GIAC

蚂蚁金服ServiceMesh新型网络代理的思考与实践

奕杉
xiaodong.dxd@antfin.com

蚂蚁金服 ANT FINANCIAL SOFASTACK

Agenda

- 背景 & 概览
- 架构 & 重点特性
- 技术案例解析
- 总结 & 展望
- QA

背景 & 概览

ServiceMesh数据平面

- C实现，支持多语言扩展
- 基于Nginx扩展
- 开发不活跃
- 老牌代理系统，业界广泛使用，服务各类场景

- C++实现
- CNCF项目， ISTIO原生数据平面
- 开发活跃，最新版为1.8.0
- Google, Lyft主导，业界众多公司使用中，重点搭载ISTIO使用，服务各类场景

- Rust实现
- CNCF项目，最早的Service Mesh数据平面
- 开发活跃，最新版为18.9.1

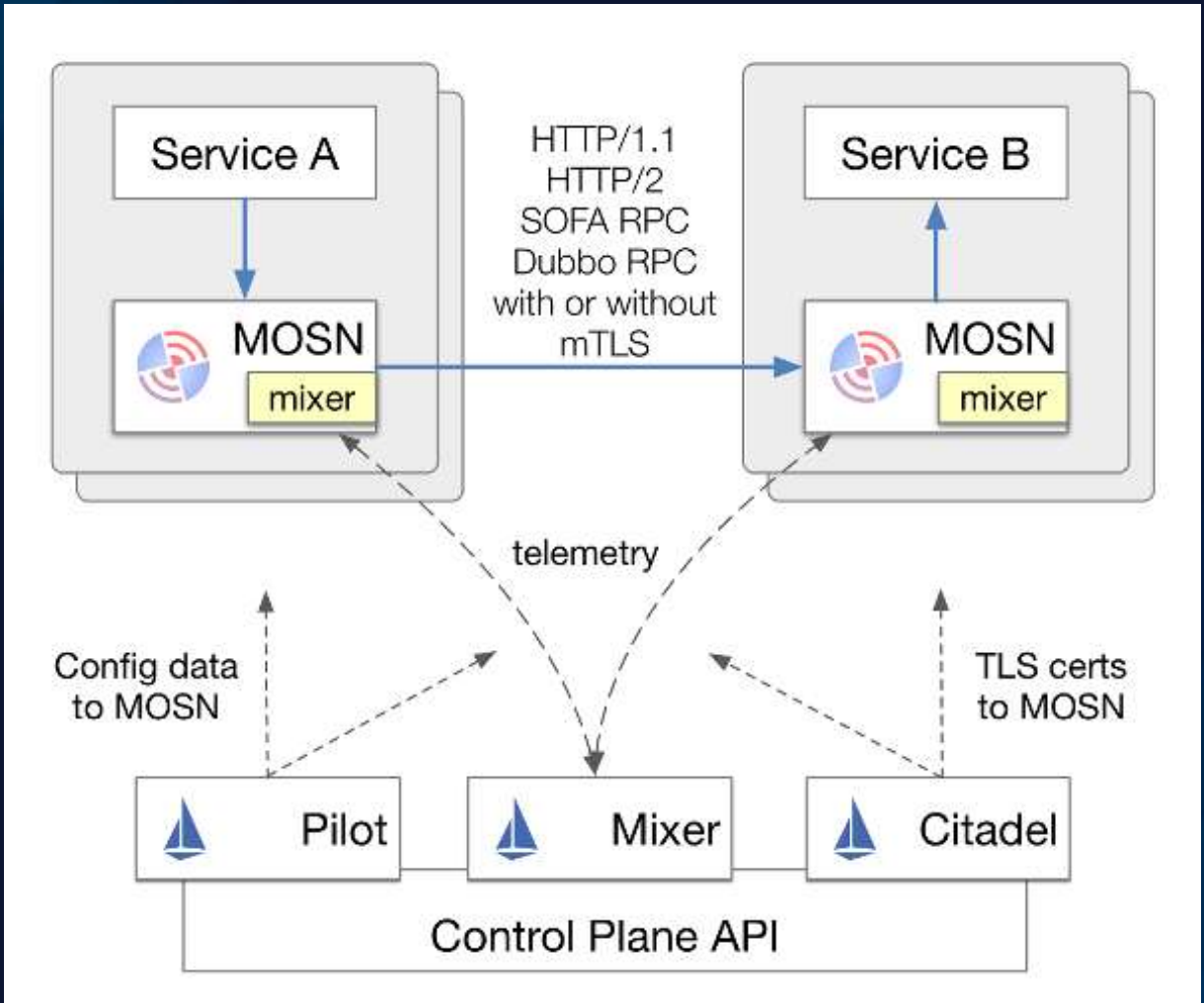
- Golang实现
- 新生项目，初期旨在搭建RPC亲和，高度可扩展性的Golang转发系统
- 开发活跃，最新版为0.3.0
- 蚂蚁+UC主导，重点搭载SOFAMesh使用，目标服务通用场景，金融场景

云原生时代数据转发平面思考

- 部署运维
- 云环境
- 微服务体系
- 云原生定义
- 测试、灰度
- 金融级零可信安全
- 异构语言、协议
- 异构计算
- 运维架构
- 东西向 VS 南北向
- …

