蚂蚁金服ServiceMesh数据平面SOFA MOSN深层揭秘

奕杉

Agenda

蚂蚁金服 ANT FINANCIAL

- 背景
- 构架
- 能力
- 性能
- RoadMap

背景

为什么蚂蚁需要ServiceMesh

蚂蚁金服 ANT FINANCIAL

- 拥抱微服务，云原生
  - SOFA 5规划落地
  - 兼容K8S的智能调度体系
- 运维体系的有力支撑
  - LDC
  - 弹性伸缩
  - 蓝绿/容灾/..
- 金融级网络安全
  - 金融级鉴权体系
  - 云原生zero trust网络安全趋势
- 异构语言体系融合
  - SOFA/NodeJS/C++/Python/..
  - 业务低成本融入服务，运维体系

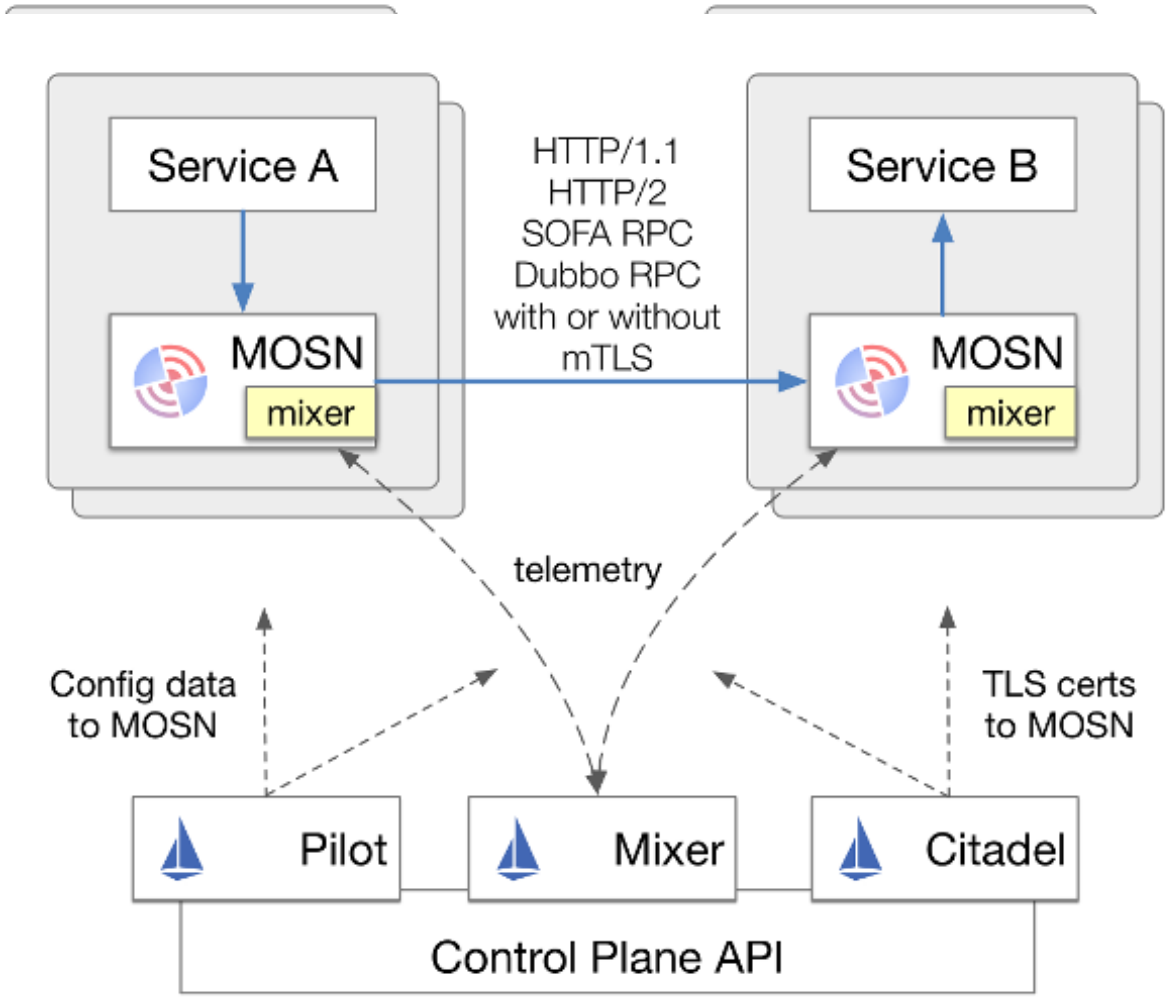
2 为什么要自研Golang版本ServiceMesh

蚂蚁金服 ANT FINANCIAL

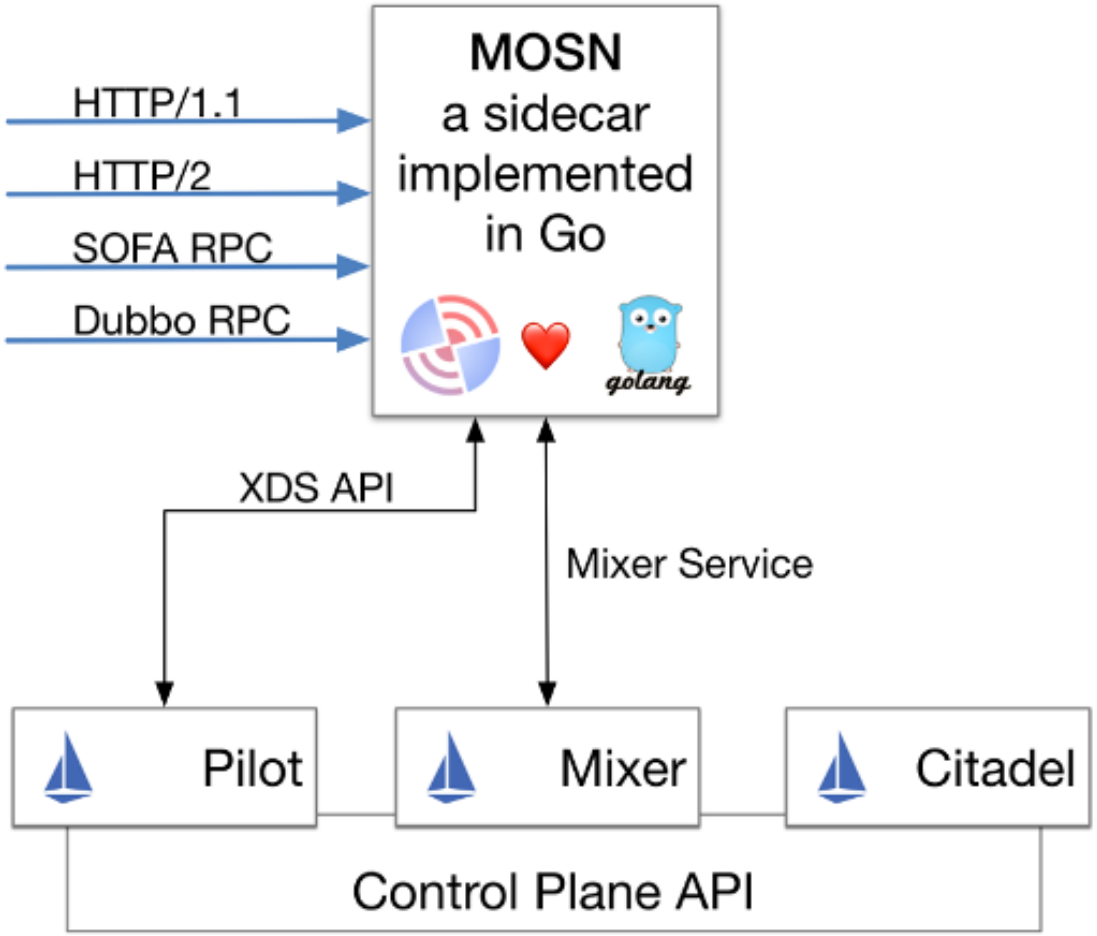
- 跨团队协作需要考虑技术栈落地成本
  - ✓ 参与团队分别使用C，Golang，Java等多种技术栈
- 基于蚂蚁SOFA体系的Mesh化思考
  - ✓ 无法保证上下游应用同时升级到Mesh模式
  - ✓ 基于RPC内容的流量调度
  - ✓ 升级窗口有限，方案必须简单高效
- 运维体系，容器化建设等方面适配
  - ✓ 蚂蚁运维架构建立在流量调度的基础上
  - ✓ 容器管理平台更替快速进行中
- Golang 性能，成本评估符合蚂蚁实际需求

