KUBERNETES COMMUNITY DAYS BEIJING 2023



使用 eBPF 实现多集群的流量调度

张晓辉@Flomesh





张晓辉

Flomesh 高级云原生工程师

资深程序员,LFAPAC 开源布道师,CNCF Ambassador,云原生社区管委会成员,微软 MVP,公众号"云原生指北"作者。

有多年的微服务和云原生实践经验,主要工作涉及微服务、容器、Kubernetes、DevOps 等。



Content

目录

01 eBPF 简介

02 eBPF 的应用场景

03 云原生网络流量

04 eBPF 助力跨集群流量调度



Part 01 eBPF 简介

BPF

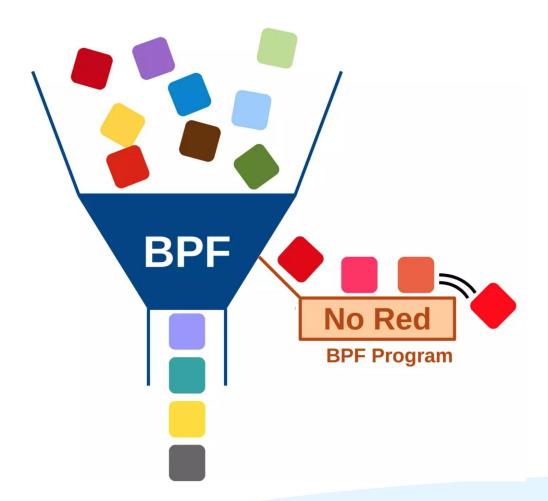


Berkeley Packet Filter

来自于 1992 年的论文<u>《The BSD Packet Filter:</u>

A New Architecture for User-level Packet Capture》

发明之初是用做网络包的过滤器, tcpdump。



eBPF



eBPF = extended Berkeley Packet Filter

Dynamically program the kernel for efficient networking, observability, tracing, and security.

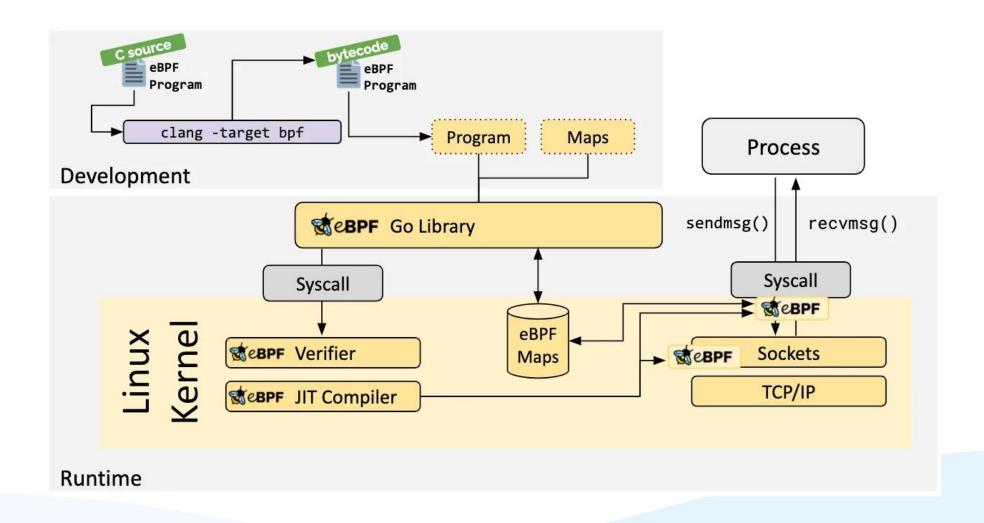
- 稳定(DAG、可达性)
- 高效(JIT 本地机器码)
- 安全 (verifier, 有限的辅助函数)
- 热加载/卸载(无需重启)



内核可编程

eBPF 加载器与验证器





eBPF 事件驱动



Event -> Action

事件:

