

- Redis 篇

#### 10.1.0 使用 Redis 有哪些好处？

##### 参考答案：

(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

#### 10.1.1 redis 相比 memcached 有哪些优势？

##### 参考答案：

(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

#### 10.1.2 redis 常见性能问题和解决方案

## 参考答案：

- (1) Master 最好不要做任何持久化工作，如 RDB 内存快照和 AOF 日志文件
- (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次
- (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内
- (4) 尽量避免在压力很大的主库上增加从库
- (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。

10.1.3 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

## 参考答案：

相关知识：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。redis 提供 6 种数据淘汰策略：

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

10.1.4 Memcache 与 Redis 的区别都有哪些?

**参考答案:**

1)、存储方式

Memecache 把数据全部存在内存之中, 断电后会挂掉, 数据不能超过内存大小。

Redis 有部份存在硬盘上, 这样能保证数据的持久性。

2)、数据支持类型

Memcache 对数据类型支持相对简单。

Redis 有复杂的数据类型。

3)、使用底层模型不同

它们之间底层实现方式 以及与客户端之间通信的应用协议不一样。

Redis 直接自己构建了 VM 机制, 因为一般的系统调用系统函数的话, 会浪费一定的时间去移动和请求。

4), value 大小

redis 最大可以达到 1GB，而 memcache 只有 1MB

10.1.5 Redis 常见的性能问题都有哪些？如何解决？

**参考答案：**

1. 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，所以 Master 最好不要写内存快照。
2. Master AOF 持久化，如果不重写 AOF 文件，这个持久化方式对性能的影响是最小的，但是 AOF 文件会不断增大，AOF 文件过大会影响 Master 重启的恢复速度。Master 最好不要做任何持久化工作，包括内存快照和 AOF 日志文件，特别是不要启用内存快照做持久化，如果数据比较关键，某个 Slave 开启 AOF 备份数据，策略为每秒同步一次。
3. Master 调用 BGREWRITEAOF 重写 AOF 文件，AOF 在重写的时候会占大量的 CPU 和内存资源，导致服务 load 过高，出现短暂服务暂停现象。
4. Redis 主从复制的性能问题，为了主从复制的速度和连接的稳定性，Slave 和 Master 最好在同一个局域网内

10.1.6 redis 最适合的场景

10.1.7 Redis 的同步机制了解么？

### 参考答案：

从从同步。第一次同步时，主节点做一次 **bgsave**，并同时 will 后续修改操作记录到内存 **buffer**，待完成后将 **rdb** 文件全量同步到复制节点，复制节点接受完成后将 **rdb** 镜像加载到内存。加载完成后，再通知主节点将期间修改的操作记录同步到复制节点进行重放就完成了同步过程。

10.1.8 是否使用过 Redis 集群，集群的原理是什么？

### 参考答案：

Redis Sentinel 着眼于高可用，在 master 宕机时会自动将 slave 提升为 master，继续提供服务。

Redis Cluster 着眼于扩展性，在单个 redis 内存不足时，使用 Cluster 进行分片存储。

10.1.9 redis 集群如何保证一致性？