2021年·第2期
栏目编辑：梁丽雯 E-mail:liven_01@163.com

微服务架构与容器技术探析

■ 中国人民银行琼海市支行 邢贞明
中国人民银行海口中心支行 李登辉 潘 博

摘 要：随着系统分布式架构技术的发展，微服务成为越来越多大型互联网企业的选择，由于其完美解决了原有单体式应用可扩展、自适应能力不足的问题，成为了当前主流的分布式应用解决方案。本文剖析微服务的优势及特点，详细分析容器基础设施实现微服务的优势，为基层央行有效促进信息系统的分布式架构转型升级探索有效途径。

关键词：微服务；容器技术；Kubernetes

一、引言

随着互联网技术的快速发展，为适应日益增长的用户访问量和产品的快速更新迭代，应用系统架构也经历了从简到繁、从单体架构到SOA架构再至微服务架构的演进过程。传统的集中式架构由于系统存在可维护性、扩展性、灵活性不足等缺点，而且对项目作进一步修改、开发、部署及测试的压力会不断增大，无法满足移动互联网时代业务快速增长和系统快速更新交付的需求。微服务分布式架构具备可扩展性、敏捷性和容错性的优势，同时，容器技术和容器编排管理平台的迅速发展为微服务架构的大规模使用提供了基础支撑，这使得微服务架构逐渐成为了目前最主流的应用解决方案。

二、微服务简介

（一）微服务概述

微服务架构是以业务功能为主的服务设计概念，系统以服务组件的方式将业务功能进行更细粒度的拆分，使单个业务功能组件化，系统由多个服务组件组成并可以单独部署，降低了系统各个服务组

作者简介：邢贞明（1991-），男，海南乐东人，助理工程师，供职于中国人民银行琼海市支行，研究方向：金融科技、大数据、信息安全等；
李登辉（1990-），男，河南禹州人，工学硕士，工程师，供职于中国人民银行海口中心支行，研究方向：金融科技、大数据、信息安全等；
潘 博（1994-），男，海南海口人，工学硕士，工程师，供职于中国人民银行海口中心支行，研究方向：金融科技、大数据、信息安全等。

收稿日期：2020-08-20

件之间的耦合度，使各个微服务可以使用不同的技术语言和存储技术开发实现，每一个服务对应单一业务功能。微服务轻量化的特征决定了微服务具备“小”“独”“轻”“松”等优点，具体见表1所列。

然而，由于微服务架构中各个微服务是分布式的，每个微服务需要独立部署，这大大增加了运维成本和服务管理的复杂性。同时，每个服务对应一个数据库，服务之间存在分布式事务的问题，难以保证多个微服务数据的一致性，此外，数据隔离也带来了报表问题。

（二）微服务与单体架构、SOA架构的对比

1. 单体架构。其将所有功能模块融合于一体，是一个统一的代码工程，容易开发、部署和维护。但是由于各个功能模块的耦合程度高，当应用系统工程过于庞大时，会伴随系统复杂度高、可维护性低、开发速度缓慢、难以扩展等问题，增加了对系统更新和升级的负担。

2. SOA架构。SOA架构是一个面向服务的架构，可将其视为组件模型，其将系统整体拆分为多个独立的功能模块，模块之间通过调用接口进行交互，有效整合了应用系统的各项业务功能，系统各个模块之间是松耦合的。SOA架构以企业服务总线链接各个子系统，是集中式的技术架构，应用服务间相互依赖导致部署复杂，应用间交互使用远程通信，降低了响应速度。

3. 微服务。微服务是SOA架构的进一步优化，去除了ESB企业服务总线，是一个真正意义上去中心化的分布式架构。其降低了微服务之间的耦合程度，不同的微服务采用不同的数据库技术，服务独立，数据源唯一，应用极易扩展和维护，同时降低了系统复杂性。架构如图1所示。

（三）微服务架构主流框架

微服务架构经多年的发展，并没有形成统一的技术规范，而是根据程序开发对框架的依赖程度，形成了以Dubbo和Spring CLoud为代表的侵入式架构和以Istio为代表的非侵入式架构两大主流框架。Dubbo是阿里开源出来的微服务化治理框架，其提供了自动服务注册和发现、高性能、智能负载均衡、RPC远程服务调用及SOA服务治理方案。Spring Cloud来自Spring公司，具有Spring社区的强大支撑，提供了一系列分布式基础设施的功能，如配置管理、服务发现、决策竞选、消息总线、负载均衡、智能路由、服务追踪等，是一个标准化的、全站式的分布式解决技术方案。Istio是一个服务网格（Service Mesh）的开源框架。服务网格是一个专注于服务间通信的基础设施层，其通过服务发现、负载平衡、检测、流控制来管理应用内部服务之间的数据共享和通信。Istio在逻辑架构上由数据平面和控制平面组成，提供了统一的连接、安全、管理和监控微服务的方案，与开发语言无绑定的以Istio为代表的服务网格框架被认为是微服务架构未来的发展趋势。

