

Redis 高级知识

一.Redis简介

- Redis是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库，并提供多种语言的API。从2010年3月15日起，Redis的开发工作由VMware主持。从2013年5月开始，Redis的开发由Pivotal赞助。
- Redis是 NoSQL技术阵营中的一员，它通过多种键值数据类型来适应不同场景下的存储需求，借助一些高层级的接口使用其可以胜任，如缓存、队列系统的不同角色

二.Redis特性

- Redis 与其他 key - value 缓存产品有以下三个特点：
- **Redis支持数据的持久化**，可以将内存中的数据保存在磁盘中，重启的时候可以再次加载进行使用。
- Redis不仅仅支持简单的key-value类型的数据，同时还提供list, set, zset, hash等数据结构的存储。
- Redis支持数据的备份，即**master-slave模式的数据备份**。

三.Redis 优势

- 性能极高 – Redis能读的速度是110000次/s,写的速度是81000次/s 。
- 丰富的数据类型 – Redis支持二进制案例的 Strings, Lists, Hashes, Sets 及 Ordered Sets 数据类型操作。
- 原子 – Redis的所有操作都是原子性的，同时Redis还支持对几个操作全并后的原子性执行。
- 丰富的特性 – Redis还支持 publish/subscribe, 通知, key 过期等等特性。
- 可实现持久化
- 主从复制，实现主写从读
- 高可用和分布式，高可用是以集群为前提，比如说哨兵机制，可实现自动监测和防止宕机就是高可用的体现

四.Redis 缺点

redis是内存型数据库，因此容易受到数据库物理内存的限制，redis的数据类型比较固定，在Redis中数据存储最多可以容纳的数据长度是512M，适用于较小数据量的高性能操作和运算。

mysql跟redis结合起来使用，用redis做缓存，会出现的问题：

1.双向写入不一致

2.缓存穿透：即黑客故意去请求缓存中不存在的数据，导致所有的请求都怼到数据库上，从而数据库连接异常。

解决：

3.缓存雪崩：即缓存同一时间大面积的失效，这个时候又来了一波请求，结果请求都怼到数据库上，从而导致数据库连接异常。

解决：给缓存的失效时间，加上一个随机值，避免集体失效

4.redis的并发key竞争问题（保存时间戳，乐观锁）

五.Redis应用场景

- 用来做缓存(ehcache/memcached)——redis的所有数据是放在内存中的（内存数据库），可实现缓存的高效查询【如注册的时候存放短信验证码】
- 可以在某些特定应用场景下替代传统数据库——比如社交类的应用
- 在一些大型系统中，巧妙地实现一些特定的功能：session共享、购物车【项目中运用】
- 可以做消息队列：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能作为很好的消息队列平台来使用。
- 排行榜/计时器：Redis在内存中对数字的进行递增或者递减的操作实现的非常好

六.Redis设置长连接

在redis配置文件中，tcp-keepalive=1,默认为0,0表示禁用长连接

Redis缓存和数据库（mysql）怎么做数据的同步

数据的脏读：脏读又称无效数据的读出，是指在数据库访问中，事务T1将某一值修改，然后事务T2读取该值，此后T1因为某种原因回滚了，这就导致了T2读到的数据是无效的。

七.redis缓存和数据库（mysql）怎么做数据同步？

答：redis缓存跟数据库mysql数据同步的话，不管是先删缓存然后写数据库，还是先写数据库，在删除缓存，都有可能出现数据不一致的问题。

举例：假如你先删除缓存，在数据没有写入到数据库mysql之前，另一个线程来读取数据，发现缓存为空，就去读取mysql，此时读到的是脏数据

假如你先写入到数据库，然后删除缓存，此时由于某种原因，删除缓存没有成功，此时另外一个用户过来读取数据，也会造成脏读

解决方法：先删除缓存，在写入数据库，然后休眠500ms(根据业务而定)，在删除缓存。

可以确保读请求结束，写请求可以删除读请求造成的缓存脏数据

八.Redis分布式锁？

是什么？redis分布式锁处理超高并发，比如发送短信量太大，一个redis承受不了，所以就要分布式配置多台redis服务器，分布式锁是控制分布式系统或者不同系统之间共同访问共享资源的一种锁实现，如果不同的系统或者同一个系统的不同主机之间共享了某个资源时，往往需要互斥来防止彼此干扰来保证一致性。