构架

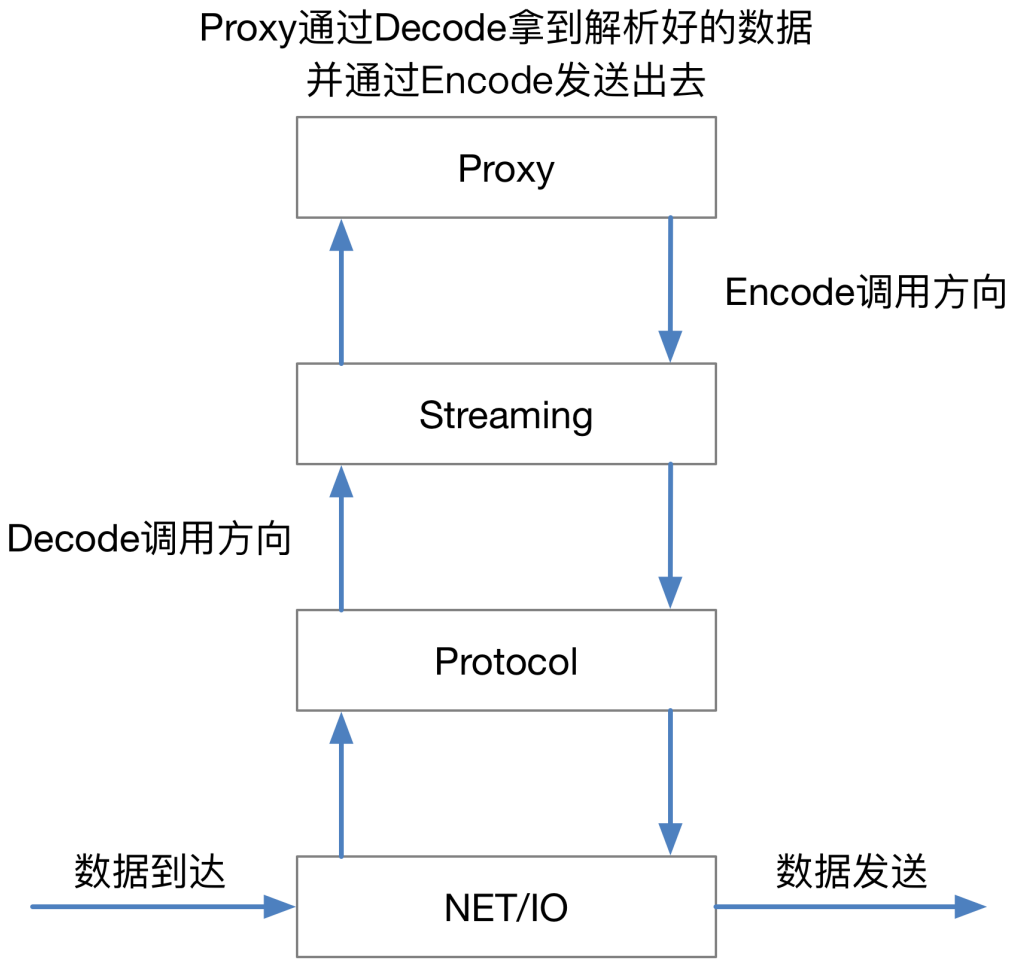
1 SOFAMesh


Service A
MOSN
mixer
HTTP/1.1
HTTP/2
SOFA RPC
Dubbo RPC
with or without
mTLS
Service B
MOSN
mixer
telemetry
Config data
to MOSN
TLS certs
to MOSN
Pilot
Mixer
Citadel
Control Plane API

2 SOFAMOSN


MOSN
a sidecar
implemented
in Go
HTTP/1.1
HTTP/2
SOFA RPC
Dubbo RPC
golang
XDS API
Mixer Service
Pilot
Mixer
Citadel
Control Plane API

3 SOFAMOSN内数据流



4 NET/IO

蚂蚁金服 ANT FINANCIAL

- 屏蔽IO处理细节
- 定义网络链接生命周期，事件机制
- 定义可编程的网络模型，核心方法，监控指标
- 定义可扩展的插件机制

5 PROTOCOL

蚂蚁金服 ANT FINANCIAL

- 定义编解码核心数据结构
  - ✓Mesh处理三段式：Headers + Data + Trailers
- 定义协议Codec核心接口
  - ✓编码：对请求数据进行编码并根据控制指令发送数据
  - ✓解码：对IO数据进行解码并通过扩展机制通知订阅方
- 定义扩展机制通知解码事件

