

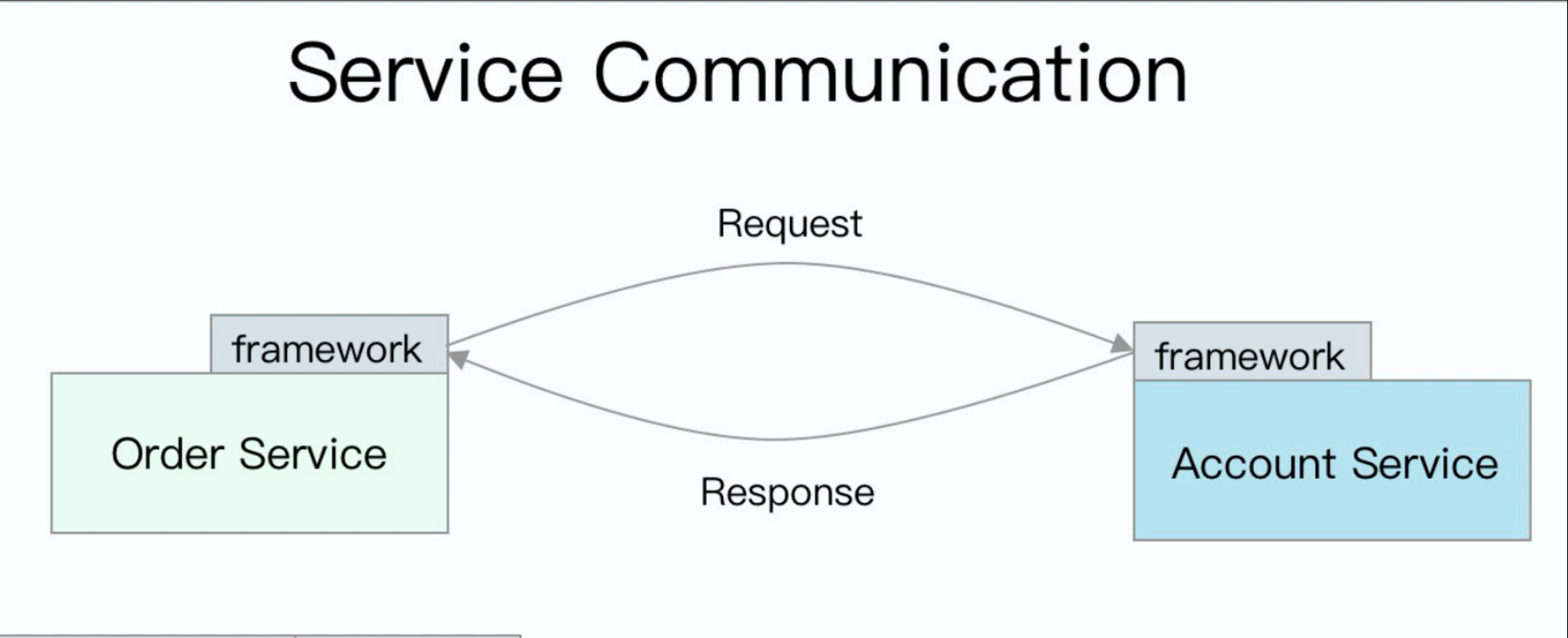
# MOSN 开源说

云原生时代的网络代理

By Xargin

# 为什么需要 Service Mesh

传统服务通信流程



ratelimit	circuit breaker	TLS
routing	load balance	tracing
service discovery	graceful	.....

