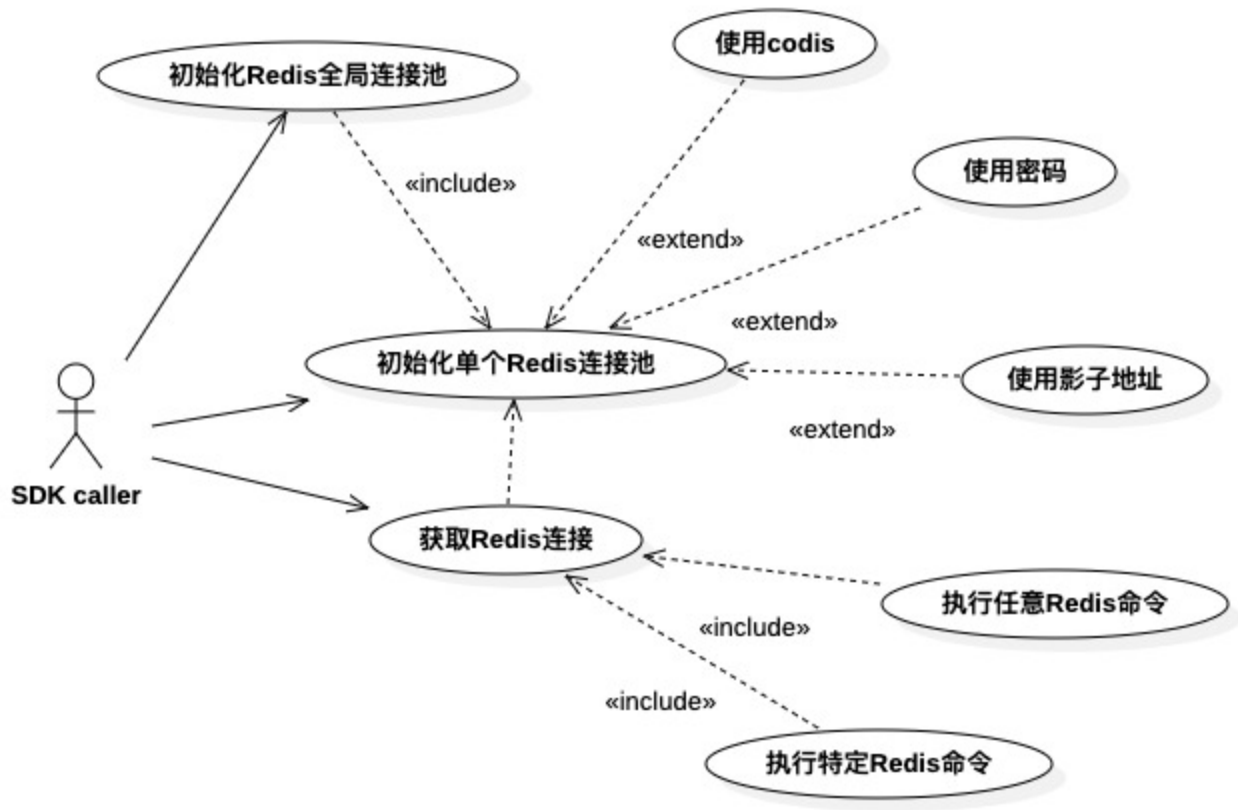


技术方案

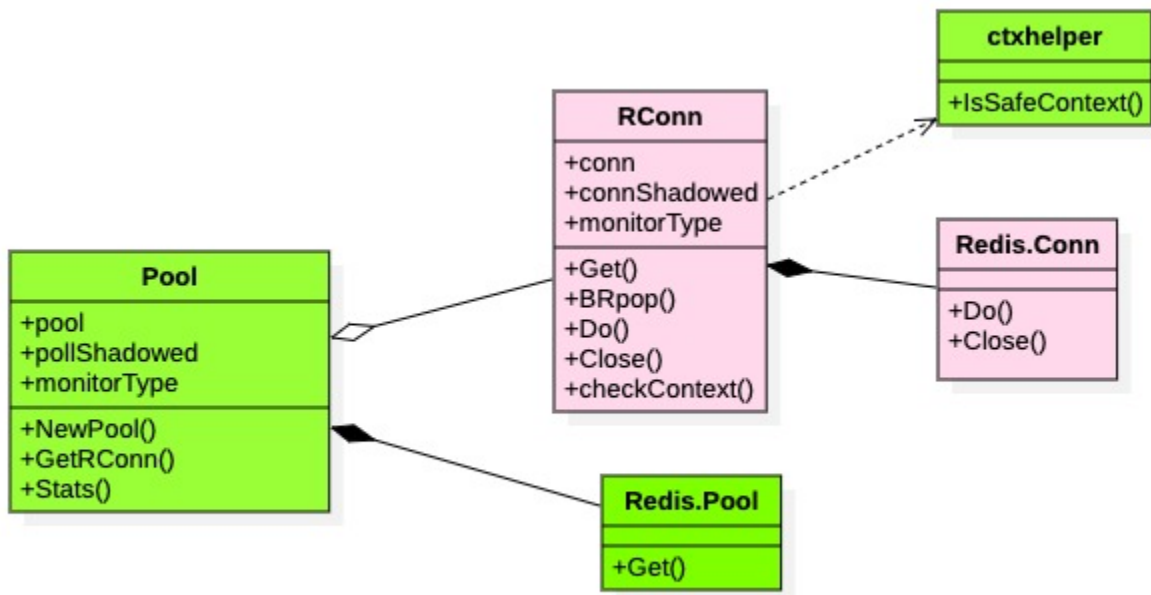
项目背景和目的

封装redis client，集成CAT监控功能，对业务方提供常见的Redis操作，并保持与官方的命令行为一致。同时提供了影子流量功能，根据传入的Ctx决定使用的是哪种类型的Redis集群。

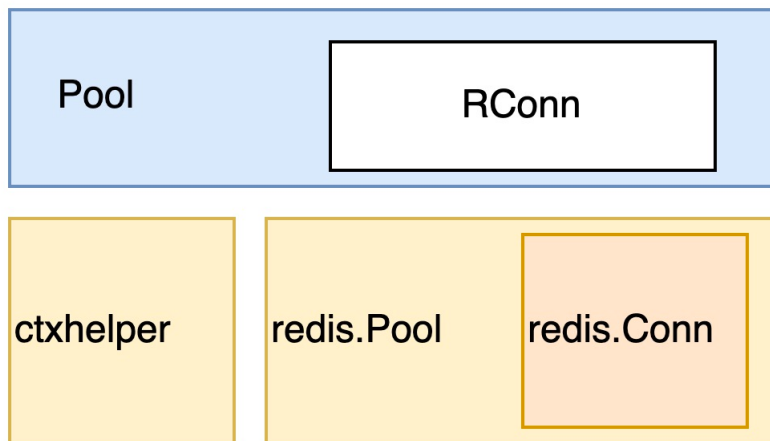
系统用例



领域模型



应用架构



详细设计

API设计

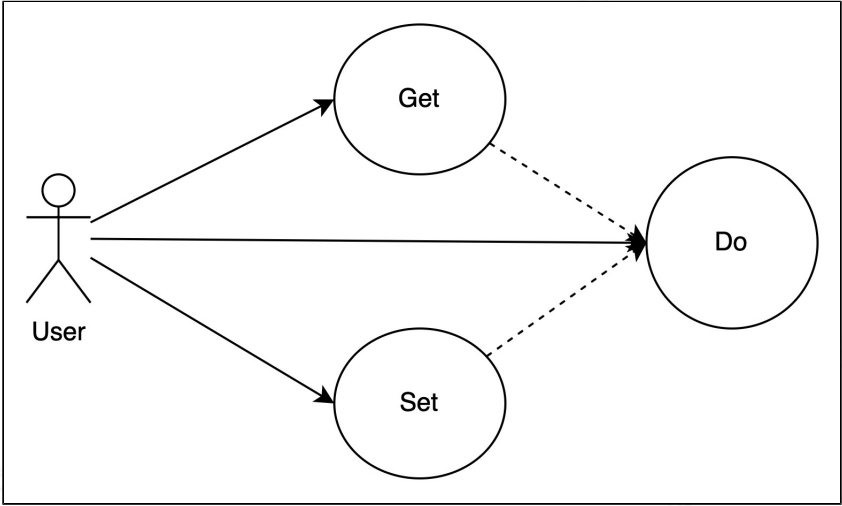
该package以RConn结构体暴露了原来redis连接的一部分API，其本质是通过redis.Conn接口中的Do()来实现的。

接口	Redis客户端	封装后的支持的接口
Do()		
Close()		
Err()		
Send()		

Revicee()		
Flush()		

封装的接口	是否与Redis行为一致
Set	
Get	
Delete	
SetEx	
SetNX	
SetNXEX	
Hget	
Hset	
Hdel	
Hmget	
HgetAll	
Hkeys	
LLen	
Lpush	
Rpoplpush	
Lrem	
BRpop	
Exists	
IncrBy	
Incr	

RConn结构带有的几乎所有的方法都是基于redis.Conn的一层解析封装，例如Hget、Hdel，都是将输入的数据转化成Redis Server接收的数据格式，然后执行RConn.Do函数，最后将返回值解析成正确的结构并返回。但BRpop函数实现不同。



