

微服务架构在智能家居网关系统中的应用研究

吴磊, 湛健, 宋丽华

(北方工业大学 计算机学院, 北京 100144)

摘要:在智能家居环境中,用户的家庭设备信息保存在智能家居网关,并且可以定制和存储个性化情景模式的设备组合。针对当前云端智能家居网关管理系统在管理所有用户家庭的设备信息时系统越来越庞大和笨重,难以维护和迭代更新的问题,引入微服务架构。从微服务架构的起源与设计理念出发,相比传统的单体架构应用,介绍了微服务架构下系统的优势,并对微服务架构带来的挑战提出了应对措施。根据当前开发和维护的智能家居网关管理系统,结合实际业务情况,采用微服务架构,将系统划分为多个微服务,选用Spring-Cloud开源微服务治理框架,对智能家居网关管理平台进行总体的架构设计和功能设计。功能与性能测试表明,微服务架构下的系统可以满足业务需求并且提升了平台的可扩展性与稳定性。

关键词:智能家居;智能家居网关;微服务;Spring-Cloud

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2019)11-0200-06

doi:10.3969/j.issn.1673-629X.2019.11.040

Research of Application of Micro-service Architecture in Smart Home Gateway System

WU Lei, ZHAN Jian, SONG Li-hua

(School of Computer Science, North China University of Technology, Beijing 100144, China)

Abstract: In the smart home, the user's home device information is stored in the smart home gateway, and the combination of devices can be customized and stored in personalized scenario mode. Aiming at the problem that the current cloud smart home gateway management system is increasingly large and bulky when managing the device information of all users' homes, and is difficult to maintain and update iteratively, we introduce the micro-service architecture. Starting from the origin and design concept of micro-service architecture, we introduce the system advantages under the micro-service architecture compared with the traditional single-frame architecture, and propose countermeasures for the challenges brought by the micro-service architecture. According to the current smart home gateway management system developed and maintained, and combined with the actual business situation, the system is divided into several micro-services by adopting micro-service architecture. The Spring-Cloud open source micro-service governance framework is selected to carry out overall architecture design and functional design of the smart home gateway management platform. After functional and performance testing, the system under the micro-service architecture can meet the business needs and improve the scalability and stability of the platform.

Key words: smart home; smart home gateway; micro-service; Spring-Cloud

0 引言

随着科技的蓬勃发展,智能家居系统功能逐渐完善。智能家居最根本的目标是为人们提供舒适、方便、安全和高效的生活环境^[1]。智能家居网关是管理智能家居环境的核心设备之一,负责对智能家居设备信息的直接管理,包括信息采集、信息输入输出与集中控制等功能^[2]。智能家居网关管理系统集中管理用户家庭

的设备信息,并制定和存储个性化情景模式的设备组合信息^[3]。随着互联网的高速发展,智能家居网关管理系统给维护人员提供了非常便利的方式,对存储在网关中各个家庭设备的信息进行辅助维护。但是随着用户的增长,不同家庭对部署在云端的网关管理软件的需求发生快速变化,系统的单体架构已经无法适应互联网时代的快速变化,导致项目的开发、维护、

收稿日期:2018-11-30

修回日期:2019-03-29

网络出版时间:2019-06-27

基金项目:北京市社会科学基金(18JYB015);北京市自然科学基金-市教委联合重点项目(KZ201810009011)

作者简介:吴磊(1963-),男,硕士,副教授,研究方向为嵌入式技术应用及智能检测技术;湛健(1992-),男,硕士研究生,研究方向为嵌入式技术应用及微服务。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20190627.1105.042.html>

升级和扩展新功能都非常困难。因此需要一种基于组件化开发、分布式部署的架构策略来支撑网关管理系统的建设。目前比较流行的微服务架构正是解决该问题的理想方案。

微服务架构是面向服务架构思想的一种体现,旨在通过将功能分解到各个离散的服务中以实现对解决方案的解耦^[4]。近年来随着敏捷开发、DevOps、持续交付等 IT 建设理念的不断发展^[5]以及虚拟化、Docker 等技术的兴起和发展,微服务架构模式受到了学术界与工程界的极大关注,已经成为计算机科学领域的重点研究对象之一^[6]。