框架在做什么？

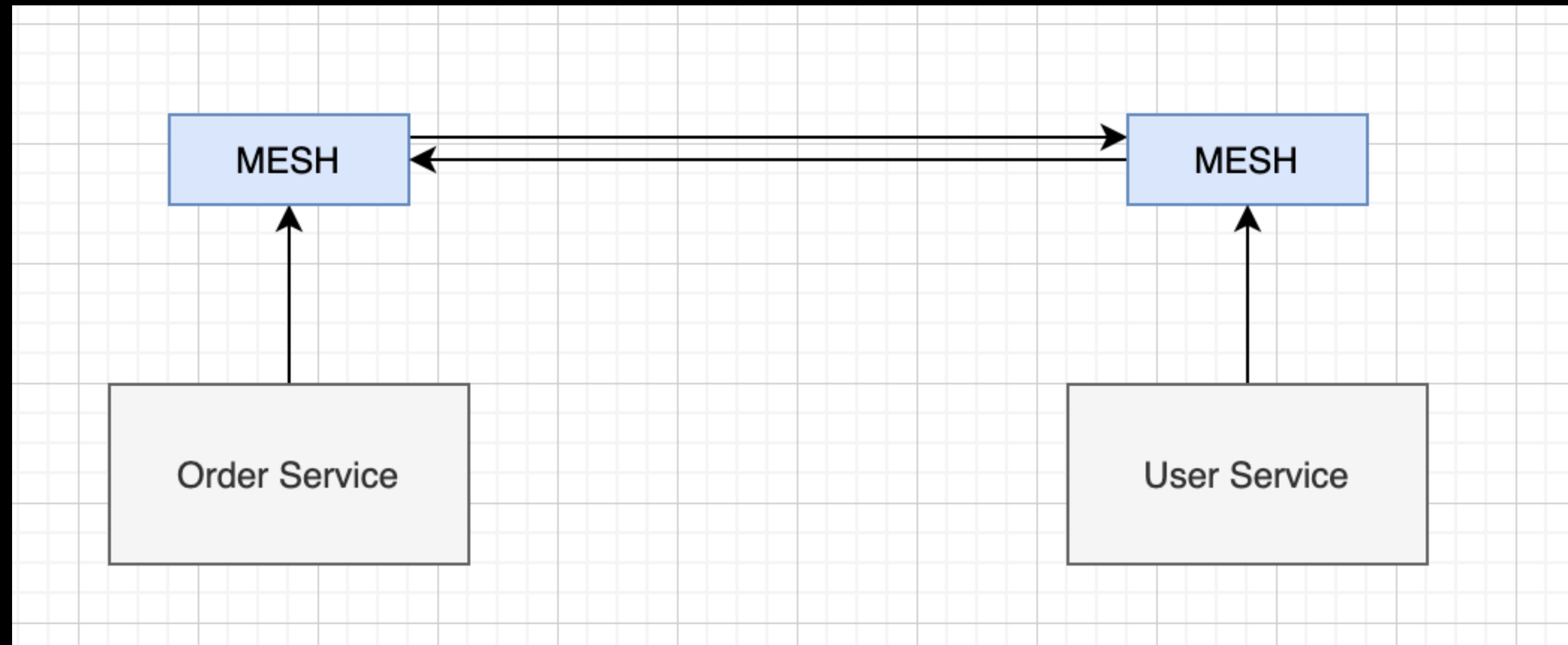
# 为什么需要 Service Mesh

基于 SDK 的集成方式：

- 基础设施团队与业务团队开发 workflow 耦合
- SDK 升级在服务上规模的公司成本很高(以千小时工时计)
- 基础设施新功能试点困难

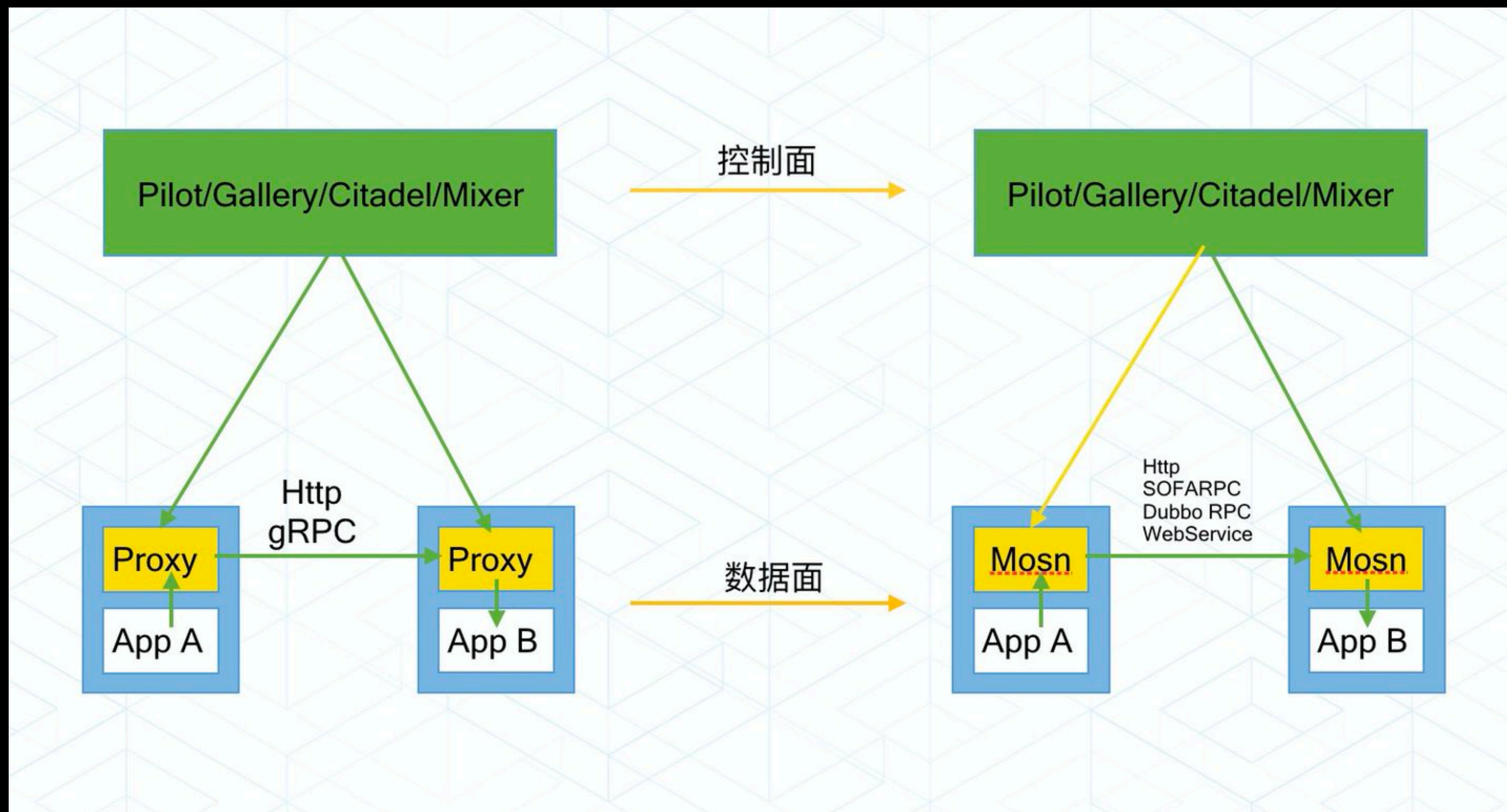
# 为什么需要 Service Mesh

SDK 剥离为单独的进程





# Service Mesh 基本概念



# Service Mesh 基本概念

简单理解：

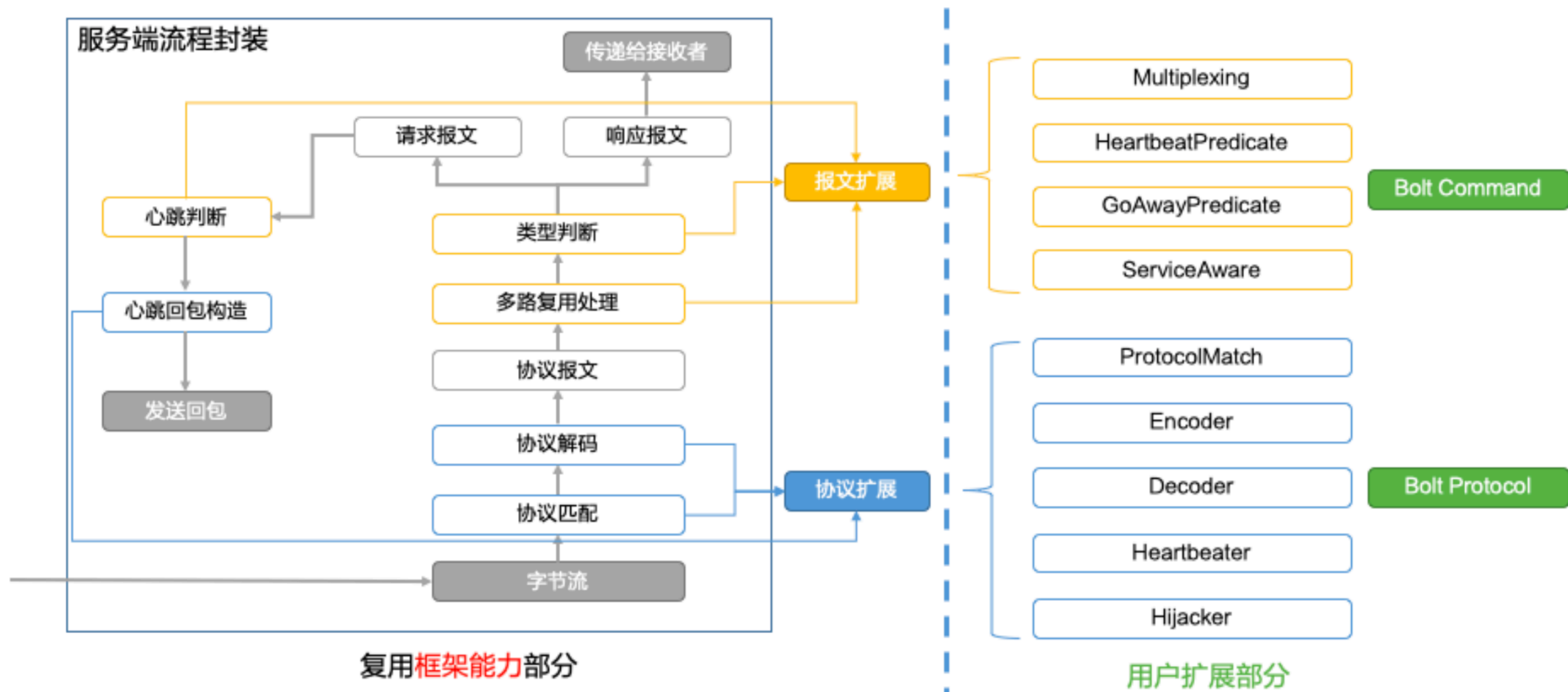
控制面 = 配置管理(服务配置、路由配置、限流配置、证书配置等)

数据面 = 通信管理(连接处理，协议处理，按配置执行 proxy 逻辑)

# MOSN 介绍

- MOSN 就是前面提到的 Service Mesh 的数据面
- 通过 xDS API 对接 Service Mesh, 支持全动态资源配置运行
- 支持 TCP 代理、HTTP 协议、多种 RPC 代理能力
- 支持丰富的路由特性
- 支持可靠后端管理, 负载均衡能力
- 支持网络层、协议层的可观察性
- 支持多种协议基于 TLS 运行, 支持 mTLS 支持丰富的扩展能力, 提供高度自定义扩展能力
- 支持无损平滑升级

