



持续更新中

回复“1024”获取Java架构师资源



微信搜一搜

Java研发军团

Java研发军团《Java面试手册》V1.0
公众号后台回复“面试手册”

MongoDB面试题

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。

MongoDB 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。

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value
← field: value
← field: value
← field: value

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.你可以在 MongoDB 记录中设置任何属性的索引(如： FirstName="Sameer",Address="8 Gandhi Road")来实现更快的排序。
- 3.你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.Mongo 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。

6. MongoDB 使用 update()命令可以实现替换完成的文档（数据）或者一些指定的数据字段。
7. MongoDB 中的 Map/reduce 主要是用来对数据进行批量处理和聚合操作。
8. Map 和 Reduce。Map 函数调用 emit(key,value)遍历集合中所有的记录，将 key 与 value 传给 Reduce 函数进行处理。
9. Map 函数和 Reduce 函数是使用 Javascript 编写的，并可以通过 db.runCommand 或 mapreduce 命令来执行 MapReduce 操作。
10. GridFS 是 MongoDB 中的一个内置功能，可以用于存放大量小文件。
11. MongoDB 允许在服务端执行脚本，可以用 Javascript 编写某个函数，直接在服务端执行，也可以把函数的定义存储在服务端，下次直接调用即可。

3、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；管理及专业性等问题时，应优先考虑关系型数据库。

4、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。

5、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

6、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：

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

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。

8、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。

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、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)。

11、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())。

12、能否使用日志特征进行安全备份？

是的。

13、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{}

14、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。写操作可能在两三秒(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??)。

15、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。可以把它类比成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。

16、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。

17、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。

18、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。

19、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。

20、我必须调用getError来确保写操作生效了么？

不用。不管你有没有调用getError(又叫"Safe Mode")服务器做的操作都一样。调用getError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。

21、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)。

22、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。

23、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。

24、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。

25、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。

26、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。

27、我怎么查看 Mongo 正在使用的链接?

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

28、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)。

29、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。

30、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。

31、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，只会在A:{B,C}上使用索引。

32、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。

33、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。

34、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。

35、什么是NoSQL数据库？NoSQL和RDBMS有什么区别？在哪些情况下使用和不使用NoSQL数据库？

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；管理及专业性等问题时，应优先考虑关系型数据库。

36、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。

37、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。

38、为什么MongoDB的数据文件很大？

MongoDB采用的预分配空间的方式来防止文件碎片。

39、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。

40、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，只会在A:{B,C}上使用索引。

41、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。

42、分析器在MongoDB中的作用是什么？

分析器就是explain 显示每次操作性能特点的数据库分析器。通过分析器可能查找比预期慢的操作

43、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()）。

44、能否使用日志特征进行安全备份？

是的

45、更新操作立刻fsync到磁盘？

一般磁盘的写操作都是延迟执行的

46、如何执行事务/加锁？

因为mongodb设计就是轻量高性能，所以没有传统的锁和复杂的事务的回滚

47、什么是master或primary？

当前备份集群负责所有的写入操作的主要节点，在集群中，当主节点（master）失效，另一个成员会变为master

48、getLastError的作用

调用getLastError 可以确认当前的写操作是否成功的提交

49、分片（sharding）和复制（replication）是怎样工作的？

分片可能是单一的服务器或者集群组成，推荐使用集群

50、数据在什么时候才会扩展到多个分片（shard）里？

mongodb分片是基于区域的，所以一个集合的所有对象都放置在同一个块中，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

51、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？

会立即更新旧的分片，然后更改会在所有权转移前复制到新的分片上

52、我怎么查看 Mongo 正在使用的链接？

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


53、mongodb的结构介绍

数据库中存储的对象设计bson，一种类似json的二进制文件，由键值对组成

54、数据库的整体结构

键值对-》文档-》集合-》数据库

55、MongoDB是由哪种语言写的

MongoDB用c++编写的,流行的开源数据库MySQL也是用C++开发的。C++1983年发行是一种使用广泛的计算机程序设计语言。它是一种通用[程序设计语言](#)，支持[多重编程模式](#)。

56、MongoDB的优势有哪些

- 面向文档的存储：以JSON 格式的文档保存数据。
- 任何属性都可以建立索引。
- 复制以及高可扩展性。
- 自动分片。
- 丰富的查询功能。
- 快速的即时更新。
- 来自 MongoDB 的专业支持。