1 微服务

1.1 微服务的来源与设计理念

微服务最早是由国际著名的 OO 专家,敏捷开发方法的创始人之一 Martin Fowler 提出。微服务是将一个单体架构的系统拆分成若干小的服务,服务与服务之间采用 http 轻量协议传输数据,每个服务独立性强,这些服务基于业务能力构建,并能够通过自动化部署机制来独立部署。这些服务使用不同的编程语言实现,利用不同的数据存储技术,并保持最低限度的集中式管理,这样的设计实现了单个服务的高内聚,各个服务间低耦合的效果。

微服务架构虽然继承了面向服务架构(SOA)的特点,但在架构本质设计方面仍存在一定差别^[7]。SOA 着重于企业服务总线,在微服务中,各个服务松散、自治,服务间低耦合,强调服务的“独立性”——独立开发、独立部署和独立运行,减少应用开发的复杂性,从而具有灵活性大、易实施和可扩展等优点。

采用一组服务的方式来构建一个应用,服务独立部署在不同的进程中,不同服务通过一些轻量级交互机制来通信,例如 RPC、HTTP 等。服务可独立扩展伸缩,每个服务定义了明确的边界,不同的服务甚至可以采用不同的编程语言来实现,由独立的团队来维护^[8]。

基于微服务架构规划设计的办公平台,可以在不影响现有业务运行的情况下,将传统 PC 端复杂的应用分解成不同功能的微服务接口,快速重构适用于移动端的低耦合、可拓展、易伸缩的移动端应用。

1.2 微服务架构的优势

微服务的一些优势是显而易见的^[9]:

(1) 将一个复杂的业务拆分成若干个小的业务,每个业务拆分成小的服务,这样将复杂的问题简单化,服务的边界变得明确。服务按照业务拆分,编写的代码也是按照业务划分的,极大提升了程序的可读性与可拓展性。

(2) 微服务的每个服务单元都是独立部署的,也

就是各自都有自己独立的进程。当某个服务模块发生代码变更时,不需要重新编译、部署整个应用,这样单个服务的重新部署不会影响到其他服务,简化了测试与应用的部署。

(3) 由于微服务架构属于分布式系统,各个服务之间没有任何的耦合关系。随着业务的增加,之后的迭代开发可以根据业务继续拆分成微服务,所以微服务架构系统具有很强的横向拓展能力^[10]。并且开发出的微服务可以进行集群部署,以解决由于用户数量的增加带来的并发访问量问题。

(4) 微服务架构中每个拆分出来的微服务都可以由负责的团队独立开发,根据团队技术栈自由选择任何合适的开发语言来实现,只要服务符合指定的 API 协议。开发人员不再被强迫使用公司以前的技术或者已经过时的技术,而是可以根据业务场景自由选择最新最合适的技术来实现服务。

1.3 微服务架构的不足与应对措施

微服务架构虽然有很多明显的优势,但是任何事物的出现都是两面性的,关注其优势的同时也不能忽视其带来的一些额外的负担:

单体应用开发完成后可以打包部署在一组应用服务器上,然后消费端使用负载均衡即可。每个应用都有相同的基础服务地址,例如数据库和消息队列。而随着业务量的增加,拆分出的微服务会越来越多。每个服务都可能拥有自己单独的配置及服务地址,从而对配置、部署以及健康性监控提出更高的要求。

应对措施:DevOps 的快速发展,使得这一问题得到解决。容器能为应用提供隔离的运行空间和完整的运行环境^[11]。在智能家居网关管理平台中利用 Docker 容器为系统服务提供隔离的运行空间,使各容器内包含一个资源独享、完整封闭的运行环境并在云端部署。平台所拆分出的微服务采用轻量级的基于 HTTP 机制的 Restful 风格的 API 设计,各服务间通过网关负载均衡模块进行通讯,如图 1 所示。

