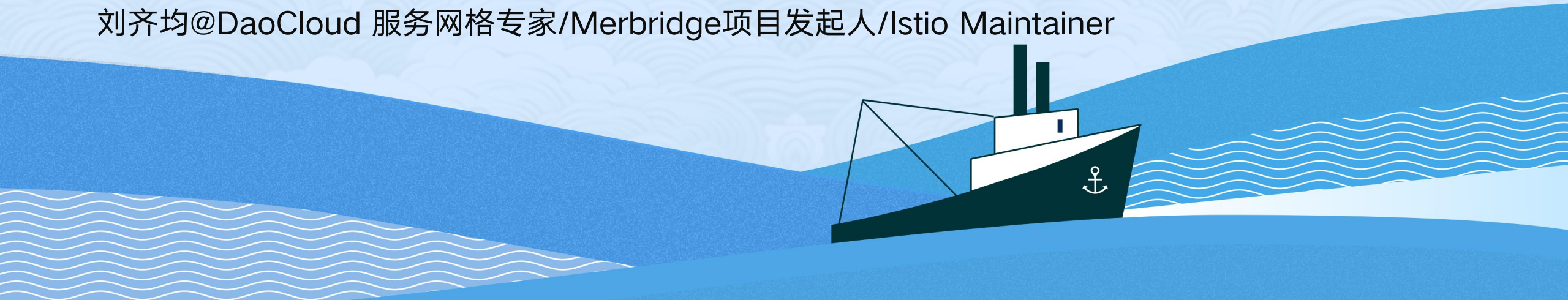
KUBERNETES COMMUNITY DAYS BEIJING 2023

KUBERNETES
COMMUNITY DAYS | BEIJING

优化服务网格以提高性能和高可用性

王夕宁@阿里云　云原生服务网格负责人

刘齐均@DaoCloud 服务网格专家/Merbridge项目发起人/Istio Maintainer



阿里云服务网格ASM的技术架构

https://www.aliyun.com/product/servicemesh

异构服务统一治理（流量管理、可观测、安全）

跨集群的应用网络管理

与CI/CD工具融合的应用灰度发布与部署

应用上云架构平滑演进

基于KServe的AI弹性服务管理

Web用户界面/被集成能力：Open API/Terraform　　声明式API，兼容社区Istio，支持控制面与数据面K8s API访问

ASM控制面:托管核心组件，标准/企业版架构统一，柔性架构、多版本支持、定制能力增强

- 托管核心组件 ASM Infra
- 流量管理& 协议增强
- 可观测性& 弹性伸缩
- 零信任安全
- 自适应 xDS优化
- 软硬一体优化
- 网格诊断 智能分析
- Envoy Filter 扩展中心
- 异构服务 注册集成

为运行在异构计算基础设施上的服务提供统一的网格化治理能力

ASM数据面

入口

出口

阿里云VPC

其他公有云或IDC

Pod
ASM 网关
WA
Envoy扩展
证书管理
协议增强

Pod
Proxy
WA
Service

ECI Pod
Proxy
WA
Service

ECS VM
Proxy
WA
Service

ENS Pod
Proxy
WA
Service

外部集群
Proxy
WA
Service

数据面的扩展及运维

网格CNI插件兼容优化

eBPF数据面性能加速

Proxyless支持及与Proxy的互通

多集群支持

混合部署

多云混合云

ACK K8s集群&ECI虚拟节点

Serverless K8s集群

ECS虚拟机

边缘集群

外部注册集群

构建网格优化中心， 提升产品的性能指标与高可用性

多维度、不同层面的性能优化

I. 收敛服务发现的范围, 提升网格配置的推送效率

II. 基于访问日志分析自动推荐生成Sidecar对象, 以减少代理资源消耗

III. AdaptiveXDS：自适应配置推送优化

IV. 软硬结合性能优化的持续推进

V. 资源超卖模式下的支持

VI. 基于eBPF tcpip-bypass的数据面性能优化

首批通过可信云服务网格性能测评
先进级认证、排名第一、全项满分

TTL泰尔实验室 CAICT 中国信通院

可信云 服务网格性能评估

证书编号：IY2022201401
报告编号：V22Y21401

服务网格性能检验证书

阿里云计算有限公司

杭州市西湖区转塘科技经济开发区块16号8幢

阿里云服务网格（ASM）

产品版本：2021SR0366408

先进级

符合性描述：“经中国信息通信研究院的检验，贵单位的阿里云服务网格（ASM）通过了Q/KXY R014-2022《基于服务网格的平台性能测试方法》先进级检验，特发此证。”

(由于该检验是针对2022年6月15日前该云产品/服务情况进行披露，用户依然应留意在具体使用过程中可能发生的风险。因云计算技术快速发展，产品/服务不断升级迭代，标准规范随行业发展不断更新等原因，当产品或检验标准发生较大变化时，建议本产品/服务进行重新检验。云计算开源产业联盟/中国信通院将在2023年6月组织新一批检验。)

