稀土掘金 CloudWeGo

字节跳动服务网格基于 Hertz 框架的实践

兰新宇

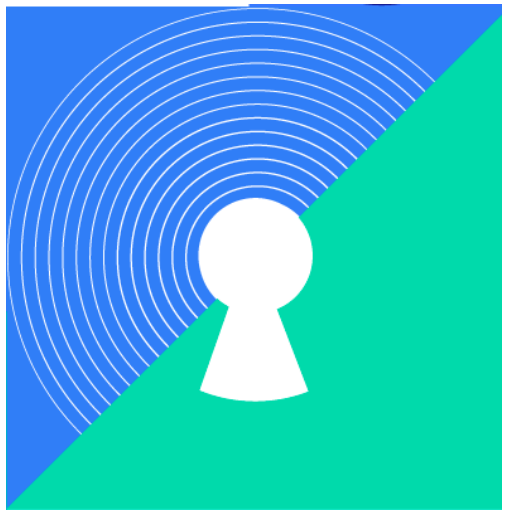
字节跳动服务网格研发工程师

CloudWeGo | 稀土掘金 出品

2022/06/25

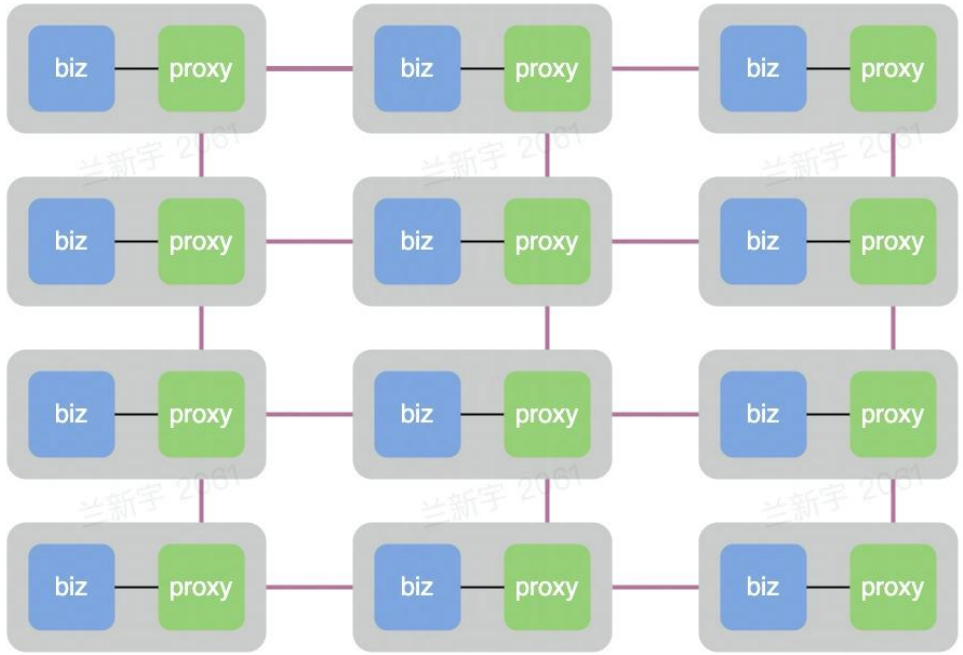
目录

CONTENT



服务网格在字节跳动

服务网格在字节跳动 – 什么是服务网格



与业务和框架解耦

单一职责在 RPC 通信领域的体现：

业务和 RPC 框架研发同学可以更专注与核心功能的实现

灵活、可靠的生命周期管理

服务网格的开发、运维者能够对生命周期做更好的把控

统一的跨语言流量治理体验

在字节跳动，多语言技术栈非常普遍：

C/C++ Python Java Golang 等服务都有很强的流量治理诉求

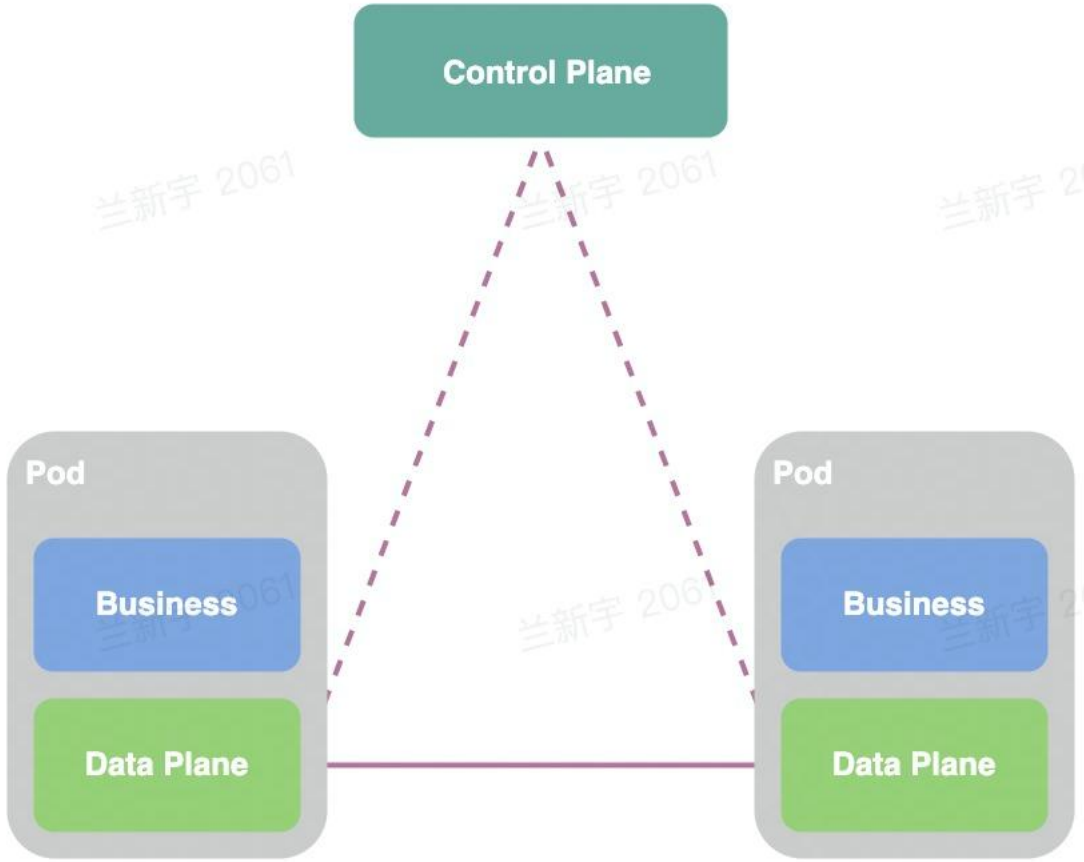
服务网格在字节跳动 – 典型架构

数据面

- 高性能网络代理
- 流量治理能力实现

控制面

- 流量控制
- 安全控制
- 可观测性



服务网格在字节跳动 – 落地挑战

01

多样的技术栈

02