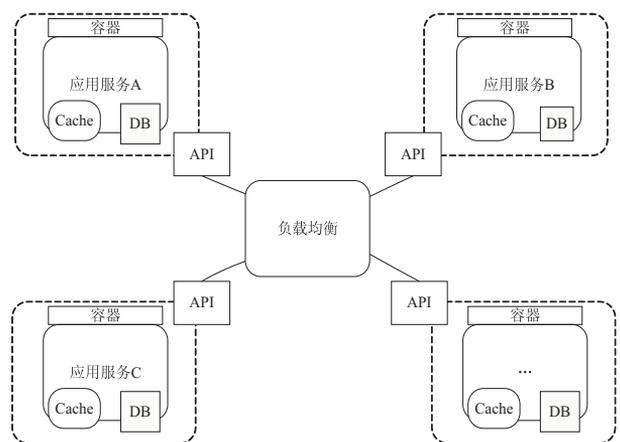


图 1 服务管理

微服务架构模式的应用中某个服务的改变由于服务间的调用关系可能会波及多个服务。在单体架构的应用中,需要对相关模块代码进行更改,整合变化,重新部署就可以。对比之下,微服务架构模式就需要考虑相关改变对不同服务的影响。许多改变一般只影响一个服务,而需要协调多服务的改变很少。

应对措施:采用异步消息通讯^[12],整合消息中间件,通过异步的生产者与消费者模式来保证服务间的数据一致性。

2 微服务智能家居网关管理平台的需求定位

伴随着互联网的发展,用户生活水平的提高,智能家居环境已经从以设备的安防和控制为中心发展到以人为中心涵盖舒适便捷、绿色节能等方面。智能家居网关管理系统需实现可以随时随地通过网络对家庭网关中的设备信息进行管理,该平台也是以此作为需求定位。

(1)集中式的需求定位。

用户家庭设备信息往往独立存储于用户家庭网关,这样管理人员对设备信息的维护往往都需要上门维护,非常不方便。这就需要有一个统一的管理中心,实现将分散的智能家居用户设备信息进行统一管理,提高管理效率。

(2)个性化的需求定位。

结合用户需求不同、天气以及在家状态,尊重用户意愿,定制多样的设备组合,实现情景模式的个性化定制。

(3)数据一致性的需求定位。

用户家庭设备信息保存在自己的家庭网关中,而管理人员统一管理的是存储在云端的设备信息,是用户家庭网关中数据的备份。在维护过程中为了保证双方的数据一致,需要采取相应的通讯措施。

3 微服务架构在智能家居网关管理平台中的应用

当前开发和维护的智能家居网关管理平台,随着用户的增加,不同用户的需求变化可能会出现单体架构应用解决不了的问题,从而引入微服务架构对该平台架构进行重构,提升平台的横向拓展能力。

智能家居网关管理平台是由服务人员通过获取用户授权来代替客户管理各自家庭网关中保存的设备数据。系统部署在云端,通过服务器端与家庭网关的通讯来保持两边数据的一致性。

3.1 服务划分

根据微服务架构原理^[13],与当前系统的实际情况,将智能家居网关管理平台进行拆分,划分出符合用户需求的定制性服务和平台的基础性服务。定制服务提供目前根据系统功能需求划分的能够实现基础业务功能的服务;基础平台服务目前包括日志服务与数据模块服务。系统划分出的微服务如表 1 所示。