- Kprobe/Kretprobe(Kernel函数入口和出口)
- Uprobe/Uretprobe(User 函数入口和出口)
- XDP (eXpress Data Path)
- Tracepoint (特定事件时触发)
- Perf(性能事件,如 CPU 周期计数)

bpftrace Probe Types hardware: Dynamic Tracing tracepoint: Static Tracing syscalls ext4 sock sched cpu-cycles Operating System task instructions signal branch-* uprobe: **Applications** timer frontend-* uretprobe: workqueue backend-* System Librarie usdt: **CPU** System Call Interface Interconnect **VFS** Sockets CPU bus Scheduler I kprobe: **▼** File Systems TCP/UDP kmem kretprobe: Volume Manager IP Virtual 4 vmscan Memory Memory writeback Block Device Interface Bus Ethernet , **Device Drivers DRAM** irq jbd2 net skb block scsi cache-* profile: BEGIN software: page-faults cpu-clock interval: cs migrations minor-faults END major-faults **Timed Events** Special Events https://github.com/iovisor/bpftrace 2018



Part 02 eBPF 的应用场景

eBPF 应用场景



eBPF 的可编程性使其能够在不离开 Linux 内核的包处理上下文的情况下,添加额外的协议解析器,并轻松编程任何转发逻辑以满足不断变化的需求。JIT 编译器提供的效率使其执行性能接近于本地编译的内核代码。

网络

将 eBPF 程序附加到跟踪点以 及内核和用户应用探针点的能 力,使得应用程序和系统本身 的运行时行为具有前所未有的 可见性。eBPF 不依赖于操作 系统暴露的静态计数器和测量, 而是实现了自定义指标的收 集和内核内聚合,并基于广泛 的可能来源生成可见性事件。

观测和跟踪

看到和理解所有系统调用的基础上,将其与所有网络操作的数据包和套接字级视图相结合,可以采用革命性的新方法来确保系统的安全。通过检测和阻止恶意行为,如 DDoS攻击、网络钓鱼等,实施网络策略、可以增强系统的安全性和稳定性。