架构 & 重点特性

SOFAMesh

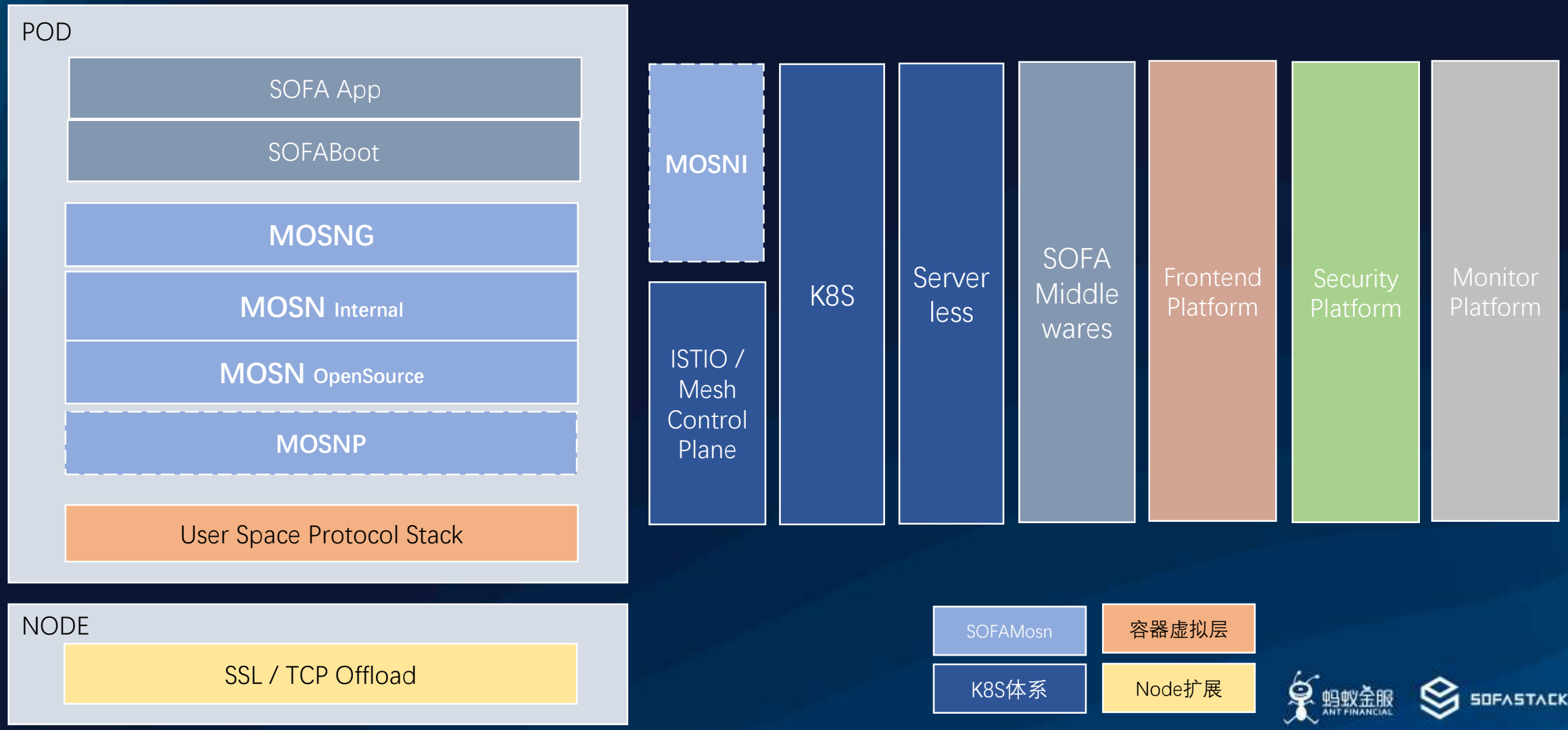


- ✓ Pilot
  - ✓ Transparent proxy integrated with K8S
  - ✓ Traffic Management
  - ✓ Traffic Shifting
  - ✓ Failure Injection
  - ✓ Control Ingress Traffic
  - ✓ Control Egress Traffic
  - ✓ MTLS over HTTPS
- ✓ Mixer
  - ✓ Report Request
- ✓ Citadel
  - ✓ MTLS
  - ✓ RBAC (doing)

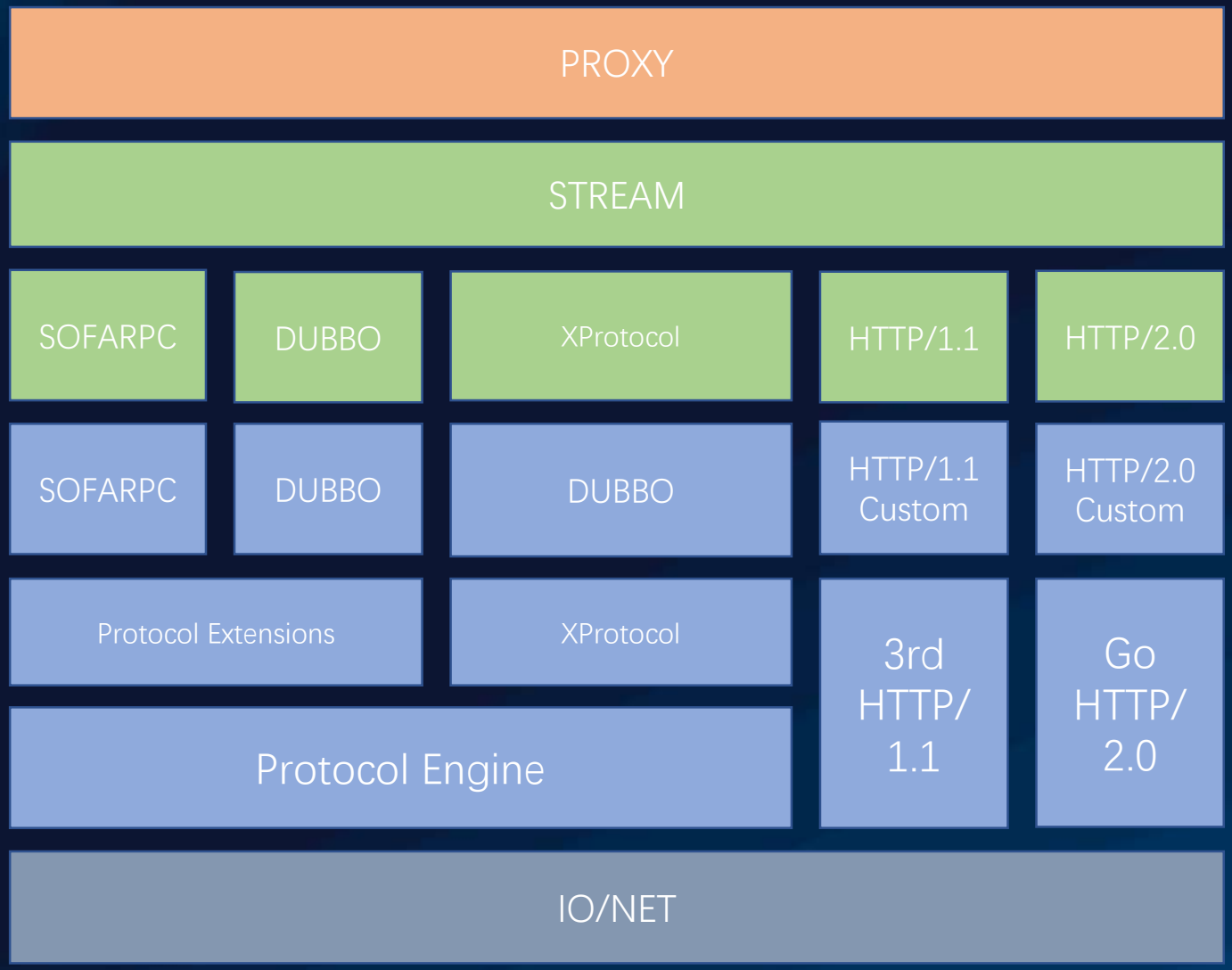
https://github.com/alipay/sofa-mosn
https://github.com/alipay/sofa-mesh

