经验交流

文章编号:1671-251X(2020)02-0094-06 DOI:10.13272/j.issn.1671-251x.2019070056

微服务在煤矿监控类软件开发框架中的应用

荆诚[1,2], 王爱军[3]

(1.中煤科工集团常州研究院有限公司,江苏 常州 213015;
2.天地(常州)自动化股份有限公司,江苏 常州 213015;
3.内蒙古准格尔旗宏丰煤炭运销有限责任公司 红树梁矿,内蒙古 鄂尔多斯 010400)

扫码移动阅读

摘要:针对煤矿监控类软件开发面临版本混乱、重复开发、维护困难,软件的定制化修改导致不同煤矿监控软件之间的通信变得困难等问题,提出了一种采用微服务架构的煤矿监控类软件开发框架。基于微服务架构,通过规范化开发流程、简化技术栈优化煤矿监控类软件的开发流程;将基础业务固化在开发框架中,专有业务通过微服务的方式进行加载运行,减少了基础代码的重复编码工作,并使得专有业务可以重用;沙盒运行方式让微服务的部署不受运行环境影响,部署方便,跨平台移植性强,微服务托管平台可对微服务进行统一的版本管理。实际应用结果表明:采用微服务架构的煤矿监控类软件通过将常用功能拆分为微服务,可以最大程度减少软件功能的重复开发,微服务的数据存储效率比现有垂直架构更高,使用 Docker 镜像部署,软件安装过程更为便捷,为监控类软件开发提供了一种更为高效的开发方式。

关键词:煤矿安全监控;煤矿监控类软件;微服务;云端托管;基础业务;专有业务;沙盒运行
中图分类号:TD672 文献标志码:A

Application of microservice in development framework of coal mine monitoring software series

JING Cheng[1,2], WANG Aijun[3]

(1.CCTEG Changzhou Research Institute, Changzhou 213015, China; 2.Tiandi(Changzhou) Automation Co., Ltd., Changzhou 213015, China; 3.Hongshuliang Coal Mine, Jungar Banner Hongfeng Coal Transport and Marketing Co., Ltd., Erdos 010400, China)

Abstract: Development of coal mine monitoring software series faces version confusion, repeated development and maintenance difficulties, and the customized modification of software makes communication between different monitoring software series of coal mine difficult. In order to solve the above problems, a development framework of coal mine monitoring software series adopting microservice architecture was proposed. The development process of coal mine monitoring software series is optimized by standardizing development process, simplifying technology stack and adopting microservice architecture. The basic business is solidified in the development framework, and the proprietary business is loaded and run by means of microservice, which reduces the repetitive coding of the basic code and makes the proprietary business reusable. Sandbox operation mode makes the deployment of microservices not affected by the running environment, easy to deploy, and has strong cross-platform portability, microservice hosting platform can carry out unified version management of microservices. The practical

收稿日期:2019-07-19;**修回日期**:2019-10-20;**责任编辑**:张强。
基金项目:天地(常州)自动化股份有限公司研发项目(2018GY003)。
作者简介:荆诚(1989-),男,江苏丹阳人,助理工程师,硕士,现主要从事煤矿相关软件系统的研发工作,E-mail:jingcl1989@163.com。
引用格式:荆诚,王爱军.微服务在煤矿监控类软件开发框架中的应用[J].工矿自动化,2020,46(2):94-99.
JING Cheng, WANG Aijun. Application of microservice in development framework of coal mine monitoring software series[J]. Industry and Mine Automation, 2020, 46(2): 94-99.

application results show that coal mine monitoring software series using microservice architecture can minimize repeated development of software functions by splitting common functions into microservices. The data storage efficiency of microservice is higher than that of the existing vertical architecture, and the software installation process is more convenient when Docker image is used for deployment, which provides a more efficient development mode for monitoring software series development.