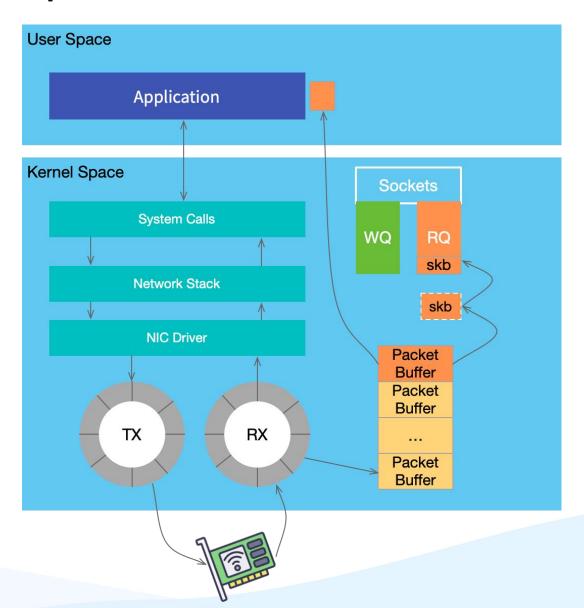
安全



Part 03 云原生网络流量

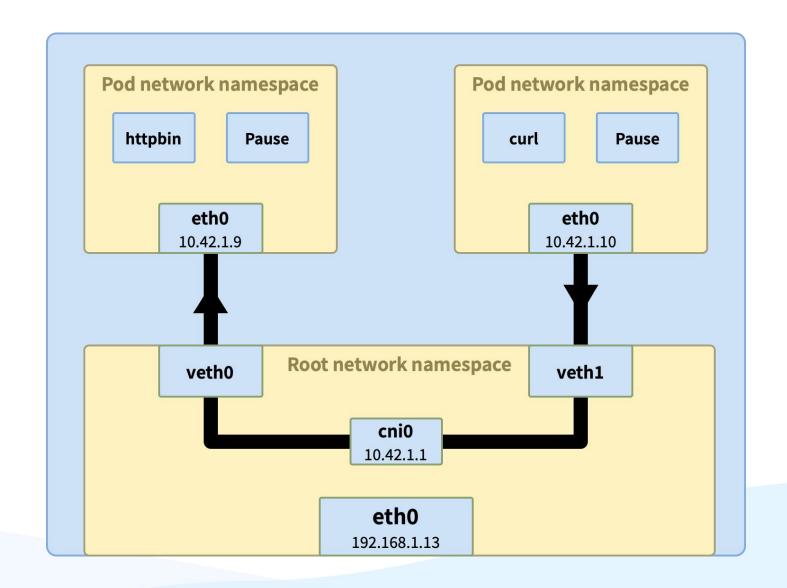
Linux Kernel Datapath





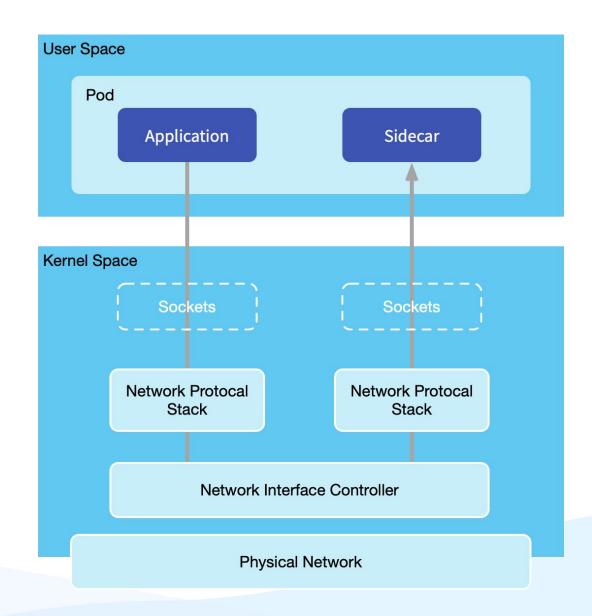
云原生容器网络





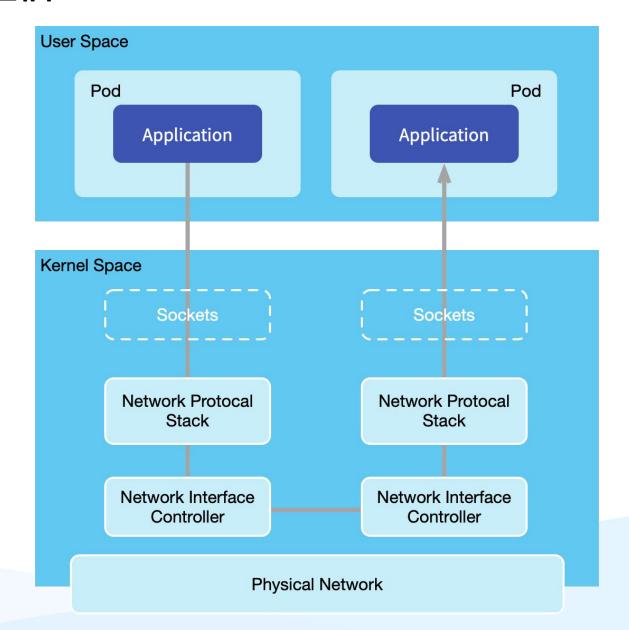
同 Pod 的网络通信





