设在 HDFS 之上的 HBase 设计的初衷却是为了解决海量数据的随机读写的请求。这种分层的结构设计主要是为了使架构更清晰，HBase 层和 HDFS 层各司其职，是却带来了潜在的性能下降。在很多业务场景中大家使用 HBase 抱怨最多的两个问题就是：Java GC 相关的问题和随机读写性能的问题。

该文章通过几种解决方案和优化策略，可以在一定程度上提高 HBase 的读写性能，接下来，我们进行逐一的分析，并给出相应的解决方案。

首先，对于操作系统而言，为了减少 JVM 虚拟机在 GC 回收时花费的时间，我们需要关闭操作系统的 swap 或是设置 swappiness 为 0，推荐设置为 0，这样只有在物理内存不够的情况下才会使用交换分区。如果虚拟机回收花费的时间过长，会导致 Region server 与 ZK 连接超期，Hmaster 会认为 Region server 已经故障，然后开始分裂 HLog 和重新分配 Region，这就导致 HBase 上的数据进行重分配，从而使一些数据的访问速度变慢。

开启特性 MLAB，MLAB 特性是在分析了 HBase 产生内存碎片的根因后给出了解决方案，这个方案虽然不能够完全解决 Full GC 带来的问题，但是一定程度上延缓了 full GC 的产生间隔。MLAB 在 0.90 版本默认是关闭的在 0.92 版本是默认打开（92 版本最近准备发布了，已经拉出分支来了）。 使用这个特性时，一定要注意如果 keyvalue，如果这个值很大的情况要增加 chunk 值（目前默认 2M）。

实际生产环境中，如果我们对数据结构比较清晰，并且能够确定数据的结构和分布情况，我们可以通过预先创建好一些空分区，达到提高性能的效果，并且避免 Region 的重分配。默认情况下，在创建 HBase 表的时候会自动创建一个 region 分区，当导入数据的时候，所有的 HBase 客户端都向这一个 region 写数据，直到这个 region 足够大了才进行切分，这样 HBase 每次需要自动创建分区，并对分区进行分割，带来一定的系统开销。当然，这种方案是指对数据分布十分明确，而且集群中的物理节点也很确定的情况下。

对于一个实际的 HBase 项目，我们应该设计好表结构以及 Rowkey，这些信息将会对整个表的访问带来很大的性能影响。一个优秀的 HBase 数据表，应该是包含很少的列簇，最好不要超过 3 个。实际上，列簇的设计需要根据你的业务。那些可能被反

复修改的数据表尽量使用单列簇。每个列簇在 HDFS 都有一个独立的 HFILE，当某个 ROWKEY 的某个列簇数据被冲刷时，这个 ROWKEY 连带的其他列簇数据也会被一起冲刷，I/O 负担很大。而持久型数据，也就是一次写入，从不修改的数据，可以使用多列簇，原理相同，但目前仍然提倡单列簇设计模式。

在 HBase 中，row key 可以是任意字符串，最大长度 64KB，实际应用中一般为 10~100bytes，存为 byte[]字节数组，一般设计成定长的。row key 是按照字典序存储，因此，设计 row key 时，要充分利用这个排序特点，将经常一起读取的数据存储到一块，将最近可能会被访问的数据放在一块。例如，如果最近写入 HBase 表中的数据是最可能被访问的，可以考虑将时间戳作为 row key 的一部分，由于是字典序排序，所以可以使用 timestamp 作为 row key，这样能保证新写入的数据在读取时可以被快速命中。

另外，为了提高访问表的性能，我们在创建表的时候，通过 setInMemory 将表放到 RegionServer 的缓存中，保证在读取的时候被 cache 命中，通过 setMaxVersions 设置表中数据的最大版本，如果只需要保存最新版本的数据，那么可以设置为 1，通过 setTimeToLive 设置表中数据的存储生命期，过期数据将自动被删除。

另外，HBase 中还有很多其他的优化参数，如客户端执行写请求时，可以通过设置 autoflush 为 false 来减少频繁的提交事件，通过启用 LZO 压缩算法，减少数据存储和网络传输等，很多优化策略都需要我们结合实际的业务场景进行分析，需要记住，这里没有绝对的最优方案，只有更适合具体业务的优化方案。

项目开发中，我们可以利用一些性能测试工具来跟踪和查找我们程序的性能问题，这里简单介绍两种供大家参考。

第一种是 HBase 自带的性能测试工具 PerformanceEvaluation，这个直接在命令行中执行，如我们需要测试一个随机读的性能，可以输入 hbase org.apache.hadoop.hbase.PerformanceEvaluation randomRead 1 得到结果，详细信息可以参照官方文档说明。

另外一种是 YCSB，它是雅虎开源的一款通用的性能测试工具。与 HBase 自带的性能测试工具相比，它的测试更加灵活，可以选择进行测试的方式：read+write，read+scan 等，还可以选择不同操作的频度与选取 Key 的方式；另外，它可以实时监控测试进度和过程。