云计算开源产业联盟
地址：北京市海淀区花园北路52号
网址：http://opensource.ecloud.cn

中国泰尔实验室
中国信息通信研究院
地址：北京市海淀区花园北路52号
网址：http://www.caict.ac.cn

授权签字人：

签发日期：2022年6月15日

KUBERNETES
COMMUNITY DAYS | BEIJING

I. 收敛服务发现的范围，提升网格配置的推送效率

I. 收敛服务发现的范围，提升网格配置的推送效率

目标

- 有效降低CPU资源消耗
- 有效降低内存资源消耗
- 有效降低带宽资源消耗

Discovery Selector：

- 基于命名空间范围的筛选,自动发现数据面 Kubernetes 集群中指定命名空间下的服务;
- 与Istiod直接交互，有效提升控制面的处理效率;

```
[
    {
        "matchLabels": {
            "kubernetes.io/metadata.name": "kube-system"
        }
    },
    {
        "matchLabels": {
            "kubernetes.io/metadata.name": "istio-system"
        }
    },
    {
        "matchExpressions": [
            {
                "values": [
                    "true",
                    "enabled"
                ],
                "key": "asm-discovery",
                "operator": "In"
            }
        ]
    },
    {
        "matchLabels": {
            "kubernetes.io/metadata.name": "default"
        }
    }
]
```

I. 收敛服务发现的范围，提升网格配置的推送效率

ExportTo + Workload Selector：

- ExportTo 用于控制规则(VirtualService, DestinationRule, ServiceEntry等)的可用性，Workload Selector 用于配置规则(DestinationRule, ServiceEntry, EnvoyFilter, Sidecar等)适用的服务；
- 既可以支持全局设置或命名级别设置，又可以支持细粒度设置(per service/config)；

```yaml
exportTo:
  - default
  - istio-system
hosts:
  - reviews.prod.svc.cluster.local
http:
  - name: reviews-route
    route:
      - destination:
          host: reviews.prod.svc.cluster.local
          port:
            number: 8080
```

```yaml
host: reviews.prod.svc.cluster.local
subsets:
  - labels:
      version: v1
    name: v1
  - labels:
      version: v2
    name: v2
workloadSelector:
  matchLabels:
    app: reviews
```

KUBERNETES
COMMUNITY DAYS | BEIJING

II. 基于访问日志分析自动推荐生成Sidecar对象，以减少代理资源消耗

II. xDS及Sidecar对象背景介绍

- xDS 是Service Mesh控制面和数据面Sidecar Proxy之间的通信协议，x 表示包含多种协议的集合，比如：LDS 表示监听器的发现，CDS 表示服务和版本的信息，EDS 表示服务和版本有哪些实例，以及每个服务实例的特征信息，RDS 表示路由的发现。
- 可以简单的把 xDS 理解为：网格内的服务发现数据和治理规则的集合。xDS 数据量的大小和网格规模是正相关的。

- 默认情况下，下发 xDS 使用的是全量下发策略，也就是网格里的所有 Sidecar代理内存里都会有整个网格内所有的服务发现数据。
- 在大多数情况下，在大规模集群中的一个简单的工作负载只与少数其他工作负载进行通信。将配置更改为仅包含一组必要的服务会对网格代理的内存占用产生很大影响。
- Sidecar资源对象可以帮助定义这种配置约束关系。

II. 基于访问日志分析自动推荐生成Sidecar对象，以减少代理资源消耗

- 工作原理：
  - 通过分析数据平面网格代理产生的访问日志获取数据平面服务之间的调用依赖关系，为数据平面中的每个工作负载自动推荐生成相应的Sidecar资源对象。
  - 根据分析结果生成出Sidecar资源对象之后，允许用户进行二次确认，或者用户基于自动生成的内容进行定制修改。
- 已知使用场景的限定：
  - 需要依赖于用户开通日志服务，并要求产生的日志需要覆盖全部业务调用，才能得到全部的依赖关系。一旦某业务路径没有调用产生日志，那么很有可能丢失对应的服务依赖关系，从而导致生成的Sidecar资源对象定义内容不准确，进而可能导致之后的该业务路径访问调用不通。

