

Apache HBase的适用场景

Mar 8th, 2015 8:03 am

原文: <http://blog.cloudera.com/blog/2011/04/hbase-dos-and-donts/>

最近我在[洛杉矶Hadoop用户组](#)做了一次关于[HBase适用场景](#)的分享。在场的听众水平都很高，给了我很多值得深思的反馈。主办方是来自Shopzilla的Jody，我非常感谢他能给我一个在60多位Hadoop使用者面前演讲的机会。可能一些朋友没有机会来洛杉矶参加这次会议，我将分享中的主要内容做了一个整理。如果你没有时间阅读全文，以下是一些摘要：

- HBase很棒，但不是关系型数据库或HDFS的替代者；
- 配置得当才能运行良好；
- 监控，监控，监控，重要的事情要说三遍。

Cloudera是HBase的铁杆粉丝。我们热爱这项技术，热爱这个社区，发现它能适用于非常多的应用场景。HBase如今已经有很多[成功案例](#)，所以很多公司也在考虑如何将其应用到自己的架构中。我做这次分享以及写这篇文章的动因就是希望能列举出HBase的适用场景，并提醒各位哪些场景是不适用的，以及如何做好HBase的部署。

何时使用HBase

虽然HBase是一种绝佳的工具，但我们一定要记住，它并非银弹。HBase并不擅长传统的事务处理程序或关联分析，它也不能完全替代MapReduce过程中使用到的HDFS。从文末的[成功案例](#)中你可以大致了解HBase适用于怎样的应用场景。如果你还有疑问，可以到[社区](#)中提问，我说过这是一个非常棒的社区。

除去上述限制之外，你为何要选择HBase呢？如果你的应用程序中，数据表每一行的结构是有差别的，那就可以考虑使用HBase，比如在标准化建模的过程中使用它；如果你需要经常追加字段，且大部分字段是NULL值的，那可以考虑HBase；如果你的数据（包括元数据、消息、二进制数据等）都有着同一个主键，那就可以使用HBase；如果你需要通过键来访问和修改数据，使用HBase吧。

后台服务

如果你已决定尝试一下HBase，那以下是一些部署过程中的提示。HBase会用到一些后台服务，这些服务非常关键。如果你之前没有了解过ZooKeeper，那现在是个好时候。HBase使用ZooKeeper作为它的分布式协调服务，用于选举Master等。随着HBase的发展，ZooKeeper发挥的作用越来越重要。另外，你需要搭建合适的网络基础设施，如NTP和DNS。HBase要求集群内的所有服务器时间一致，并且能正确地访问其它服务器。正确配置NTP和DNS可以杜绝一些奇怪的问题，如服务器A认为当前是明天，B认为当前是昨天；再如Master要求服务器C开启新的Region，而C不知道自己的机器名，从而无法响应。NTP和DNS服务器可以让你减少很多麻烦。

我前面提到过，在考虑是否使用HBase时，需要针对你自己的应用场景来进行判别。而在真正使用HBase时，监控则成了第一要务。和大多数分布式服务一样，HBase服务器宕机会有多米诺骨牌效应。如果一台服务器因内存不足开始swap数据，它会失去和Master的联系，这时Master会命令其他服务器接过这部分请求，可能会导致第二台服务器也发生宕机。所以，你需要密切监控服务器的CPU、I/O以及网络延迟，确保每台HBase服务器都在良好地工作。监控对于维护HBase集群的健康至关重要。

HBase架构最佳实践

当你找到了适用场景，并搭建起一个健康的HBase集群后，我们来看一些使用过程中的最佳实践。键的前缀要有良好的分布性。如果你使用时间戳或其他类似的递增量作为前缀，那就会让单个Region承载所有请求，而不是分布到各个Region上。此外，你需要根据Memstore和内存的大小来控制Region的数量。RegionServer的JVM堆内存应该控制在12G以内，从而避免过长的GC停顿。举个例子，在一台内存为36G的服务器上部署RegionServer，同时还运行着DataNode，那大约可以提供100个48M大小的Region。这样的配置对HDFS、HBase、以及Linux本身的文件缓存都是有利的。

其他一些设置包括禁用自动合并机制（默认的合并操作会在HBase启动后每隔24小时进行），改为手动的方式在低峰期间执行。你还应该配置数据文件压缩（如LZO），并将正确的配置文件加入HBase的CLASSPATH中。

非适用场景

上文讲述了HBase的适用场景和最佳实践，以下则是一些需要规避的问题。比如，不要期许HBase可以完全替代关系型数据库——虽然它在许多方面都表现优秀。它不支持SQL，也没有优化器，更不能支持跨越多条记录的事务或关联查询。如果你用不到这些特性，那HBase将是你的不二选择。

在复用HBase的服务器时有一些注意事项。如果你需要保证HBase的服务器质量，同时又想在HBase上运行批处理脚本（如使用Pig从HBase中获取数据进行处理），建议还是另搭一套集群。HBase在处理大量顺序I/O操作时（如MapReduce），其CPU和内存资源将会十分紧张。将这两类应用放置在同一集群上会造成不可预估的服务延迟。此外，共享集群时还需要调低任务槽（task slot）的数量，至少要留一半的CPU核数给HBase。密切关注内存，因为一旦发生swap，HBase很可能会停止心跳，从而被集群判为无效，最终产生一系列宕机。

总结

最后要提的一点是，在加载数据到HBase时，应该使用MapReduce+HFileOutputFormat来实现。如果仅使用客户端API，不仅速度慢，也没有充分利用HBase的分布式特性。

用一句话概述，HBase可以让你用键来存储和搜索数据，且无需定义表结构。

使用案例

- Apache HBase: [Powered By HBase Wiki](#)
- Mozilla: [Moving Socorro to HBase](#)
- Facebook: [Facebook's New Real-Time Messaging System: HBase](#)
- StumbleUpon: [HBase at StumbleUpon](#)

Posted by Ji ZHANG Mar 8th, 2015 8:03 am [big data](#), [translation](#)