

# MongoDB 2021最新高频面试题

## MongoDB 2021最新高频面试题

- 1、MongoDB是什么？
- 2、MongoDB有哪些特点？
- 3、MySQL与MongoDB之间最基本的差别是什么？
- 4、monogodb中的分片什么意思？
- 5、MongoDB中的命名空间是什么意思？
- 6、在MongoDb中索引是什么？
- 7、MongoDB成为最好NoSQL数据库的原因是什么？
- 8、解释一下什么是MongoDB中的GridFS？
- 9、分析器在MongoDB中的作用是什么？
- 10、MongoDB更新操作会立刻 fsync 到磁盘？
- 11、MongoDB副本集选举条件有那些？
- 12、简单的描述下MongoDB选举流程
- 13、什么是MongoDB分片集群？
- 14、MongoDB中为何需要水平分片？
- 15、MongoDB中分片键的意义何在？
- 16、什么情况下需要用到MongoDB的分片？
- 17、构建一个分片集群需要用的那些角色？分别是什么？
- 18、如何执行事务/加锁？
- 19、副本集角色有那些？做是什么？
- 20、非关系型数据库有哪些类型？
- 21、数据在什么时候才会扩展到多个分片(shard)里？
- 22、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？
- 23、如何理解MongoDB中的GridFS机制，MongoDB为何使用GridFS来存储文件？
- 24、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？
- 25、MongoDB适合应用在那些场景？
- 26、"ObjectID"有哪些部分组成
- 27、如何使用"AND"或"OR"条件循环查询集合中的文档
- 28、在MongoDB中如何排序？
- 29、什么是聚合？
- 30、raft 选举过程，投票规则？

## 1、MongoDB是什么？

MongoDB是由C++语言编写的，是一个基于分布式文件存储的开源数据库系统。在高负载的情况下，添加更多的节点，可以保证服务器性能。 MongoDB旨在为WEB应用提供可扩展的高性能数据存储解决方案，将数据存储为一个文档，数据结构由键值(key=>value)对组成。MongoDB文档类似于 JSON 对象。字段值可以包含其他文档，数组及文档数组。

## 2、MongoDB有哪些特点？

- (1) MongoDB 是一个面向文档存储的数据库，操作起来比较简单和容易；
- (2) 可以在MongoDB记录中设置任何属性的索引；
- (3) 可以通过本地或者网络创建数据镜像，这使得MongoDB有更强的扩展性；
- (4) 如果负载的增加（需要更多的存储空间和更强的处理能力），它可以分布在计算机网络中的其他节点上这就是所谓的分片；
- (5) 支持丰富的查询表达式，查询指令使用JSON形式的标记，可轻易查询文档中内嵌的对象及数组。

### 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查询、关系、事务、schema的设计和定义、标准化、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。

### 4、mongodb中的分片什么意思？

分片是将数据水平切分到不同的物理节点，当应用数据越来越大的时候，数据量也会越来越大。当数据量增长时，单台机器有可能无法存储数据或可接受的读取写入吞吐量，利用分片技术可以添加更多的机器来应对数据量增加以及读写操作的要求。

### 5、MongoDB中的命名空间是什么意思？

mongodb存储bson对象在从集(collection)中，数据库名字和从集名字以句点连结起来叫做名字空间，一个集合命名空间又有多个数据域(extent)，集合命名空间里存储着集合的元数据，比如集合名称，集合的第一个数据域和最后一个数据域的位置等等。而一个数据域由若干条文档(document)组成，每个数据域都有一个头部，记录着第一条文档和最后一条文档的指针，以及该数据域的一些元数据。extent之间，document之间通过双向链表连接，索引的存储数据结构是B树，索引命名空间存储着对B树的根节点的指针。

### 6、在MongoDb中索引是什么？

索引用于高效的执行查询，没有索引的MongoDB将扫描整个集合中的所有文档，这种扫描效率很低，需要处理大量的数据，索引是一种特殊的数据结构，将一小块数据集合保存为容易遍历的形式。索引能够存储某种特殊字段或字段集的值，并按照索引指定的方式将字段值进行排序。

### 7、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。

### 8、解释一下什么是MongoDB中的GridFS？

为了存储和检索大文件，例如图像，视频文件和音频文件，使用GridFS。默认情况下，它使用两个文件fs.files和fs.chunks来存储文件的元数据和块。

### 9、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。

### 10、MongoDB更新操作会立刻 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)

## 11、MongoDB副本集选举条件有那些？

1. 复制集初始化。
2. 主节点挂掉。
3. 主节点脱离副本集（可能是网络原因）。
4. 参与选举的节点数量必须大于副本集总节点数量的一半，如果已经小于一半了所有节点保持只读状态。