Key words: coal mine safety monitoring; coal mine monitoring software series; microservice; cloud hosting; basic business; proprietary business; sandbox operation mode

0　引言

煤矿监控类软件对煤矿的安全生产至关重要,在煤矿日常生产中监控类软件可提供 24 h 不间断的各项指标监控,并可根据监控数据的变化采取告警、断电等措施[1-2]。传感器设备监控、人员位置监控、井下车辆位置监控等均属于监控软件功能范畴。煤矿现场根据自身业务需求会配备相关的煤矿监控类软件,由于不同的煤矿有不同的使用场景和个性化的业务需求[3],这使得煤矿监控类软件虽然主要功能类似,但同一款煤矿监控类软件却不能不加修改就应用于多个煤矿[4],同款软件的不同配置及二次开发使得煤矿监控软件版本复杂,难以管理[5]。这种软件开发模式给开发人员带来了巨大的工作量,同时也使得运维人员的维护工作变得复杂和充满不确定性[6-7]。

个性化定制使得煤矿监控类软件的主数据结构也经常变化,主数据结构的不确定性导致不同煤矿监控软件之间的通信也因为软件定制化修改而变得困难重重,影响煤矿监控类软件的融合工作。

针对煤矿监控类软件开发面临的现状和问题,提出了一种采用微服务架构的煤矿监控类软件统一开发框架,采用微服务架构的开发框架将基础业务固化在框架中,专有业务提取为公共组件,以微服务的方式供开发人员按需加载调用。该开发框架采用了当前成熟、流行的开发语言,开发模式为前后端分离。使用开发框架进行软件开发,可降低技术复杂度,同时借助技术中的跨平台特性将降低多平台版本软件的重复编码工作。

1　煤矿监控类软件开发框架

煤矿监控类软件开发框架架构如图 1 所示,自下而上分为设备层、通信接入层、应用服务层、数据发布层和数据交互层。组件和版本管理平台作为支撑模块,提供例如基础设施、通用组件、存储、规范、公共技术栈和组件版本管理等支撑功能。

1.1　设备层

设备层包含传感器、读卡器、执行器、分站、网关等,设备可以采集所处环境的各种数据,并通过网络与上位机通信,上位机可以利用设备数据来监控煤矿井下的各项数据指标,也可以通过下发指令来控制设备的运行。

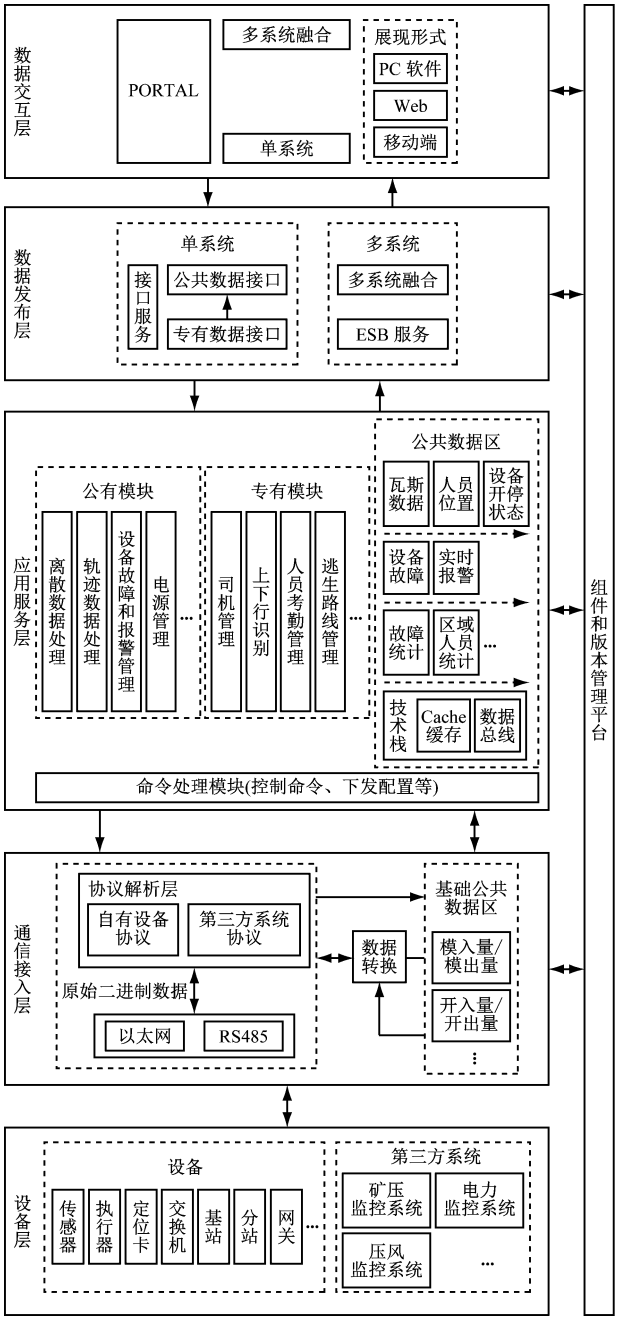

图 1　煤矿监控类软件开发框架架构

Fig. 1　Development framework architecture of coal mine monitoring software series