200w+

5w+

海量的微服务

03

TikTok

头条

抖音

复杂的业务形态

服务网格在字节跳动 – 落地实践

数据面

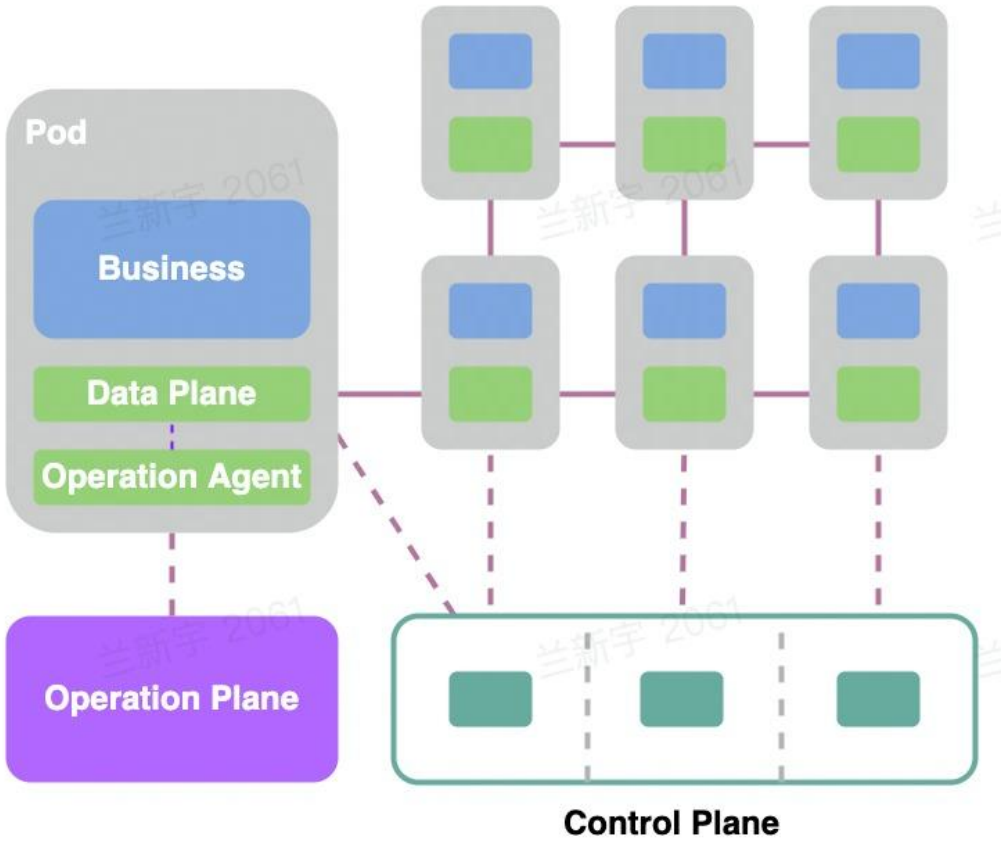
- 动态配置能力
- Share Memory IPC
- LTO

控制面

- 纯自研，支持服务动态接入网格
- 多集群部署，有效控制爆炸半径
- 多样的流量治理能力，持续为业务赋能

运维面

- 全面掌控网格内部通信
- 发布确认机制
- 版本锁定



超复杂调用网络下的流量治理体系

超复杂调用网络下的流量治理体系 – 核心能力

01

微服务访问控制

- 授权
- 鉴定
- 流量加密

02

单元化

- HTTP
- RPC
- MQ

03

动态治理

- 动态过载控制
- 动态负载均衡

超复杂调用网络下的流量治理体系 – 落地规模

13000+

2500+

微服务访问控制

1000+

单元化

7500+

动态过载控制

2500+

动态负载均衡

超复杂调用网络下的流量治理体系 – 微服务访问控制

授权

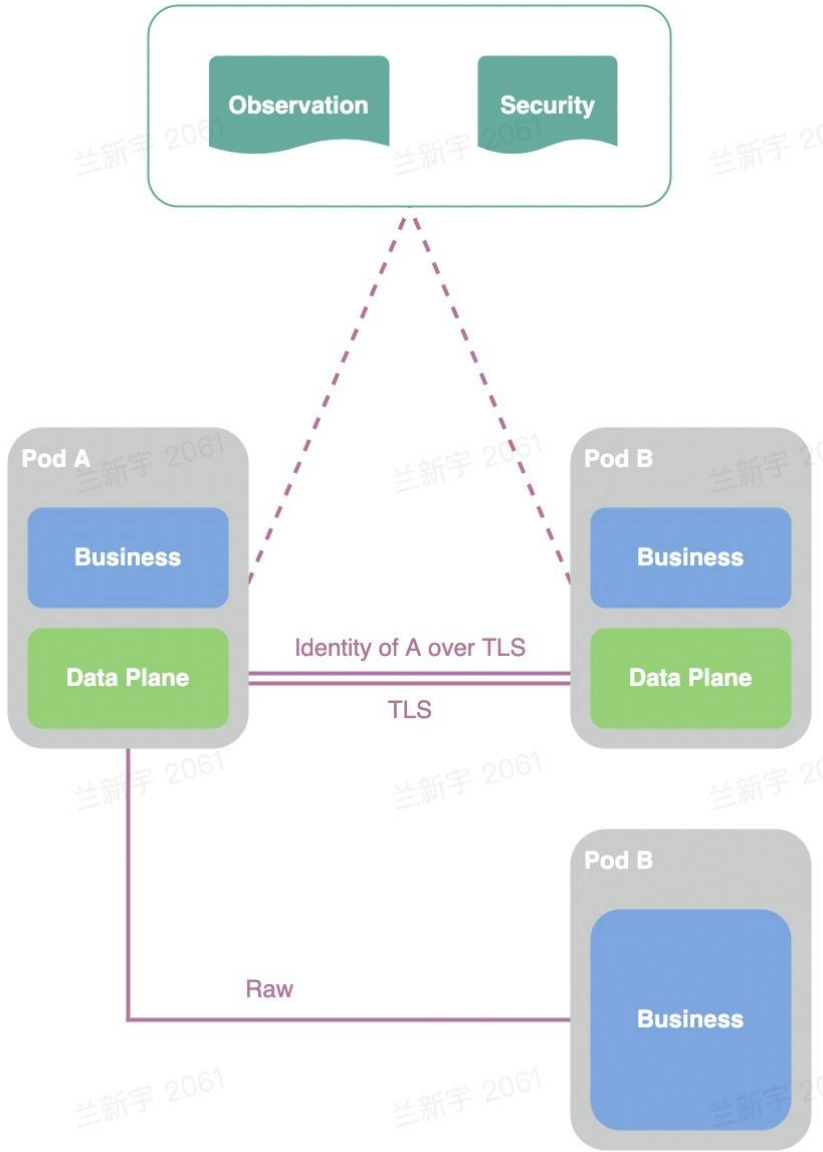
- 观测能力（Dry Run）
