

# 初识 Service Mesh



黄祁 (/u/7fbe09a81fa8) [+关注](#)

2.3 2018.01.16 18:24 字数 1773 阅读 47176 评论 3 喜欢 65

(/u/7fbe09a81fa8)

## 初识 Service Mesh

### 1. 介绍

#### Service Mesh 概念

Service Mesh又译作“服务网格”，作为服务间通信的基础设施层。Willian Morgan ( Linkerd的CEO ) 如下定义Service Mesh。

A service mesh is a dedicated infrastructure layer for handling service-to-service communication. It's responsible for the reliable delivery of requests through the complex topology of services that comprise a modern, cloud native application. In practice, the service mesh is typically implemented as an array of lightweight network proxies that are deployed alongside application code, without the application needing to be aware. (But there are variations to this idea, as we'll see.)

Service Mesh 是一个基础设施层，用于处理服务间通信。云原生应用有着复杂的服务拓扑，Service Mesh 保证请求可以在这些拓扑中可靠地穿梭。在实际应用当中，Service Mesh 通常是由一系列轻量级的网络代理组成的，它们与应用程序部署在一起，但应用程序不需要知道它们的存在。

Service Mesh 实际上就是处于 TCP/IP 之上的一个抽象层，它假设底层的 L3/L4 网络能够点对点地传输字节（当然，它也假设网络环境是不可靠的，所以 Service Mesh 必须具备处理网络故障的能力）。

#### Service mesh 有如下几个特点：

1. 应用程序间通讯的中间层；
2. 轻量级网络代理；
3. 应用程序无感知；
4. 解耦应用程序的重试、超时、监控、追踪和服务发现；



## 2. 原理

### Service Mesh 基本原理

如果用一句话来解释什么是 Service Mesh，可以将它比作是应用程序或者说微服务间的 TCP/IP，负责服务之间的网络调用、限流、熔断和监控。对于编写应用程序来说一般无须关心 TCP/IP 这一层（比如通过 HTTP 协议的 RESTful 应用），同样使用 Service Mesh 也就无须关系服务之间的那些原来是通过应用程序或者其他框架实现的事情，比如 Spring Cloud、OSS，现在只要交给 Service Mesh 就可以了。

Phil Calçado 在他的这篇博客 Pattern: Service Mesh 中详细解释了 Service Mesh 的来龙去脉：

1. 从最原始的主机之间直接使用网线相连
2. 网络层的出现
3. 集成到应用程序内部的控制流
4. 分解到应用程序外部的控制流
5. 应用程序的中集成服务发现和断路器
6. 出现了专门用于服务发现和断路器的软件包/库，Twitter's Finagle和 Facebook's Proxygen。这时候还是集成在应用程序内部
7. 出现了专门用于服务发现和断路器的开源软件，如：NetflixOSS ecosystem
8. 最后作为微服务的中间层Service Mesh出现

Service Mesh 的架构如下图所示

service-mesh-control-plane

## 3. 方案



目前社区Service Mesh的开源解决方案有：Buoyant 公司推出的 Linkerd 和 Google、IBM 等厂商牵头的 Istio。Linkerd 更加成熟稳定些，Istio 功能更加丰富、设计上更为强大，社区相对也更加强大一些。所以普遍认为 Istio 的前景会更好，但是毕竟还处于项目的早期，问题还很多。

(/apps/  
utm\_sc  
banner

### 3.1 Istio 介绍

Istio是由Google、IBM和Lyft开源的微服务管理、保护和监控框架。Istio为希腊语，意思是”起航“。官方中文文档地址：<https://istio.doczh.cn> (<https://link.jianshu.com?t=https%3A%2F%2Fistio.doczh.cn>)

**Istio架构图：**

istio-arch

---

Istio架构分为控制层和数据层。

1. 数据层：由一组智能代理（ Envoy ）作为sidecar部署，协调和控制所有microservices之间的网络通信。
2. 控制层：负责管理和配置代理路由流量，以及在运行时执行的政策。

Istio架构各个组成部分。

- Envoy：Istio使用Envoy代理的扩展版本，该代理是以C++开发的高性能代理，用于调解service mesh中所有服务的所有入站和出站流量。
- Mixer：Mixer负责在service mesh上执行访问控制和使用策略，并收集Envoy代理和其他服务的遥测数据。
- Istio Manager：Istio-Manager用作用户和Istio之间的接口，收集和验证配置，并将其传播到各种Istio组件。



