

QuickStart

一、可选（建议直接连我电脑上的Etcd集群，不然代码里./etcd/etcd.go需要同步修改）

1. 在个人服务器使用docker部署一个Etcd集群

首先下载Etcd

```
sudo docker pull quay.io/coreos/etcd:v3.5.5
```

然后修改成自己的部署路径和配置路径（以下为参考），配置文件可参考etcd0.yaml、etcd1.yaml、etcd2.yaml（主要修改ip地址）

```
docker run -d -p 2379:2379 -p 2380:2380 -v /tmp/etcd0-data:/etcd-data -v /data/home/shenyntong/workspace/etcdconf:/etcd-conf --name etcd0 quay.io/coreos/etcd:v3.5.5 /usr/local/bin/etcd --config-file=/etcd-conf/etcd0.yaml
docker run -d -p 12379:12379 -p 12380:12380 -v /tmp/etcd1-data:/etcd-data -v /data/home/shenyntong/workspace/etcdconf:/etcd-conf --name etcd1 quay.io/coreos/etcd:v3.5.5 /usr/local/bin/etcd --config-file=/etcd-conf/etcd1.yaml
docker run -d -p 22379:22379 -p 22380:22380 -v /tmp/etcd2-data:/etcd-data -v /data/home/shenyntong/workspace/etcdconf:/etcd-conf --name etcd2 quay.io/coreos/etcd:v3.5.5 /usr/local/bin/etcd --config-file=/etcd-conf/etcd2.yaml
```

进入docker bash中

```
docker exec -it etcd0 bash
```

查看集群状态

```
etcdctl endpoint status --cluster -w table
```

二、必选

1. 下载我的项目

```
git clone https://git.woa.com/shenyntong/MoreFunProJect.git
```

2. 启动kafka组件

```
bash StartMQ.sh
docker ps -a #查看是否启动完成
```

3. 开启一个终端启动NameServer

```
cd MoreFun/
go run ./nameServer/nameServer.go
```

4. 开启一个终端启动服务器

```
bash Start_All_Svr.sh #启动了ABCDEF服务各20个
#当窗口不再输出代表启动完成-----Response: send ns success
#此时NameServer显示-----[120]: receive succss
```

5. 开启一个可以键入的服务器

```
go run ./server/server.go --name A --port 40002 --test=false
#此时NameServer显示-----[121]: receive succss
#此窗口显示:
#Choose a test scenario:
#1. Fixed target routing test
#2. Other types of routing tests
#3. Batch set dynamic key-value routing test
#4. Batch access dynamic key-value routing test
#5. Flexible experience
#Enter your choice:
```

6. 测试完成后关闭服务

```
killall sidecar
killall server
```

三、完成上诉操作后，服务启动完成，开始测试

1. 测试制定目标路由

```
Enter your choice: 1
Enter the type of service: B #输入B类服务的目的是，假如指定目标路由不存在，会返回一个B类服务的所有ip:port列表
Enter the target route: 9.135.119.71:30000 #这里的ip填自己服务器的ip
Enter the number of times to send: 1000 #发送信息的次数
```

2.测试其他路由选择

```
"A": {ServiceName: "A", Strategy: ConsistentHash},           //访
问A是一致性哈希
"B": {ServiceName: "B", Strategy: Random},                   //访
问B是随机
"C": {ServiceName: "C", Strategy: WeightedRoundRobin},       //访
问C是平滑加权轮询
"D": {ServiceName: "D", Strategy: LeastConnections},         //访
问D是最少连接
"E": {ServiceName: "E", Strategy: WeightedLeastConnections}, //访
问E是加权最少连接(未作平滑处理)
"F": {ServiceName: "F", Strategy: FastestResponse},          //访
问D是最快响应
```

```
Enter your choice: 2
Enter the type of service or dynamicKey: A           #访问想要访问的服务类型，测试
对应的路由选择策略
Enter the number of times to send: 1000              #发送信息的次数
```

3.测试批量的动态键值路由注册

由于这里是批量的注册，所有的dynamicValue都是一样的，但是dynamicKey会以 [yourName]_[num]注册，如: tsy_1、tsy_2, tsy_3

```
Enter your choice: 3
Enter the dynamicKey: [key]                             #注册的动态key
前缀
Enter the dynamicValue: 9.135.119.71:30000              #注册的
endpoint路径
Enter the timeout: 100                                  #timeout时间
Enter the number of times to set: 1000                  #批量注册的个数
```

想要不一样的dynamicValue就只能在times处设1独立注册了

4.测试动态键值路由的访问

```
Enter your choice: 4
Enter the type of service or dynamicKey: [key]         #访问想要访问的动态key
Enter the number of times to send: 1000                #发送信息的次数
```

5.灵活测试

这种方式包含以上全部的功能，但是都不具备大批量的效果

| | |
|------------------------------|------------------------------------------|
| A 9.135.119.71:30000 | #直接输入服务类型和目标路由达成1的测试效果 |
| A | #直接输入服务类型达成2的测试效果 |
| set | #输入set进入设定动态键值 |
| [key] 9.135.119.71:30000 100 | #分别输入key,value,timeout达成3的测试效果,且这种注册不带编号 |
| [key] | #输入访问想要访问的动态key达成4的测试效果 |
| exit | #退出灵活测试 |

6.其他测试

基本测试已经完成了，如果想要测试服务掉线后状态维护，两种方法：

1. 直接kill掉对应服务的sidecar和server
2. 独立启动服务按Ctrl+C中断，如执行多条类似 `go run ./server/server.go --name A --port 40002 --test=false`，这种方式的好处是更容易看到消息发给了哪个服务器

```
#go run ./server/server.go --name A --ip 9.135.119.71 --port 30000 --weight 5 --test=false
--name      [服务类型]          #如 A
--ip        [服务ip]            #如 9.135.119.71
--port      [服务端口]          #如 30000
--weight    [服务权重]          #如 5
--test      [是否是测试模式]    #如--test=false, false时服务可以键入, true不能键入
```

独立启动服务更加容易测试带权重的路由选择策略，因为批量启动的权重是随机的（1-30）

感谢您已经看完了全部的QuickStart，祝您工作顺利！！每天MoreFun！！