解决方案介绍

英特尔助力移动云云原生服务网格之中国首例社区版本双栈落地实现

中国移动 China Mobile

目录

背景介绍

众所周知，由于 IPv4 地址的消耗殆尽，且在未来 5G 和物联网等行业趋势的推动下，IPv6 地址的普遍运用是大势所趋。然而网络中主要服务提供商在网络及其应用全部升级到 IPv6 协议之前，通信兼容是必须要解决的问题，而双协议栈技术在其中必将扮演着举足轻重的角色。

所谓双栈是双协议栈（Dual Stack）技术的简称，即是指在一台设备上同时启用 IPv4 和 IPv6 两套协议栈。因此该设备既能和 IPv4 网络端点通信，又能和 IPv6 网络端点通信。网络端点一般是指通信端节点，包括主机和路由器等设备。尽管有其他技术（比如网络隧道和 NAT-TP 技术等）可以使得 IPv6 结点保持与纯 IPv4 节点的通信兼容，但双栈技术始终是最直接的方式。

云原生领域 IPv4/6 双栈介绍

随着云计算及云原生相关标准和技术的普遍运用落地，Kubernetes 容器编排系统和服务网格（Service Mesh）等基础技术得到行业的深度应用。其中，Kubernetes 的双栈支持作为测试特性早在 Kubernetes v1.16（alpha）就已经存在，并在进行了全面测试以后，双栈特性作为功能完备的特性被集成在 Kubernetes v1.21 之中，自从 Kubernetes v1.22 开始，Kubernetes 双栈技术的支持已经被作为稳定的特性而存在，并于 Kubernetes v1.23 版本中进入正式发布（GA）阶段[1]。同时越来越多的云服务提供商也为双栈 Kubernetes 集群提供了支持。

尽管如此，在业界广泛应用的服务网格项目（以大家所熟知且应用最为广泛的 Istio 和 Linkerd 为例）中，双栈技术的支持依然是缺失的。

移动云双栈需求介绍

随着使用 IPv6 成为大势所趋，2020 年工信部发布了《工业和信息化部关于开展 2020 年 IPv6 端到端贯通能力提升专项行动的通知》（工信部通信函〔2020〕57 号）[2]，其中提到一项重点工作任务为大幅提升包括移动云在内的多个云服务平台 IPv6 业务承载能力，2021 年工信部和网信办又发布了《两部门关于印发《IPv6 流量提升三年专项行动计划（2021-2023 年）》的通知》（工信部联通信〔2021〕84 号）[3]，为了全面落实国家部门关于 IPv6 提升的专项行动计划，移动云将通过双栈方式实现公有云产品 IPv4/6 访问作为 2021 年重点工作。

移动云绝大部分公有云产品都实现了云原生化，Istio 作为云原生服务网格领域主流技术，在移动云上获得了广泛的使用，目前将近 50 款产品通过 Istio 及生态组件进行灰度发布、流量治理及服务可观测，为移动云产品快速迭代、线上稳定提供了有力支撑。

但由于 Istio 在双栈技术上的缺失，移动云产品面临双栈访问需求和 Istio 使用需求的矛盾，急需 Istio 在双栈技术方面的实现方案解决这个问题。

为了更好的实现服务网格与 Kubernetes 在双栈支持上的协同工作，满足移动云等用户在 Istio 双栈方面的需求，本文主要介绍了服务网格的代表项目 Istio 在双栈技术中的实现方案以及该方案在移动云的落地实现场景。

Istio 双栈解决方案

本文对 Istio 的双栈支持的实现方案主要参考社区提议（Proposal）[4]。根据其设计思路并基于 Istio v1.14-dev 的版本进行代码改造。同时，目前的实现代码已经放到社区的 Istio repository 中一个名为 **experimental-dual-stack** 的分支[5]。

双栈方案实现

方案中对 Istio 双栈实现的平台有如下基本要求：

组件	组件名称	组件版本
容器编排系统	Kubernetes	推荐：1.21+
DNS 服务	CoreDNS	推荐：1.8.0+
网络 CNI	Calico	推荐：3.17.1+
容器运行时	containerd	推荐：1.5.4+

除了上述的平台要求外，目前引入环境变量 **ISTIO_DUAL_STACK** 来启动 Istio 的双栈功能，同时要求 **PILOT_USE_ENDPOINT_SLICE** 也是被启用的（对于之前提到的开发和测试环境而言，该环境变量是默认启动的）。

