

分享到

# Istio是啥？一文带你彻底了解！

2018-10-19 18:05

公司 / 开发 / 技术

“

如果你比较关注新兴技术的话，那么很可能在不同的地方听说过 Istio，并且知道它和 Service Mesh 有着牵扯。



这篇文章可以作为了解 Istio 的入门介绍，了解什么是 Istio，Istio 为什么最近这么火，以及 Istio 能给我们带来什么好处。

什么是 Istio？

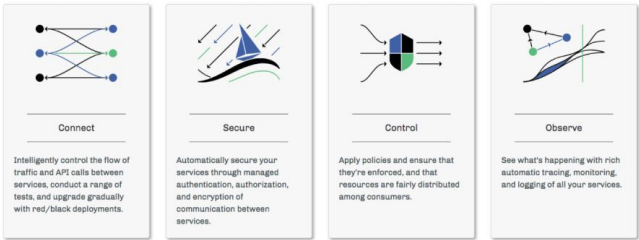
官方对 Istio 的介绍浓缩成了一句话：

An open platform to connect, secure, control and observe services.

翻译过来，就是“连接、安全加固、控制和观察服务的开放平台”。开放平台就是指它本身是开源的，服务对应的是微服务，也可以粗略地理解为单个应用。

合 搜狐 |

0 评论



中间的四个动词就是 Istio 的主要功能，官方也各有一句话的说明。这里再阐释一下：

- 连接（Connect）：智能控制服务之间的调用流量，能够实现灰度升级、AB 测试和红黑部署等功能
- 安全加固（Secure）：自动为服务之间的调用提供认证、授权和加密。
- 控制（Control）：应用用户定义的 policy，保证资源在消费者中公平分配。
- 观察（Observe）：查看服务运行期间的各种数据，比如日志、监控和 tracing，了解服务的运行情况。

大家都在搜：女生元旦宿舍自缢



## 热门图集

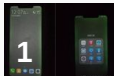


潘玮柏再被曝和空姐女友已在美国登记结婚

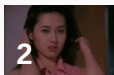
她是天生的尤物，美得男女通杀



## 24小时热文



华为知错就改：凡是出现绿屏的 Mate 20 Pro 都可更换新机  
6400万 阅读



从三级女王到金马影后，蜜桃在这里成熟  
3902万 阅读



百度Q3财报透露出的自信与危机  
6283万 阅读



突发！腾讯损失近万亿，马云重回中国首富！腾讯危机来了？  
6277万 阅读



代购们哭了！代购3年，罚款550万，淘宝店主被判十年！  
6181万 阅读

说了跟没说一样。要想理解上面这几句话的含义，我们还是从头说起，先聊聊 Service Mesh。

分享到 **NOTE：**其实 Istio 的源头是微服务，但这又是一个比较大的话题，目前可以参考网络上各种文章。如果有机会，我们再来聊聊微服务。

什么是 Service Mesh

一般介绍 Service Mesh 的文章都会从网络层的又一个抽象说起，把 Service Mesh 看做建立在 TCP 层之上的微服务层。我这次换个思路，从 Service Mesh 的技术根基——网络代理来分析。

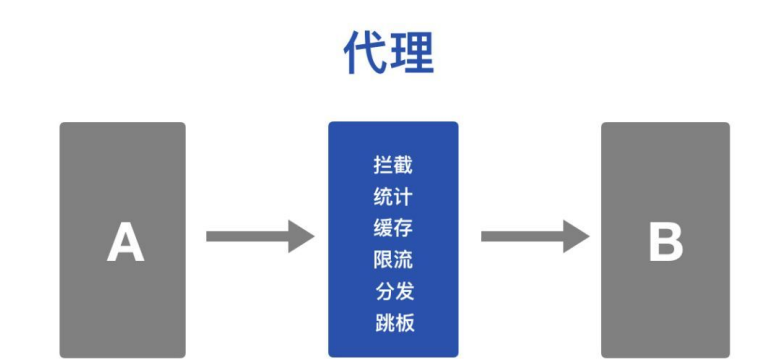
说起网络代理，我们会想到翻墙，如果对软件架构比较熟悉的会想到 Nginx 等反向代理软件。

其实网络代理的范围比较广，可以肯定的说，有网络访问的地方就会有代理的存在。

Wikipedia 对代理的定义如下：

In computer networks, a proxy server is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers.