需要解决问题：

- 1.互斥性：任何时候，只能有一个客户端获取锁
- 2.安全性：锁只能被持有该锁的客户端删除，不能由其他客户删除
- 3.死锁：获取锁的客户端因为某些原因未能释放锁，其他客户段再也无法获得该锁
- 4.容错：当部分节点宕机时，客户端仍然能够获取锁和释放锁。

九.redis宕机如何处理？

redis如果只有一台，肯定会丢失数据

多态redis或者redis集群，宕机分两种情况

一.如果是redis从服务器宕机

配置主从的时候才配置的从服务器，从服务器会自动读取主服务器的操作日志，达到主从复制的目的，如果从服务器宕机，可以从如下解决：

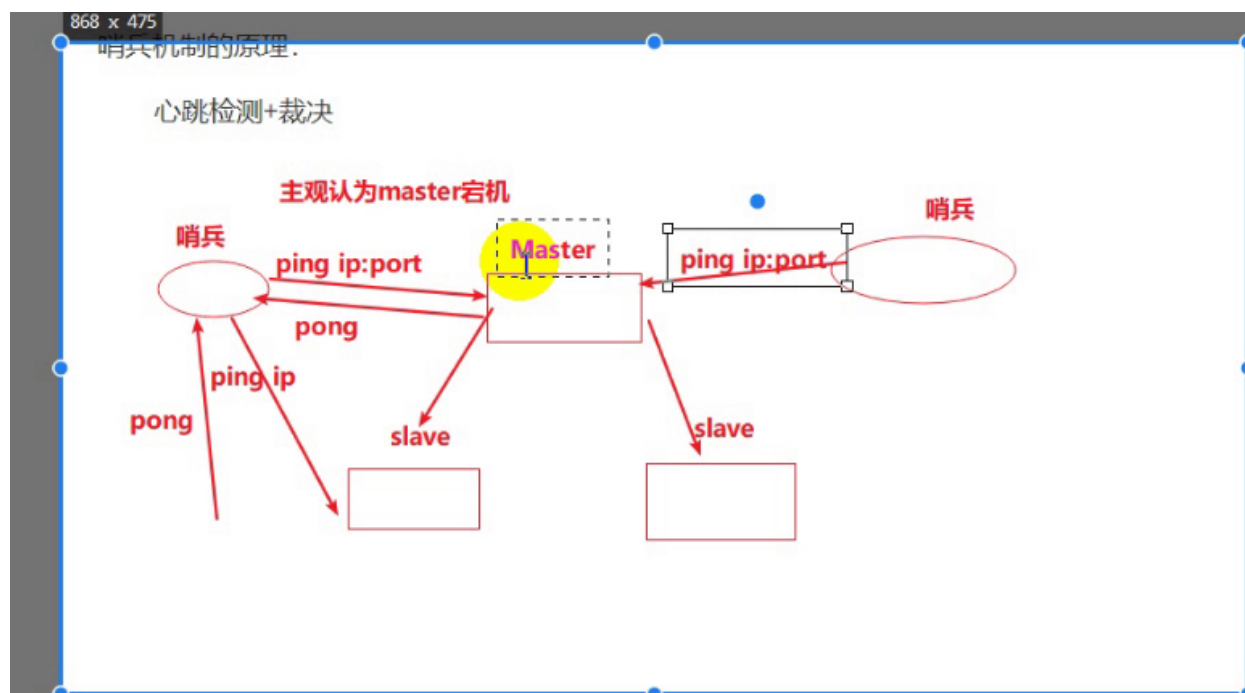
- 1.从服务器重新启动，连接主服务器，自动复制主服务器的数据；
- 2.从服务器如果实现数据的持久化，可以直接连接主服务器，实现数据的增量就可以了（数据增量是指从服务器宕机到连接到主服务器这段时间的数据复制到从服务器）