在“Istio 双栈支持示意图”中，简化了流量劫持的具体细节并展示了从 service productpage 到 service review 的双栈支持流程。简单来说，Istio 双栈技术的主要实现思路是：如果用户的 Istio 启用了双栈特性，那么 Istio 控制面会为用户的边车代理同时创建基于 IPv4 和 IPv6 地址家族所需的各类 xDS 配置资源，这些由 Istio 控制面下发的核心配置资源包括 LDS，RDS，CDS 和 EDS。

以 Virtual Listener 为例，对于 VirtualInboundListeners 任播地址的监听会分别为边车代理生成 IPv4 和 IPv6 的监听器：**0.0.0.0:15006** 和 **[::]:15006**。对于其他的资源也是类似的处理。需要特别说明的是 Route 相关的配置生成，如图所示，Review 的监听端口是 **9080**，为了让不同地址家族对应不同的 CDS 和 EDS，于是在代码中特地让原始的端口对应 IPv4，而原始端口通过添加 **.IPv6** 后缀来对应 IPv6。

除此以外，Cluster 相关配置的生成中，使用原本的 **inbound** 和 **outbound** 标记 IPv4 的 cluster 配置，相应的使用 **inbound6** 和 **outbound6** 标记 IPv6 的 cluster 配置。

落地场景

基于对 IPV4/IPV6 双栈的市场/客户需求，英特尔携手移动云梳理了 Istio 在移动云上的落地场景，抽象出对应的模型 Demo 进行测试验证（目前 Istio 经典 Demo bookinfo 部分微服务不支持双栈，不能使用 bookinfo 进行验证），共同排查遇到问题并将解决代码贡献给社区，最终双栈方案通过了所有测试。

