微服务架构Service Mesh的设计与应用

陶 志，向忠清

（武汉邮电科学研究院，湖北 武汉 430074）

摘 要：微服务架构正处于兴起阶段，在SpringCloud，Dubbo框架的支持下，微服务框架成为了最主流的分布式应用解决方案，然而随着微服务实施水平的不断深化，不论是市面上的哪种框架，都会存在一些问题，本文将分析当前微服务架构存在的一些问题，并且针对这些问题找到基于Service Mesh的Istio框架这种比较好的解决方案。最后在实际的运用场景区域卫生综合管理平台进行探讨，说明了Service Mesh的这种架构思想优于传统的微服务的架构思想。

关键词：微服务；Service Mesh；Istio；区域卫生综合管理平台

中图分类号：TP393.09 文献标志码：B 文章编号：1003-7241(2020)01-0049-05

Research And Implement of Micro-Services Architecture Service Mesh

TAO Zhi, XIANG Zhong-qing

(Wuhan Research Institute of Posts and Telecommunications, Wuhan 430074 China)

Abstract: Micro-services Architecture is on the rise. with the support of SpringCloud and Dubbo, it has become the main solution for the distributed application. However, with the deepening of Micro-services implementation level, it will be some problems no matter which framework is on the market. This article analysises the key problem in the current Micro-services Architecture. At the same time, this article finds the relatively good solution for the Istio architecture which is based on Service Mesh for these problems. At last, this article discusses in the practical application scenario on region health management platform. It demonstrates this framework is better than the traditional micro-services framework.

Key words: Micro-services; Service Mesh; Istio; region health management platform

1 引言

自从微服务在2014年三月被首次提出以来，它很快的就成为了最主流的分布式应用的解决方案，第一代的微服务框架以SpringCloud和Dubbo为主，但无论是SpringCloud还是Dubbo都存在技术门槛高，多语言支持不足和代码侵入性强等问题，本文会详细探讨这些问题，并会针对这些问题提出基于Service Mesh的Istio的这种解决方案，并根据这种解决方案构建区域卫生综合管理平台系统，分析这种解决方案的优势。

2 研究现状

2.1 微服务架构

微服务架构是一种新的软件体系设计模式，它并没有形成统一，严格的定义，但是基于其分布式环境应用的场景，却拥有一些共同的特征：比如开发敏捷性、持续交付、可伸缩性、最终一致性等[1]。

微服务的基本思想在于考虑围绕着业务领域组件来创建应用，这些应用可独立地进行开发、管理和加速[2]。在分散的组件中使用微服务云架构和平台，使部署、管理和服务功能交付变得更加简单[3]。

微服务架构建议将大型复杂的单体架构应用划分为一组微小的服务，每个微服务根据其负责的具体业务职责提炼为单一的业务功能[4]。每个服务可以很容易地部署并发布到生产环境里隔离和独立的进程内部，它可以很容易地扩展和变更。对于一个具体的服务来说可以采用任何适用的语言和工具来快速实现，服务之间基于基

收稿日期：2018-10-31

础设施互相协同工作。

2.2 当前微服务架构存在的问题

2.2.1 技术门槛高

随着微服务实施水平的不断深化，除了基础的服务发现，配置中心和授权管理之外，在实施的过程中不可避免的需要在服务治理层面面临着各种新的挑战如分布式跟踪，熔断降级，灰度发布，故障切换等等，这对整个团队提出了非常高的技术要求[5]。

2.2.2 多语言支持不足

对于稍具规模的团队，尤其是在高速成长的互联网创业公司，多语言的技术栈是常态，跨语言的服务调用也是常态，但目前开源社区上并没有一套统一的、跨语言的微服务技术栈。

2.2.3 代码侵入性强

主流的微服务框架（比如Spring Cloud、Dubbo）或多或少都对业务代码有一定的侵入性，框架替换成本高，导致业务团队配合意愿低，微服务落地困难。

2.3 解决方案

综合本文提到的现在微服务框架存在的一些问题，比较容易想到的解决方案就是代理模式，在负载均衡层（LB）处理所有的服务调用以及服务治理（分布式跟踪，熔断降级）[6]。

这样设计会带来的另外一个问题就是中心化架构，需要一个代理中心来处理所有的服务调用和服务治理，因此代理端的可用性和性能成为了整个系统的瓶颈，并且这样会给运维团队带来高难度的工作。Service Mesh却能够很好的解决这一问题。