表 1 微服务

序号	分类	服务名称
1	定制服务	用户服务
2		设备服务
3		情景模式服务
1	基础平台服务	操作日志服务
2		数据模块服务

各微服务都定义成独立的服务接口,相比于之前难以维护的单体结构的情况,新的智能家居网关管理平台由不同的微服务构成,每个服务只关注单一独立的业务功能,

3.2 技术架构选择

开发平台微服务时使用的技术架构如图 2 所示。

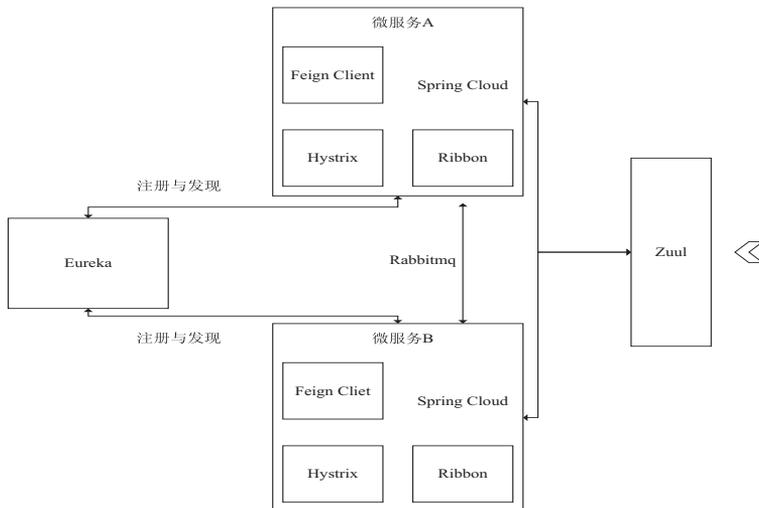


图 2 技术架构

其中微服务的基础开发框架选用 Spring-Boot。Spring-Boot 提供了各个服务敏捷开发的方式并包含了依赖注入、持久化和嵌入式容器等功能。服务的注册与发现通过集成 Eureka 组件进行实现;各个服务的客户端调用引入 Feign;Hystrix 组件实现各个服务的熔断机制与延迟;API 网关的路由选择,监控及负载均衡通过整合 Zuul 路由网关加以实现。最后通过各个功能框架组件的整合实现了平台微服务的设计开发和调用。

由于各个组件相互独立,为了开发方便,引入了 Spring-Cloud^[14] 组件。Spring-Cloud 是一个基于 Spring-Boot^[15] 实现的云应用开发工具,为基于 JVM 的云应用开发中的配置管理、服务发现、熔断器、动态路由、分布式与集群状态管理等操作提供了一种简单的开发方式。