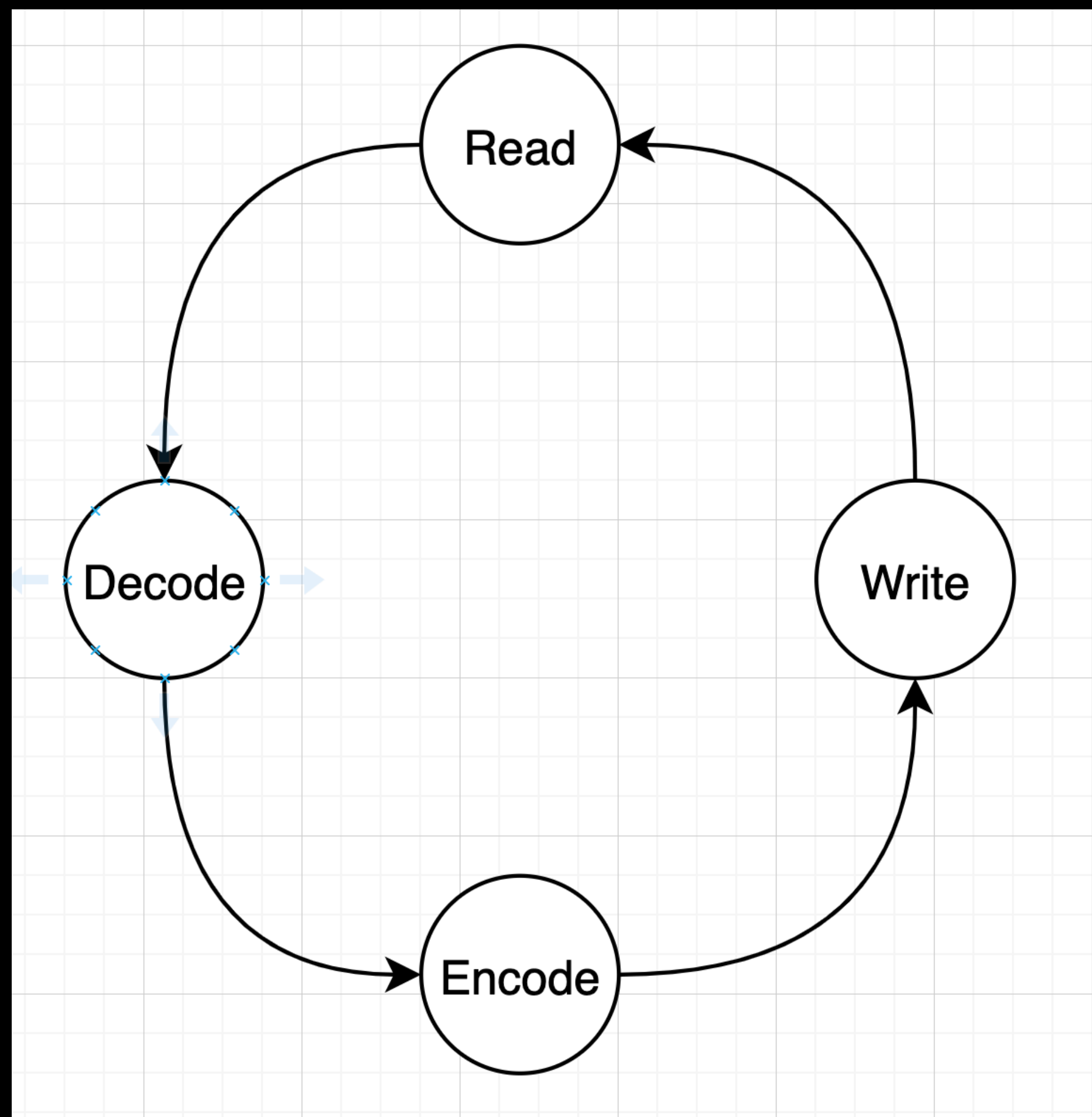
# 多协议支持





# Go 语言开发网络应用

- 所有操作看起来都是同步操作



- 实际上 read/write 是阻塞在了
- gopark 上

```
goroutine profile: total 4
1 @ 0x10324a0 0x102d45a 0x102ca25 0x10bfba5 0x10c0b7b 0x10c0b5d 0x11ede9f 0x105f721
#      0x102ca24      internal/poll.runtime_pollWait+0x54      /usr...
#      0x10bfba4      internal/poll.(*pollDesc).wait+0x44      /usr...
#      0x10c0b7a      internal/poll.(*pollDesc).waitRead+0x22a /usr...
#      0x10c0b5c      internal/poll.(*FD).Read+0x20c          /usr...
#      0x11ede9e      net.(*netFD).Read+0x4e                  /usr...
#      0x1200687      net.(*conn).Read+0x67                   /usr...
#      0x1306ac3      net/http.(*connReader).Read+0xf3        /usr...
#      0x10e32d2      bufio.(*Reader).fill+0x102              /usr...
#      0x10e402c      bufio.(*Reader).ReadSlice+0x3c          /usr...
#      0x10e4263      bufio.(*Reader).ReadLine+0x33           /usr...
#      0x1291e3b      net/textproto.(*Reader).readLineSlice+0x6b /usr...
#      0x13010a1      net/textproto.(*Reader).ReadLine+0x91   /usr...
#      0x13010d0      net/http.readRequest+0xc0               /usr...
#      0x1307dbe      net/http.(*conn).readRequest+0x15e      /usr...
#      0x130c343      net/http.(*conn).serve+0x6d3            /usr...
```