- 巡检系统，控制增量风险

鉴定

- 观测能力（Dry Run）
- 动态降级能力（过期票/无效票/空票）

流量加密

- 宽松/严格 模式



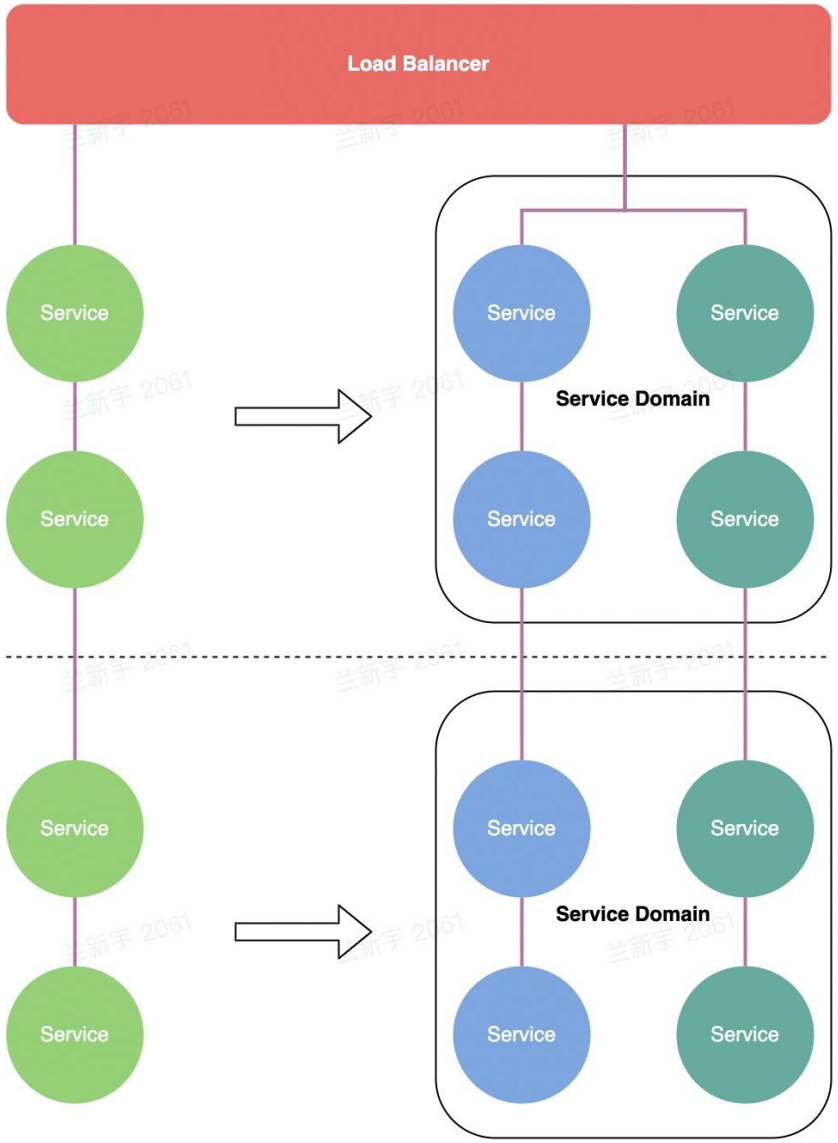
超复杂调用网络下的流量治理体系 – 单元化

特点

- 请求特征级别的智能路由能力
- 请求特征级别的治理能力
- 动态更新，无业务入侵
- 支持观测、按比例灰度上量

应用场景

- 多租户
- 常态化容灾演练
- 流量隔离



超复杂调用网络下的流量治理体系 – 动态治理（过载保护）

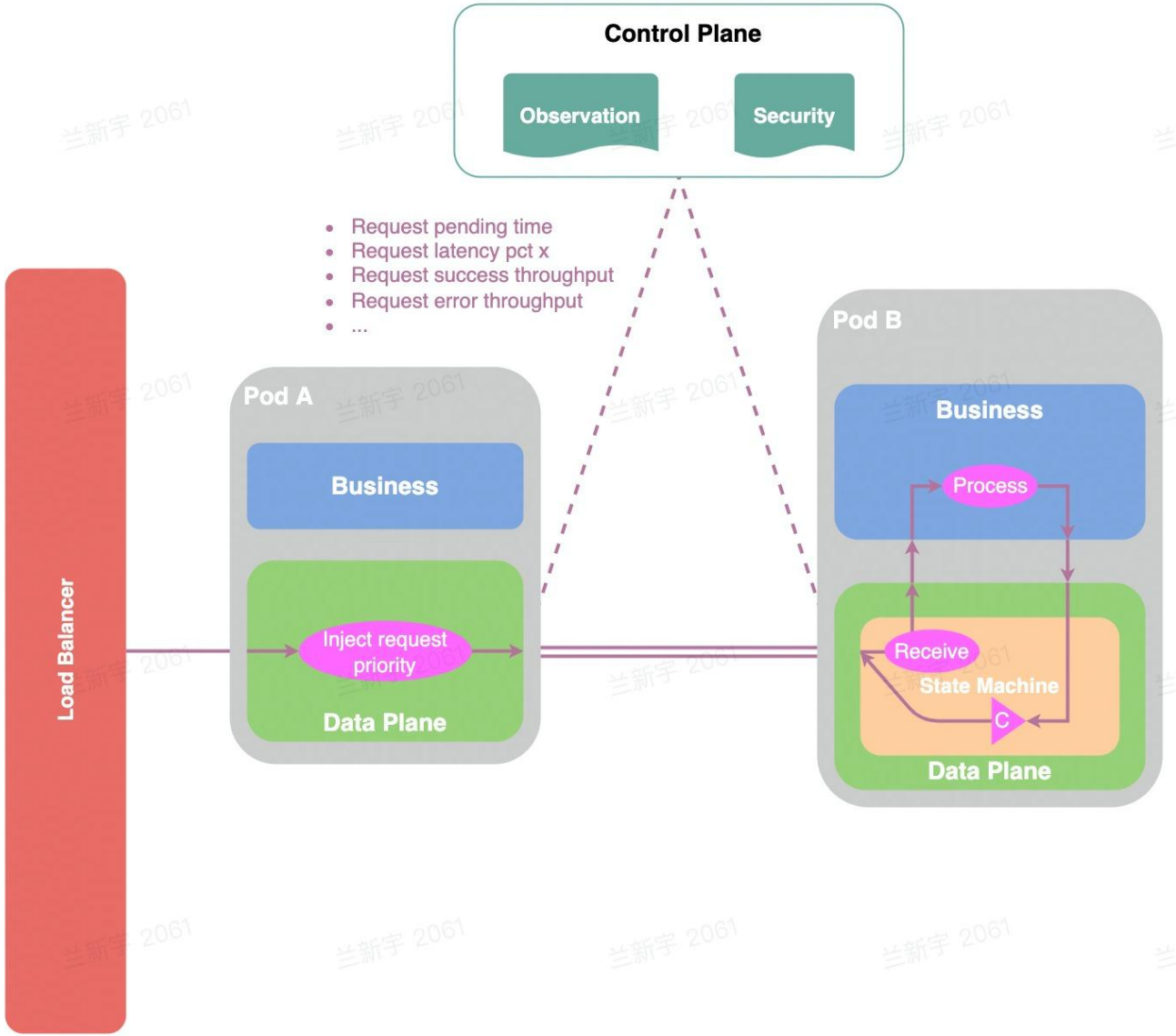
过载依据

输入：

- CPU 利用率 > x
- **排队时间 > x**
- QPS > x
- 客户端超时断连事件

输出：

- 请求优先级 < x -> 丢弃



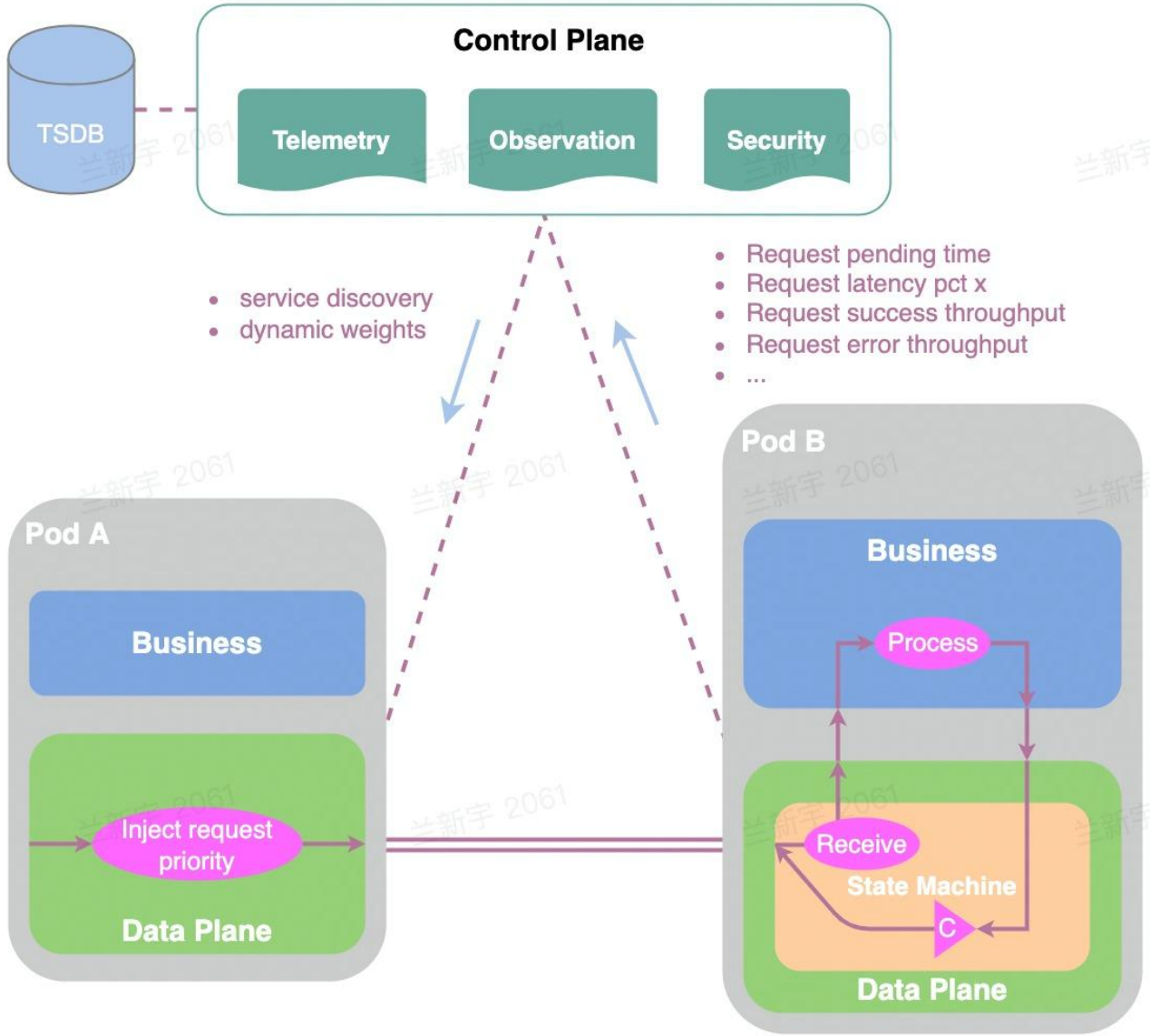
超复杂调用网络下的流量治理体系 – 动态治理（负载均衡）

数据面

- 高精度指标的采集、上报
- 基于权重的负载均衡算法

控制面

- 基于指标的动态权重计算
- 基于动态权重的流量控制



Hertz 在服务网格的落地实践

