

## 期中测试题-一套习题，测出你的掌握程度

你好，我是蒋德钧。

咱们的课程已经更新过半了，我看到很多同学一直在坚持学习，课程每次更新后，都会认真回答课后题，而且还会分享自己的思考和经验。但是，我也发现，最近有不少同学都掉队了，积累了很多节课都没有学习过。

从今天开始，我们就进入期中周了。我知道，很多同学平时确实比较忙，想等到有了大块的时间再来学习。所以，在刚开始设计课程时，我就特意设置了期中周。巧的是，我们的期中周刚好和国庆黄金周重合了。那么，现在，就是你赶上进度的好机会。

在开始做题之前，我想多说几句。

Redis的知识点比较多，而且一旦涉及到性能优化、可靠性保证等需求时，我们就需要和进程、线程、内存管理、磁盘IO、网络连接等计算机底层系统知识打交道。如果你不熟悉底层系统的知识，在学习Redis时，就需要进一步查资料。但是我们平时都很忙，可能会来不及查资料，过一段时间可能就忘了，再想学习时，就需要重新了解，学习成本比较高。

针对这个问题，我想给你分享一下我自己的学习方法。我会用一个word文档或者其他的笔记软件，把涉及到的知识点先记录下来。对于那些我没搞清楚的知识，我会把它们标记为红色，表明这是一个to-do项。等我有空的时候，我就会把这个文档拿出来，挨个儿去查看那些标红的知识点，查找相关的资料，补上知识漏洞。

你可以不要小瞧这个文档，日积月累下来，这就是你的知识宝库。你已经了解的知识点以及还需要进一步学习的知识点，在文档中一目了然。而且，咱们常说“温故而知新”，这个文档就是一个“温故知新”的好材料。

另外，我知道很多同学还有一个疑惑：在学习的时候感觉自己都明白了，但是，真正应用的时候，发现自己又说不清楚或者是想不明白。其实，一个潜在的原因就是，我们对技术点的掌握还不牢固，没有形成自己内在的一套知识体系。

所以，我再给你推荐一个非常有用的学习方法，那就是“**转述**”。每学完一节课之后，就找一个小伙伴，把你学到的内容讲给他/她听。如果对方能听明白，就表示你理解了这些内容。我们自己在讲述内容的时候，潜意识会自动梳理知识点以及它们之间的逻辑关系。当然，你也可以写成一篇文章，如果你发现自己讲不清楚，或者是写不出来，那就代表，你对这些内容的理解有偏差，或者是没有把它们纳入你自己的知识体系。这个时候，你一定要找出来知识盲区，及时在留言区提出来，和我或者是其他小伙伴一起交流讨论。

好了，那话不多说，接下来就准备来自测一下吧。我给你出了一套测试题，包括一套选择题，和一套问答题。

- 选择题：满分共100分，包含15道单选题和5道多选题。提交试卷之后，系统自动评分。
- 问答题：包括3道题目，不计入分数，但我希望你能认真回答这些问题，可以把你的答案写在留言区。在10月7日这一天，我会公布答案。

### 选择题

## 问答题

### 第一题

Redis在接收多个网络客户端发送的请求操作时，如果有一个客户端和Redis的网络连接断开了，Redis会一直等待该客户端恢复连接吗？为什么？

### 第二题

Redis的主从集群可以提升数据可靠性，主节点在和从节点进行数据同步时，会使用两个缓冲区：复制缓冲区和复制积压缓冲区，这两个缓冲区的作用各是什么？会对Redis主从同步产生什么影响吗？

### 第三题

假设在业务场景中，我们有20GB的短视频属性信息（包括短视频ID、短视频基本信息，例如短视频作者、创建时间等）要持久化保存，并且线上负载以读为主，需要能快速查询到这些短视频信息。

现在，我们想使用Redis来实现这个需求，请你来设计一个解决方案。我来提几个问题，你可以思考下。

首先，你会用Redis的什么数据类型来保存数据？如果我们只用单个实例来运行的话，你会采用什么样的持久化方案来保证数据的可靠性？

其次，如果不使用单实例运行，我们有两个备选方案：一个是用两台32GB内存的云主机来运行主从两个Redis实例；另一个是用10台8GB的云主机来运行Redis Cluster，每两台云主机分别运行一个Redis实例主库和从库，分别保存4GB数据，你会用哪种方案呢？请聊一聊你的想法。

