什么是 Redis?

> Redis 是一个使用 C 语言写成的,开源的 key-value 数据库。。和 Memcached 类似,它支持存储的 value 类型相对更多,包括 string(字符串)、list(链表)、set(集合)、zset(sorted set --有序集合)和 hash（哈希类型）。这些数据类型都支持 push/pop、add/remove 及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。在此基础上，redis 支持各种不同方式的排序。与 memcached 一样，为了保证效率，数据都是缓存在内存中。区别的是 redis 会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了 master-slave(主从)同步。目前，Vmware 在资助着 redis 项目的开发和维护。

Redis 与 Memcached 的区别与比较

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。memcache 支持简单的数据类型，String。

2 、Redis 支持数据的备份，即 master-slave 模式的数据备份。

3 、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,而 Memecache 把数据全部存在内存之中

4、 redis 的速度比 memcached 快很多

5、Memcached 是多线程，非阻塞 IO 复用的网络模型；Redis 使用单线程的 IO 复用模型。

对比参数	Redis	Memcached
类型	1、支持内存 2、非关系型数据库	1、支持内存 2、key-value键值对形式 3、缓存系统
数据存储类型	1、String 2、List 3、Set 4、Hash 5、Sort Set 【俗称ZSet】	1、文本型 2、二进制类型【新版增加】
查询【操作】类型	1、批量操作 2、事务支持【虽然是假的事务】 3、每个类型不同的CRUD	1、CRUD 2、少量的其他命令
附加功能	1、发布/订阅模式 2、主从分区 3、序列化支持 4、脚本支持【Lua脚本】	1、多线程服务支持
网络IO模型	1、单进程模式	2、多线程、非阻塞IO模式
事件库	自封装简易事件库AeEvent	贵族血统的LibEvent事件库
持久化支持	1、RDB 2、AOF	不支持

如果想要更详细了解的话，可以查看慕课网上的这篇手记（非常推荐） ：**《脚踏两只船的困惑 - Memcached 与 Redis》**：https://www.imooc.com/article/23549

Redis 与 Memcached 的选择

终极策略： 使用 Redis 的 String 类型做的事，都可以用 Memcached 替换，以此换取更好的性能提升； 除此以外，优先考虑 Redis；

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

Redis 常见数据结构使用场景

1. String

常用命令: set,get,decr,incr,mget 等。

String 数据结构是简单的 key-value 类型，value 其实不仅可以是 String，也可以是数字。 常规 key-value 缓存应用； 常规计数：微博数，粉丝数等。

2.Hash

常用命令： hget,hset,hgetall 等。

Hash 是一个 string 类型的 field 和 value 的映射表，hash 特别适合用于存储对象。 比如我们可以 Hash 数据结构来存储用户信息，商品信息等等。

举个例子： 最近做的一个电商网站项目的首页就使用了 redis 的 hash 数据结构进行缓存，因为一个网站的首页访问量是最大的，所以通常网站的首页可以通过 redis 缓存来提高性能和并发量。我用 **jedis 客户端**来连接和操作我搭建的 redis 集群或者单机 redis，利用 jedis 可以很容易的对 redis 进行相关操作，总的来说从搭一个简单的集群到实现 redis 作为缓存的整个步骤不难。感兴趣的可以看我昨天写的这篇文章：

《一文轻松搞懂 redis 集群原理及搭建与使用》： https://juejin.im/post/5ad54d76f265da23970759d3

3.List

常用命令: lpush,rpush,lpop,rpop,lrange 等

list 就是链表，Redis list 的应用场景非常多，也是 Redis 最重要的数据结构之一，比如微博的关注列表，粉丝列表，最新消息排行等功能都可以用 Redis 的 list 结构来实现。

Redis list 的实现为一个双向链表，即可以支持反向查找和遍历，更方便操作，不过带来了部分额外的内存开销。

4.Set

常用命令： sadd,spop,smembers,sunion 等

set 对外提供的功能与 list 类似是一个列表的功能，特殊之处在于 set 是可以自动排重的。 当你需要存储一个列表数据，又不希望出现重复数据时，set 是一个很好的选择，并且 set 提供了判断某个成员是否在一个 set 集合内的重要接口，这个也是 list 所不能提供的。

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。Redis 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。

5.Sorted Set

常用命令： zadd,zrange,zrem,zcard 等

和 set 相比，sorted set 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。

举例： 在直播系统中，实时排行信息包含直播间在线用户列表，各种礼物排行榜，弹幕消息（可以理解为按消息维度的消息排行榜）等信息，适合使用 Redis 中的 SortedSet 结构进行存储。

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 有哪些数据淘汰策略？？？）

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（回收策略）。redis 提供 6种数据淘汰策略：

1. **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
2. **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
3. **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
4. **allkeys-lru**：从数据集（server.db[i].dict）中挑选最近最少使用的数据淘汰
5. **allkeys-random**：从数据集（server.db[i].dict）中任意选择数据淘汰
6. **no-enviction**（驱逐）：禁止驱逐数据

Redis 的并发竞争问题如何解决?

