

基于微服务架构的高铁列控仿真测试系统的研究与设计

陈汇远

(中国铁道科学研究院集团有限公司通信信号研究所 北京 100081)

摘要 为保障高铁列控车载设备安全可靠运行,有必要对其进行充分全面测试和验证。针对既有列控仿真测试系统研发的可用性、可扩展性和可维护性低的问题,提出采用面向微服务架构构建高铁列控仿真测试系统。在深入探讨系统结构和主要功能的基础上,建立基于微服务架构的系统模型。通过采用层次分析法对模型各层的工作原理和模型实现的关键技术进行深入研究。根据实验验证与分析,基于微服务架构建模的高铁列控仿真测试系统具备高可用性、扩展性、可靠性和可维护性。

关键词 高铁列控 仿真测试 微服务

中图分类号 TP391.9

文献标志码 A

DOI:10.3969/j.issn.1000-386x.2024.07.048

STUDY AND DESIGN OF A SIMULATION TEST SYSTEM FOR HIGH-SPEED RAILWAY TRAIN CONTROL BASED ON MICROSERVICE ARCHITECTURE

Chen Huiyuan

(Signal & Communication Research Institute, China Academy of Railway Science Corporation Limited, Beijing 100081, China)

Abstract In order to ensure the reliability and safety of high-speed railway train control on-board equipment, it is necessary to conduct full and comprehensive testing and verification. Aimed at the low usability, scalability and maintainability of the existing train control simulation test system, the high-speed railway train control simulation test system based on microservice is proposed. On the basis of deep discussion over the systematical architecture and main function, the architecture of the systematical model based on microservice was built. The working principle of each layer of the model and the key technology of the model implementation were deeply studied by using the analytic hierarchy process. According to the experimental verification and analysis, the high-speed railway train control simulation test system based on microservice has high availability, high expandability, high reliability and high maintainability.

Keywords High-speed railway train control Simulation test Microservice

0 引言

高速铁路列控系统是提高铁路运营效率、保障铁路运输安全的核心装备。车载设备是列控系统的关键部分,它的运行直接关系旅客生命和财产安全。为了保障车载设备安全可靠地运行,因此有必要对列控系统车载设备的系统功能进行充分全面的测试和验证。

目前,列控仿真测试系统仿真建模实现方式主要有

两种。一种是采用基于形式化语言的仿真系统^[1-4];另一种是采用基于模型驱动方式的仿真系统^[5-8]。基于形式化语言的仿真主要用于系统规范和逻辑流程的验证,其本质是通过数学方法描述目标软件系统属性。基于模型驱动的仿真主要用于系统方案验证和系统功能测试。当今,基于模型驱动的列控仿真测试系统占据了列控仿真测试系统应用的75%,这种基于计算机程序开发的模型构建仿真系统的方式是实现仿真资源可重用和可组合的基本条件^[9]。当前的列控测试系统仿真建模做了大量研究,有的通过软件实现了模型的

应用^[5-8],但大多数研究采用传统单体架构设计,代码耦合性强,资源可重用性和业务扩展性低,无法实现对既有仿真资源进行有效管理和整合。此外,因为列控仿真测试系统涉及大量软、硬件开发,结构和接口复杂,既有的仿真测试系统主要针对列控系统车载设备不同厂商的硬件接口和协议分别编写测试脚本,测试用例通用性差,系统的可用性、可扩展性和可维护性低。如何行之有效地实现列控仿真测试系统高可用性、可靠性、可扩展性和可维护性是迫切需要解决的问题,也是本文研究的最终目的。

1 微服务架构

1.1 微服务定义和特点

微服务指围绕系统业务功能,构建能够协同工作的小而自治的服务,这些独立服务通过组合共同构成整个应用系统。微服务根据系统业务划分,具有高内聚、高扩展和高自治性特点,使得系统易于扩展和维护^[10]。

