

1. 说一下你在项目中的redis的应用场景？

1、5大value类型

2、基本做缓存

3、业务场景：服务状态 token、session 延伸思考，有哪些数据结构

2. Set、Zset分 别用于哪些场景？

3. redis是单线程还是多线程？

1、工作线程单线程

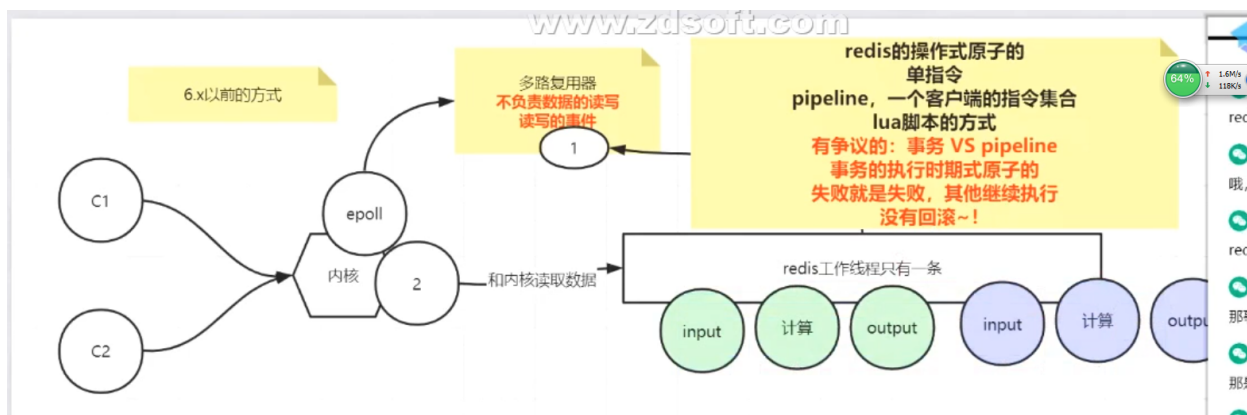
2、高版本IO多线程

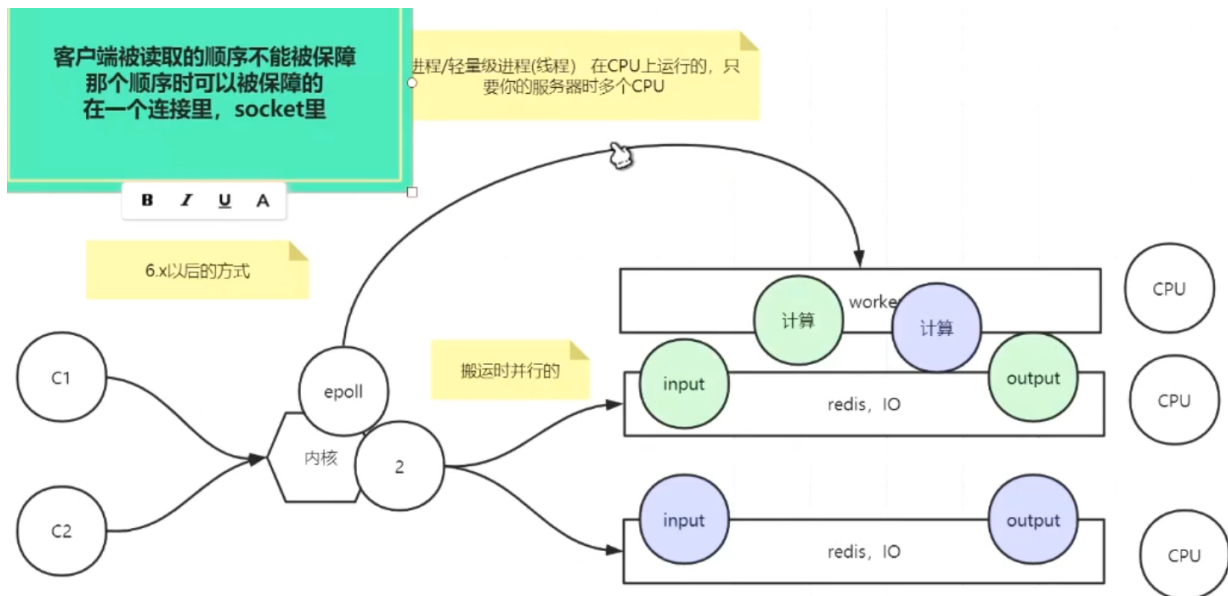
3、【io课程】面向IO模型，（react模型）内核吧数据搬运到程序里，netty

