

东山絮柳仔

[博客园](#)[首页](#)[新随笔](#)[联系](#)[订阅](#)[管理](#)

MongoDB 提升性能的18原则（开发设计阶段）

MongoDB 是高性能数据，但是在使用的过程中，大家偶尔还会碰到一些性能问题。MongoDB和其它关系型数据库 Server、MySQL、Oracle 相比来说，相对较新，很多人对其不是很熟悉，所以很多开发、DBA往往是注重功能性的要求。其实，MongoDB和 SQL Server、MySQL、Oracle 一样，一个数据库对象的设计调整、索引的创建会对性能产生巨大的影响。

为了充分挖掘MongoDB性能，现简单总计了以下18条，欢迎大家一起来持续总结完善。

- (1) 文档中的_id键推荐使用默认值，禁止向_id中保存自定义的值。

解读：MongoDB文档中都会有一个“_id”键，默认是个ObjectID对象（标识符中包含时间戳、机器ID、进程ID），MongoDB在指定_id与不指定_id插入时，速度相差很大，指定_id会减慢插入的速率。

- (2) 推荐使用短字段名。

解读：与关系型数据库不同，MongoDB集合中的每一个文档都需要存储字段名，长字段名会需要更多的存储空间。

- (3) MongoDB索引可以提高文档的查询、更新、删除、排序操作，所以结合业务需求，适当创建索引。

- (4) 每个索引都会占用一些空间，并且导致插入操作的资源消耗，因此，建议每个集合的索引数尽量控制在5个以内。

(5) 对于包含多个键的查询，创建包含这些键的复合索引是个不错的解决方案。复合索引的键值顺序很重要，原则。

解读：例如在test集合上创建组合索引{a:1,b:1,c:1}。执行以下7个查询语句：

1. db.test.find({a:"hello"})
2. db.test.find({b:"sogo", a:"hello"})
3. db.test.find({a:"hello",b:"sogo", c:"666"})
4. db.test.find({c:"666", a:"hello"})
5. db.test.find({b:"sogo", c:"666"})
6. db.test.find({b:"sogo" })
7. db.test.find({c:"666"})

- 以上查询语句可能走索引的是1、2、3、4
- 查询应包含最左索引字段，以索引创建顺序为准，与查询字段顺序无关。
- 最少索引覆盖最多查询。

(6) TTL 索引 (time-to-live index，具有生命周期的索引)，使用TTL索引可以将超时时间的文档老化，一个文档后就会被删除。

解读：创建TTL的索引必须是日期类型。TTL索引是一种单字段索引，不能是复合索引。TTL删除文档后台线程档。不支持定长集合。

(7) 需要在集合中某字段创建索引，但集合中大量的文档不包含此键值时，建议创建稀疏索引。

解读：索引默认是密集型的，这意味着，即使文档的索引字段缺失，在索引中也存在着一个对应关系。在稀疏索引键值的文档才会出现。

(8) 创建文本索引时字段指定text，而不是1或者-1。每个集合只有一个文本索引，但是它可以为任意多个字段

解读：文本搜索速度快很多，推荐使用文本索引替代对集合文档的多字段的低效查询。

(9) 使用findOne在数据库中查询匹配多个项目，它就会在自然排序文件集合中返回第一个项目。如果需要返回：find方法。

(10) 如果查询无需返回整个文档或只是用来判断键值是否存在，可以通过投影（映射）来限制返回字段，减少网存使用。

解读：既可以通过设置{key:1}来显式指定返回的字段，也可以设置{key:0}指定需要排除的字段。

- (11) 除了前缀样式查询，正则表达式查询不能使用索引，执行的时间比大多数选择器更长，应节制性地使用它们。
- (12) 在聚合运算中，\$match 要在 \$group前面，通过 \$match 前置，可以减少\$group 操作符要处理的文档数量。
- (13) 通过操作符对文档进行修改，通常可以获得更好的性能，因为，不需要往返服务器来获取并修改文档数据，数据上花费更少的时间。
- (14) 批量插入 (batchInsert) 可以减少数据向服务器的提交次数，提高性能。但是批量提交的BSON Size不能超过16MB。
- (15) 禁止一次取出太多的数据进行排序，MongoDB目前支持对32M以内的结果集进行排序。如果需要排序，请分批排序。
- (16) 查询中的某些\$操作符可能会导致性能低下，如\$ne, \$not, \$exists, \$nin, \$or，尽量在业务中不要使用
- a) \$exists:因为松散的文档结构导致查询必须遍历每一个文档；
 - b) \$ne:如果当取反的值为大多数，则会扫描整个索引；
 - c) \$not:可能会导致查询优化器不知道应当使用哪个索引，所以会经常退化为全表扫描；
 - d) \$nin:全表扫描；
 - e) \$or:有多个条件就会查询多少次，最后合并结果集，应该考虑转换为\$in。
- (17) 固定集合可以用于记录日志，其插入数据更快，可以实现插入数据时，淘汰最早的数据。需求分析和设计即提高了性能，有省去了删除动作。
- 解读：固定集合需要显式创建，指定Size的大小，还能够指定文档的数量。集合不管先达到哪一个限制，都会把最老的文档移出。
- (18) 集合中文档的数据量会影响查询性能，为保持适量，需要定期归档。

本文版权归作者所有，未经作者同意不得转载,谢谢配合!!!

标签： MongoDB 设计

好文要顶

关注我

收藏该文