1.2 微服务架构优点

微服务架构是近年来新兴的软件架构,它将一个大型的应用程序分解成若干个微服务,其本质是通过将应用程序的功能细化,拆分到各个离散的服务中,微服务之间相对独立且松耦合,实现对解决方案的解耦和降低了系统的复杂度。微服务架构具备服务化、组件化、分散化和独立部署等优点,它将应用程序通过组件化的方式分解为多个微服务,使得系统更易开发和维护。微服务系统实质是分布式系统,随着系统新业务增加,可以将现有的服务单元进一步拆分,也可以根据新增需求添加新的服务单元,提高了系统的可扩展性。此外,微服务架构强调去中心化的软件组织架构,每个微服务都可独立测试、部署和运行,单个服务的变更不会影响到其他服务,持续化部署和维护更加便利,实现了系统高可靠性、可用性和可维护性^[11]。

2 系统构成及主要功能

2.1 构成

高速铁路列控仿真测试系统是个半实物仿真系统,它包括主控模块、无线消息处理模块和硬件模块,其系统结构如图1所示。

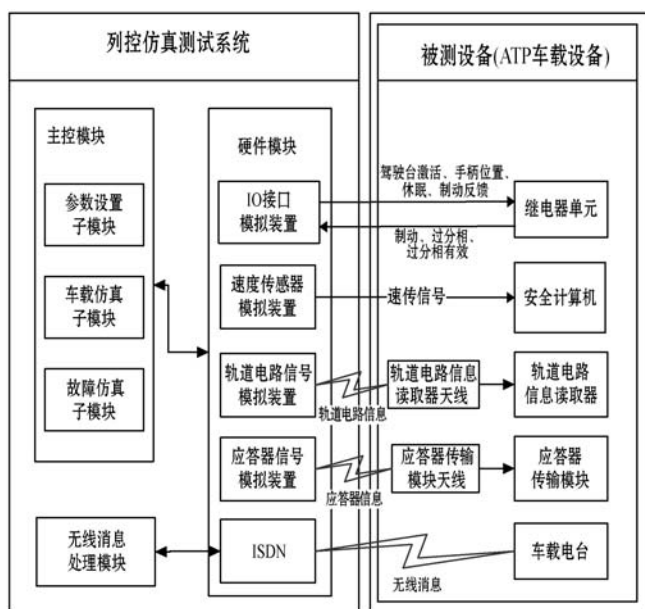


图1 高速铁路列控仿真测试系统结构

2.2 主要功能

2.2.1 主控模块

用于设置列车参数以及与硬件模块通信的参数。加载、检测并执行列控工程脚本文件,通过模拟列车速度信号、车辆输入输出(Input Output, IO)信号、轨道电路信号和应答器信号计算列车的位置和速度。通过故障注入模拟制动反馈故障、速度差故障、手柄故障、轨道电路载频低频异常信息和应答器报文异常信息验证列车自动防护(Automatic Train Protection, ATP)车载设备在异常条件下的功能处理。

主控模块包括参数设置子模块、车载仿真子模块和故障仿真子模块。

1) 参数设置子模块。用于设置列车参数以及与硬件模块通信的参数。列车参数主要包括:列车型号、轮径值、齿轮数、列车长度、列车实时信息发送的通信参数。与硬件模块通信的参数主要包括:与硬件模块中IO接口模拟装置通信参数、与速度传感器模拟装置通信参数、与轨道电路信号模拟装置通信参数、与应答器信号模拟装置通信参数。

2) 车载仿真子模块。用于模拟地面线路仿真信号和车辆IO功能。通过加载与校核列控工程数据脚本文件产生列车地面线路数据仿真信号;根据列车速度计算列车位置,并通过列车位置逐条执行脚本指令;同时具备模拟车辆IO功能。列控工程数据脚本文件主要包括列车位置、相对距离、载频、低频、区段名称、区段长度等。所产生的仿真信号主要包括:车辆输入输出信号、轨道电路信号、应答器信号、速度传感器信号。通过IO接口模拟装置接收车载设备的制动信息、切除牵引、过分相、过分相有效等信号;通过IO接口模

