

面向纺织制造的数据一致性与服务解耦微服务 MES 系统设计与实现

王金永¹, 党向盈¹, 石 炜¹, 郁雯杰², 孙长征¹

(1. 徐州工程学院 信息工程学院, 江苏 徐州 221018; 2. 江苏恒辉安防股份有限公司, 江苏 南通 226499)

摘 要: 纺织行业面临市场需求增长与“智改数转”转型的双重挑战, 传统制造执行系统在定制化生产、精细化管理及实时数据监控方面逐渐暴露出局限性。针对纺织业工艺流程复杂、数据异构、实时控制与协同管理需求高的行业特征, 基于SpringCloud微服务架构设计并实现了一套纺织业制造执行系统(MES)。系统采用Seata分布式事务管理方案, 确保跨服务数据一致性, 同时结合MyBatis-Plus构造器函数优化后端数据处理性能, 从而满足多源数据接入、异构环境适配及高并发响应需求。系统基于纺织制造流程进行功能拆分, 构建了订单管理、设备控制、生产执行和质量监控等多个独立微服务模块, 并完成了数据库结构设计与模块间调用机制。通过接口测试、业务流程测试与系统压力测试, 验证了系统在数据一致性、模块隔离性及系统稳定性方面的可行性与有效性。该系统已成功部署于江苏恒辉安防股份有限公司, 实现了产线自动化管控与流程数字化协同。研究结果表明, 基于微服务架构的纺织业MES系统在系统可扩展性、实时监控、故障隔离及资源协同管理方面具有显著优势, 具备较强的实际应用价值与行业推广潜力。

关键词: 纺织制造执行系统; 微服务架构; 系统集成; 数据一致性; 系统可扩展性

DOI: 10.11907/rjdk.241426

开放科学(资源服务)标识码(OSID):

中图分类号: TP319

文献标识码: A

文章编号: 1672-7800(2025)009-0112-10



Design and Implementation of Microservices-Based MES System for Textile Manufacturing with Data Consistency and Service Decoupling

WANG Jinyong¹, DANG Xiangying¹, SHI Wei¹, YU Wenjie², SUN Changzheng¹

(1. School of Information Engineering, Xuzhou University of Technology, Xuzhou 221018, China;

2. Jiangsu Hanvo Safety Product Co., Ltd., Nantong 226499, China)

Abstract: The textile industry is facing dual challenges of surging market demand and the imperative for intelligent and digital transformation. Traditional Manufacturing Execution Systems (MES) have shown limitations in supporting customized production, refined process control, and real-time data monitoring. To address key industry-specific issues such as complex workflows, heterogeneous data sources, and high demands for real-time coordination and collaborative management in textile manufacturing, this paper designs and implements a microservices-based MES using the SpringCloud framework. The system adopts Seata for distributed transaction management to ensure cross-service data consistency, and integrates MyBatis-Plus query builders to optimize backend data processing, thereby meeting the requirements of multi-source data access, heterogeneous environment compatibility, and high concurrency responsiveness. Based on the analysis of textile manufacturing workflows, the system is modularized into independent microservices, including order management, equipment control, production exe-

收稿日期: 2024-04-30

扫描二维码阅读全文:



基金项目: 教育部“春晖计划”国际合作科研项目(HZKY20220164); 教育部产学研合作协同育人项目(220607026242907); 徐州市科技计划项目(KC23012)

作者简介: 王金永(1983-), 男, 博士, CCF会员, 徐州工程学院信息工程学院讲师, 研究方向为数智协同系统设计、安全强化学习; 党向盈(1978-), 女, 博士, 徐州工程学院信息工程学院教授、硕士生导师, 研究方向为基于搜索的软件工程、进化算法应用; 石炜(2003-), 男, 徐州工程学院信息工程学院学生, 研究方向为数据驱动软件设计; 郁雯杰(2000-), 男, 江苏恒辉安防股份有限公司工程师, 研究方向为系统设计与软件测试; 孙长征(2004-), 男, 徐州工程学院信息工程学院学生, 研究方向为系统设计与软件测试。本文通讯作者: 党向盈。

cution, and quality monitoring. The system architecture also includes comprehensive database schema design and service interaction mechanisms. Through interface testing, process flow validation, and system stress testing, the feasibility and effectiveness of the system have been verified in terms of data consistency, module-level isolation, and operational stability. The system has been successfully deployed at Jiangsu Hanvo Safety Product Co., Ltd., achieving automated production control and digital process coordination. Results demonstrate that a microservices-based MES for the textile industry offers significant advantages in scalability, real-time monitoring, fault isolation, and resource coordination, and holds strong potential for practical application and industry-wide adoption.

Key Words: textile manufacturing execution system; microservice architecture; system integration; data consistency; system scalability

0 引言

随着信息技术的持续演进,智能化改造与数字化转型已成为制造业提升核心竞争力的重要手段。纺织业作为我国传统支柱产业之一,当前正面临“智改数转”的紧迫需求。然而,传统制造执行系统(Manufacturing Execution System, MES)在支持定制化生产、流程协同管理与实时数据监控等方面逐渐暴露出局限,难以满足纺织行业多工序、长链条、设备异构和高并发的复杂生产场景。传统纺织MES系统多采用单体架构模式,业务流程常常涉及多工序、多设备与多部门协同,存在以下典型问题:①系统采用单体架构,部署和更新需整体打包,一旦某一模块功能调整,将影响全局;②数据采集接口高度依赖于特定硬件和协议,扩展性与复用性差;③部分核心模块(如称重系统、质量检测)存在并发瓶颈,难以支撑车间高频业务;④不同车间、分厂区之间存在管理边界,无法通过统一系统灵活部署。例如,在江苏恒辉安防股份有限公司的原系统中,面对织机扩容和功能优化时需整体停机打包更新,严重影响产线连续性与生产节奏。同时,系统数据采集接口高度依赖特定硬件与通信协议,导致设备接入兼容性差,维护成本高、故障定位难。

针对以上问题,本文提出采用SpringCloud构建微服务架构的MES系统。具体优势如下:①每个核心业务模块(如设备管理、订单管理、工艺执行和质量控制等)可按业务粒度拆分为独立微服务单元;②系统部署灵活支持模块级更新,具备异地、异网环境部署能力;③支持多协议适配器设计(如COM口采集、REST等),极大提升对设备接入的兼容性;④结合服务注册与发现、统一网关、远程调用与分布式事务协调机制,实现高内聚、低耦合的服务通信与自治运行;⑤微服务架构天然支持高并发访问和异步任务处理,适用于称重、扫码和质检等业务场景;⑥通过API网关统一接口输出,提升与供应链系统协同能力。江苏恒辉安防股份有限公司原系统在面对织机数量增加时,需要进行整体升级部署,这严重影响了生产节奏^[1]。在出账与称重系统改造过程中,曾尝试基于原有单体架构进行拓展,但每次功能优化都需要停机维护并进行整体打包部署,这导致了生产中断。通过采用微服务重构后,实现了称重服务与FQC检测服务的独立部署,系统支持热更新和异步处理,不仅显著降低了更新维护成本,还提高了在高并发场