推荐的Sidecar资源

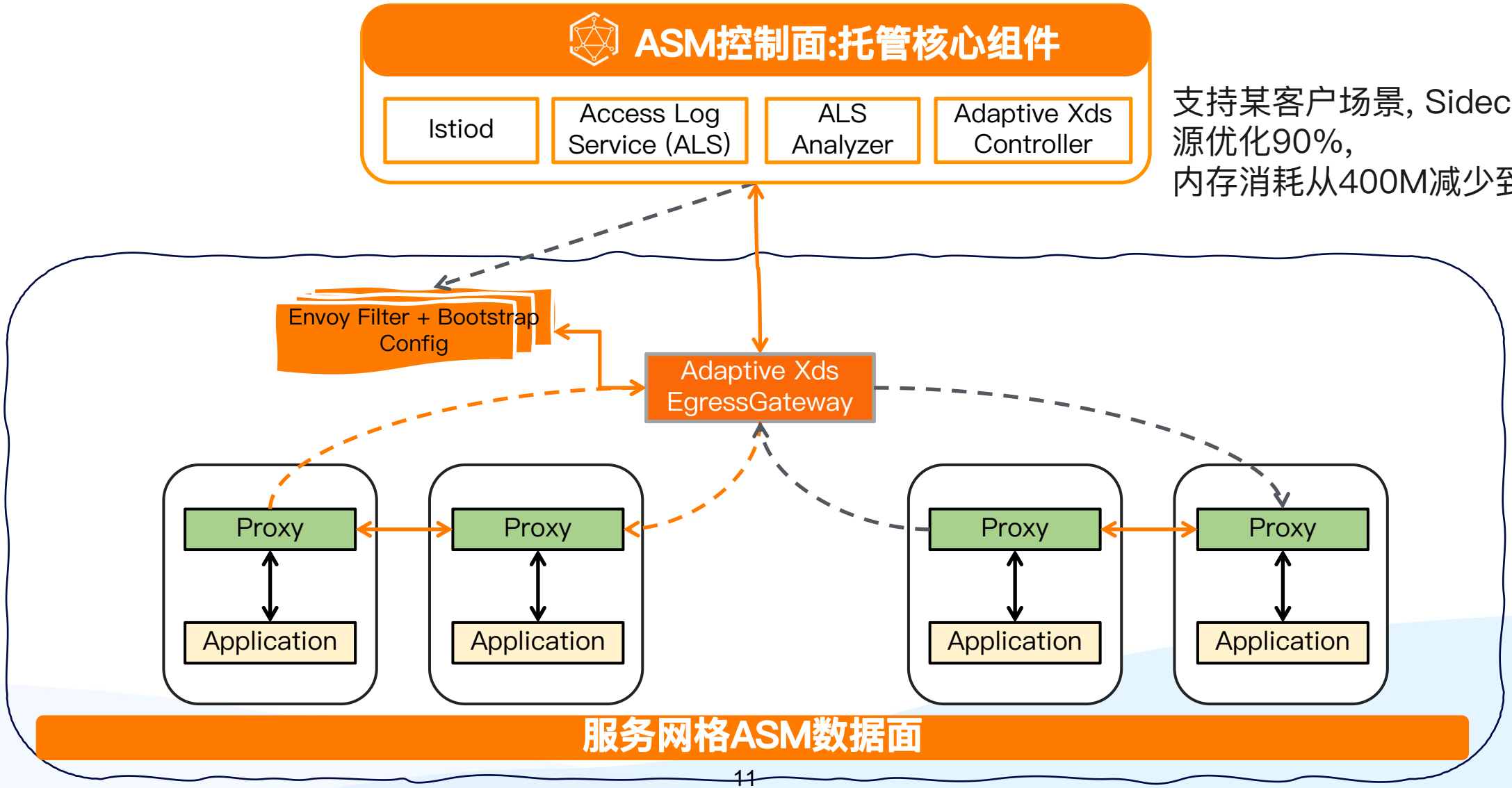
```yaml
apiVersion: networking.istio.io/v1beta1
kind: Sidecar
metadata:
  labels:
    provider: asm
    taskid: '297'
  name: asm-rec-for-default-productpage-v1
  namespace: default
spec:
  egress:
    - hosts:
        - '*/details.default.svc.cluster.local'
        - '*/reviews.default.svc.cluster.local'
      port:
        name: HTTP-9080
        number: 9080
        protocol: HTTP
    - hosts:
        - istio-system/*
        - kube-system/*
  workloadSelector:
    labels:
      app: productpage
      version: v1