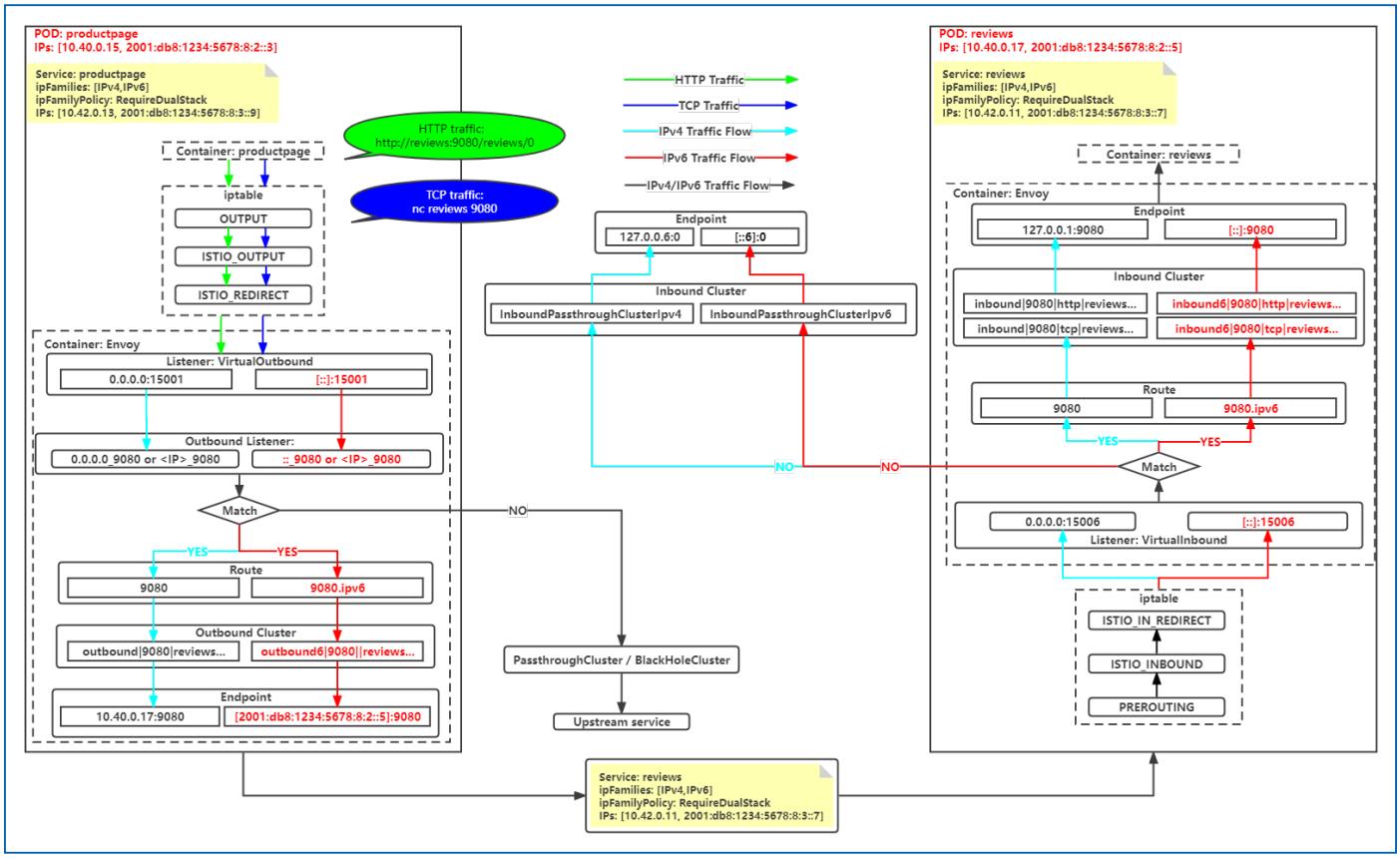
测试环境为 3 台物理机构成的一套 Kubernetes 集群。

物理机硬件配置如下：

CPU	内存
英特尔® 至强® E5-2620 v3 处理器（2*6 核，2.4 GHz）	128G

组件相关信息如下：

组件	组件名称	组件版本
容器编排系统	Kubernetes	v1.21.5
DNS 服务	CoreDNS	1.8.0
网络 CNI	Calico	3.20.2
容器运行时	containerd	1.6.4



图： Istio 双栈支持示意图

灰度发布

Istio 在移动云中应用最广泛的场景是灰度发布，灰度发布细分场景较多，按照产品在 Kubernetes 集群中的部署方式，可以分为单租户和多租户场景下的灰度发布，按照灰度发布范围，可以分为入口微服务灰度发布、内部微服务灰度发布、全链路微服务灰度发布。

我们进行了单租户和多租户场景下的灰度发布测试，在单租户场景中又按灰度发布范围进行了细分场景测试，以下是我们的测试验证介绍。

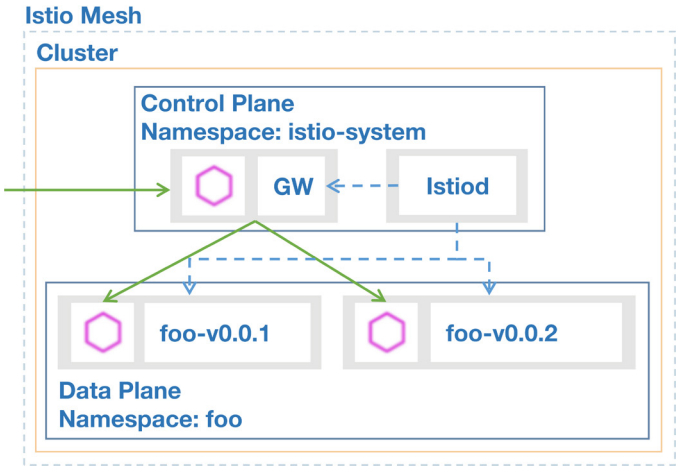
单租户

单租户场景即 Istio 部署模型中的 Single Cluster 模型[6] 在移动云中的应用，指一款产品独占一套 Kubernetes 集群，并部署在一个 namespace 中，通过一套 Istio 管理该 Kubernetes 集群，这是最简单部署场景。

入口微服务灰度发布

入口微服务灰度发布指对产品入口处微服务进行灰度发布的场景，该场景需要 Istio Ingress Gateway 结合入口微服务进行灰度发布。

我们的验证模型架构图如下：

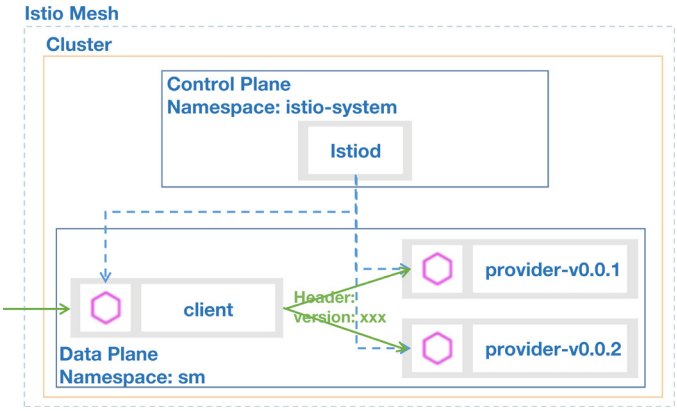


数据面流量通过 Istio Ingress Gateway，转到微服务 foo，正常请求访问 v0.0.1 版本 foo 服务，通过 HTTP 请求 Header 中设置 key-value 对，key 为 version，value 为 v0-0-2，流量流转到 v0.0.2 版本的 foo 服务。