4、单线程，满足redis的串行原子，只不过IO多线程后，把输入/输出放到更多的线程里去并

行，好处如下:1，执行时间缩短，更快;2，更好的压榨系统及硬件的资源(网卡能够高效的使

用);





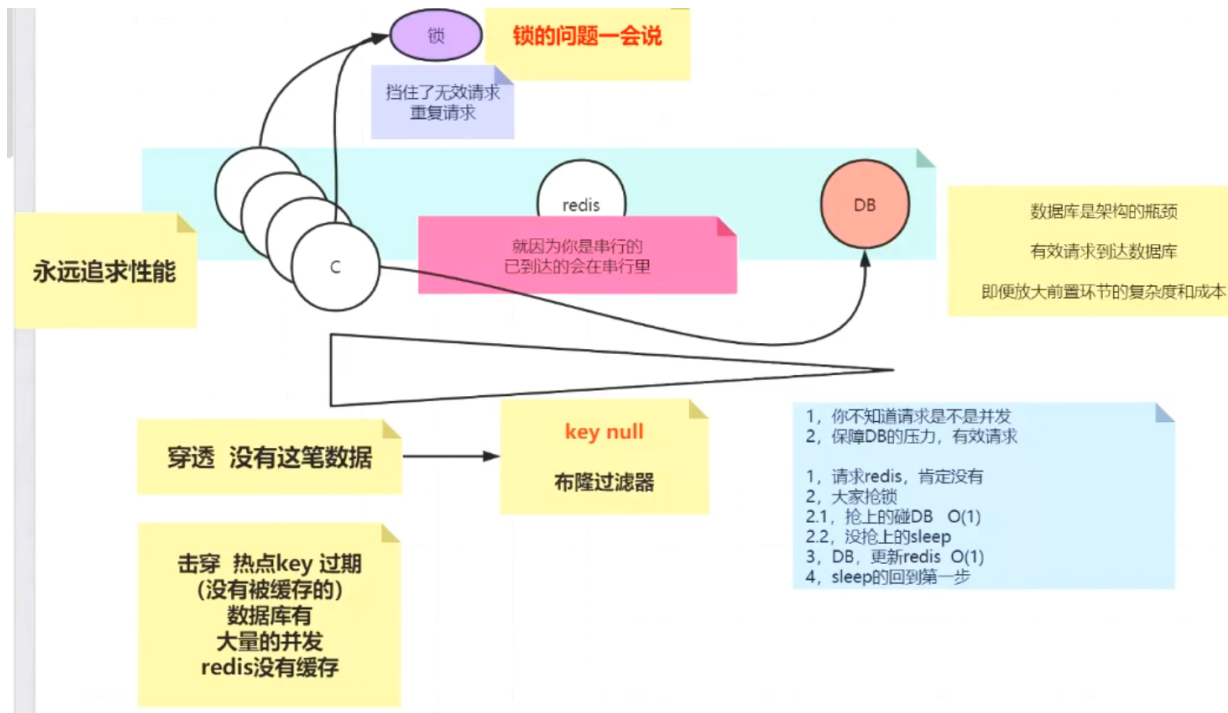
4. redis存在线程安全的问题吗?为什么?

J重复2中的单线程串行

redis可以保障内部串行

外界使用的时候要保障, 业务上要自行保障顺序~!

5. 遇到过缓存穿透吗?详细描述一下。



6. 遇到过缓存击穿吗?详细描述一下。

7. 如何避免缓存雪崩?

以上问题, 核心就是避免DB无效/重复请求, 结合图去理解

8. 缓存如何回收的?(怎么删除过期key?)