```

KUBERNETES
COMMUNITY DAYS | BEIJING

III. AdaptiveXDS：自适应配置推送优化

III. AdaptiveXDS：自适应配置推送优化

提供一种托管式的按需推送xDS配置的能力，自适应应用服务的变更



支持某客户场景，Sidecar资源优化90%，
内存消耗从400M减少到50M。

KUBERNETES
COMMUNITY DAYS | BEIJING

IV. 软硬结合性能优化的持续推进

IV. 从节点特征探测到自适应动态配置相应的特性

阿里云 | 奥运会全球指定云服务商

节点特征探测 NFD → 检测 Kubernetes 集群中每个节点上可用的硬件功能特征，包括CPUID特征、指令集扩展等；

→ 自适应动态配置

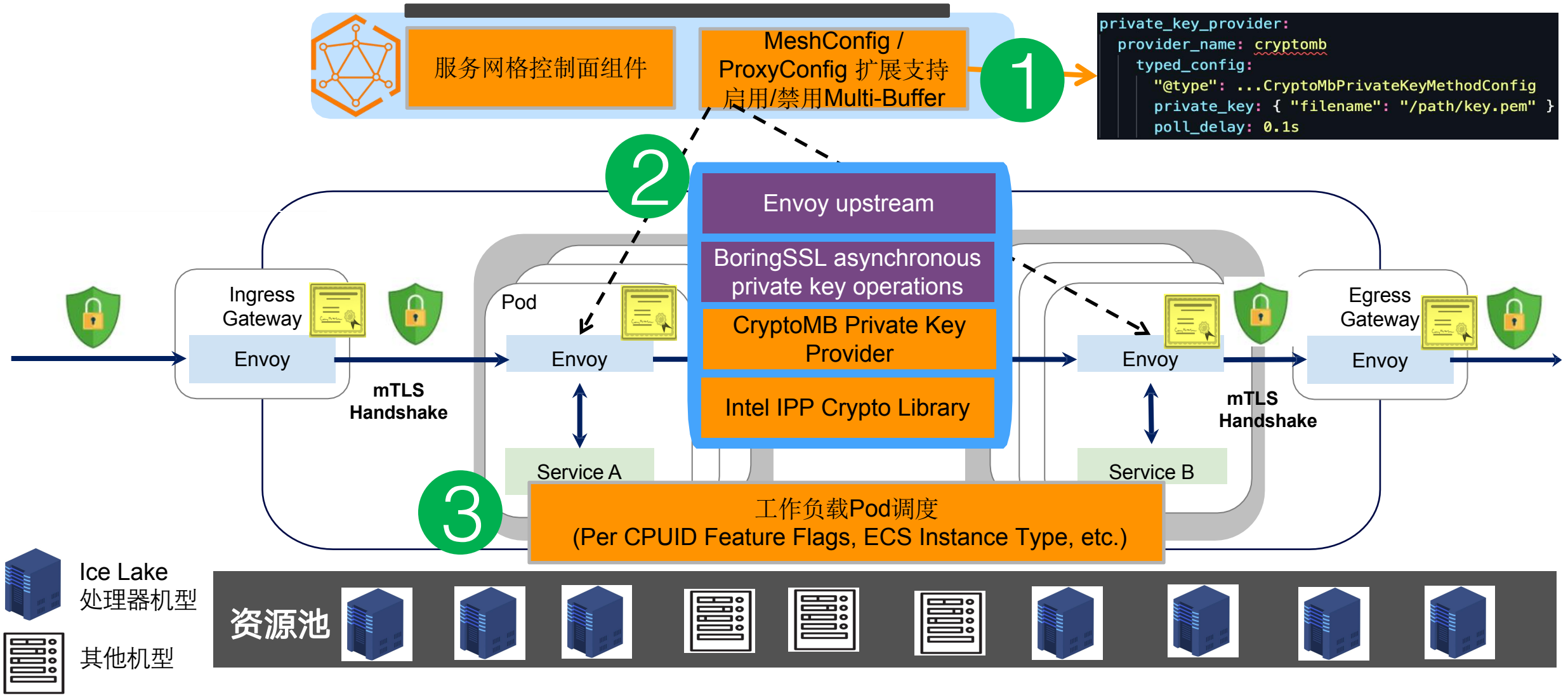
根据探测到的特征，自适应动态配置相应的特性，用户无感知；例如根据是否支持AVX指令集动态启用Multi-Buffer或QAT特性，提升TLS加解密性能；

技术白皮书https://developer.aliyun.com/ebook/7817

IV. 软硬结合性能优化的持续推进

阿里云 | 云服务商

✓ 结合Intel处理器指令集来优化和加速服务间的安全通信能力; 业界首个基于Intel Multi-Buffer技术提升TLS加解密的服务网格平台， 并服务于多个客户案例；并同Intel一起发布相关的技术白皮书。