# 依旧令人头疼的 C10K 问题

例如下面这样的代码：

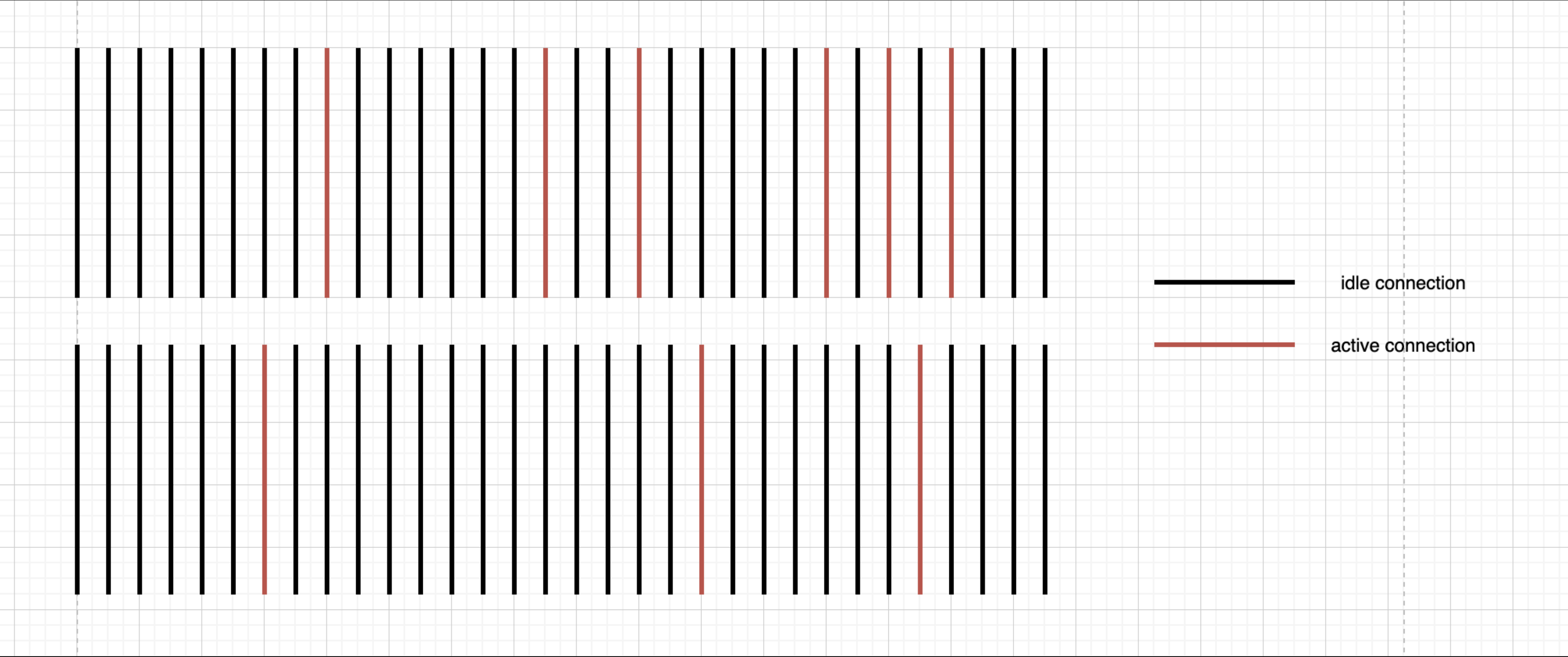
```
5 func main() {  
6     var l net.Listener  
7     for {  
8         c, _ := l.Accept()  
9         go func() {  
10             var buf = make([]byte, 4096)  
11             for {  
12                 c.Read(buf)  
13             }  
14         }()  
15     }  
16 }  
17
```

```
393 // need to recheck error states after setting gpp to WAIT  
394 // this is necessary because runtime_pollUnblock/runtime_pollSetDeadline/de  
395 // do the opposite: store to closing/rd/wd, membarrier, load of rg/wg  
396 if waitio || netpollcheckerr(pd, mode) == 0 {  
397     gopark(&netpollblockcommit, unsafe.Pointer(gpp), waitReasonIOWait, trace  
--- }  
  
// be careful to not lose concurrent READY notification  
old := atomic.Xchguintptr(gpp, new: 0)  
if old > pdWait {  
    throw(s: "runtime: corrupted polldesc")  
}  
return old == pdReady  
}  
  
func netpollunblock(pd *pollDesc, mode int32, ioready bool) *q {  
    netpollblock(pd *pollDesc, mode int32, waitio bool) bool
```