景下的响应能力。

本文不仅聚焦于工程实践层面的系统构建,更在制造业务流程分析、微服务模块划分、链路一致性保障与系统扩展能力等方面提出了适用于纺织行业的通用设计思路。该工作的开展为传统纺织企业的信息化转型提供了技术参考,并为制造业MES系统的微服务架构化演进提供了现实样本。

本文系统设计融合了以下关键技术:①采用Seata框架,实现了跨服务的分布式事务一致性保障;②利用MyBatis-Plus构造器,提升了数据库访问效率;③采用MySQL与Redis的混合数据架构,实现了高频实时数据的采集与历史数据的沉淀管理。该系统已成功部署于江苏恒辉安防股份有限公司,初步测试结果表明,与传统架构相比,其在系统吞吐量、模块隔离性、数据一致性与稳定性方面均有显著提升,展现了良好的工程应用前景与学术价值。

1 相关工作

本文将从以下2个方面进行国内外研究现状分析:①企业制造执行系统的设计与实现;②微服务架构的设计与应用。在制造业的工业实践中,世界各国都在积极开展新一代信息技术与制造业的深度融合,形成整个国家政产学研高度共识的发展蓝图。例如,美国的先进制造业伙伴计划、德国工业4.0战略、“英国制造2050”与“中国制造2025”等。工业互联网的主要特征体现为数字化、智能化和网络化,可以实现智能制造纵向集成、端到端集成和横向集成,并驱动制造执行系统向生产要素管理和决策一体化系统发展^[2]。人工智能、云计算、大数据、物联网等技术和工业融合发展形成了工业智能,工业智能的本质是人工智能通用技术与工业知识、工业场景结合,实现具备MAPE-K自感知、自分析、自决策、自执行和自反馈等能力,系统可以动态适应变化的工业环境,完成定制化工业系统的开发^[3]。工业软件的深入应用是加速制造执行系统智能化改造和数字化转型的关键,智能制造对MES提出了模块化、组件化的需求,需要对业务功能进行微服务的划分,设计基于微服务架构的MES框架^[4]。在传统的单体架构中,MES功能之间耦合程度高,难以快速地根据企业需求变化进行重构。为提升MES的可重构性,并解决其网络化模块高耦合、通用性不足及可扩展性差等问题,本研究构建了基于微服务架构的多层系统架构,以消除系统流

程与业务逻辑之间的耦合度^[5-8]。

微服务架构在系统设计中有着广泛的学术研究和应用实践。采用微服务架构设计,可以构建一个可扩展的数据采集和数据处理的数据平台,提供智能管理、定位监控、任务调度与危险报警的头盔 PasS 平台,同时减少安全事故的发生^[9]。基于 SpringCloud 微服务架构设计思想,系统重构后各功能模块之间的耦合度较低,能够实现独立部署和运行^[10]。同时,通过引入微服务和 Docker 技术对系统进行重新架构,系统能够实现高并发和动态伸缩性^[11-12]。系统采用图聚类算法对其图结构进行聚类处理,以实现将服务拆分为微服务系统。同时,采用双 Redis 数据库设计以满足系统的实时性和并发性需求^[13]。设计一种基于预分配额度进行主动推送配额更新的限流算法,使基于微服务的系统能够更好地适应线上线下精准限流的需求^[14]。通过设计流程层和服务层系统双层依赖关系模型和微服务系统中流程演化的实现逻辑,解决了微服务流程演化以适应业务变化的需求^[15]。通过引入因果贝叶斯网络和干预数据,设计基于因果干预的微服务系统故障根因分析方法,可以更好地分析微服务系统中的故障根因^[16]。综上,采用软件监控、故障诊断、故障预测、故障复现和测试可以进行微服务故障检测^[17]。

2 纺织业制造执行系统

2.1 研究内容与方法

为满足纺织行业“工艺流程复杂、数据多源异构、实时管控要求高、柔性生产与协同调度并存”等特性,文章基于 SpringCloud 构建纺织制造执行系统(MES),围绕系统架构设计、数据管控能力、系统协同可视化与安全性开展研究。研究内容和方法主要包括以下 3 个方面:

2.1.1 基于业务边界的微服务架构拆分与集成机制

在纺织制造场景中,不同车间/工艺段之间业务耦合复杂、流程灵活多变。如何依据业务边界合理划分微服务模块,并通过服务注册、统一入口、远程调用与分布式事务协调机制,构建一个既解耦又协同的系统架构,是本研究的关键问题之一。本研究主要关注微服务架构设计与系统集成,具体包括:①基于 SpringCloud 框架,将传统单体 MES 系统拆分为多个可独立部署的微服务(如订单管理、设备管理、质量控制、生产执行等);②借助 Eureka(服务注册)、Feign(远程调用)、Gateway(统一入口)和 Seata(分布式事务),以实现系统的高可用、解耦与事务一致性控制;③有效解决传统系统模块间耦合度高、功能扩展困难及部署更新复杂等问题。

2.1.2 多源异构数据环境下的高效数据采集与一致性控制策略

基于电子秤、扫码器、COM 接口设备和 K3 等系统的多源异构数据,如何实现低延迟、高频率和可追溯的数据采

集与状态同步,以及如何在多微服务间保持数据一致性,是数据管理与控制机制中的技术难点。本研究主要关注数据管理与实时控制机制,具体包括:①结合 MySQL+Redis 的混合数据架构,实现高频采集数据的快速响应与低频业务数据的稳定存储;②引入消息队列与事件驱动机制,实现设备状态与生产数据的实时同步传输至各业务微服务,确保生产过程的可控性与数据的全程可追溯性;③使用 Mybatis-plus 提供的数据库访问能力与查询构造器,支撑高效的数据处理与统计分析。

2.1.3 面向供应链协同的可视化平台构建与安全治理机制

在制造执行系统中集成上下游数据协同能力的同时,如何基于 API 网关实现平台可视化交互,并保障系统在跨企业、多角色访问情境下的数据安全与权限控制,是本研究重点关注的协同与安全研究问题。本文主要关注供应链协同、可视化与系统安全保障,具体包括:①构建供应链可视化平台,基于 API 网关实现纺织企业与上下游供应链系统的数据共享与协同调度;②通过 Spring Security 实现用户身份认证与权限控制,结合 JWT(JSON Web Token)数字签名令牌机制保障接口访问安全;③对关键业务数据采用权限隔离与访问控制策略,防止外泄与非授权操作。