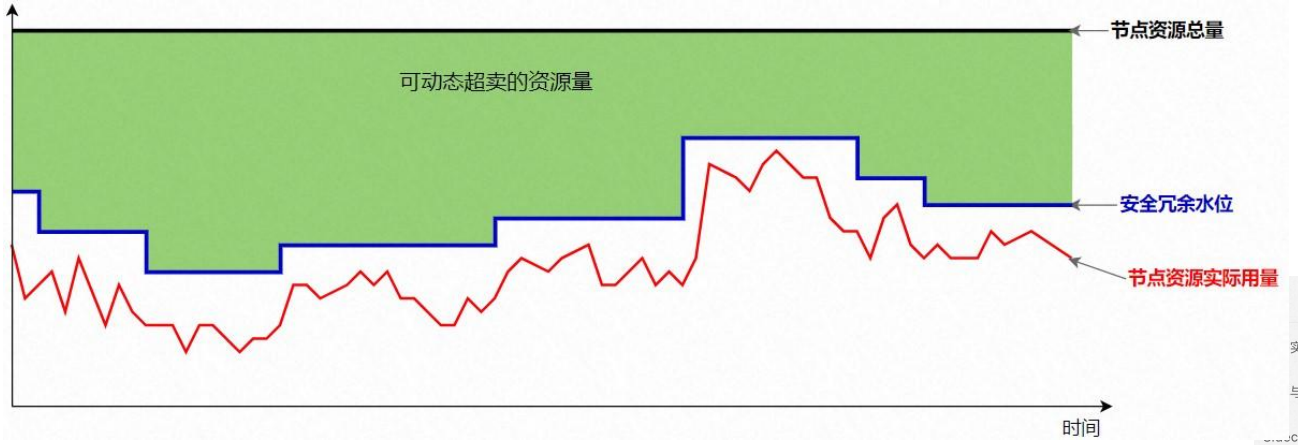
KUBERNETES
COMMUNITY DAYS | BEIJING

V. 资源超卖模式下的支持

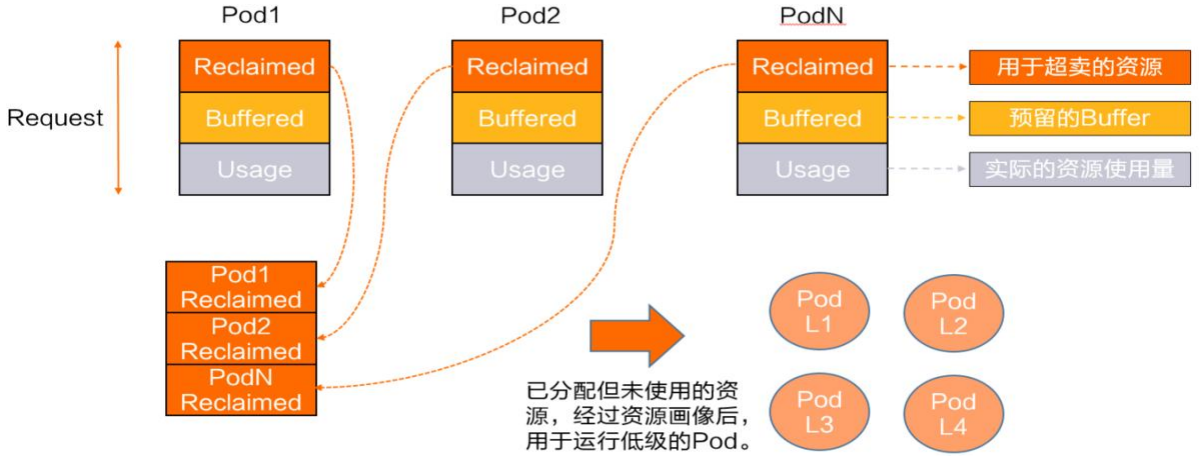
V. 资源超卖模式下的支持

阿里云 | 奥运会全球指定云服务商

Koordinator： 基于 QoS 的 Kubernetes 混合工作负载调度系统



- 一般不强制资源限制和所需资源配置为相同值。
- 若QoS为Guaranteed类型的工作负载，建议您两者配置为相同值。
- 若为其他类型的Pod，建议您保持与原生资源类型中所需资源小于资源限制的约定。