3 Service Mesh

3.1 Service Mesh定义

服务网格由控制面板和数据面板组成，它是专门用于处理服务到服务通信的基础结构层。服务网格是独立于具体的服务而存在的，这种模式从根本上解决了多语言支持不足以及代码侵入性的问题，并且由于服务网格的独立性，业务团队不需要关心服务治理相关的复杂度，可以全权交给服务网格处理[7]。

3.2 边车模式

由服务网格的定义可知，服务网格由控制模块和数据模块组成，而Sidecar就是完成控制模块的这一部分的功能，与其它模式不同的是边车模式会针对每一个服务实例，服务网格都会在同一主机上一对一的部署一个边车进程，接管这个服务对外所有的通信，如图1所示。

这样每一个服务都会有一个边车进程处理业务逻辑之外的事情，很好的解决了代理模式的中心化的性能瓶颈问题。

3.3 服务网格的由来

由于每一个服务实例都会有一个边车代理与之配对，服务之间的通信都是通过边车代理进行的，因此就会

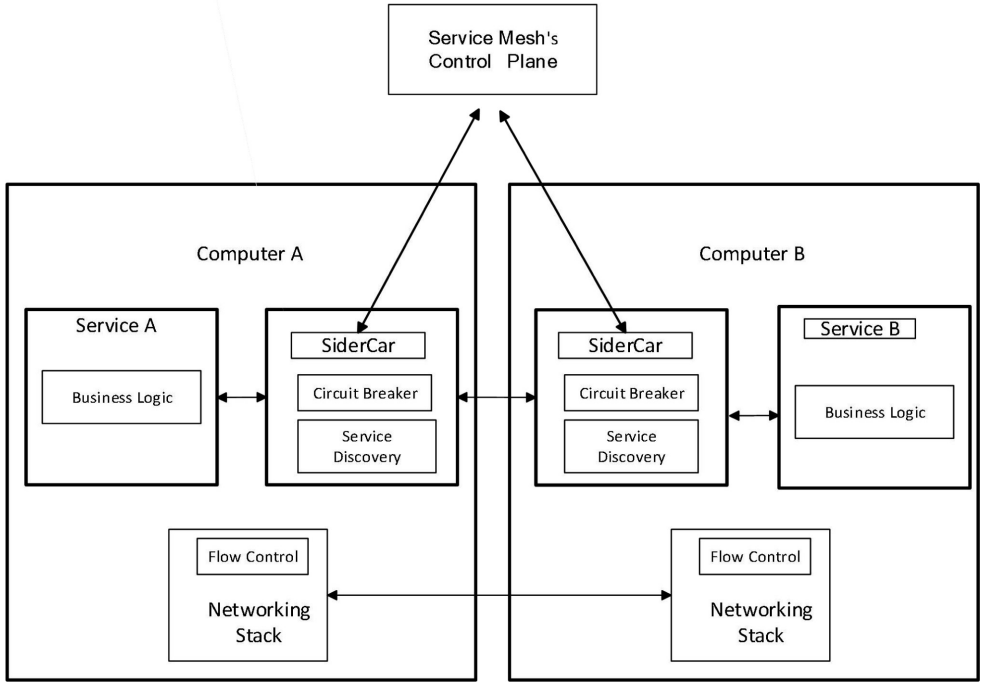

图1 边车模式示意图

得到图2的部署图：

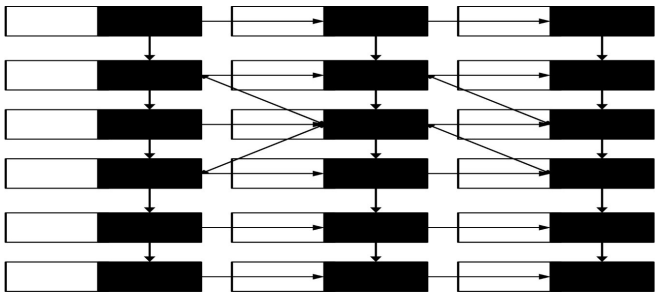

图2 服务网格部署图

这种代理之间的交叉连接形成了一种网络网格。

4 Istio微服务框架

4.1 Istio介绍

Istio是由Google，IBM和Lyft这三家互联网巨头联合开发的一个基于服务网格的开源项目。Istio首先是一个服务网格，但是Istio却不仅仅是服务网格，它在Linkerd，Envory这样典型的服务网格之上，Istio提供了一个完整的解决方案，为整个服务网格提供行为洞察和操作控制，以满足微服务应用程序的的多样化需求。

