1. 管道

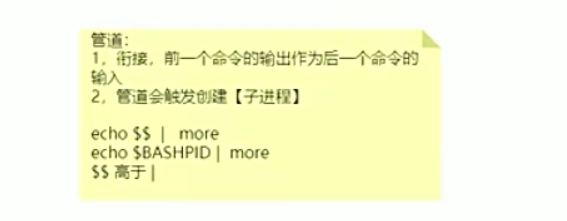
（1）出现原因：

客户端发送很多命令给Redis，每发一条命令就要进行一次IO通信，并且一条命令结果返回之后才能发送第二条命令。

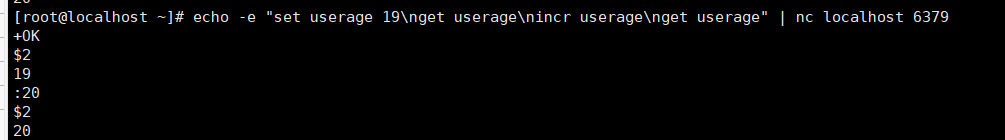
导致系统调用频繁、网络IO频繁，效率不高。

为了减少系统调用、网络IO，降低通信成本，出现管道技术。

1. 定义



通过管道nc可以往Redis发送多个命令。



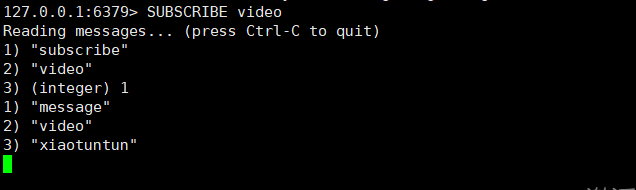
1. Redis冷启动

Redis进程启动是空的，什么数据也没有。

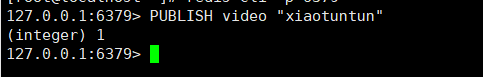
此时，如果希望Redis启动时预加载数据，可以通过文件批量插入数据。

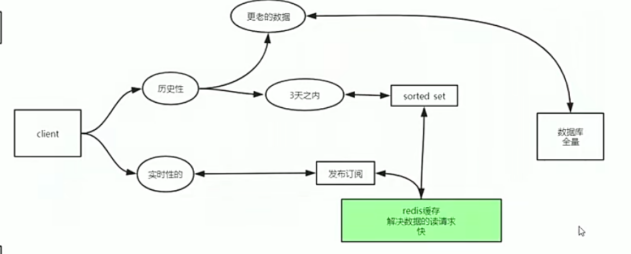
1. 发布订阅（观察者模式）

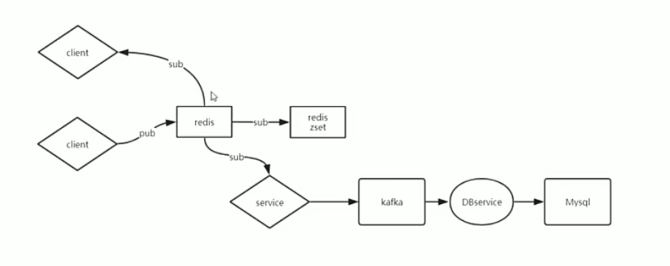
订阅subscribe



发布publish







1. 事务

事务是一个原子操作，事务中的所有命令要么全部执行，要么全部不执行。

1. 指令

乐观锁（CAS：check-and-set操作）：watch监控key，事务执行的时候发现key被更改（key被删除、key对应的value被修改）则不执行事务。取消是unwatch。