简单的算术题，10w 条连接，至少要占多少内存？

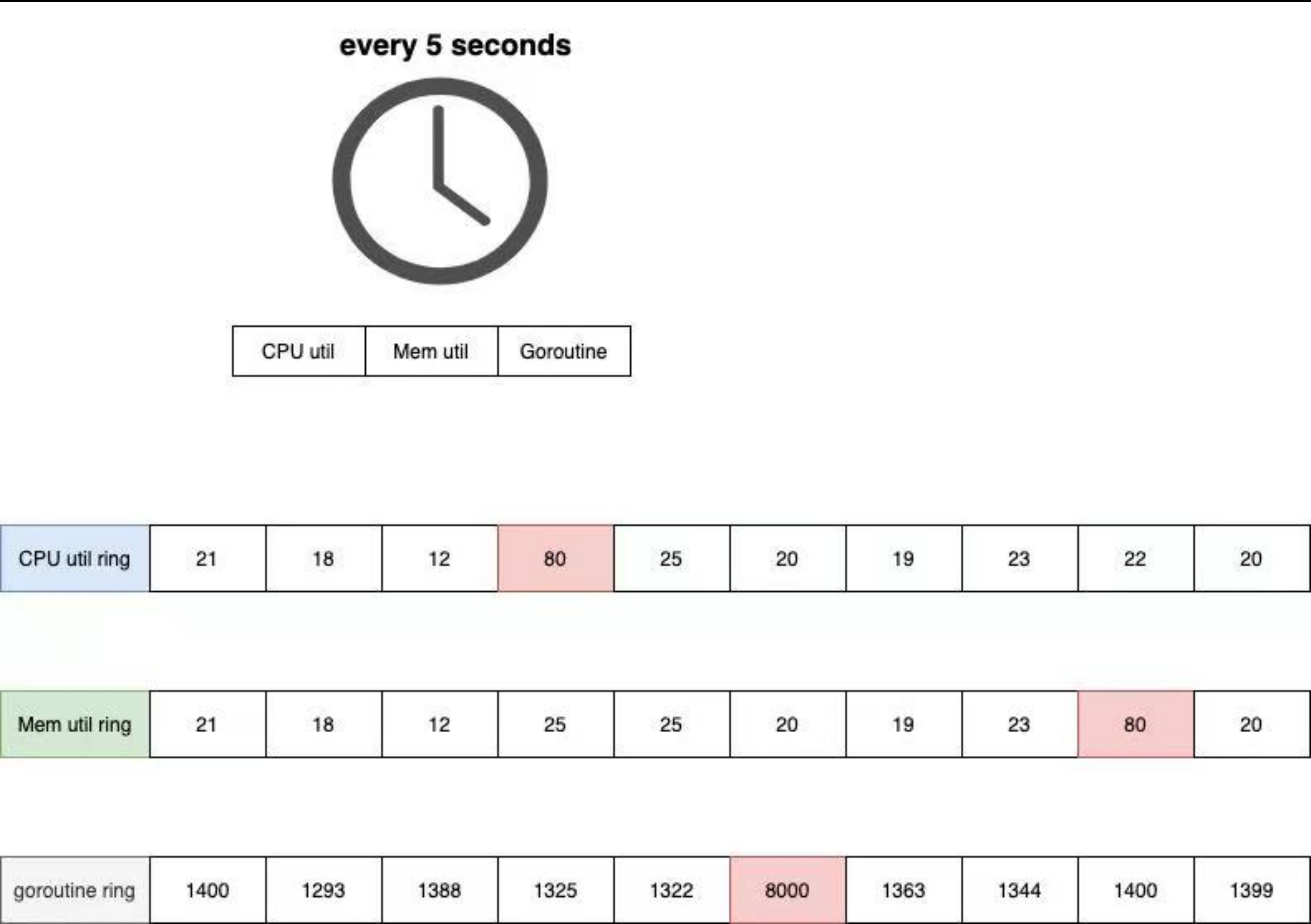
# Mesh 语境下的性能优化

sync.Pool