蚂蚁金服 ANT FINANCIAL SOFASTACK

蚂蚁云原生数据平面架构



MOSN – 多协议支持



- 支持常用协议
- 支持自定义扩展模式
- XProtocol快速接入SOFAMesh

蚂蚁金服 ANT FINANCIAL SOFASTACK

MOSN – 性能

IO

- Raw Epoll 模式
- Writev
- 读优化
- 无损迁移

协议

- SofaRPC 深度优化
- XProtocol 扩展机制
- HTTP/1.1, HTTP/2.0 优化

TLS

- 官方库IO 优化
- 无损状态迁移

内存优化

- 内存复用框架
- Slab style buffer
- 内存使用深度优化

协程

- 协程池化
- 同步操作异步化

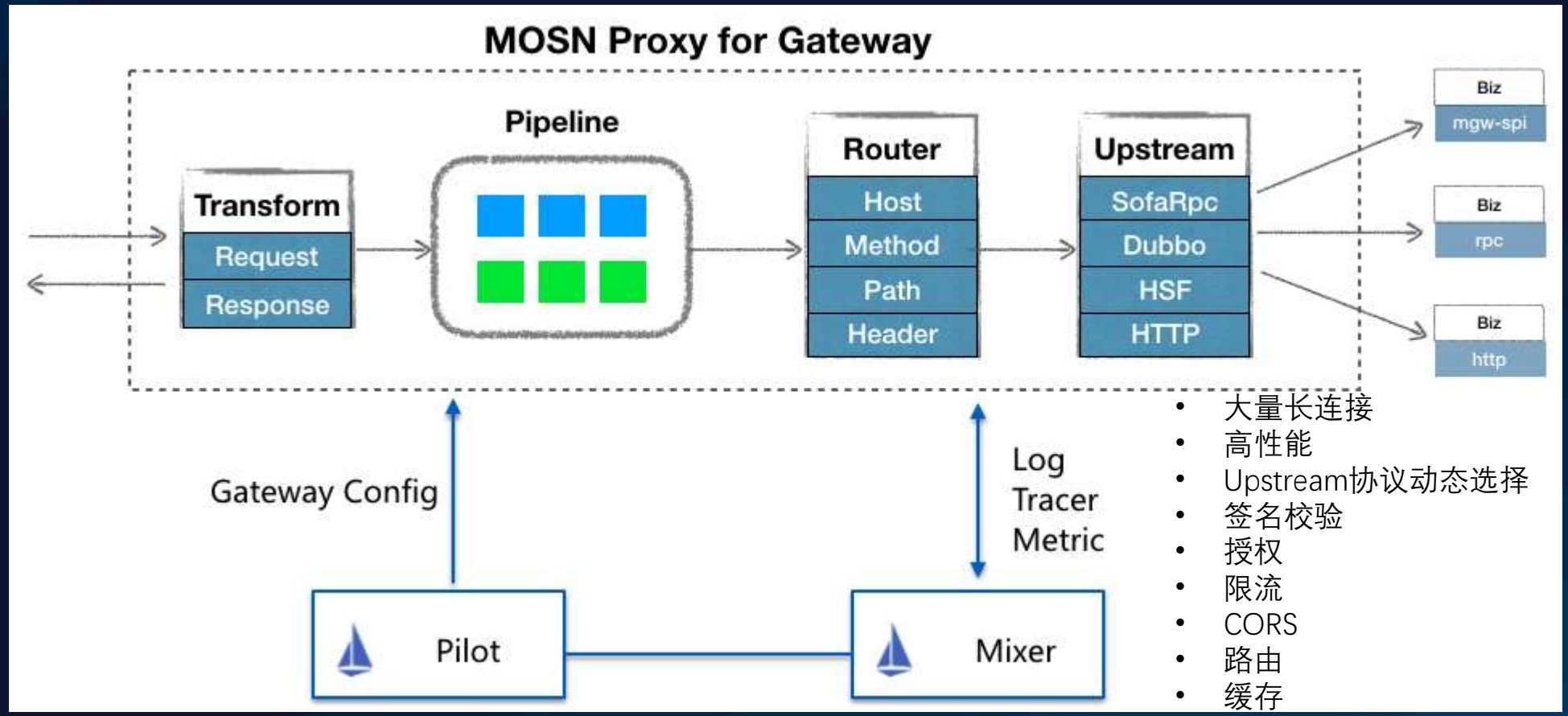
网络层

- 协议栈接入试点

MOSN – 安全

- ✓RBAC（ing）
- ✓双向链路加密
- ✓WAF (todo)
- ✓流量镜像 (todo)

MOSNG 网关 (待开源)



- 大量长连接
- 高性能
- Upstream协议动态选择
- 签名校验
- 授权
- 限流
- CORS
- 路由
- 缓存
- …

蚂蚁金服 ANT FINANCIAL SOFASTACK

MOSN - 多场景支持

- 开源版保证充分的可扩展性，性能支持
- 基于开源版MOSN建设蚂蚁内部版MOSN，通过扩展的方式实现LDC/弹性路由选择，加密机证书集成，基于配置中心的后端服务发现等蚂蚁定制需求
- 基于开源版MOSN建设网关类转发产品MOSNG对功能扩展，性能等方面的需求（待开源）

MOSN 0.4.0 功能特性

分类	特性	已支持	待支持
协议	HTTP	支持高性能HTTP/1.1转发，并接入MOSN IO/Net，Metrics收集等基础能力	XF Headers HTTP/2.0 初始化参数 HTTP/2.0 Server Push
		性能优化的HTTP/2.0转发，并接入MOSN IO/Net，Metrics收集等基础能力	
		支持协议自动识别、Upgrade、GRPC、WebSocket	
	RPC	支持高性能SOFARPC转发	
		基于Xprotocol浅解包支持Dubbo转发	
		提供两种模式的可扩展的RPC协议框架，支持自定义RPC，支持协议自动识别	
	TCP	支持TCP代理	
	TLS	支持端口维度TLS，mTLS	XFCC 域名维度TLS Context
		支持SNI多域名接入	
		支持服务端明文、密文自动识别	
		支持可扩展的证书获取方式	