**NOTE：**代理可以是嵌套的，也就是说通信双方 A、B 中间可以多多层代理，而这些代理的存在有可能对 A、B 是透明的。



简单来说，网络代理可以简单类比起现实生活中的中介，本来需要通信的双方因为各种原因在中间再加上一道关卡。本来双方就能完成的通信，为何非要多此一举呢？

那是因为代理可以为整个通信带来更多的功能，比如：

- 拦截：代理可以选择性拦截传输的网络流量，比如一些公司限制员工在上班的时候不能访问某些游戏或者电商网站，再比如把我和世界隔离开来的 GFW，还有在数据中心中拒绝恶意访问的网关。
- 统计：既然所有的流量都经过代理，那么代理也可以用来统计网络中的数据信息，比如了解哪些人在访问哪些网站，通信的应答延迟等。
- 缓存：如果通信双方比较“远”，访问比较慢，那么代理可以把最近访问的数据缓存在本地，后面的访问不用访问后端来做到加速。CDN 就是这个功能的典型场景。
- 分发：如果某个通信方有多个服务器后端，代理可以根据某些规则来选择如何把流量发送给多个服务器，也就是我们常说的负载均衡功能，比如著名的 Nginx 软件。
- 跳板：如果 A、B 双方因为某些原因不能直接访问，而代理可以和双方通信，那么通过代理，双方可以绕过原来的限制进行通信。这应该是广大中国网民比较熟悉的场景。
- 注入：既然代理可以看到流量，那么它也可以修改网络流量，可以自动在收到的流量中添加一些数据，比如有些宽带提供商的弹窗广告。



搜狐号推荐

- 
- 搜狐科技视界  
搜狐科技官方原创账号。聚焦TMT领域大事件、大趋势和新变化，用我们的视角观...
- 
- 果壳网  
面向都市科技青年们的社交网站。开放、多元的泛科技兴趣社区，并提供负责任...
- 
- 科技说说  
科技说说，说说科技，用最通俗易懂的文字解读行业。专注文娱、金融、电商、...
- 
- 科技小李  
科技评测与最新的科技信息
- 
- 创业家精选  
推送及时的创业创新新闻，传递融资信息，提供行业热点报道。



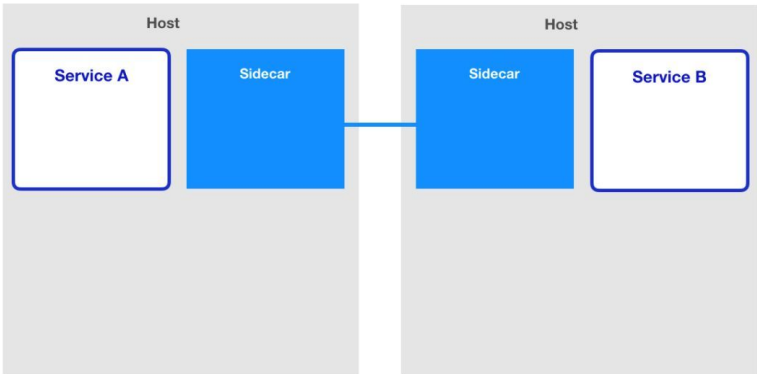
分享到

不是要讲 Service Mesh 吗？为什么扯了一堆代理的事情？因为 Service Mesh 可以看做是传统代理的升级版，用来解决现在微服务框架中出现的问题，可以把 Service Mesh 看做是分布式的微服务代理。

在传统模式下，代理一般是集中式的单独的服务器，所有的请求都要先通过代理，然后再流入转发到实际的后端。

而在 Service Mesh 中，代理变成了分布式的，它常驻在了应用的身边（最常见的就是 Kubernetes Sidecar 模式，每一个应用的 Pod 中都运行着一个代理，负责流量相关的事情）。

这样的话，应用所有的流量都被代理接管，那么这个代理就能做到上面提到的所有可能的事情，从而带来无限的想象力。



此外，原来的代理都是基于网络流量的，一般都是工作在 IP 或者 TCP 层，很少关心具体的应用逻辑。

但是 Service Mesh 中，代理会知道整个集群的所有应用信息，并且额外添加了热更新、注入服务发现、降级熔断、认证授权、超时重试、日志监控等功能，让这些通用的功能不必每个应用都自己实现，放在代理中即可。