内部微服务灰度发布

内部微服务灰度发布指对产品内部的微服务进行灰度发布的场景。

我们的验证模型架构图如下：

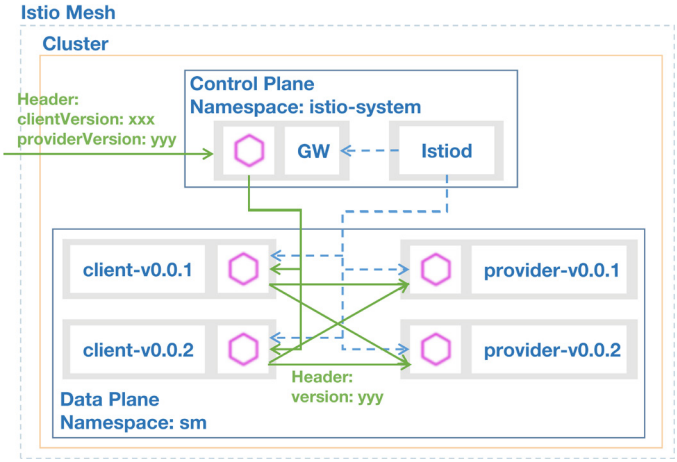


上游访问 client 服务，client 服务再请求 provider 服务，provider 提供两个版本的服务：v0.0.1 和 v0.0.2，正常情况下访问 v0.0.1 的服务，当 Header 中设置 key-value 对，key 为 version，value 为 v0-0-2 时，流量流转到 v0.0.2 版本的 provider 服务。

全链路微服务灰度发布

全链路微服务灰度发布指在产品微服务链路中，对多个微服务进行灰度发布及验证的场景。

我们的验证模型架构图如下：

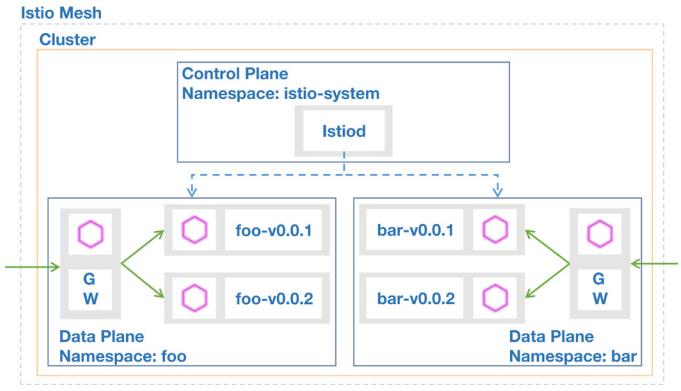


用户通过 Istio Ingress Gateway 访问 client 服务，client 服务再请求 provider 服务。client 提供两个版本的服务：v0.0.1 和 v0.0.2，正常情况下访问 v0.0.1 的服务，当用户通过 Gateway 访问 client 服务时，Header 中设置 key-value 对，key 为 clientVersion，value 为 v0-0-2 时，流量流转到 v0.0.2 版本的 client 服务。provider 也提供两个版本的服务：v0.0.1 和 v0.0.2，正常情况下访问 v0.0.1 的服务，当用户通过 Gateway 访问 client 服务时，Header 中设置 key-value 对，key 为 providerVersion，value 为 v0-0-2 时，流量流转到 v0.0.2 版本的 provider 服务。

多租户

多租户场景即 Istio 部署模型中的 Namespace Tenancy 模型[7] 在移动云中的应用，指多款产品共享一套 Kubernetes 集群，每款产品部署在单独的 namespace 中，通过探索的多租户场景部署方案[8]，实现多款产品共享控制面，独占数据面的场景。

我们在多租户场景下进行了入口微服务灰度发布细分场景的测试验证，验证模型架构图如下：



foo 和 bar 两款产品共享一套 Kubernetes 集群，foo 和 bar namespace 下分别部署 foo、foo 独占的 Istio Ingress Gateway 及 bar、bar 独占的 Istio Ingress Gateway。

产品 foo 的访问流量通过 foo namespace 的 Istio Ingress Gateway，转到微服务 foo，正常请求访问 v0.0.1 版本 foo 服务，通过 HTTP 请求 Header 中设置 key-value 对，key 为 version，value 为 v0-0-2，流量流转到 v0.0.2 版本的 foo 服务。产品 bar 同理。