MOSN 0.4.0 功能特性

分类	特性	已支持	待支持
路由 & LB	域名匹配	多层域名匹配	CORS策略 基于请求header控制的路由方式 更多LB算法
	简单路由匹配	Path / Headers / Prefix	
	复杂路由匹配	Label based SubSet	
	流量分发	Cluster / Host Weight支持	
	拦截路由	Direct Response	
	请求处理	HTTP Rewrite & Headers Custom	
	流量镜像	Traffic Shadow	
	重试	简单计数重试	
	路由扩展机制	可扩展的链式自定义路由	
	LB算法	Random / SRR	

MOSN 0.4.0 功能特性

分类	特性	已支持	待支持
后端集群管理	Cluster管理	Cluster / SubSet Cluster	优先级等功能
	后端池化	支持各协议后端连接池	被动健康检查 完善后端熔断机制 实现更多后端获取方式
	主动健康检查	支持各协议健康检查、心跳	
	后端TLS	支持可配置的后端TLS	
	后端资源熔断	支持简单的后端熔断	
	扩展性	可扩展的后端获取方式	
流量控制	故障注入	故障注入	基于请求参数控制的故障注入
	流控	支持QPS、Rate限流	支持黑/白名单

MOSN 0.4.0 功能特性

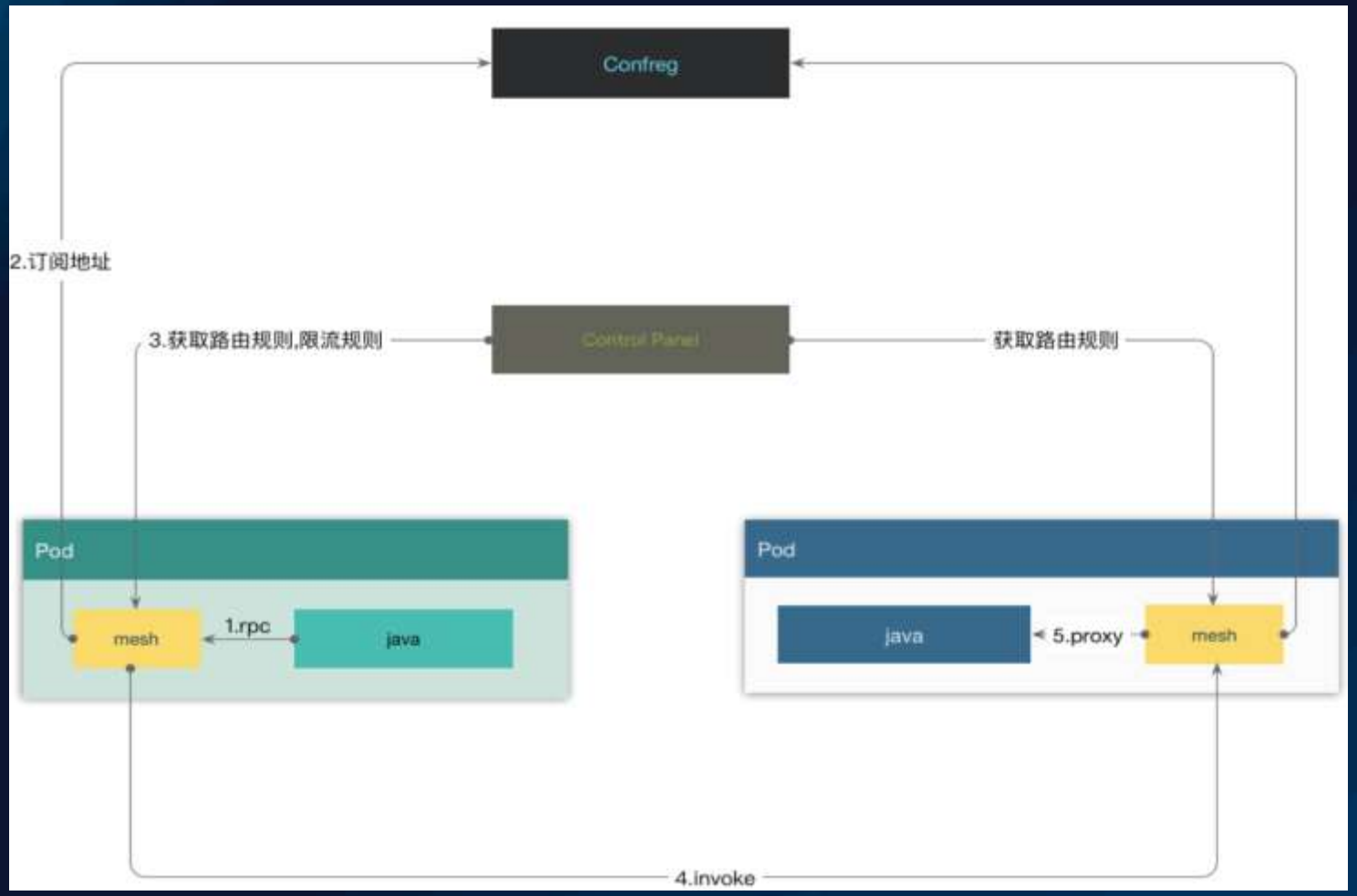
分类	特性	已支持	待支持
监测	Metrics	IO、协议、前后端核心metrics	
		支持无损迁移	
	Tracing	Tracing Framework	
		SOFARPC Tracer	HTTP Tracing完善
	Logging	Request Access Log	
遥感	Report	Report Request / Response 信息	Report更多监测信息
配置更新	XDS	支持基于ADS的配置更新	支持多种可扩展的配置模式
		支持ISTIO sidecar、gateway、router 配置模式	
	高性能配置更新	基于RCU的动态配置更新	
ADMIN	查询	查询当前生效Config	
		查询当前Metrics	