6 STREAMING

蚂蚁金服 ANT FINANCIAL

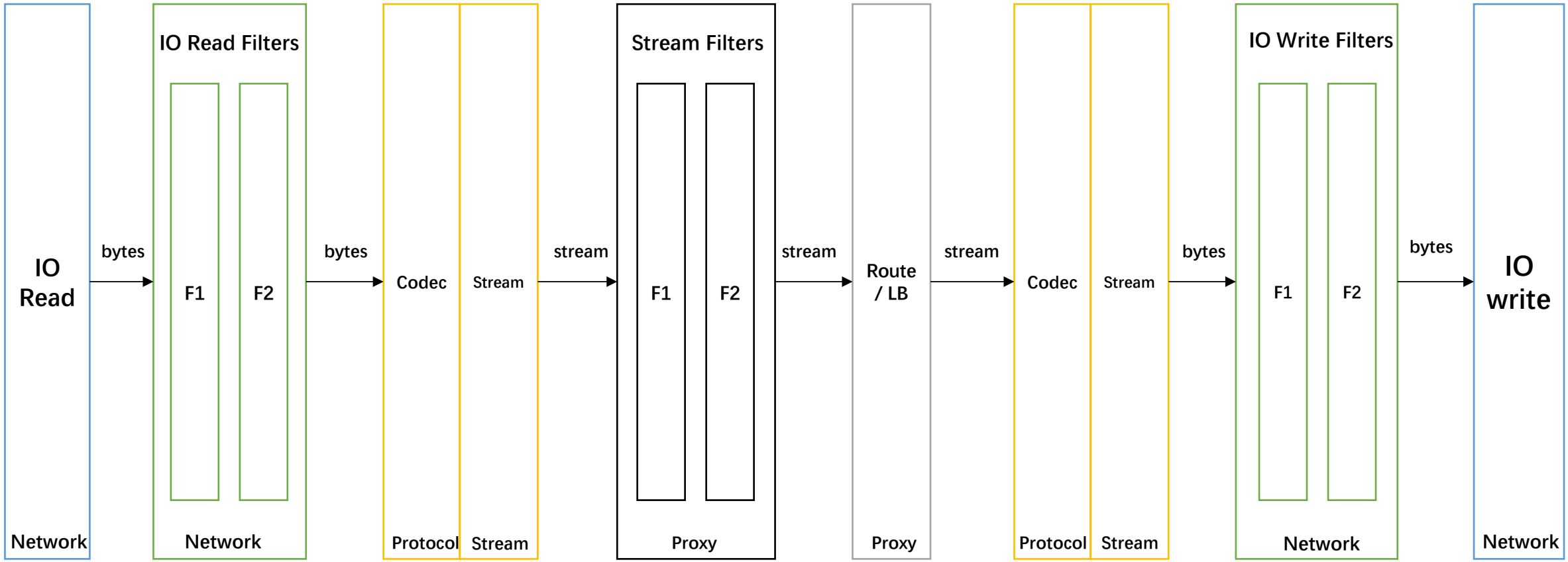
- 定义Stream模型
  - ✓ 向上确保协议行为一致性
  - ✓ 为网络协议请求/响应提供可编程的抽象载体
  - ✓ 考虑PING-PONG，PIPELINE，分帧STREAM三种典型流程特征
- 定义Stream生命周期，核心事件
- 定义Stream层编/解码核心接口
  - ✓ 核心数据结构复用Protocol层
- 定义可扩展的插件机制
- 对于满足请求Stream池化的需求
- 需处理上层传入的状态事件

7 PROXY

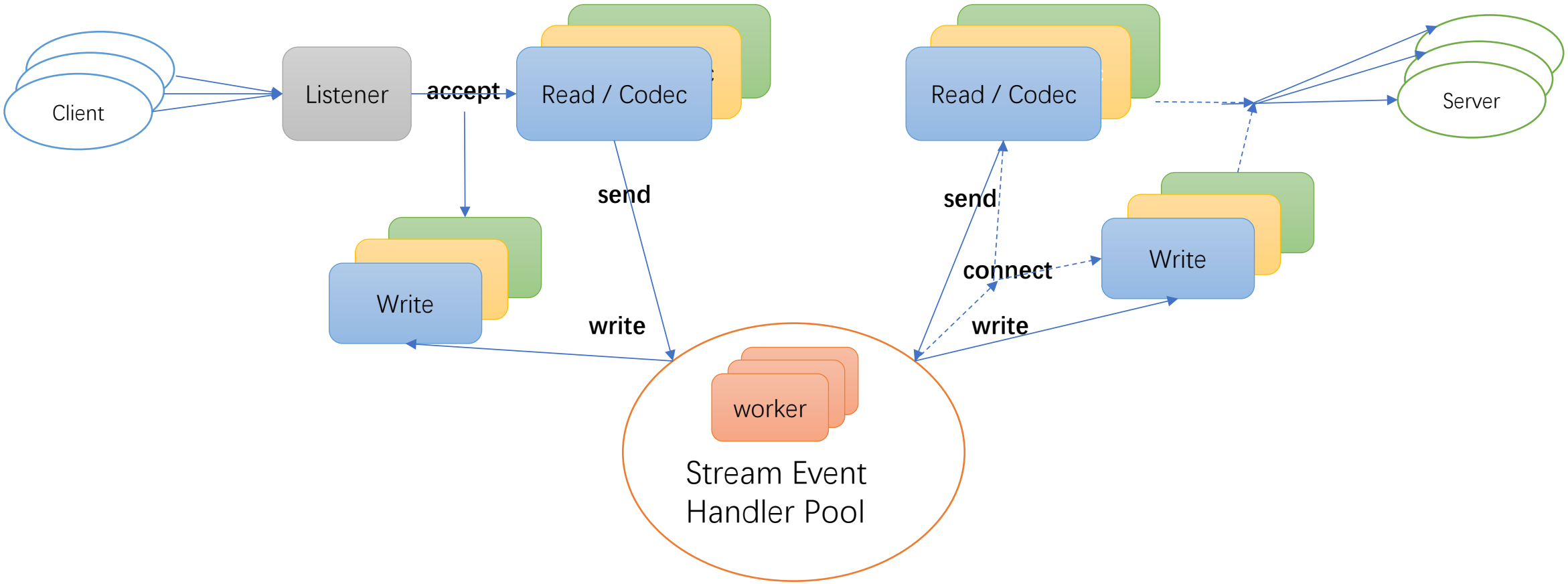
蚂蚁金服 ANT FINANCIAL

- 基于Stream抽象提供多协议转发能力
- 执行Stream扩展Filters
- 提供可扩展的路由寻址能力
- 提供可扩展的后端管理，负载均衡，健康检查能力
- 维护上/下游核心指标