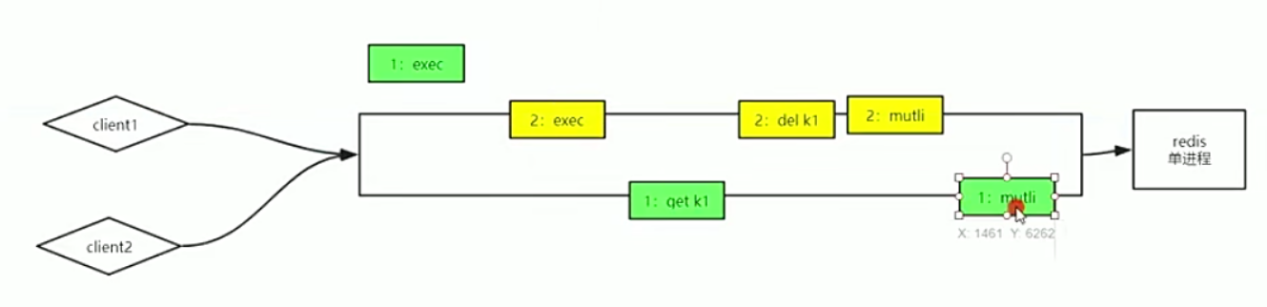
开启事务：multi

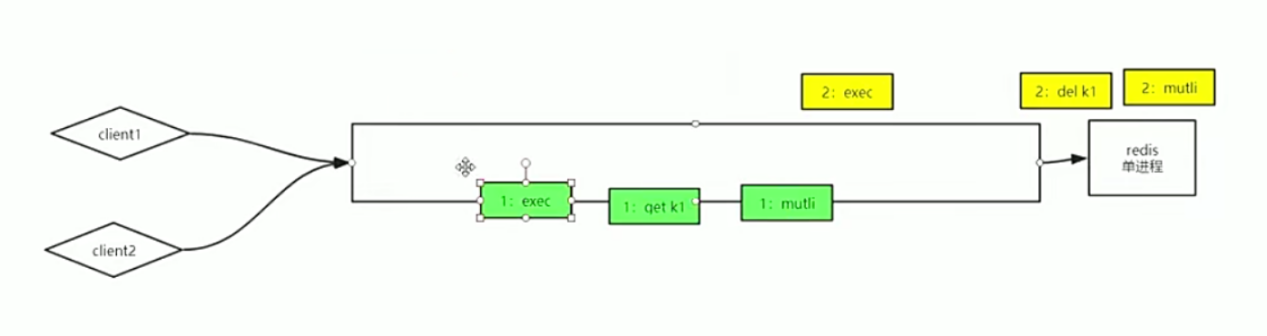
执行事务：exec

取消事务：discard

（2）客户端2的exec先到达，则先执行客户端2的事务

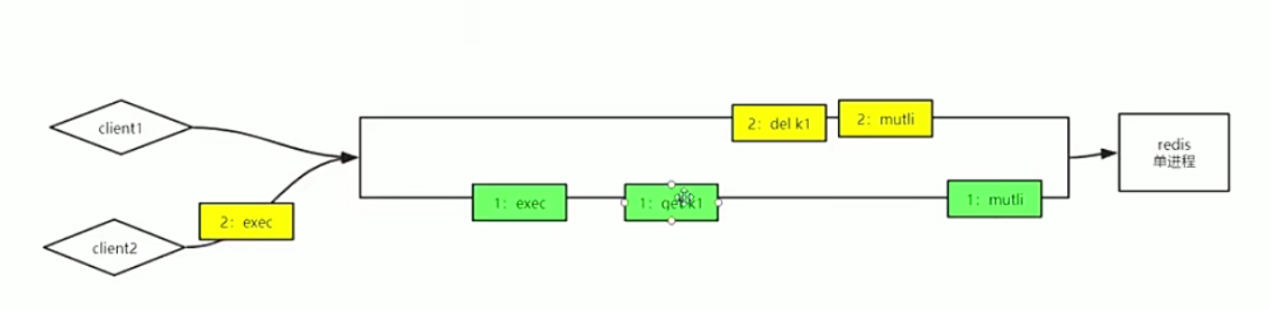
谁的exec先到达，谁就先执行



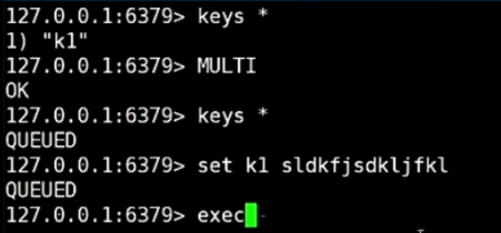
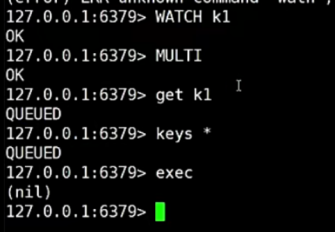


