

# 基于微服务架构的充电桩管理平台

吴洁, 周杰, 符叙伦

(深圳市英可瑞科技股份有限公司, 广东 深圳 518000)

**摘要:** 随着充电桩管理平台业务和功能的迅速增加, 传统的单块架构模式处理起来显得过于复杂, 同时得面对开发效率低, 代码后期维护难等问题。结合当前网络技术, 移动应用技术以及云技术, 通过建立统一的服务间通信机制以及服务注册与发现机制、服务网关等组件, 提出并实现了基于微服务架构的充电桩管理平台。最后对微服务架构下的充电桩管理平台框架设计进行了分析与探讨。充电桩管理平台的微服务化方案解决了平台存在的标准不统一、结构复杂等问题, 为实现平台的低耦合、易伸缩、高可扩展提供了坚实的技术支撑。

**关键词:** 微服务; 管理平台; 充电桩; 系统架构

## Charging pile management platform based on microservice architecture

WU Jie, ZHOU Jie, FU Xulun

(Shenzhen Yingkere Technology Co., Ltd., Shenzhen 518000, China)

**Abstract:** With the rapid increase of business and functions of charging pile management platform, the traditional single-block architecture model is too complex to deal with. At the same time, it has to face the problems of low development efficiency and difficult maintenance of code in the later stage. Combining the current network technology, mobile application technology and cloud technology, a charging pile management platform based on micro-service architecture is proposed and implemented by establishing a unified inter-service communication mechanism, service registration and discovery mechanism, service gateway and other components. Finally, the framework design of charging pile management platform under micro-service architecture is analyzed and discussed. The micro-service scheme of charging pile management platform solves the problems of inconsistent standards and complex structure, and provides a solid technical support for realizing low coupling, easy scalability and high scalability of the platform.

**Key words:** microservice; management platform; charging pile; system architecture

## 1 概述

随着国家鼓励新能源汽车的发展, 充电桩的规模会越来越大, 业务和功能越来越复杂, 相应的管理平台原本简单的应用体积会随着变大, 导致维护、升级和扩展新功能都非常困难; 如果要更新它的某个部分, 需要重新部署整个应用。传统单块平台架构耦合性强, 很难适应应用变更的需求。相同业务需求因业务管理模块分割、管理边界重复交叉、管理方式各异等原因导致系统功能重复建设等<sup>[1]</sup>。因此, 需要一种基于模块化开发, 分布部署的架构策略支撑充电桩管理平台的建设。为构建具有易维护, 易扩展的充电桩管理平台, 提出了基于微服务架构的系统解决方案, 本文对微服务架构的应用及优点进行了分析, 并围绕微服务架构下的平台架构, 微服务封装和交互机制给出具体规划设计<sup>[2]</sup>。

## 2 微服务简介

### 2.1 微服务架构概念

微服务架构(Microservice Architecture)是一种架构概念, 旨在通过将功能分解到各个离散的服务中以实现解决方案的解耦。把一个大型的单个应用程序和服务拆分为数个甚至数十个的支持微服务, 它可扩展单个组件而不是整个的应用程序堆栈, 从而满足服务等级协议<sup>[3]</sup>。

### 2.2 微服务架构特点

1) 通过服务实现应用的组件化: 在应用架构设计中, 通过将整体应用切分成可独立部署及升级的微服务方式, 进行组件化设计。

2) 围绕业务能力组织服务: 微服务架构采取以业务能力为出发点组织服务的策略, 因此微服务团队的组织结构必须是跨功能的, 强搭配的 DevOps

开发运维一体化团队,通常这些团体不会太大<sup>[4]</sup>。

3) 基础设施自动化:云化及自动化部署等技术极大地降低了微服务构建,部署和运维的难度,通过应用持续集成和持续交付等方法有助于达到加速推出市场的目的。

4) 故障处理设计:微服务架构所带来的一个后果是必须考虑每个服务的失败容错机制。因此微服务非常重视架构以业务相关指标的实时监控和日志机制。

5) 演进式的设计:微服务应用更注重快速更新,因此系统的设计随时间不断变化及演进<sup>[5]</sup>。

### 2.3 微服务的优势

1) 原子服务:微服务架构的主要思想是将庞大的整体系统在整体功能不变的提前下分解成具有单一功能的一系列服务,每个服务的功能越单一,对其他服务的依赖越少。

2) 独立部署:由于微服务具备独立的运行进程,所以每个微服务也可以独立部署。在传统架构中,如果要对应用程序中的某个服务进行需求变更。及时改变再小也要对就整体架构进行重新部署。而当要对应用微服务架构的某个服务进行需求变更时,可独立构建并重新部署而不依赖其他微服务。可以利用 docker 等虚拟机在云端进行部署,可以提高资源利用率,从而降低部署成本<sup>[6]</sup>。

