Software Engineering and Applications 软件工程与应用, 2022, 11(4), 751-758
Published Online August 2022 in Hans. http://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2022.114078

Hans汉斯

基于微服务的模具生产管理系统的设计与实现

黄　静，何　岳

浙江理工大学，浙江　杭州

收稿日期：2022年6月15日；录用日期：2022年7月27日；发布日期：2022年8月4日

摘　要

近年来，模具企业蓬勃发展，与此同时先进的计算机技术也广泛地应用在模具企业上，模具设计过程的信息化、自动化和集中化管理是模具公司得以发展和生存的唯一途径，在这样的情况下，利用各种先进的计算机技术、SQL以及NoSQL技术来构建效率高、响应快的模具企业管理系统，进而使企业运转的更加高效，已逐渐成为模具企业自动化管理过程中的至关重要的一步。本文采用Redis缓存、RabbitMQ消息中间件、服务注册与发现等处理机制开发出一套成熟的信息化模具生产管理系统。采用该系统能确保企业内数据传输安全，信息沟通及时，缩短生产周期，使得企业能够建立良好的供应关系。经测试，在系统功能保障的基础上，系统内部稳定运行的性能指标主要包括响应时间、吞吐量、并发数量等各项指标检测均符合需求。

关键词

Redis，RabbitMQ，微服务，模具

Design and Implementation of Mold Production Management System Based on Microservice

Jing Huang, Yue He

Zhejiang Sci-Tech University, Hangzhou Zhejiang

Received: Jun. 15th, 2022; accepted: Jul. 27th, 2022; published: Aug. 4th, 2022

Abstract

In recent years, mold enterprises have developed vigorously. At the same time, advanced computer technology is also widely used in mold enterprises. Informatization, automation and centra-

文章引用: 黄静, 何岳. 基于微服务的模具生产管理系统的设计与实现[J]. 软件工程与应用, 2022, 11(4): 751-758.
DOI: 10.12677/sea.2022.114078

lized management of mold design process are the only way for mold companies to develop and survive. In this case, various advanced computer technologies, SQL and NoSQL technologies are used to build an efficient and responsive mold enterprise management system, thus making the enterprise run more efficiently has gradually become a crucial step in the process of automatic management of mold enterprises. This paper uses Redis cache, RabbitMQ message middleware, service registration and discovery and other processing mechanisms to develop a mature information-based mold production management system. The system can ensure the safety of data transmission and timely information communication, shorten the production cycle and enable enterprises to establish a good supply relationship. After testing, on the basis of system function guarantee, the performance indicators of stable operation within the system mainly include response time, throughput, number of concurrent and other indicators. The tests meet the requirements.

Keywords

Redis, Rabbit MQ, Microservices, Mold

1. 引言

我国的模具公司，主要是在最近十几年里不断发展壮大起来的，但他们却天天都忙于接单、忙于制造、忙于完成、忙于建设，一味追求表面的规模和公司的形式，却忽略了现代公司的信息化管理水平[1]。模具制造行业中一般都是以小型企业居多，它既是典型的精密铸造类行业，也是典型的面向订单式生产企业，具有大量单件无重复制造，对车间生产工艺路线的手工程度要求较高，以机器化生产占据了主导地位，按单制造的优点明显[2]。现有的模具生产管理系统大多是以 ERP (Enterprise Resource Planning，即企业资源计划)系统为辅，主要以表单为介质进行信息的传递，效率极其低下，对人员以及模具的调度较为混乱。

本文为了解决以上问题，开发出一套完全信息化的系统，选取前后端分离的风格作为本系统的开发模式，松解前后端耦合。后端选取 Spring Cloud 作为核心框架，以微服务的风格构建。微服务的架构风格就是将一个服务整体按照功能的不同分解成多个单独的模块；前端选取 Vue.js 作为核心框架，建立该系统的前端整体架构；前端通过 ajax 发送请求到后端，后端进行处理，然后将得到数据返回给前端，前端根据需求将需要展示的数据，采用相关组件来进行渲染，展现在页面上。将该系统部署到企业的服务器上，经过用户的使用反馈，系统能很好地解决企业在业务上的需求，大大提高了企业对模具的调度效率和对产品的产出效率。