57、什么是集合

集合就是一组 MongoDB 文档。它相当于关系型数据库（RDBMS）中的表这种概念。集合位于单独的一个数据库中。一个集合内的多个文档可以有多个不同的字段。一般来说，集合中的文档都有着相同或相关的目的。

58、什么是文档

文档由一组key value组成。文档是动态模式,这意味着同一集合里的文档不需要有相同的字段和结构。在关系型数据库中table中的每一条记录相当于MongoDB中的一个文档。

59、什么是“mongod”

mongod是处理MongoDB系统的主要进程。它处理数据请求，管理数据存储，和执行后台管理操作。当我们运行mongod命令意味着正在启动MongoDB进程,并且在后台运行。

60、"mongod"参数有什么

- 传递数据库存储路径，默认是"/data/db"

- 端口号 默认是 "27017"

61、什么是"mongo"

它是一个命令行工具用于连接一个特定的mongod实例。当我们没有带参数运行mongo命令它将使用默认的端口号和localhost连接

62、MongoDB哪个命令可以切换数据库

MongoDB 用 `use +数据库名称` 的方式来创建数据库。`use` 会创建一个新的数据库，如果该数据库存在，则返回这个数据库。

```
>use database_name
```

63、什么是非关系型数据库

非关系型数据库是对不同于传统关系型数据库的统称。非关系型数据库的显著特点是不使用SQL作为查询语言，数据存储不需要特定的表格模式。由于简单的设计和非常好的性能所以被用于大数据和Web Apps等

64、非关系型数据库有哪些类型

- -Key-Value 存储 Eg:Amazon S3
- 图表 Eg:Neo4j
- 文档存储 Eg:MongoDB
- 基于列存储 Eg:Cassandra

65、为什么用MongoDB ?

- 架构简单
- 没有复杂的连接
- 深度查询能力,MongoDB支持动态查询。
- 容易调试
- 容易扩展
- 不需要转化/映射应用对象到数据库对象
- 使用内部内存作为存储工作区,以便更快的存取数据。

66、在哪些场景使用MongoDB

- 大数据
- 内容管理系统
- 移动端Apps
- 数据管理

67、MongoDB中的命名空间是什么意思？

MongoDB内部有预分配空间的机制，每个预分配的文件都用0进行填充。

数据文件每新分配一次，它的大小都是上一个数据文件大小的2倍，每个数据文件最大2G。

MongoDB每个集合和每个索引都对应一个命名空间，这些命名空间的元数据集中在16M的*.ns文件中，平均每个命名占用约 628 字节，也即整个数据库的命名空间的上限约为24000。

如果每个集合有一个索引（比如默认的_id索引），那么最多可以创建12000个集合。如果索引数更多，则可创建的集合数就更少了。同时，如果集合数太多，一些操作也会变慢。

要建立更多的集合的话，MongoDB也是支持的，只需要在启动时加上"--nssize"参数，这样对应数据库的命名空间文件就可以变得更大以便保存更多的命名。这个命名空间文件（.ns文件）最大可以为 2G。

每个命名空间对应的盘区不一定是连续的。与数据文件增长相同，每个命名空间对应的盘区大小都是随分配次数不断增长的。目的是为了平衡命名空间浪费的空间与保持一个命名空间数据的连续性。

需要注意的一个命名空间\$freelist，这个命名空间用于记录不再使用的盘区（被删除的Collection或索引）。每当命名空间需要分配新盘区时，会先查看\$freelist是否有大小合适的盘区可以使用，如果有就回收空闲的磁盘空间。

68、哪些语言支持MongoDB？

- C
- C++
- C#
- Java
- Node.js
- Perl
- Php 等

69、在MongoDB中如何创建一个新的数据库

MongoDB 用 `use + 数据库名称` 的方式来创建数据库。`use` 会创建一个新的数据库，如果该数据库存在，则返回这个数据库。

```
>use mydb
switched to db mydb
```

70、在MongoDB中如何查看数据库列表

使用命令"show dbs"

```
>show dbs
```

71、MongoDB中的分片是什么意思

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。

72、如何查看使用MongoDB的连接Sharding - MongoDB Manual21.如何查看使用MongoDB的连接

使用命令"db.adminCommand("connPoolStats")"

```
>db.adminCommand("connPoolStats")
```

