

微服务架构的发展与影响分析

◆ 李贞昊

摘要: 本文以当前“互联网+”和双创时代信息服务平台软件开发的要求为切入点,总结和梳理了微服务架构的发展过程,分析了微服务架构在开发成本、运行速度、可维护性等方面的优缺点。并结合软件行业和“互联网+”相关行业的变化分析了微服务架构正在和即将带来的影响。

关键词: 微服务; 软件架构; 服务商店

引言

随着“互联网+”时代的到来以及智能硬件的兴起,无论是PC应用、移动应用、还是可穿戴设备、机载车载设备,越来越多的终端应用场景下都需要在线信息服务。通过信息服务,不同客户端和服务端在云端服务的支撑下,可以实现快速的信息获取、分发和同步。这些信息以某种特定的格式传输,绝大多数通过HTTP协议实现。在大众创业、万众创新的今天,各行业都以“互联网+”相关应用作为增加自身行业优势,加快服务效率的重要途径。面对需求的瞬息万变、以及跨行业的特殊要求,使自身信息服务的软件架构成本可控、扩展性强、易于维护成为非常迫切的要求^[1]。在微服务(Microservices)架构风格被提出之前,在服务端构建、开发并部署信息服务时,通常需要以统一的、完整的开发框架作为实现基础。然而,面对业务的不断更变,尤其是在企业面对新需求,需要在旧系统和框架上作调整时,“一站式”的信息服务软件架构显得越来越力不从心。微服务的提出和应用无疑为这类企业和项目带来了曙光。它使得信息服务的软件架构在演化和产生变动时的维护成本大大降低,并显著提高了稳定性。

一、微服务的演化过程

在微服务出现之前,网站服务(web)、接口服务(移动端)的服务架构通常是“一站式”的。一个服务应用中通常包含多个模块和接口,传统的开发模式和设计风格是将所有功能、接口统一设计,分层设计,逐层开发实现。以一个典型的web网站为例。它至少包含表示层(UI)、业务层(逻辑控制)、服务层(services)、数据访问层(DAO)、持久层(数据库)5层。在这5层的几十上百个模块中,若有任何一处没有完成开发或编译成功,都可能影响整个站点的部署。

针对上述问题,20世纪90年代末,SOA(Service Oriented Architecture)面向服务架构被提出^[2],SOA首次提出了在软件架构设计时,使用低耦合并面向服务流程的思想。将流程分解为步骤,而每个步骤的实现都可以定义为一个服务。但SOA的架构中,复杂的ESB企业服务总线依然处于

非常重要的位置,整个系统的架构并没有实现完全的组件化以及面向服务,它的学习和使用门槛依然偏高。

微服务架构的思想本质上来源于项目设计中对业务功能和模块的水平、垂直切割、拆分。2012年,微服务的架构被提出,并陆续出现了诸多设计案例。在随后的几年里,Amazon、Uber等企业进行了各自的实践并取得了成功。微服务强调完全的组件化和面向服务。所有微服务均是独立的,微服务对外以REST API形式暴露给调用者。

二、微服务架构的特点

微服务是一种软件架构的设计风格^[3],整个软件服务架构由多个微服务构成,它并没有一成不变的规定,而是需要根据业务需求来做设计。

2.1 微服务架构原理。在微服务架构中,每个微服务只负责非常明确、独立、简单的任务处理,并将处理结果以API的形式返回给外部。从实践的角度看,微服务即是对整个软件平台或项目的细粒化拆分,拆分后的所有微服务独立运行,互不干扰,每个微服务都运行在独立的容器中。在以往的架构设计中,服务层是以参数或对象传递的方式实现细粒化过程的使用,这一过程在内存中完成。而微服务将其搬到了“外部”。例如,用户的登录过程,传统的接口或动作(Action)中需要经历黑白名单检查、密码口令比对、登录日志记录3个过程。而在微服务架构中,可将这3个过程都以微服务方式构件,独立工作,进而使微服务的代码成为组件。微服务架构使用一种服务管控模块的思想来代替ESB总线,根据业务要求的不同,服务控制模块至少需要包含服务的注册、代理、发布、路由功能。

2.2 微服务架构优势。微服务架构的主要优点在于它可以使软件架构变得灵活,经过低耦合拆分的微服务在开发、测试、部署、更新阶段都带来了极高的效率,业务的变更变得低成本和低风险。越是在规模庞大的软件项目中,微服务架构的优势越明显。从企业角度来看,由于微服务架构只与API相关,它使得不同服务可以使用不同的语言和开发框架进行开发,极大地扩大了企业对人才招聘的限定范围。从工程实现的角度来看,微服务架构能将以往开发中的代码、对