拟装置向 ATP 车载设备发送方向手柄位置、牵引手柄位置、制动手柄位置、驾驶室激活、休眠、制动反馈等信号。

3) 参数设置子模块。用于实现故障模拟功能,产生列车 ATP 的故障信号。列车故障不仅包括紧急制动反馈故障、常用制动反馈故障、速度跳变、速度差故障等,也包括模拟牵引手柄、制动手柄、方向手柄的状态故障等。

2.2.2 无线消息处理模块

用于模拟无线闭塞中心(Radio Block Center, RBC),加载并校验无线脚本文件,执行无线消息序列,通过硬件模块中的综合业务数字网(Integrated Services Digital Network, ISDN)服务器与真实 ATP 进行无线消息交互,实现 ATP 在 C3 等级下控车。无线脚本文件主要包含列车位置、RBC 发送的无线消息、RBC 接收的无线消息、无线消息长度、无线消息内容和无线消息传输方向等信息。

2.2.3 硬件模块

包括 IO 接口模拟装置、速度传感器模拟装置、轨道电路信号模拟装置、应答器信号模拟装置和 ISDN 服务器。其中:IO 接口模拟装置用于模拟列车的手柄位置、驾驶室激活、休眠、制动反馈、制动、过分相等功能;速度传感器模拟装置,用于模拟列车的速度脉冲信号;轨道电路信号模拟装置,用于模拟地面线路的轨道电路信号;应答器信号模拟装置,用于模拟地面线路的应答器信号;ISDN 服务器用于 RBC 与 ATP 车载设备的无线信息交互。

3 系统设计

3.1 微服务架构设计原则

设计出一种良好的微服务架构需遵循以下四个原则:单一职责、服务自治、轻量级通信方式和接口明确。根据这些设计原则,微服务基础框架应包括:服务注册、服务发现、服务通信、日志管理和接口管理等功能。在此基础上为了增强微服务架构的完整性及先进性应加入限流容错、负载均衡、服务配置、服务监控等功能。这些不同的服务功能相互组合使用,保证了微服务架构的高效、稳定、可移植和跨平台等特性^[12]。

3.2 体系结构

综合微服务架构的设计原则、优点和仿真测试系统的主要功能,设计出基于微服务架构的高速铁路列车控制仿真测试系统。微服务架构如图 2 所示。

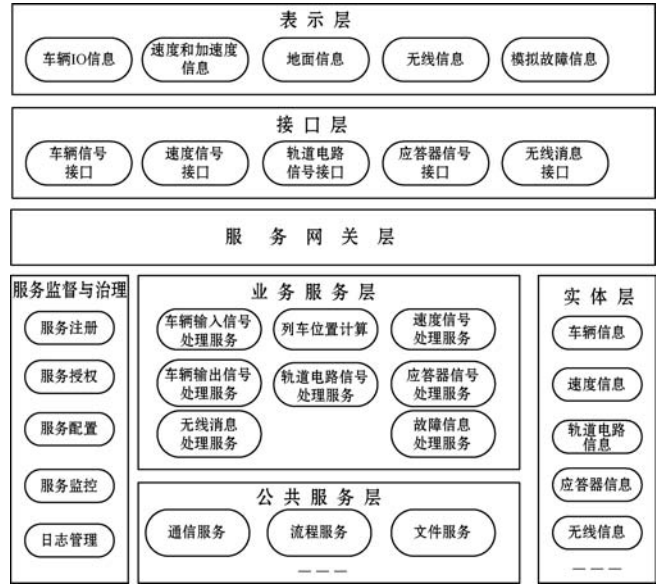


图 2 微服务架构

1) 表示层。使用前后端分离技术,前端实现与用户的交互:提取列控工程数据脚本文件轨道电路信息、应答器信息及无线脚本文件的无线消息数据,通过界面模拟司机操控台,进行速度和加速度的控制,并通过界面模拟地面和车辆等故障信息,测试 ATP 车载设备的异常信息处理。

2) 接口层。对外提供数据和服务。将轨道电路信息、应答器信息、速传信息和无线消息等信息进行数据提取,并进行数据适配和格式转换。针对第三方车载设备测试,根据业务规则定制好接口,进行接口信息接入、数据适配和格式的转换。