1.2 通信接入层

通信接入层主要用于上传井下设备采集的数据和利用上位机下发配置和控制命令，可提供各种设备的协议解析和数据包装功能，设备和第三方系统可以通过以太网和 RS485 等通道连接到上位机监控系统。

协议解析的主要作用：针对不同设备协议的数据进行数据格式的拆解，并转换成通用的数据结构，或者把通用数据结构封装为设备协议格式的数据。协议包括：自有设备所使用的协议：Caribus、自定义二进制协议、Modbus TCP/Modbus RTU、CAN；第三方系统协议：OPC/OPC UA、自定义文本协议等。根据接入设备的不同，通信接入层需要提供配套的解析协议。相对于非独立的协议解析模块，通信接入层可以将各种协议解耦包装为可替换模块，根据监控系统接入的设备不同按需加载所需的解析协议，独立的通信接入层可以大大减少硬件协议的适配工作。

1.3 应用服务层

应用服务层将煤矿监控软件中的很多典型业务解耦抽离，区分公有和专有业务，并使用订阅发布机制来进行数据通信。

(1) 订阅发布机制。该机制提供了一对多的消息发布，以便在耦合度较低的系统间进行数据通信。例如，经过协议解析后的数据对外发布，发布者称为生产者，其他应用服务模块可以采取订阅的方式来获取生产者的数据，接收数据的服务模块称为消费者，订阅发布机制因此也被称为生产者消费者模式。

(2) 业务模块(公有、专有)。煤矿监控类软件开发框架可提供监控类软件中具有共性的公有模块，比如离散数据处理、轨迹数据处理、设备故障和报警管理、电源管理等。这些业务被固化在框架中，以便框架使用者随时调用。专有模块由具体的监控类软件开发人员根据专有业务需求来编写，比如胶轮车监控的司机管理模块、上下行识别模块、人员定位系统的考勤模块、逃生路线管理模块等。

(3) 公共数据区。存储经过处理后的业务数据，比如瓦斯数据、人员位置数据、设备开停状态数据、设备故障和报警数据、电源状态数据等。

(4) 命令处理模块。在应用中，用户通过 APP 或者 PC 可以下发对底层设备的控制指令，控制指令经过命令处理模块处理，再通过通信通道下发至设备实现控制操作。

1.4 数据发布层

数据发布分为单系统的接口服务和多系统融合的 ESB(Enterprise Service Bus，企业服务总线)服务。WebAPI 应用在移动端和 Web 端。ESB 宜用于多系统的集成类应用中，特别是涉及到第三方厂家开发的系统。

数据发布层负责将数据发布至数据交互层，用户通过数据发布层提供的数据获取方式获取所需数据。对于单系统的软件，通过接口(WebAPI)请求的方式获取数据。对于多系统融合的软件，依赖 ESB 技术来进行数据通信，ESB 技术提供了公共数据通道，各系统将其他系统所需的数据对 ESB 开放，其他软件可通过消息总线获取本软件所关注的数据，达到了跨系统通信的目的。

1.5 数据交互层

数据交互层的主要功能是与用户交互，将底层数据通过层层通信，最终以可视化的方式展示在用户面前，展示的方式包括图形(车辆轨迹、人员轨迹、环境参数)、报表(瓦斯数据、考勤)、表格(设备状态、下井人员信息)等。数据交互层通过捕获用户的点击、滑动和其他交互动作进行数据查询，下发指令控制设备等操作。可视化展示有多重途径，例如 Web 端网页展示、PC 软件展示、移动端展示(适配 Android 和 IOS 等智能手机操作系统)。以往，开发一款软件如果要适配 PC、Web 和移动端，需要进行多次设计和开发，并且需要有对应专业技能的研发人员，人力物力消耗巨大。微服务软件开发框架采用的开发技术可以一次开发编译为多种平台的运行版本，这大大减少了开发成本和开发时间。同时，统一的开发技术、精简的技术栈让研发人员可以高效、高质地完成研发工作。