2. 微服务架构

随着互联网技术的不断迭代发展，软件开发技术已由第一代的单体架构发展到面向服务的体系结构(SOA) [3]，再到现在的微服务架构。单体架构是指整个系统需要创建一个独立单元作为所有功能模块的基础。所有数据库的操作、后台的业务逻辑以及对后台代码的处理都集中在该独立单元中，后续对后端业务进行扩展与优化也都在该独立单元中进行。随着业务量的不断增加，系统会变得十分臃肿，代码也会十分冗余。以至于处理起来十分困难，让开发人员无从下手。如果其中某个模块存在问题，那么整个

系统将崩溃，导致用户无法使用。

SOA 是面向服务的一种体系结构。SOA 架构将系统里的每一个功能模块划分开来，并通过这些模块之间定义清晰的接口和约束将它们连接起来。SOA 结构根据服务功能的不同将最初的一个结构分解为不同的子结构，SOA 结构如图 1 所示。

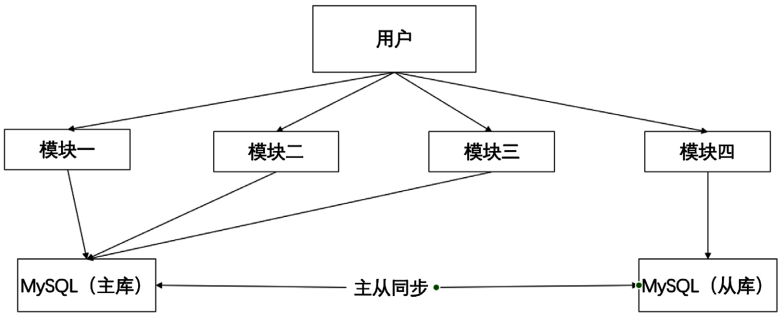

Figure 1. SOA architecture diagram
图 1. SOA 架构示意图

由上图可知，将一个完整的项目划分为多个模块，并且数据库进行主库与从库设计与数据同步，以解决单体式架构所遗留下的问题。

SOA 架构一般会建立一个模块来主导其他模块的集中式管理模式，并且被管理的模块使用的语言与技术集可以是一致的，也可以是不一致的，十分灵活。但是 SOA 也存在一个弊端，就是对系统和服务的定义不清晰，除此之外，SOA 对接口的协议也没有一个统一的规定，开发人员维护起来十分繁琐。

基于 SOA 架构出现的弊端，从而衍生出了一种变体——微服务体系架构。微服务架构，提倡将整个服务模块按照各自功能的不同分解为一组组小型服务，这些被分解后的小型服务通过 Http 通信机制，相互协调合作，互不干扰。每项服务都基于特定的业务，可以在生产环境、类生产环境等中独立运行。微服务架构示意图如图 2 所示。