换句话说，Service Mesh 中的代理对微服务中的应用做了定制化的改进！

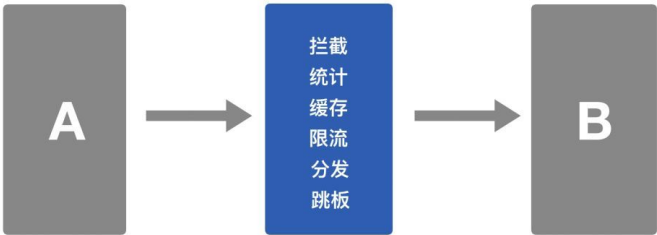


就这样，借着微服务和容器化的东风，传统的代理摇身一变，成了如今炙手可热的 Service Mesh。

应用微服务之后，每个单独的微服务都会有很多副本，而且可能会有多个版本，这么多微服务之间的相互调用和管理非常复杂，但是有了 Service Mesh，我们可以把这块内容统一在代理层。

代理

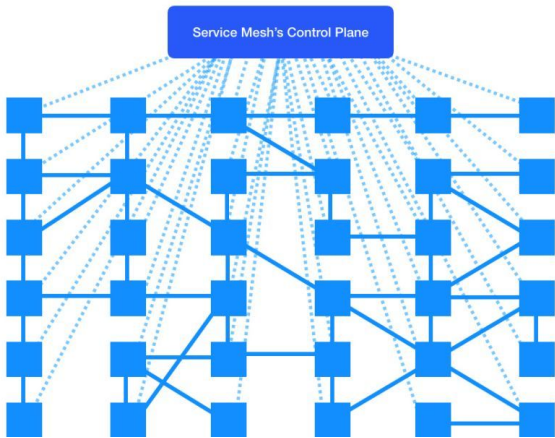
分享到



有了看起来四通八达的分布式代理，我们还需要对这些代理进行统一的管理。

手动更新每个代理的配置，对代理进行升级或者维护是个不可持续的事情，在前面的基础上，在加上一个控制中心，一个完整的 Service Mesh 就成了。

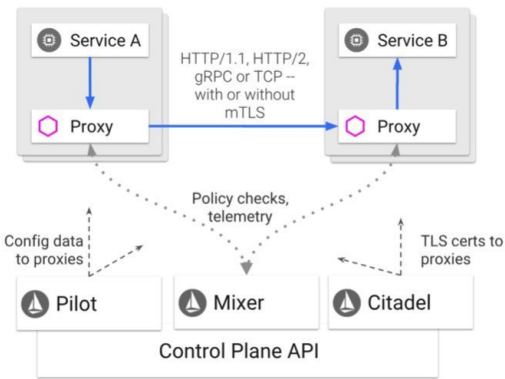
管理员只需要根据控制中心的 API 来配置整个集群的应用流量、安全规则即可，代理会自动和控制中心打交道根据用户的期望改变自己的行为。



**NOTE：**所以你也可以理解 Service Mesh 中的代理会抢了 Nginx 的生意，这也是为了 Nginx 也要开始做 NginMesh 的原因。

再来看 Istio

了解了 Service Mesh 的概念，我们再来看 Istio ，也许就会清楚很多。首先来看 Istio 官方给出的架构图：



可以看到，Istio 就是我们上述提到的 Service Mesh 架构的一种实现，服务之间的通信（比如这里的 Service A 访问 Service B）会通过代理（默认是 Envoy）来进行。

信协议。

控制中心做了进一步的细分，分成了 Pilot、Mixer 和 Citadel，它们的各自功能如下：

分享到

- Pilot：为 Envoy 提供了服务发现，流量管理和智能路由（AB 测试、金丝雀发布等），以及错误处理（超时、重试、熔断）功能。用户通过 Pilot 的 API 管理网络相关的资源对象，Pilot 会根据用户的配置和服务的信息把网络流量管理变成 Envoy 能识别的格式分发到各个 Sidecar 代理中。
- Mixer：为整个集群执行访问控制（哪些用户可以访问哪些服务）和 Policy 管理（Rate Limit, Quota 等），并且收集代理观察到的服务之间的流量统计数据。
- Citadel：为服务之间提供认证和证书管理，可以让服务自动升级成 TLS 协议。

