



《手写OS操作系统》小班二期招生，全程直播授课，大牛带你掌握硬核技术！

点此查看

首页 > 慕课教程 > Go工程师体系课全新版 > 9. grpc的负载均衡策略

- 全部开发者教程
- 第17周 分布式理论基础、分布式事务解决方案
1. 事务和分布式事务

2. 程序出哪些问题会导致数据不一致?

3. CAP和BASE理论

4. 两_三阶段提交

5. tcc分布式事务

6. 基于本地消息表的最终一致性

7. 基于可靠消息的最终一致性- 最常用

8. 最大努力通知

9. mq (message queue) 的使用场景

10. mq技术选型



bobby · 更新于 2022-11-08

◀ 上一节 8.常见的负载均衡策略 10. 为什么需要... 下一节 ▶

1. grpc的负载均衡策略

[文档](#)

2. go使用grpc负载均衡

[grpc-consul-resolver地址](#)

3. 关于serverconfig

[官方文档](#)

4. go的grpc测试

<> 代码块

```
1 package main
2
3 import (
4     "0ldPackageTest/grpc_lb_test/proto"
5     "context"
6     "fmt"
7     "log"
8
9     _ "github.com/mbobakov/grpc-consul-resolver" // It's important
10
11     "google.golang.org/grpc"
12 )
13
14 func main() {
15     conn, err := grpc.Dial(
16         "consul://192.168.1.103:8500/user-srv?wait=14s&tag=srv",
17         grpc.WithInsecure(),
18         grpc.WithDefaultServiceConfig(`{"loadBalancingPolicy": "round_robin"}`),
19     )
20     if err != nil {
21         log.Fatal(err)
22     }
23     defer conn.Close()
24 }
```

意见反馈

收藏教程

标记书签



```
27         rsp, err := userSrvClient.GetUserList(context.Background(), &proto.PageIn
28         Pn:    1,
29         PSize: 2,
30     })
31     if err != nil {
32         panic(err)
33     }
34     for index, data := range rsp.Data{
35         fmt.Println(index, data)
36     }
37 }
38
39 }
```