2 微服务和 Docker

2.1 微服务架构

软件研发初期，架构的选择是相当重要的，软件架构决定了该款软件的各方面特性[8]，软件架构的选择通常是根据实际需求来决定的。软件架构从简单到复杂、从单体到分布式出现了很多种，可以应对各种需求。

(1) 单体架构：即所有软件功能集成在单独的项目工程中，应用界面和后台数据是分离的，性能提升和扩展主要是通过后端部署数据库集群。这种架构比较简单，优点是开发成本低、效率高，通常应用于小型项目。其缺点也同样明显，由于全部功能都在一个项目中，这使得后期的维护和扩展会变得越来越困难，同时性能提升也有瓶颈，提升性能所需部署的设备成本较高[9]。

(2) 垂直架构：这种架构通常应用于由多个子系统组成的项目，由于项目之间有数据交互需求，

多个单体架构的项目会组成一个更大的项目,这种以单体架构项目为单位的架构就是垂直架构,单体架构项目之间会有相互调用,数据共享,为了保证项目数据一致,还需要数据库同步。其优点就是架构依然不算复杂,同时可以通过垂直拆分,让项目中的子系统不会无限扩展,而且项目的划分可以在不同的项目里采用不同的技术实现。其缺点与单体架构类似,因为其本质上就是单体架构项目的一种融合,最终所有的功能依然是在一个完整的大项目中,这使得项目规模庞大,维护和扩展都不简单,性能提升成本较高,且有性能瓶颈。

(3) SOA:SOA(Service Oriented Architecture,面向服务的架构)的思想是面向服务,将可重用的功能抽离变为组件,并将这些公用组件以服务的方式提供给各系统。为了减少系统中各种接口的耦合,SOA 中组件之间的通信采用的是 ESB。SOA 的优点是组件可重复利用,提高了系统的可重用性,提高了开发效率,同时还可以针对不同的业务需求定制集群和优化方案。SOA 的缺点是这种架构中系统和服务边界模糊,其服务抽取是站在系统的角度抽取的,粒度较大,大部分情况下抽取的组件是松耦合的,但有时服务之间会有相互影响。SOA 架构采用中央管理方式,采用这种架构的系统模板时各服务之间可以正常交互,保证了系统的正常运行。

(4) 微服务架构:与 SOA 有诸多相似之处,不同之处在于 SOA 是从系统往组件设计的,而微服务是直接从组件出发,在需要的时候任何业务都可以抽取为组件,从而快速开发迭代[10]。微服务更像是 SOA 的特殊化版本,为了彻底解耦组件,微服务中所有的服务都是松耦合的,且单个服务的部署无需考虑其他服务的影响,这使得业务可以变得更为专注[11-12]。不同于 SOA 的中央管理方式,微服务采用分散管理方式[13],采用这种架构最大的好处就是可以独立开发某个业务,开发人员无需考虑自己与其他业务的交互,每个微服务都有自己私有的数据库持久化业务数据[14],同时微服务采用的接口为 WebAPI 接口[15],使得数据的输入输出更加便捷、规范。

煤矿监控类软件框架采用微服务架构使得不同业务的开发人员可以独立开发自己的服务,并且可以便捷地发布自己的服务,同时又可以享受到使用公共服务带来的开发效率的提升,减少了重复代码的编写。

2.2　Docker

Docker 是一种开源的应用容器,微服务中的每个服务通过打包封装成镜像,利用 Docker 创建容器后加载镜像就可以进行独立的维护和部署,且借助 Docker 的沙盒机制,这个封装在 Docker 中的服务可以正常运行在 Windows、Mac OS 或是 Linux 下,这意味着每个服务封装后即具备了跨平台的特性。而一台计算机同时运行多个容器,即可将各种服务组合成一个复杂的系统,通过这种方式来完成各种业务需求。