Hertz 在服务网格的落地实践 – 技术选型

性能

13M+

易用性

- 容易写出高质量代码
- 易用且丰富的 API

Bugs
Issues
Features

长效支持

- 开源方案 or 内部自研

Hertz 在服务网格的落地实践 – Hertz Introduction

CloudWeGo - Hertz

- Hertz 是字节跳动开源的基于 Golang 的高性能 HTTP 框架

特点

- 高易用性：字节跳动内部沉淀两年
- 高性能：高吞吐、低时延
- 高拓展性
- 多协议支持
- …

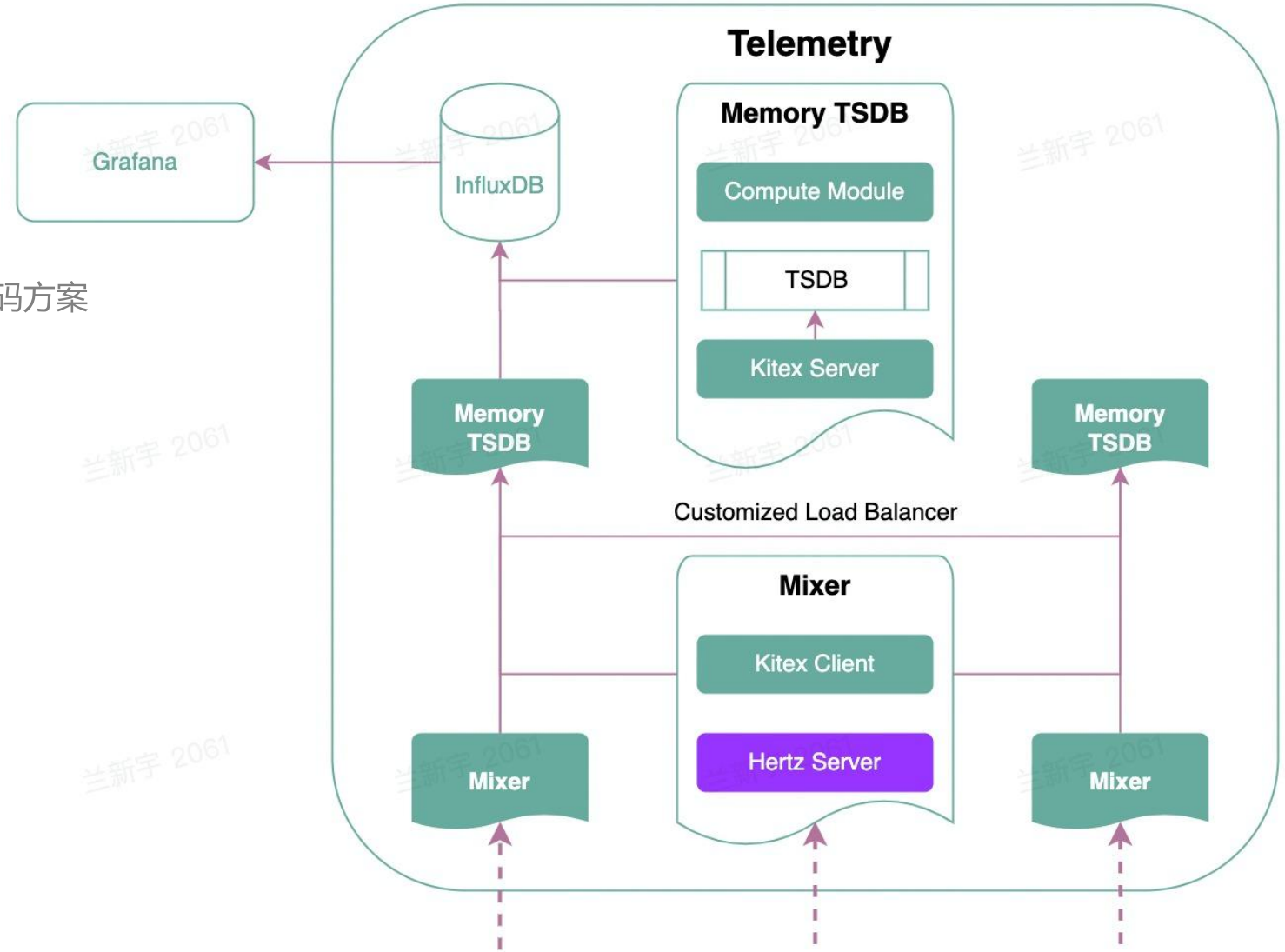
Hertz 在服务网格的落地实践 – Telemetry Mixer

Hertz Server

- 高并发、高吞吐的 HTTP Server
- 无缝集成基于 Sonic 的高性能 JSON 编解码方案

Kitex

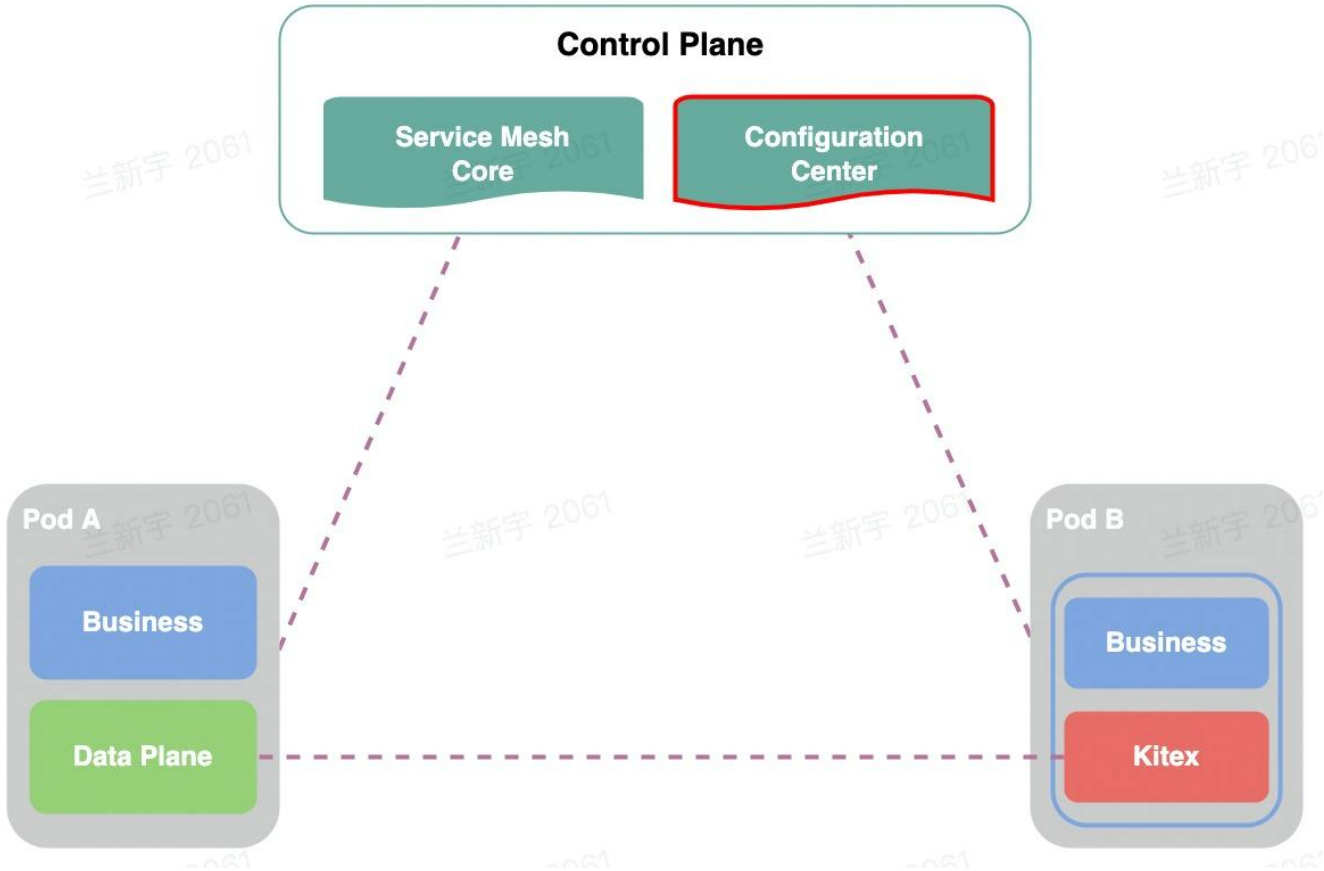
- 自定义负载均衡策略
- Thrift + Frugal



Hertz 在服务网格的落地实践 – Configuration Center

Hertz Server

