

1. 你说的 NoSQL 数据库是什么意思?NoSQL 与 RDBMS 直接有什么区别?为什么要使用和不使用 NoSQL 数据库?说一说 NoSQL 数据库的几个优点?

NoSQL 是非关系型数据库，NoSQL = Not Only SQL。

关系型数据库采用的结构化的数据，NoSQL 采用的是键值对的方式存储数据。

在处理非结构化/半结构化的大数据时；在水平方向上进行扩展时；随时应对动态增加的数据项时可以优先考虑使用 NoSQL 数据库。

在考虑数据库的成熟度；支持；分析和商业智能；管理及专业性问题时，应优先考虑关系型数据库。

2. NoSQL 数据库有哪些类型?

NoSQL 数据库的类型

例如：MongoDB, Cassandra, CouchDB, Hypertable, Redis, Riak, Neo4j, HBASE, Couchbase, MemcacheDB, RevenDB and Voldemort are the examples of NoSQL databases.详细阅读。

3. MySQL 与 MongoDB 之间最基本的差别是什么?

MySQL 和 MongoDB 两者都是免费开源的数据库。MySQL 和 MongoDB 有许多基本差别包括数据的表示(data representation)，查询，关系，事务，schema 的设计和定义，标准化(normalization)，速度和性能。

通过比较 MySQL 和 MongoDB，实际上我们是在比较关系型和非关系型数据库，即数据存储结构不同。
[详细阅读](#)

4. 你怎么比较 MongoDB、CouchDB 及 CouchBase?

MongoDB 和 CouchDB 都是面向文档的数据库。MongoDB 和 CouchDB 都是开源 NoSQL 数据库的最典型代表。除了都以文档形式存储外它们没有其他的共同点。MongoDB 和 CouchDB 在数据模型实现、接口、对象存储以及复制方法等方面有很多不同。

细节可以参见下面的链接：

[MongoDB vs CouchDB](#)

[CouchDB vs CouchBase](#)

5. MongoDB 成为最好 NoSQL 数据库的原因是什么?

以下特点使得 MongoDB 成为最好的 NoSQL 数据库：

- 面向文件的
- 高性能
- 高可用性
- 易扩展性
- 丰富的查询语言

6.32 位系统上有什么细微差别?

journaling 会激活额外的内存映射文件。这将进一步抑制 32 位版本上的数据库大小。因此，现在 journaling 在 32 位系统上默认是禁用的。

7. journal 回放在条目(entry)不完整时(比如恰巧有一个中途故障了)会遇到问题吗?

每个 journal (group)的写操作都是一致的，除非它是完整的否则在恢复过程中它不会回放。

8. 分析器在 MongoDB 中的作用是什么?

MongoDB 中包括了一个可以显示数据库中每个操作性能特点的数据库分析器。通过这个分析器你可以找到比预期慢的查询(或写操作);利用这一信息，比如，可以确定是否需要添加索引。

9. 名字空间(namespace)是什么?

MongoDB 存储 BSON 对象在丛集(collection)中。数据库名字和丛集名字以句点连结起来叫做名字空间(namespace)。

10. 如果用户移除对象的属性，该属性是否从存储层中删除?

是的，用户移除属性然后对象会重新保存(re-save())。

11. 能否使用日志特征进行安全备份?

是的。

12. 允许空值 null 吗?

对于对象成员而言，是的。然而用户不能够添加空值(null)到数据库丛集(collection)因为空值不是对象。然而用户能够添加空对象{}。

13. 更新操作立刻 fsync 到磁盘?

不会，磁盘写操作默认是延迟执行的。写操作可能在两秒(默认在 60 秒内)后到达磁盘。例如，如果一秒内数据库收到一千个对一个对象递增的操作，仅刷新磁盘一次。(注意，尽管 fsync 选项在命令行和经过 getLastError_old 是有效的)(译者：也许是坑人的面试题??)。

14. 如何执行事务/加锁?

MongoDB 没有使用传统的锁或者复杂的带回滚的事务，因为它设计的宗旨是轻量，快速以及可预计的高性能。可以把它类比为 MySQL MyISAM 的自动提交模式。通过精简对事务的支持，性能得到了提升，特别是在一个可能会穿过多个服务器的系统里。

15. 为什么我的数据文件如此庞大?

MongoDB 会积极的预分配预留空间来防止文件系统碎片。

16. 启用备份故障恢复需要多久?