3.3 总体框架

基于微服务架构的智能家居网关管理云平台架构如图 3 所示。

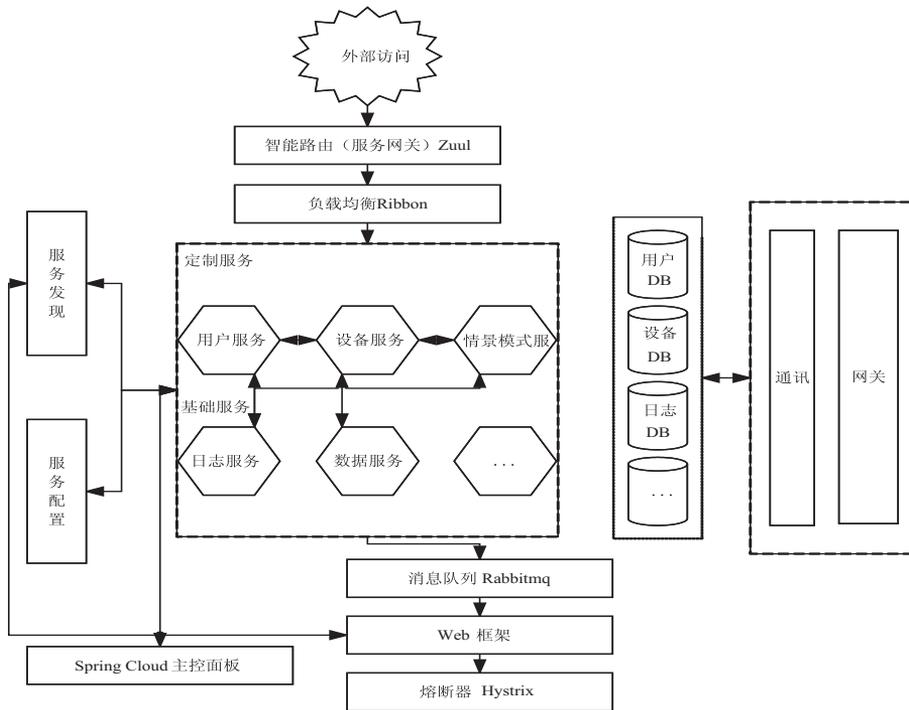


图 3 系统架构

进行基于微服务的智能家居网关管理平台的架构设计时,遵循“高内聚、松耦合”的设计思想,实现平台的开放性、软性和可扩展性。基于微服务架构理念,对系统提供的业务功能加以重构,使之解耦成相互独立且功能专一的服务,服务间以轻量级的通讯方式进行沟通,通过服务间的相互协作构建供外部访问的应用平台。

智能家居网关管理平台框架主要包括四层:

(1) 外部访问层:管理员获取用户授权,通过建立的 web 服务来聚合调用所需要的微服务接口,从而对授权家庭的网关设备信息加以维护。

(2) 服务网关:系统中拆分出每个微服务一般有不同的访问地址,在完成一个业务需求时客户端调用方可能需要调用多个微服务接口才能满足需要。如果客户端直接和拆分出的不同微服务进行通信,可能会存在以下问题:

① 客户端会多次请求不同微服务,增加客户端的复杂性;

② 存在跨域请求,在一定场景下处理相对复杂;

③ 认证复杂,每一个服务都需要独立认证;

④ 难以复用和重构,随着项目的迭代更新,可能需要根据业务需求重新拆分微服务,如果客户端直接和微服务通信,那么代码的重构将会变得难以实施;

⑤ 某些微服务可能使用了其他协议,直接访问有一定困难。

上述问题,都可以借助微服务 API 路由网关解决。微服务网关是介于调用端和接口提供端之间的中间层,所有的外部请求都会先经过微服务的 API 网关,这样调用端只需要和 API 网关交互,而无需直接调用特定微服务的接口,而且方便对所有微服务接口进行监控,并且易于权限认证,减少调用端和各个微服务接口之间的交互次数。

(3) 服务层:由拆分出来的多个相互独立、可拓展的微服务构成,包括业务相关的微服务与基础服务。业务微服务包括用户微服务、设备微服务、情景模式微服务等,基础微服务包括权限认证、日志管理等服务。

根据服务的重要性排序后,进行早期版本开发,服务根据业务需要封装扩展和实现迭代开发。在微服务架构中,每个服务根据实际情况采用各自独立数据库或共享数据库,采用 Mysql、redis 等数据库进行相关数据信息的存储。

(4)通讯层:智能家居网关管理平台是为了辅助维护用户家庭网关数据,要保证两边数据一致性,平台通过实验室制定的通讯协议,实现网关与平台的及时通讯,以实现异构数据库的增量同步。

智能家居网关管理平台的主要功能模块:

