



链滴

## Redis-04- 事务与锁机制

作者: [Anileh](#)

原文链接: <https://ld246.com/article/1652105918363>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 1.Redis的事务

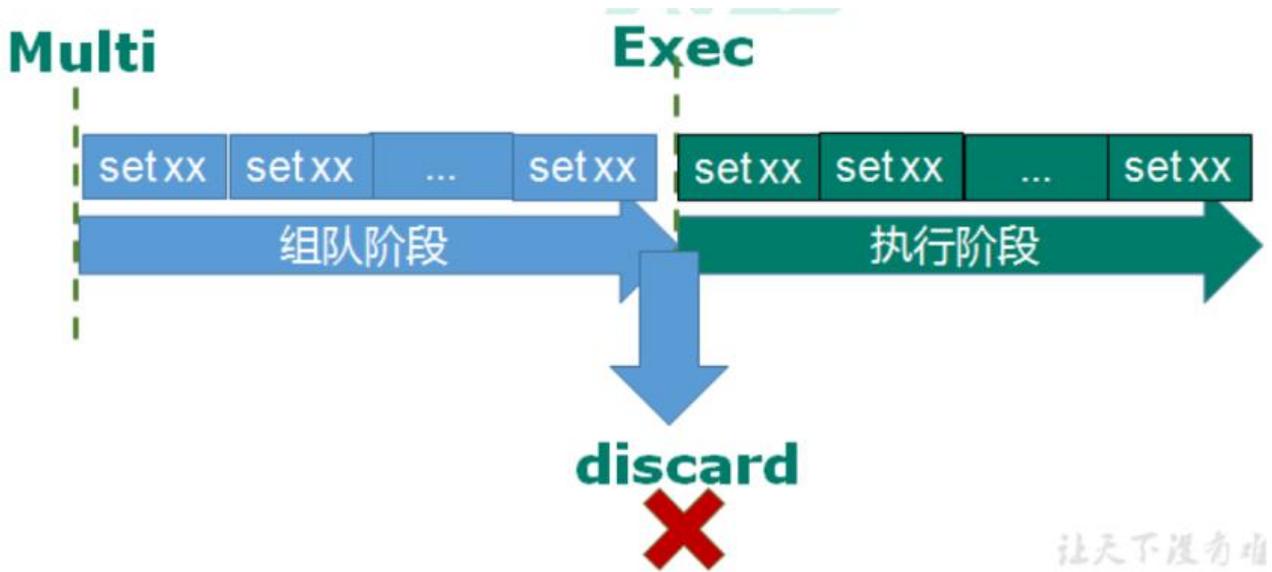
## 1.1 事务简介

Redis事务是一个单独的隔离操作，事务中的所有命令都会序列化、按顺序执行。事务在执行的过程中，不会被其他客户端发送的命令打断。

主要作用：串联多个命令防止别的命令插队。

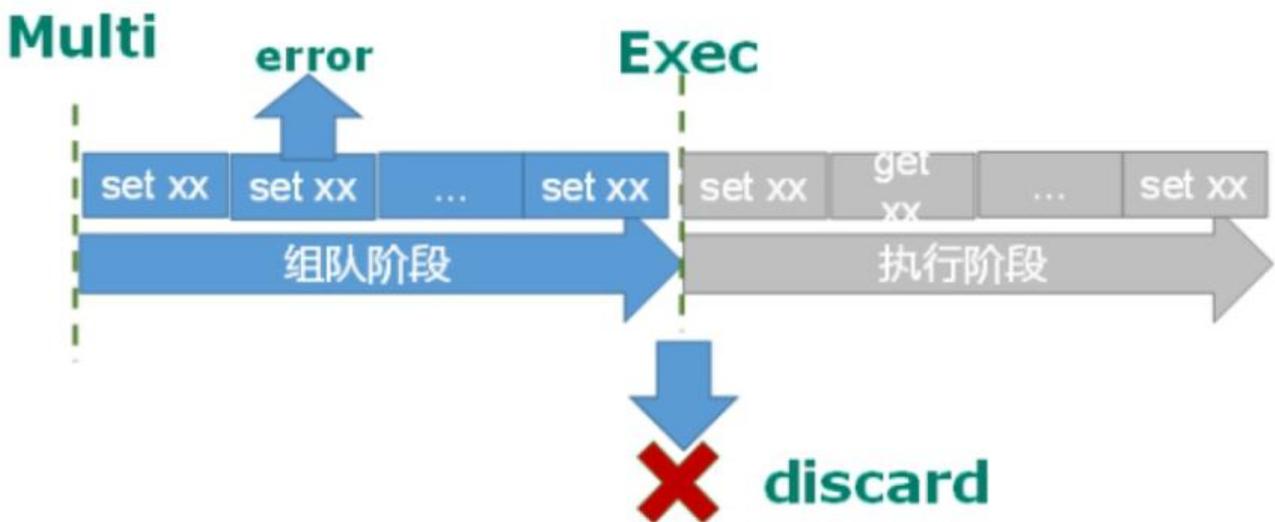
从输入Multi命令开始，输入的命令都会一次进入命令队列中，但不会执行，直到输入Exec后，Redis将之前的命令队列中的命令依次执行。