73、什么是复制

复制是将数据同步到多个服务器的过程，通过多个数据副本存储到多个服务器上增加数据可用性。复制可以保障数据的安全性，灾难恢复，无需停机维护（如备份，重建索引，压缩），分布式读取数据。

74、在MongoDB中如何在集合中插入一个文档

要想将数据插入 MongoDB 集合中，需要使用 insert() 或 save() 方法。

```
>db.collectionName.insert({"key":"value"})
>db.collectionName.save({"key":"value"})
```

75、在MongoDB中如何除去一个数据库Collection Methods24.在MongoDB中如何除去一个数据库

MongoDB 的 dropDatabase() 命令用于删除已有数据库。

```
>db.dropDatabase()
```

76、在MongoDB中如何创建一个集合。

在 MongoDB 中，创建集合采用db.createCollection(name, options)方法。options 是一个用来指定集合配置的文档。

```
>db.createCollection("collectionName")db.createCollection() - MongoDB
Manual>db.createCollection("
```

77、在MongoDB中如何查看一个已经创建的集合

可以使用show collections 查看当前数据库中的所有集合清单

```
>show collections
```

78、在MongoDB中如何删除一个集合

MongoDB 利用 `db.collection.drop()` 来删除数据库中的集合。

```
>db.CollectionName.drop()
```

79、为什么要在MongoDB中使用分析器

数据库分析工具(Database Profiler)会针对正在运行的mongod实例收集数据库命令执行的相关信息。包括增删改查的命令以及配置和管理命令。分析器(profiler)会写入所有收集的数据到 `system.profile` 集合，一个capped集合在管理员数据库。分析器默认是关闭的你能通过per数据库或per实例开启。

80、MongoDB支持主键外键关系吗

默认MongoDB不支持主键和外键关系。用Mongodb本身的API需要硬编码才能实现外键关联，不够直观且难度较大。可以参考以下链接

81、MongoDB支持哪些数据类型

- String
- Integer
- Double
- Boolean
- Object
- Object ID
- Arrays
- Min/Max Keys
- Datetime
- Code
- Regular Expression等

82、为什么要在MongoDB中用"Code"数据类型

"Code"类型用于在文档中存储 JavaScript 代码。

83、为什么要在MongoDB中用"Regular Expression"数据类型

"Regular Expression"类型用于在文档中存储正则表达式

84、为什么在MongoDB中使用"Object ID"数据类型

"ObjectID"数据类型用于存储文档id

85、如何在集合中插入一个文档

要想将数据插入 MongoDB 集合中，需要使用insert()或save()方法。

```
>db.collectionName.insert({"key":"value"})
>db.collectionName.save({"key":"value"})
```

86、"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。

87、在MongoDb中什么是索引

索引用于高效的执行查询。没有索引MongoDB将扫描查询整个集合中的所有文档这种扫描效率很低，需要处理大量数据。索引是一种特殊的数据结构，将一小块数据集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。

88、如何添加索引

使用 db.collection.createIndex() 在集合中创建一个索引

```
>db.collectionName.createIndex({columnName:1})
```

89、用什么方法可以格式化输出结果

使用pretty() 方法可以格式化显示结果

```
>db.collectionName.find().pretty()
```

90、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()
```

91、在MongoDB中如何更新数据

`update()` 与 `save()` 方法都能用于更新集合中的文档。`update()` 方法更新已有文档中的值，而 `save()` 方法则是用传入该方法的文档来替换已有文档。

```
>db.collectionName.update({key:value},{ $set:{newkey:newValue}})
```

92、如何删除文档

MongoDB 利用 `remove()` 方法 清除集合中的文档。它有 2 个可选参数：

- `deletion criteria`：（可选）删除文档的标准。
- `justOne`：（可选）如果设为 `true` 或 `1`，则只删除一个文档。

```
>db.collectionName.remove({key:value})
```

93、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})
```

94、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)
```

95、在MongoDB中什么是副本集

在MongoDB中副本集由一组MongoDB实例组成，包括一个主节点多个次节点，MongoDB客户端的所有数据都写入主节点(Primary),副节点从主节点同步写入数据，以保持所有复制集内存储相同的数据，提高数据可用性。



持续更新中

回复“1024”获取Java架构师资源



微信搜一搜

Java研发军团

Java研发军团《Java面试手册》V1.0
公众号后台回复“面试手册”

Java研发军团
<https://www.ycbbs.vip>