3　微服务在煤矿监控类软件中的应用

3.1　煤矿监控软件中的微服务提取

以煤矿监控系统中的监控软件为例来说明煤矿监控类软件中微服务的提取。煤矿监控系统的功能模块众多,包含了各种图形和报表,且有多重告警方式和多种实时状态。根据微服务框架架构,将煤矿监控系统中的业务划分为公共业务和专有业务,其中日志管理、用户权限、短信通知等作为监控类软件的常用功能,将其解耦抽离成为公共业务,并固化在监控类软件开发框架中,在加载框架时即可以启用这些功能。瓦斯日报、设备状态、实时数据和历史曲线等这些与监控业务密切相关的模块则以微服务的方式进行开发和加载。煤矿监控软件微服务架构如图 2 所示。

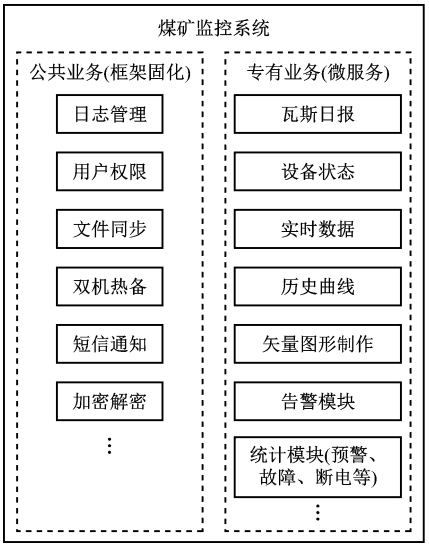

图 2　煤矿监控软件微服务架构

Fig. 2　Microservice architecture of coal mine monitoring software

煤矿监控系统中常用的查询微服务都是幂等性(一次和多次请求某一个资源对于资源本身应该具有同样的结果)的,这就意味着查询业务的水平扩展性会相当出色。微服务本身是一种云原生技术,其最大优势是支持便捷的云端部署,当云端托管条件成熟时,可以在云端直接部署微服务架构的监控软件,利用负载均衡技术,云端部署的监控软件可以按需扩展性能,低成本地实现高质量的软件部署,改变了煤矿软件性能受现场运行环境约束的现状。

对于微服务架构性能的提升,可以通过典型业

务的查询时间对比来进行说明。不同架构的数据查询速度对比结果见表 1。从表 1 可看出，采用垂直架构的历史数据查询，耗时要远远大于采用微服务架构的历史数据查询，因为在微服务架构中将数据区分为热数据和冷数据，热数据由于访问频繁，将其加载入高速缓存中，由专门的热数据管理微服务来进行管理，高速缓存中的数据查询操作速度相比于常规的磁盘查询速度有了质的提升。

表 1 不同架构的数据查询速度对比

Table 1 Comparison of data query speeds of different architectures

查询数据量/条	垂直架构查询速度/ms	微服务架构查询速度/ms
10 000	16 400	1 300

3.2 煤矿监控软件中的微服务开发

微服务开发流程如图 3 所示。首先熟悉软件框架技术体系选定的开发技术，然后按照普通软件的开发流程正常开发，前端项目或后端项目都可以生成微服务。

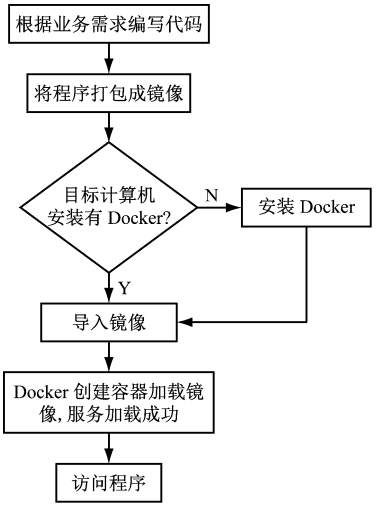

图 3 微服务开发流程

Fig. 3 Microservice development process

计算机根据需求编写业务代码，经测试后通过 Docker 打包工具将项目打包为镜像，若将要部署的计算机没有 Docker 环境，则需要安装 Docker。在 Docker 环境下，导入镜像后创建容器并加载镜像，通过对外部映射的端口或磁盘镜像来访问项目内容。自此微服务从开发至运行的完整流程完毕，流程中对运行环境的依赖很少，节约了以往煤矿监控软件需要大量环境配置所耗费的时间。

