基于硬件卸载的云原生网关连接平衡实现

戴翔 (Intel 云原生工程师)

戴翔 Loong

Intel Cloud Software Engineer

从事云原生行业多年，深耕开源

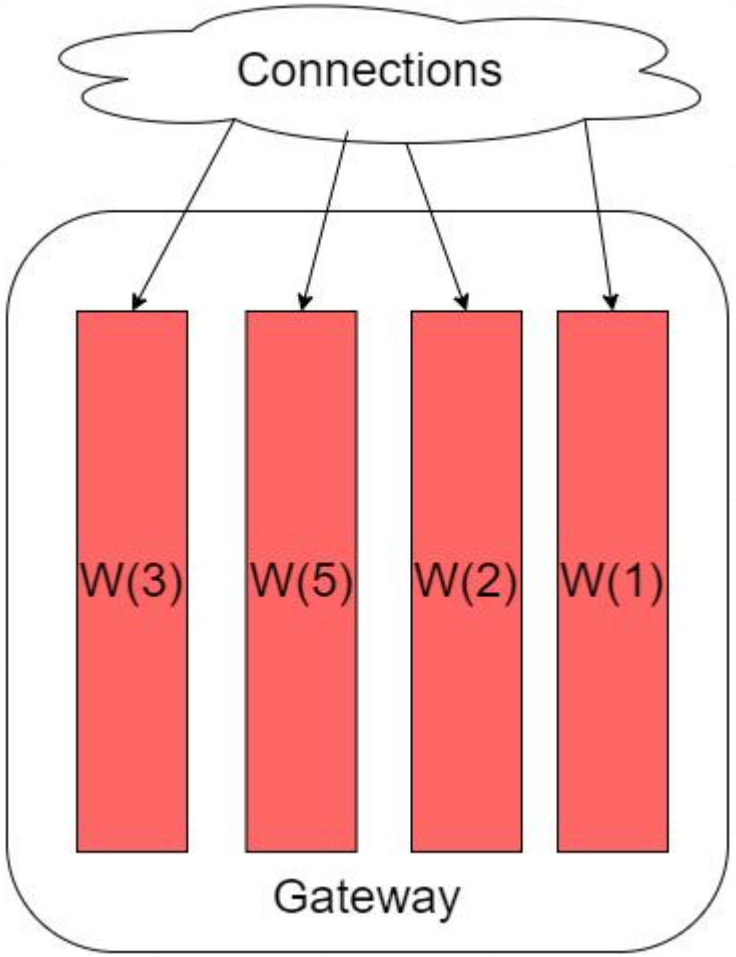
Dapr/Thanos/Golangci-lint Maintainer

目前专注于服务网格领域

云原生网关的连接均衡

- 负载均衡是一种核心的网络解决方案，用于在服务器场中的多个服务器之间分配流量。负载均衡器提高了应用程序的可用性和响应能力，并防止服务器过载。每个负载均衡器位于客户端设备和后端服务器之间，接收传入请求，然后将其分发到任何能够满足它们的可用服务器。
- 一个常见的网关通常有多个worker（进程或线程）。如果多个客户端同时连接到同一个worker，这个worker会变得忙碌，带来很大的尾部延迟。此时其他worker会处在相对空闲的状态，影响Web服务器的整体性能。连接负载均衡器就是针对这种情况的解决方案
- 接下来以 Envoy 为例，介绍网关时如何接收连接并进行连接均衡，以及基于硬件卸载的实现。