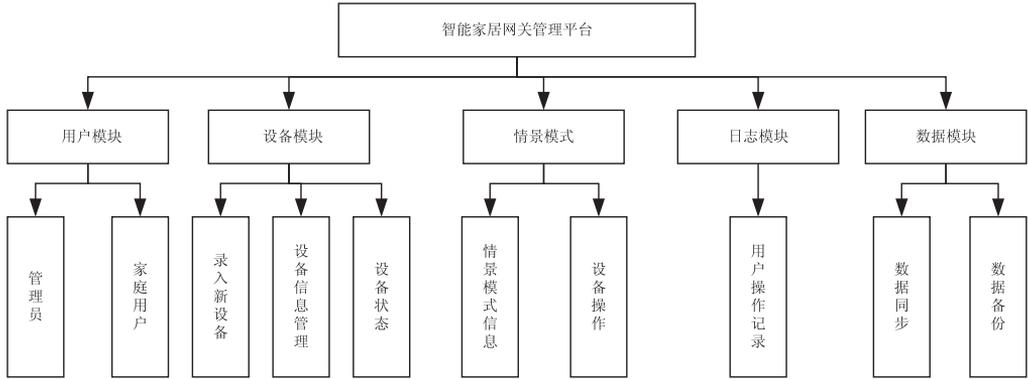


图 4 智能家居网关管理平台功能

用户模块:用户模块是管理网关管理平台的两类用户信息(管理员和家庭用户),模块包括注册、登录和用户授权三项子业务。平台管理员由系统预先设定,家庭用户信息由用户自行录入。在登录界面,管理员录入登录信息,申请用户授权,通过 API 接口获取返回的用户授权码,输入之后完成系统登录。

设备模块:该模块是管理用户家庭网关中的设备相关信息的微服务。包括新设备信息的录入与修改等功能。在管理员获取用户授权登录后,可以在智能家居网关管理平台上查看该用户家庭网关中已录入的设备信息,根据用户的意愿进行辅助维护。

情景模式模块:管理员获取用户授权后,根据当时天气情况、用户归家与离家状态以及用户需求,通过设备服务接口对用户家庭设备进行按需组合,从而一键控制设备的运行状态。

日志模块:记录平台登录人员的信息与操作记录,从而可以在出现某些人为操作失误后可以找出失误源头,从而可以回滚误操作结果。

数据模块:该模块定时对云端数据库数据信息在每日的凌晨 2 点到 4 点这段时间段内随机选择一个时间点,执行数据备份操作,将数据库中存放的所有设备信息和情景模式信息进行一次备份操作,备份的文件存放在默认备份目录中,以日期命名,且覆盖掉上个月同一天备份数据。同时可以与用户家庭网关进行通讯以保持两边的数据一致。

由于微服务架构中每个服务仅包含功能单一、独立、复杂度可控的业务,开发人员可根据自己的技术栈负责某个服务整个生命周期的开发和维护工作,从而保证整个应用系统的可维护性和开发效率,实现系统的快速迭代与升级。该智能家居网关管理平台结合互联网技术部署在服务器上,管理员通过获取用户授权与用户均可以远程管理用户自己家庭网关的设备数据。未来还可以根据不同家庭的各自需求,对系统平台做出快速的修改和升级。根据现有业务功能,将平台划分为以下模块,如图 4 所示。

4 实验与分析

4.1 实验环境

该实验在 1 MB/S 以太网的网络环境中进行测试,阿里云服务器(4 GB 内存、Windows Server 2012,处理器为 Intel 双核 2.60 GHz),软件环境有 Jdk1.8 的 Java 平台,数据库 Mysql 5.5.27,项目管理工具 Apache Maven 3.5.0,各微服务部署在 Docker 容器。记录智能家居网关管理系统在调用服务时产生的数据,包括当前调用的客户端、调用时间等等。

下面对比分析重构前方法调用与重构后微服务调用在性能方面的差异。以管理员登录耗时以及获取房间设备列表为例进行分析。

4.2 实验结果

4.2.1 管理员登录耗时对比分析

图 5 展示了在使用用户服务时,管理员进行登录,

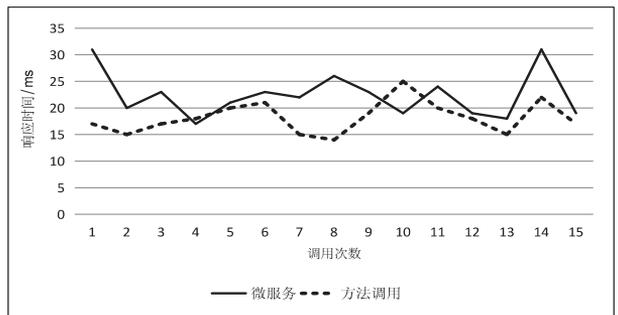


图 5 管理员登录耗时

智能家居网关管理系统通过方法调用与服务调用所消耗的时间,耗时差距平均基本维持在 20 ms 以内。

4.2.2 获取房间设备列表耗时对比分析

图 6 展示了在使用用户服务与设备服务时,根据用户家庭信息,获取房间设备列表,智能家居网关管理系统通过方法调用与服务调用所消耗的时间,耗时差距平均基本维持在 30 ms 以内。