3) 技术灵活:在微服务架构中,技术选型不需要再完全同一。传统的一体化运用方式,往往需要采用某种确定的技术栈。而在微服务架构模式下,技术选型是去中心化的,将庞大的整体应用拆分为微服务后,由于每个服务有自己独立的应用架构和数据存储,可以根据不同服务和业务需求的特点选择最合适的技术栈,技术实现平台和数据库方案。

4) 扩展及容错:在微服务架构下,每个服务都可以根据需求变化进行独立扩展。在微服务架构下,由于服务间松耦合的特点。故障被隔离在微服务内部。可以采用超时重试,多副本策略等容错机制。避免全局不可用<sup>[7]</sup>。

### 2.4 微服务解决方案 Spring Cloud 简介

Spring Cloud 为开发者提供了在分布式系统中快速构建的工具,它提供了微服务开发所需的配置管理、服务发现、断路器、智能路由、微代理、控制总线、全局锁、决策竞选、分布式会话和集群状态管理等组件。使用 Spring Cloud 的开发者可以快速的启动服务或构建应用,同时能够快速和云平台资源进行对接<sup>[8]</sup>。SpringCloud 五大核心组件介绍如下。

Eureka: 服务注册中心,特性有失效剔除、服务保护。

Zuul: API 服务网关,功能有路由分发和过滤。

Config: 分布式配置中心,支持本地仓库、SVN、Git、Jar 包内配置等模式。

Ribbon: 客户端负载均衡,特性有区域亲和、重试机制。

Hystrix: 客户端容错保护,特性有服务降级、服务熔断、请求缓存、请求合并、依赖隔离。

Dashboard: Hystrix 仪表盘,监控集群模式和单点模式。

## 3 电桩管理平台框架设计

### 3.1 业务功能划分

微服务充电桩管理平台在功能设计方面,需要根据实际情况,将原有系统的功能进行充分解耦,形成小服务群,这些小的服务彼此独立,单独部署,并都可以完成自身的任务。每个小服务都是从交互,业务逻辑在到底层存储,独立形成一个应用。每个小应用之间通过相互调用 API 实现通信<sup>[9]</sup>。

1) 数据采集系统:是整个管理平台运行的基础,主要负责采集充电桩数据和移动 APP 用户数据,并将这些信息实时写入数据服务器,供各个管理模块调用。

2) 充电桩运维管理:充电桩是管理平台的主要实体,也是提供服务的基础。对充电桩的运维管理主要分为充电桩注册,充电桩的实时监控,充电服务实时监控,远程配置参数,告警处理,费率下发,远程升级等。

3) 平台运维管理:平台维护是系统运营过程中的一个重要方面,它能对系统平台的日常工作状况进行跟踪调查,能够解决系统平台在运营过程中产生的问题,确保系统能够提供各项服务。运营平台的维护主要包含软件维护,数据维护,日志管理,充电桩维护等方面。

4) 充电支付结算:实现充电费用实时计算,充电完成后进行费用结算,支付以及发票等功能<sup>[10]</sup>。

5) 系统用户管理:运营系统中的用户大致可以分为系统管理员、监控人员、运维人员和充电用户等角色。各角色拥有不同的权限和功能,彼此之间分工明确,相互联系,保证系统正常的运行。

### 3.2 容器技术的应用

传统的虚拟化技术为每个虚拟机都虚拟出一套完整的操作系统。部署应用时需要为每个虚拟系统都配置运行环境应用在微服务上时,由于一个应用系统被分解成不同服务,部署及维护难度将大幅升高。容器技术是一个可移植,跨平台的解决方案,将它用于微服务可以大幅提高部署以及维护的效