3) 服务网关层。服务网关提供服务统一代理入口与调用分发,是连接客户端和后台的中介。网关是系统中的门户,它通过执行服务的路由和消费契约的方式,确保微服务的垂直扩展。同时,网关也具备负责鉴权、认证、安全和跳转等作用。

4) 服务监督与治理层。包括服务注册、发现、负载均衡、限流、熔断、配置、监控和日志分析。

(1) 服务注册。去中心化的服务注册,注册节点共同组成一个服务注册系统,每个服务节点均可以提供服务注册和发现功能,系统无须维护一个高可用的注册中心,有效避免了只有一个注册中心因宕机而导致的系统瘫痪风险^[13]。当某微服务注册的注册中心发生故障时,构建高可用的多注册中心能够使每个注册中心同时具备服务端和客户端的功能,实现交叉注册,既可以向其他注册中心注册自己,又可以向其他注册中心提供注册功能,注册中心之间互备,防止单个注册中心故障导致注册该中心的微服务不可用。通过与面向服务架构中使用具备中心化特点的企业服务总线比较,微服务架构系统具备高可靠性和高可用性。

(2) 服务发现。在微服务架构中,服务发现是最为核心和基础的组件,它为各个服务的相互识别提供了前提条件,并提供服务实例的具体网络位置以便其他服务调用。

(3) 负载均衡。通过标准的接口,由具体组件负责负载均衡。它将对同一微服务的多个请求分配到该微服务的特定实例进行处理,从而保证同一微服务的每个实例所处理的请求在数目上大致保持一致,提高了服务的执行效率。

(4) 配置中心。在微服务框架中,配置中心管理所有微服务的配置。它做到应用配置与应用部署完全分离。它实现在分布式环境下对微服务运行时所需参数的集中管理、动态调整和自动更新。

(5) 服务监控。主要目的是在系统运行时对基础设施和微服务等性能进行观察,它提供微服务状态和调用链路监控,方便用户根据微服务状态和执行情况进行系统性能参数的调优和日常运维。

(6) 日志管理。日志用于在微服务等发生故障后快速排除错误。日志管理包括:调试、警告、错误和异常,提高了系统的可维护性。

(7) 容错保护。当某个服务单元发生故障时,通常需要经过长时间的等待,然后返回连接超时等错误信息。故障监控的断路器能够监测服务的状态,当服务发生故障时,会立即向调用方法返回一个异常响应,而无须长时间等待,解决了线程因调用故障使服务长时间不能够释放资源的问题,提高了系统的性能和可用性。当服务器压力剧增影响整体服务性能的时候,实施服务降级策略,保证核心服务的稳定,以此增强系统的稳定性和可用性。

5) 业务服务层。涵盖本系统的主要业务,包括车辆输入信号处理服务、车辆输出信号处理服务、列车位置计算服务、速度信号生成服务、轨道电路信号生成服务、应答器信号生成服务、无线消息处理服务、故障信息处理服务等。业务服务层可根据系统需求的新变化添加新的服务,方便了系统功能扩充,提高了系统可扩展性。针对复杂的业务,可以进行颗粒度细化,拆分成若干微服务,分别进行开发和维护,提高了系统可维护性。

6) 公共服务层。系统中多处用到的功能统一封装成可复用的公共服务,作为公共服务层,它为业务服务层提供更便利的支撑。它包括通信服务、流程服务、文件服务等。其中流程服务能够将已有的多个功能独立的服务根据业务需要通过服务组合的方式组合成更为复杂和完善的整体应用,提高了系统可扩展性。服务组合方式有编排和协同两种方式:服务编排是采用

中央协调器方式来协调对多个服务的调用;服务协同是指所有服务以对等方式相互协作而无须中央协调器参与。本系统采用服务协同的方式实现微服务的去中心化和独立性,提高了系统的可靠性。

7) 实体层。将系统所需要的关键数据封装为实体信息,供业务服务层和公共服务层调用。它包括车辆信息、轨道电路信息、应答器信息、无线信息等。

3.3 关键技术

1) 领域驱动的中台设计。领域驱动设计是一系列软件设计实践、技术和原则的集合,它是服务划分的一种方法,能够将领域模型解耦成包含业务流程且相互区别的限界上下文并表达它们之间的关系。

