Mongodb 实战优化

Mongodb是一个高性能，可扩展数据库，并具有低延迟，高吞吐率的性能。但是使用过程中难免会有所坑，下面将介绍一些优化方案。

以下建议翻译自 亚马逊的 《Performance Best Practices for MongoDB 2015》

补充是自己在Mongodb实践中的总结

1、Mongodb引擎

Mongodb 3.0 支持了2种引擎：

A、默认的 MMAPv1 引擎，老版本一种使用

B、WiredTiger 引擎，新支持的db引擎

这两种引擎可以在一个副本集中共存，他们之前的迁移是很容易，也就是说，从 MMAPv1 老的引擎升级到新的 WiredTiger 引擎是不会破坏原有数据的。

这两种引擎是为不同的业务场景做的优化，下面的优化方案会提及支持的相关引擎。

2、硬件优化

2.1、确保足够的内存：

保证内存大于数据存储大小，优化内存是最显而易见的方案

补充：实践证明，尽量保证数据块大小不要超出系统内存的一半，性能最优

2.2、在高写入应用中，使用SSD硬盘

Mongodb的落盘操作是随机的，一般60s落盘，100ms写journal，所以使用SSD硬盘一般会有很显著的写入20-50倍的提升。

补充：当内存在合理范围内，更换SSD硬盘并不会显著提升写性能，关闭journal会提升一些性能，但是可能会带来数据丢失的风险

2.3、压缩存储和I/O密集型负载

新的存储引擎 WiredTiger 允许压缩存储，设置合理的压缩配置，可以防止压缩过大导致的I/O密集型的CPU负载过高的情况。

2.4、使用更快的CPU

这点毋容置疑，新的存储引擎WiredTiger可以比MMAPv1更好的利用多核。

2.5、尽量将Mongodb实例独享物理设备

这儿也不用多说，独享设备肯定是最好

2.6、使用多个mongos

当集群增加后，也要相应的增加mongos

3、应用层优化

3.1、只更新需要更新的字段：

不是更新所有的字段，而只是更新需要更新的改变过的字段，这样可以避免检索文档所有的字段，减少网络传输和数据库负载。

3.2、避免单一 非 条件的查询：

如果单一的非条件查询，会进行全表扫描，尽量避免它

3.3、在你的应用中，使用 explain() 测试你的每一条语句

执行计划explain()可以让你有很多意想不到的收获

3.4、覆盖索引的查询

这个无需多说了，查询的条件尽可能多的覆盖索引

3.5、避免不带片键查询

在集群中，如果不带片键查询，会去每片查询，避免这样的操作

3.6、读写分离

因为主库和从库的同步延迟，如果应用能容忍延迟，最好是从库读，主库写

3.7、使用最新的客户端连接库

mongodb一直在不断的更新，尽量使用官方提供的最新连接库

3.8、分片均匀

如果 mongodb 采用分片，片键的选择非常重要，又要能分片均匀，又要保证读写性能高，还要保证可扩展

3.9、选择合适的片键

同上

补充：

4.0、大量更新批量操作

实际经验，如果一个集合有大量的upate或insert操作，那么与其一个个操作的去连接，不如汇聚到1000-5000条左右，一次性批量写入，可以有效减少锁的竞争情况，性能有显著提高。

4、结构设计和索引

4.1、尽量把数据存成一个文档

在合适的时候把数据存储为一个文档，可以非常高效的利于查询，这样搜索的条数就很少，打开展平对于搜索是不利的，但是过重的一个文档可能也会带来写的问题。

4.2、避免过大文档

Mongodb一个文档最大为16MB，在实际情况中，很多文档都是小于几KB的，比如在文档里维护一个很大的列表，应该将列表打开，一行一条记录。

4.3、避免文档无限的增长

Mongodb为了节约存储空间，会以 usePowerOf2Sizes 来增长一个文档的存储空间，举个例子：

比如：第一次插入这个文档，文档大小为1KB，这时Mongodb会以2KB来存储，在文档增长到2KB以内时，是不会移动文档和索引存储块的。当这个文档增长到 2KB 后，Mongodb又会设置 4KB 来存储它，每次都是以2倍的量存储，保证了空间的利用和存储块移动之间的平衡。如果一个文档不停的无限增长，可能会因为频繁的移动存储块而影响性能。

补充：实际情况中设置usePowerOf2Sizes这个参数，并没有显著的性能提升

4.4、避免大的索引数组

多字段索引可能会利于查询，但是会让写操作变慢，所以要权衡一下。

4.5、避免过长的字段名

Mongodb的字段名和sql不同，会占用文档的空间，增加网络传输压力，尽量用短小精简的字段名，可以在程序那边做映射。

补充：实际情况，设置短小精简的字段名，可以显著提升查询性能

4.6、避免 select * 的查询

这点和sql一样，用不到的字段不要返回了

补充：实际情况也是如此，不查询不需要的字段，显著提升性能

4.7、优化索引

避免索引命中不散列，比如对一个枚举，布尔值做索引，都是无效的

4.8、组合索引可以包含单个索引

比如有一个组合索引 {"lastname":1, "firstname":1}，是没必要建立 {"lastname":1} 这个索引的，因为已经被包括在了组合索引里了

4.9、避免正则表达式没有命中索引

这个类似sql的like，如果要命中索引，必须从头开始匹配

4.10、使用 wiredtiger 引擎，索引放其他地方存储

使用 wiredtiger 引擎，索引可以放在其他地方存储，比如更快的硬盘，资源争抢I/O更少的磁盘

5、磁盘I/O

5.1、readhead需要设置为32

为了mongodb的性能，readhead这个值不能设置少于32，如果设置过多会浪费内存。

5.2、使用EXT4或者XFS文件系统，避免EXT3

EXT3文件系统太过陈旧，没有对数据库进行优化，不够高效

5.3、关闭 access time 设置

关闭对文件最后一次访问的时间记录，因为database一直在访问文件，所以关闭它能够提升性能

5.4、不要使用大量的虚拟内存

将虚拟内存设置为合理大小，不要设置过大

5.5、使用Raid10

使用Raid10来做存储，性能和数据安全都有保证

6、性能配置

6.1、做好性能测试

在应用上线之前，需要做性能测试来评估

6.2、片键

如果没有范围的片键搜索，建议使用hash来使用片键，如果有范围的搜索，建议将他们提前处理，然后并行搜索获取。

6.3、关闭块自动平衡

关闭自动移动块平衡，提升性能

补充：实际情况中，这个设置没有显著提升性能，建议还是打开

6.4、预热系统几分钟

这个看上去没啥用

6.5、对系统瓶颈进行监控

这个也是必须的