3.3 煤矿监控软件中的微服务部署

在微服务架构下，煤矿监控软件微服务的部署流程如图 4 所示。

每个相关的业务都可成为一个微服务，并且界面和数据库可以分开部署，部署完成后，通过

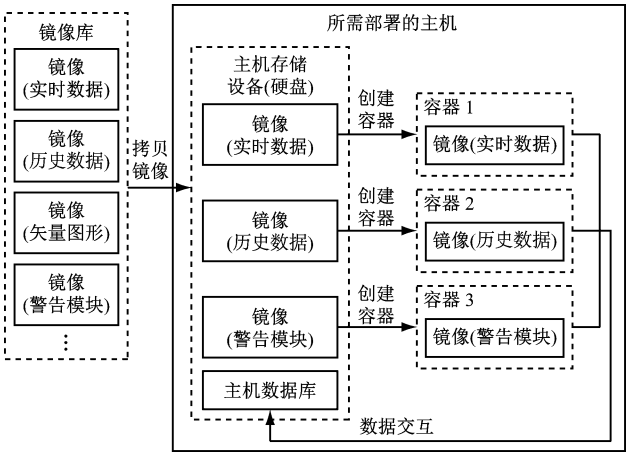

图 4 微服务部署流程

Fig. 4 Microservice deployment process

Docker 对外映射的地址即可正常访问。镜像库保存了所有已经开发完成的微服务，镜像库的维护和托管对于框架的正常运行至关重要，如使用本地服务器托管可以通过 FTP 访问拷贝；如使用云服务器仓库托管可使用 Docker 相关命令拉取所需的镜像。镜像库管理可以借助图形界面工具，管理工具允许预先创建业务模板，提前配置好相关参数，以便操作人员根据需求一键部署所需服务，通过图形管理工具，还可以便捷地查看镜像运行状态和资源占用情况，以便在资源不足时提前做出响应。

以一个前端项目为例进行项目部署，首先将镜像导入，再创建服务映射端口，该项目映射的前端端口为 8088，通过浏览器访问 8088 端口(图 5)，简单几步部署，一个微服务即可正常运行，根据需求，可以从镜像库中拷贝所需镜像进行部署，并且通过磁盘映射，各微服务之间可以共享交互数据，从而实现一个满足复杂需求的完整功能。

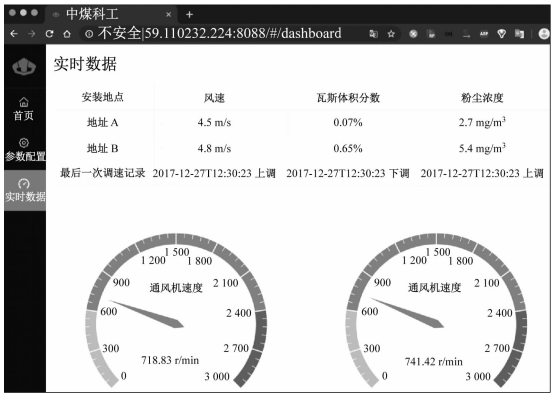

图 5 微服务运行界面

Fig. 5 Microservice running interface

4 结论

(1) 采用微服务架构的煤矿监控类软件开发框架消除了煤矿监控软件之间的技术壁垒，减少了相同需求的软件的重复开发劳动，从源头上降低了煤

矿监控类软件的运维工作复杂度,改变了现有监控类软件的开发模式,大型项目可从框架中获得基础服务的支持,专有业务可拆分为多个微服务,便于开发者并发进行开发工作,提高工作效率。

(2) 微服务云原生特性为监控类软件的云端托管提供了底层的支持,为软件实现分布式运行打下了基础。微服务大大降低了功能之间的耦合度,使得功能开发可以独立完成,同时使后期维护、修改也变得更为可控。沙盒运行方式让微服务的部署不受运行环境影响,部署更方便,跨平台移植性强。

参考文献(References):

[1] 贺耀宜.煤矿本质安全支撑体系平台研究[J].工矿自动化,2012,38(11):5-8.
HE Yaoyi. Research of support system platform of intrinsic safety of coal mine [J]. Industry and Mine Automation,2012,38(11):5-8.