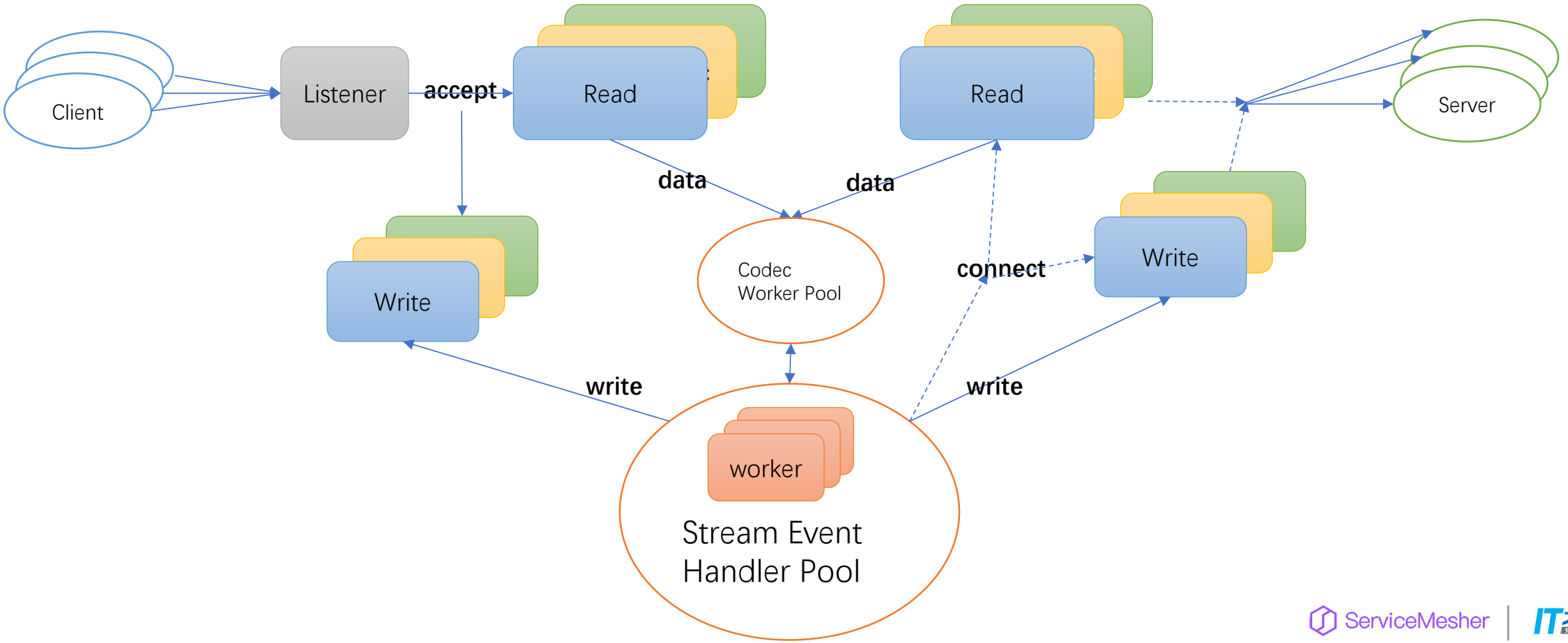
8 转发流程



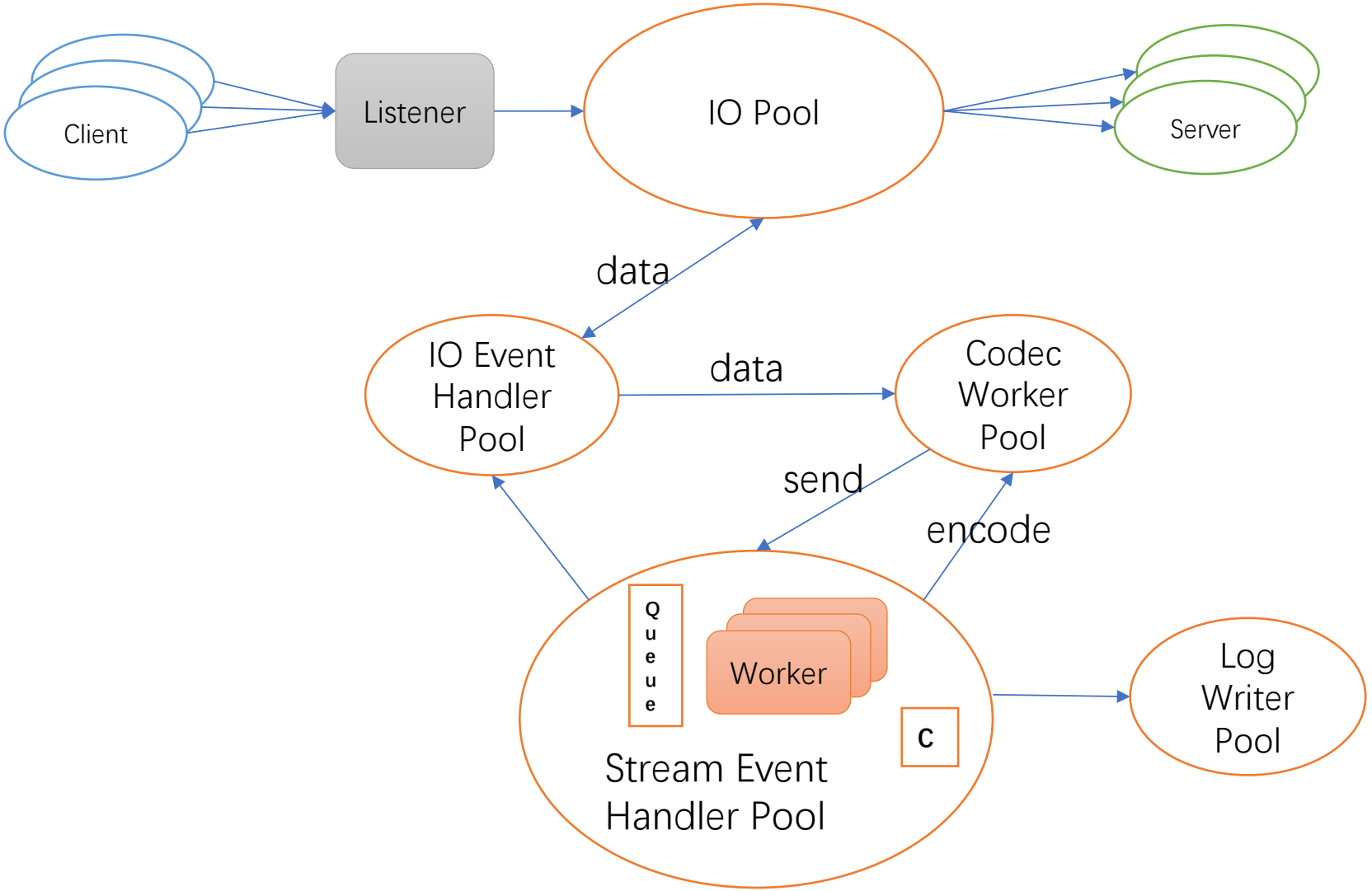
9 线程模型 0.1.0


Client
Listener
accept
Read / Codec
send
Write
write
worker
Stream Event
Handler Pool
Read / Codec
send
connect
Write
write
Server