象和模块的复用转变为服务复用,大大降低项目在开发过程的管理成本,这对于一些需要基于基础版的软件功能进行个性化开发场景是帮助巨大的。

在实际应用中,不同的微服务通常以 Docker 作为独立容器部署^[4,5],Docker 容器可以共享一台主机的内核代码部分,相比独立的资源消耗较大的虚拟机而言更加小巧。同时它在进程级别是相互隔离的,使得微服务的代码执行可以完全独立,即便其中某些微服务因故障无法工作,也不影响其它微服务,从而使整个软件服务的健壮性得到提升。Docker 的这种恰到好处的“半隔离半虚拟化”特点为微服务的应用提供了强有力的支撑。

2.3 微服务架构不足。微服务架构诚然也有它的短处。首先,微服务架构的设计并不是将所有操作都服务化、组建件,例如一些数据库层的底层操作是不推荐服务化的。架构设计者需要根据业务具体情况进行合理的划分。此外,微服务架构的目的是解决软件开发和迭代过程中的效率问题,由于将本应该在内部使用的服务性功能以 API 的方式暴露出来,使得软件平台的一次对外数据服务过程中,需要增加数次基于 HTTP 协议的内部数据传输。虽然传输在内网完成,但相比内存中函数接口的调用慢了几个数量级。这对于后期服务用量增加时,软件平台服务器的设计和部署是一个不小的挑战。最后,由于 API 在内网的暴露,设计者还需要对关键微服务进行鉴权保护,以保证一些敏感微服务不被滥用。

三、微服务架构的影响

目前微服务架构仅在一些中大型公司得到了实践,还没有完全被中小型企业接受。由于微服务架构的优势,给相关行业带来的将不止是软件开发管理上的影响。

3.1 加速互联网信息服务繁荣。微服务架构的灵活性可以使项目的开发过程更易于管理,从而大幅度提升项目开发团队的开发效率。微服务独立性、分外分割的设计思想可以使开发人员更加关心组件内部的构造,从而提高软件代码的质量。同时,服务的复用使开发人员从以往的代码、对象、模块复用中解脱出来。服务复用的优势在于开发人员不需要在工程中去关心代码级别的构件和测试工作,仅需要依据定义好的 REST API 向相应地址发送请求即可,真正实现“一次部署,无限使用”。这将使企业或项目团队在构建后续功能时,时间和人力成本大幅降低,从而整个行业将有更多时间基于已有微服务开发更多新的微服务,也可将一个项目过程中的某些微服务对外开放调用,获得收益。

3.2 微服务商店的兴起。随着微服务开发者的增多,越来越多开发者愿意将自己开发的微服务开放给全网的用户使用,甚至已有开发者专门开发微服务为职业。SAP 公司于 2016 年下半年推出了 Yaas 平台^[6],专门为微服务进行销售,Yaas 平台类似于手机端的应用商城、App Store,它是基于微服务架构实现的服务管理整合台。Yaas 开放全世界的开发者接入他们自己开发的微服务应用并以服务调用次数作为计费依

据,为开发者获取收益。目前 Yaas 平台中已经集成了众多与电子商务相关的微服务,如订单管理、支付管理、商品管理等。未来随着微服务开发者的增加,Yaas 平台的微服务应用将更加丰富。

3.3 加速创业项目的成形。2014 年以来,随着国家创新创业政策的支持,越来越多的行业与互联网行业跨界融合,创业项目以井喷式涌现。通常创业项目中的信息服务平台开发和外包式的软件开发的过程有其极大的不同。创业项目中很多业务和商业模式都是试探性和实验性的,这导致其需求变化更为频繁。传统的统一式、整体式的服务开发框架难以适应,若创业团队不能把握好需求变更和生产环境的节奏,很容易造成服务整体性不可用。微服务架构的出现使得初创项目的软件开发可以遵循由中心到外围,由重要到次要的微服务开发次序,降低因需求临时变动带来的对项目整体性的冲击,从而加速项目的成形。

四、结束语

微服务架构的发展已过走了 3 年多的历程,从最初的概念到一些大企业的积极实践。尽管实际应用中各家设计的细节不尽相同,但它引领了一种全新的大规模软件平台的设计样式。从行业的反应来看,微服务架构将在未来几年中更加深刻地影响软件行业和其它与“互联网+”相关行业。

参考文献

- [1] 徐斌.“互联网+”时代软件和信息服务业的发展路线研究[J].中国商论,2015(34):66-68.
- [2] 吴家菊,刘刚,席传裕.基于 Web 服务的面向服务(SOA)架构研究[J].现代电子技术,2005,28(14):1-4.
- [3] Fowler M, Lewis J. Microservices (2014)[J].2016.
- [4] 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.
- [5] 孙海洪.微服务架构和容器技术应用[J].金融电子化,2016 (5): 63-64.
- [6] 孙杰贤.从 YaaS 看传统软件企业转型[J].中国信息化,2016 (8): 60-61.

(基金编号:科技部,国家科技支撑计划课题:数字音乐服务模式研究及新型营销关键技术研发(2012BAH44F02))
(作者单位:电子科技大学信息与软件工程学院)