MOSN 0.4.0 功能特性

分类	特性	已支持	待支持
基础能力	无损升级	HTTP/1.1 长、短链接无损迁移	HTTP/2.0 基于goaway无损迁移
		无状态RPC无损迁移	
		TLS无损迁移	
	扩展性	支持IO处理扩展	
		支持监听器扩展	
		支持请求处理扩展	
	IO/Net模式	Golang原生IO方式	
		NetPoll网关模式	
	通用能力	内存池化复用机制	
		Stream处理引擎	

技术案例解析

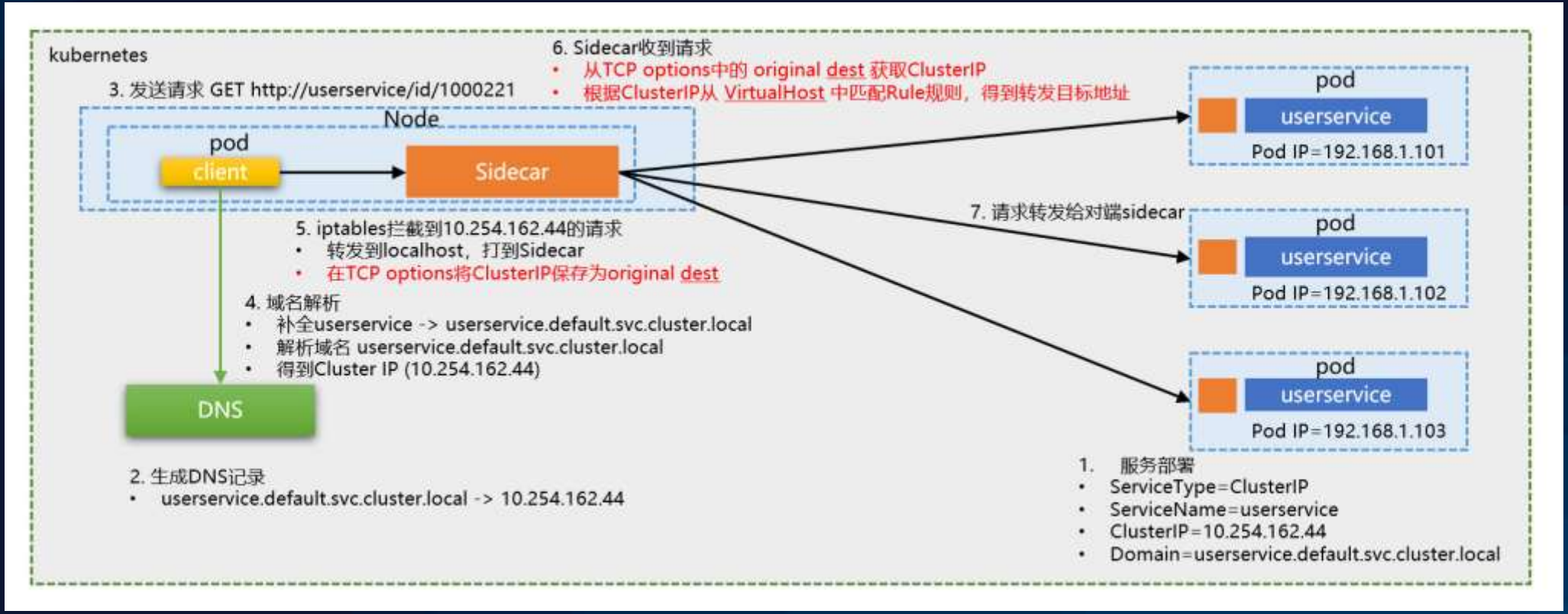
服务接入 – 蚂蚁配置中心方案



- ✓ 通过中间件通道对应用推送MOSN调用地址
- ✓ 通过扩展cluster类型的方式动态获取配置中心后端
- ✓ MOSN出向路由基于明确的服务依赖关系生成
- ✓ 服务通过 id:version 定义
- ✓ 适用于SOA化服务，标准微服务