- Istio-auth：Istio-Auth提供强大的服务间和最终用户认证，使用相互TLS，内置身份和凭据管理。

### 3.2 Linkerd 介绍

Linkerd 是开源网络代理，设计为以服务网格部署：用于管理，控制和监控应用程序内的服务与服务间通讯的专用层。

#### Linkerd 架构图

what\_it\_does

why\_is\_it\_necessary

**Linkerd 基本功能** 原文链接 (<https://link.jianshu.com?t=https%3A%2F%2Flinkerd.io%2Ffeatures%2F>)

1. Load balancing：负载均衡算法，它们使用实时性能指标来分配负载并减少整个应用程序的尾部延迟。
2. Circuit breaking：自动熔断，将停止将流量发送到被认为不健康的实例，从而使他们有机会恢复并避免连锁反应故障。



3. Service discovery : 服务发现后端集成, 通过删除特定的(ad-hoc)服务发现实现来帮助降低代码的复杂性。
4. Dynamic request routing : 动态请求路由和重新路由, 允许您使用最少量的配置来设置分段服务(staging service), 金丝雀(canaries), 蓝绿部署(blue-green deploy), 跨DC故障切换和黑暗流量(dark traffic)。
5. Retries and deadlines : 在某些故障时自动重试请求, 并且可以在指定的时间段之后让请求超时。
6. TLS : 可以配置为使用 TLS 发送和接收请求, 您可以使用它来加密跨主机边界的通信, 而不用修改现有的应用程序代码。
7. HTTP proxy integration : 可以作为 HTTP 代理, 几乎所有现代 HTTP 客户端都广泛支持, 使其易于集成到现有应用程序中。
8. Transparent Proxying : 在主机上使用 iptables 规则, 设置通过 linkerd 的透明代理
9. gRPC : 支持 HTTP/2 和 TLS, 允许它路由 gRPC 请求, 支持高级 RPC 机制, 如双向流, 流程控制和结构化数据负载。
10. Distributed tracing : 分布式跟踪和度量仪器, 可以提供跨越所有服务的统一的可观察性。
11. Instrumentation : 支持分布式跟踪和度量仪器, 可以提供跨越所有服务的统一的可观察性。

(/apps/  
utm\_sc  
banner

## 4. 扩展阅读

- 极客时间: 什么是Service Mesh (<https://link.jianshu.com?t=https%3A%2F%2Ftime.geekbang.org%2Farticle%2F2355>)
- 极客时间: Service Mesh深度解析 (<https://link.jianshu.com?t=https%3A%2F%2Ftime.geekbang.org%2Farticle%2F2360>)
- 极客时间: 解读2017之Service Mesh: 群雄逐鹿烽烟起 (<https://link.jianshu.com?t=https%3A%2F%2Ftime.geekbang.org%2Farticle%2F2362>)
- kubernetes-handbook (<https://link.jianshu.com?t=https%3A%2F%2Fjimmysong.io%2Fkubernetes-handbook>)
- Jimmy Song (<https://link.jianshu.com?t=https%3A%2F%2Fjimmysong.io%2F>)
- Service Mesh 在华为公有云的实践 (<https://link.jianshu.com?t=http%3A%2F%2Fgitbook.cn%2Fbooks%2F5a1e7dca387c5b4ee351790b%2Findex.html>)

## 5. 参考资料

- Service Mesh 服务网格 (<https://link.jianshu.com?t=https%3A%2F%2Fjimmysong.io%2Fkubernetes-handbook%2Fusecases%2Fservice-mesh.html>)
- Pattern: Service Mesh ([https://link.jianshu.com?t=http%3A%2F%2Fphilcalcado.com%2F2017%2F08%2F03%2Fpattern\\_service\\_mesh.html](https://link.jianshu.com?t=http%3A%2F%2Fphilcalcado.com%2F2017%2F08%2F03%2Fpattern_service_mesh.html))



- What's a service mesh? And why do I need one? (<https://link.jianshu.com?t=https%3A%2F%2Fbuoyant.io%2F2017%2F04%2F25%2Fwhats-a-service-mesh-and-why-do-i-need-one%2F>)
- linkerd.io (<https://link.jianshu.com?t=https%3A%2F%2Flinkerd.io%2F>)
- Linkerd官方文档中文版 (<https://link.jianshu.com?t=https%3A%2F%2Fwww.gitbook.com%2Fbook%2Fdoczhcn%2Flinkerd%2Fdetails>)
- Istio官方文档中文版 (<https://link.jianshu.com?t=http%3A%2F%2Fistio.doczh.cn%2F>)