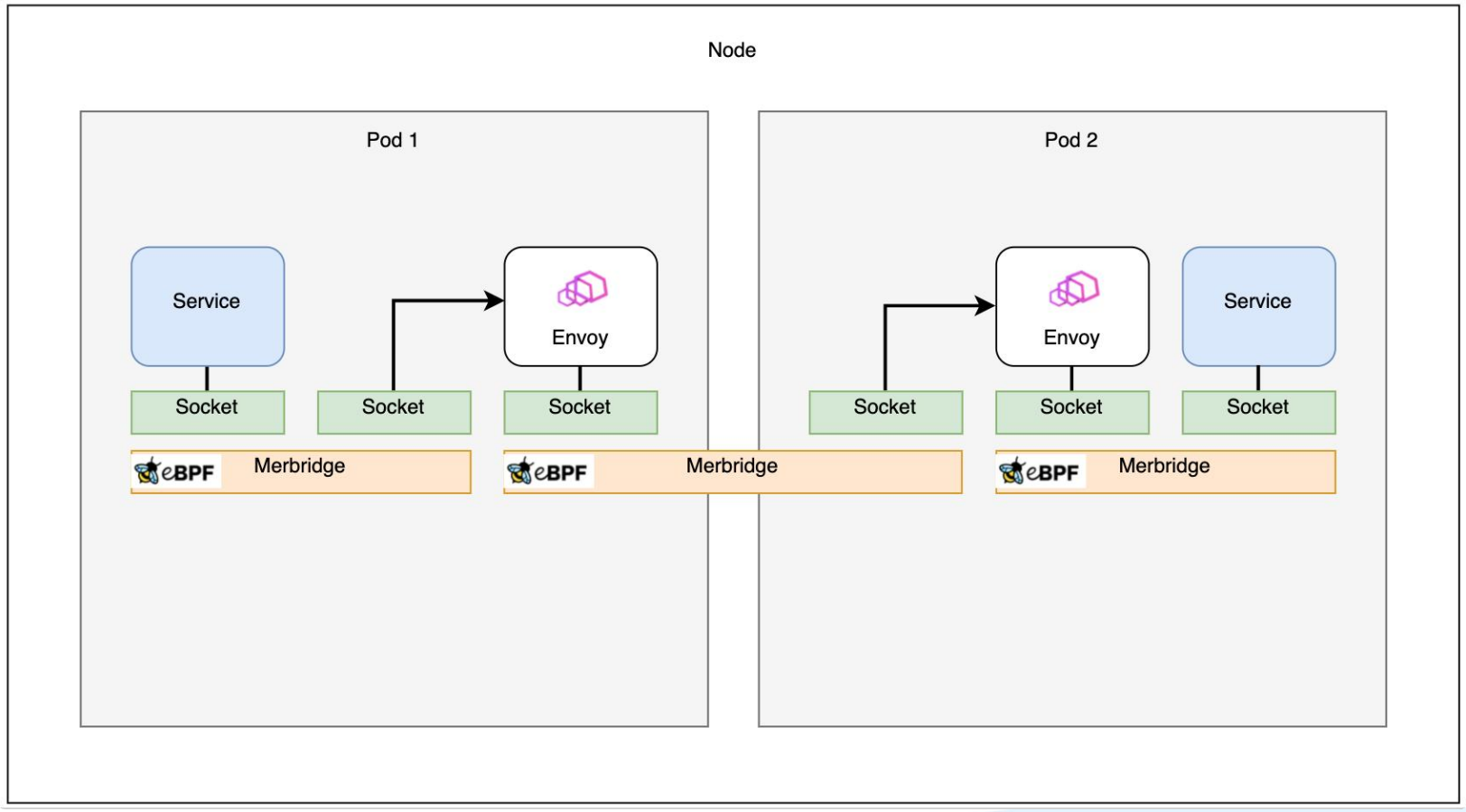
KUBERNETES
COMMUNITY DAYS | BEIJING

VI. MerBridge：基于 eBPF tcpip-bypass 的数据面性能优化

Merbridge 是什么？

Merbridge 在服务网格中使用 eBPF 技术代替 iptables，实现流量拦截。

借助 eBPF 和 msg_redirect 技术，Merbridge 可以提高 Sidecar 和应用之间的传输速度，降低延迟。



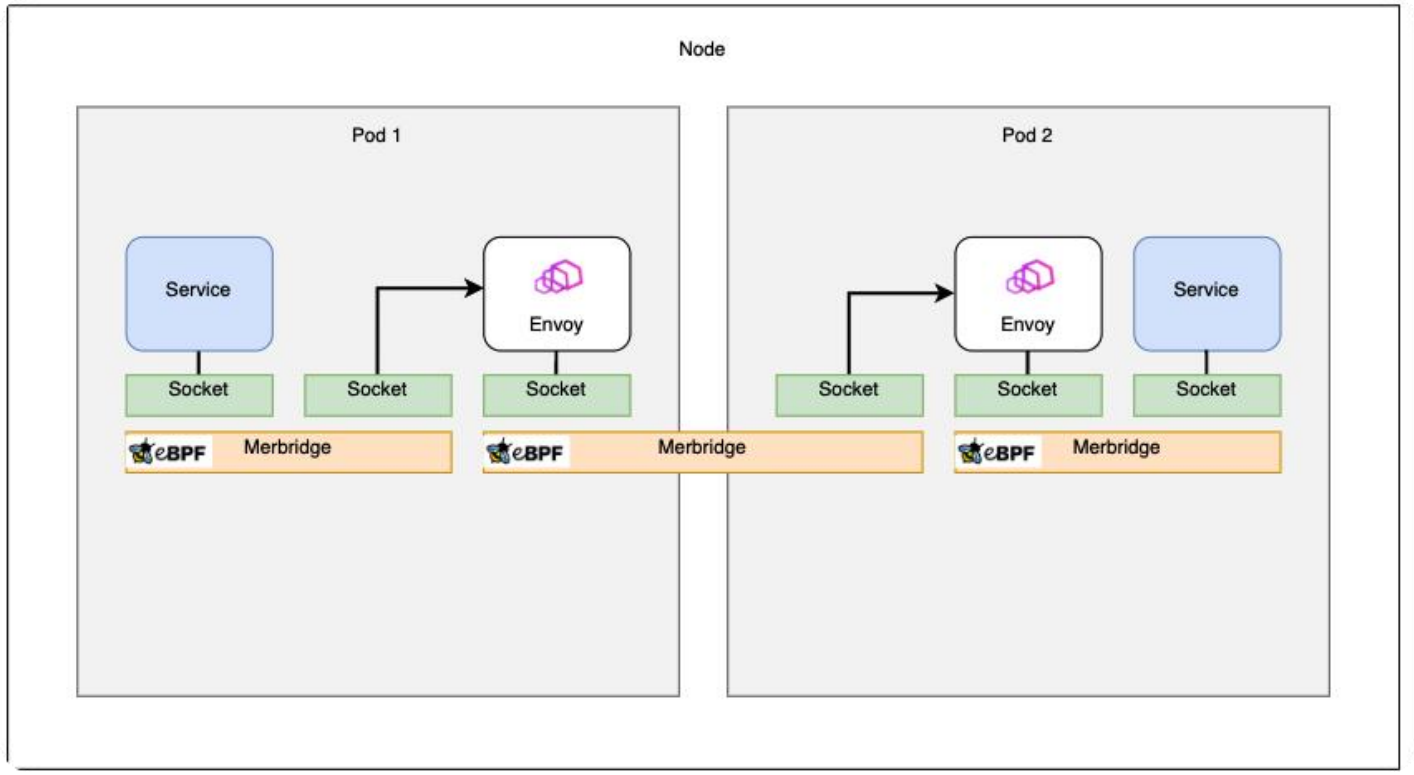
CLOUD NATIVE COMPUTING FOUNDATION

如何加速？

eBPF 提供重写 tcp_sendmsg 函数的能力，让 eBPF Program 接管 tcp_sendmsg。

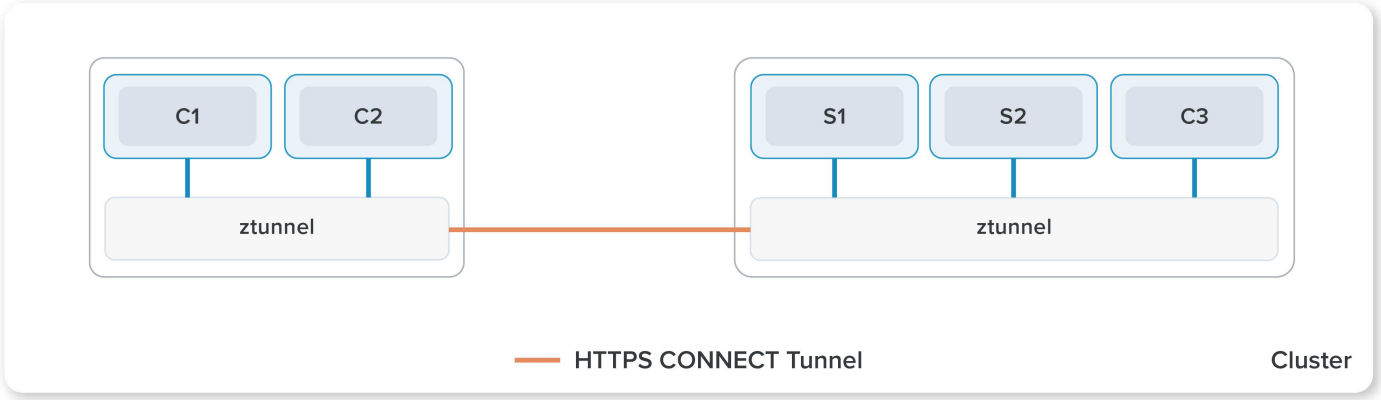
同时，eBPF 提供了 bpf_msg_redirect_hash 相关的 helper 函数，可以在被接管的 tcp_sendmsg 内将主机上的两个 socket 传输路径进行短接，绕过内核态协议栈，从而加速进程间的访问。

Helper 函数依赖一个内核级别的 map 保存连接信息，需要确保每个连接在 map 中的 key 互不冲突。



实现原理

Ambient 模式支持难题



- Ambient 模式下，加入/移除网格将更加灵活，CNI 模式将不再适用（CNI 只会在 Pod 创建的时候生效，Ambient 模式允许通过修改 annotation 的方式将 Pod 纳入网格）。
- Container 出口流量拦截的地址将不再是 127.0.0.1:15001，而需要转发到当前节点 ztunnel 实例。
- 由于 Pod 中不存在 Sidecar，之前通过判断是否监听 15001 端口等方案来确定是否需要拦截的方案将不再生效。