蚂蚁金服 ANT FINANCIAL SOFASTACK

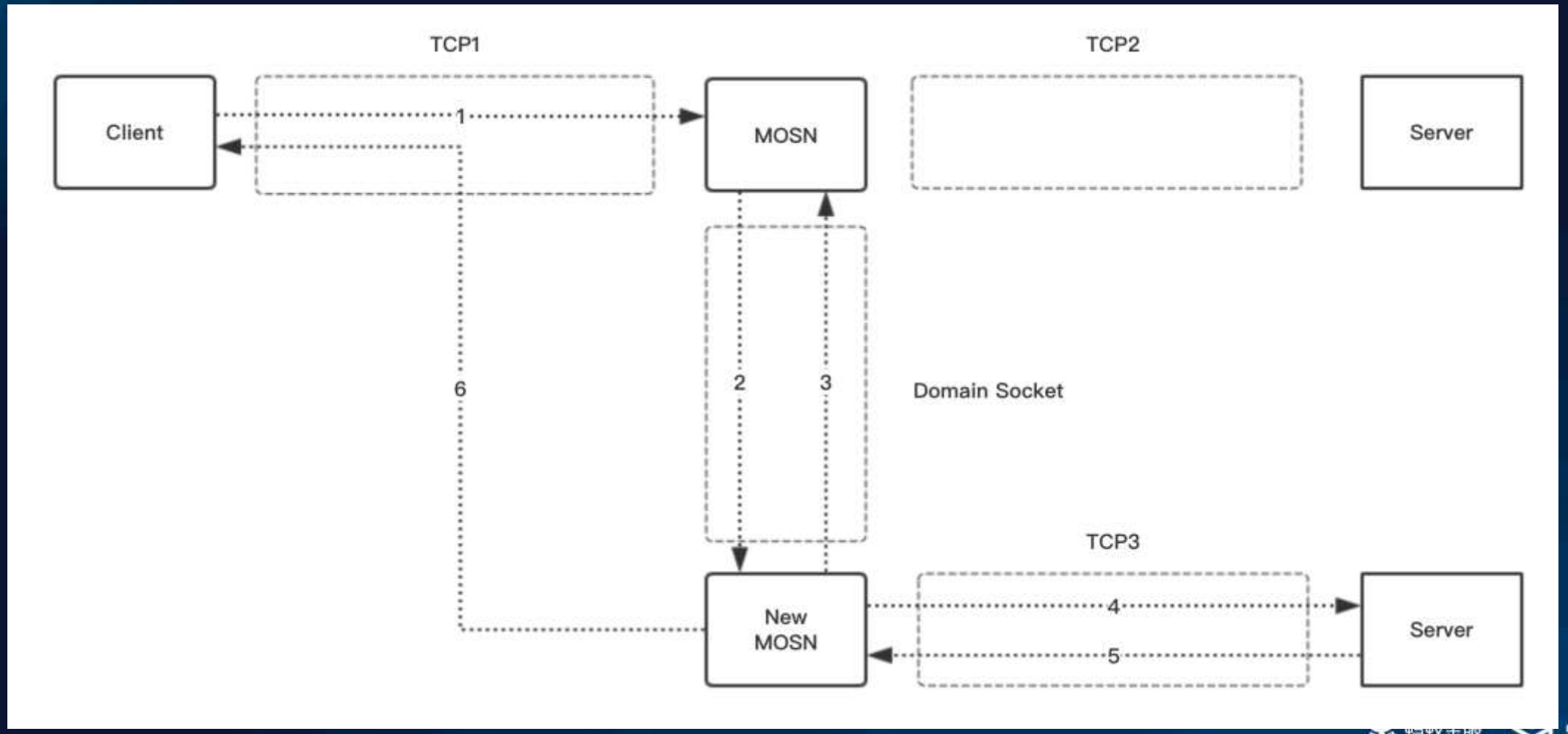
服务接入 – XProtocol DNS方案



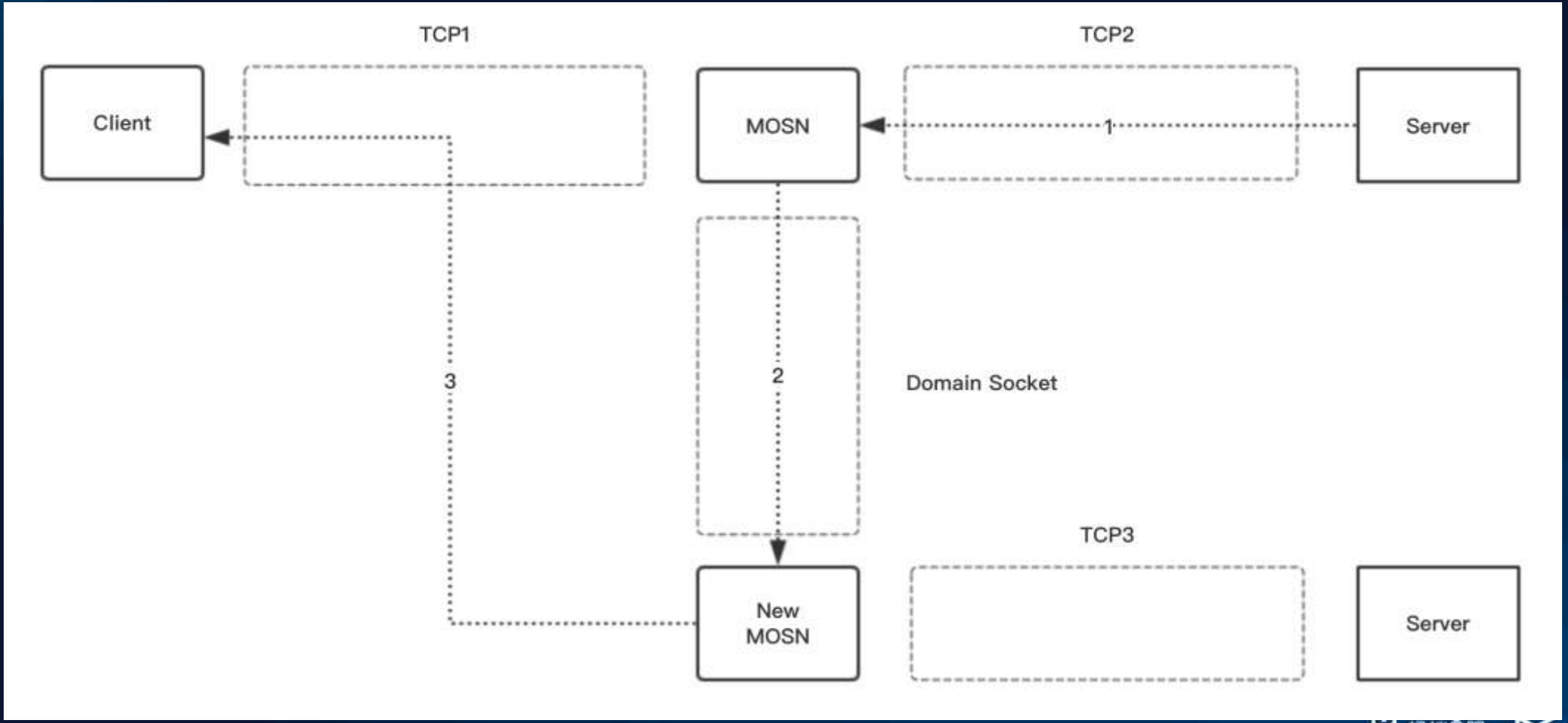
无损平滑迁移

- ✓支持平滑升级，平滑reload
- ✓支持存量链接无损迁移
- ✓支持无状态RPC迁移
- ✓支持HTTP/1.1迁移，含长连接
- ✓支持TLS
- ✓较好解决MOSN升级/reload造成业务抖动等问题

IO平滑迁移


TCP1
TCP2
Client
1
MOSN
Server
6
2
3
Domain Socket
TCP3
New
MOSN
4
5
Server

IO平滑迁移


TCP1
TCP2
Client
MOSN
1
Server
2
3
Domain Socket
TCP3
New
MOSN
Server

TLS平滑迁移

- ✓支持AEAD模式Cipher套件
- ✓支持TLS状态迁移
  - ✓加密秘钥
  - ✓Seq序列
  - ✓TLS读写缓存
  - ✓秘钥状态
- ✓明/密文链路自动探测

单核性能测试

- ✓压测版本: 0.2.1 – 2018.08版本
- ✓部署模式: ServiceA <-> MOSNA <-> MOSNB <-> ServiceB
- ✓压测机型

节点	OS	CPU
MosnA	3.10.0-327.ali2010.rc7.alios7.x86_64	Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
MosnB / ServiceB	3.10.0-327.ali2010.rc7.alios7.x86_64	Intel(R) Xeon(R) CPU E5-2430 0 @ 2.20GHz