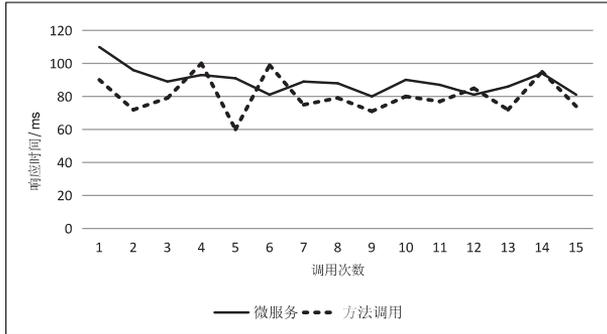


图 6 获取房间设备列表耗时

经过实验结果分析,重构后采用微服务调用与之前方法调用,实现功能耗时在可接受范围内。以微服务架构实现智能家居网关管理系统方案实现了内部逻辑的松耦合,服务可复用,提高了平台的可拓展性。

5 结束语

文中详细介绍了微服务架构的发展来源和概念,并结合传统的单体应用架构与微服务架构的对比,阐述了微服务架构的优点、劣势以及应对的方法,然后阐明了智能家居网关管理平台目前的实际需求,最后基于微服务架构设计并实现了服务高可用、可拓展的智能家居网关管理平台。

目前在微服务架构的实践中,仍处于起步阶段,还有很大的发展空间,在后续智能家居网关管理平台建设中将不断摸索并完善。

参考文献:

- [1] 陈思运,刘 焱,沈 超,等.基于可穿戴设备感知的智能家居能源优化[J].计算机研究与发展,2016,53(3):704-715.
- [2] 林子雨,赖永炫,林 琛,等.云数据库研究[J].软件学报,2012,23(5):1148-1166.
- [3] 吴 磊,原 鹏,丁维龙.智能家居网关与云服务器数据同步协议的研究[J].计算机技术与发展,2018,28(9):151-155.
- [4] 杨俊伟,纪 鑫,胡强新.基于微服务架构的电力云服务平台[J].电力信息与通信技术,2017,15(1):8-12.
- [5] 王 磊.微服务架构与实现[M].北京:电子工业出版社,2015.
- [6] BALALAE A, HEYDARNOORI A, JAMSHIDI P. Microservices architecture enables devops: migration to a cloud-native architecture[J]. IEEE Software, 2016, 33(3):42-52.
- [7] 辛园园,钮 俊,谢志军,等.微服务体系结构实现框架综述[J].计算机工程与应用,2018,54(19):10-17.
- [8] VIDELA A, WILLIAMS J J W. RabbitMQ 实战:高效部署分布式消息队列[M].汪佳南,译.北京:电子工业出版社,2015.
- [9] 张 晶,黄小锋.一种基于微服务的应用框架[J].计算机系统应用,2016,25(9):265-270.
- [10] 方志朋.深入理解 Spring Cloud 与微服务构建[M].北京:人民邮电出版社,2018.
- [11] 陆远蓉.微服务架构下的移动教学平台的规划和设计[J].工程技术研究,2017(7):234-236.
- [12] 李春阳,刘 迪,崔 蔚,等.基于微服务架构的统一应用开发平台[J].计算机系统应用,2017,26(4):43-48.
- [13] THONES J. Microservices[J]. Software, 2015, 32(1):113-116.
- [14] COSMINA I. Spring microservices with spring cloud[M]// Pivotal certified professional spring developer exam. Berkeley, CA: Apress, 2017:435-459.
- [15] WALLS G. SpringBoot in action[M].北京:人民邮电出版社,2016:42-64.