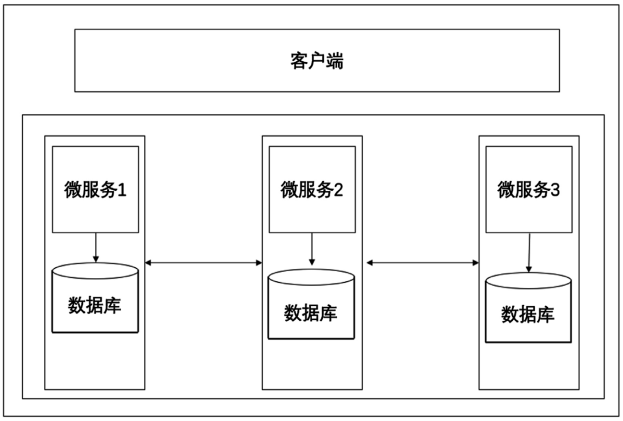

Figure 2. Schematic diagram of microservice architecture
图 2. 微服务架构示意图

3. 系统相关技术选型

本系统采用前后端分离的开发模式，前后端各负其责，提高开发效率。后端采用的是 Spring Boot 框

架，结合 Spring Cloud 构建微服务，数据库采用 MySQL 8.0.18 实现，数据缓存采用 Redis 实现，消息队列采用 RabbitMQ 实现，前端架构采用 Vue.js 开发框架[4]。

Spring 是一个开源且轻量的框架，任何的基于 Java 的程序都可以从该框架中获得好处，降低代码耦合度与复杂性。但随着程序功能越来越丰富，越来越复杂，Spring 也逐渐力不从心，满足不了日益增多的需求。所以 Spring Boot 在这种场景下横空出世。Spring Boot 是基于 Spring4.0 设计的，既有 Spring 框架的出色能力，而且在此之上还简化了 Spring 的配置，更进一步降低了应用程序的构建以及开发过程[5]。

Spring Cloud 并不是一个独立的框架，它是一系列框架的集合。Spring Cloud 必须依赖于 Spring Boot，它利用 Spring Boot 操作简便的风格高明地使分布式系统基础设施的开发变得更加容易，像服务注册与发现、路由中心、配置中心、消息总线、负载均衡等，都可以做到一键启动和部署[6]。Spring Cloud 通过 Spring Boot 风格进行再封装屏蔽掉了复杂的配置和实现原理，最终给开发者留出了一套简单易懂、易部署和易维护的分布式系统开发工具包[7]。

Redis (Remote Dictionary Server)，即远程字典服务，是一个开源的使用 C 语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value 数据库，并提供多种语言的 API [8]。它支持存储的 value 类型有很多，包括 string (字符串)、list (链表)、set (集合)、zset (sorted set--有序集合)和 hash (哈希类型)。这些数据类型都支持 push/pop、add/remove 及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。为了保证效率，数据都是缓存在内存中。区别的是 Redis 会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了 master-slave (主从)同步。开发者可将短期内不会发生变化的数据存入 Redis 中，从而减轻数据库的压力，提高系统响应速度，增强用户体验。

RabbitMQ 是实现了高级消息队列协议(AMQP)的开源消息代理软件(亦称面向消息的中间件)。RabbitMQ 服务器是用 Erlang 语言编写的，而集群和故障转移是构建在开放电信平台框架上的[9]。所有主要的编程语言均有与代理接口通讯的客户端库。开发者可以使用 RabbitMQ 来进行服务间的通信，松解服务与服务之间的耦合度。与此同时，RabbitMQ 还可以对服务进行流量削峰，缓解服务器压力。

4. 系统设计与实现

4.1. 系统总体架构设计

本系统总体架构图如图 3 所示。

各个模块详细职能如下：

1) 系统前端

本系统前端主要使用 Vue.js 框架、Element UI 和 Echarts 实现，以图表的形式完整清晰地展示相关数据。

2) 数据库、Redis 缓存

本系统数据库采用 MySQL 8.0.18，将本系统的所有数据都放在其中。MySQL 是最流行的数据库之一，是一个免费开源的关系型数据库管理系统。MySQL 不仅可以在 Windows 系列的操作系统上运行，还可以在 UNIX、Linux 和 Mac OS 等操作系统上运行。所以 MySQL 的跨平台性保证了其在 Web 应用方面的优势[10]。本系统将用户、部门、商品生产订单、生产订单、派工表、模具申请记录、出入库记录等所有模具生产数据信息都存储在数据库中。由于在模具生产的过程中所产生的数据信息都会存在数据库中，经过长时间的积累，数据量已十分庞大，从而会导致数据库检索的速度慢等问题。为了解决该问题，在系统设计初期，将访问频率高、体量庞大的数据以缓存的形式存入 Redis 中，并设置相关限定条件，如数据过期时间等，从而实时可控地更新数据，增强数据读写性，提升系统响应速度。