- ✓Client模拟方式
  - ✓ SOFARPC
    - ✓ 通过蚂蚁内部压测平台建立500条链接
  - ✓ HTTP/1.1
    - ✓ ab -n 2000000 -c 500 -k
  - ✓ HTTP/2.0
    - ✓ h2load -n1000000 -c5 -m100 -t4
- ✓压测内容: 1K 请求/响应

蚂蚁金服 ANT FINANCIAL SOFASTACK

单核性能测试

场景	QPS	RT(ms)	MEM(K)	CPU(%)
SOFARPC	16000	15.8	77184	98
HTTP/1.1 (原生FastHTTP)	4610	67	47336	90
HTTP/2.0 (原生Golang官方实现)	5219	81	31244	74

性能优化实践总结

- Golang经典的短超时读IO模式在较老版本Linux(2.6.2)下性能损失较大(约30%)，可通过绑核或升级内核(如到4.13.0)解决
- Golang官方HTTP/2.0在单核场景下性能一般，满足生产高性能要求需要深度优化
- 一些有效的优化手段
  - 单次尽量多读 / writev写合并
  - 100K以下的内存避免入堆；避免临时内存入堆
  - 通过内存复用降低GC占比
  - 协程池化，减少协程调度，减少stack扩容/缩容
  - 注意避免G饥饿
  - 性能热点函数优化/替换
  - …

多核性能测试

- ✓压测版本: 0.2.1 – 2018.08版本
- ✓部署模式: Client <-> MOSN<-> Service
- ✓压测机型

节点	OS	CPU
MOSN	3.10.0-327.ali2010.rc7.alios7.x86_64	Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
Service	3.10.0-327.ali2010.rc7.alios7.x86_64	Intel(R) Xeon(R) CPU E5-2430 0 @ 2.20GHz

- ✓Client模拟方式
  - ✓ SOFARPC
    - ✓ 通过蚂蚁内部压测平台建立500条链接,
  - ✓ HTTP/1.1
    - ✓ ab -n 2000000 -c 500 -k
  - ✓ HTTP/2.0
    - ✓ h2load -n1000000 -c5 -m100 -t4
- ✓压测内容: 1K 请求/响应

蚂蚁金服 ANT FINANCIAL SOFASTACK

多核性能测试

场景	QPS	RT(ms)	MEM(K)	CPU(%)
SOFARPC	45000	23.4	544732	380
HTTP/1.1 (原生FastHTTP)	21584	23	42768	380
HTTP/2.0 (原生Golang官方实现)	8180	51.7	173180	300

性能优化实践总结

- Golang官方HTTP/2.0在多核场景下性能不佳，满足生产要求需要深度优化
- G-P-M协程调度模型在单进程模式下可以压到4-8C，更高多核要求可以通过多进程解决
- 在内存复用等优化的前提下，GC表现稳定
- 多进程绑核+reuse port方案在多核场景性能优于单进程多协程方案（约15%），但从进程结构简单出发MOSN选择单进程模型
- 尽量减少系统调用等导致G切换上下文
- 注意避免P饥饿

长连接模式

- ✓压测版本: 0.2.1 – 2018.08版本
- ✓部署模式: Client <-> MOSN<-> Service
- ✓压测机型

节点	OS	CPU
MOSN	3.10.0-327.ali2010.rc7.alios7.x86_64	Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
Service	3.10.0-327.ali2010.rc7.alios7.x86_64	Intel(R) Xeon(R) CPU E5-2430 0 @ 2.20GHz

- ✓Client模拟方式
  - ✓SOFARPC
    - ✓通过蚂蚁内部压测平台建立10w条链接
- ✓压测内容: 1K 请求/响应

长连接模式

场景	QPS	MEM(K)	CPU(%)	goroutine
原生IO模式	1000	3.3	60	200028
Raw Epoll模式	1000	2.5	18	28