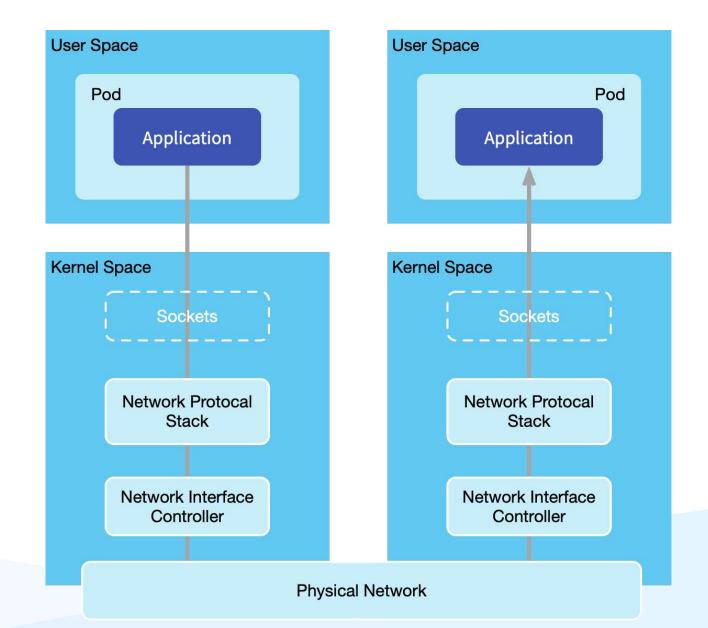
同节点的网络通信





跨节点的网络通信

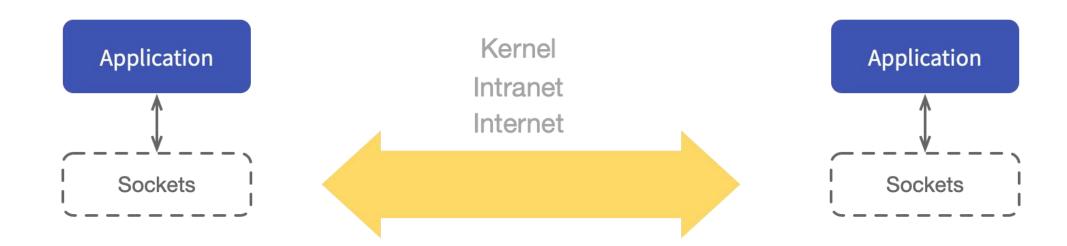




连接建立

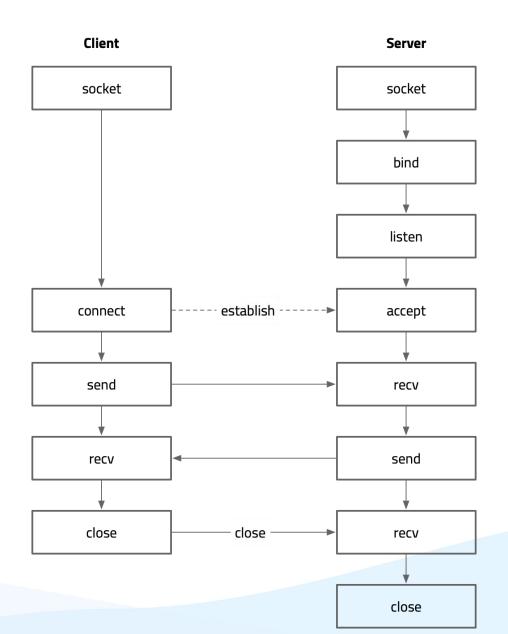


连接建立是一个 socket 寻址 过程



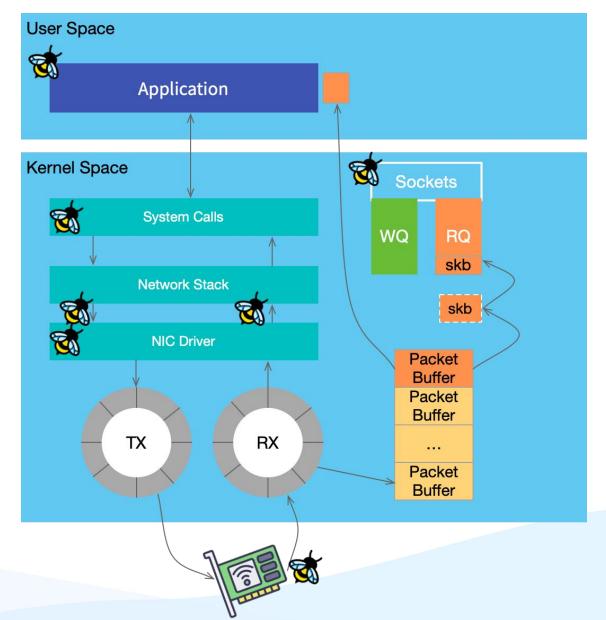
Socket 操作





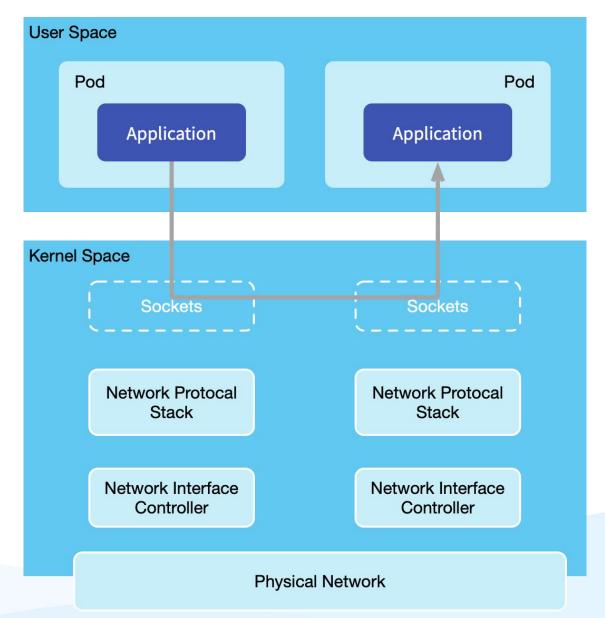