(/apps/  
utm\_sc  
banner

小礼物走一走，来简书关注我

赞赏支持

📖 技术笔记 (/nb/14286560) 举报文章 © 著作权归作者所有



黄祁 (/u/7fbe09a81fa8) ♂

写了 15893 字，被 62 人关注，获得了 93 个喜欢

(/u/7fbe09a81fa8)

一直在探索.....

+ 关注

喜欢 | 65



更多分享





下载简书 App ▶  
随时随地发现和创作内容



(/apps/redirect?utm\_source=note-bottom-click)

被以下专题收入，发现更多相似内容






- 

程序员 (/c/NEt52a?utm\_source=desktop&utm\_medium=notes-included-collection)
- 

后台开发专题 (/c/87fe68cf5803?utm\_source=desktop&utm\_medium=notes-included-collection)
- 

我爱编程 (/c/7847442e0728?utm\_source=desktop&utm\_medium=notes-




-  included-collection) (/c/96078504df73?utm\_source=desktop&utm\_medium=notes-included-collection)
-  收藏 (/c/96078504df73?utm\_source=desktop&utm\_medium=notes-included-collection)
-  新技术 (/c/9daacdefb5e0?utm\_source=desktop&utm\_medium=notes-included-collection)
-  MicroS... (/c/95be06eed051?utm\_source=desktop&utm\_medium=notes-included-collection)
-  技术杂谈 (/c/583cff2e52c5?utm\_source=desktop&utm\_medium=notes-included-collection)

展开更多 ▾

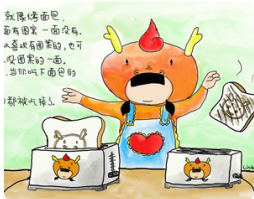
(/apps/  
utm\_sc  
banner

### 全面对比指南：Service Mesh能否成为下一代SDN (/p/964f2f998706?utm...

本文获得jameskelly.net授权翻译发表，转载需要注明来自公众号EAWorld。 作者：James Kelly 译者：月满西楼 原题：Are Service Meshes the Next-Gen SDN? 全文7500字，阅读约需要18分钟 2017年6月28日更...


 EAII企业架构创新研究院 (/u/72a1f772fe47?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommenc

(/p/23fdaa6bb65b?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommenc  
**数人云|万字解读:Service Mesh服务网格新生代--Istio (/p/23fdaa6bb65b?u...**

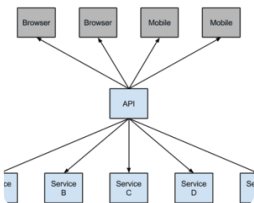
Service Mesh新秀，初出茅庐便声势浩荡，前有Google，IBM和Lyft倾情奉献，后有业界大佬俯首膜拜，这就是今天将要介绍的主角，扛起Service Mesh大旗，掀起新一轮微服务开发浪潮的Istio！讲师简介：敖小剑...


 优云数智 (/u/8d891d8ffdd7?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommenc

(/p/46fd0faecac1?

utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommenc  
**Spring Cloud (/p/46fd0faecac1?utm\_campaign=ma...**

Spring Cloud为开发人员提供了快速构建分布式系统中一些常见模式的工具（例如配置管理，服务发现，断路器，智能路由，微代理，控制总线）。分布式系...



 卡卡罗2017 (/u/d90908cb0d85?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommenc

(/p/987155ce2298?

utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommenc  
**Qcon2017实录|Service Mesh：下一代微服务 (/p/987155ce2298?utm\_ca...**


▲扫码报名活动 数人云11月Meetup报名开启，看中西方大神如何论道云原生与





全案整装模式-林凤装饰龙湖唐宁one-北欧风格效果图 (...)

本案整体风格是北欧风格，硬装上没有做过多造型，主要是在软装搭配方面进行色彩调控，和整体风格把控。 客厅说明: 客厅整体运用浅蓝色的墙面乳胶漆做...

 装修达人关小婆儿 (/u/219e972fba03?)



utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation\_banner (/apps/utm\_source=recommendation\_banner)