为了进一步说明本系统采用微服务架构的技术优势,表 1 对比了传统单体架构 MES 系统与本文微服务架构在核心特征上的区别。通过结构对比可以看出,微服务架构在模块解耦、可扩展性、容错能力和自动化实现等方面具有明显优势,尤其适用于纺织行业制造执行系统中“模块多、流程长、接口复杂”的典型场景。

Table 1 Comparison between monolithic architecture and microservices architecture in mes system implementation

表 1 微服务与单体架构的区别

序号	结构特征	单体架构传统 MES 系统	MES 系统基于微服务实现
1	模块部署	所有模块打包为一个整体,统一部署	各模块独立部署,独立运行
2	服务间依赖	紧耦合,改动一处,常需重构整体系统	松耦合,服务独立更新
3	可扩展性	水平扩展困难	可对热点微服务单独扩展,比如 FQC 检测服务
4	数据一致性	基于本地事务,简单处理	使用 Seata 解决分布式事务一致性问题
5	故障影响范围	任一模块异常可能影响整个系统	局部故障隔离,不影响其他服务
6	技术异构支持	技术选型受限	各服务可根据需要选择不同框架、数据库等 多数操作流程通过服务逻辑
7	自动化能力	需依赖手动流程控制	实现自动化,例如投料校验流程

由表 1 可知,纺织业制造执行系统在多个关键技术点(如服务拆分、分布式事务、自动化处理流程)上均体现了微服务架构的优势。例如,通过将 FQC 质量检测、设备状态采集和订单执行等功能模块独立部署,不仅提升了系统

运行效率,还提高了模块级的可维护性和可扩展性,有效支持了企业多车间、多品类的柔性化生产管理需求。

2.2 系统总体功能架构设计

纺织业制造执行系统功能模块分为管理员和用户 2 个部分:①管理员负责系统的整体管理与维护,主要功能模块如图 1 所示;②用户通过管理员分配的权限进行不同的操作,主要功能模块如图 2 所示。

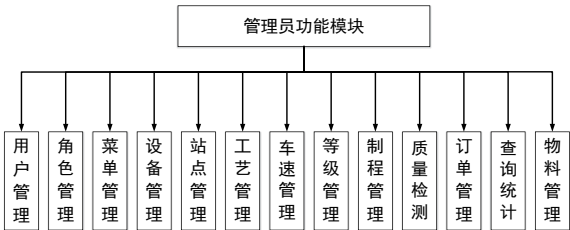


Fig. 1 Administrator function module

图 1 管理员功能模块

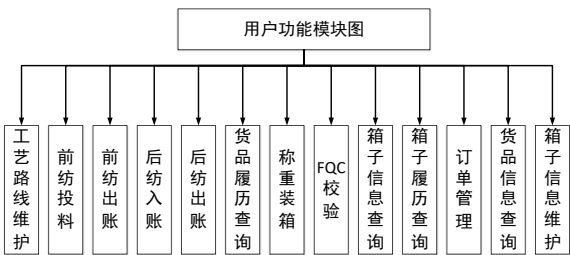


Fig. 2 User function module

图 2 用户功能模块

其中,前纺模块的功能流程如下:用户登录系统后,首先生成生产订单。随后进入制程管理中的前纺投料环节进行投料操作。在投料过程中,用户选择相应的生产订单,并执行以下 4 个步骤:第一步为复核物料是否用错(生产任务单+工艺单中的物料信息进行对比,错误则提示原料错误);第二步为系统记录数据配一釜多少 kg 原料+白油,数据来源于工艺路线设置;第三步为系统时间默认为投料日期时间,此时间作为前纺换桶后的追溯时间,默认为 48 小时;第四步为每配一釜编号(任务单号+年月日+计时器),记录一行流水数据可导出,投料后会进行预溶胀、溶胀,溶胀完成后开始纺丝,投料结束后换桶,换桶时系统自动关联釜编号,记录数据(此处需要手动修正功能,可进行人工关联桶号与釜号,可一桶多釜)。

后纺模块功能分析:在前纺静置完成后,所有纺丝进入集束架,经集束、萃取、干燥、牵伸和卷绕等一系列处理流程,最终制成 PE 丝。在集束过程中,产线班长根据生产任务单所需的筒数进行选择(默认为无限,当输入具体数量时,按数量自动分配筒数),并选择车速以绑定筒信息和后纺地标线的位置号信息。筒信息与卷信息的绑定过程可追溯至釜编号。卷绕过程中,需进行换桶操作,并打印上一个筒的标签信息。

最终质量控制(Final Quality Control,FQC)校验模块功能分析:在后纺出账后,出账的货品需进行 FQC 校验。此模块需进行电子秤称重,系统通过读取电子秤的 COM 口

获取数据信息,基于这些数据自动计算该成丝品的百米克重,并通过百米克重判断其质量等级。

2.3 系统数据库总体设计

该系统主要有 2 个对象:管理员(车间主任)和用户(各条生产线的员工)。由管理员给每个生产线的员工分配账户,不同生产线的用户授予不同的权限,确保用户仅能访问其对应岗位的功能模块。在基础模块中,用户需要对生产线上的数据进行维护,如桶号、制程设备、站点、工艺路线、车速和等级等。在制程模块中,首先需要生成订单。根据生成的订单,前往前纺模块选择合适的设备进行投料。投料完成后,将订单号与桶号进行绑定,并开始出账操作。出账完成后,需登记对应的静置时间,至此前纺流程结束,流程进入后纺阶段。其次,在后纺入账时,系统会根据桶号绑定后纺所需的设备和地标号,并校验是否达到或超过静置时间要求;若已满足要求,即可进行入账操作。然后,入账成功后,需要执行上筒操作。上筒后的丝将自动进入集束架,依次完成集束、萃取、干燥等一系列工艺流程,并在完成后自动进行出账处理。至此,后纺流程结束,产品进入全检打包区。最终,全检环节包括 FQC 检验和 FQC 复检,主要对成丝产品的质量进行判定。检验合格的产品将进入打包和装箱流程,完成整个制程模块。

2.4 微服务架构层次划分

系统按照数据层、平台层、服务层和应用层进行拆分,如图 3 所示。该系统通过重点呈现系统架构图及各模块的职责说明,清晰地展示了微服务架构的具体实现方式。

(1)数据层(Data Layer)。数据层是系统的底层支撑,提供持久化数据存储和高速缓存服务。其中,数据库(MySQL)用于存储系统核心业务数据,比如订单信息、设备状态和工艺参数等;缓存数据库(Redis)提高数据访问效率,支持实时数据采集、设备 COM 口数据传输等。系统中每个微服务(如生产订单、设备管理、质量控制等)都拥有独立的数据模型,实现了逻辑隔离。采用 Seata 分布式事务协调器实现跨微服务数据一致性管理,同时支持异构数据接入(包括 K3 系统推送数据及 COM 口设备数据等)。

