# Go缓存流转介绍

## 概括

为了保证我们的大集群在前端访问时高并发、高可用，我们使用了redis作为缓存。但是我们还要获取实时信息，这就需要专门对设备集群进行遍历，并将实时的设备信息写入redis缓存中。Go语言在并发编程中有着极为显著的优势，结合我们自己实现的使用队列方式的设备轮询和处理机制，实现了这种特别的缓存流转机制。除了可以按照设定的时间进行设备状态轮询并将信息写入缓存中，设备上的可操控原件阈值性控制也在这里实现。

我们使用xorm连接mysql数据库，使用Redigo操作redis缓存，并且使用Gin提供了刷新、插入新设备、插入新的区间等功能的tcp请求接口。实现了轮询服务强制更新和热更新等。

其中最值得一提的是我们基于java多线程底层的队列思想和go程信道机制制作的设备轮询机制。首先我们定义了一个储存设备地址的自定义字符串队列，之后我们启动了一个go程，并在go程中的逻辑执行之前进行信道值等待，只有当上一个go程结束后，才会有值传入信道，使得当前在等待信道传值的go程继续执行。go程中逻辑即从队列中取出一个设备地址，之后对设备进行读取、控制之后将设备信息写入redis缓存，最后在将刚刚操作完成的设备放到队列尾部实现环形读取。

Go缓存流转也可可以使用grpc等技术或直接相互请求的形式形成集群。

## 各功能模块

### “刷新状态”：

请求方式：GET

请求路径：/flush/:way

参数类型：路径参数

参数实例：/flush/192.168.100.106:80

说明：刷新指定设备当前信息并且写入redis缓存，返回从设备中读取到的所有数据

### “添加设备”：

请求方式：GET

请求路径：/addAddress

参数类型：address

参数实例：/addAddress?address=192.168.100.106:80

说明：添加设备地址到遍历的队列中

### “添加区间”：

请求方式：GET

请求路径：/addArea

参数类型：address,min,max

参数实例：/addArea?address=192.168.100.106:80&min=10&max=20

说明：添加或更新设备上可控原件的触发区间