中台分为业务中台和数据中台,业务中台是将后台的系统资源进行有效组织,转化为前端可调用的业务服务,使系统中的业务能力沉淀到业务中台,提升业务的协作效率。业务中台依据“高内聚、低耦合”原则,将复杂业务系统进行细粒度拆分,降低业务系统复杂度,有利于软件的开发和维护。数据中台是指通过数据技术对系统中的数据进行采集和处理,把数据封装成服务,并以接口方式输出^[14]。中台利用平台化手段,将公共业务和数据沉淀部分进行复用,它为前台减负,提高前台快速响应能力;同时整合系统的通用化能力,快速支持系统的业务扩展,提高了系统的可扩展性。

2) 网关。网关是连接客户端和后台的中介,具备请求接入、业务聚合、中介策略和统一管理的作用。它关注于系统的业务可维护性、系统吞吐能力、客户端请求接入能力和实现安全、验证、过滤、限流、监控等各种策略。通过采用 API 网关,实现统一接入、协议适配转换、流量管控、安全防护、服务的路由和执行。

3) 通信。通信方式包括同步通信和异步通信,同步通信主要采用 http 方式,异步通信主要是基于消息的通信,它通过消息中间件的发布订阅模式实现消息的异步传输。消息中间件是事件驱动机制的重要部分,负责将事件源产生的事件发布给事件订阅者,实现事件驱动业务在执行过程的协同一致。消息中间件实现微服务低耦合、持久化和分布式异步通信,避免因消息接收端宕机或离线等故障而导致信息丢失,并在此基础上,通过 Web Service 接口实现与第三方系统的集成^[15]。

4 实验验证与分析

4.1 实验设置和验证

针对测试的 ATP 车载设备需求大量变更,采用微

服务架构建模的仿真测试系统能够快速响应需求的新变化和功能的扩展,大幅提高软件开发的效率。此外,搭建实验平台对系统进行验证。实验平台如图 3 所示。

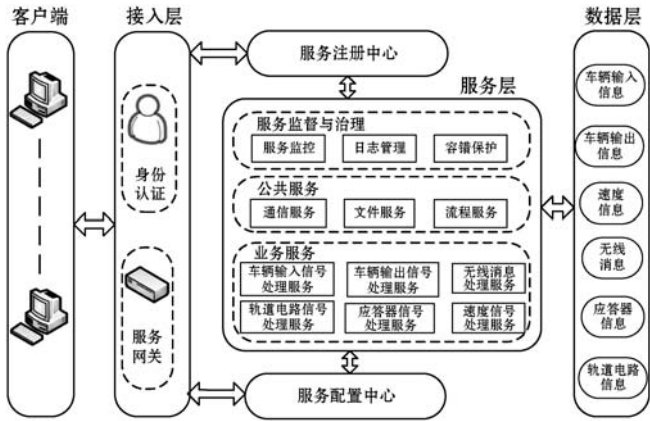


图 3 实验平台构成

实验平台构成主要包括客户端、接入层、服务注册中心、服务配置中心、服务层和数据层。客户端是模拟列车在地面线路上运行的仿真程序。接入层主要实现身份鉴权和分发请求。服务注册中心提供服务注册和发现功能。服务配置中心提供微服务动态配置和微服务信息更新的同步。服务层包括各种服务:服务监督与治理、公共服务和业务服务。数据层包括重要的业务数据信息。