1、工作线程单个, 后台轮询分批删除过期

2、请求事件判断是否已经过期

尽量把内存无用的空间回收

8.1 缓存如何淘汰？

0、内存空间不足的情况下

1、淘汰机制里有不允许淘汰

2、lru/lfu/random/ttl

3、全空间

4、设置过期的集合

9. 如何进行缓存预热？

1、提前把数据塞入redis，你知道那些是热数据吗？

2、开发逻辑上也要规避差集（没缓存的），会造成击穿，穿透，雪崩，实施锁方案

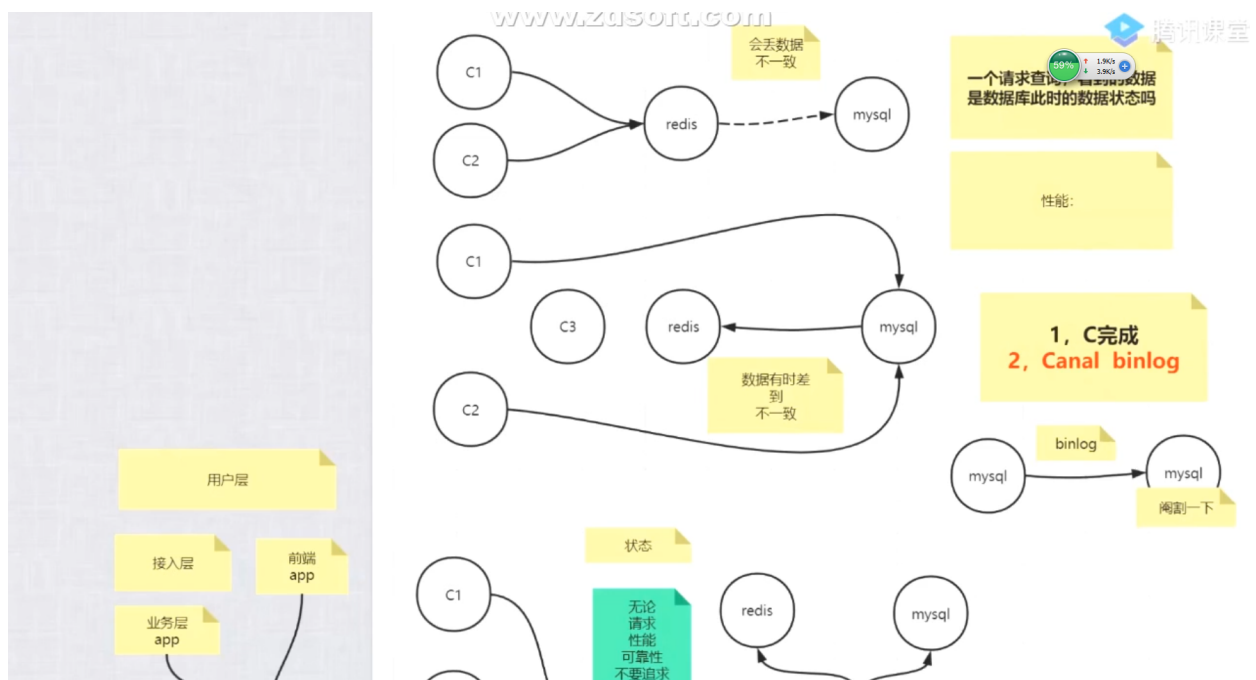
3、一劳永逸，未来也不怕了

10. 数据库与缓存不一致如何解决？

1、使用分布式事务来解决，（意义不大），读多，写稀有情况下

2、Canal binlog

3、MQ 最终一致性



11. 简述一下主从不一 致的问题？

1, redis 的默认是弱一一致性， 异步的同步

2, 锁不能用主从(单实例/分片集群/redlock)==>redisson

3, 在配置中提供了必须有多少个Client连接能同步， 你可以配置同步因子， 趋向于强制一致性

4、wait 2 0

5、3 4 违背reids初衷

12. 描述一下redis持久化原理?

当前线程阻塞服务 异步后台进程完成持久化, fork+copywrite

13. Redis也打不住了, 万级流量会打到DB上, 该怎么处理?

4. 5. 6

14. Redis是怎么删除过期key的?

15. Redis有哪些持久化方式?

1、rdb , aof ;主从同步

2、高版本: 开启AOF , AOF可以通过执行日志得到全部内存数据的方式, 追求性能

2.1, 体积变大, 重复无效指令重写, 后台用线程把内存的kv生成指令写个新的aof

2.2, 4.x新增更有性能模式:把重写方式换成直接RDB放到aof文件的头部, 比2.1的方法快

了, 再追加日志

17. redis中的事务三条指令式什么, 第三条指令到达后执行失败了, 怎么处理

16. 为什么使用setnx?

1、原子性操作, 不存在的情况下完成创建

2、如果要做分布式锁, 用set k v nx ex(不存在, 过期时间, 避免死锁)

18. redis实现分布式锁的指令