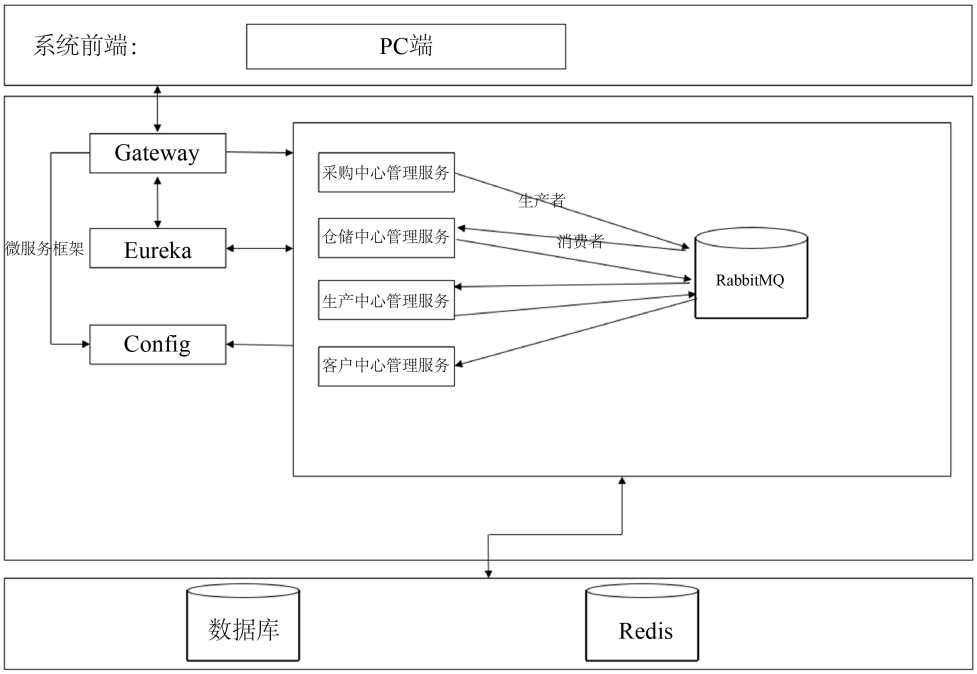

Figure 3. System overall architecture diagram
图 3. 系统总体架构图

3) 服务端

微服务框架无论是对于技术开发，还是对于企业业务需求，都能够贴切地满足，不仅在开发阶段易于上手，而且在维护升级阶段也具有易维护与可拓展的特性。针对企业采购成本节约问题与库存压力，利用 JIT 采购模式调整采购与库存关系，再进行采购、生产、库存成本建模，整体求得最优并给出采购建议与计划调度，有利于企业资源调度与辅助决策。根据模具生产流程划分出四个微服务，分别是采购中心管理服务、仓储中心管理服务、生产中心管理服务、客户中心管理服务，各个模块对应模具生产管理的各个流程。