实验平台采用分布式部署,客户端程序运行在多个终端 PC 上,服务器 A 包括接入层、服务注册中心和服务配置中心。服务器 B 包括服务监督与治理和公共服务。服务器 C 包括业务服务和数据层。微服务协同工作原理如下:首先,客户端向接入层发送应用请求,经过接入层身份认证后,进入服务网关。服务网关读取数据请求,并从服务注册中心获取相关的服务信息,然后向服务层分发请求。服务层中的微服务根据请求进行相应的业务逻辑处理,并访问数据层中的相关数据信息,最后将请求处理的结果返回给客户端。以测试列车 C2/C3 等级转换场景为例:系统用户经过身份认证登录后,执行客户端程序,经过服务网关读取数据请求,并通过服务注册中心获取服务层中的文件服务信息,向服务层文件服务分发请求,加载线路脚本文件和无线脚本文件,从数据层中获取相应的线路轨道电路信息、应答器信息、无线消息和速度信息。通过轨道电路信号服务、应答器信号服务和速度信号服务处理后,通信服务将轨道电路信息、应答器信息和速度信息发送给列控系统 ATP 车载设备。ATP 车载设备 C2 等级发车,接收到应答器信息的通信会话管理包后呼叫 RBC,在成功与 RBC 建立无线通信会话后,RBC 执行无线脚本文件与 ATP 车载设备进行信息的实时

交互。ATP 车载设备 C3 后备接收 RBC 发送的行车许可,经过等级转换预告应答器和等级转换执行应答器后完成 C2/C3 等级转换,从而实现 C3 主控控车。在此期间,客户端程序界面中能够实时回显轨道电路信息、应答器信息、无线消息和列车速度等信息。

4.2 实验分析

同时,对系统的性能单独进行压力测试。测试工具使用 Loadrunner,并采用 Http/Winsocket 协议。主要思想是通过虚拟用户模拟实际用户对系统施加压力。测试过程中考虑以下性能测试指标:

- 1) 数据端到端的总时间:从客户端数据请求发起,经过服务网关、服务注册中心获取相应微服务,调用并执行微服务业务逻辑处理,返回微服务处理的结果给客户端程序的时间。
- 2) 读取数据流及获取相应微服务接口的时间:服务网关读取客户端数据请求,并从服务注册中心获取相关微服务的时间。
- 3) 执行及获取应答时间:执行微服务业务逻辑处理,并将处理结果返回客户端的时间。

通过对系统进行大规模的压力测试分析,以某典型业务场景中高频发生的多个复杂业务混合操作为例,采用 20、40 和 100 个用户分别进行并发操作,每个并发操作选取 1 个用户登录系统,循环测试 40 min,实现测试目标。测试结果如表 1 所示。

表 1 并发用户数与事务执行情况

并发用户数	20	40	100
通过的虚拟用户数	20	40	100
总吞吐/Byte	3 369 214 563	5 003 267 458	7 056 708 073
平均吞吐量/(Byte · s ⁻¹)	1 293 365	1 986 212	2 892 093
事务平均响应时间/s	0.385	1.026	1.742
事务最大响应时间/s	0.524	1.568	3.136
总点击次数	180 165	267 678	376 722
每秒 HTTP 响应数	69.161	106.264	154.394

通过表 1 可以分析得出:随着测试中并发用户数目的增加,事务平均响应时间和最大响应时间都在相应增加,但是即使在高并发达到 100 户的情况下,事务平均响应时间也在 1 s 多以内,而系统实际使用的用户数主要是在 20 户以下,事务平均响应时间和最大响应时间远远小于 1 s。总体上分析,在大吞吐量高并发

情况下,系统运行平稳可靠,执行效率高,具备高可用性、高可扩展性和高可靠性。

5 结 语

高铁列控仿真测试系统涉及大规模软硬件开发,并且接口众多结构复杂。本文结合微服务架构的优点和设计原则,设计出一种行之有效的微服务架构模型。通过实验验证与分析,基于微服务架构实现的高铁列控仿真测试系统促进了信息系统的开发与协同效率,使业务系统能快速地响应新需求的变化,方便了系统扩展和运维,实现了系统的高可用性、可靠性、可扩展性和可维护性。此外,采用本文提供的微服务架构模型,在涉及到复杂系统软件开发和大型遗留系统升级改造方面提供了一种解决方案。