(2)平台层(Platform Layer)。平台层为微服务运行提供基础设施支持,从而确保微服务的可注册、可发现、可监控和高可用。系统通过 SpringCloud 构建微服务平台,以支持服务弹性扩展和动态发现,具备高可用性。具体包括:①Eureka,服务注册与发现中心,实现服务间动态源路由;②Ribbon,客户端负载均衡机制,提升请求分发效率;③Hystrix,实现服务熔断与降级,提升系统容错能力。

(3)服务层(Service Layer)。服务层是系统的核心业务实现层,基于 Spring Cloud 框架构建。通过 API 网关实现统一接入,并利用 Feign 实现服务间的远程调用。系统采用 Gateway 作为统一 API 网关,为外部提供访问入口,并通过 Seata 处理分布式事务。服务层进一步划分为 3 个子模块:①核心业务微服务,包括订单管理、设备管理、生产执

行、质量控制和仓储物流等,覆盖纺织制造执行的主要环节;②系统管理微服务,包括用户、菜单、角色、工艺和数据查询等,主要负责系统运行过程中的配置与管理功能;③平台支撑微服务,提供制程管理、API网关、安全认证、配置中心和链路追踪等辅助支撑服务,增强系统整体协同能力。

在微服务架构中,这些功能被明确拆分为多个独立的微服务:每个服务都可以独立部署、维护和扩展;服务之间通过轻量级接口进行通信,从而实现松耦合。

(4)应用层(Application Layer)。应用层面面向最终用户提供服务,通过前端 Web 界面和移动端应用构建人机交互入口,分别面向管理人员和一线操作人员。所有用户操作请求均通过 API 网关接入,实现了前后端的分离与业务逻辑的解耦:前端界面可通过 API 网关调用后端微服务,所有功能操作均通过服务接口进行封装调用,例如投料绑定、FQC 称重校验等。此外,系统支持事件驱动机制,可将各端口数据(如电子秤 COM 口数据)实时写入后端的质量管理微服务中,确保数据处理的高效与可靠。

该架构设计体现了良好的可扩展性、模块解耦性与灵活部署能力,为面向纺织行业的定制化、实时化和智能化制造提供了技术支撑,也为系统后续迭代升级和多场景应用打下了良好基础。

3 模块设计与系统实现

3.1 平台后台主要功能实现

(1)制程管理功能。用户点击“制程管理”后,进入系统最核心的流程—前纺与后纺操作。首先,在前纺投料环节,用户需根据生产订单号选择对应的前纺设备进行投料。其次,在前纺出账时,需要将生产订单与对应的桶进行绑定。一条生产订单可绑定多个桶。绑定完成后,系统需登记静置开始时间。至此前纺流程结束,系统进入后纺阶段。用户需选择后纺设备、地标、车速(即上集束架从头到尾的时间)以及待入账的桶。系统将根据生产订单表中静置要求对静置时间进行校验,达到要求后方可入账成功。随后,看板系统会依据设定的车速自动触发定时任务,并开始倒计时。当倒计时剩余 2 min 时,系统将红预警提示。倒计时结束后,系统自动触发上筒(绳丝筒)操作。由于自动化机制的限制,上筒操作中系统会先将前一个筒的货品记录出账,再生成新的货品记录,因此第一个上筒和最后一个下筒的记录必须由人工手动完成出账操作。接着,在后纺出账环节,用户可选择批量或单个方式进行出账。同时,还需判断是否为断丝出账,若发生断丝,系统将自动生成新的货品记录。所有出账信息均通过 LODOP

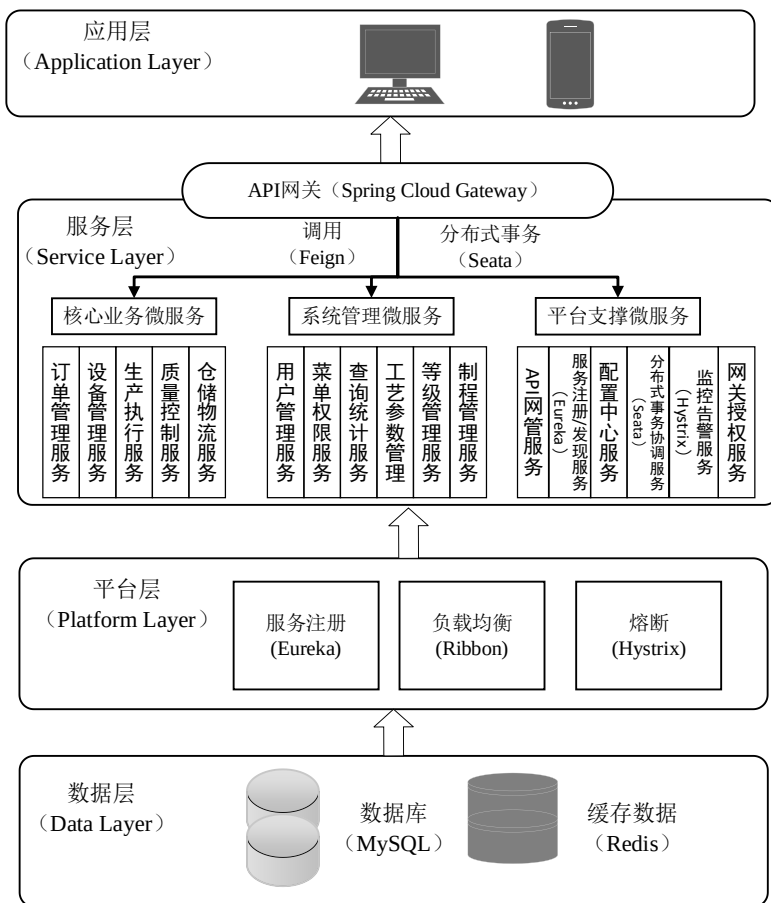


Fig. 3 Microservices architecture of the manufacturing execution system

图3 制造执行系统微服务架构图

打印输出。最终,通过系统流程的控制与人机协同机制,保障前纺与后纺操作的高效与准确。

(2)质量管理功能。在 FQC 检验界面,系统通过定时器自动读取 4 个 COM 口的数据,并将这些数据以 key-value 的形式存储在缓存中。当用户输入货品 ID 并按下回车键后,前端会将货品 ID 和对应的 COM 口数据发送至后端。后端执行信息查询、百米克重计算和自动判断等操作,最终将计算结果返回前端展示。

Table 2 System functional modules and their corresponding microservices
表 2 系统功能模块与微服务映射关系表

