

后端开发常问面试题集锦

——算法问题



欢迎做 Java 的工程师朋友们加入 Java 高级架构进阶群；群内有技术大咖指点难题，还提供免费的 Java 架构学习资料（里面有高可用,高并发,高性能及分布式,Jvm 性能调优,Spring 源码,MyBatis,Netty,Redis,Kafka,Mysql,Zookeeper,Tomcat,Docker,Dubbo,Nginx 等多个知识点的架构资料）

群名称:Java高级架构进阶@群
群 号:963944895

Redis 中 5 种数据结构的使用场景介绍

一、redis 数据结构使用场景

原来看过 redisbook 这本书，对 redis 的基本功能都已经熟悉了，从上周开始看 redis 的源码。目前目标是吃透 redis 的数据结构。我们都知道，在 redis 中一共有 5 种数据结构，那每种数据结构的使用场景都是什么呢？

String——字符串

Hash——字典

List——列表

Set——集合

Sorted Set——有序集合

下面我们就来简单说明一下它们各自的使用场景：

1. String——字符串

String 数据结构是简单的 key-value 类型，value 不仅可以是 String，也可以是数字（当数字类型用 Long 可以表示的时候 encoding 就是整型，其他都存储在 sdshdr 当做字符串）。使用 Strings 类型，可以完全实现目前 Memcached 的功能，并且效率更高。还可以享受 Redis 的定时持久化（可以选择 RDB 模式或者 AOF 模式），操作日志及 Replication 等功能。除了提供与 Memcached 一样的 get、set、incr、decr 等操作外，Redis 还提供了下面一些操作：

1. LEN niushuai: 0(1) 获取字符串长度
2. APPEND niushuai redis: 往字符串 append 内容，而且采用智能分配内存（每次 2 倍）
3. 设置和获取字符串的某一段内容
4. 设置及获取字符串的某一位（bit）
5. 批量设置一系列字符串的内容
6. 原子计数器
7. GETSET 命令的妙用，请于清空旧值的同时设置一个新值，配合原子计数器使用

2. Hash——字典

在 Memcached 中，我们经常将一些结构化的信息打包成 hashmap，在客户端序列化后存储为一个字符串的值（一般是 JSON 格式），比如用户的昵称、年龄、性别、积分等。这时候在需要修改其中某一项时，通常需要将字符串（JSON）取出来，然后进行反序列化，修改某一项的值，再序列化成字符串（JSON）存储回去。简单修改一个属性就干这么多事情，消耗必定是很大的，也不适用于一些可能并发操作的场合（比如两个并发的操作都需要修改积分）。而 Redis 的 Hash 结构可以使你像在数据库中 Update 一个属性一样只修改某一项属性值。

存储、读取、修改用户属性

3. List——列表

List 说白了就是链表（redis 使用双端链表实现的 List），相信学过数据结构知识的人都应该能理解其结构。使用 List 结构，我们可以轻松地实现最新消息排行等功能（比如新浪微博的 TimeLine）。List 的另一个应用就是消息队列，可以利用 List 的 *PUSH 操作，将任务存在 List 中，然后工作线程再用 POP 操作将任务取出进行执行。Redis 还提供了操作 List 中某一段元素的 API，你可以直接查询，删除 List 中某一段的元素。

1. 微博 TimeLine
2. 消息队列

4. Set——集合

Set 就是一个集合，集合的概念就是一堆不重复值的组合。利用 Redis 提供的 Set 数据结构，可以存储一些集合性的数据。比如在微博应用中，可以将一个用户所有的关注人存在一个集合中，将其所有粉丝存在一个集合。因为 Redis 非常人性化的为集合提供了求交集、并集、差集等操作，那么就可以非常方便的实现如共同关注、共同喜好、二度好友等功能，对上面的所有集合操作，你还可以使用不同的命令选择将结果返回给客户端还是存集到一个新的集合中。

1. 共同好友、二度好友
2. 利用唯一性，可以统计访问网站的所有独立 IP
3. 好友推荐的时候，根据 tag 求交集，大于某个 threshold 就可以推荐

5. Sorted Set——有序集合

和 Sets 相比，Sorted Sets 是将 Set 中的元素增加了一个权重参数 score，使得集合中的元素能够按 score 进行有序排列，比如一个存储全班同学成绩的 Sorted Sets，其集合 value 可以是同学的学号，而 score 就可以是其考试得分，这样在数据插入集合的时候，就已经进行了天然的排序。另外还可以用 Sorted Sets 来做带权重的队列，比如普通消息的 score 为 1，重要消息的 score 为 2，然后工作线程可以选择按 score 的倒序来获取工作任务。让重要的任务优先执行。

1. 带有权重的元素，比如一个游戏的用户得分排行榜
2. 比较复杂的数据结构，一般用到的场景不算太多

二、redis 其他功能使用场景

1. 订阅-发布系统

Pub/Sub 从字面上理解就是发布（Publish）与订阅（Subscribe），在 Redis 中，你可以设定对某一个 key 值进行消息发布及消息订阅，当一个 key 值上进行了消息发布后，所有订阅它的客户端都会收到相应的消息。这一功能最明显的用法就是用作实时消息系统，比如普通的即时聊天，群聊等功能。

2. 事务——Transactions

谁说 NoSQL 都不支持事务，虽然 Redis 的 Transactions 提供的并不是严格的 ACID 的事务（比如一串用 EXEC 提交执行的命令，在执行中服务器宕机，那么会有一部分命令执行了，剩下的没执行），但是这个 Transactions 还是提供了基本的命令打包执行的功能（在服务器不出问题的情况下，可以保证一连串的命令是顺序在一起执行的，中间有会有其它客户端命令插进来执行）。Redis 还提供了一个 Watch 功能，你可以对一个 key 进行 Watch，然后再执行 Transactions，在这过程中，如果这个 Watched 的值进行了修改，那么这个 Transactions 会发现并拒绝执行。