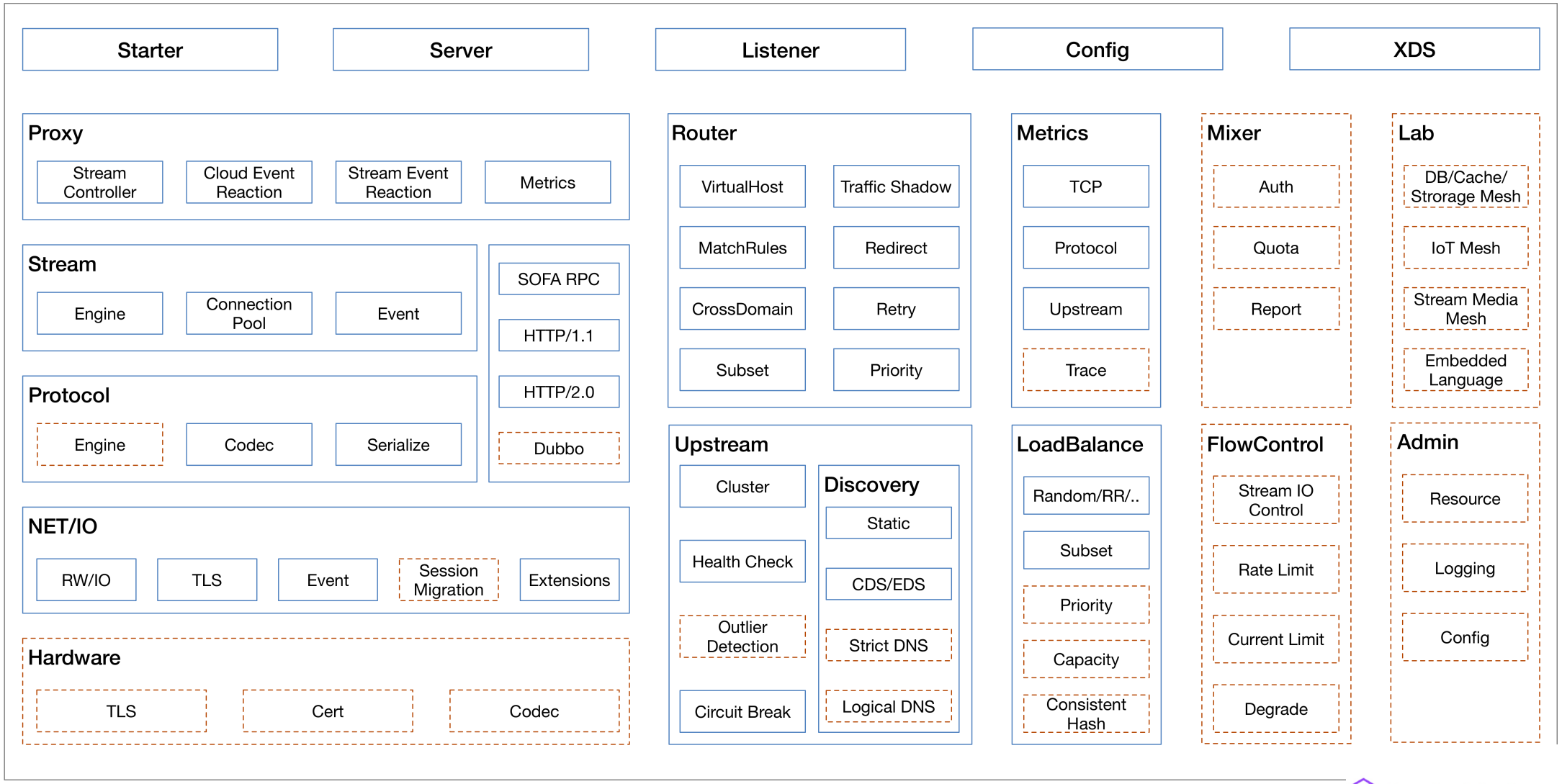
9 线程模型 0.2.0



线程模型规划: SEDA


Client
Listener
IO Pool
Server
data
IO Event
Handler
Pool
data
Codec
Worker
Pool
send
encode
Q
u
e
u
e
Worker
C
Stream Event
Handler Pool
Log
Writer
Pool

模块划分


Starter
Server
Listener
Config
XDS
Proxy
Stream Controller
Cloud Event Reaction
Stream Event Reaction
Metrics
Stream
Engine
Connection Pool
Event
Protocol
Engine
Codec
Serialize
SOFA RPC
HTTP/1.1
HTTP/2.0
Dubbo
NET/IO
RW/IO
TLS
Event
Session Migration
Extensions
Hardware
TLS
Cert
Codec
Router
VirtualHost
Traffic Shadow
MatchRules
Redirect
CrossDomain
Retry
Subset
Priority
Upstream
Cluster
Health Check
Outlier Detection
Circuit Break
Discovery
Static
CDS/EDS
Strict DNS
Logical DNS
Metrics
TCP
Protocol
Upstream
Trace
LoadBalance
Random/RR/..
Subset
Priority
Capacity
Consistent Hash
Mixer
Auth
Quota
Report
FlowControl
Stream IO Control
Rate Limit
Current Limit
Degrade
Lab
DB/Cache/ Strorage Mesh
IoT Mesh
Stream Media Mesh
Embedded Language
Admin
Resource
Logging
Config

要点总结

- 模块化，分层解耦
- 统一的编程模型接口
- 可扩展的事件驱动模型
- 可扩展的路由/后端管理机制
- 更好的吞吐量

能力

1 核心能力

蚂蚁金服 ANT FINANCIAL

网络处理

- 网络编程接口
  - 链接管理
  - 事件机制
  - Metrics 收集
- TCP 代理
- TLS 支持
- TProxy 支持
- 平滑 reload
- 平滑版本升级

多协议

- SOFA RPC
- HTTP 1.x (待优化)
- HTTP 2 (待优化)
- Dubbo (研发中)
- HSF (研发中)
- On TLS

核心路由

- 支持 virtual host 路由
- 支持 headers/url/prefix 路由
- 支持基于 host metadata 的 subset 路由
- 支持重试

后端管理

- 基础负载均衡算法
- 主动健康检查
- Subset 负载策略

2 Highlights

- X-Protocol: 支持 RPC on HTTP2的通用方案（完善中）
- 支持平滑升级中协议无关存量链接迁移
- 支持指定 / 更新 Downstream / Upstream 协议配置
- SOFARPC 支持 Upstream 反向请求

3 Istio集成

- 支持 Istio 0.8 版本 Pilot V4 API
  - ✓支持xDS on ADS
- 支持全动态配置启动运行
- 支持API核心功能点，不断完善中

4 扩展机制

- 网络层扩展
  - 事件订阅
  - 自定义filter
- 协议层处理扩展
  - 事件订阅
  - 自定义filter
- TCP层自定义私有协议
  - 自定义Codec
  - 自定义Stream
- 其他
  - 路由
  - 后端管理