率。容器技术调用内核接口，通过内核共享的方式运行，容器之间共享内核并完全隔离<sup>[5]</sup>。从移植的图中可以看出，虚拟容器占用了更少的物理资源，增大了发布能力，同时虚拟容器技术为系统级虚拟技术，启动和停止相对于传统虚拟技术更快，也使部署在上面的微服务比传统虚拟技术有更好的移植性。微服务之间通过暴露的 API 供其他接口调用，实现微服务之间的进行通信，接口由 URL 请求地址、请求、相应消息等组成<sup>[11]</sup>。

### 3.3 架构设计实现

充电桩管理平台架构设计，目标是将现有的充电桩管理平台的业务服务进行拆分，独立成微服务，建设成一个个高性能独立的子系统聚合的平台。

基于当下业界最火的 Spring Cloud 去实现这微服务的架构。微服务整体架构如图 1。

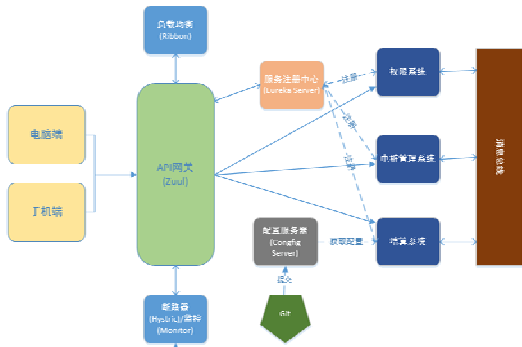


图 1 微服务整体架构

Fig. 1 Micro service overall architecture

下面详细介绍微服务整体架构的主要特点。

#### 3.3.1 安全认证

整体系统的安全由 API 网关控制，ss 鉴权逻辑统一网关过滤器判断。当用户未认证时，网关路由到登录接口，登录成功生成 TOKEN，返回给调用方。后面 TOKEN 作为用户凭证去控制其权限<sup>[12]</sup>。过程如图 2。

#### 3.3.2 高可用注册中心

服务注册中心在这个架构中显得尤为重要，所有的服务提供者和发布者都依赖注册中心，所以采用主备的方式多注册中心，每个注册中心的收到服务注册/续约/下线等请求都会转发给其他注册中心，始终保持注册中心状态一致<sup>[13]</sup>。服务注册流程如图 3。