序号	功能模块	所属微服务(服务名称)	描述说明
1	用户登录/权限管理	用户认证服务(auth-service)	独立服务,负责登录鉴权、权限控制,支持JWT令牌与角色绑定
2	菜单与角色配置	权限配置服务(rbac-service)	支持不同角色菜单绑定、权限层级配置
3	用户信息维护	用户管理服务(user-service)	用户信息的增删改查与组织架构维护
4	工艺路线与工艺参数	工艺管理服务(process-service)	管理工艺路线配置、工序顺序、工艺版本等信息
5	订单管理	订单服务(order-service)	订单创建、拆分、派发、状态跟踪
6	投料、出账、称重	生产执行服务(exec-service)	涉及生产投料、流程控制、称重逻辑、状态驱动链路
7	FQC 质量检测	质量服务(quality-service)	包括终检、校验、复测、质检判级等,需事务协调与追溯支持
8	物料/产品追溯	查询服务(trace-service)	跨服务聚合数据,支持箱码反查、订单反查、批次流转记录查询
9	箱子与标签管理	包装服务(package-service)	支持箱码生成、箱子绑定、标签打印、装箱记录存储
10	仓储入库管理	仓储服务(warehouse-service)	包括成品入库、库位分配、库存状态更新等
11	数据统计与报表	报表服务(report-service)	提供订单、生产、质量等维度的数据统计与可视化报表输出
12	设备接入与配置	设备服务(device-service)	对接扫码枪、电子秤、串口设备配置、在线状态采集
13	系统日志与审计	日志服务(log-service)	记录系统操作行为、错误信息、事件触发日志
14	消息与通知推送	消息服务(msg-service)	支持任务提醒、异常通知、系统公告等异步推送
15	网关入口与路由	网关服务(gateway-service)	Spring Cloud Gateway,统一服务入口、接口鉴权、限流
16	服务注册与配置	注册与配置中心(eureka + config)	服务注册发现、动态配置下发,支持服务热加载

3.3 系统实现细节

为增强系统实现部分对微服务架构设计的体现,揭示其在系统可靠性、可维护性和可扩展性方面的特点,以下将详细介绍具体实现细节。系统通过标准化 REST 接口、串口协议及消息中间件等方式,实现与上层 ERP 系统、下层执行设备及仓储系统的解耦通信。同时,结合各服务之间的调用路径和消息投递流程,明确展示了微服务架构如何支撑异构数据接入、事件驱动处理和服务间松耦合协同。此外,在系统链路一致性保障方面,介绍了对标识符绑定机制、分布式事务控制(Seata)、状态驱动流程校验和全链路追溯回查等核心机制,强化了对各业务微服务之间在执行链条上的逻辑连贯性与数据一致性控制能力。MES 系统基于微服务架构实现与上下游系统(如 K3 系统、仓储系统、设备数据采集模块等)的对接,通过标准化接口方式实现解耦通信,具体包括:①与上层业务系统接口(K3 ERP 系统),通过标准化 RESTful API 接口实现订单数据、物料库存、库存反馈和发货记录等的双向交互;②与下层执行设备接口,通过串口、PLC 对接程序采集电子秤、扫码枪和 RFID 设备状态等生产数据;③与仓储系统接口,实现出入库信息联动和库位状态更新,保障库存实时准确。

通过统一的数据标准和消息队列(MQ)中间件对接异构系统,确保接口通信的可靠性,并在微服务之间形成缓冲层。避免直接调用核心服务,改为通过 MQ 进行缓冲。例如,设备服务可将采集到的数据发送到 MQ,由 MES 进行

3.2 功能模块的微服务映射表

为体现系统在微服务架构下的服务拆分逻辑与功能分层设计,本文根据业务职责、模块耦合度和服务部署独立性原则,对纺织制造执行系统的各功能模块进行了系统性微服务划分。每个微服务围绕一个核心业务功能构建,具备独立部署、独立运行和独立扩展的能力,并通过 API 网关、服务注册中心与其他服务进行协同通信。表 2 展示了系统中各功能模块与其所归属微服务的映射关系。

异步处理;同时,各微服务的事件可统一投递到 MQ,供订阅者监听和处理。通过这种方式,实现微服务之间的解耦,使其彼此无需强依赖,从而在不同系统和微服务之间传递消息,达到系统解耦、提高性能并增强可扩展性的目的。

为保证投料、执行、称重、质检和入库等环节数据一致性,系统设计了生产链路一致性保障机制,具体包含:①统一业务标识符绑定机制,系统通过订单编号、物料编号、箱码等关键标识,将各生产环节流程绑定,实现从计划任务下发到现场作业执行全过程的数据贯通,确保上下游服务间拥有一致的业务上下文;②分布式事务协调机制,在微服务架构下,系统采用 Seata 实现分布式事务管理,协调如订单状态变更、库存更新与扫码记录入库等多个微服务中的数据一致性,避免跨服务调用导致的数据偏差或写入失败;③状态驱动的链路控制机制,通过在微服务中引入“状态机制”,对每个业务流程节点设置严格的状态变迁规则与依赖校验(如 FQC 检测前必须完成出账,称重操作需验证箱码生成状态等),防止跳步、错步、数据遗漏等问题;④全链路追溯与回查机制,系统支持箱码或产品批次的反向追溯,基于唯一标识可查询其对应的订单来源、投料信息、工艺路线、加工设备及质检结果等,实现全过程数据回溯与责任定位;⑤多层级一致性保障策略,通过“先验证(输入校验)、过程监控(状态管控)、结果回查(追溯分析)”三位一体的保障策略,进一步提升系统运行过程中的

流程闭环性与数据准确性。

4 系统测试

4.1 测试环境与工具说明

为确保系统部署稳定性与测试结果准确性,本文系统测试基于如下环境进行:

(1)系统运行环境。系统部署于一台具备多核 CPU、64 GB 内存与 SSD+RAID 存储架构的中型服务器,运行环境基于 CentOS 7.9,采用 MySQL 与 Redis 构建混合数据架构,微服务容器通过 Docker 封装、Kubernetes 自动编排实现动态扩展与高可用部署,充分满足中小型企业数字化生产系统的性能与稳定性需求。具体配置如下:服务器采用 CPU 为 Intel Xeon Silver 4210×2,内存为 64 GB DDR4 ECC,硬盘为 SSD 1 TB(用于应用)+RAID 存储 2 TB(用于数据库);操作系统采用开源、稳定的企业级 Linux 系统 CentOS 7.9;数据库环境为 MySQL 8.0 +Redis 6.2,MySQL 为主数据库,Redis 为缓存数据库;部署方式为 Docker 容器化部署,服务编排通过 Kubernetes 实现自动扩展与负载均衡;网络架构为企业级千兆局域网,公网访问采用 VPN 加密接入。