[2] 原雅茹,谢斌红,潘理虎,等.煤矿安全监控领域可变性模型[J].工矿自动化,2017,43(10):43-47.
YUAN Yaru, XIE Binhong, PAN Lihu, et al. Variability modeling of coal mine safety monitoring and control field[J]. Industry and Mine Automation, 2017,43(10):43-47.

[3] 潘涛.煤矿生产系统集成的层次结构及其标准化问题研究[J].工矿自动化,2014,40(9):19-23.
PAN Tao. Research of integrated architecture of coal mine production system and its standardization problems[J]. Industry and Mine Automation, 2014, 40(9):19-23.

[4] 张小艳,蔡攀亮.大型煤炭企业运销信息化框架研究[J].工矿自动化, 2014,40(4):93-95.
ZHANG Xiaoyan, CAI Panliang. Research of informatization framework of transportation and sale for large coal enterprise [J]. Industry and Mine Automation, 2014,40(4):93-95.

[5] 王维.煤矿设备信息管理系统的构件化设计方法[J].工矿自动化,2012,38(5):33-35.
WANG Wei. Component design method of information management system of mine equipment [J]. Industry and Mine Automation, 2012, 38(5): 33-35.

[6] 朱哲君.煤炭企业信息服务能力评价模型[J].工矿自动化,2014,40(1):54-58.
ZHU Zhejun. Evaluation model of information service capability of coal enterprise[J]. Industry and Mine Automation, 2014,40(1):54-58.

[7] 陈峤鹰.矿用设备检测实验室智慧服务体系[J].工矿自动化,2015,41(2):79-82.
CHEN Qiaoying. Intelligent service system for mine-used equipment testing laboratory[J]. Industry and Mine Automation, 2015,41(2):79-82.

[8] 徐晓耘,左松林,倪妍妍.基于微服务架构的电力营销信息系统研究[J].信息技术,2019(2):160-166.
XU Xiaoyun, ZUO Songlin, NI Yanyan. Design and research of integrated power marketing information system based on micro-application and micro-service architecture:[J]. Information Technology ,2019(2): 160-166.

[9] 欧阳荣彬,王倩宜,龙新征.基于微服务的数据服务框架设计 [J].华中科技大学学报(自然科学版),2016,44(增刊 1):126-130.
OUYANG Rongbin, WANG Qianyi, LONG Xinzheng. A data service framework based onmicroservices[J]. Journal of Huazhong University of Science and Technology (Natural Science Edition),2016,44(Sl): 126-130.

[10] 侯一鸣,徐泉,李亚杰,等.基于物联网和工业云的选矿设备状态监控系统[J].计算机集成制造系统,2017,23(9):1972-1982.
HOU Yiming,XU Quan, LI Yajie,et al. Monitoring system for mineral processing equipment based on IoT and industrial cloud computing [J]. Computer Integrated Manufacturing Systems, 2017, 23(9): 1972-1982.

[11] 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.
LI Chunyang, LIU Di, CUI Wei, et al. Unified application development platform based on micro-service architecture [J]. Computer Systems & Applications, 2017,26(4):43-48.

[12] 张杰,司维超,王丽娜,等.一种面向微服务的通用考核系统设计与应用 [J].计算机与数字工程, 2018,46(12):2463-2467.
ZHANG Jie, SI Weichao, WANG Lina, et al. Design and application of general assessment system for micro service [J]. Computer and Digital Engineering, 2018,46(12):2463-2467.

[13] 方意,朱永强,宫学庆.微服务架构下的分布式事务处理[J].计算机应用与软件, 2019,36(1):152-158.
FANG Yi, ZHU Yongqiang, GONG Xueqing. Distributed transaction processing under microservice architecture [J]. Computer Applications and Softvware, 2019,36(1):152-158.

[14] 曹宏宇,胡恒.基于微服务架构的智能终端软件架构探讨[J].科技创新与应用,2019(20):17-19.
CAO Hongyu,HU Heng. Discussion on the software architecture of intelligent terminal based on micro-service architecture[J]. Technology Innovation and Application, 2019(20):17-19.

[15] 钟俊林. 基于微服务架构的自助微商城的研究与实现[D].北京:北京邮电大学,2019.
ZHONG Junlin. Research and implementation of self-service micro-mall based on micro-service architecture [D]. Beijing: Beijing University of Posts and Telecommunications, 2019.