Redis 为单进程单线程模式，采用队列模式将并发访问变为串行访问。Redis 本身没有锁的概念，Redis 对于多个客户端连接并不存在竞争，但是在 Jedis 客户端对 Redis 进行并发访问时会发生连接超时、数据转换错误、阻塞、客户端关闭连接等问题，这些问题均是由于客户端连接混乱造成。对此有 2 种解决方法：

1.客户端角度，为保证每个客户端间正常有序与 Redis 进行通信，对连接进行池化，同时对客户端读写 Redis 操作采用内部锁 synchronized。 2.服务器角度，利用 setnx 实现锁。

注：对于第一种，需要应用程序自己处理资源的同步，可以使用的方法比较通俗，可以使用 synchronized 也可以使用 lock；第二种需要用到 Redis 的 setnx 命令，但是需要注意一些问题。

Redis 回收进程如何工作的？Redis 回收使用的是什么算法？

Redis 内存回收:LRU 算法（写的很不错，推荐）：https://www.cnblogs.com/WJ5888/p/4371647.html

Redis 大量数据插入

官方文档给的解释：http://www.redis.cn/topics/mass-insert.html

Redis 分区的优势、不足以及分区类型

官方文档提供的讲解：http://www.redis.net.cn/tutorial/3524.html

Redis 持久化数据和缓存怎么做扩容？

《redis 的持久化和缓存机制》：https://blog.csdn.net/tr1912/article/details/70197085?foxhandler=RssReadRenderProcessHandler

扩容的话可以通过 redis 集群实现，之前做项目的时候用过自己搭的 redis 集群 然后写了一篇关于 redis 集群的文章：**《一文轻松搞懂 redis 集群原理及搭建与使用》**：https://juejin.im/post/5ad54d76f265da23970759d3

Redis 常见性能问题和解决方案:

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

Redis 与消息队列

作者：翁伟 链接：https://www.zhihu.com/question/20795043/answer/345073457

不要使用 redis 去做消息队列，这不是 redis 的设计目标。但实在太多人使用 redis 去做去消息队列，redis 的作者看不下去，另外基于 redis 的核心代码，另外实现了一个消息队列 disque：antirez/disque:https://github.com/antirez/disque 部署、协议等方面都跟 redis 非常类似，并且支持集群，延迟消息等等。

我在做网站过程接触比较多的还是使用 redis 做缓存，比如秒杀系统，首页缓存等等。

好文 Mark

非常非常推荐下面几篇文章。。。

《Redis 深入之道：原理解析、场景使用以及视频解读》：https://zhuanlan.zhihu.com/p/28073983：主要介绍了：Redis 集群开源的方案、Redis 协议简介及持久化 Aof 文件解析、Redis 短连接性能优化等等内容，文章干货太大，容量很大，建议时间充裕可以看看。另外文章里面还提供了视频讲解，可以说是非常非常用心了。

《阿里云 Redis 混合存储典型场景：如何轻松搭建视频直播间系统》：https://yq.aliyun.com/articles/582487?utmcontent=m46529：主要介绍视频直播间系统，以及如何使用阿里云 Redis 混合存储实例方便快捷的构建大数据量，低延迟的视频直播间服务。还介绍到了我们之前提高过的 redis 的数据结构的使用场景

《美团在 Redis 上踩过的一些坑-5.redis cluster 遇到的一些问》：http://carlosfu.iteye.com/blog/2254573：主要介绍了 redis 集群的两个常见问题，然后分享了一些关于 redis 集群不错的文章。

参考：

https://www.cnblogs.com/Survivalist/p/8119891.html

http://www.redis.net.cn/tutorial/3524.html

https://redis.io/

一 redis 的安装

Redis 是 c 语言开发的。 安装 redis 需要 c 语言的编译环境。如果没有 gcc 需要在线安装：**yum install gcc-c++**

第一步：获取源码包：**wget http://download.redis.io/releases/redis-3.0.0.tar.gz**

第二步：解压缩 redis：**tar zxvf redis-3.0.0.tar.gz**

第三步：编译。进入 redis 源码目录(**cd redis-3.0.0**)。执行 **make**

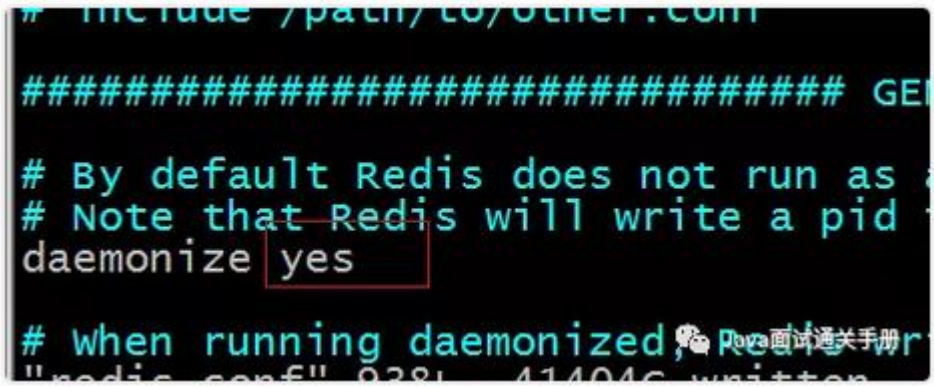
第四步：安装。**make install PREFIX=/usr/local/redis**