4.2 Istio架构

Istio架构图如图3所示。

Istio服务网格逻辑上分为数据面板和控制面板。数据面板由一组智能代理(Envoy)组成，代理部署为sidecar，调解和控制微服务之间所有的网络通信；控制面板负责管理和配置代理路由流量以及在运行时执行的策略。

4.2.1 Envoy

Envoy是一种高性能的代理，可以完成负载均衡，TLS终止，动态服务发现，HTTP/2&gRPC代理，熔断器，健康检查，基于百分比流量拆分的分段推出，故障注入和丰富指标。

Envoy被部署为sidecar，和对应服务在同一个Kubernetes pod中。这允许Istio将大量关于流量行为的信号作为属性提取出来，而这些属性又可以在Mixer中用于执行策略决策，并发送给监控系统，以提供整个网格行为的信息。Sidecar代理模型还可以将Istio的功能添加到现有的部署中，而无需重新构建或重写代码。

4.2.2 Mixer

Mixer负责在服务网格上执行访问控制和使用策略，并从Envoy代理和其他服务收集遥测数据。代理提取请求级属性，发送到Mixer进行评估。Mixer包括一个灵活的插件模型，使其能够接入到各种主机环境和基础设施后端，从这些细节中抽象出Envoy代理和Istio管理的服务。

4.2.3 Pilot

Pilot负责收集和验证配置并将其传播到各种Istio组件。它从Mixer和Envoy中抽取环境特定的实现细节，为他们提供服务的抽象表示，独立于底层平台。此外，流量管理规则可以在运行时通过Pilot进行编程。

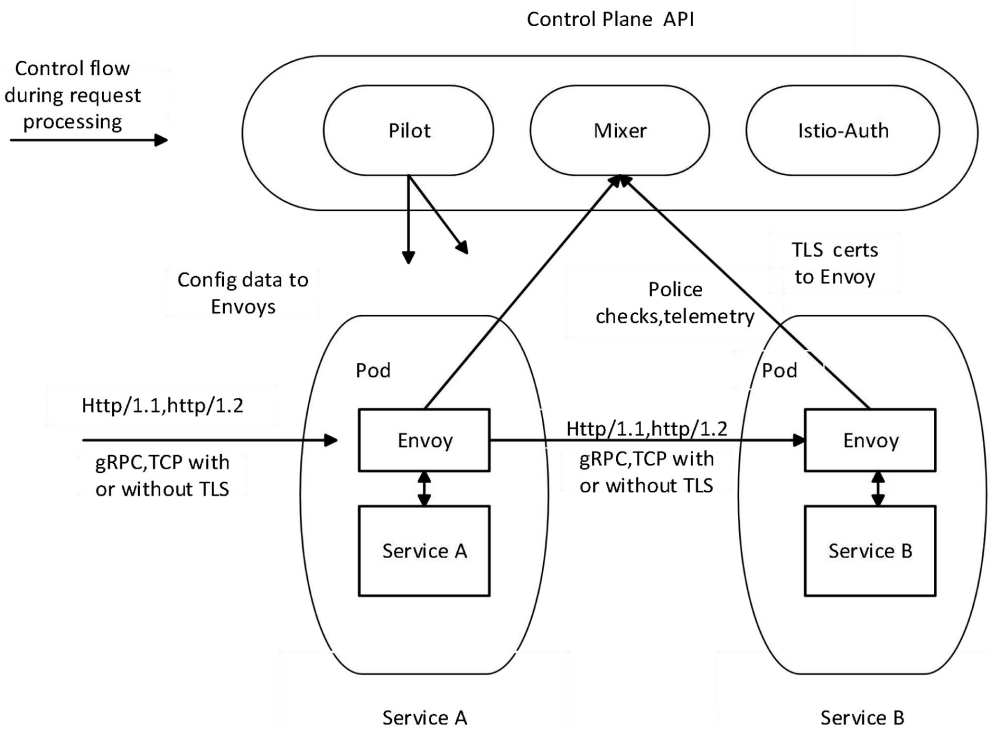

图3 Istio架构图

4.2.4 Istio-Auth

Istio-Auth提供强大的服务间认证和终端用户认证，使用交互TLS，内置身份和证书管理。可以升级服务网格中的未加密流量，并为运维人员提供基于服务身份而不是网络控制来执行策略的能力。

