链滴

四、 go-kit 微服务的限流实现

作者：450370050

原文链接：https://ld246.com/article/1524901808498

来源网站：链滴

介绍

go-kit提供了限流模块，该模块采用令牌桶算法实现，其实是封装了一下golang自带的golang.org/x time/rate包来实现的。

令牌桶

令牌桶这种控制机制基于令牌桶中是否存在令牌来指示什么时候可以发送流量。令牌桶中的每一个令都代表一个字节。如果令牌桶中存在令牌，则允许发送流量；而如果令牌桶中不存在令牌，则不允许送流量。因此，如果突发门限被合理地配置并且令牌桶中有足够的令牌，那么流量就可以以峰值速率送。

令牌桶算法的基本过程如下：

假如用户配置的平均发送速率为r，则每隔1/r秒一个令牌被加入到桶中；

假设桶最多可以存发b个令牌。如果令牌到达时令牌桶已经满了，那么这个令牌会被丢弃；

当一个n个字节的[数据包]到达时，就从令牌桶中删除n个令牌，并且数据包被发送到网络；

如果令牌桶中少于n个令牌，那么不会删除令牌，并且认为这个数据包在流量限制之外；

两种限流

1、DelayingLimiter【限流延迟访问】

2、ErroringLimiter【限流错误返回】

Middleware

因为endpoint的封装，我们在使用go-kit提供的其它中间件时十分简单。下面就是一个完整的限流延中间件

把已有的endPoint外再包一层endPoint,再从最外层向内一层层调用

```
func NewDelayingLimiter(limit Waiter) endpoint.Middleware {
  return func(next endpoint.Endpoint) endpoint.Endpoint {
    return func(ctx context.Context, request interface{}) (interface{}, error) {
      if err := limit.Wait(ctx); err != nil {
        return nil, err
      }
      return next(ctx, request)
    }
  }
}
```

使用延迟限流

把之前我们的bookListEndPoint进行更改

添加限流处理前的bookInfoEndPoint

```
bookServer := new(BookServer)
bookInfoHandler := grpc_transport.NewServer(
   makeGetBookInfoEndpoint(),
   decodeRequest,
   encodeResponse,
)
bookServer.bookInfoHandler = bookInfoHandler
```

添加限流后的代码

```
bookInfoEndPoint := makeGetBookInfoEndpoint()
//创建限流器 1r/s
limiter := rate.NewLimiter(rate.Every(time.Second * 1), 1)
//通过DelayingLimiter中间件，在bookInfoEndPoint 的外层再包裹一层限流的endPoint
bookInfoEndPoint = ratelimit.NewDelayingLimiter(limiter)(bookInfoEndPoint)

bookInfoHandler := grpc_transport.NewServer(
   bookInfoEndPoint,
   decodeRequest,
   encodeResponse,
)
bookServer.bookInfoHandler = bookInfoHandler
```

测试

```
fmt.Println("请求服务: ", instanceAddr, "当前时间: ", time.Now().Format("2006-01-02 15:04:05.9
"))
```

我们把之前的客户端代码加上输出请求的时间点, 查看下多次请求的情况

```
请求服务:  127.0.0.1:50051 当前时间:  2018-04-28 15:47:20.8
        获取书籍详情
        bookId: 1  =>  bookName: 21天精通php
请求服务:  127.0.0.1:50051 当前时间:  2018-04-28 15:47:20.8
        获取书籍详情
        bookId: 1  =>  bookName: 21天精通php
请求服务:  127.0.0.1:50051 当前时间:  2018-04-28 15:47:21.8
        获取书籍详情
        bookId: 1  =>  bookName: 21天精通php
请求服务:  127.0.0.1:50051 当前时间:  2018-04-28 15:47:22.8
        获取书籍详情
        bookId: 1  =>  bookName: 21天精通php
请求服务:  127.0.0.1:50051 当前时间:  2018-04-28 15:47:23.8
        获取书籍详情
        bookId: 1  =>  bookName: 21天精通php
请求服务:  127.0.0.1:50051 当前时间:  2018-04-28 15:47:24.8
        获取书籍详情
        bookId: 1  =>  bookName: 21天精通php
请求服务:  127.0.0.1:50051 当前时间:  2018-04-28 15:47:25.8
        获取书籍详情
        bookId: 1  =>  bookName: 21天精通php
请求服务:  127.0.0.1:50051 当前时间:  2018-04-28 15:47:26.8
        获取书籍详情
        bookId: 1  =>  bookName: 21天精通php
```

```
Process finished with exit code 0
```

原文链接：四、 go-kit 微服务的限流实现