性能

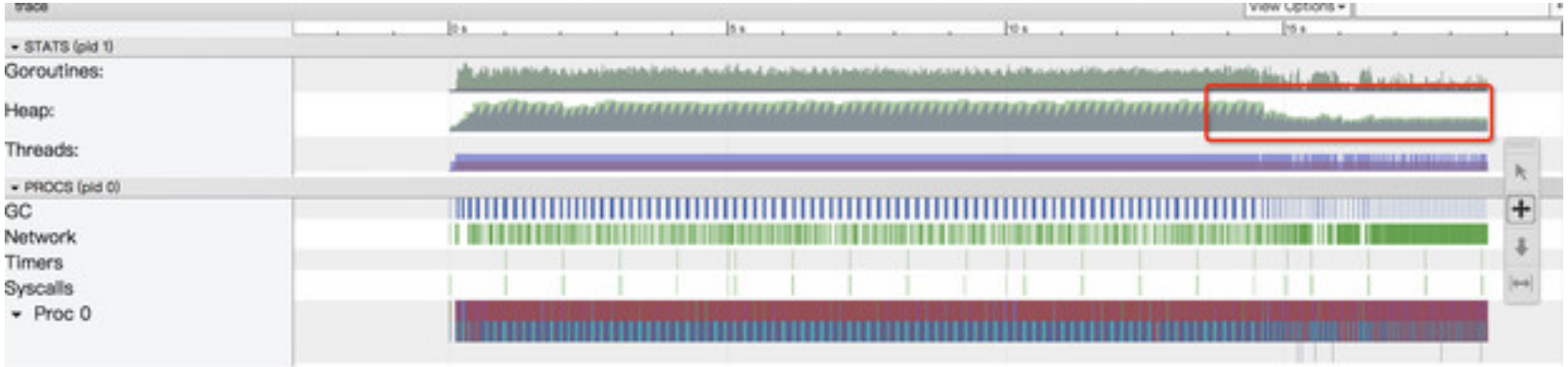
1 MOSN单核优化点

蚂蚁金服 ANT FINANCIAL

- 绑核
  - ✓ 更好的 runtime G-P-M data locality亲和性
  - ✓ 整体性能提升约 30%
- 内存
  - ✓ SLAB-style buffer pool
  - ✓ 内存优化
- IO
  - ✓ IO 优化
  - ✓ IO 均衡
- 调度
  - ✓ P调度均衡
  - ✓ 池化 worker 协程
- 其他
  - ✓ Log 优化
  - ✓ Codec 优化：减少解包等

2 绑核

- 单核绑核 runtime 执行性能更好
  - ✓单次写操作绑核 3us，不绑核 5us
- 更好的 G-P-M cache locality 亲和性
  - ✓Runtime 内存使用率提升，arena 区内存申请频率低，大小更小
  - ✓Mheap 申请系统内存减少约60%

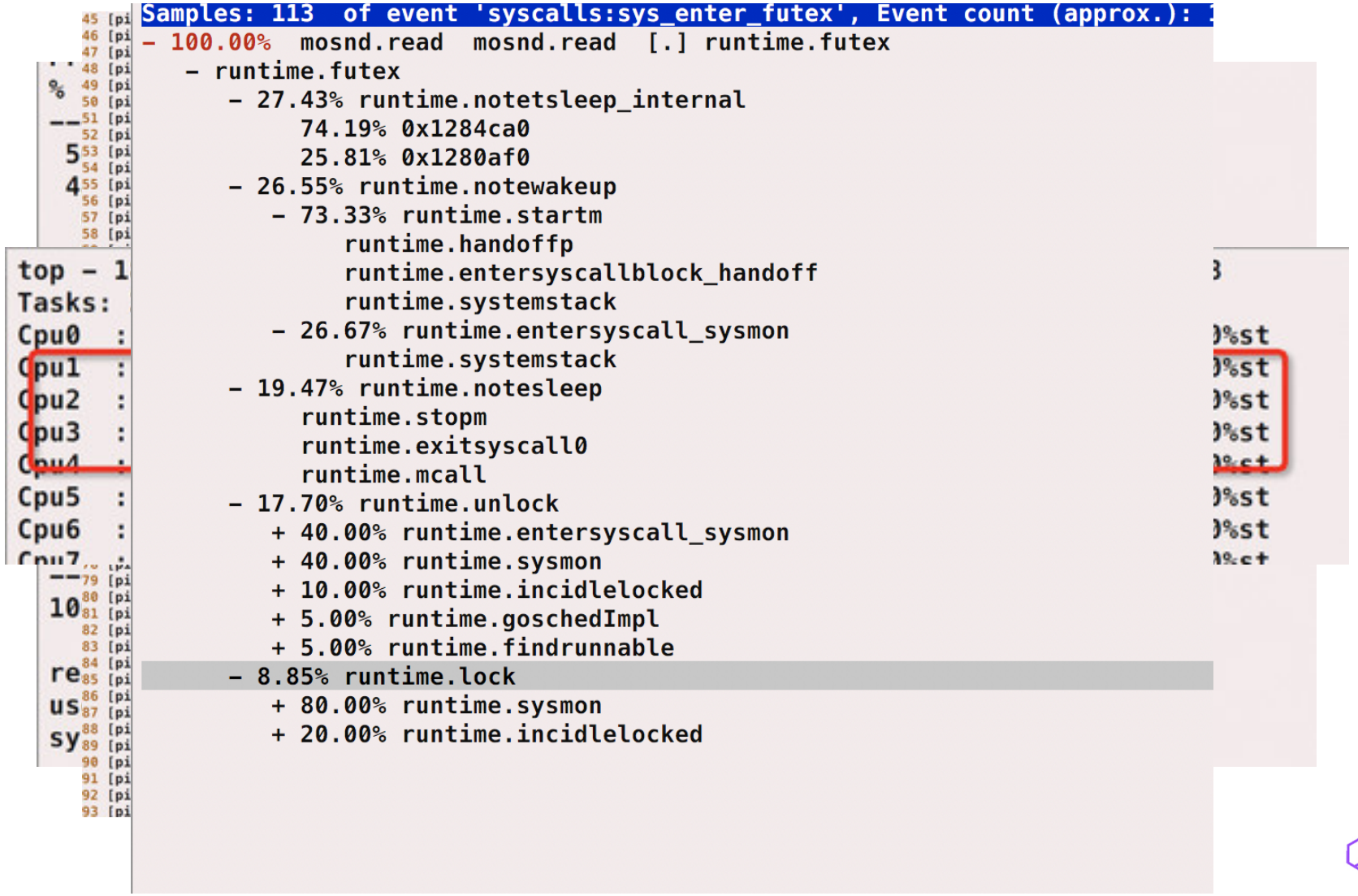


3 内存

- SLAB-style buffer pool
  - ✓减少内存 copy
  - ✓压测场景下内存复用率90%
- Golang 内存模型亲和
  - ✓P中 mcache 缓存小于 32K 的小内存块，最大 2M
  - ✓小内存分配顺序 Pmcache -> mcentral -> mheap -> arena
  - ✓大于 32K 的大内存分配顺序 mheap -> arena
- GC 优化
  - ✓避免入堆
  - ✓减少内存 copy
  - ✓内存使用整体化，降低 scanobject 成本
  - ✓使用 GC 亲和的数据结构
  - ✓适度使用 sync.Pool
  - ✓…