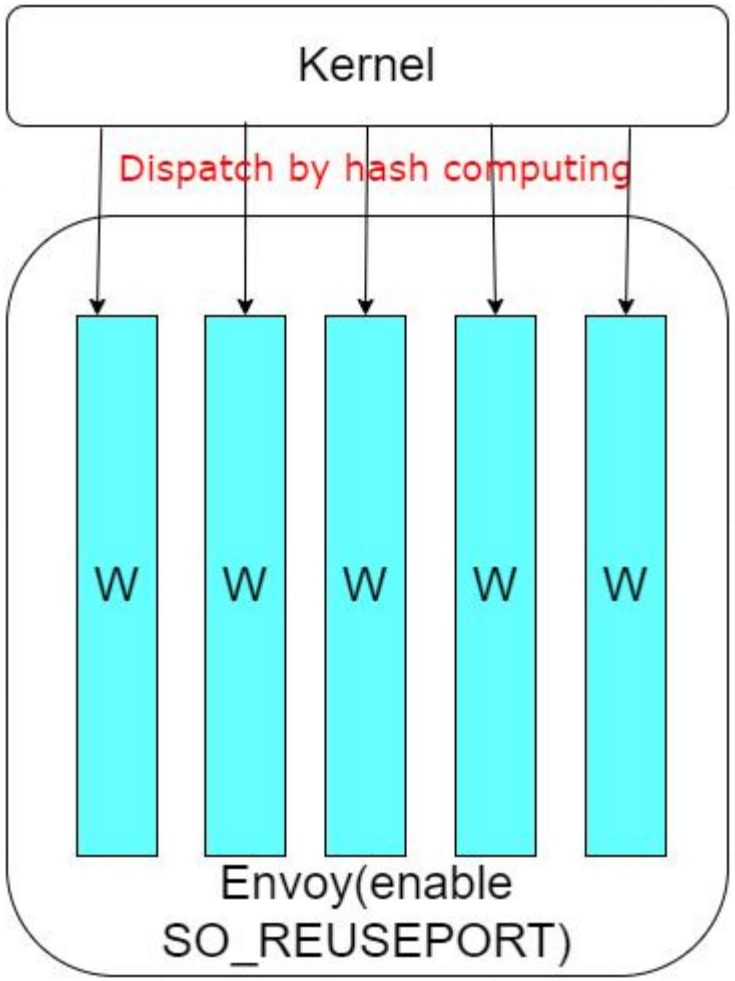
抽象模型



Envoy的连接处理

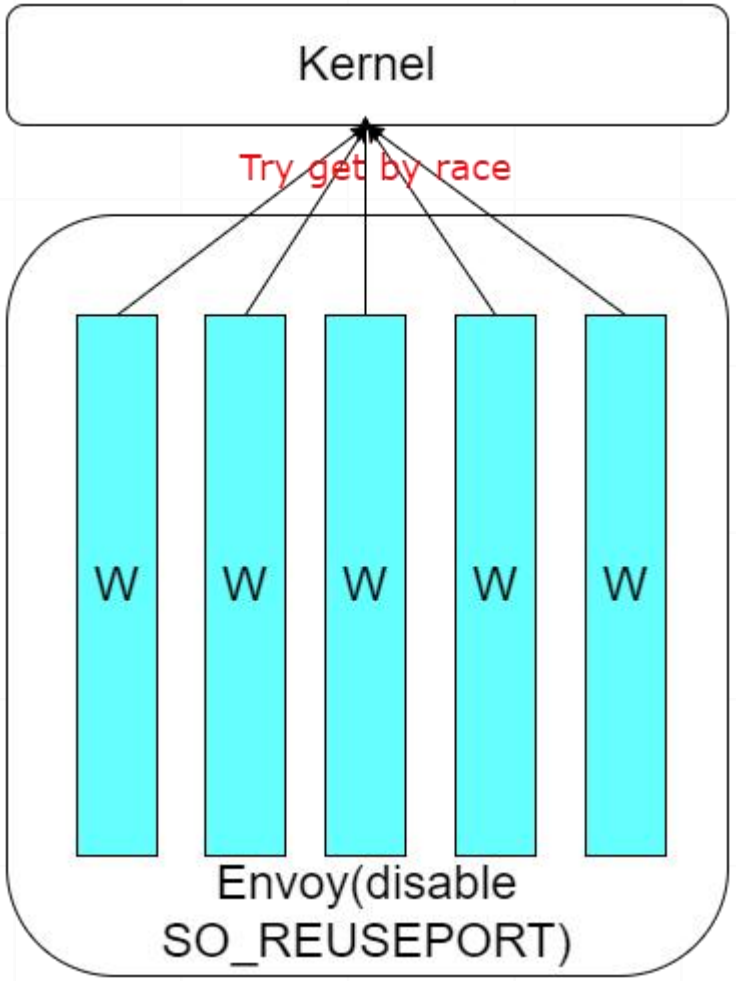
当开启SO_REUSEPORT（默认开启）时，Kernel会基于四元组信息（源地址、源IP、目标地址、目标IP）进行hash计算，然后将连接分配到具体的worker。