(2)软件开发技术栈。后端框架为 SpringBoot+ Spring-Cloud (Eureka、Gateway、Feign、Hystrix);数据库访问层为 MyBatis-Plus,封装通用 CRUD 操作;分布式事务管理为 Seata;前端技术为 Vue+ ElementUI;消息队列为 RabbitMQ (用于异步任务与日志通知)。

(3)压力测试环境与结果。测试工具为 Postman 或 ApiPost7、Apache JMeter 5.4.3。测试场景为模拟 1 000 个并发用户同时进行投料、订单提交和称重出账等核心操作。关键指标结果为平均响应时间 325 ms,最大响应时间 1.2 s,TPS(每秒事务处理数)约 520 次/s,错误率在 0.1% 以下。

测试结果表明:系统在微服务架构支持下具备良好的并发处理能力;关键业务服务部署了副本节点,在测试过程中无明显性能瓶颈。

4.2 接口与功能测试

接口测试、流程测试和压力测试是基于 SpringCloud 的纺织业制造执行系统中常用的测试方法。项目在开发完成后,经过测试可以确保软件获取的数据准确,系统性能稳定。在实际应用中可能会遇到一些未在开发阶段充分考虑到的情况,因此,开发完成后的测试对于保障数据的准确性、系统的稳定性能至关重要。通过测试可以提升系统质量,发现潜在的问题和风险,同时也为系统的改进和优化提供依据和方向。

该系统的测试步骤分为三步:第一步系统开发时对开发的接口进行接口测试;第二步系统开发完成后到车间产线的流程测试;第三步流程测试完成后对整个产线进行压力测试。在接口功能测试阶段,采用 Postman 和 ApiPost

(版本 7)等工具对各业务接口进行功能性测试,确保所有服务调用逻辑正确、响应结构符合预期。在接口功能验证通过的基础上,进一步使用 Apache JMeter 5.4.3 工具进行系统压力测试。模拟高并发访问场景,采集吞吐量(TPS)、平均响应时间和并发处理能力等核心性能指标,测试验证系统在高并发环境下是否依然能够保持良好响应能力与模块级稳定性。

(1)接口测试。接口测试是验证系统接口是否按照预期工作的过程。它主要关注接口的输入和输出,以及接口在各种情况下的行为和返回结果。接口测试可以通过模拟用户请求、发送合法和非法的输入数据,从而验证系统对请求的响应是否符合预期。这种测试方法可以帮助发现接口设计或实现中的错误、异常处理能力和数据一致性问题。接口开发完成后利用 Postman 或 ApiPost7 等测试工具进行接口测试,确保每个调用的接口能正常调用。接口测试用例如表 3 所示。

Table 3 Interface test cases

表 3 接口测试用例

序号	测试对象	测试内容	预期结果	实际结果
1	ProductionOrderDetail	添加订单详情信息	添加成功	添加成功
2	ProductionOrderDetail	修改订单详情信息	修改成功	修改成功
3	SpinningTrackOut	批量出账后纺记录	出账成功	出账成功
4	AddProductionBom	K3 推送的物料清单添加到该系统	添加成功	添加成功

(2)流程测试。流程测试是针对系统中的具体业务流程进行的测试,可以验证整个流程的正确性和稳定性。在基于 SpringCloud 的纺织业制造执行系统中,流程测试可以通过模拟用户的实际操作路径和场景,执行一系列的操作和交互,并验证系统是否按照预期流程进行处理和输出结果。流程测试可以帮助发现系统中的流程逻辑错误、数据传递问题和系统集成的异常情况。根据系统的执行流程进行测试,确保每个环节所写的程序代码能正常运行,比如用户对信息的增删改查是否能操作成功。流程测试具体可分为前纺流程测试、后纺流程测试和 FQC 测试。其中,前纺测试用例如表 4 所示,用于订单生成后,前纺投料和出账是否正常;后纺测试用例如表 5 所示,用于将原材料转换为最后成品的 PE 丝;FQC 测试用例如表 6 所示,用于检验后纺出账成功的货品是否合格。

(3)压力测试。压力测试是为了评估系统在高负载条件下的性能和稳定性而进行的测试。在基于 SpringCloud 的纺织业制造执行系统中,压力测试可以通过模拟大量并发用户请求和数据负载,对系统进行长时间运行和高负载下的性能监测。压力测试可以帮助评估系统的吞吐量、响应时间、资源利用率和扩展性等指标,以发现系统中的性能瓶颈和潜在的问题。由于系统主要给生产车间流水线使用,所以对系统进行压力测试是非常重要的,必须确保系统在长时间的工作情况下不出现任何问题。

在完成整体流程测试并确认无重大问题后,对系统进

Table 4 Pre-spinning test cases

表 4 前纺测试用例

序号	测试对象	测试内容	预期结果	实际结果
1	生产订单号	一条生产订单号是否能生成多条生产订单	一个生产订单号可以生成多条生产订单	一个生产订单号可以生成多条生产订单
		投料所消耗的原材料能否通过物料编码进行追溯	投料所消耗的原材料可以通过物料编码进行追溯	
2	前纺投料	生产订单能否和桶进行正确绑定	生产订单可以正确绑定	由于 K3 系统没有将所有的物料信息推送给系统,导致有些订单校验失败
3	前纺出账		生产订单可以和桶进行正确绑定	生产订单可以和桶进行正确绑定

Table 5 Post-spinning test cases

表 5 后纺测试用例

序号	测试对象	测试内容	预期结果	实际结果
1	后纺入账	前纺出账的桶绑定后纺设备和地标后是否能正常入账	前纺出账的桶绑定后纺设备和地标后能正常入账	前纺出账的桶登记过静置时间,且时间超过了生产订单的静置时间才入账成功。
		入账成功的筒(绳丝筒)是否能自动出账	可以自动出账	自动化流程中,除了第一个和最后一个能自动出账
2	后纺出账	根据车速倒计时自动出账,看板展示 2 分钟倒计时是否能报警	倒计时 2 分钟时看板该地标会爆红	倒计时 2 分钟时看板该地标会爆红
		人工断丝操作能否生成新的货品记录	人工断丝操作可以生成新的货品记录	人工断丝操作可以生成新的货品记录

Table 6 FQC test cases

表 6 FQC 测试用例

序号	测试对象	测试内容	预期结果	实际结果
1	电子秤	通过电子秤的多个 COM 能否自动读取多个数据	可以获取所有数据	可以读取所有数据,但由于当前方法依赖浏览器缓存,通过 key-value 的形式获取,在数据量大的情况下容易造成前一次数据被覆盖的情况
		通过货品 ID 及电子秤读取的数据,判断系统是否能够自动计算百米克重并识别对应商品信息等	可以自动计算百米克重并识别对应商品信息等	可以自动计算百米克重并识别对应商品信息等