从备份数据库声明主数据库宕机到选出一个备份数据库作为新的主数据库将花费 10 到 30 秒时间。这期间在主数据库上的操作将会失败--包括写入和强一致性读取(strong consistent read)操作。然而，你还能在第二数据库上执行最终一致性查询(eventually consistent query)(在 slaveOk 模式下)，即使在这段时间里。

17. 什么是 master 或 primary?

它是当前备份集群(replica set)中负责处理所有写入操作的主要节点/成员。在一个备份集群中，当失效备援(failover)事件发生时，一个另外的成员会变成 primary。

18. 什么是 secondary 或 slave?

Secondary 从当前的 primary 上复制相应的操作。它是通过跟踪复制 oplog(local.oplog.rs)做到的。

19. 我必须调用 getLastError 来确保写操作生效了么?

不用。不管你有没有调用 getLastError(又叫"Safe Mode")服务器做的操作都一样。调用 getLastError 只是为了确认写操作成功提交了。当然，你经常想得到确认，但是写操作的安全性和是否生效不是由这个决定的。

20. 我应该启动一个集群分片(sharded)还是一个非集群分片的 MongoDB 环境?

为开发便捷起见，我们建议以非集群分片(unsharded)方式开始一个 MongoDB 环境，除非一台服务器不足以存放你的初始数据集。从非集群分片升级到集群分片(sharding)是无缝的，所以在你的数据集还不是很大的时候没必要考虑集群分片(sharding)。

21. 分片(sharding)和复制(replication)是怎样工作的?

每一个分片(shard)是一个分区数据的逻辑集合。分片可能由单一服务器或者集群组成，我们推荐为每一个分片(shard)使用集群。

22. 数据在什么时候才会扩展到多个分片(shard)里?

MongoDB 分片是基于区域(range)的。所以一个集合(collection)中的所有的对象都被存放到一个块(chunk)中。只有当存在多余一个块的时候，才会有多个分片获取数据的选项。现在，每个默认块的大小是 64Mb，所以你需要至少 64 Mb 空间才可以实施一个迁移。

23. 当我试图更新一个正在被迁移的块(chunk)上的文档时会发生什么?

更新操作会立即发生在旧的分片(shard)上，然后更改才会在所有权转移(ownership transfers)前复制到新的分片上。

24. 如果在一个分片(shard)停止或者很慢的时候，我发起一个查询会怎样?

如果一个分片(shard)停止了，除非查询设置了“Partial”选项，否则查询会返回一个错误。如果一个分片(shard)响应很慢，MongoDB 则会等待它的响应。

25. 我可以把 moveChunk 目录里的旧文件删除吗?

没问题，这些文件是在分片(shard)进行均衡操作(balancing)的时候产生的临时文件。一旦这些操作已经完成，相关的临时文件也应该被删除掉。但目前清理工作是需要手动的，所以请小心地考虑再释放这些文件的空间。

26. 我怎么查看 Mongo 正在使用的链接?

```
db._adminCommand("connPoolStats");
```

27. 如果块移动操作(moveChunk)失败了，我需要手动清除部分转移的文档吗?

不需要，移动操作是一致(consistent)并且是确定性的(deterministic);一次失败后，移动操作会不断重试;当完成后，数据只会出现在新的分片里(shard)。

28. 如果我在使用复制技术(replication)，可以一部分使用日志(journaling)而其他部分则不使用吗？

可以。

29.当更新一个正在被迁移的块（Chunk）上的文档时会发生什么？

更新操作会立即发生在旧的块（Chunk）上，然后更改才会在所有权转移前复制到新的分片上。

30.MongoDB 在 A:{B,C}上建立索引，查询 A:{B,C}和 A:{C,B}都会使用索引吗？

不会，只会 A:{B,C}上使用索引。

31.如果一个分片（Shard）停止或很慢的时候，发起一个查询会怎样？

如果一个分片停止了，除非查询设置了“Partial”选项，否则查询会返回一个错误。如果一个分片响应很慢，MongoDB 会等待它的响应。

32. MongoDB 支持存储过程吗？如果支持的话，怎么用？

MongoDB 支持存储过程，它是 javascript 写的，保存在 db.system.js 表中。

33.如何理解 MongoDB 中的 GridFS 机制，MongoDB 为何使用 GridFS 来存储文件？

GridFS 是一种将大型文件存储在 MongoDB 中的文件规范。使用 GridFS 可以将大文件分隔成多个小文档存放，这样我们能够有效的保存大文档，而且解决了 BSON 对象有限制的问题。