好了，这节课就到这里。希望你能抓住期中周的机会，查漏补缺，快速地提升Redis实战能力。我们10月7日见！

## 精选留言：

• Mr.蜜 2020-10-03 00:55:01

- 1.redis不会等待客户端重新连接，做客户端断开处理。如果redis等待客户端连接，会影响其他客户端连接的数据处理，从而影响性能。或者说，redis服务器会等待任何客户端的连接，而不仅仅只等待先前断开的客户端连接，按照epoll模型等待着客户端的连接并做accept和命令处理。
- 2.复制缓冲区是COW（写时复制）时，对RDB备份和主从数据同步的同时，还有写的操作的缓存。复制积压缓冲区是主从数据同步的环形缓冲区，是一个环形窗口机制，这样在增量同步时，主机可以知道需要同步多少数据给从机。
- 3.短视频属性信息，一般以K-V键值对数据，所以使用hashmap更合适（使用string+数据序列化，会使得数据的读取需要在客户端做，整存整取，如果发生多客户端写一个数据时，无法保证数据的安全性），这样可以获取单独的数据，也可以使用hgetall获取单个短视频的全量书信信息。在总量20GB的容量需求情况下，使用Redis Cluster更合适，这样保证单个实例在4G左右，保证单实例的响应速度；也保证了数据的安全性，在主从同步时，也不会因为数据量大，而长时间阻塞主机主线程。

- 卫江 2020-10-02 22:43:35

1 对于redis来说，连接的建立是很普遍的操作，如果等待回复，可能造成不必要的内存使用问题。

2 复制缓存区用于保存全量复制期间的变化，如果全量复制太大，又有大量的修改，可能引发缓存溢出，造成主从复制中断，最严重的后果可能造成死循环，从服务器一直启动不了，且对于主的压力也很大。复制积压缓存区用于全量完成之后如果发生断线重连做的优化。为了控制它的大小，使用了环形队列，但是如果修改太频繁，会很快覆盖头部，在主从发生断线之后，就只能从头开始进行全量同步了

3 对于实用的数据结构，不是很清楚查询的需求，如果只是根据id进行查询的话，是可以使用string，不过string对象的空间利用率不是很高，所以也可以使用hash，控制hash的大小，把所有的数据分片到不同的小hash里面，保证内部使用压缩列表来实现，对于持久化方案，最好是rdb+aof，如果是老版本不支持，可以使用aof，因为是以读为主，修改少，自然产生的aof日志就小，最后是选择分片更多，每个主库数据更少肯定更好，就更不用说加上从库来保证更好的可靠性了，理论上来说，主库的内存占有肯定是越小越好的，这样最起码rdb，主从复制，io的压力更小，阻塞我们主线程的元素更少，同时分片更多，并发度也更好，所以不论从那个方面来说，分片越多，每个分片内存越小，都是好的

- sid 2020-10-02 17:46:19

假期是拉开差距的最好时间！！

- 漫步oo0云端 2020-10-02 13:06:20

1：我认为不会等待恢复连接，断开的连接个人认为会视作处理完成，如果有没处理完的操作，客户端重新请求操作即可。

2：主从复制-复制缓冲区，用于全量复制时临时保存新增数据变更和写入操作。等全量复制完成后，再把复制缓冲区中的数据发送到从库。

主从复制-复制积压缓冲区，是一个环形缓冲区，会不断的写入新增数据，当从库和主库断连，锻炼时间内的新增数据会从复制积压缓冲区同步到从库，当新增数据太多发生溢出时会触发全量同步。

3：我觉得就用String保存就可以因为，因为短视频本身就是bigkey，如果放到集和类型中，会导致一个集和变得超级大。

然后因为主要是读请求，因此数据实时持久化到磁盘也没问题，因为写操作很少。

然后我会用集群方式，因为高并发访问时，bigkey会导致阻塞主进程，因此多台机器分摊并发压力会提升性能。