# 平台型应用的问题定位

## Moving Average

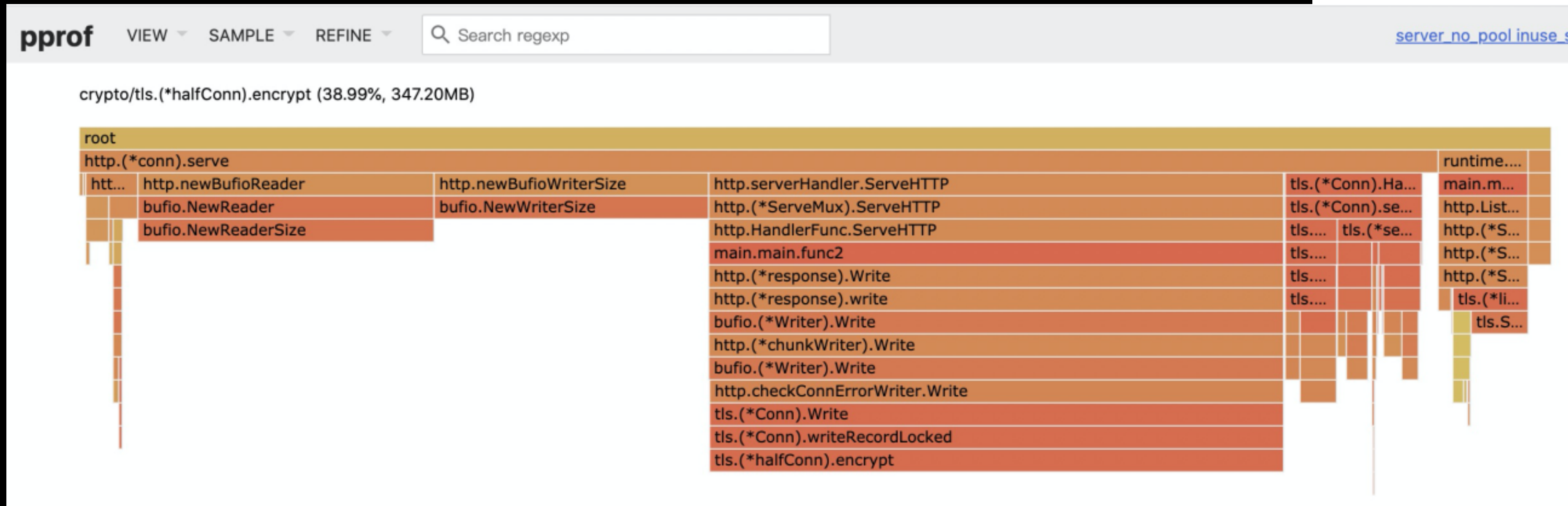
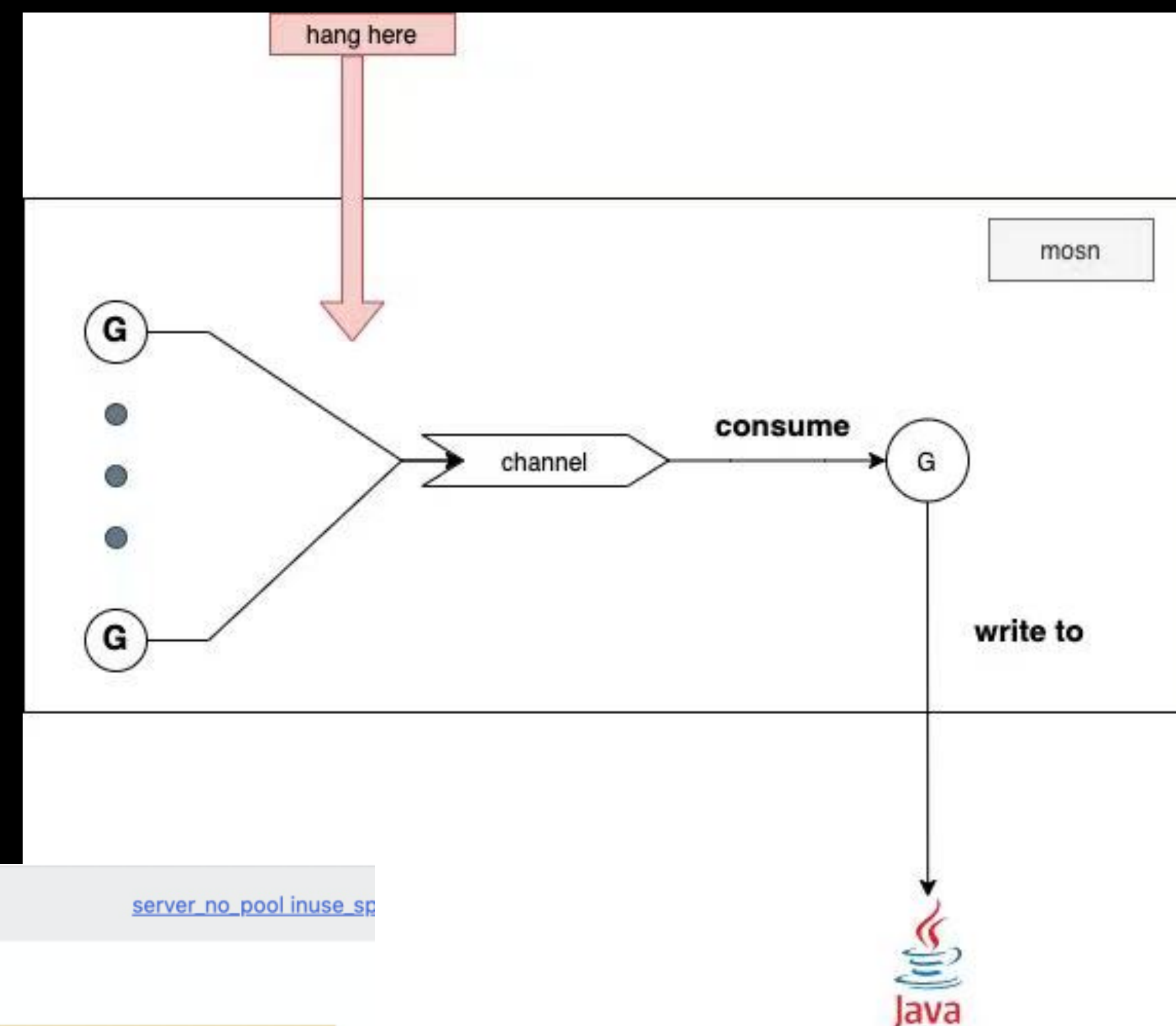


