

• 经典评述 •

微服务架构评述

赵 然^{1 2} 朱小勇¹

(¹ 中国科学院声学研究所 国家网络新媒体工程技术研究中心 北京 100190 ² 中国科学院大学 北京 100049)

摘要: 微服务架构在 2012 年开始出现技术雏形, 并逐步取代传统的单体式架构。2014 年学者 Martin Fowler 正式提出微服务架构的概念, 与此同时, 容器技术的快速发展为微服务架构的大规模使用提供了基础支撑。2014 年至今, 微服务架构已成为行业内最流行的服务架构。本文首先介绍了微服务架构的概念和主要特点; 然后详细描述了国内外学术界对于微服务架构的学术论文与研究工作; 最后介绍了微服务架构的优点和缺点以及与传统的单体式架构特点的对比。

关键词: 微服务, 服务架构, 综述

A Review of the Microservice Architecture

ZHAO Ran^{1 2}, ZHU Xiaoyong¹

(¹National Network New Media Engineering Research Center, Institute of Acoustics,

Chinese Academy of Sciences, Beijing, 100190, China,

²University of Chinese Academy of Sciences, Beijing, 100049, China)

Abstract: The microservice architecture started to appear in technology prototypes in 2012, and it gradually replaced the traditional monolithic architecture. Martin Fowler proposed the concept of the microservice architecture in 2014. At the same time, the rapid development of the container technology provided the foundation for the large – use of the microservice architecture. Since then, the microservice architecture has become the most popular service architecture in the industry. This paper first introduces the concept and the main characteristics of the microservice. Then the literatures and the research about the microservice at home and aboard are described specifically. At last, the advantages and disadvantages of the microservice architecture are introduced, and they are compared with the traditional monolithic architecture.

Keywords: microservice, service architecture, review

0 引言

随着互联网时代的到来, 云计算技术和物联网技术快速发展, 互联网应用提供商的硬件资源与软件资源越来越多地以服务的形式提供给用户^[1]。越来越大的用户体量、越来越复杂的服务内容对互联网公司和服务提供商的可扩展性、敏捷性、容错性等方面的能力提出了越来越高的要求, 这也促使后台网络应用服务的架构形式在不断地演进, 从早期的单体式架构(monolithic architecture)到后来的面向服务架构 SOA(service – oriented architecture)。

在这个演进过程中, 微服务架构(microservice architecture)由面向服务架构 SOA 慢慢发展而来。微服务架构于 2012 年开始出现技术雏形, 并逐步取代传统的单体式架构。2014 年学者 Martin Fowler 正式提出微

本文于 2018-08-12 收到 2018-12-04 收到修改稿。

服务架构的概念,与此同时,容器技术的快速发展为微服务架构的大规模使用提供了基础支撑。2014年至今,微服务架构已成为行业内最流行的服务架构。

1 微服务架构概述

1.1 微服务架构概念

微服务架构是一种互联网应用服务的软件架构,主要应用于互联网应用服务的服务端软件开发。微服务架构由面向服务架构 SOA 发展而来,其核心理论基础来自于康威定律[2]中关于组织结构与其设计的系统结构之间关系的描述,即任何组织设计的系统,其结构都是组织本身沟通结构的复制。

2014 年学者 Martin Fowler 正式提出微服务架构的概念^[3]:微服务架构以一套微小的服务的方式来开发和部署一个单独的应用,这些微小的服务根据业务功能来划分,通过自动化部署机制独立部署运行在自己的进程中,微服务之间使用轻量级通信机制来进行通信。一个典型的微服务架构应该包括客户端、微服务网关、服务发现、微服务原子层、数据库、部署平台等模块,根据不同应用类型及服务规模,可以增加负载均衡、权限认证、服务熔断、日志监控等模块,来满足服务的非功能性需求。

1.2 微服务架构特点

虽然 Martin Fowler 给出了微服务的一种定义,但是他也同时指出,微服务并不局限于该定义。Martin 尝试归纳和描述微服务架构风格所具有的共同特点,这些特点并不是所有微服务架构风格都要拥有的,也不是用来定义微服务架构本身的,而是微服务架构风格被希望要拥有的特点。也就是说,微服务架构风格不是微服务化的终点,而是微服务化的方向。下面简单介绍一下文献[3]中总结的几个微服务架构风格的特点。

1.2.1 服务组件化

微服务中,服务可以被当作进程外组件,独立进行部署,服务之间利用网络服务请求或者远程过程调用进行通信。一个好的微服务架构的目标是通过服务合同中的解耦服务边界和进化机制来帮助各个微服务独立部署运行。微服务架构的设计者希望对任何一个组件或者服务的改动和升级都只需要重新部署该服务而不需要重新部署整个应用程序,并且在升级过程中尽可能少地改变服务间通信的接口。