## 12、简单的描述下MongoDB选举流程

- 1、副本集中的主节点选举必须满足“大多数”的原则，所谓“大多数”是指副本中一半以上的成员。副本集中成员只有在得到大多数成员投票支持时，才能成为主节点。例如：有N个副本集成员节点，必须有 $N/2+1$ 个成员投票支持某个节点，此节点才能成为主节点。注意：副本集中若有成员节点处于不可用状态，并不会影响副本集中的“大多数”，“大多数”是以副本集的配置来计算的。
- 2、仲裁节点（Arbiter）它并不保存数据，并且不能被选举为主节点，但是具有投票权。仲裁节点使用最小的资源，不能将Arbiter部署在同一个数据集节点中。
- 3、副本集中最好是有奇数个成员节点，如果有偶数个节点，最好加一个仲裁节点。若副本集中有偶数个成员节点，如图2所示，IDC1网络连不通IDC2，IDC1和IDC2内的成员节点分别会发生选举主节点的行为，然而选举因都无法满足大多数的原则，都不能选出主节点；加入一个仲裁节点之后，则副本集就能满足大多数原则，从中选出主节点了。
4. 如果副本集成员节点数量是奇数，就不再需要仲裁者。但是如果在成员节点是奇数时，强行使用仲裁者，会导致选举耗时变长。由于添加了仲裁者就可能出现两个成员节点票数相同的情况，从而导致选举耗时变长。
5. 若在一轮投票中，副本集中成员节点被投了反对票，则本轮不能被选为主节点。例如，在一个10个成员节点的副本集，某轮投票中，成员节点A由于数据延迟较大被某个成员节点投了反对票，则A同时收到了9票赞成票，然而A仍然不能被选为主节点。
6. 集群中的优先级为0的节点不能成为主节点，并且不能触发选举，但是具有投票权，并且拥有与主节点一致的数据集。

## 13、什么是MongoDB分片集群？

Sharding cluster是一种可以水平扩展的模式，在数据量很大时特给力，实际大规模应用一般会采用这种架构去构建。sharding分片很好的解决了单台服务器磁盘空间、内存、cpu等硬件资源的限制问题，把数据水平拆分出去，降低单节点的访问压力。每个分片都是一个独立的数据库，所有的分片组合起来构成一个逻辑上的完整的数据库。因此，分片机制降低了每个分片的数据操作量及需要存储的数据量，达到多台服务器来应对不断增加的负载和数据的效果。

## 14、MongoDB中为何需要水平分片？

- 1) 减少单机请求数，将单机负载，提高总负载
- 2) 减少单机的存储空间，提高总存储空间

## 15、MongoDB中分片键的意义何在？



- 1、一个好的片键对分片至关重要。片键必须是一个索引,通过 `sh.shardCollection` 加会自动创建索引。一个自增的片键对写入和数据均匀分布就不是很好,因为自增的片键总会在一个分片上写入,后续达到某个阈值可能会写到别的分片。但是按照片键查询会非常高效。随机片键对数据的均匀分布效果很好。注意尽量避免在多个分片上进行查询。
- 2、在所有分片上查询, `mongos` 会对结果进行归并排序,提高查询效率和速度

## 16、什么情况下需要用到MongoDB的分片?

- 1) 机器的磁盘不够用了。使用分片解决磁盘空间的问题。
- 2) 单个 `mongod` 已经不能满足写数据的性能要求。通过分片让写压力分散到各个分片上面,使用分片服务器自身的资源。
- 3) 想把大量数据放到内存里提高性能。和上面一样,通过分片使用分片服务器自身的资源。

## 17、构建一个分片集群需要用的那些角色? 分别是什么?

- 1) 分片服务器 (Shard Server)  
`mongod` 实例,用于存储实际的数据块,实际生产环境中一个 `shard server` 角色可由几台机器组一个 `relica set` 承担,防止主机单点故障  
这是一个独立普通的 `mongod` 进程,保存数据信息。可以是一个副本集也可以是单独的一台服务器。
- 2) 配置服务器 (Config Server)  
`mongod` 实例,存储了整个 `Cluster Metadata`,其中包括 `chunk` 信息。  
这是一个独立的 `mongod` 进程,保存集群和分片的元数据,即各分片包含了哪些数据的信息。最先开始建立,启用日志功能。像启动普通的 `mongod` 一样启动  
配置服务器,指定 `configsvr` 选项。不需要太多的空间和资源,配置服务器的 `1KB` 空间相当于真是数据的 `200MB`。保存的只是数据的分布表。
- 3) 路由服务器 (Route Server)  
`mongos` 实例,前端路由,客户端由此接入,且让整个集群看上去像单一数据库,前端应用起到一个路由的功能,供程序连接。本身不保存数据,在启动时从配置服务器加载集群信息,开启 `mongos` 进程需要知道配置服务器的地址,指定 `configdb` 选项。