行压力测试,系统压力测试用例如表 7 所示。

通过对系统进行接口测试、流程测试和压力测试,结果表明系统整体运行基本符合预期,能够实现大部分操作的自动化,有效提升了系统稳定性与产线效率。但在测试过程中也发现了一些影响性能与使用体验的问题,主要包括:①SQL 查询效率问题,部分查询语句未经过优化,导致在数据量较大的情况下查询响应时间显著增加,页面数据

Table 7 System stress test cases

表 7 系统压力测试用例

序号	测试对象	测试内容	预期结果	实际结果
1	前纺出账	前纺出账的桶自动绑定同一个生产订单的最早一条记录,超过 8 条自动绑定到下一个生产订单。	超过 8 条后自动绑定到下一个生产订单	超过 8 条后自动绑定到下一个生产订单
		货品自动出账后打印机能否正常打印标签内容	货品自动出账后打印机能正常打印标签内容	货品自动出账后,打印机能正常打印标签内容
2	后纺出账			由于使用的 lodop 插件打印标签函数和读取 COM 口数据的函数有时会同时调用,但是 lodop 插件同一时间只能执行一个函数,导致无法打印标签。
3	FQC 校验	校验完成后,打印机能否正常打印标签	校验完成后,打印机能正常打印标签	校验完成后,打印机能正常打印标签
4	基础物料	远程搜索基础物料	能快速搜索想要的基础物料	在数据量较大的情况下,有概率导致页面卡死现象

无法及时回显,影响用户操作体验;②LODOP 插件依赖问题,系统对 LODOP 插件的使用缺乏深入理解和控制,尤其在打印标签等高频操作中,依赖性过高。在并发场景下,LODOP 插件存在兼容性与性能瓶颈,无法满足多用户同时操作的业务需求。

综上,虽然系统在整体功能和流程自动化方面达到了预期目标,但仍需对数据库查询性能和打印模块的兼容性进行进一步优化,以提升系统在高负载与高并发场景下的稳定性与响应效率。

4.3 微服务架构下性能与可靠性对比测试

为进一步验证本文所采用的微服务架构在制造执行系统中的性能优势,特别是在系统响应能力、服务容错、部署效率与可扩展性等方面的改进效果,设计了针对典型系统指标的测试对比。表 8 对比展示了同一制造执行系统在单体架构与微服务架构下的运行表现,涵盖吞吐量、响应时间、并发承载、服务隔离和部署运维等 10 个关键维度,可以系统性揭示微服务架构所带来的性能优化与架构优势。

目前该系统已部署于江苏恒辉安防股份有限公司,处于现场测试和优化阶段。从企业初步反馈与实际运行情况看,微服务架构在模块解耦、独立部署、高并发处理和异构系统集成等方面表现出良好效果,尤其在快速适配多生产工序、异步任务调度与模块级维护方面展现出明显优势。根据运行数据和用户反馈分析,微服务架构相比传统单体系统在以下方面展现出显著优势。

(1)系统性能提升。在相同硬件配置下,系统吞吐能力(TPS)由原单体架构的约 220 req/s 提升至 520 req/s,系

Table 8 Performance and reliability comparison between monolithic and microservices architectures in MES system testing

表 8 微服务架构与单体架构在 MES 系统测试中的性能与可靠性对比

序号	测试项目	单体架构版本	微服务架构版本	对比说明
1	吞吐量(TPS)	220 req/s(每秒请求数)	520 req/s	微服务架构下可单独扩展高频服务模块,提高处理能力
2	平均响应时间	890 ms	330 ms	模块间调用并行化,数据流转时间更短
3	最大并发用户数	120	400	微服务架构下分布式部署,提高系统弹性
4	出账模块部署时间	需整体重启	可热更新	微服务支持模块独立部署,升级维护不影响整体系统
5	模块故障影响范围	全系统卡顿	仅影响本模块	Hystrix 等熔断机制,提升服务容错能力
6	服务扩展难度	高(需整体评估重构)	低(单服务可水平扩展)	微服务具备天然可扩展性
7	日志审计追踪难度	高(日志集中耦合)	低(服务内独立日志,便于链路追踪)	日志清晰分离,便于排查异常链路
8	接口响应稳定性(高并发)	偶发超时、接口阻塞	稳定响应	使用消息队列和限流机制,避免系统崩溃
9	部署方式	一体部署,版本统一	多模块独立部署,支持容器化与 K8s 编排	可适配多场景、多节点灵活部署
10	系统恢复时间(服务异常)	整体重启耗时 5~10 分钟	模块自动重启,60 秒内恢复	微服务结合容器平台具备故障自动恢复能力

统响应速度提升 60% 以上。通过独立部署高并发模块(如称重服务、质量检测服务),实现了弹性扩展,显著缓解了业务高峰期系统响应压力。

(2)开发与运维效率提高。各业务模块以独立服务形式开发与部署,前后端分离,服务之间通过 API 网关统一管理。新业务模块上线不再依赖整体打包与重启,大幅减少了系统维护窗口期,支持灵活迭代和快速更新。

(3)系统稳定性增强。通过服务注册与熔断机制(如 Eureka + Hystrix),系统具备服务级别的故障隔离能力,单个模块故障不会导致整体宕机。日志分析服务独立管理,便于快速定位问题与追溯链路。

(4)业务扩展与异构对接更灵活。各服务支持容器化部署,易于实现多车间、异地分布式部署。系统已成功对接 K3 系统、仓储系统、电子秤和扫码设备等多个异构子系统,体现出良好的平台开放性和集成能力。

5 结语

本文围绕纺织制造行业“流程复杂、数据异构、协同强度高”的行业特点,基于 SpringCloud 微服务架构设计并实现了一套制造执行系统(MES),完成了从系统架构设计、模块功能实现到系统部署测试的完整闭环。系统采用模块化设计理念,结合微服务拆分原则,对订单管理、投料出账、质量检测和设备采集等关键业务进行了服务解耦与逻辑重构,具备良好的可扩展性与可维护性。通过引入服务注册与发现、网关路由、远程调用和分布式事务协调等关键微服务组件,系统实现了跨模块的协同控制与服务弹性扩展能力。同时,系统采用混合数据架构(MySQL + Redis)、状态驱动机制及事件总线,实现了对全过程数据的实时处理与链路一致性保障。性能测试结果表明,在微服务架构下系统吞吐量、响应时间和模块独立部署能力等方面均优于传统单体架构,尤其在服务隔离性与系统稳定性方面表现出明显优势。