1.2.2 围绕业务功能组织服务

当把一个大的应用拆分成小的部分的时候,通常的方法都是根据技术层面分为 UI 团队、服务端逻辑团队和数据库团队。但是这种拆分团队的方式会使得即使一个简单的变动都会导致整个团队需要耗费时间和预算来适应和协调。

康威在文献[2]中提出了康威定律,其中有一条提到:任何组织设计的系统,其结构都是组织本身沟通结构的复制。根据康威定律的这条描述,微服务架构采用围绕业务功能来拆分应用和组织服务的方法。在微服务架构中,设计组织被分为小的开发团队,与之相对应的是,应用被拆分为小的服务,应用之间的沟通过程就是团队之间的沟通过程。为保证应用之间沟通过程清晰明确,团队之间需要划分清晰的服务边界。这样的划分方法也要求每个小团队本身提供开发过程中所需要的所有技能。

1.2.3 基础设施自动化

许多开发团队都是使用持续交付和持续集成技术来构建微服务架构的应用和系统的,这使得基础设施自动化技术得到了广泛的应用。而随着容器技术、云计算技术等技术在过去几年的快速发展,基础设施自动化技术取得了长足的进步,这也间接降低了微服务构建、部署、运行的复杂性。

2 国内外研究现状

在互联网服务行业中,微服务架构逐渐成为非常流行的架构,在学术界关于微服务架构也同样有一些研究,下面分别从国内研究和国外研究两方面介绍国内外关于微服务架构的研究工作。

2.1 国内研究现状

国内相关研究文献中对于微服务的研究工作主要集中在利用微服务框架搭建新的服务系统或者将原有的服务系统微服务化,并分析服务系统的功能性需求和非功能性需求,解决微服务化过程中遇到的一些问题。

北京大学的龙新征等人在文献[4]中针对微服务架构中的服务注册、服务发现、负载均衡等常见问题,提出了一种分层的微服务框架。基于这个框架设计并实现了“北京大学校园移动信息服务平台”,为校内师生提供信息服务。同济大学的郭栋等人在文献[5]中设计了一种基于微服务架构构建的云件 PaaS 平台,可以在不需要对传统软件做出任何改动的情况下,将软件部署到云端 WEB 服务器,并通过浏览器以 WEB 的形式为终端用户提供软件服务。北京交通大学的谭一鸣在文献[6]中对平台化服务框架的功能性需求和非功能性需求进行了分析,设计并实现了使用 API 网关构建的微服务系统。南京大学的唐文宇在文献[7]中分析了微服务的安全性访问需求,设计并实现了一个安全系统来解决微服务中的第三方应用审核认证及微服务权限访问控制问题。北京交通大学的肖仲垚在文献[8]中分析了业务微服务化过程中对于通信框架的高效简洁高可用等需求,设计并实现了一种微服务通信框架。北京邮电大学的宋鹏威在文献[9]中设计了一个“开放式微服务框架”,并归纳出了一组行业相关开发者在开发过程中所需要注意的内容和原则。作者还基于这些内容和原则完成了一个“开放式数据采集、存储和分析服务”,证明了该框架对微服务的开发指导工作的有效性。

2.2 国外研究现状

国外相关研究文献中对于微服务技术的研究工作可以分为 4 个方面:介绍微服务架构,并结合具体项目来说明效果;开发微服务相关组件或工具,实现微服务架构下的某些特殊功能;评估比较微服务架构的性能;宏观讨论企业从单体式架构转变为微服务架构的原因及可能遇到的问题。

