Envoy Intro

Lizan Zhou
06-25-2019
State of microservice networking in industry

- **Languages** and frameworks.
- **Protocols** (HTTP/1, HTTP/2, gRPC, databases, caching, etc.).
- **Infrastructures** (IaaS, CaaS, on premise, etc.).
- Intermediate **load balancers** (AWS ELB, F5, etc.).
- **Observability** output (stats, tracing, and logging).
- Implementations (often partial) of **retry**, **circuit breaking**, **rate limiting**, **timeouts**, and other distributed systems best practices.
- **Authentication** and **Authorization**.
- Per language **libraries** for service calls.
State of microservice networking in industry

- Likely already in a **world of hurt** or rapidly approaching that point.
- **Debugging** is difficult or impossible (each application exposes different stats and logs with no tracing).
- **Limited visibility** into infra components such as hosted load balancers, databases, caches, network topologies, etc.
- Multiple and **partial implementations** of circuit breaking, retry, and rate limiting (If I had a $ for every time someone told me that retries are “easy” …).
- Furthermore, if you do have a good solution, you are likely using a **library** and are locked into a particular technology stack essentially forever.
- Libraries are incredibly **painful to upgrade**. (Think CVEs).
What is Envoy?

The network should be transparent to applications.

When network and application problems do occur it should be easy to determine the source of the problem.
Service mesh refresher

- Service A
  - Sidecar proxy
- Service B
  - Sidecar proxy
- Service C
  - Sidecar proxy
- Service D
  - Sidecar proxy
Envoy design goals

- Out of process architecture
- High performance / low latency code base
- L3/L4 filter architecture
- HTTP L7 filter architecture
- HTTP/2 first
- Service discovery and active/passive health checking
- Advanced load balancing
- Best in class observability (stats, logging, and tracing)
- Authentication and authorization
- Edge proxy
Envoy config management via xDS APIs

- Envoy is a **universal data plane**
- xDS == * Discovery Service (various configuration APIs). E.g.,:
  - LDS == Listener Discovery Service
  - CDS == Cluster Discovery Service
- Both gRPC streaming and JSON/YAML REST via proto3!
- Central management system can control a fleet of Envoys **avoiding per-proxy config file hell**
- **Global bootstrap config** for every Envoy, rest taken care of by the management server
- Envoys + xDS + management system == **fleet wide traffic management distributed system**
Envoy architecture

- L4 Read Filters
- L4 Write Filters
- HTTP/L7 codec
- HTTP/L7 conn manager
- HTTP/L7 read filters
- HTTP/L7 write filters
- Service Router
- Upstream conn pool
- Backend services
- Stats
- Admin
- Cluster/Listener/Route/Secret Manager
- xDS API
- Filter Chains
- Connection
- Listener filters
- Worker
Extension and pluggability

Envoy is designed to have multiple extension point. E.g.:

- L4/L7 filters
- Access loggers
- Tracers
- Health checkers
- Transport sockets
- Retry policy
- Resource monitors
- Stats sink
### Envoy Adoption

- Airbnb
- Amazon Web Services
- Booking.com
- Cookpad
- DigitalOcean
- eBay
- Figma
- Google
- Gojek
- GrubHub
- IBM
- Medium
- Microsoft
- Netflix
- Pinterest
- Skype
- Stripe
- Square
- Tencent
- Twilio
- Uber
- Verizon
- VMware
- Yahoo Japan
- Yelp
Adoption in China

NetEase, Inc.
Envoy 在网易内部的实践

曾宇星 网易资深架构师
目录

01  Envoy 在内部微服务框架中的应用适配

02  扩展Envoy 实现微服务多环境治理
现状

• 以Nginx作为流量出口代理
• 与Consul紧密结合
• 服务发现基本实现业务无感知

挑战

• 数据面能力弱, 无控制面
• 维护成本高
• 开源社区偏离
• 整体基于Envoy+Istio方案
• Outbound指向Sidecar
• Inbound可配置流量拦截
• 控制面以Pilot为核心
• 注册中心以K8S原生方式
• 通过MCP机制扩展
适配原有服务调用配置

127.0.0.1:8550/proxy/servicea/path/xxx
127.0.0.1:8550/hash/serviceb/path/xxx

proxy 和 hash 为 prefix，可配置

Istio 适配

默认下发8550的LDS至Envoy，其中route_config_name 配置为80

Envoy适配

在http_connection_manager 下增加url_transformer 配置选项，针对类似URL 127.0.0.1:8550/proxy/servicea/path/xxx更改http请求的host 和 path 分别为 “servicea” 和 “path/xxx”
Envoy XDS

LDS:
listeners

RDS:
routes

CDS:
clusters

EDS:
endpoints

Rewrite hostname and path

127.0.0.1:8550/proxy/servicea/path/xxx
需求:

- 流量染色
- 流量穿梭
- 业务无感知
控制面通过 sourceLabels 下发定向路由，环境较多时配置太麻烦，而且不能解决 subset 不存在时 cluster 降级问题

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
    - ratings
  http:
    - match:
        - sourceLabels:
            color: red
          route:
            - destination:
                host: ratings
                subset: red

apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: ratings
spec:
  host: ratings
  subsets:
    - name: red
      labels:
        color: red
    - name: green
      labels:
        color: green
```
- 支持L4&L7
- 支持流量染色
- 支持流量穿梭
- 应用无侵入
// get the downstream color
auto downstream_color = downstreamConn->getPreferClusterColor();
auto listener_config_color = transport_socket_options->getDefualtDownStreamColor();

if (!downstream_color.empty()) {
    send_color = downstream_color.data();
    ENAVY_LOG(debug, "using downstream connection color: ", downstream_color);
} else if (!listener_config_color.empty()) {
    send_color = listener_config_color;
    ENAVY_LOG(debug, "using listener config default color: ", send_color);
}

if (!send_color.empty()) {
    // add extention color property
    // defined in <type>
    proxy_data->tlv.type = 0x39; // PP2_TYPE_NETNS;

    len = strlen(send_color.data());
    len = std::min(len, 15);
    proxy_data->tlv.length = htons(len);

    strncpy(reinterpret_cast<char*>(proxy_data->tlv.value), send_color.data(), len);

    proxy_data->length = len + PP2_TLV_HEADER_SIZE;

    uint8_t* proxy_data_buff = reinterpret_cast<uint8_t*>(proxy_data->get());
    //
    // memmove(proxy_data_buff[8], &proxy_data->tlv, sizeof(Network::ProxyProtocol::pp2_tlv));
    //
    else {
        ENAVY_LOG(debug, "proxy protocol: not color data -");
    }

**Sidecar 通过 header (x-prefer-color) 、Proxy protocol 协议的扩展字段传递 color 属性**
注意：类似分布式追踪应用回传x-request-id相关头，流量穿梭功能需要业务应用回传x-prefer-color头
Q&A

- Thanks for coming!
- [https://envoyproxy.io/](https://envoyproxy.io/)
- Talk to us if you need help getting started with Envoy.
- **Tetrate is hiring**: Contact us if you want to work on the problems and solutions based on Service Mesh.