三、微服务相关技术及编排工具简介

微服务架构虽然能快速实现系统的更新迭代以满足业务需求，但由于系统微服务应用数量的增长，

表1 微服务的优点

体积小、架构小	微服务体积小、架构小。系统以业务功能为模块细粒化成一个个独立、专注的功能组件，每一个微服务是对单一职责的业务功能的封装，只专注于单一功能
单独的进程	单独的进程，微服务具备独立的运行进程，可以被单独部署，这种运行和部署方式提高了系统代码组织方式的灵活性，加快系统发布节奏，降低了对生产环境造成的风险
轻量级通信机制	轻量级通信机制，微服务之间采用轻量级通信机制，通常使用HTTP/REST接口进行交互，该通信方式不受语言和平台的限制，每个微服务都有足够的独立性，技术栈不受限，使系统更简洁轻量化，架构更灵活
松耦合性	松耦合性，服务之间是松耦合的，系统各个应用组件在功能和数据层面均采用相对独立的松耦合设计，每个微服务可以很灵活地按需扩展

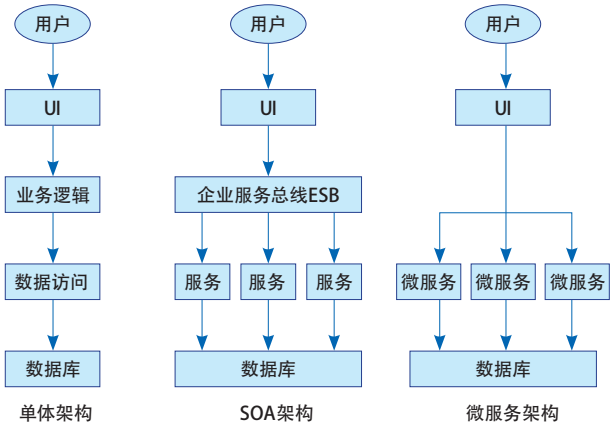

图1 单体架构、SOA架构和微服务架构

随之带来的是运维管理和测试成本的增加。容器技术的发展可消除本地、测试、线上环境的隔离，解决部署服务初始化繁琐的问题，应用容器编排管理平台加强了容器集群的管理，提供了完善的分布式系统支撑平台，成为了微服务应用的关键技术。

（一）容器技术

容器技术（Linux Container）是一种轻量级内核的操作系统层虚拟化技术，由Namespace和Cgroup两大机制实现。其中，Namespace负责进程资源的隔离，将内核的全局资源进行隔离封装，各个Namespace拥有独立的资源，不同的进程在各自的Namespace中对系统资源的使用不冲突。Cgroup属于Linux内核的一个特性，用于限制和隔离进程对系统资源的使用，负责对全局系统资源的管理和精细化控制。

容器的内核机制提供了容器之间互相联系的桥梁，同时，容器之间资源独立，互相隔离。容器技术的特点与微服务架构相辅相成，为微服务提供了良好的运行环境，大大提升了对微服务的运维管理效率，Docker是容器典型的代表。

（二）Docker技术应用特性

1. 快速交付和部署。Docker允许直接集成到可持续的开发流程中，支持应用程序版本的快速迭代，同时支持单个项目和整套项目的打包部署，项目的启动时间达到了秒级，大大地节约了开发、测试和部署时间。

2.环境一致性和高效扩容。Docker容器支持在任意平台上运行，不依赖于任何语言、框架和系统，这种兼容性支持了应用程序的部署平台的快速迁移，Docker具有的兼容性和轻量特性实现了对负载的动态管理，可快速扩容应用和服务，速度趋于实时。

3.资源利用率高。Docker具有高效系统资源利用率，除了运行其中的应用外，不消耗额外的系统资源，大大提高了系统的性能。

4. 更简单的更新管理。以增量的形式对应用程序所有的改动进行更新和发布，实现了自动化和高效的管理。

（三）基于Docker技术的微服务架构

基于Docker的技术特性，微服务体系很适用于Docker技术来架构。在微服务架构中，应用系统被拆分成小且松散耦合的分布式业务组件，Docker作为这些独立服务组件的部署单元，一个容器运行一个微服务，容器之间实行资源隔离，保证了微服务之间不会相互影响。由于每个容器对一个特定的微服务功能进行独立地部署与运维，因此整个微服务架构以业务单元的方式进行拆分后，以细粒度的方式部署在各个Docker容器中，大大提高了对微服务体系的管理和运维效率。

