

Redis 面试题

1、Redis 集群最大节点个数是多少？

16384 个。

2、Redis 的特点

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。

3、Redis 最适合的场景？

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：ZRRANGE 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！

4、使用 Redis 有哪些好处？

1、速度快，因为数据存在内存中，类似于 HashMap，HashMap 的优势就是

查找和操作的时间复杂度都是 $O(1)$ 。

2、支持丰富数据类型，支持 string，list，set，sorted set，hash。

3、支持事务，操作都是原子性，所谓的原子性就是对数据的更改要么全部执行，要么全部不执行。

4、丰富的特性：可用于缓存，消息，按 key 设置过期时间，过期后将会自动删除。

5、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入内存中，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。

6、Redis 的内存用完了会发生什么？

如果达到设置的上限，Redis 的写命令会返回错误信息（但是读命令还可以正常返回。）或者你可以将 Redis 当缓存来使用配置淘汰机制，当 Redis 达到内存上限时会冲刷掉旧的内容。

7、Redis 的回收策略（淘汰策略）

`volatile-lru`：从已设置过期时间的数据集（`server.db[i].expires`）中挑选最近最少使用的数据淘汰。

`volatile-ttl`：从已设置过期时间的数据集（`server.db[i].expires`）中挑选将要过期的数据淘汰。

`volatile-random`：从已设置过期时间的数据集（`server.db[i].expires`）中任意选择数据淘汰。

`allkeys-lru`：从数据集（`server.db[i].dict`）中挑选最近最少使用的数据淘汰。

`allkeys-random`：从数据集（`server.db[i].dict`）中任意选择数据淘汰。

`no-eviction`（驱逐）：禁止驱逐数据。

注意这里的 6 种机制，`volatile` 和 `allkeys` 规定了是对已设置过期时间的数据集淘汰数据还是从全部数据集淘汰数据，后面的 `lru`、`ttl` 以及 `random` 是三种不同的淘汰策略，再加上一种 `no-eviction` 永不回收的策略。

使用策略规则：

- 1、如果数据呈现幂律分布，也就是一部分数据访问频率高，一部分数据访问频率低，则使用 `allkeys-lru`。
- 2、如果数据呈现平等分布，也就是所有的数据访问频率都相同，则使用 `allkeys-random`。

8、假如 Redis 里面有 1 亿个 key，其中有 10w 个 key 是以某个固定的已知的前缀开头的，如果将它们全部找出来？

使用 keys 指令可以扫出指定模式的 key 列表。

对方接着追问：如果这个 Redis 正在给线上的业务提供服务，那使用 keys 指令会有什么问题？

这个时候你要回答 Redis 关键的一个特性：Redis 的单线程的。keys 指令会导致线程阻塞一段时间，线上服务会停顿，直到指令执行完毕，服务才能恢复。这个时候可以使用 scan 指令，scan 指令可以无阻塞的提取出指定模式的 key 列表，但是会有一定的重复概率，在客户端做一次去重就可以了，但是整体所花费的时间会比直接用 keys 指令长。

9、Memcached 与 Redis 的区别？

1、Redis 不仅仅支持简单的 k/v 类型的数据，同时还提供 list，set，zset，hash 等数据结构的存储。而 memcache 只支持简单数据类型，需要客户端自己处理复杂对象。

2、Redis 支持数据的持久化，可以将内存中的数据保持在磁盘中，重启的时候可以再次加载进行使用（PS：持久化在 rdb、aof）。

10、Redis 常见性能问题和解决方案：

- 1、Master 最好不要写内存快照，如果 Master 写内存快照，save 命令调度 rdbSave 函数，会阻塞主线程的工作，当快照比较大时对性能影响是非常大的，会间断性暂停服务。
- 2、如果数据比较重要，某个 Slave 开启 AOF 备份数据，策略设置为每秒同步一。
- 3、为了主从复制的速度和连接的稳定性，Master 和 Slave 最好在同一个局域网。
- 4、尽量避免在压力很大的主库上增加从。
- 5、主从复制不要用图状结构，用单向链表结构更为稳定，即：Master <- Slave1<- Slave2 <- Slave3... 这样的结构方便解决单点故障问题，实现 Slave 对 Master 的替换。如果 Master 挂了，可以立刻启用 Slave1 做 Master，其他不变。