代理会和控制中心通信，一方面可以获取需要的服务之间的信息，另一方面也可以汇报服务调用的 Metrics 数据。

知道了 Istio 的核心架构，再来看看它的功能描述就非常容易理解了：

- 连接：控制中心可以从集群中获取所有服务的信息，并分发给代理，这样代理就能根据用户的期望来完成服务之间的通信（自动地服务发现、负载均衡、流量控制等）。
- 安全加固：因为所有的流量都是通过代理的，那么代理接收到不加密的网络流量之后，可以自动做一次封装，把它升级成安全的加密流量。
- 控制：用户可以配置各种规则（比如 RBAC 授权、白名单、Rate Limit 或者 Quota 等），当代理发现服务之间的访问不符合这些规则，就直接拒绝掉。
- 观察：所有的流量都经过代理，因此代理对整个集群的访问情况知道得一清二楚，它把这些数据上报到控制中心，那么管理员就能观察到整个集群的流量情况了

Istio 解决什么问题

虽然看起来非常炫酷，功能也很强大，但是一个架构和产品出来都是要解决具体的问题。所以这部分我们来看看微服务架构中的难题以及 Istio 给出的答案。

首先，原来的单个应用拆分成了许多分散的微服务，它们之间相互调用才能完成一个任务，而一旦某个过程出错（组件越多，出错的概率也就越大），就非常难以排查。

用户请求出现问题无外乎两个问题：错误和响应慢。如果请求错误，那么我们需要知道那个步骤出错了，这么多的微服务之间的调用怎么确定哪个有调用成功？哪个没有调用成功呢？

如果是请求响应太慢，我们就需要知道到底哪些地方比较慢？整个链路的调用各阶段耗时是多少？哪些调用是并发执行的，哪些是串行的？这些问题需要我们能非常清楚整个集群的调用以及流量情况。

故障排查

这个请求在哪里失败了？A 有调用 B 吗？

为什么用户的请求/页面 hung 住了？

为什么系统这么慢？哪个组件最慢？

此外，微服务拆分成这么多组件，如果单个组件出错的概率不变，那么整体有地方出错的概率就会增大。服务调用的时候如果没有错误处理机制，那么会导致非常多的问题。



加，对于用户来说就是请求卡住了。

分享到

如果没有重试机制，那么因为各种原因导致的偶发故障也会导致直接返回错误给用户，造成不好的用户体验。

此外，如果某些节点异常（比如网络中断，或者负载很高），也会导致应用整体的响应时间变长，集群服务应该能自动避开这些节点上的应用。

最后，应用也是会出现 Bug 的，各种 Bug 会导致某些应用不可访问。这些问题需要每个应用能及时发现问题，并做好对应的处理措施。

应用容错性

客户端没有配置 timeout，导致应用整个卡住

没有重试机制，某个 pod 偶尔出现异常也会导致用户页面错误

某些节点异常（负载高），导致应用整体响应时间变长

某个 pod 有 bug，会耗尽 TCP 连接数或者网络流量

应用数量的增多，对于日常的应用发布来说也是个难题。应用的发布需要非常谨慎，如果应用都是一次性升级的，出现错误会导致整个线上应用不可用，影响范围太大。

而且，很多情况我们需要同时存在不同的版本，使用 AB 测试验证哪个版本更好。

如果版本升级改动了 API，并且互相有依赖，那么我们还希望能自动地控制发布期间不同版本访问不同的地址。这些问题都需要智能的流量控制机制。

应用升级发布

新版本都是一次性升级，出错回滚造成的影响范围很大

无法进行 A/B 测试，根据用户属性访问不同的应用版本

为了保证整个系统的安全性，每个应用都需要实现一套相似的认证、授权、HTTPS、限流等功能。

一方面大多数的程序员都对安全相关的功能并不擅长或者感兴趣，另外这些完全相似的内容每次都要实现一遍是非常冗余的。这个问题需要一个能自动管理安全相关内容的系统。

[分享到](#)

