

MongoDB优化技巧

程显峰

kidrane@gmail.com
@kidrane

- mongo-cn @ googlegroups

技巧I 存储就是存储

- 不要用Map/Reduce, Group
应该用Hadoop, 客户端来完成
- 不要用\$where
要从新设计文档结构
{apple: 3, pear: 4} 查询苹果比梨多1个
- 不要用服务端JavaScript

技巧2 一次完成查询

- 尽量使用内嵌文档，比如博客的评论，订单中的产品详情
- 原则：几乎不改的数据，如生日等个人信息；需要有历史快照的数据，如打折价格
- 反范式化（denormalized）
- no join any more

技巧3 预分配空间

- 虽然文档型数据库有很大的灵活性，但是要尽量少地发挥这种灵活性。比如在线分析系统，或者其他数据流系统（Data Stream Processing System）数据年复一年都一个模子。
- 使用数据模板，整数用0，字符串用够长的垃圾数据，然后\$unset

技巧4 避免skip

- 一般的结果会有sort排序
- 利用辅助的键，比如timestamp来分页

技巧5 慎用数组操作

- \$push, \$pushAll
- 有可能改变文档大小，产生文档拆迁
- 但是真的很灵活

技巧6 使用explain

- 看看到底使用的哪种索引
- 看看扫描了多少文档
- 看看执行时间

提示 I 出问题看日志

- 无法获得锁，一般是由于非正常宕机
- 需要恢复，其实就是导入导出
- 1.8有日志系统，能够快速恢复

提示2 getLastError

- 其实看上一条命令是否成功
- 也可以变成阻塞IO
- 神奇的w参数
- shell中是返回错误的

提示3 典型的复制组

- 1个活跃
- 1个备份
- 1个仲裁（没有数据）

提示4 典型分片

- 3个配置服务器
- 若干复制组作为片（逻辑上的）
- 每个应用客户端一个Mongos

提示5 片键选择

- 不能是随机数
不能确定热数据
- 也不能是按顺序增加的
会在一个分片上狂写
- 组合的, 比如{month, cookie}

提示6 使用puppet

- 使用puppet管理集群
- 减少手工操作
- 做到配置统一
- 自动灾难恢复

提示7 机器配置

- 64位系统 Linux 生产系统不考虑Windows
- 大内存，64G
- SSD?

提示8 合理的类型

- Date
- ObjectId
- 整数 (32, 64)

警告！对齐时间

- 利用NTP对齐时间
- 集群时间差绝对不可以超过15分钟，否则分片会崩溃，这是一个已知的bug
- 注意时区设置

警告2 预警系统

- 监控系统的负载
CPU，磁盘IO，内存，网络负载
- MongoDB的读写效率
重点关注：锁

题外篇

- memcached? 过时了吧 redis? 不过随你需要了
- Hbase不了解 :(

语言支持

- C#
- Java
- Python
- Ruby
- PHP
- 稳定性不是问题，可能效率有差异

mmap

- 成也萧何败萧何
- 性能几乎全仰赖mmap

估计数据文件大小

- 几乎不可能
- 文档大小和文档占用的空间可能不一致
- 用很多空间碎片，会非常非常多

大规模使用案例

- <http://www.mongodb.org/display/DOCS/Production+Deployments>
- 5 Billion +