

1、什么是 Redis？简述它的优缺点？

Redis 本质上是一个 Key-Value 类型的内存数据库，很像 memcached，整个数据库统统加载在内存当中进行操作，定期通过异步操作把数据库数据 flush 到硬盘上进行保存。

因为是纯内存操作，Redis 的性能非常出色，每秒可以处理超过 10 万次读写操作，是已知性能最快的 Key-Value DB。

Redis 的出色之处不仅仅是性能，Redis 最大的魅力是支持保存多种数据结构，此外单个 value 的最大限制是 1GB，不像 memcached 只能保存 1MB 的数据，因此 Redis 可以用来实现很多有用的功能。

比方说用他的 List 来做 FIFO 双向链表，实现一个轻量级的高性能消息队列服务，用他的 Set 可以做高性能的 tag 系统等等。

另外 Redis 也可以对存入的 Key-Value 设置 expire 时间，因此也可以被当作一个功能加强版的 memcached 来用。Redis 的主要缺点是数据库容量受到物理内存的限制，不能用作海量数据的高性能读写，因此 Redis 适合的场景主要局限在较小数据量的高性能操作和运算上。

2、Redis 相比 memcached 有哪些优势？

(1) memcached 所有的值均是简单的字符串，redis 作为其替代者，支持更为丰富的数据类型

(2) redis 的速度比 memcached 快很多

(3) redis 可以持久化其数据

3、Redis 支持哪几种数据类型？

String、List、Set、Sorted Set、hashes

4、Redis 主要消耗什么物理资源？

内存。

5、Redis 的全称是什么？

Remote Dictionary Server。（这里加粗一下）

6、Redis 有哪几种数据淘汰策略？

noeviction: 返回错误当内存限制达到并且客户端尝试执行会让更多内存被使用的命令（大部分的写入指令，但 DEL 和几个例外）

allkeys-lru: 尝试回收最少使用的键（LRU），使得新添加的数据有空间存放。

volatile-lru: 尝试回收最少使用的键（LRU），但仅限于在过期集合的键，使得新添加的数据有空间存放。

allkeys-random: 回收随机的键使得新添加的数据有空间存放。

volatile-random: 回收随机的键使得新添加的数据有空间存放，但仅限于在过期集合的键。

volatile-ttl: 回收在过期集合的键，并且优先回收存活时间（TTL）较短的键，使得新添加的数据有空间存放。

7、Redis 官方为什么不提供 Windows 版本？

因为目前 Linux 版本已经相当稳定，而且用户量很大，无需开发 windows 版本，反而会带来兼容性问题。

8、一个字符串类型的值能存储最大容量是多少？

512M

9、为什么 Redis 需要把所有数据放到内存中？

Redis 为了达到最快的读写速度将数据都读到内存中，并通过异步的方式将数据写入磁盘。

所以 redis 具有快速和数据持久化的特征。如果不将数据放在内存中，磁盘 I/O 速度为严重影响 redis 的性能。

在内存越来越便宜的今天，redis 将会越来越受欢迎。如果设置了最大使用的内存，则数据已有记录数达到内存限值后不能继续插入新值。

10、Redis 集群方案应该怎么做？都有哪些方案？

1.codis。

目前用的最多的集群方案，基本和 twemproxy 一致的效果，但它支持在节点数量改变情况下，旧节点数据可恢复到新 hash 节点。

2.redis cluster3.0 自带的集群，特点在于他的分布式算法不是一致性 hash，而是 hash 槽的概念，以及自身支持节点设置从节点。具体看官方文档介绍。

3.在业务代码层实现，起几个毫无关联的 redis 实例，在代码层，对 key 进行 hash 计算，然后去对应的 redis 实例操作数据。这种方式对 hash 层代码要求比较高，考虑部分包括，节点失效后的替代算法方案，数据震荡后的自动脚本恢复，实例的监控，等等。

11、Redis 集群方案什么情况下会导致整个集群不可用？

有 A, B, C 三个节点的集群,在没有复制模型的情况下,如果节点 B 失败了，那么整个集群就会以为缺少 5501-11000 这个范围的槽而不可用。

12、MySQL 里有 2000w 数据，redis 中只存 20w 的数据，如何保证 redis 中的数据都是热点数据？

redis 内存数据集大小上升到一定大小的时候，就会施行数据淘汰策略。

13、Redis 有哪些适合的场景？

（1）会话缓存（Session Cache）

最常用的一种使用 Redis 的情景是会话缓存（session cache）。用 Redis 缓存会话比其他存储（如 Memcached）的优势在于：Redis 提供持久化。当维护一个不是严格要求一致性的缓存时，如果用户的购物车信息全部丢失，大部分人都会不高兴的，现在，他们还会这样吗？