（3）客户端1的exec先到达，则先执行客户端1的事务

谁的exec先到达，谁就先执行



（4）watch监控



1. 布隆过滤器（通过位图代替key-value）
2. 安装

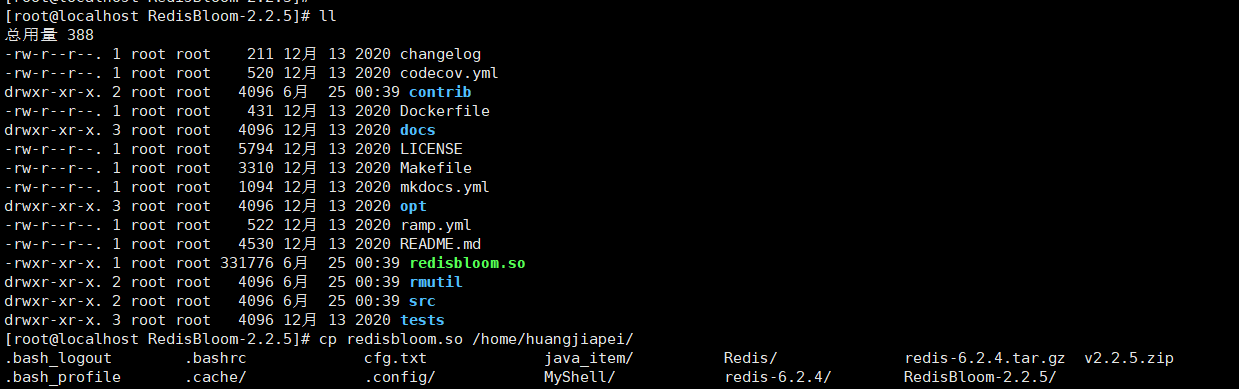


下载源码：

wget <https://github.com/RedisBloom/RedisBloom/archive/refs/tags/v2.2.5.zip>

编译so：

Make



Redis启动加载布隆过滤器：

20210625011342

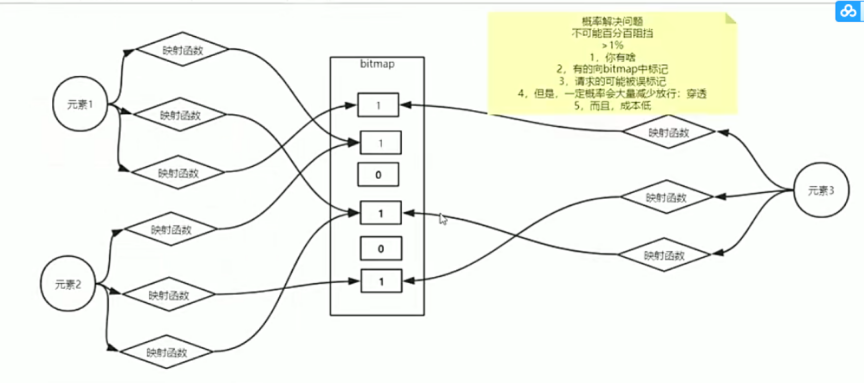
1. 介绍

原理：通过hash计数如果每位都为1，则表示数据库可能存在，不进行过滤；否则只要有1位为0，则表示数据库肯定不存在，对其进行过滤。

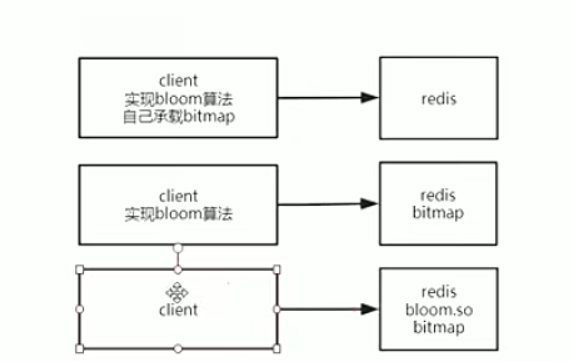
优点：通过位图，节省内存空间。如果不使用位图，则每个商品都要设置key-value，就很耗费空间。

问题：防止攻击，大概率过滤掉不存在的东西，但是可能会有穿透（过滤器无法过滤掉，但是Redis不存在的情况，直接访问数据库）。

解决方法：Redis设置该key-value，接下来不走布隆过滤器，查询该key-value，发现不存在则过滤掉。



（3）三种架构:



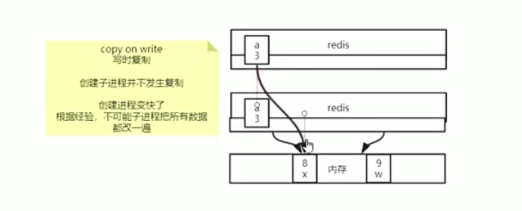
1. fork
2. 含义

创建子进程。每个进程都有自己的虚拟地址，需要通过cpu的mmu映射到实际的内存地址（物理内存地址），创建的子进程也有自己的虚拟地址。

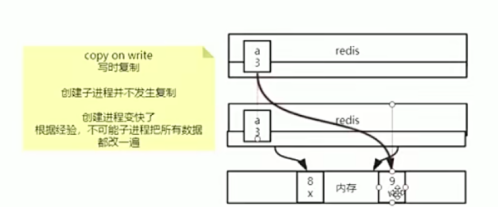
1. 父子进程的数据

需要保证父子进程之间数据隔离

* 创建子进程时，不发生数据复制



* 写时复制，copy on write



1. 优点

fork创建进程速度快，只有在数据发生变化的时候才会复制。

占用空间小。