4.3 Istio功能以及特性

4.3.1 流量管理

流量管理的核心组件是Pilot，它管理和配置部署在特定Istio服务网格中所有Envoy代理实例。它允许指定Envoy代理之间使用什么样的路由流量规则，并配置如超时，重试和熔断器这些故障发生时的故障恢复功能。Pilot还维护了网格中所有服务的规范模型，并使用这个模型，来通过发现服务让Envoy了解网格中的其他实例。

每个Envoy实例都会维护负载均衡信息，负载均衡信息从Pilot获得，并且Pilot会在其负载均衡池中的定期进行健康检查，从而实现智能分配流量，同时遵循指定的路由规则。这种实现方式使得服务之间的调用更加可靠，并且使系统在恶劣放入网络环境下更加健壮。

4.3.2 策略与执行

Istio使用属性来控制在服务网格中运行的服务的运行时行为，属性是具有名称和类型的元数据片段，用以描述入口和出口流量，以及这些流量所属的环境。Istio携带特定信息片段，如API请求的错误代码，API请求的延迟或TCP连接的原始IP地址。有一个有关Istio部署的固定的属性词汇表，这个表里面定义了常用的基准属性集。

Istio完成策略控制主要在Mixer当中完成，Mixer在应用程序代码和基础架构后端之间提供通用中介层，这种设计将策略决策移出应用层，用运维人员能够控制的配置取而代之。如图4所示。

应用程序代码不再将应用程序代码与特定的后端集成在一起，而是与Mixer进行相当简单的集成，然后Mixer负责与后端系统连接。这样能够改变层次之间的边界，以此降低总体的复杂性。从服务代码中剔除策略逻辑，改由运维人员进行控制。

4.3.3 服务身份以及安全

服务身份以及安全是由Istio Auth这个模块进行完成的，Istio Auth的目标是提高微服务及通信的安全性，而不需要修改服务代码。它主要由身份，秘钥管理和通信安全三个部分组成，这三个组件完成了加密服务间的通信的功能。为网格中的服务提供可验证身份，并提供保护服务流量的能力，使其可以在不同可信度的网络上流转。

5 Service Mesh的应用

为实现医疗卫生信息的共享和互联互通，将以社区医院为中心的的信息资源进行整合以及利用，将会利用Service Mesh微服务架构搭建一个卫生区域综合管理平台，来综合管理珠海市高栏岗区的卫生信息。

5.1 区域卫生综合管理平台实例分析

通过合理的组织信息，使得不同的用户能够通过区域卫生管理平台来得到他们想要获取的信息。整个系统的架构图如图5所示。

这几个微服务分别使用多种不同的语言编写，并且这些服务和Istio没有依赖关系，在这一点上，Istio架构很好的解决了多语言支持不足和代码侵入性的问题。并且不需要再次开发一个鉴权服务去验证用户的身份，因为Istio Auth能够完成这个功能，在使用的过程中进行相关配置即可以。