（四）Docker环境下容器编排工具的选择

采用Docker容器技术搭建的微服务架构，由于其技术特性导致生命周期短，加之应用部署密度的增长，使得对基础设施的监控愈加重要，需要被单独监控的事物以指数级数量增长，要想更好地使用容器，急需要优秀的容器编排管理平台。容器编排管理平台负责基于容器组成的分布式集群应用的管理工作，对容器进行监控和控制。容器编排工具用于协调创建和管理多个跨主机容器，并提供了功能强大的分布式集群解决方案。目前，主流容器编排管理平台是Kubernetes和Docker Swarm，由于Docker Swarm的调度策略均没有结合集群节点的实际情况进行容器调

度，且市场占有率和使用普及度不如Kubernetes，因此本文着重探析Kubernetes的技术特性。

Kubernetes是一个开源的容器集群管理系统，近几年来已经发展成为具有全球影响力的基础设施平台，推动了微服务架构和服务网格等热门技术的普及和落地。Kubernetes提供了一套完整的管理工具，包含了开发、部署测试、运维监控等各个节点，同时也是个完善的分布式系统支撑平台，具有完善的集群管理能力，涵盖了多租户应用支撑能力、服务注册和发现机制、智能负载均衡器、多层次的安全防护机制和在线扩容能力等一整套功能。Kubernetes实现了操作的自动化，用户由yaml部署到Kubernetes时，平台会根据应用程序计算资源需求，自动分配到node，大大简化了部署的流程。对于重启失效的容器节点，其自我修复功能可自动更替并重新调度容器。Kubernetes具有优质的动态资源分配机制，可实现Pod根据使用率的横向自动伸缩，并为每个容器分配IP地址和DNS，在容器之间实现服务发现和负载均衡。除此之外，Kubernetes还拥有可视化界面和稳定成熟的工具，为用户在集群管理层面提供了极大的便利。

四、人民银行部署微服务的建议

目前，基层央行多数信息系统都采用单体架构或以SOA架构为主的技术架构体系，传统技术架构难以适应新形势下金融创新业务系统的灵活扩展和大数据应用需求，信息系统亟须向平台化、微服务化、模块化转型。笔者根据微服务架构的特点，结合人民银行信息化建设的实际情况，提出以下建议。

（一）积极探索新技术落地，实现分布式架构转型

目前，基层央行多数使用虚拟机部署应用服务。与传统虚拟机相比，容器的轻量级和隔离特性能轻易实现轻量的应用运行环境，且拥有比虚拟机更高的硬件资源利用率，微服务结合容器部署，并选择容器编排工具如Kubernetes或以Kubernetes为底层技术的可视化平台进行管理，容器提供的资源分配和监控功能正好可以满足微服务的独立部署原则，结合容器集群管理工具类来管理容器，实现部署流程自动化、系统性能实时监控与应用运维智能化，提高运维管理效率。

（二）认真做好顶层设计，立足实际业务数据建模

由于微服务的数据库具有分布特性，一个微服务对应一个数据库，在对系统进行微服务改造或设计时需要对各微服务中的数据进行建模，明确组件之间数据共享的界限和范围。不同业务系统数据存在一定差异，在进行系统拆分时要以业务逻辑、可扩展、数据性能为原则，在数据建模过程中，在保证各业务互不影响的情况下找到数据治理和数据规划之间的平衡点。

（三）稳步推进队伍建设，建立微服务人才储备机制

在辖区范围内，调动专业性人才的积极性，构建一支可以独立开发、独立部署、独立发布并且采用去中心化管理的敏捷微服务团队。微服务具有“服务即业务“的特征，因此，要以业务为主技术为辅的原则构建团队，将业务和技术相结合，培养精通日常业务、掌握专业技术的复合型人才，建设统一的技术架构标准，实现对系统架构转型的规范化、流程化管理，推进分布式架构的转型。FTT

参考文献：

[1]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.

[2]姚刚，王从镔，吴海莉. 基于Docker技术的微服务架构探析[J]. 信息系统工程，2020(4):127-128.

[3]王义. 微服务架构特点、技术趋势及在行业应用中关键问题研究[J]. 软件，2020(6):132-136.

[4]邓杰文，曹彩凤. 微服务若干关键问题研究[J]. 五邑大学学报(自然科学版)，2016(2):49-54.

[5]赵然，朱小勇. 微服务架构评述[J]. 网络新媒体技术，2019(1):58-61.