- 优化
  - ✓尽可能多读，同时减少SetReadDeadline频繁调用，实现见IOBuffer.ReadOnce
  - ✓适度 buffer 写数据，频繁写系统 IO 会造成写效率下降
- 均衡
  - ✓读写均衡是高吞吐量的保证
  - ✓大量读/写会增加系统时间消耗， runtime 调度成本

5 IO Bad Case


Samples: 113 of event 'syscalls:sys_enter_futex', Event count (approx.): 1
- 100.00% mosnd.read mosnd.read [.] runtime.futex
- runtime.futex
- 27.43% runtime.notetsleep_internal
74.19% 0x1284ca0
25.81% 0x1280af0
- 26.55% runtime.notewakeup
- 73.33% runtime.startm
runtime.handoffp
runtime.entersyscallblock_handoff
runtime.systemstack
- 26.67% runtime.entersyscall_sysmon
runtime.systemstack
- 19.47% runtime.notesleep
runtime.stopm
runtime.exitsyscall0
runtime.mcall
- 17.70% runtime.unlock
+ 40.00% runtime.entersyscall_sysmon
+ 40.00% runtime.sysmon
+ 10.00% runtime.incidlelocked
+ 5.00% runtime.goschedImpl
+ 5.00% runtime.findrunnable
- 8.85% runtime.lock
+ 80.00% runtime.sysmon
+ 20.00% runtime.incidlelocked

6 调度均衡

- 池化：避免 runtime.morestack 连续栈扩容性能损耗
- 单核：避免G饥饿
- 多核：避免P饥饿

```
G284: status=1(select) m=-1 locked=-1
G286: status=4(select) m=-1 locked=-1
G287: status=4(select) m=-1 locked=-1
G289: status=4(semacquire) m=-1 locked=-1
G290: status=4(select) m=-1 locked=-1
G291: status=4(select) m=-1 locked=-1
```

单核TCP转发

- 环境
  - ✓独占CPU容器
  - ✓CPU： Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz X 1
  - ✓OS： 3.10.0-327.ali2008.alios7.x86_64
- 软件
  - ✓MOSN 0.1.0
  - ✓Envoy 1.7
- 场景
  - ✓代理模式: client -> mesh -> server
  - ✓Client直连server请求耗时1.6ms

8 SOFARPC + 1K字符串

蚂蚁金服 ANT FINANCIAL

指标\软件	MOSN	Envoy
QPS峰值	103500	104000
RT(avg)	16.23ms	15.88ms
MEM	31m	18m
CPU	100%	100%

9 HTTP/1.1 + 1K字符串

蚂蚁金服 ANT FINANCIAL

指标\软件	MOSN	Envoy
QPS峰值	29670	38800
RT(avg)	5.715ms	5.068ms
RT(P99)	16ms	7ms
RT(P98)	13ms	7ms
RT(P95)	11ms	6ms
MEM	56m	24m
CPU	100%	95%

单核SOFARPC转发

- 环境
  - ✓独占CPU容器
  - ✓CPU: Intel(R) Xeon(R) CPU E5620 @ 2.40GHz X 16 X 1
  - ✓OS: 2.6.32-431.17.1.el6.FASTSOCKET
- 软件
  - ✓MOSN 0.1.0
- 场景
  - ✓代理模式: client -> mesh -> server
  - ✓Client直连server请求耗时1.6ms