- 高并发、高吞吐的 HTTP Server
- 无缝集成基于 Sonic 的高性能 JSON 编解码方案



Hertz 在服务网格的落地实践 – 收益分析

性能

- 稳定承载线上超 13M QPS 流量
- CPU 火焰图占比符合预期

易用性

- 使用者更能专注于业务逻辑
- 使用者更容易写出高效代码

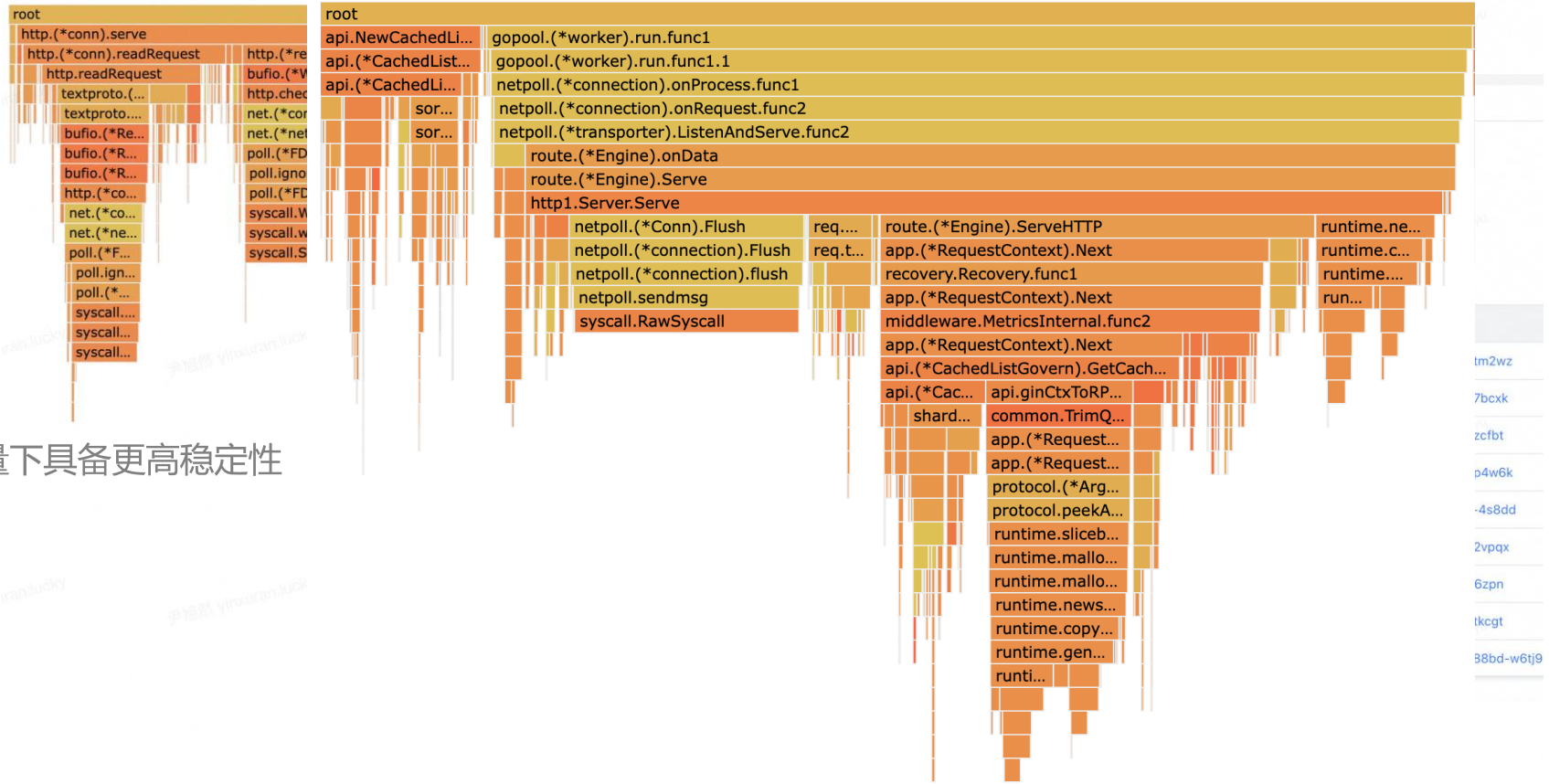
长效支持

- Hertz 已贡献给 CloudWeGo 开源社区
- 开源和内部为统一版本

Hertz 在服务网格的落地实践 – 收益分析之性能

痛点

- Goroutine 数量较多，稳定性差
- Gin 框架自身开销较大



Hertz 收益

- 基于 Netpoll 网络库，同等吞吐量下具备更高稳定性
- Goroutine 数量 6w -> 80
- 框架开销大幅降低
- 稳定承载线上超 13M QPS 流量

Hertz 在服务网格的落地实践 – 收益分析之易用性

痛点

- 不容易写出高性能代码
- 不容易设计好的容错机制

Hertz 收益

- 易用的对象回收机制
- 易用的连接超时+重试机制

```
var (
    requestPool sync.Pool