很明显，当某个客户端发出了对同一资源的大量请求时，这些请求会堆积在某个worker上，此时其他worker的相对空闲，造成很大的尾部延迟和资源浪费。



Envoy的连接处理

当关闭SO_REUSEPORT时，Envoy的每个worker会以竞争的方式请求接受连接。



Envoy的连接均衡策略

Envoy有两种连接均衡策略：

- Nop Connection Balance
- Exact Connection Balance

Nop Connection Balance实际上不做任何处理，是Envoy的默认策略。

例外：

Windows平台存在多worker无法正常工作问题，只有开启Exact Connection Balance才能让所有worker工作。

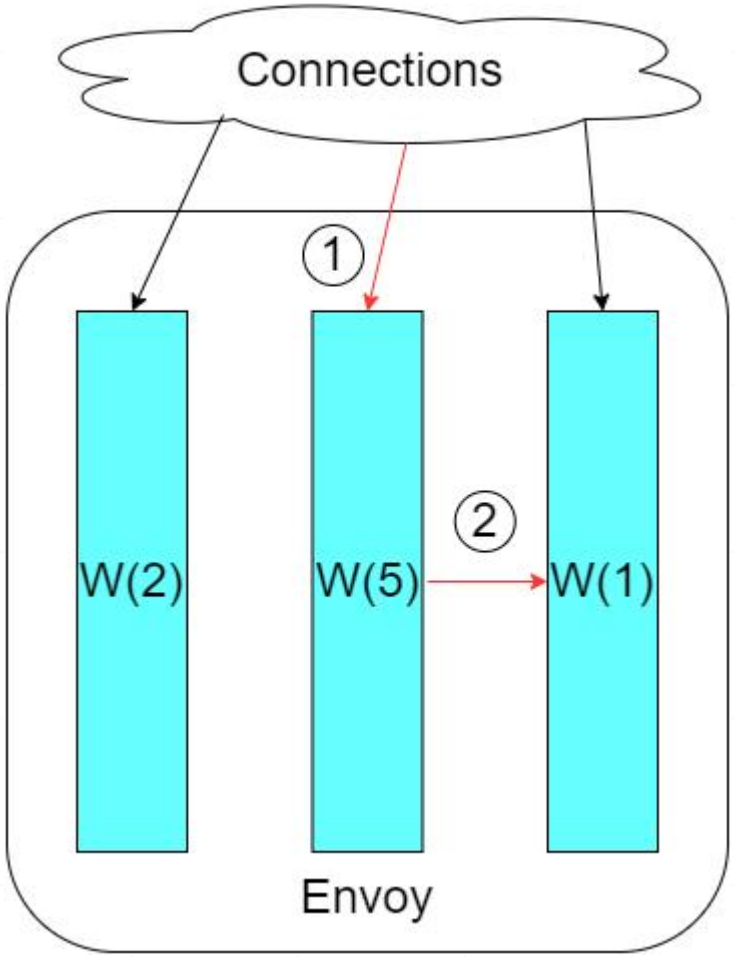
Envoy的精确连接均衡策略

精确连接均衡策略，顾名思义，尽可能让每个worker都处理相同数量的连接（**不区分长短连接**）。

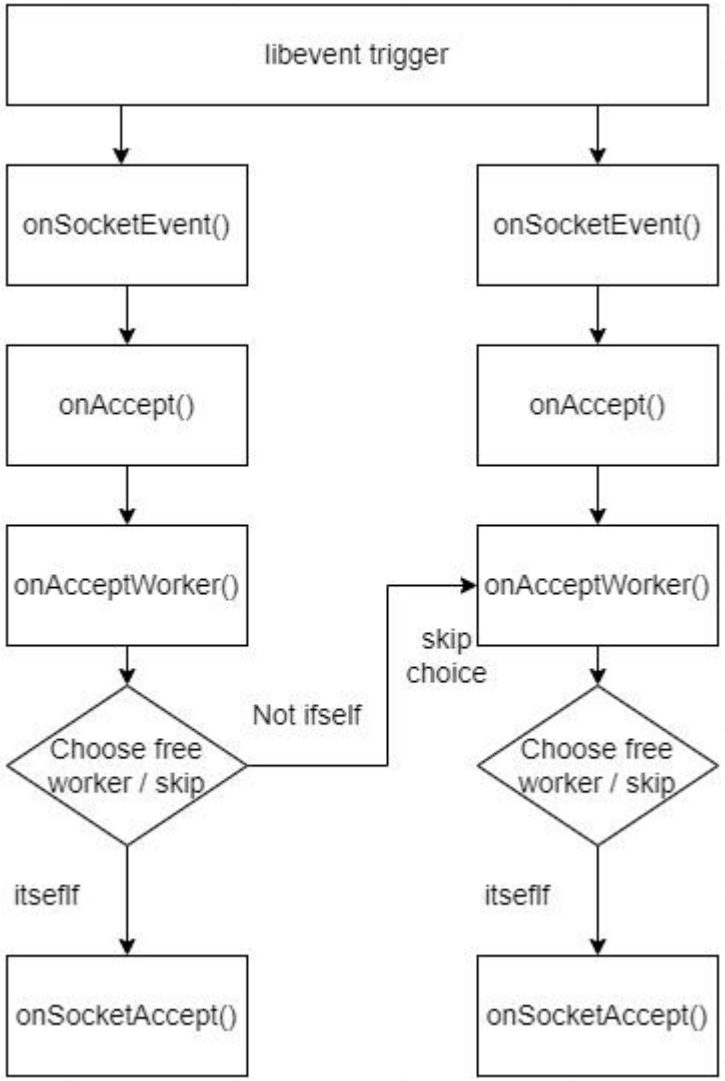
精确连接均衡主要分两步：

1. 接收连接，加锁，找出最空闲即连接数最少的worker。解锁。