4.2. 系统数据库设计

本系统数据库采用 MySQL 8.0.18，在此仅对部分库表的结构设计进行说明。数据库部分表设计如图 4 和图 5 所示，其中产品生产订单表(pdm_production_order)存放企业的所有的生产订单信息，产品销售订单(pdm_sale_order)存放企业的所有销售订单，产品分类表(pdm_product_category)存放企业生产所有的产品分类信息，计划生产表单(pdm_plan_schedule)存放生产中心的计划生产信息，计划生产派工表(psm_assign_process)存放生产中心的派工信息，派工后报工表(psm_report_process)存放生产中心进行派工后的报工信息，工序工价表(fdm_process_price)存放生产中心的工价工序信息，产品打样表(pdm_product_proof)存放对即将要生产的产品进行打样的信息，首件确认报告基本信息(qc_report)存放的是对首件产品进行确认后的信息，首件报告检查项(qc_reportdtl)存放的是对首件确认报告的补充信息，巡检记录表详情(qc_inspection_orderdtl)存放的是对生产的产品进行巡检时记录详情。pdm_production_order 表和 pdm_sale_order 表是一对多的关系，每一张产品生产订单包含多张销售订单，两表通过外键 work_order_id 关联。pdm_product_proof 表和 qc_report 表是一对多的关系，每张产品打样表对应多条首件确认报告信息，两表通过外键 proof_id 关联。

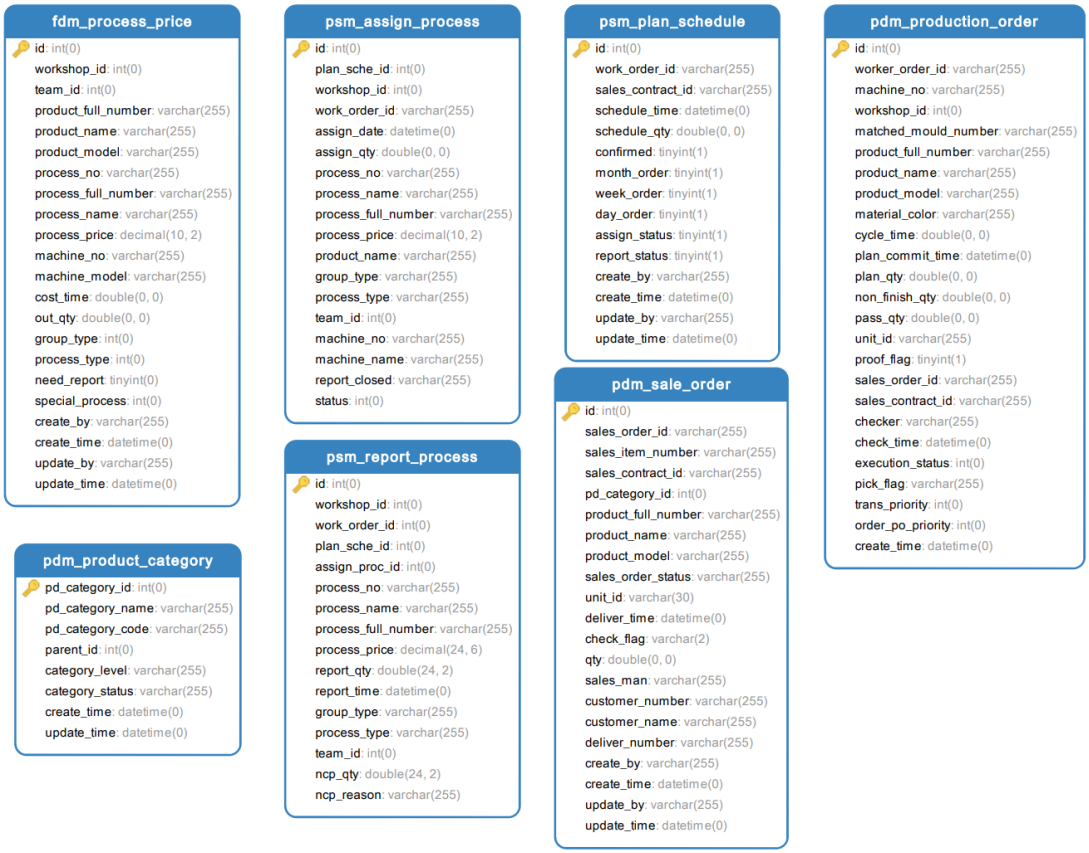

Figure 4. Part of the database table and the corresponding diagram
图 4. 数据库部分表及对应关系图

