MongoDB优化及使用限制

导语：优化是个长期且需要通过不断实战才能做出一些总结，故此文档会不定期更新。但是在优化之前，也必须了解一些**MONGODB**内在的限制要求，避免陷入“已知”坑里。

1、使用限制

BSON文档

1.1BSON文档大小

BSON文档的最大为16M。限制文档的大小有助于确保单个文档不会占用太多的内存，在传输时也能节省带宽。如果存储的文档需要超过最大大小的限制，可以使用MongoDB提供的GridFS API。

1.2BSON文档嵌套层级

MongoDB中BSON文档嵌套层级不能超过100层。

1.3命名空间

命名空间长度：每个命名空间，包括数据库名和集合名，加起来要小于123字节。

命名空间个数：Mongodb命名空间存储文件(即*.ns文件，下文统一描述为*.ns文件)所占字节数除以628就是当前系统最大能支持的命名空间数量。
一个16M大小的*.ns文件可以支持约24000左右的命名空间。每个collection和索引都对应一个命名空间。

命名空间存储文件大小（MMAPv1引擎）：
*.ns文件不能超过2047M；
*.ns文件默认是16M。可以在启动时通过设置nssize参数来调整大小。
注：WiredTiger引擎没有此限制

1.4数据库名称限制

数据库名称不区分大小写，故MongoDB不能只依赖大小写字符去区分不同的数据库。

Windows系统开发，数据库名称不能包含以下特殊字符：/\.
“$*<>:|?

Unix and Linux系统开发，数据库名称不能包含：/\. “$

数据库名称不能为空且必须小于64个字符。

另外，以上系统开发过程中，数据库名称不能包含空字符。

1.5集合名限制

集合名可以以下划线开头，或者以字母开头。
但是下面都是不允许的：
包含 “$” 符号
集合名为空字符串 (例如 "")
包含空字符
以system.做为前辍 (因为system是MongoDB内部预留)

集合命名空间最大长度是120个字节（包括数据库名，dot(“.”)，集合名），如<database>.<collection>。

1.6Field名称限制

不能有dot(.)符号。
不能以$符号做为前缀。
更多请移步https://docs.mongodb.com/manual/faq/fundamentals/#faq-dollar-sign-escaping

1.7索引

索引键，被索引的字段值必须小于1024字节，凡是超过大小的值，MongoDB不会记录到索引中。

单个Collection索引数量不能创建超过64个索引。

索引名长度，包括命名空间在内（即数据库名，集合名，dot 分隔符）不能大于128个字符（如<database name>.<collection name>.$<index name>）。如果觉得默认索引名称太长的话，可以通过createIndex()显示指定一个名称。具体可参考https://docs.mongodb.com/manual/reference/method/db.collection.createIndex/#db.collection.createIndex

复合索引中字段的数量不能超过31个Field（字段）。

查询操作不能同时使用text和地理空间索引，text命令需要针对text类型索引进行操作，它不能联合其它不同类型的特殊索引一起使用。使如，$text和$near操作不能同时使用。

1.8数据

Capped Collection的最大文档数，如果你在创建capped collection时使用max参数显式指定了最大文档数，这个数字必须小于2**32（2的32次方）。 如果没有指定，那么就没有文档数的限制。

数据库大小限制，MMAPv1存储引擎不准许超过16000个数据文件。这就意味着单个MMAPv1引擎可以拥有最大32TB文件。如果需要，可以通过修改storage.mmapv1.smallFiles参数，使其最多处理8TB文件。具体请参考https://docs.mongodb.com/manual/reference/configuration-options/#storage.mmapv1.smallFiles

单个mongod实例数据大小不能管理超过底层操作系统所能提供的最大虚拟内存大小的数据集。

虚拟内存限制		
操作系统	Journaled	Not Journaled
Linux	64TB	128TB
Windows Server 2012 R2 and Windows 8.1	64TB	128TB
Windows（其他系列）	4TB	8TB

注：WiredTiger引擎没有此限制。

数据库中集合的数量，最大集合的数量是由数据库的*.ns文件大小以及索引个数计算出来的，具体见上文“命名空间限制”。

1.9复制集

复制集的成员个数，3.0版的复制集最大可以支持50个成员，之前的版本不能超过12个成员。目前可以支持大于12个复制集的客户端驱动如下：
C# (.NET) Driver 1.10
Java Driver 2.13
Python Driver (PyMongo) 3.0+
Ruby Driver 2.0+
Node.JS Driver 2.0+