## 系统安全

服务都是 HTTP，而不是 HTTPS

没有流量限制，任何人都可以发送请求进行服务攻击

上面提到的这些问题是不是非常熟悉？它们就是 Istio 尝试解决的问题，如果把上面的问题和 Istio 提供的功能做个映射，你会发现它们非常匹配，毕竟 Istio 就是为了解决微服务的这些问题才出现的。

用什么姿势接入 Istio？

虽然 Istio 能解决那么多的问题，但是引入 Istio 并不是没有代价的。最大的问题是 Istio 的复杂性，强大的功能也意味着 Istio 的概念和组件非常多，要想理解和掌握 Istio，并成功在生产环境中部署需要非常详细的规划。

一般情况下，集群管理团队需要对 Kubernetes 非常熟悉，了解常用的使用模式，然后采用逐步演进的方式把 Istio 的功能分批掌控下来。

第一步，自然是在测试环境搭建一套 Istio 的集群，理解所有的核心概念和组件。

了解 Istio 提供的接口和资源，知道它们的用处，思考如何应用到自己的场景中，然后是熟悉 Istio 的源代码，跟进社区的 Issues，了解目前还存在的 Issues 和 Bug，思考如何规避或者修复。

这一步是基础，需要积累到 Istio 安装部署、核心概念、功能和缺陷相关的知识，为后面做好准备。

第二步，可以考虑接入 Istio 的观察性功能，包括 **Logging**、**Tracing**、**Metrics** 数据。

应用部署到集群中，选择性地（一般是流量比较小，影响范围不大的应用）为一些应用开启 Istio 自动注入功能，接管应用的流量，并安装 Prometheus 和 Zipkin 等监控组件，收集系统所有的监控数据。

这一步可以试探性地了解 Istio 对应用的性能影响，同时建立服务的性能测试基准，发现服务的性能瓶颈，帮助快速定位应用可能出现的问题。

此时，这些功能可以是对应用开发者透明的，只需要集群管理员感知，这样可以减少可能带来的风险。

第三步，为应用配置 **Time Out** 超时参数、自动重试、熔断和降级等功能，增加服务的容错性。

这样可以避免某些应用错误进行这些配置导致问题的出现，这一步完成后需要通知所有的应用开发者删除掉在应用代码中对应的处理逻辑。这一步需要开发者和集群管理员同时参与。

第四步，和 **Ingress**、**Helm**、应用上架等相关组件和流程对接，使用 Istio 接管应用的升级发布流程。

让开发者可以配置应用灰度发布升级的策略，支持应用的蓝绿发布、金丝雀发布以及 AB 测试。

提升整个集群的安全性。

分享到

因为安全的问题配置比较繁琐，而且优先级一般会比功能性相关的特性要低，所以这里放在了最后。

当然这个步骤只是一个参考，每个公司需要根据自己的情况、人力、时间和节奏来调整，找到适合自己的方案。

总结

Istio 的架构在数据中心和集群管理中非常常见，每个 Agent 分布在各个节点上（可以是服务器、虚拟机、Pod、容器）负责接收指令并执行，以及汇报信息。

控制中心负责汇聚整个集群的信息，并提供 API 让用户对集群进行管理。

Kubernetes 也是类似的架构，SDN（Software Defined Network）也是如此。

相信以后会有更多类似架构的出现，这是因为数据中心要管理的节点越来越多，我们需要把任务执行分布到各节点（Agent 负责的功能）。

同时也需要对整个集群进行管理和控制（Control Plane 的功能），完全去中心化的架构是无法满足后面这个要求的。

Istio 的出现为负责的微服务架构减轻了很多的负担，开发者不用关心服务调用的超时、重试、Rate Limit 的实现，服务之间的安全、授权也自动得到了保证。

集群管理员也能够很方便地发布应用（AB 测试和灰度发布），并且能清楚看到整个集群的运行情况。

但是这并不表明有了 Istio 就可以高枕无忧了，Istio 只是把原来分散在应用内部的复杂性统一抽象出来放到了统一的地方，并没有让原来的复杂消失不见。

因此我们需要维护 Istio 整个集群，而 Istio 的架构比较复杂，尤其是它一般还需要架在 Kubernetes 之上，这两个系统都比较复杂，而且它们的稳定性和性能会影响到整个集群。