2. 如果最空闲的worker不是自己，将连接交给最空闲的worker，让最空闲的worker继续处理连接。

此策略牺牲了吞吐量以获得准确性，适合在有少量很少循环的连接的场景，不适用于大连接场景如网关。



Envoy的精确连接均衡策略



基于硬件卸载的连接均衡

- 英特尔® 动态负载均衡器（英特尔® DLB）是一个由队列和仲裁器组成的硬件管理系统，连接生产者和消费者。它是一种 PCI 设备，存在于服务器 CPU 非核心中，可以与运行在核心上的软件交互，并可能与其他设备交互。
- 英特尔 DLB 实现了以下负载平衡功能：
  - 无锁多生产者/多消费者操作
  - 针对不同流量类型的多个优先级
  - 多种负载均衡队列类型

基于硬件卸载的连接均衡

负载均衡队列有 4 种类型：

- 直接：不进行任何负载均衡操作。
- 无序：将数据包分散到多个worker上，并且不保留顺序。
- 有序：与无序类似，只是系统提供了一种恢复原始流顺序的方法，但软件中可能仍需要同步机制。
- 原子：动态地将数据流固定到某个或某些worker，在需要时在多worker之间迁移流以实现负载平衡。这保留了数据流本身的顺序，在现代数据包处理设备（如网卡）中是非常需要的。