## 18、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。

## 19、副本集角色有那些? 做什么?

1. 主节点 (Primary)  
接收所有的写请求,然后把修改同步到所有Secondary。一个Replica Set只能有一个Primary节点,当Primary挂掉后,其他Secondary或者Arbiter节点会重新选举出来一个主节点。  
默认读请求也是发到Primary节点处理的,可以通过修改客户端连接配置以支持读取Secondary节点。
2. 副本节点 (Secondary)  
与主节点保持同样的数据集。当主节点挂掉的时候,参与选主。
3. 仲裁者 (Arbiter)  
不保有数据,不参与选主,只进行选主投票。使用Arbiter可以减轻数据存储的硬件需求,Arbiter几乎没什么大的硬件资源需求,但重要的一点是,在生产环境下它和其他数据节点不要部署在同一台机器上。

## 20、非关系型数据库有哪些类型?

1、Key-Value 存储 Eg:Amazon S3

2、图表 Eg:Neo4J

3、文档存储 Eg:MongoDB

4、基于列存储 Eg:Cassandra

## 21、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)中的所有对象都被存储到一个块(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。

## 22、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。

## 23、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。

## 24、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)。

## 25、MongoDB适合应用在那些场景?

从目前阿里云 MongoDB 云数据库上的用户看，MongoDB 的应用已经渗透到各个领域，比如游戏、物流、电商、内容管理、社交、物联网、视频直播等，以下是几个实际的应用案例。

游戏场景，使用 MongoDB 存储游戏用户信息，用户的装备、积分等直接以内嵌文档的形式存储，方便查询、更新

物流场景，使用 MongoDB 存储订单信息，订单状态在运送过程中会不断更新，以 MongoDB 内嵌数组的形式来存储，一次查询就能将订单所有的变更读取出来。

社交场景，使用 MongoDB 存储存储用户信息，以及用户发表的朋友圈信息，通过地理位置索引实现附近的人、地点等功能

物联网场景，使用 MongoDB 存储所有接入的智能设备信息，以及设备汇报的日志信息，并对这些信息进行多维度的分析

视频直播，使用 MongoDB 存储用户信息、礼物信息等

## 26、"ObjectID"有哪些部分组成

一共有四部分组成：时间戳、客户端ID、客户进程ID、三个字节的增量计数器。

\_id是一个 12 字节长的十六进制数，它保证了每一个文档的唯一性。在插入文档时，需要提供\_id。如果你不提供，那么 MongoDB 就会为每一文档提供一个唯一的 id。\_id的头 4 个字节代表的是当前的时间戳，接着的后 3 个字节表示的是机器 id 号，接着的 2 个字节表示 MongoDB 服务器进程 id，最后的 3 个字节代表递增值。

## 27、如何使用"AND"或"OR"条件循环查询集合中的文档

在 find() 方法中，如果传入多个键，并用逗号(，)分隔它们，那么 MongoDB 会把它看成是AND条件。

```
>db.mycol.find({key1:value1, key2:value2}).pretty()
```

若基于OR条件来查询文档，可以使用关键字\$or。

```
>db.mycol.find(
{
  $or: [
    {key1: value1}, {key2:value2}
  ]
}
).pretty()
```

## 28、在MongoDB中如何排序?

MongoDB 中的文档排序是通过 sort() 方法来实现的。sort() 方法可以通过一些参数来指定要进行排序的字段，并使用 1 和 -1 来指定排序方式，其中 1 表示升序，而 -1 表示降序。

```
>db.connectionName.find({key:value}).sort({columnName:1})
```

## 29、什么是聚合?



聚合操作能够处理数据记录并返回计算结果。聚合操作能将多个文档中的值组合起来，对成组数据执行各种操作，返回单一的结果。它相当于 SQL 中的 `count(*)` 组合 `group by`。对于 MongoDB 中的聚合操作，应该使用 `aggregate()` 方法。

```
>db.COLLECTION_NAME.aggregate(AGGREGATE_OPERATION)
```

## 30、raft 选举过程，投票规则？

选举过程：

当系统启动好之后，初始选举后系统由1个Leader和若干个Follower角色组成。然后突然由于某个异常原因，Leader服务出现了异常，导致Follower角色检测到和Leader的上次RPC更新时间超过给定阈值时间时。此时Follower会认为Leader服务已出现异常，然后它将会发起一次新的Leader选举行为，同时将自身的状态从Follower切换为Candidate身份。随后请求其它Follower投票选择自己。

投票规则：

当一个候选人获得了同一个任期号内的大多数选票，就成为领导人。

每个节点最多在一个任期内投出一张选票。并且按照先来先服务的原则。

一旦候选人赢得选举，立刻成为领导，并发送心跳维持权威，同时阻止新领导人的诞生

**获取更多资源，对话微软MVP，微信扫一扫！**



微信公众号



助教小姐姐



助教小仙女