PREFIX 参数指定 redis 的安装目录。一般软件安装到/usr 目录下

这样 Redis 就成功装在了我们的 usr/local/redis 目录下。

第五步：设置后台启动： [root@localhost redis-3.0.0]# cp redis.conf /usr/local/redis/bin/（把/root/redis-3.0.0/redis.conf 复制到/usr/local/redis/bin 目录下）

修改配置文件：把 **daemonize** 后面的参数改为 **yes**



测试启动：[root@localhost bin]# **./redis-server redis.conf**

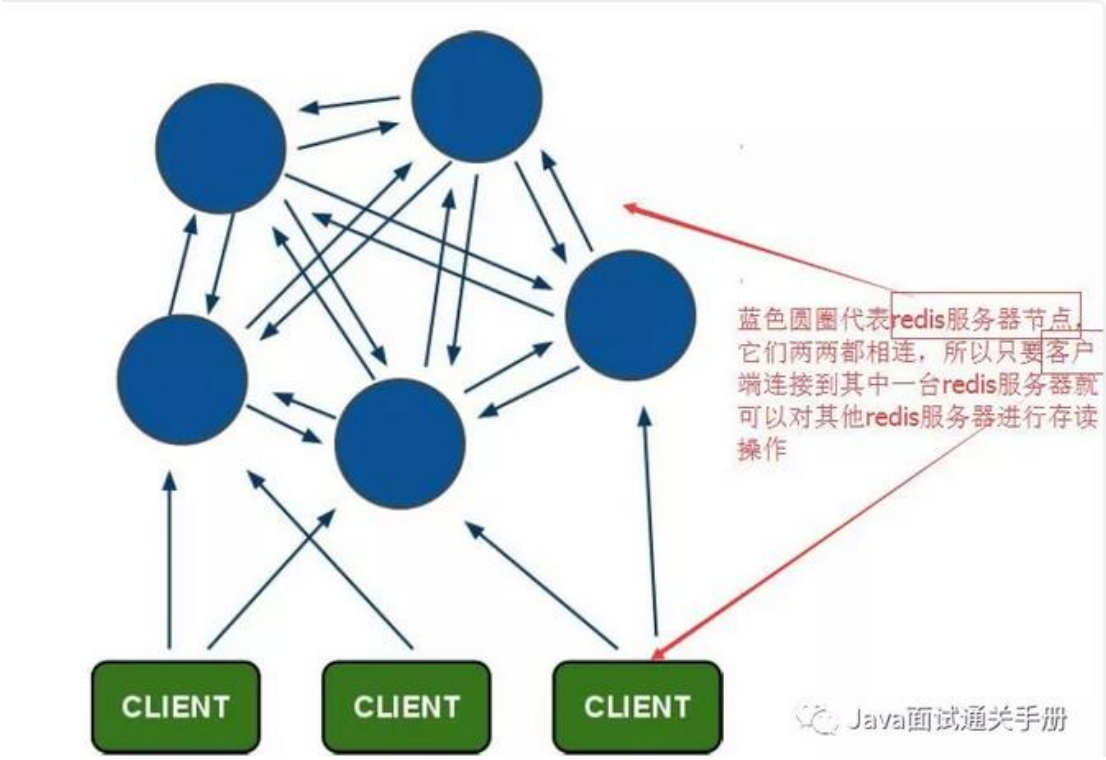
查看 redis 进程：[root@localhost bin]# **ps aux|grep redis**

二 redis 集群的搭建

2.1 redis 集群(redis-cluster)原理

3.0 版本之前的 redis 是不支持集群的，3.0 版本之前想要搭建 redis 集群的话需要中间件来找到存值和取值的对应节点。

3.0 版本以后的 redis 集群架构图：



那么这是如何实现的呢？？？

Redis 集群中内置了 **16384** 个哈希槽，当需要在 Redis 集群中放置一个 key-value 时，redis 先对 key 使用 crc16 算法算出一个结果，然后把结果对 16384 求余数，这样每个 key 都会对应一个编号在 **0-16383** 之间的哈希槽，redis 会根据节点数量大致均等的将哈希槽映射到不同的节点。

redis 集群投票机制

redis 集群中有多台 redis 服务器不可避免会有服务器挂掉。redis 集群服务器之间通过互相的 ping-pong 判断是否节点可以连接上。如果有一半以上的节点去 ping 一个节点的时候没有回应，集群就认为这个节点宕机了。

上面就是我们常说的为了容错而生的 **redis 集群投票机制**。