因此再采用 Istio 之前，必须做好清楚的规划，权衡它带来的好处是否远大于额外维护它的花费，需要有相关的人才对整个网络、Kubernetes 和 Istio 都比较了解才行。

参考资料：

- Istio / What is Istio?: istio 官网上对 istio 进行介绍的文档
- Pattern: Service Mesh: service mesh pattern 详解的文章

作者: cizixs

编辑: 陶家龙、孙淑娟

出处: <http://cizixs.com/2018/08/26/what-is-istio>  [返回搜狐](#), [查看更多](#)

声明: 该文观点仅代表作者本人, 搜狐号系信息发布平台, 搜狐仅提供信息存储空间服务。

阅读 (827)

 不感兴趣  投诉



分享到

淘宝网

¥19.90

¥39

¥68

¥68

iphone

手机壳

葡萄干

小米8

广告

我来说两句

0人参与，0条评论

来说两句吧.....

登录并发表

搜狐“我来说两句”用户公约



还没有评论，快来抢沙发吧！

推荐

- 创业武林大会
- 移动互联网
- 工业技术
- 国家电网
- 阿里星球
- Surface
- 陈竺
- 运动手环
- 智能手表
- Apple Watch
- 测评
- AR

推荐阅读

雷军承诺未来一年不出售小米股份 上市前所获百亿期权将捐赠慈善

和讯 搜狐科技视界 · 今天 09:11

182

雷军关注知乎问题：为什么华为是民族品牌 小米却不是

站长之家 · 昨天 16:46

1748

CES探趋势：为什么说今年5G商用稳了？

和讯 搜狐科技视界 · 今天 12:12

华为欲通过笔记本和平板电脑 在美国杀出重围

和讯 环球网 · 昨天 15:09

222

《40周年999纯银纪念币》  
40周年纪念币震撼发行 999纯银

广告 · 今天 14:11

库克不满苹果被唱衰：1000亿美元收入不靠iPhone

和讯 环球网 · 今天 11:09

170

滴滴内部反腐：去年83人涉腐败舞弊被解聘，8人被移送司法

澎湃 澎湃新闻 · 今天 11:16

2018微信数据报告：快来看不同年龄段喜爱表情都是什么？

搜狐科技快讯 · 今天 12:09

快手收购在线文档“一起写” 意图布局B端业务？

和讯 搜狐科技视界 · 今天 12:33

又一批假蜂蜜被查封，赴深山调查，终于揭开蜂蜜惊天骗局，引发国人震怒！



广告 · 今天 14:11

【原创】高通苹果案再生悬念 5G时代专利之争激战正酣

财联社 · 今天 09:29



“艺术升”App受质疑后下架部分收费项目！系统维护暂停6小时

南方都市报 · 昨天 23:34

26

22亿市值 vs 8亿亏损：一杯咖啡背后的资本游戏



Bianews · 昨天 20:31

4

华为P20手机“狂降”遭网友吐槽：下手太早，下次你还支持吗？



第一监控 · 今天 09:31



起底冬虫夏草：一个“未知”大骗局！宋同仁曝光虫草行业掺假黑幕！



广告 · 今天 14:11



世界上第一款真正的可折叠手机问世，网友：然而，我无法停止打开和关闭它

搞笑精选app · 今天 10:05



中关村在线：荣耀V20性能均强于万元笔记本电脑！



数码辣条 · 今天 11:04

8

红米品牌独立：首个产品4800万像素只能拍1200像素照片！



手机观察室 · 昨天 16:52

183



华为/苹果/小米手机供应商大盘点！

模切易得通 · 今天 09:06

1

为何要吃干海参，而不是鲜海参？答案在这！



“孟晚舟事件”39天



铉刻度 · 昨天 19:37

11

行业标杆应该有的样子 CES 戴尔发新款XPS 13



中关村在线 · 今天 11:01

4

库克现身回应苹果风波：猛烈抨击高通 苹果生态无懈可击  
手表耳机收入超iPod巅峰

TECH2IPO创见 · 今天 10:55

36

华为已经在测试新操作系统了, 比iOS还快, 值得期待



核舟POST · 今天 13:23

148

40个漂亮的html5网站欣赏

广告 · 今天 14:11

加载更多