eBPF in Datapath





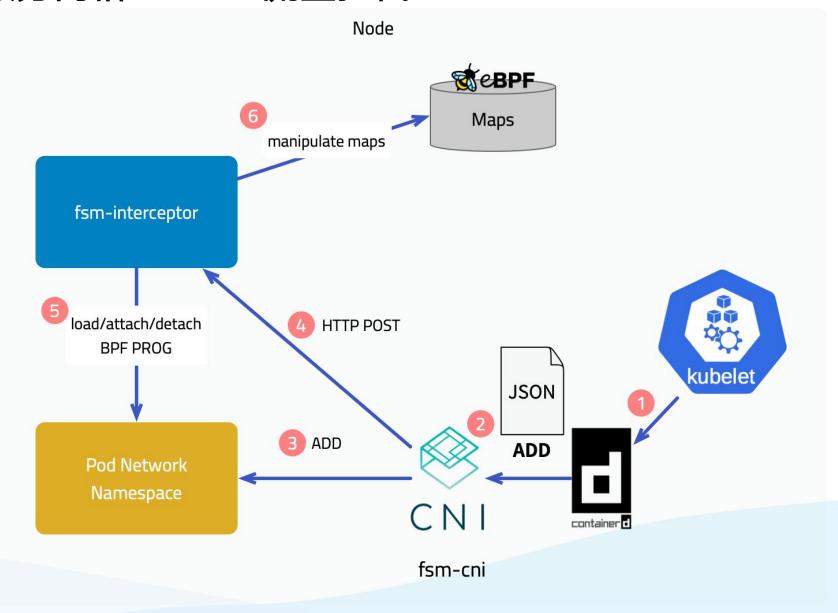
Bypass 网络协议栈





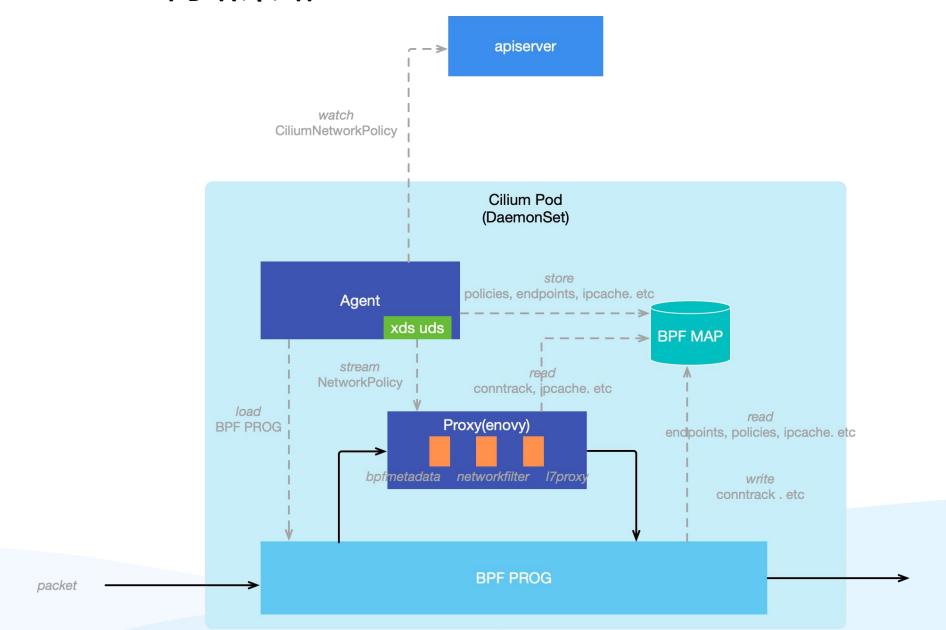
Flomesh 服务网格: FSM 流量拦截





Cilium L3/4/7 网络策略



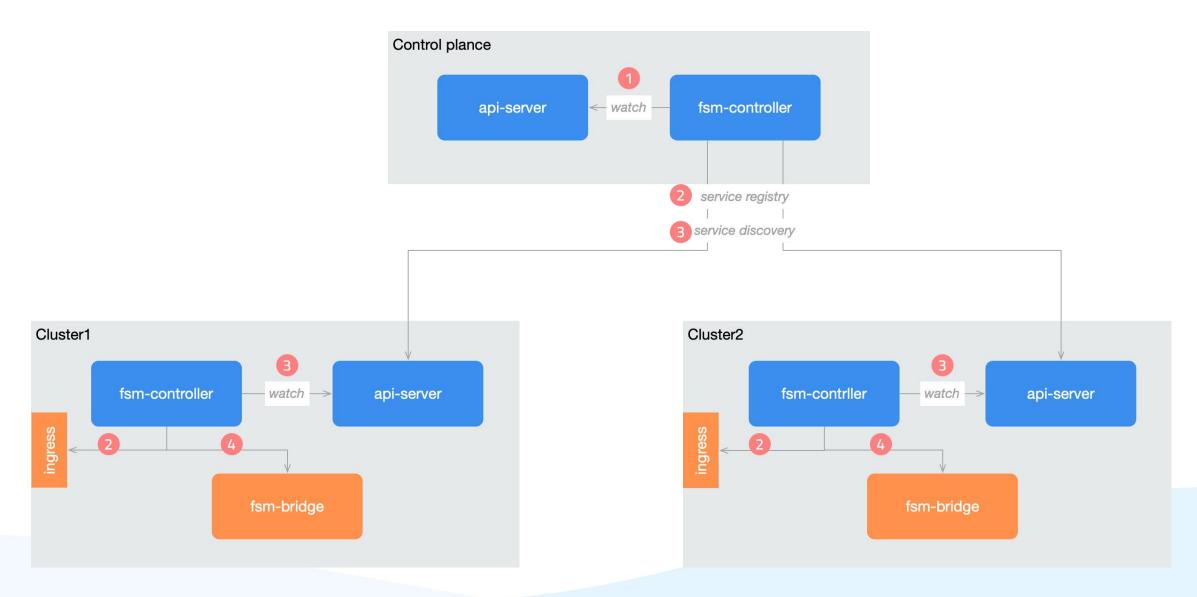




Part 04 eBPF 助力跨集群流量调度