    cli         http.Client
)

func AcquireRequest() *http.Request {
    v := requestPool.Get()
```

```
req := hertz_protocol.AcquireRequest()
defer hertz_protocol.ReleaseRequest(req)
req.SetMethod(consts.MethodPost)
req.SetRequestURI("xxx")
req.Header.SetContentTypeBytes([]byte("application/json"))
req.SetBody(mysqlReqBytes)

res := hertz_protocol.AcquireResponse()
defer hertz_protocol.ReleaseResponse(res)
err = cli.Do(ctx, req, res)
```

```
    cli = http.Client{
        Transport: yy, // Control connection timeout.
        Timeout:   zz, // Control read/write timeout.
    }
}

func main() {
    req := AcquireRequest()
    // Set correct URL and other necessary members...
    // req.URL = ...
    cli.Do(req)
    ReleaseRequest(req)
}
```

Hertz 在服务网格的落地实践 – 收益分析之长效支持

痛点

- 开源项目迭代速度往往不达预期
- 被迫维护开源项目的克隆

Hertz 收益

- 强大的 CloudWeGo 社区运营
- 字节跳动内部广泛使用
- 字节跳动内部版本亦基于开源版本构建

内外版本维护

完整的微服务体系离不开基础的云生态，无论在
的治理，比如治理平台、监控、链路跟踪、注册
持微服务体系，但这些服务短期还无法开源，那

CloudWeGo 下与内部生态没有耦合的项目，如

而需要集成治理能力融入微服务体系的 Kitex 则
内部用户无感知升级。集成内部治理特性的模块
开源库。

对于使用 Kitex 的开源用户，同样可以对 Kitex
为更多用户提供便利。

发布

Kitex

- **Release v0.3.2**
- Release v0.3.0
- Release v0.2.1
- Release v0.2.0
- Release v0.1.4
- Release v0.1.3
- Release v0.1.2
- Release v0.1.0
- Release v0.0.8
- Release v0.0.5
- Release v0.0.4
- Release v0.0.3
- Release v0.0.2
- Release v0.0.1

Netpoll

- Release v0.2.2
- Release v0.2.0
- Release v0.1.2
- Release v0.1.1
- Release v0.1.0
- Release v0.0.4

施开发微服务，都需要搭建额外的服务以很好的支持微服务
存在一些定制的规范。字节跳动自然也有完善的内部服务支
统一迭代呢?

调整为开源库。

分，Kitex 的核心代码迁移到开源库，内部库封装一层壳保证
对于一些新的特性也会优先在内部库支持，稳定后迁移到

系中，也希望开发者能贡献自己的扩展到 kitex-contrib，

FAQ

沟通群

公众号

THANKS