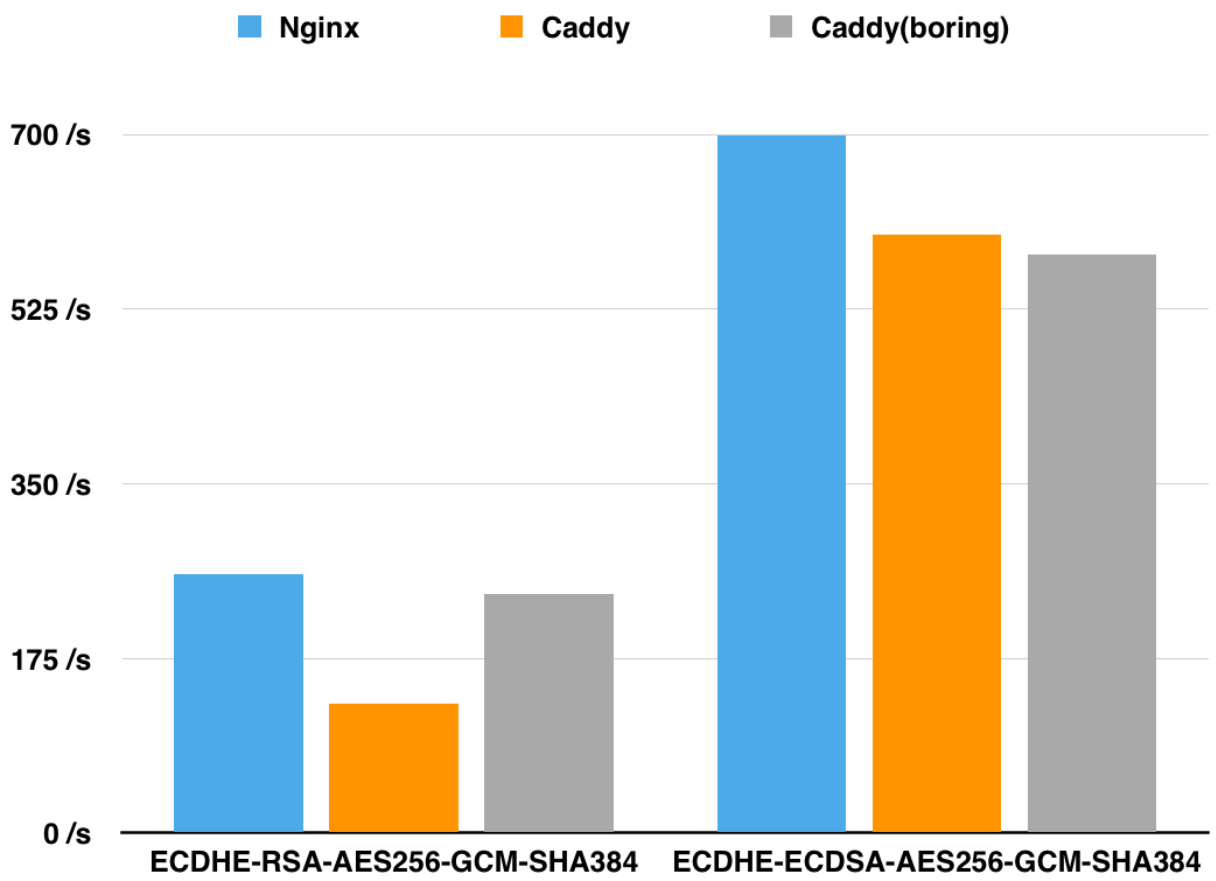
单核SOFARPC转发

指标\软件	MOSN	Envoy
QPS峰值	18000	N/A
RT(avg)	12.354ms	N/A
MEM	100m	N/A
CPU	100%	N/A

Golang TLS单核性能测试

- 环境
  - ✓ CPU： Intel(R) Xeon(R) CPU E5-2430 0 @ 2.20GHz
  - ✓ 内存： 1.5G
- 软件
  - ✓ Nginx-1.13.8 with OpenSSL
  - ✓ Caddy on go-1.10.2
  - ✓ Caddy-boring on go-1.10.2
- 场景
  - ✓ Case1
    - ✓ 证书：RSA 2048
    - ✓ Cipher：ECDHE-RSA-AES256-GCM-SHA384
  - ✓ Case2
    - ✓ 证书：ECC p256
    - ✓ Cipher： ECDHE-ECDSA-AES256-GCM-SHA384
- 命令
  - ✓ ab -f TLS1.2 -Z $cipher -c 100 -n 200000 https://$ip

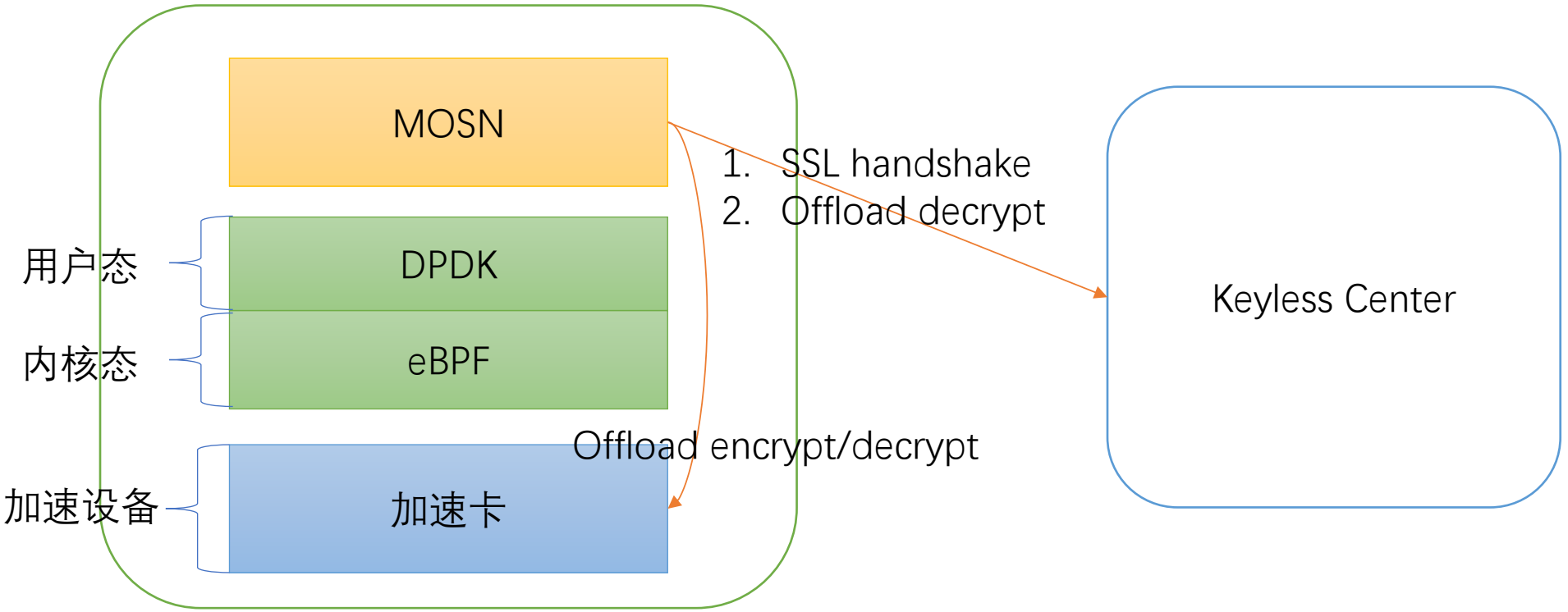
Golang TLS单核性能测试



Golang 单核加解密性能分析

- Golang 对 RSA 上没有优化，并且暂无优化计划
- Golang 对 p256 有汇编优化， p256MulInternal， p256SqrInternal等椭圆曲线函数实现与OpenSSL相同
- Golang 对 p384 没有优化，boring SSL golang 性能是 golang 实现6倍
- Golang 对 AES-GCM 有汇编优化，性能是 boring SSL golang 版本的20倍
- Golang 对 SHA, MD 等 HASH 算法都有汇编优化
- 除 p384，RSA 外，其他算法 golang 原生性能好于boring SSL golang

性能优化规划


MOSN
DPDK
eBPF
加速卡
用户态
内核态
加速设备
1. SSL handshake
2. Offload decrypt
Offload encrypt/decrypt
Keyless Center

RoadMap

SOFA Mesh Roadmap

by Ant Financial Open Source

v0.1.0

- SOFA MOSN实现Proxy的基本功能
- SOFA MOSN对其Envoy的基本功能
- SOFA MOSN支持Envoy的XDS API V0.4版本，实现和Pilot的对接
- 支持通讯协议：HTTP/1.1，REST，SOFARPC

v0.3.0

- 重点是SOFA MOSN对接Mixer
- SOFA MOSN提供熔断和限流功能
- 支持UDP通讯，支持QUIC协议

v0.2.0

- 完善SOFA MOSN的功能
- 提供k8s operator
- HTTP/2增强，包括性能和功能
- 支持xprotocol通用通讯协议，以提供高性能，并可以更好的扩展支持私有TCP
- 支持Dubbo/HSF通讯协议和兼容Dubbo基于Zookeeper的服务注册中心
- 性能优化，包括多核优化/核心链路内存优化
- 代码重构和优化，完善测试案例
- 文档补充

v0.4.0

- 重点是实现和Istio Citadel的对接，实现加密，认证/授权/鉴权等功能
- 支持eBPF和Cilium

蚂蚁金服 ANT FINANCIAL

Q&A

欢迎加入

蚂蚁金服 ANT FINANCIAL

- 系统部
  - 容器，K8S，智能调度，网络，Linux内核..
- 中间件
  - 微服务，容器框架，数据，通信，搜索，OLAP..