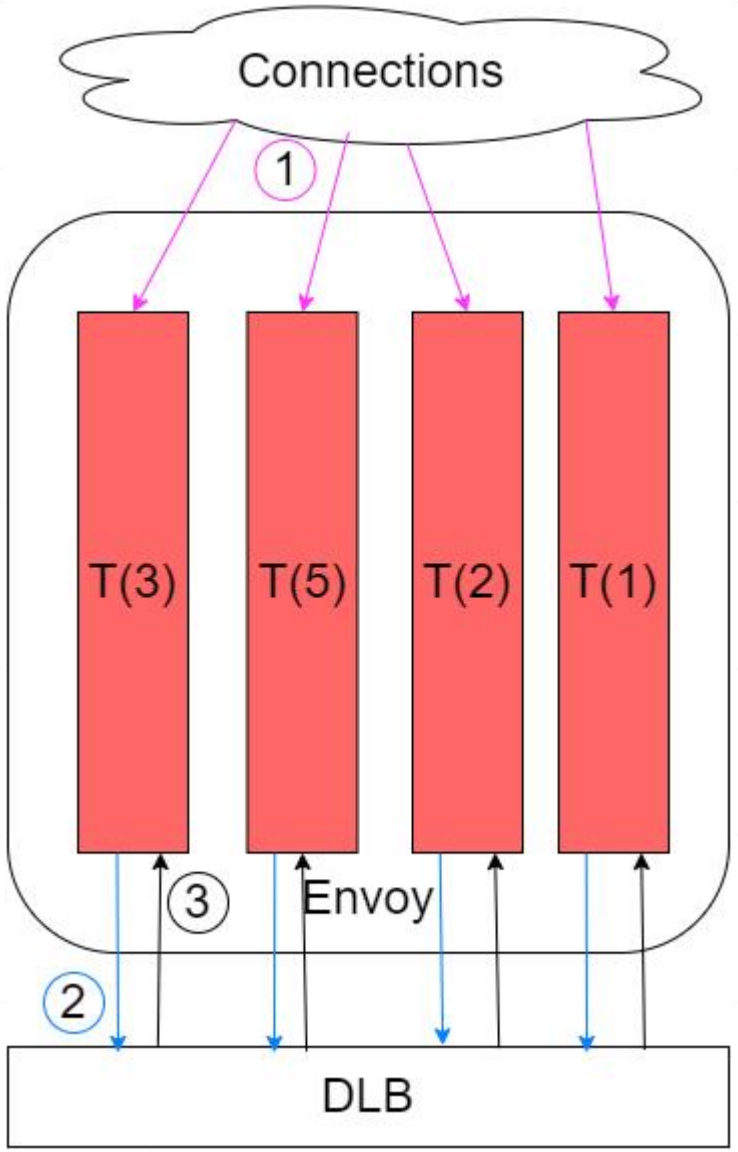
网关的设计初衷是尽可能快地处理尽可能多的数据，所以我们选择了无序队列。

DLB连接均衡策略实现

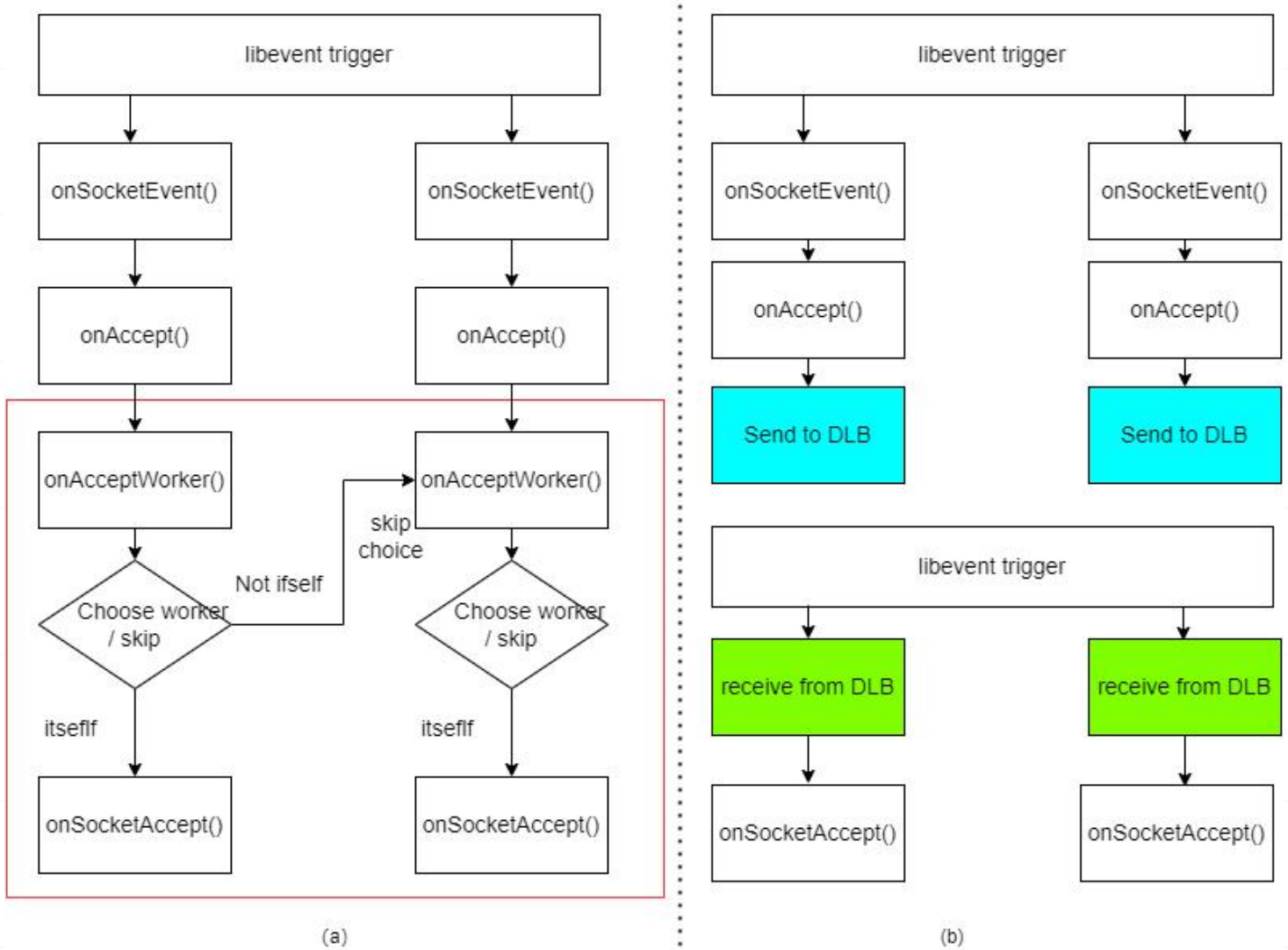
DLB连接均衡主要分两步：

1. 接收连接，将连接发送给DLB设备。
2. DLB设备通过eventfd通知具体的worker处理连接。

整个负载均衡过程完全卸载到硬件，避免了锁开销，减少了CPU资源的消耗。



DLB连接均衡实现


libevent trigger
onSocketEvent()
onSocketEvent()
onAccept()
onAccept()
onAcceptWorker()
onAcceptWorker()
skip
choice
Not ifself
Choose worker
/ skip
Choose worker
/ skip
itseflf
itseflf
onSocketAccept()
onSocketAccept()
(a)
libevent trigger
onSocketEvent()
onSocketEvent()
onAccept()
onAccept()
Send to DLB
Send to DLB
libevent trigger
receive from DLB
receive from DLB
onSocketAccept()
onSocketAccept()
(b)

扩展连接均衡实现

在过去，Envoy的连接均衡策略是无法扩展的，因此需要使用一种灵活的机制扩展现有的连接均衡策略。

旧的的配置方式（不配置即为默认的Nop Connection Balance）：

```
static_resources:
  listeners:
  - connection_balance_config:
      exact_balance: {}
```

扩展连接均衡实现

现在，通过extend_balance的name和typed_config指定一种具体的扩展连接均衡。

DLB连接均衡属于扩展连接均衡。

旧的的配置方式：

```
static_resources:
  listeners:
  - connection_balance_config:
      exact_balance: {}
```

新的的配置方式：

```
static_resources:
  listeners:
  - connection_balance_config:
      extend_balance:
        name: envoy.network.connection_balance.dlb
        typed_config:
          "@type": type.googleapis.com/envoy.extensions.network.connection_balance.dlb.v3alpha.Dlb
```

DLB连接均衡策略

现在，Envoy根据组件的成熟程度等维度，将代码分为source和contrib两个目录进行管理。

新的组件一般是先进入contrib目录，再进入source目录。

官方镜像也因此分为两个：
- envoyproxy/envoy
- envoyproxy/envoy-contrib

DLB连接均衡策略已被社区采纳，代码在contrib目录，最新的1.23 release里已包含此特性。

目前可以通过envoyproxy/envoy-contrib镜像直接使用，Kubernetes环境需要配合DLB device plugin(https://github.com/intel/intel-device-plugins-for-kubernetes#dlb-device-plugin)使用

官方文档：
- https://www.envoyproxy.io/docs/envoy/latest/configuration/other_features/dlb

感谢观看

云原生社区
Cloud Native Community