1. **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，泛指非关系型的数据库。随着互联网[web2.0](https://baike.baidu.com/item/web2.0/97695" \t "/home/anwc/文档\\x/_blank)网站的兴起，传统的关系数据库在应付web2.0网站，特别是超大规模和高并发的[SNS](https://baike.baidu.com/item/SNS/10242" \t "/home/anwc/文档\\x/_blank)类型的web2.0纯[动态网](https://baike.baidu.com/item/%E5%8A%A8%E6%80%81%E7%BD%91" \t "/home/anwc/文档\\x/_blank)站已经显得力不从心，暴露了很多难以克服的问题，而非关系型的数据库则由于其本身的特点得到了非常迅速的发展。NoSQL数据库的产生就是为了解决大规模数据集合多重数据种类带来的挑战，尤其是大数据应用难题。

虽然NoSQL的流行与火起来才短短一年的时间，但是不可否认，现在已经开始了第二代运动。尽管早期的堆栈代码只能算是一种实验，然而现在的系统已经更加的成熟、稳定。不过现在也面临着一个严酷的事实：技术越来越成熟——以至于原来很好的NoSQL数据存储不得不进行重写，也有少数人认为这就是所谓的2.0版本。该工具可以为大数据建立快速、可扩展的存储库。

NoSQL与RDBMS之间的区别:

1 理解ACID与BASE的区别（ACID是关系型数据库强一致性的四个要求，而BASE是NoSQL数据库通常对可用性及一致性的弱要求原则，它们的意思分别是，ACID：atomicity, consistency, isolation, durability;BASE：Basically Available, Soft-state, Eventually Consistent。同时有意思的是ACID在英语里意为酸，BASE意思为碱）

2 理解持久化与非持久化的区别。这么说是因为有的NoSQL系统是纯内存存储的。

3 你必须意识到传统有关系型数据库与NoSQL系统在数据结构上的本质区别。传统关系型数据库通常是基于行的表格型存储，而NoSQL系统包括了列式存储（Cassandra）、key/value存储（Memcached）、文档型存储（CouchDB）以及图结构存储（Neo4j）

4与传统关系数据库有统一的SQL语言操作接口不同，NoSQL系统通常有自己特有的API接口。

5 在架构上，你必须搞清楚，NoSQL系统是被设计用于成百上千台机器的集群中的，而非共享型数据库系统的架构。

6在NoSQL系统中，可能你得习惯一下不知道你的数据具体存在何处的情况。

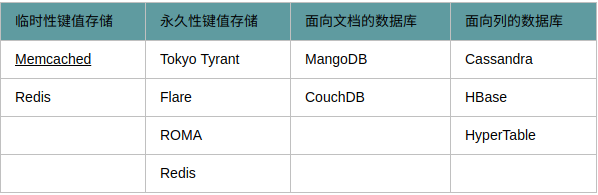
7 在NoSQL系统中，你最好习惯它的弱一致性。”eventually consistent”(最终一致性)正是BASE原则中的重要一项。比如在Twitter，你在Followers列表中经常会感受到数据的延迟。

8 在NoSQL系统中，你要理解，很多时候数据并不总是可用的。

9 你得理解，有的方案是拥有分区容忍性的，有的方案不一定有。

Nosql数据库优势:  
 1.擅长大量数据的写入和读取  
 2.快速的查询响应,灵活的数据模型  
 3.数据结构变更或更新非常方便，不需要更改已有数据的数据结构  
 4.击碎了性能瓶颈，可以使执行速度变的更快

1. **NoSQL数据库都有哪些？**

****

1. **MySQL与MongoDB的区别在哪？**

****

1. **MongoDB成为最好NoSQL数据库的原因是什么？**

面向文档存储(类JSON数据模式简单而强大)

动态查询

全索引支持,扩展到内部对象和内嵌数组

查询记录分析

快速,就地更新

高效存储二进制大对象 (比如照片和视频)

复制和故障切换支持

Auto- Sharding自动分片支持云级扩展性

MapReduce 支持复杂聚合

1. **分析器在MongoDB中的作用是什么？**

[mongodb](http://www.ttlsa.com/mongodb" \o "mongodb" \t "/home/anwc/文档\\x/_blank)索引和查询分析器dex，是一种MongoDB的性能调整工具，比较MongoDB的日志文件和索引条目并给出索引建议。目前，必须提供一个连接数据库的URI。 dex只建议完整的索引，而不是部分索引。不支持[Windows](http://www.ttlsa.com/windows/" \o "windows" \t "/home/anwc/文档\\x/_blank)平台。

dex在运行过程中主要会进行下面三个步骤： 1. 解析query 2. 通过已存在的索引对当前query进行判断 3. 如果发现索引不当，就推荐合适的索引

1. **如果用户移除对象的属性，改属性是否需要从存储层中删除？**

是的，用户移除属性然后对象会重新保存(re-save())。

1. **能否使用日志特征进行安全备份？如何使用？**

可以

1. **允许空值null么？**

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{}。

1. **启用备份故障恢复需要多少时间呢？为什么？在这期间会发生什么？**

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。

1. **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。

1. **什么是secondary或slave?**

Seconday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。

1. **是否必须调用getLastError才能确保写操作生效?**

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。

1. **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。

1. **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。

1. **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。

1. **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。

1. **我怎么查看 Mongo 正在使用的链接?**

db.\_adminCommand("connPoolStats");

1. **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)。

1. **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

1. **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。

1. **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。

1. **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。

1. **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。

1. **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。

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。

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。