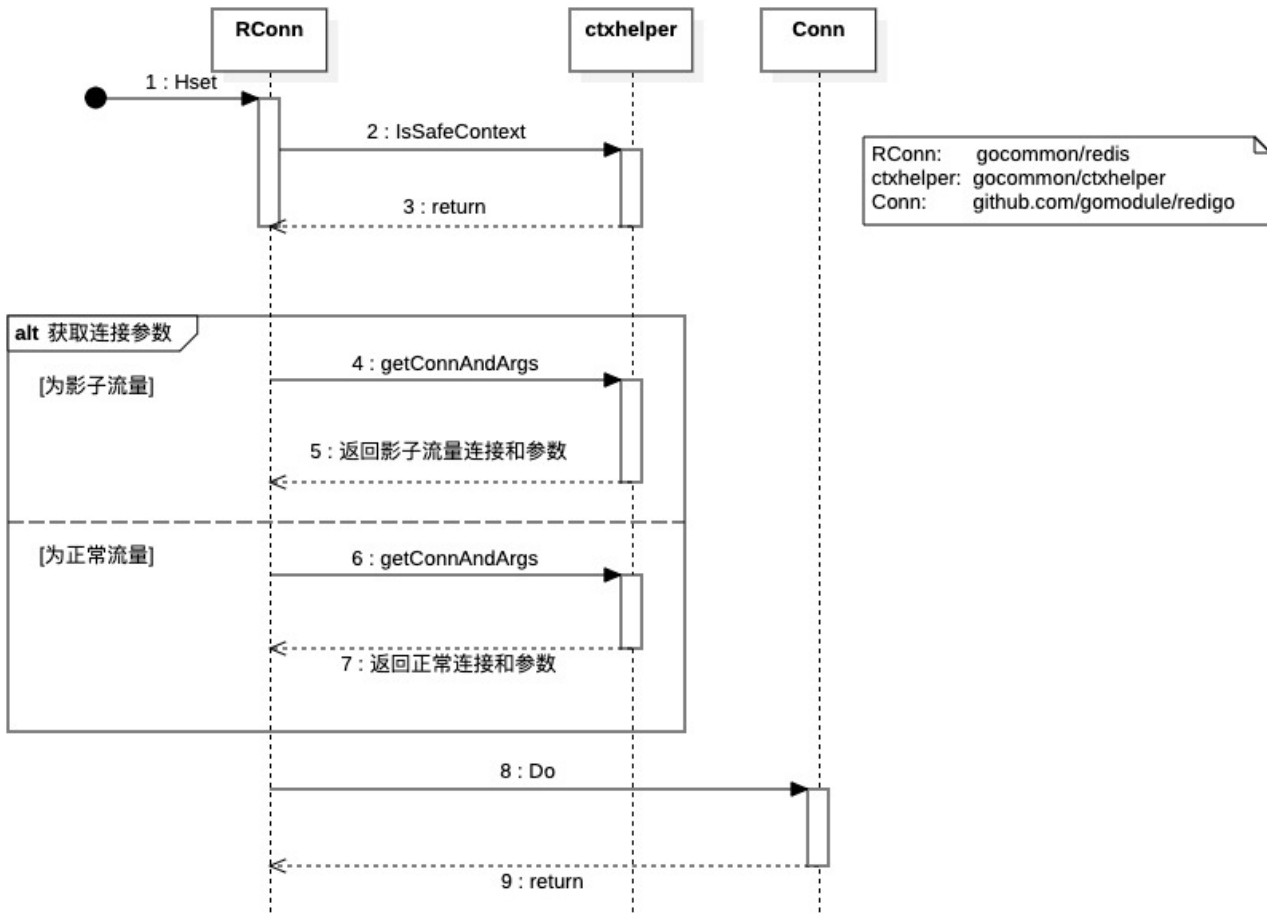
该图描述了RConn提供的API之间的依赖关系，仅展示了Get、Set与Do的关系，其余的API均与Get类似。

RConn.Do

RConn.Do函数是对于redis.Conn的一层封装，通过调用RConn.Do函数可以为函数体中的操作全都加上一层Transation的监控，其他函数不需要自己添加监控代码，从而减少了代码的冗余量。

getConnAndArgs调用了ctxhelper中的函数，检查ctx中是否带有影子流量的标记，这次redis读写操作是影子流量，则会将数据读写都引到对应的影子redis连接。

下图展示了除BRpop之外的API，调用RConn.Do函数的流程



BRpop

BRpop因为Codis集群不支持该命令，因此在此处使用轮询来模拟服务端阻塞的状态。因此也不能直接调用上面的RConn.Do函数，而是使用了轮询Rpop来模拟实现阻塞状态。但使用轮询来实现这个阻塞效果，会出现对应的时间延迟，即可能n毫秒时业务发送了请求，此时启动计时器，在n+100时才会发送下一次请求。假设n+5时这个key对应的list有push操作，则依然需要等到n+100时才能获取这个被push进来的值。

BRpop正确的行为是：

- 如果对应的key存在，并且list是有值的，则会直接弹出第一个非空列表的尾元素，行为等同于Rpop。
- 如果key不存在，或者list为空，则服务端会阻塞；当有另外的命令插入这个key时，或者超时后，会解除阻塞状态并返回nil。

下图展示了BRpop函数的调用流程