流量治理

我们选择移动云产品广泛使用的限流功能进行了流量治理相关的验证测试。我们对入口微服务灰度发布场景中 foo 服务应用限流规则，通过 fortio 向 Istio Ingress Gateway + foo 应用发起 IPv4 和 IPv6 协议的并发请求，请求结果符合限流预期。

可观测性

我们按照 Istio 官方的运维组件集成文档[9]，在双栈环境中集成

部署了 Prometheus、Kiali、Jaeger 等可观测性组件。通过双栈方式请求 Istio 纳管的验证 Demo，查看 Kiali 和 Jaeger 系统功能，均符合预期。

总结与展望

移动云双栈落地总结与展望

Istio 双栈方案通过验证后，我们第一时间将方案落地，把双栈版 Istio 集成到移动云云原生平台产品中并进行了充分测试，使其成为了国内首款为客户提供在双栈环境下通过 Istio 进行灰度发布、流量治理、服务拓扑、链路追踪等能力的产品[10]；同时我们在和多款移动云产品进行试点对接，未来会有更多移动云产品通过 Istio 实现双栈访问、灰度发布、流量治理及服务观测。

社区双栈特性总结与展望

事实上，社区是非常欢迎 Istio 双栈特性能够被支持的，并且有许多企业级的用户正在想办法完成他们自己的双栈实现，正因为如此才有 Istio 双栈特性在独立的分支上持续演进。

基于当前的实现虽然能够满足双栈的支持，但是其最大的问题在于 Istio 的控制面分别为 IPv4 和 IPv6 家族地址生成了重复的边车代理配置信息，这会带来更多的资源损耗。因此，最终的方案是消除这些重复的配置信息，并将双栈特性的最终实现推进到 Istio 的主干分支中。

目前为了实现这个目标，英特尔和 Aspen Mesh 等企业正积极的在 Istio 和 Envoy 社区合作，并推进这项工作。而且目前也取得了很多实质性的进展。相信在不远的未来，Istio 的双栈特性的支持将会变得越来越好，而 Istio 也将会成为更多用户首选的服务网格产品。

另外值得一提的是，当前的实现方案由于是实验性的分支，因此并没有完善单元测试和集成测试，但是一些企业级用户已经基于当前的代码实现完成了数百个测试用例的校验，因此对于基本通用的用户场景的校验已经是完成了的。同时需要补充一点，根据当前的实现版本，Istio CNI 和 Calico IPVS 等特性都已经做过验证。我们欢迎并鼓励更多有双栈需求的用户在非生产环境上去体验 Istio 的双栈支持，同时如果有任何问题需要交流，也请踊跃加入我们 Istio 社区的 slack channel: **#dual-stack-support**。

作者

赵复生
英特尔先进软件技术事业部
云计算软件开发经理

张怀龙
英特尔先进软件技术事业部
云原生开发工程师

吕锐新
英特尔销售与市场部
运营商事业部技术总监

参考连接:

[1]: https://kubernetes.io/zh-cn/blog/2021/12/08/dual-stack-networking-ga/
[2]: http://www.gov.cn/zhengce/zhengceku/2020-03/23/content_5494661.htm
[3]: https://www.miit.gov.cn/zwgk/zcwj/wjfb/txy/art/2021/art_3c42d01d124b426e98b4400830da8fd8.html
[4]: https://docs.google.com/document/d/1oT6pmRhOw7AtsldU0-HbfA0zA26j9LYiBD_eepeErsQ/
[5]: https://github.com/istio/istio/tree/experimental-dual-stack
[6]: https://istio.io/latest/docs/ops/deployment/deployment-models/#single-cluster
[7]: https://istio.io/latest/docs/ops/deployment/deployment-models/#namespace-tenancy
[8]: https://cloudnative.to/blog/istio-multi-tenancy-exploration/
[9]: https://istio.io/latest/docs/ops/integrations/
[10]: https://ecloud.10086.cn/home/support/cloudnative

英特尔并不控制或审计第三方数据。请您审查该内容，咨询其他来源，并确认提及数据是否准确。

英特尔技术特性和优势取决于系统配置，并可能需要支持的硬件、软件或服务得以激活。产品性能会基于系统配置有所变化。没有任何产品或组件是绝对安全的。更多信息请从原始设备制造商或零售商处获得，或请见 intel.com

描述的成本降低情景均旨在在特定情况和配置中举例说明特定英特尔产品如何影响未来成本并提供成本节约。情况均不同。英特尔不保证任何成本或成本降低。

英特尔、英特尔标识以及其他英特尔商标是英特尔公司或其子公司在美国和/或其他国家的商标。

© 英特尔公司版权所有