在具体微服务架构项目方面,Hasselbring 在文献[10]中介绍了在欧洲最大的电子商务平台之一的 otto.de 上引入微服务架构,沿业务服务进行垂直分解,为应用和服务提供高可扩展性和高可用性的能力,并解决了微服务化过程中的耦合、集成、可伸缩性、监视和开发等问题。Innerbichler 等人在文献[11]中设计并开发了一种基于微服务架构的物联网平台 NIMBLE,该平台通过分散,可扩展服务的组合来实现平台的核心业务功能,服务之间以及平台用户、制造商、供应商、传感器和 Web 资源之间的通信通过简单的协议和轻量级机制来支持。在开发微服务组件方面,Granchelli 等人在文献[12]中设计了一种用于微服务体系结构恢复的原型工具 MicroART,能够生成基于微服务的系统软件架构模型。Mayer 等人在文献[13]中提出了一种用于微服务的监控和管理的实验仪表,能够满足用户不同需求并支持集成不同监控设施来收集微服务运行数据。在评估微服务架构性能方面,Amaral 等人在文献[14]中利用主从式和嵌套式容器两种模型来比较微服务架构中的 CPU 和网络性能。Ueda 等人在文献[15]中使用了 Acme Air 来分析比较微服务和单体架构运行 Node.js 和 Java 的性能,实验结果表明相同的硬件配置下微服务的额外开销更大,消耗时间也更多。在宏观讨论企业微服务化可能遇到的问题方面,Kalske 等人在文献[16]中指出,迈向微服务架构的典型原因是复杂性、可扩展性和代码所有权,企业微服务化面对的挑战可以分解为技术挑战和组织挑战,前者包括微服务的解耦、划分服务边界代码重构等,后者则包括根据康威定律将大的团队分为可以自主工作的小型团队等等。

3 微服务架构与单体式架构特点对比

提到微服务架构,就不得不提到经常与其对比的传统单体式架构(也叫整体式架构)。单体式架构是指应用被构建为一个整体的单元,对系统做出任何改动都需要重新构建、部署新版本的服务端应用。单体式架构是一种很自然的应用构建方式,其优点有:比较适合小项目,开发过程简单,开发团队统一,集中式服务,没有分布式的管理问题。但是单体式架构同样有很多缺点,比如:单体式架构所有的代码都在一个项目

里,开发团队沟通成本高,开发效率低,对于任何一个小小的改动都需要重新构建和部署应用,同时单体式架构的容错性也较低,一个微小的错误也可能导致整个系统崩溃。

与之相对应的,微服务架构虽然不够符合人的直觉,但是更加符合康威定律,团队职责和业务边界也更加明确。微服务架构的优点是整体的服务可以被切分得比较小,代码开发效率高;每个微服务都可以独立开发,对开发团队的技术栈要求更低;每个微服务独立部署,更有利于持续集成,可扩展性也更高;每个微服务可以独立更新,升级迭代更方便,容错性也更高。微服务架构在带来诸多的优点的同时,也还有着很多缺点,比如数据一致性问题、性能监控问题、服务间通信机制问题等等,归结起来这些都是微服务架构的分布式系统带来的复杂性问题。

总体而言,单体式架构由于其集中式的特点,更适合服务功能单一、开发团队规模小的应用项目,在应用复杂程度较低的情况下,单体式架构使得应用的额外开销更小,整体性能更好。而相比单体式架构,微服务架构更加适合服务功能多样、开发团队规模庞大的应用项目,微服务架构的服务解耦合、独立部署、容错性高等特点可以使得在应用复杂程度较高的情况下,应用的性能损耗更小,以满足应用服务的非功能性需求。另外,由于微服务架构的可扩展性好的特点,微服务架构也比单体式架构更适合业务规模增长较快、服务内容变化较大的应用服务,这一点也与现今互联网服务行业的快速发展的特点更加契合。

4 结束语

自2014年微服务架构的概念被正式提出以来,微服务技术凭借其独立部署、可扩展性、快速迭代、容错性、易于开发等优点正在快速发展并逐渐成为业界最流行的软件架构。本文首先介绍了微服务架构的概念和主要特点,然后详细描述了国内外学术界对于微服务架构的学术论文与研究工作,最后介绍了微服务架构的优点和缺点以及与传统的单体式架构特点的对比。

微服务架构是互联网时代的特点与需求、分布式架构与敏捷开发的流行、持续集成持续部署、容器技术的慢慢成熟等条件结合的产物,是互联网时代技术浪潮中的一朵小小的浪花,其核心仍然是康威定律中探讨的组织结构与其设计的系统结构之间的关系。微服务技术对于应用服务开发团队来说,与其说是一种技术,不如说是一种针对软件开发的指导思想和设计理念。微服务架构本身不是目的,也不是能够解决所有问题的灵药,更重要的还是开发团队结合自身实际需求与具体问题选择更合适的应用架构。