#### 3.3.3 接口超时容错处理

通过服务熔断器 Hystrix(也可以称为断路)、降级、限流(隔离)、异步 RPC 等手段控制依赖服务的延迟与失败。

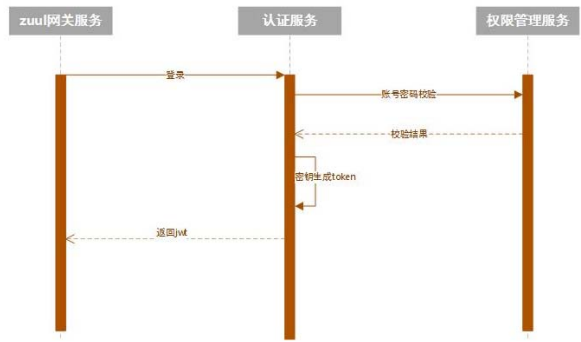


图 2 网关服务鉴权流程

Fig. 2 Gateway service authentication process

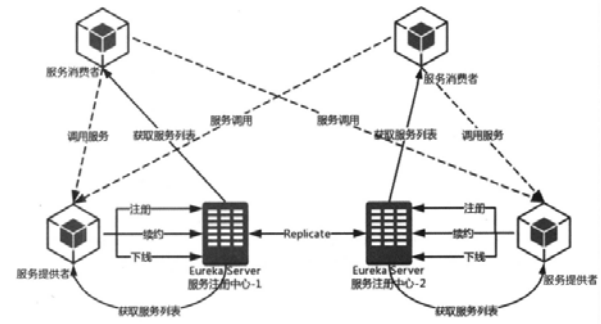


图 3 服务注册流程

Fig. 3 Service registration process

熔断器为每一个依赖服务维护一个线程池(或者信号量)，当线程池占满，该依赖服务将会立即拒绝服务而不是排队等待。

每个依赖服务都被隔离开来，熔断器会严格控制其对资源的占用，并在任何失效发生时，执行失败回退逻辑<sup>[14]</sup>。

熔断器开关由关闭到打开的状态转换是通过当前服务健康状况和设定阈值比较决定的，服务的健康状况=请求失败数/请求总数。Hystrix 服务调用的内部逻辑如图 4。

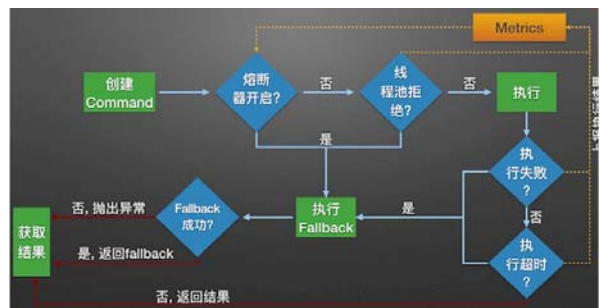


图 4 Hystrix 服务调用的内部逻辑

Fig. 4 Hystrix internal logic of service invocation

### 3.3.4 服务状态监控(图 5)

利用熔断器衡量健康状况的度量指标, 对这些指标进行汇总, 整合到 Hystrix DashBoard, 监控服务状态, 为运维提供帮助。

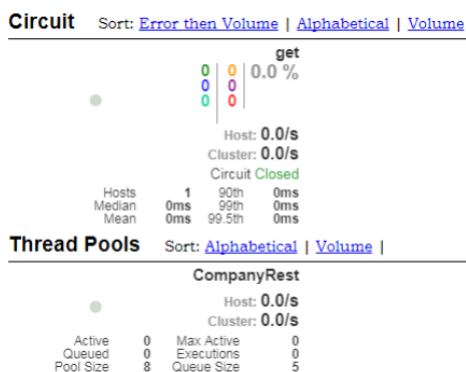


图 5 服务状态监控

Fig. 5 Service state monitor

### 3.3.5 充电桩通讯服务改造

针对电桩是 TCP 直连到服务器, 设计时进行了优化, 把和电桩通讯的服务全部独立出去, 与平台业务系统隔离。充电桩和服务器对应关系存在 ConfigServer, 网关自定义网关路由规则, 实现平台动态向桩发送控制命令<sup>[15]</sup>。电桩通讯服务流程如图 6。

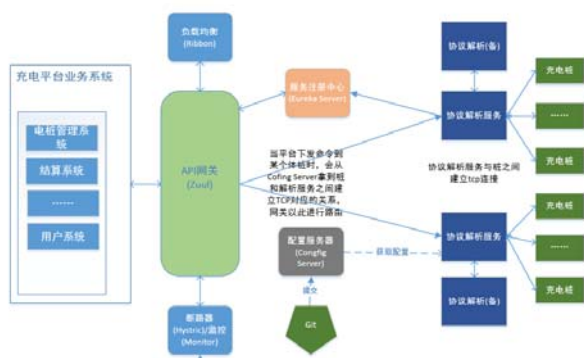


图 6 电桩通讯服务流程

Fig. 6 Electric pile communication service flow

## 4 结论

将微服务架构应用于充电桩管理平台的优点显著, 采用基于虚拟容器的微服务架构来构建平台, 能将软件从硬件中分离出来, 既增加了系统维护下, 扩张性, 易于部署管理, 又使得技术路线可以灵活应用, 同时还解决了平台下物联网异构问题。除此之外, 其分布式设计又易于通过 API 实现跨服务通信。从而, 可以将充电桩相关应用系统整合到充电