二.如果是主服务器宕机

确认是否做了持久化，如果没做持久化，不能立刻重启服务器，不然会造成数据丢失，应该如下操作

- 1.将一台从服务器升级为主服务器，执行命令 `slave of one` ,
- 2.重新启动主服务器，主服务器作为从的服务器，连接到主的上做主从复制

上面操作太过繁琐，如果操作失误，就会造成数据丢失，使用哨兵机制（配置就可以了），哨兵机制的原理是用心跳检测和裁决来完成的



十.redis为什么比mysql快？

- 答：
- 1.redis是内存型数据库，mysql数据是存在磁盘中的，从内存取数据当然比从磁盘取数据快
 - 2.redis是以键值对的形式存储，查找时间复杂度是 $O(1)$ 常数级别，而mysql查询存储是依靠引擎INNODB和MYISAM，数据的存储和查找是依靠B-tree结构，查找时间复杂度是 $O(\log n)$ 对数级别
 - 3.redis查询数据只要找到对应的键就能得到值，而mysql的数据是保存在表中，查询数据需要查表或者需要联表查询或者通过外键查询，需要操作磁盘，速度相对较慢
 - 4.redis是单线程多路I/O复用（IO复用机制可以同时监控多个描述符，当某个描述符就绪（读或写就绪），则立即通知相应程序进行读或写操作），避免了开多线程的系统资源的消耗和时间，对于多核的电脑操作redis,可以把数据进行分区。

十一.redis如何实现持久化

redis实现持久化的方式有两种，一种是RDB（快照），另一种是AOF（append only file）

RDB

是redis默认的持久化的方式，它的实现原理是将redis中内存中的数据写入到本地的二进制文件dump.rdb

具体实现方法：redis会调用fork(分叉)函数，会从主进程复制出一个子进程，主进程继续处理用户请求，子进程会去遍历内存中的所有数据，然后生成一个副本二进制文件dump.rdb，将完成副本文件之后，就替换磁盘中旧的文件，通过配置文件做到

#RDB方式的持久化是通过快照(snapshotting)完成的,当符合一定条件时Redis会自动将内存中的所有数据进行快照并存储在硬盘上。进行快照的条件可以由用户在配置文件中自定义,由两个参数构成:时间和改动的键的个数。当在指定的时间内被更改的键的个数大于指定的数值时就会进行快照。RDB是Redis默认采用的持久化方式,在配置文件中已经预置了3个条件:

```
save 900 1 # 900秒内有至少1个键被更改则进行快照
save 300 10 # 300秒内有至少10个键被更改则进行快照
save 60 10000 # 60秒内有至少10000个键被更改则进行快照
```

上述命令之的是如果900秒内有1个key发生了变化，生成一份快照文件,如果再300秒内有10个key发生了改变，生成一份快照文件，如果在60秒内有10000个key发生了改变，生成一份快照文件。

save可以设置多个，就是多个snapshotting检查点，每到一个检查点，就会去检查一下，是否有指定的key数量发生变化，如果有就生成一个新的dump.rdb文件

RDB模式优缺点：

优点：1.使用起来简单，只要修改配置文件就可以了

2.读写速度快，因为数据保存在二进制文件中

缺点：生成快照的时候，如果出现断电，容易造成数据丢失

AOF

实现的原理：将redis所有的写的操作都追加到日志文件中（磁盘），当redis重启时，会执行日志文件中的写的命令来重建内存的数据，可以在配置文件中通过配置告诉redis我们想要使用fsync函数强制写入到磁盘的时间。

配置分为三种：

1. appendfsync no（不写）

2. appendfsync everysec（每一秒）

3. appendfsync always（每一次写操作）

优缺点：

1.优点：1.可以更好地保存数据，持久化数据，保存的aof日志文件是以redis协议格式保存的，非常容易读取。

2.缺点：数据保存在日志文件中，受文件大小的影响，而且数据读取的速度没有RDB快，文件的大小也比RDB大。