- 为什么 OOM 了？
- 为什么 CPU 监控有个尖刺？
- 为什么 goroutine 暴涨又恢复？
- 为什么实例有 5000 个线程？



# 问题定位

# 业务进程 GC 卡死



# TLS 库占用的内存优化



# MOSN 社区

MOSN 用户群/开发者群

MOSN committers

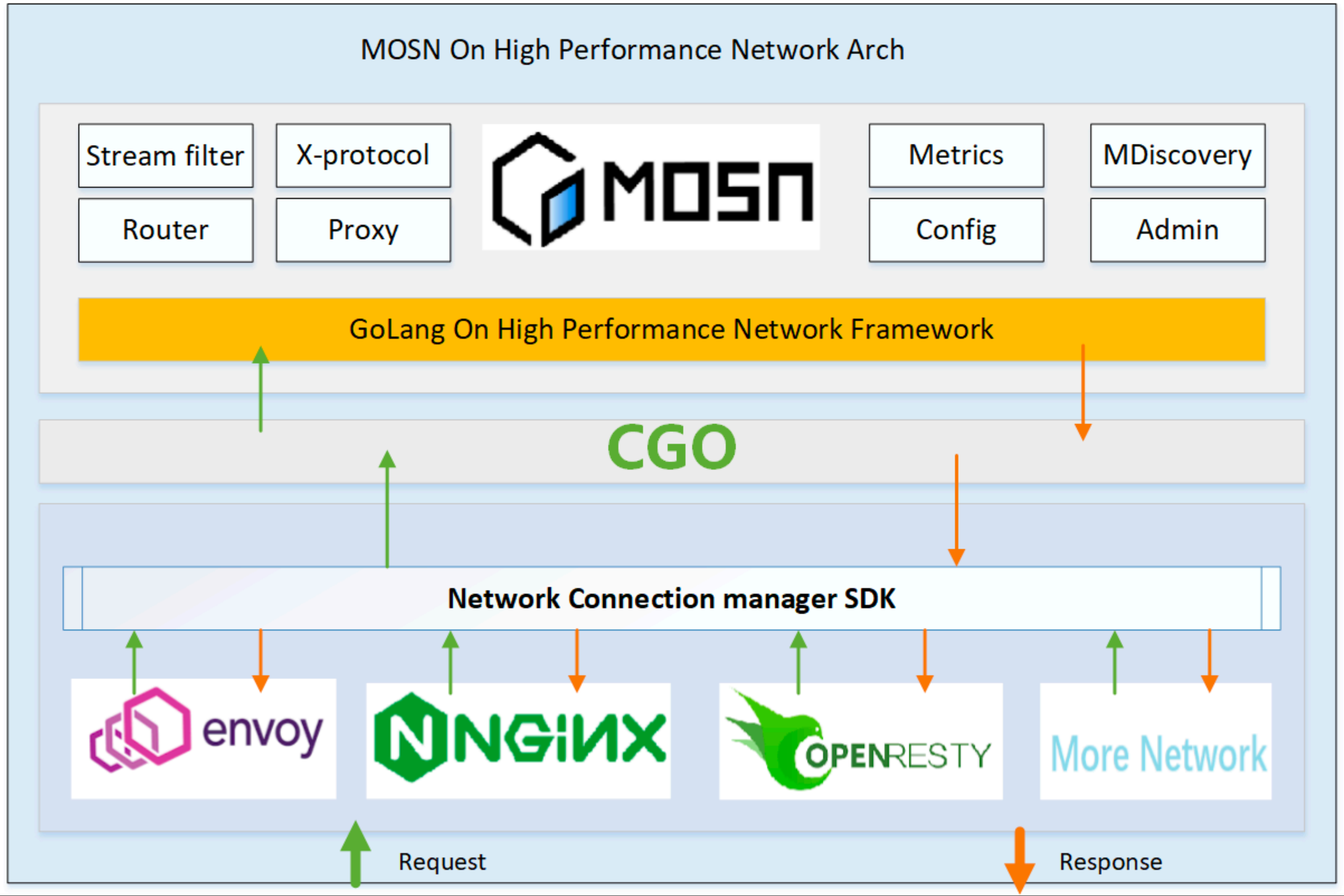
MOSN SIG

MOSN 开发者会议

# MOSN 的后续发展

- 云原生网络平台建设
- 升级Xprotocol框架
- 支持WASM
- 区块链网络框架
- 代码热更新
- 支持网络层可扩展
- fastGRPC
- 协程收敛epoll模型
- cgo性能优化
- 支持zipkin, Jaeger等观测产品
- 支持zk, nacos等注册中心
- 支持Dubbo 3.0新特性
- 支持thrift, kafka等协议
- 支持Istio 1.8
- 支持Ingress和Gateway
- 推动UDPA多协议

# MOSN 后续发展



# 加入钉钉用户群~