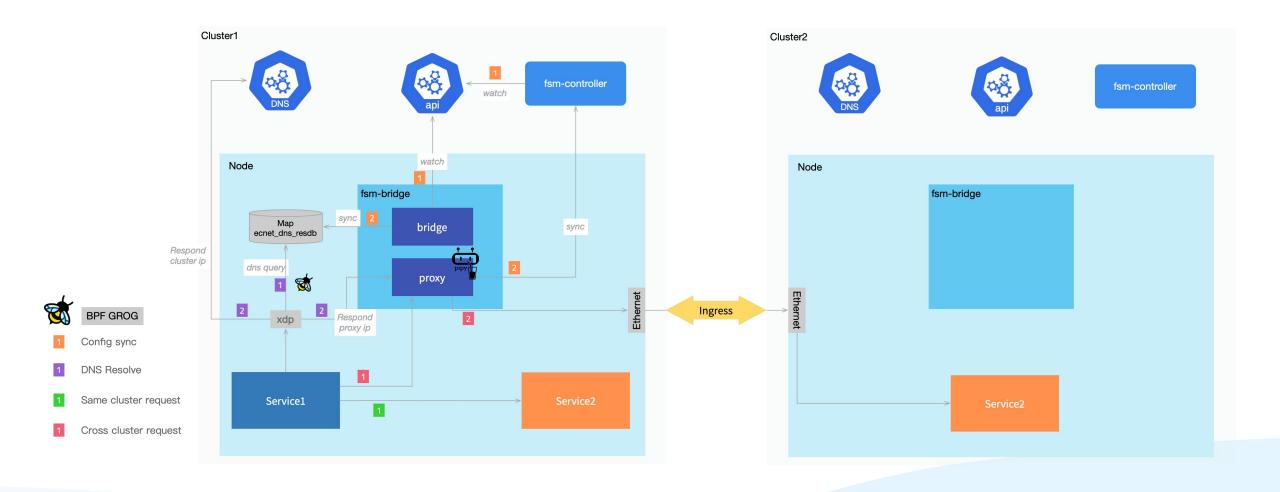
FSM 多集群服务注册发现











展望:可编程



灵活、高效

API **User Space** 代理可编程 **Kernel Space** 内核可编程 Helper Call

https://github.com/flomesh-io/pipy



Thanks.

关注我们





flomesh.io



github.com/flomesh-io



flomesh-io.slack.com