幸运的是，随着 Redis 这些年的改进，很容易找到怎么恰当的使用 Redis 来缓存会话的文档。甚至广为人知的商业平台 Magento 也提供 Redis 的插件。

（2）全页缓存（FPC）

除基本的会话 token 之外，Redis 还提供很简便的 FPC 平台。回到一致性问题，即使重启了 Redis 实例，因为有磁盘的持久化，用户也不会看到页面加载速度的下降，这是一个极大改进，类似 PHP 本地 FPC。

再次以 **Magento** 为例，**Magento** 提供一个插件来使用 **Redis** 作为全页缓存后端。

此外，对 **WordPress** 的用户来说，**Pantheon** 有一个非常好的插件 **wp-redis**，这个插件能帮助你以最快速度加载你曾浏览过的页面。

（3）队列

Redis 在内存存储引擎领域的一大优点是提供 **list** 和 **set** 操作，这使得 **Redis** 能作为一个很好的消息队列平台来使用。**Redis** 作为队列使用的操作，就类似于本地程序语言（如 **Python**）对 **list** 的 **push/pop** 操作。

如果你快速的在 **Google** 中搜索“**Redis queues**”，你马上就能找到大量的开源项目，这些项目的目的就是利用 **Redis** 创建非常好的后端工具，以满足各种队列需求。例如，**Celery** 有一个后台就是使用 **Redis** 作为 **broker**，你可以从这里去查看。

（4）排行榜/计数器

Redis 在内存中对数字进行递增或递减的操作实现的非常好。集合（**Set**）和有序集合（**Sorted Set**）也使得我们在执行这些操作的时候变的非常简单，**Redis** 只是正好提供了这两种数据结构。

所以，我们要从排序集合中获取到排名最靠前的 10 个用户-我们称之为“**user_scores**”，我们只需要像下面一样执行即可：

当然，这是假定你是根据你用户的分数做递增的排序。如果你想返回用户及用户的分数，你需要这样执行：

```
ZRANGE user_scores 0 10 WITHSCORES
```

Agora Games 就是一个很好的例子，用 **Ruby** 实现的，它的排行榜就是使用 **Redis** 来存储数据的，你可以在这里看到。

（5）发布/订阅

最后（但肯定不是最不重要的）是 **Redis** 的发布/订阅功能。发布/订阅的使用场景确实非常多。我已看见人们在社交网络连接中使用，还可作为基于发布/订阅的脚本触发器，甚至用 **Redis** 的发布/订阅功能来建立聊天系统！

14、Redis 支持的 Java 客户端都有哪些？官方推荐用哪个？

Redisson、**Jedis**、**lettuce** 等等，官方推荐使用 **Redisson**。

15、Redis 和 Redisson 有什么关系？

Redisson 是一个高级的分布式协调 **Redis** 客户端，能帮助用户在分布式环境中轻松实现一些 **Java** 的对象（**Bloom filter**, **BitSet**, **Set**, **SetMultimap**, **ScoredSortedSet**, **SortedSet**, **Map**, **ConcurrentMap**, **List**, **ListMultimap**, **Queue**, **BlockingQueue**, **Deque**, **BlockingDeque**, **Semaphore**, **Lock**, **ReadWriteLock**, **AtomicLong**, **CountDownLatch**, **Publish / Subscribe**, **HyperLogLog**）。

16、Jedis 与 Redisson 对比有什么优缺点？

Jedis 是 **Redis** 的 **Java** 实现的客户端，其 **API** 提供了比较全面的 **Redis** 命令的支持；

Redisson 实现了分布式和可扩展的 **Java** 数据结构，和 **Jedis** 相比，功能较为简单，不支持字符串操作，不支持排序、事务、管道、分区等 **Redis** 特性。**Redisson** 的宗旨是促进使用者对 **Redis** 的关注分离，从而让使用者能够将精力更集中地放在处理业务逻辑上。

17、Redis 如何设置密码及验证密码？

设置密码: `config set requirepass 123456`

授权密码: `auth 123456`

18、说说 Redis 哈希槽的概念？

Redis 集群没有使用一致性 hash,而是引入了哈希槽的概念,Redis 集群有 16384 个哈希槽,每个 key 通过 CRC16 校验后对 16384 取模来决定放置哪个槽,集群的每个节点负责一部分 hash 槽。

19、Redis 集群的主从复制模型是怎样的？

为了使在部分节点失败或者大部分节点无法通信的情况下集群仍然可用,所以集群使用了主从复制模型,每个节点都会有 N-1 个复制品。

20、Redis 集群会有写操作丢失吗？为什么？

Redis 并不能保证数据的强一致性,这意味这在实际中集群在特定的条件下可能会丢失写操作。