复制集中可投票成员数，在任意给定的时刻一个复制集中只能有7个可投票的成员。（不可投票成员有否决权，并且可以被选为主节点）

Oplog日志限制，在2.6版本之后，默认上限是50GB。可以通过oplogSizeMB/oplogSize去修改（设置此参数需谨慎，请参考《最优实战》）。

2.0分片集群

分片中的操作限制和不支持的操作：
group操作不能在分片环境中执行，可以用mapReduce或者aggregate框架替代。

db.eval()不支持在被分片的集合中执行。在分片集群中只能通过db.eval()操作未被分片的集合。

$where不允许使用db对象。而未被分片的集合可以使用。（此处文档官方未更新，实际上在Mongodb2.4版本之后，map/reduce操作、group操作、$where操作等已经不能使用某些全局的函数或属性，例如这里所说的db对象）

$isolated 修改器不能在分片环境中使用。

$snapshot不能在分片环境中使用。

分片环境不支持geoSearch命令。

对已经有数据的集合进行分片：
对于已有数据的集合，只要数据少于256G，MongoDB都可以对这个集合开启分片支持。 MongoDB甚至可能对400G大小的集合进行分片，不过这要取决于各个不同大小文档的分布情况。具体的数据需要通过chunk大小和整个数据集的大小计算得出。具体操作请参考https://docs.mongodb.com/manual/tutorial/modify-chunk-size-in-sharded-cluster/

已分片的集合中对单个文档进行修改：

当指定justOne 或者 multi: false选项对文档进行所有的update()和remove()操作时，都必须在查询块中包含shard key或者 _id字段。否则的话，会返回错误。

分片集合中的唯一索引：
MongoDB不支持在分片集合中创建唯一索引，除非这个唯一索引包含了完整的shard key，并做为索引的前辍。通过这种方式MongoDB强制保证唯一性，而不是依靠单个字段。（由于分片后具体的数据库操作经过Mongos最终路由于具体的Monogd进程上，所以要保证全局唯一性必须加上shard key。）

2.1Shard Key限制

Shard Key不能超过512个字节。

Shard Key不能直接被更换：
当对集合进行分片后，不能改换shard key。如果必须要更换的话，按照下面的步骤：
将所有数据从MongoDB中导出到外部存储介质。
Drop掉原始已分片的集合。
重新设置shardkey。
对分片集合进行预拆分操作，以确保数据在初始导入的时候能够均匀分布。
将数据导入到MongoDB。

Shard Key的值是不能被修改：
当对分片集合insert一个文档后，这个文档的shard key或者包含了shard key的字段不能被修改。例如update()操作就不能修改已有文档中的shard key。

递增的ShardKey会影响Insert性能：
对于一个有大量insert操作的集群来说，递增类型的shard key会影响insert的性能。如果shard key是_id，实际上取值也就是ObjectIDs。ObjectIDs 是自增长的，跟时间戳类似。

当shard key是递增字段时，所有的insert操作都会写入到同一个shard的同一个chunk上。系统分根据chunk大小和边界将数据拆分，然后移动到其它节点以保证集群数据分布均匀。但是，在任意时刻，所有的写操作都指向同一个shard节点，这就造成了数据库insert瓶颈。如果集群上的操作主要是读和更新，那这不会造成太大影响。要避免这种情况，可以使用hashed shard key或者选择一个非递增也非递减的字段。

2.2操作

排序操作：
当MongoDB在一次查询中使用了排序，但是并没有通过索引，这个过程返回的数据占用的空间必须少于32M。

聚合Pipeline操作：
PipeLine最多能使用内存是100MB。如果超过此限制，MongoDB将会发生错误。如果要处理大数据集，使用allowDiskUse选项可以准许聚合Pipeline把数据写到临时文件。具体请参考：
排序时内存限制https://docs.mongodb.com/manual/reference/operator/aggregation/sort/#sort-memory-limit

分组时内存限制https://docs.mongodb.com/manual/reference/operator/aggregation/group/#group-memory-limit

多个$in集合达式组合的限制：
查询时使用2个或多个$in集合达式 (比如 { a: { $in: ["a", "b", "c"] }, b: { $in: ["b", "c"] } }) 如果查询使用了这些字段的索引(例如{ a: 1, b: 1 })，可能会触碰到集合达式组合的限制。具体来说，如果组合的数量 (各个不同数组里元素组合个数)大于等于4000000，MongoDB会抛出异常"combinatorial limit of $in partitioning of result setexceeded”。

2D GEO查询时不能使用$or。

2.3官方最新的客户端驱动包支持WiredTiger 引擎的版本如下

语言	版本
C	1.1.0
C++	1.0.0
C#	1.10
Java	2.13
Node.js	1.4.18
Perl	0.708.0.0
PHP	1.5
Python	3.0
Motor	0.4
Ruby	1.12
Scala	3.0

2、参数优化

3、内核优化

4、索引优化

参考资料：
https://docs.mongodb.com/master/administration/analyzing-mongodb-performance/

http://www.slideshare.net/andrew311/optimizing-mongodb-lessons-learned-at-localytics

https://www.youtube.com/watch?v=lnY8D3TL_tM