参 考 文 献

- [1] 赵显琼,李开成,唐涛,等. 基于 TTCN-3 的 CTCS-3 级列车运行控制系统自动测试方法[J]. 铁道通信信号,2010,46(7):15-20.
 - [2] Jo H J, Yoon Y K, Hwang J G. Analysis of the formal specification application for train control systems[J]. Journal of Electrical Engineering & Technology,2009,4(1):87-92.
 - [3] 张勇,王超琦. CTCS-3 级列控系统车载设备测试序列优化生成方法[J]. 中国铁道科学,2011,32(3):100-106.
 - [4] 谢雨飞,唐涛,徐田华,等. CTCS-3 级列控系统规范的建模与形式化验证方法研究[J]. 铁道学报,2011,33(7):67-72.
 - [5] Dong H, Dong W, Ji Y. A HLA-based hierarchical architecture for the CTCS hardware-in-the-loop simulation system [C]//2nd IEEE International Conference on Computer Science and Information Technology,2009:86-91.
 - [6] 张维维. CTCS-3 级列控系统仿真测试平台-车载设备仿真子系统的研究[D]. 北京:北京交通大学,2006.
 - [7] 卢楠. 基于高速列控仿真测试平台的专家系统[D]. 北京:中国铁道科学研究院,2015.
 - [8] 庞家杰. CTCS-3 级列控系统多车仿真测试平台的设计与实现[D]. 北京:北京邮电大学,2017.
 - [9] 陈建球. 基于 SOA 的列控仿真系统设计及可行性研究[D]. 北京:北京交通大学,2017.
 - [10] 于曼,黄凯,张翔. 基于微服务架构的 ETC 系统设计[J]. 计算机科学,2020,47(S1):643-647.
 - [11] Balalaie A, Heydarnoori A, Jamshidi P. Microservices architecture enables DevOps: Migration to a cloud-native architecture[J]. IEEE Software,2016,33(3):42-52.
 - [12] 徐天桥. 微服务对传统应用的改造及设计[J]. 电子制作,2020(8):38-39,63.
 - [13] 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,2016,14(9):12-17.
 - [14] 屈艳玲. 面向微服务架构的数字化图书馆构建[J]. 内蒙古科技与经济,2020(3):150-151,155.
 - [15] 黄启启,项前,程茂上,等. 基于微服务的仓储管理与控制系统[J]. 东华大学学报(自然科学版),2020,46(1):83-90.
-
- (上接第 245 页)
- [4] Mumtaz J, Guan Z, Yue L, et al. Multi-level planning and scheduling for parallel PCB assembly lines using hybrid spider monkey optimization approach[J]. IEEE Access,2019,7:18685-18700.
 - [5] Benoit A, Le Fèvre V, Raghavan P, et al. Resilient scheduling heuristics for rigid parallel jobs[J]. International Journal of Networking and Computing,2021,11(1):2-26.
 - [6] Cheng C Y, Pourhejazy P, Ying K C, et al. Learning-based metaheuristic for scheduling unrelated parallel machines with uncertain setup times[J]. IEEE Access,2020,8:74065-74082.
 - [7] Nazari M, Oroojlooy A, Takáč M, et al. Reinforcement learning for solving the vehicle routing problem[C]//32nd International Conference on Neural Information Processing Systems,2018:9861-9871.
 - [8] Bello I, Pham H, Le Q V, et al. Neural combinatorial optimization with reinforcement learning [EB]. arXiv:1611.09940,2016.
 - [9] Shi D, Fan W, Xiao Y, et al. Intelligent scheduling of discrete automated production line via deep reinforcement learning[J]. International Journal of Production Research,2020,58(11):3362-3380.
 - [10] 沙宗轩,薛菲,朱杰. 基于并行强化学习的云机器人任务调度策略[J]. 计算机应用,2019,39(2):501-508.
 - [11] Chen X, Tian Y. Learning to perform local rewriting for combinatorial optimization[EB]. arXiv:1810.00337,2018.
 - [12] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine Learning,1992,8(3/4):229-256.
 - [13] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[EB]. arXiv:1707.06347,2017.
 - [14] Boney R, Di Palo N, Berglund M, et al. Regularizing trajectory optimization with denoising autoencoders[J]. Advances in Neural Information Processing Systems,2019,32:2859-2869.
 - [15] Raphan M, Simoncelli E P. Least squares estimation without priors or supervision[J]. Neural Computation,2011,23(2):374-420.