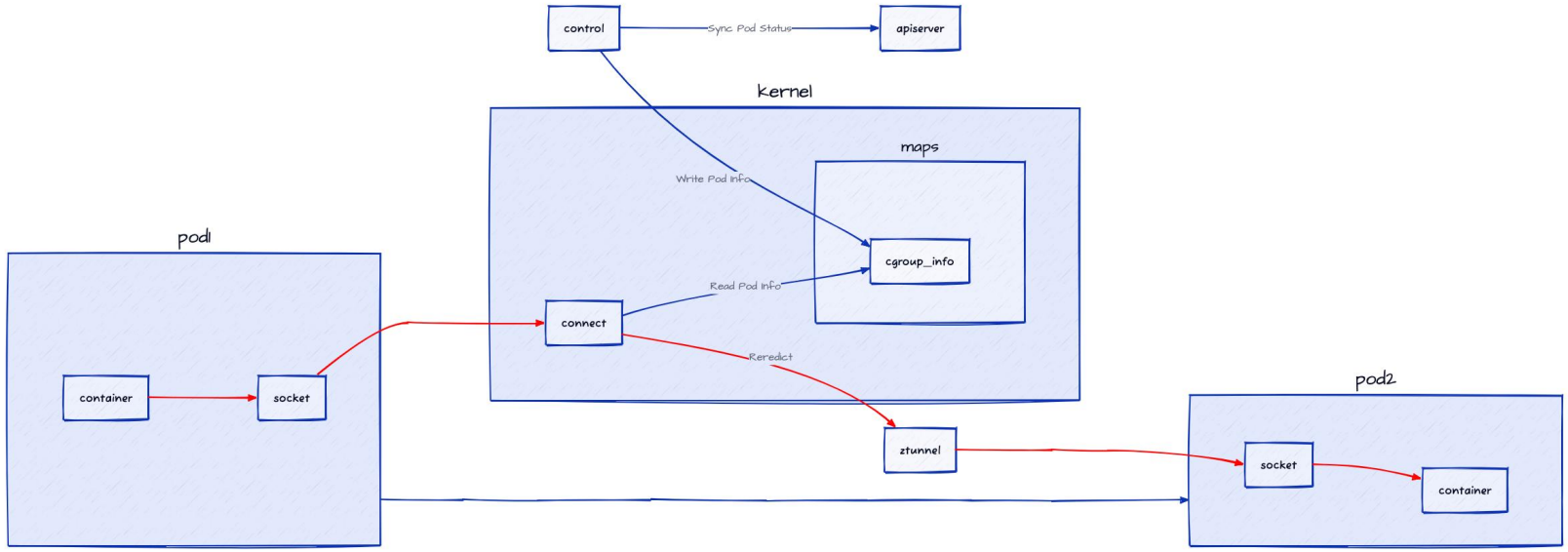
实现原理

Ambient 模式支持方案

1. 通过 eBPF 观察进程创建事件，在用户态关联 cgroup id 和 Pod IP。同时，将 Pod 信息等进行储存。

2. 在 eBPF 程序中，通过当前进程的 cgroup id 查找当前进程的 Pod IP。（一个容器中的进程应该具有相同的 cgroup id，且不会变更）

3. 如果是 Ambient 模式的 Pod 发出的流量，将流量转发到 ztunnel。

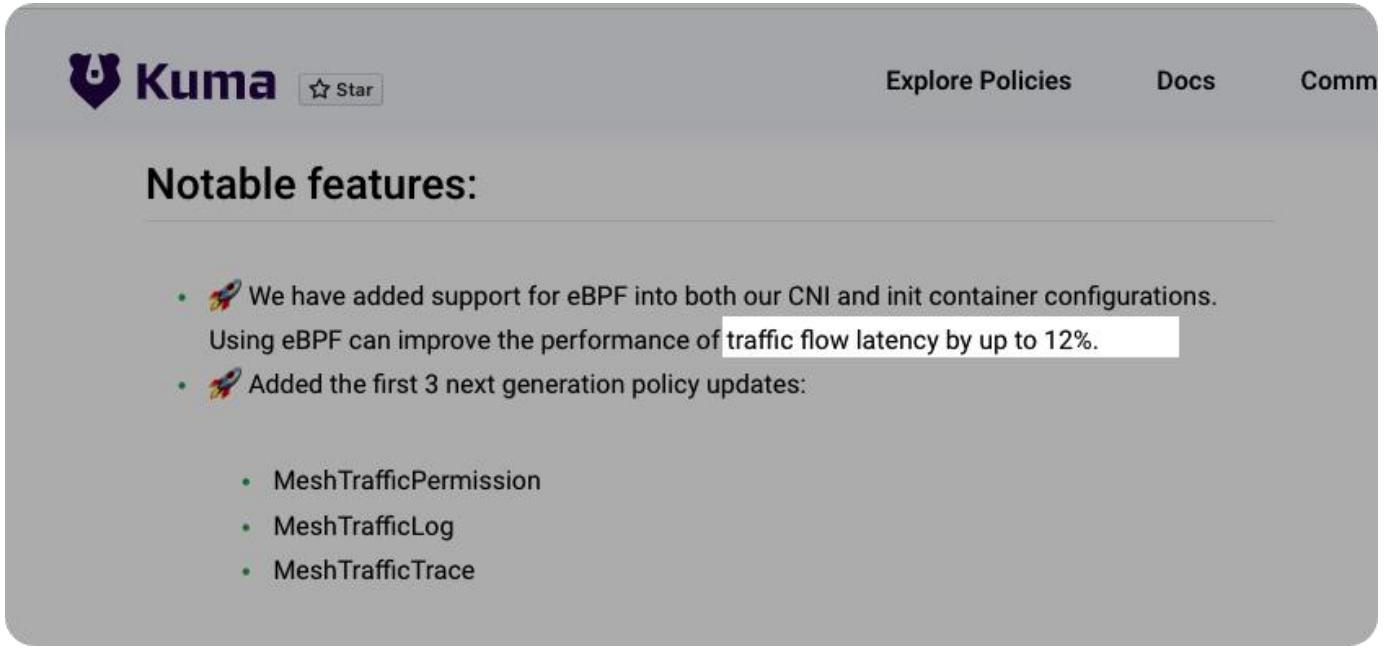


实现效果

一般情况下，可以实现 5-12% 的延迟提升。

可应对高并发场景。

无其它性能损耗或要求。



未来展望

持续性能测试；

更低的内核版本要求；

跨节点加速；

双栈支持；

Ambient 生产就绪；

……

KUBERNETES
COMMUNITY DAYS | BEIJING

Thanks.