目前该系统已部署于江苏恒辉安防股份有限公司,并处于实际测试阶段。从企业初步反馈来看,微服务架构显著提升了系统的部署效率与柔性响应能力,在异构设备接

入、流程分布式管理和模块级运维等方面具有良好的适应性。系统在真实企业环境中运行稳定,展现出良好的性能表现与工程可推广价值。该研究在传统纺织制造系统与现代微服务架构融合的探索中,提出了一套适用于纺织行业的信息系统拆分策略与数据闭环控制机制,为行业数字化转型提供了可借鉴的技术路径。未来研究中,可进一步聚焦于制造执行中的微服务调度优化、流程挖掘与过程智能分析,引入人工智能与大数据技术提升系统智能决策能力,以更好地满足纺织行业在高并发、智能化生产管理中的发展需求。

参考文献:

[1] XIONG J L, REN Q R, SHMUEL T, et al. MSA-Lab: integrated design platform for model-driven development of microservices [J]. Journal of Software, 2024, 35(3): 1280-1306.
熊靖渊,任秋蓉,Tyszberowicz Shmuel,等. MSA-Lab:模型驱动的微服务集成设计平台[J]. 软件学报,2024,35(3): 1280-1306.

[2] CHAI T Y, LIU Q, DING J L, et al. Prospects for research on new intelligent optimization manufacturing model of process industry driven by industrial Internet [J]. Science in China: Technology Science, 2022, 52(1): 14-25.
柴天佑,刘强,丁进良,等. 工业互联网驱动的流程工业智能优化制造新模式研究展望[J]. 中国科学:技术科学,2022,52(1):14-25.

[3] ZHAO F Q, LIU H, ZHU B, et al. Key common technology of industrial intelligence and industrial Internet [J]. Software Guide, 2022, 21(10): 1-8.
赵付青,刘欢,朱波,等. 工业智能与工业互联网共性关键技术[J]. 软件导刊,2022,21(10):1-8.

[4] SHI H J, WANG L P, WANG D. Research on manufacturing execution system architecture based on microservices[J]. Modern Manufacturing Engineering, 2024(1):45-50.
史慧杰,王立平,王冬. 基于微服务的制造执行系统架构研究[J]. 现代制造工程,2024(1):45-50.

[5] LI Y J, LI Z N. Research on key technologies of microservice based manufacturing execution system [J]. Intelligent Manufacturing, 2023(9): 95-101.
李亚杰,李昭楠. 面向微服务的制造执行系统关键技术研究[J]. 制造技术与机床,2023(9):95-101.

- [6] LIAO J B. Design and implementation of MES system based on microservice architecture [J]. Computer Programming Skills and Maintenance, 2023(11):76-78.
廖俊波. 基于微服务架构的MES系统设计与实现[J]. 电脑编程技巧与维护, 2023(11):76-78.
- [7] WANG K S, YANG L. Research and application of intelligent manufacturing warehousing system based on microservices [J]. Computer Knowledge and Technology, 2024, 20(6):68-70.
王魁生, 杨柳. 基于微服务的智能制造仓储系统研究与应用[J]. 电脑知识与技术, 2024, 20(6):68-70.
- [8] LIANG J. Exploration and practice of industrial internet collaborative office based on microservice architecture [J]. Smart Rail Transit, 2023, 60(3):91-94.
梁军. 基于微服务架构的工业互联网的协同办公探索与实践[J]. 智慧轨道交通, 2023, 60(3):91-94.
- [9] LIN Y S, ZENG Z R, LIAO Z X, et al. Design of helmet PaaS platform based on microservice architecture [J]. Journal of Guangdong University of Petrochemical Technology, 2024, 34(1):60-64.
林燕珊, 曾梓锐, 廖振兴, 等. 基于微服务架构的头盔PaaS平台设计[J]. 广东石油化工学院学报, 2024, 34(1):60-64.
- [10] DAI L C, WANG C. Reconstruction and optimization of online lawyer websites under microservice architecture [J]. Software Guide, 2021, 20(9):144-149.
代立晨, 王晨. 微服务架构下的在线律师网站重构与优化[J]. 软件导刊, 2021, 20(9):144-149.
- [11] LI X M, LIANG Z C, LIU L H, et al. Design and implementation of online evaluation system based on micro-service architecture [J]. Software Guide, 2023, 22(8):144-150.
李西明, 梁志才, 刘龙浩, 等. 基于微服务架构的在线评判系统设计与实现[J]. 软件导刊, 2023, 22(8):144-150.
- [12] ZHANG L. Design and implementation of business-to-customer e-mall website based on microservice architecture [J]. Software Engineering, 2021, 24(9):55-57.
张林. 基于微服务架构的商对客模式电商网站的设计与实现[J]. 软件工程, 2021, 24(9):55-57.
- [13] KOU B Y, WANG S L. Research and application of smart lamppost system based on microservice architecture [J/OL]. Software Guide, 2024-01-29. <https://link.cnki.net/urlid/42.1671.TP.20240126.0947.004>.
寇柏源, 王少林. 基于微服务架构的智慧灯杆系统研究与应用[J/OL]. 软件导刊, 2024-01-29. <https://link.cnki.net/urlid/42.1671.TP.20240126.0947.004>.
- [14] ZHENG X, FAN H J, LIU J F. A pre-allocated capacity quota limiting system based on microservice [J/OL]. Computer Science, 2024-04-02. <https://link.cnki.net/urlid/50.1075.TP.20240402.1325.004>.
郑旭, 范红杰, 柳军飞. 基于微服务的预分配额度限流设计研究[J/OL]. 计算机科学, 2024-04-02. <https://link.cnki.net/urlid/50.1075.TP.20240402.1325.004>.
- [15] ZHENG W B, LI Z C, LIU J Z, et al. Business process and service evolution analysis based on relationship dependency model [J]. Journal of Nanjing University (Natural Science), 2022, 58(2):309-319.
郑伟波, 李志超, 刘纪遵, 等. 基于关系依赖模型的微服务流程演化分析[J]. 南京大学学报(自然科学), 2022, 58(2):309-319.
- [16] DING J L, HE Y F, WANG J. Causal intervention-based root cause analysis method for microservices system faults [J/OL]. Journal of Computer Applications, 2024-04-01. <https://link.cnki.net/urlid/51.1307.TP.20240328.1702.014>.
丁建立, 何雨峰, 王静. 基于因果干预的微服务系统故障根因分析方法[J/OL]. 计算机应用, 2024-04-01. <https://link.cnki.net/urlid/51.1307.TP.20240328.1702.014>.
- [17] WANG L, JIANG Y X, LI Q S, et al. A review of microservice fault detection [J]. Chinese Journal of Computers, 2023, 46(11):2342-2369.
王璐, 姜宇轩, 李青山, 等. 微服务故障检测研究综述[J]. 计算机学报, 2023, 46(11):2342-2369.

(责任编辑:陈维捷)