桩管理平台系统中, 提供充电桩所需的各类信息化应用, 满足后期充电桩行业发展需求。

## 参考文献

- [1] 黄小锋, 张晶. 微服务框架介绍与实现[J]. 电脑与信息技术, 2016, 24(6): 14-16.  
HUANG Xiaofeng, ZHANG Jing. Introduction and implementation of microservice architecture[J]. Computer and Information Technology, 2016, 24(6): 14-16.
- [2] 黄嘉诚, 董晶. 基于微服务的智能档案服务系统设计与实现[J]. 电子设计工程, 2017, 26(2): 26-30.  
HUANG Jiacheng, DONG Jing. Design and implementation of intelligent archive service system based on micro-service[J]. Electronic Design Engineering, 2017, 26(2): 26-30.
- [3] 蒋勇. 基于微服务架构的基础设施设计[J]. 软件, 2016, 37(5): 94-97.  
JIANG Yong. Design of infrastructure based on micro-services architecture[J]. Computer Engineering & Software, 2017, 37(5): 94-97.
- [4] 胡启敏, 薛锦云, 钟林辉. 基于 Docker 的 DevOps 系统设计与实现[J]. 指挥信息系统与技术, 2017, 3(8): 87-93.  
HU Qimin, XUE Jinyun, ZHONG Linhui. Design and implementation of DevOps system based on Docker[J]. Command Information System and Technology, 2017, 3(8): 87-93.
- [5] 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): 9-12.  
YANG Junwei, JI Xin, HU Qiangxin. SPower cloud service platform based on microsoft service architecture[J]. Electric Power Information and Communication Technology, 2017, 15(1): 9-12.
- [6] 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.  
XIN Yuanyuan, NIU Jun, XIE Zhijun, et al. Survey of implementation framework for microservices architecture[J]. Computer Engineering and Applications, 2018, 54(19): 10-17.
- [7] 杨鸥, 张羿, 耿贞伟. 微服务架构在容器云中的应用实践[J]. 电脑与电信, 2017(7): 79-81.  
YANG Ou, ZHANG Yi, GENG Zhenwei. Application of microservice architecture in container cloud[J]. Computers and Telecommunications, 2017(7): 79-81.
- [8] 胡启敏, 薛锦云, 钟林辉. 基于 Spring 框架的轻量级 J2EE 架构与应用[J]. 计算机工程与应用, 2008, 44(5): 115-118.  
HU Qimin, XUE Jinyun, ZHONG Linhui. Lightweight

- J2EE architecture based on spring framework and its application[J]. Computer Engineering and Applications, 2008, 44(5): 115-118.
- [9] 李洪峰, 李红霞, 陈志刚, 等. 一种新型电动汽车充电桩技术方案探讨[J]. 电力系统保护与控制, 2017, 45(6): 142-147.  
LI Hongfeng, LI Hongxia, CHEN Zhigang, et al. Discussion on technology scheme of a new EV charging pile[J]. Power System Protection and Control, 2017, 45(6): 142-147.
- [10] 杨校辉, 张娟, 史志鸿, 等. 电动汽车充电桩认证结算单元的设计与实现[J]. 电力系统保护与控制, 2018, 46(11): 118-123.  
YANG Xiaohui, ZHANG Juan, SHI Zhihong, et al. Design and implementation of the EV charging pile certification settlement unit[J]. Power System Protection and Control, 2018, 46(11): 118-123.
- [11] 徐琛杰, 周翔, 彭鑫, 等. 面向微服务系统的运行时部署优化[J]. 计算机应用与软件, 2018, 35(10): 86-93.  
XU Chenjie, ZHOU Xiang, PENG Xin, et al. Runtime deployment optimization for micro-service systems[J]. Computer Applications and Software, 2018, 35(10): 86-93.
- [12] 庄璐, 路学刚. 微服务架构中认证与鉴权的探讨[J]. 金融科技时代, 2018(10): 40-42.  
ZHUANG Lu, LU Xuegang. Discussion on authentication and authentication in microsoft architecture[J]. Financial Technology Time, 2018(10): 40-42.
- [13] 冯显力, 韦化, 韦洪波, 等. 含微服务的调度自动化系统分布式实时数据库[J]. 电力系统保护与控制, 2018, 46(21): 138-144.  
FENG Xianli, WEI Hua, WEI Hongbo, et al. Distributed real-time database for dispatching automation system with micro-services[J]. Power System Protection and Control, 2018, 46(21): 138-144.
- [14] 崔蔚, 李春阳, 刘迪, 等. 面向微服务的统一应用开发平台[J]. 电力信息与通信技术, 2016(9): 12-17.  
CUI Wei, LI Chunyang, LIU Di, et al. Unified application development platform for micro service[J]. Electric Power Information and Communication Technology, 2016(9): 12-17.
- [15] 郑明钊, 张建强. 基于微服务的大平台系统架构演进探讨[J]. 软件, 2017, 38(12): 165-169.  
ZHENG Mingzhao, ZHANG Jianqiang. Discussion on the evolution of big platform system architecture based on microservice[J]. Computer Engineering & Software, 2017, 38(12): 165-169.

---

收稿日期: 2018-12-01

作者简介:

吴 洁(1989—), 男, 学士, 工程师, 研究方向为应用软件; E-mail: 291218887@qq.com

周 杰(1990—), 男, 学士, 工程师, 研究方向为应用软件; E-mail: 418087535@qq.com

符叙伦(1990—), 男, 学士, 工程师, 研究方向为应用软件。E-mail: 13790403112@163.com