请求的入口是Show这个服务，Show服务是一个使

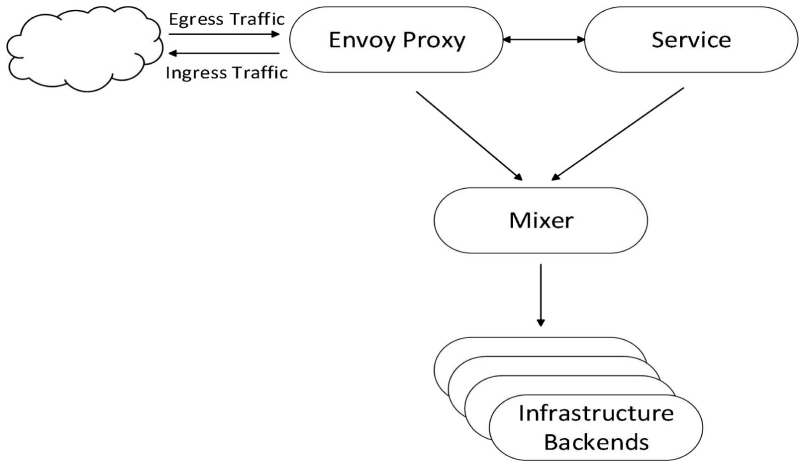

图4 Mixer执行策略控制流程图

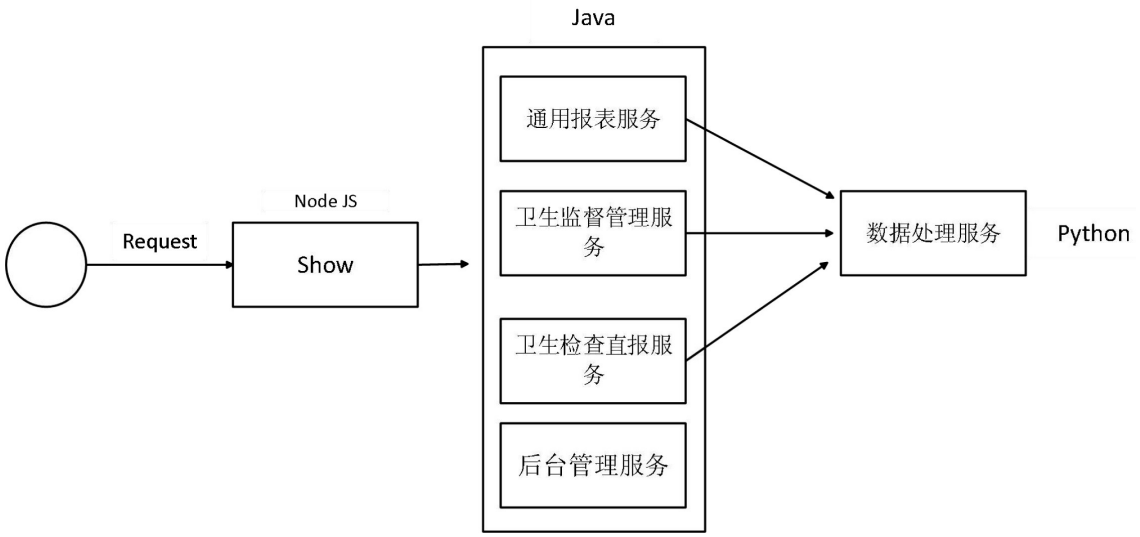

图5 区域卫生管理平台整体架构图

用node js写的提供页面展示服务，这个服务在请求不同的接口的时候会调用后台管理服务，卫生监督管理服务，卫生监督管理直报服务和通用报表服务，这四个服务是使用java语言编写的。于此同时，还有一个专门进行数据处理的服务，这个服务是使用python语言写的，专门用来处理数据。不同的服务根据需求的不同采用合适的编程语言完成服务的开发。

它对于后端程序的调用都是通过域名调用的，它不需要通过服务发现系统获取这个域名，也不需要关心转发，不会意识到自己是否使用了Service Mesh，自己是否部署在Kubenetes上，它把服务之间的通信完全的交给了基础设施层，自己无需关心除业务之外的其他事情。

6 结束语

本文将传统的微服务架构和基于Service Mesh的微服务架构进行对比，指出了传统微服务架构现在存在的问题，以及Service Mesh这种新的微服务架构使用边车模式来解决传统微服务框架存在的问题，同时说明了基于服务网格的Istio框架在具体实现中是如何解决这些问题的，最后使用Istio框架构建了区域卫生综合管理平台这个系统，分析了这个系统使用Istio框架的好处。当然Istio微服务框架当前还存在一些需要进一步完善的问题，这些还需要在今后的实践中不断探索与学习。

参考文献：

[1] 王磊.微服务架构与实践[M].北京:电子工业出版社,2015.

[2] 黄小锋,张晶.微服务架构框架介绍与实现[J].电脑与信息技术,2016(6):14-16.

[3] 蒋勇.基于微服务架构的基础设施设计[J].软件,2016,37(5):93-97.

[4] 王健,李冬睿.从单一模式系统架构往微服务架构迁移转化技术研究[J].科教导刊,2016(9):43-44.

[5] 毕小红,刘渊,陈飞.微服务应用平台的网络性能研究与优化[J].计算机工程,2018,44(5):53-59.

[6] 毛润菲.微服务系统的可靠性分析及关键问题[J].电子技术与软件工程,2018(14):51.

[7] 孙杰平.论Service Mesh在微服务架构中的优势[J].通信世界,2018(5):49.

作者简介：陶志(1992-)，男，硕士，研究方向：互联网技术。