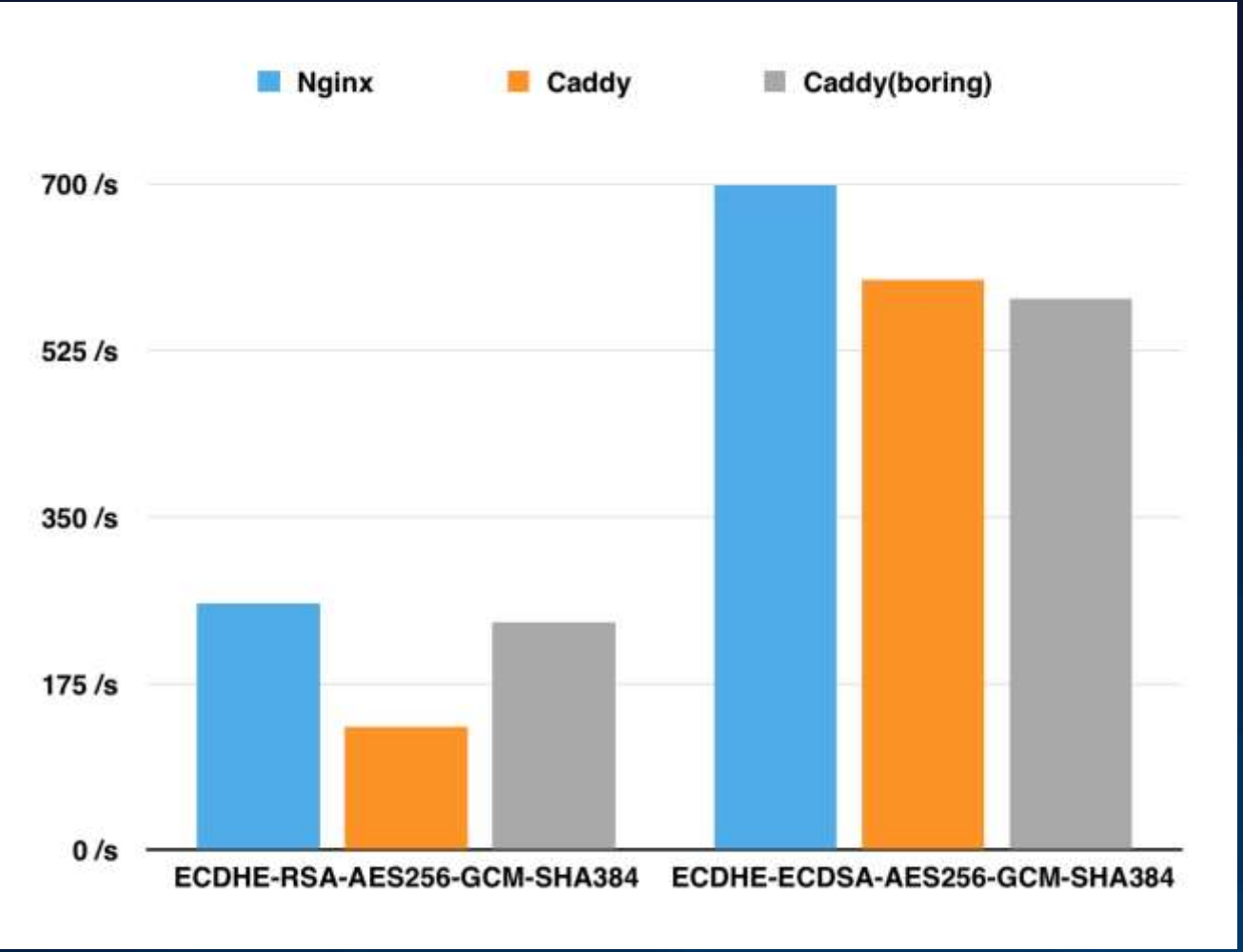
性能优化实践总结

- 原生协程IO模式暂时无法很好满足C10K场景性能要求，不适合高性能网关场景

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



蚂蚁金服 ANT FINANCIAL SOFASTACK

性能优化实践总结

- 除 p384，RSA 外，Golang原生对很多算法有汇编优化，性能好于boring SSL golang
- 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

性能优化实践总结

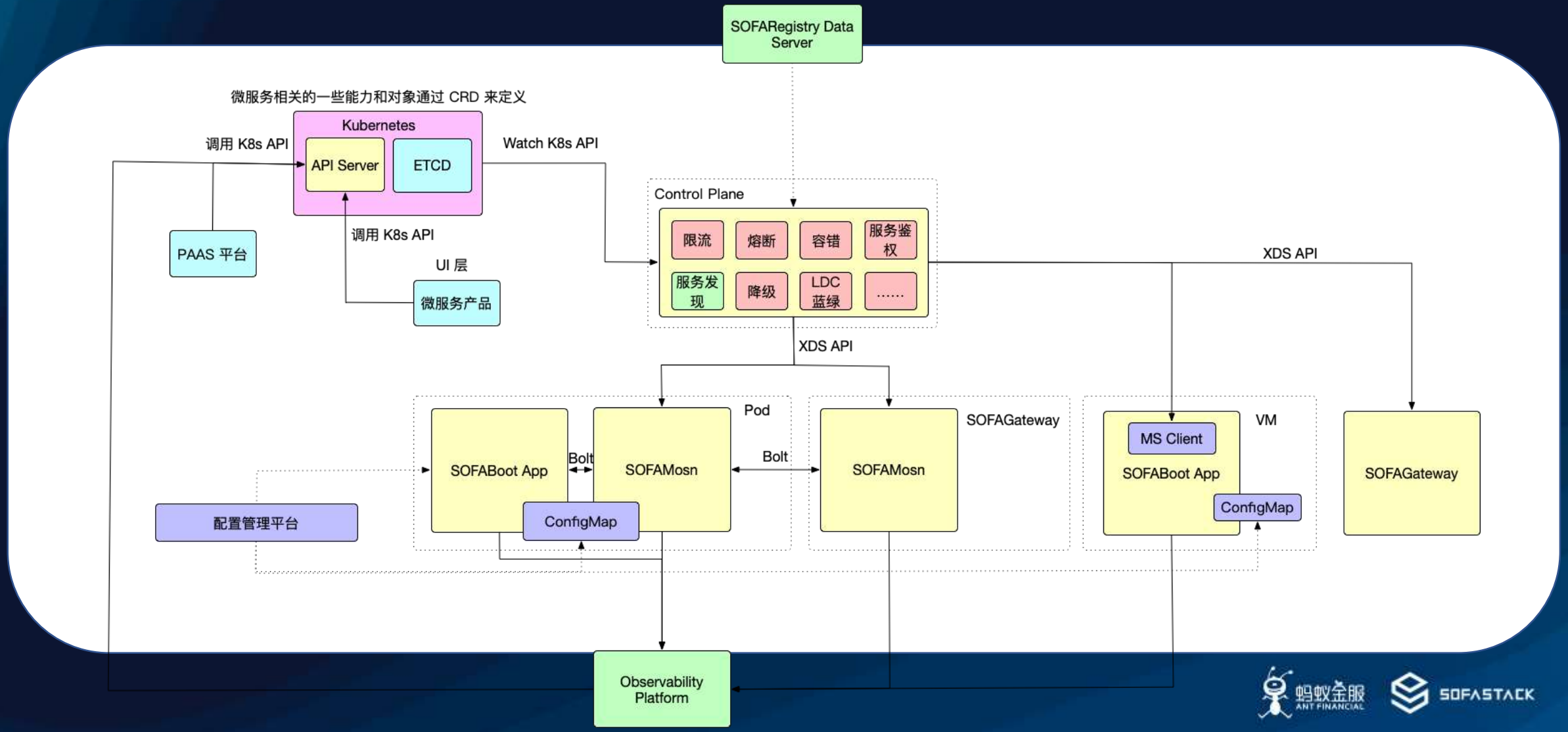
- SOFARPC 经过IO，内存，协议等优化，较0.1.0版本QPS提升50%，内存使用减少40%
- HTTP/2.0 优化针对IO，内存，流处理等，仍在进行中
- Golang内存/线程模型与C/C++有较大区别，思路/算法/工具迁移有一定成本
- 系统编程领域Golang体系是“新世界”，在协议栈加速等深度优化方面仍有大量工作要做，但潜力无限

更多技术案例

- 协议 / TLS自动识别
- 基于RCU机制的高性能动态配置更新
- 串行化stream事件处理
- QPS / 令牌桶限流
- 链式路由、 Multiple Cluster
- Metrics平滑迁移
- …

总结 & 展望

蚂蚁微服务Mesh化探索



蚂蚁金服 ANT FINANCIAL SOFASTACK

MOSN - X

蚂蚁金服 ANT FINANCIAL SOFASTACK

Q&A

金融级分布式架构

ServiceMesher

欢迎关注微信公众号，获取更多技术干货！

https://github.com/alipay/sofa-mosn
https://github.com/alipay/sofa-mesh
xiaodong.dxd@antfin.com

蚂蚁金服 ANT FINANCIAL SOFASTACK

GIAC