组队的过程可以通过discard来终止

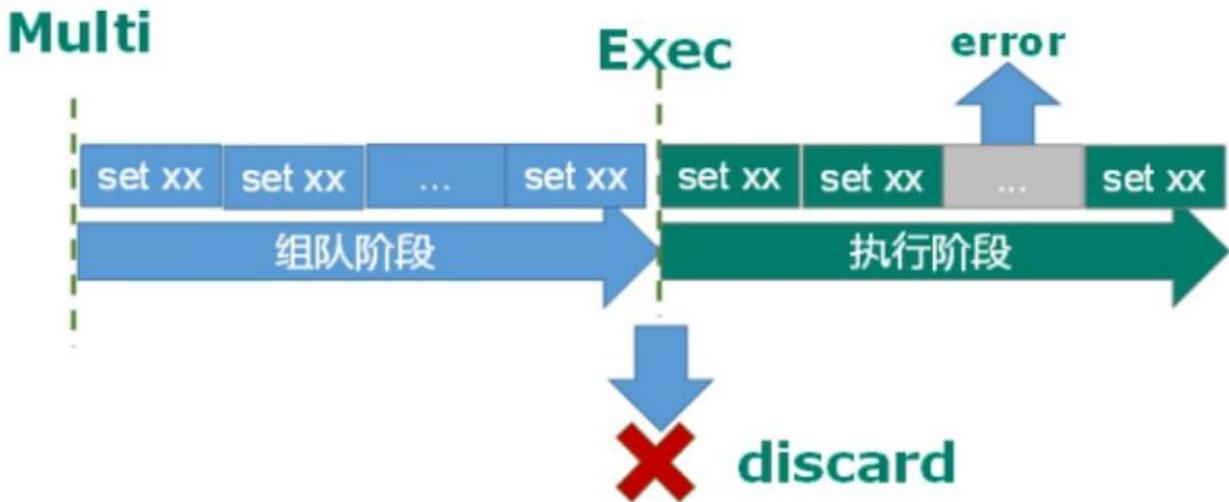


## 1.2 事务的错误处理

组队中某个命令出现了错误，执行时整个队列都会被取消



如果执行阶段某个命令报告错误，则只要报错的命令不执行，而其他的命令照常执行，不进行回滚。



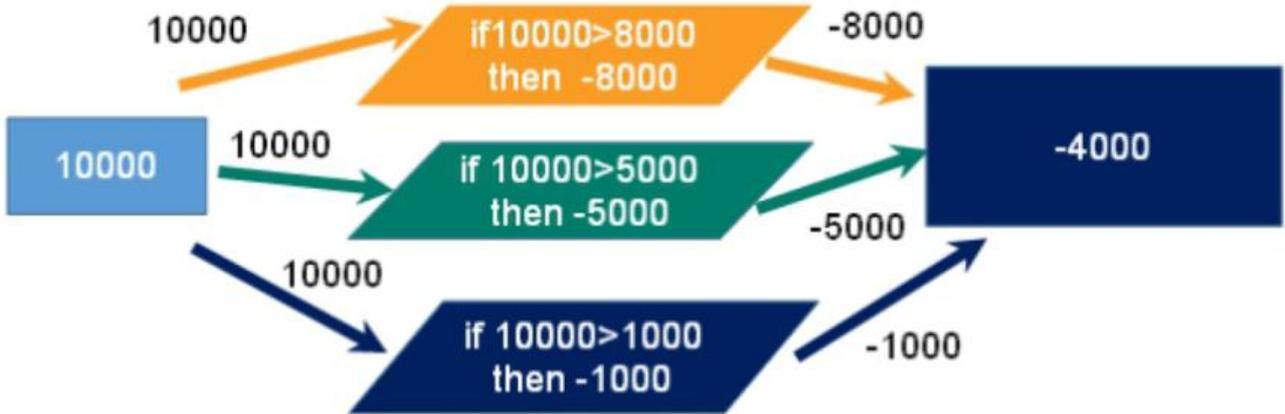
## 2 锁机制

案例：

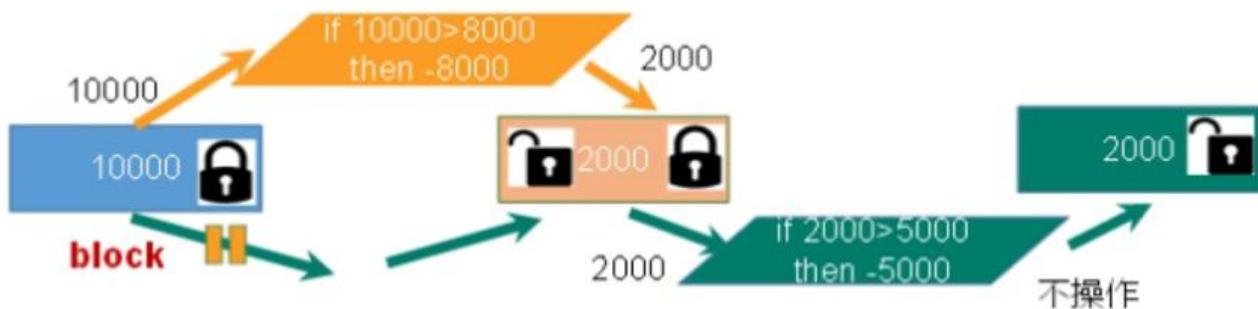
一个请求想给金额减8000

一个请求想给金额减5000

一个请求想给金额减1000

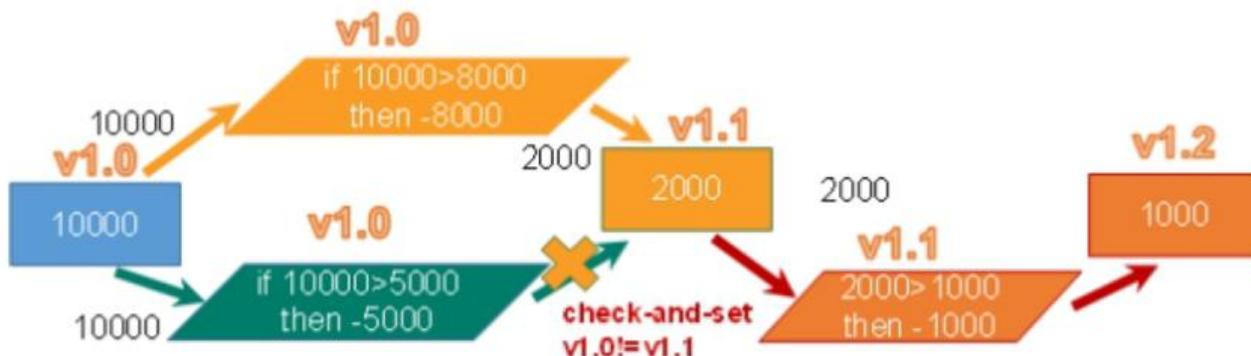


### 2.1 悲观锁



**悲观锁(Pessimistic Lock)**, 顾名思义, 就是很悲观, 每次去拿数据的时候都认为别人会修改, 所以每次在拿数据的时候都会上锁, 这样别人想拿这个数据就会 block 直到它拿到锁。传统的关系型数据库里边就用到了很多这种锁机制, 比如行锁, 表锁等, 读锁, 写锁等, 都是在做操作之前先上锁。

## 2.2 乐观锁



**乐观锁(Optimistic Lock)**, 顾名思义, 就是很乐观, 每次去拿数据的时候都认为别人不会修改, 所以不会上锁, 但是在更新的时候会判断一下在此期间别人有没有去更新这个数据, 可以使用版本号等机制。乐观锁适用于多读的应用类型, 这样可以提高吞吐量。Redis 就是利用这种 check-and-set 机制实现事务的。

## 2.3 WATCH key [key ...]

在执行multi之前, 先执行watch key1 ....可以监视多个key, 如果在事务执行之前这些key被其他命令动, 那么改事务将被打断执行。

```
127.0.0.1:6379> WATCH balance
OK
127.0.0.1:6379> MULTI
OK
127.0.0.1:6379> DECRBY balance 10
QUEUED
127.0.0.1:6379> INCRBY debt 10
QUEUED
127.0.0.1:6379> EXEC
1) (integer) 60
2) (integer) 40
```

## 2.4 unwatch

取消watch命令的监视

如果在执行watch命令之后, exec命令或discard命令先被执行, 那么此UNwatch命令将不会执行

## 3.Redis事务的特性

- 单独的隔离操作：事务中的所有命令都会序列化执行，执行过程不会被其他客户端的请求打断
- 没有隔离级别的概念：队列中的命令在没有提交前都不会被实际执行
- 不保证原子性：事务中如果有命令执行失败，之后的命令仍然会被执行，即没有回滚操作