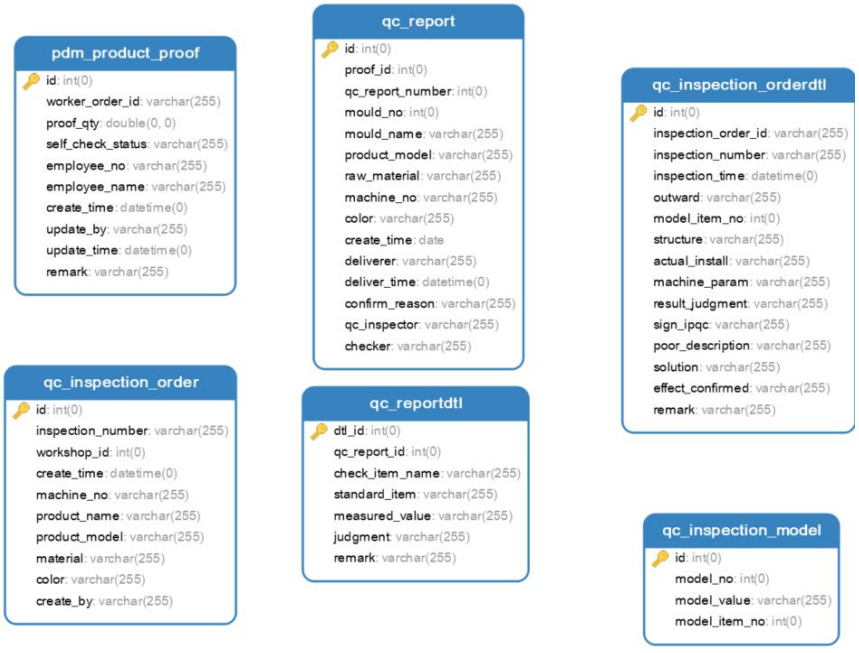

Figure 5. Part of the database table and the corresponding diagram
图 5. 数据库部分表及对应关系图

4.3. 系统服务端设计与实现

4.3.1. 微服务框架设计

本系统微服务框架采用 Spring Cloud 的微服务架构进行搭建。Spring Cloud 全家桶包括服务注册与发现——Netflix Eureka、负载均衡：客户端负载均衡——Netflix Ribbon，服务端负载均衡——Feign (其也是依赖于 Ribbon，只是将调用方式 RestTemplete 更改成 Service 接口)、断路器——Netflix Hystrix、服务网关——Netflix Zuul、分布式配置——Spring Cloud Config [11]。

数据缓存采用 Redis，它是一个开源的高性能的 key-value 非关系型数据库。Redis 通过 RDB 和 AOF 两种方式来实现数据持久化，可以将内存中的数据存入磁盘，再次加载时可以使用；Redis 不仅有 key-value 类型，还有 string、set、zset、list 等数据类型；Redis 读数据的速度可达到 110,000 次/秒，写数据的速度可达到 81,000 次/秒。在本系统中，运用 Redis 实现业务数据的缓存，将体量庞大且短时间内不会发生变化的数据存入其中，减小了数据库压力，提升了系统响应速度，增强了用户体验。

4.3.2. 微服务功能设计

根据模具生产管理整体流程，本系统划分出四个微服务，分别是仓储中心管理微服务、采购中心管理微服务、生产中心微服务、客户关系中心管理微服务，以下为各个微服务具体的功能描述。

1) 仓储中心

仓储中心主要包括模具仓储、原料仓储、成品仓储三部分。

模具仓储负责模具类别与模具基本信息的维护；对生产任务中，上下模具申请进行审批并伴随模具相关文件的发放、回收；记录模具出入库；审批过后需要追踪模具与生产活动。物料仓储负责物料类别与物料基本信息的维护；对物料出入库进行记录；对物料设置安全库存。成品仓储主要负责成品的信息维护以及成品出入库记录。