参 考 文 献

- [1] 马斌,冯波.海云协同环境下服务运营环境关键技术的探讨[J].网络新媒体技术 2015 4(02):1-9.
- [2] Melvin E. Conway. How do committees invent? [EB/OL]. http://www.melconway.com/Home/Committees_Paper.html, 1968.
- [3] Martin Fowler James Lewis. Microservices a definition of this new architectural term [EB/OL]. <https://martinfowler.com/articles/microservices.html>, 2014.
- [4] 龙新征,彭一明,李若森.基于微服务框架的信息服务平台.东南大学学报(自然科学版) 47(S1):48-52, 2017.5
- [5] 郭栋,王伟,曾国荪.一种基于微服务架构的新型云件 PaaS 平台 [J]. 信息网络安全 2015 (11):15-20
- [6] 谭一鸣. 基于微服务架构的平台化服务框架的设计与实现 [D]. 北京:北京交通大学 2017.
- [7] 唐文宇. 面向 SOA 架构微服务的安全系统的设计与实现 [D]. 南京:南京大学 2016.
- [8] 肖仲垚. 微服务通信框架的设计与实现 [D]. 北京:北京交通大学 2017.
- [9] 宋鹏威. 开放式微服务框架的设计与应用 [D]. 北京:北京邮电大学 2017.
- [10] Hasselbring W, Steinacker G. Microservice Architectures for Scalability, Agility and Reliability in E-Commerce [C]//ieee international conference on software architecture workshops, 2017: 243-246.
- [11] J. Innerbichler, S. Gonul, V. Damjanovic-Behrendt, B. Mandler and F. Strohmeier, NIMBLE collaborative platform: Microservice architectural approach to federated IoT [C]//2017 Global Internet of Things Summit (GIoTS), Geneva, 2017: 1-6.

(下转封三)

今天的商业模式很难设想出来,而是在实践不断摸索出来,才能被各个行业所接受。一个停车场能接受多大的通讯量?收什么样的费用?和一个养猪场、一辆汽车、一辆波音飞机是完全不同的。现在,运营商正在花大量的时间测试网络调控、时延多少……我认为应该有一个新的实验室,在进行5G的技术实验的同时,运营商、参与者与投资者都应该有商业模式的实验室。这种实验室的价值和意义非常重大。

在5G时代,网络安全变成了非常重要的领域。没有网络安全,就没有5G,就像一辆汽车没有油门就没有闸一样。作为运营商来讲,未来没有安全就没有业务,而不同的业务需求有不同等级的安全,安全实时性的保障,24小时监控,是运营商非常强大的能力。

总结一下:第一,当我们做5G网络技术实验的同时,也应该参与商业模式的实验,针对不同的客户和应用类型,作出正确的反馈;第二,把安全视作和5G发展同样重要的领域,把安全与业务连接在一起,才能真正改变业务模式,让整个生态系统繁荣起来。

Q3 对新的移动生态系统、对垂直行业在5G时代的发展有什么建议?

田溯宁:我认为面临这样一场大的变革,第一点,开放非常重要,开放自己的大脑,不断去学习,到自己不熟悉不舒适的领域去了解。第二点,大家都在谈生态价值,我觉得说起来容易做起来难。企业管理需要有新的位置,过去CEO是“Chief Executive Officer”,未来CEO可能是“Chief Ecosystem Officer”。你能不能建立这样一个生态系统,能不能让大家都获利?在这个过程中,你要谦虚,要为别人去学习。最后一个感受很有意思,我参加过互联网的创业,参加过国企,又做了投资,你会发现,每个体系相对都比较封闭。我们怎么才能拥抱不同的体系?不管是文化、语言还是利益。我觉得这跟寒武纪一样,是一个大爆发,新物种产生的时代。做好了,就能把握一个全新的变革时机;做不好,就会很快被淘汰。所以学会开放、学会生态、学会协作,这三点非常重要。

(源自:亚信安全官方微信,发布时间:2018-11-29)

(上接第61页)

- [12] G. Granchelli, M. Cardarelli, P. D. Francesco, I. Malavolta, L. Iovino and A. D. Salle, MicroART: A Software Architecture Recovery Tool for Maintaining Microservice – Based Systems [C] // 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, 2017: 298 – 302.
- [13] Mayer B, Weinreich R. A Dashboard for Microservice Monitoring and Management [C] // ieee international conference on software architecture workshops, 2017: 66 – 69.
- [14] Amaral M, Polo J, Carrera D, et al. Performance Evaluation of Microservices Architectures Using Containers [C] // network computing and applications, 2015: 27 – 34.
- [15] Ueda T, Nakaike T, Ohara M, et al. Workload characterization for microservices [C] // ieee international symposium on work-load characterization, 2016: 85 – 94.
- [16] Kalske M, Makitalo N, Mikkonen T, et al. Challenges When Moving from Monolith to Microservice Architecture [C] // international conference on web engineering, 2017: 32 – 47.

作者简介

赵然 男 (1992-) 博士研究生 研究方向:边缘计算,容器虚拟化技术。

朱小勇 男 (1982-) 博士 副研究员 研究方向:嵌入式系统,多媒体技术。