2) 采购中心

采购中心主要负责供应商基本信息的管理，包括供应商基本信息、供应商合同、供应商发票信息维护；采购过程发生的物料采购订单信息维护。

3) 生产中心

工厂管理：工厂的基本产能信息维护如员工、车间、班组、机器、工序工价等。计划管理：依据销售订单制定产品生产订单与物料需求单，此时需要判断库存水位，以此判断是否需要进行物料采购计划；依据生产订单制定主生产计划；主生产计划审核后进行车间派工、任务下达。生产调度：车间领班接收到派工单后，依据派工单进行生产，生产后需要 QC 质检员进行物料标识，检验是否合格。由于厨电产品生产需要模具与机器的配合，在此之前需要进行上模申请，一批半成品生产结束后需要下模并归还模具；该批订单完工后可报工核算工资。需要注意的是，某些新产品需要打样，经 QC 质检员首件确认后方可开机生产。

4) 客户关系中心

一方面负责客户管理即客户信息维护；另一方面客户的销售订单、销售合同、销售发票的管理；此外还需要维护产品的基本信息、相关经营信息等。

4.4. 系统前端设计与实现

PC 端和移动端共设计了五个模块，分别是仓储中心管理、采购中心管理、生产中心管理、客户关系中心管理。

5. 结论

本系统对模具的生产管理系统进行了在数据结构和功能上的设计实现，在设计过程中充分考虑了系统的实际应用情况，从系统数据架构方面和系统的功能两个方面对系统进行了详细的设计实现，与试点单位现有的模具管理 ERP 系统相比，在一定程度上实现了企业完全信息化，脱离了纸质文件传递信息的弊端。通过系统的相应功能，有助于模具的设计与产品生产的全面集成，并提高了客户的满意度。

参考文献

[1] 模具行业信息[J]. 模具工业, 2022, 48(5): 6-9.

[2] 罗志良, 胡永珊, 柳志林, 张斌, 肖莹, 黎力超. 增材制造技术在模具行业的应用现状及展望[J]. 机械工程与自动化, 2022(2): 224-226.

[3] 周海荣. 从应用场景来探讨物联网时代应急广播体系技术架构搭建[J]. 中国传媒科技, 2021(6): 143-148. https://doi.org/10.19483/j.cnki.11-4653/n.2021.06.046

[4] 黄静, 吴涵. 基于 Spring Cloud 的工厂可视化管理系统的设计与实现[J]. 软件工程, 2021, 24(10): 59-62+58. https://doi.org/10.19644/j.cnki.issn2096-1472.2021.010.014

[5] 牛宵, 董林, 苏醒. 基于容器技术的空间数据管理方案[J]. 地理空间信息, 2022, 20(4): 143-146.

[6] 杨晟, 罗奇. 基于 Spring Boot 的在线选课系统的设计[J]. 网络安全技术与应用, 2022(6): 53-54.

[7] 沈萍萍, 关辉, 韦阳, 梅秀玲. 分布式集群系统架构设计及应用部署[J]. 信息技术与信息化, 2021(1): 159-162.

[8] 刘世超, 杨斌, 刘卫国. 高性能高可用 Redis 客户端的设计与实现[J]. 电子技术应用, 2022, 48(1): 46-52+58. https://doi.org/10.16157/j.issn.0258-7998.212432

[9] 余社平, 刘自立, 何芳. 基于 RabbitMQ 构建在线教育培训平台的设计与实现[J]. 大学, 2021(11): 113-114.

[10] 陈中凯. 试论在计算机软件开发中数据库安全设计的应用实践[J]. 信息系统工程, 2020(12): 119-120+122.

[11] 薛云兰, 黄嘉浩, 邵桐杰. 微服务架构的在线课程学习系统的研究与设计[J]. 计算机时代, 2022(5): 130-133+137. https://doi.org/10.16644/j.cnki.cn33-1094/tp.2022.05.034