

引文格式: 欧阳森山, 雷浩, 陈琛, 等. 基于微服务的MES-MSA架构研究[J]. 航空制造技术, 2021, 64(20): 47-55.

OUYANG Senshan, LEI Hao, CHEN chen, et al. Research on MES-MSA architecture based on microservices[J]. Aeronautical Manufacturing Technology, 2021, 64(20): 47-55.

# 基于微服务的MES-MSA 架构研究

欧阳森山, 雷浩, 陈琛, 彭春

(航空工业成都飞机工业(集团)有限责任公司, 成都 610092)

**[摘要]** 随着智能制造的兴起, MES系统将成为制造业贯通物理空间与虚拟空间的使能器, 现有MES系统在架构设计中不足以满足爆发式、颠覆式的业务需求。提出了一种基于微服务的MES系统架构, 由交互层、决策层、数据层、感知层4个层次构成, 由事件中心连接各个层次, 由IT模型和支撑平台提供基础设施。阐述了MES-MSA架构的关键技术, 给出了该架构的一些应用场景, 有效解决了现有系统架构耦合度高、扩展性不强、集成困难等技术问题的同时, 满足了智能制造对MES系统业务上的新要求。

**关键词:** 制造执行系统(MES); 微服务架构(MSA); 系统架构; 智能制造; 容器化部署

**DOI:** 10.16080/j.issn1671-833x.2021.20.047



欧阳森山

高级工程师, 硕士, 研究方向为大  
数据技术和系统架构设计。

以德国“工业4.0”、美国工业互联网为代表的工业革命席卷全球, 智能制造成为中国制造的主攻方

向。中国制造业从手工作坊、大批量流水线到精益制造高速发展, 批量定制成为下一阶段制造业的关键特征。基于大数据的自适应加工、精益为导向的自主化装配、以人为本的智能人工增强成为新工业革命下制造过程的典型场景, 传统的加工和装配模式被颠覆, 操作人员的角色被彻底改变, 与之相适应的MES系统(Manufacturing execution system)已不能胜任新的模式。

MES系统在制造企业的产品供应链中起到了承上启下的作用, 是生产活动与管理活动信息沟通的桥梁, 同时也是贯通虚拟空间与物理空间的使能器。产品、资源、流程的模型在MES系统中关联, 进一步去引导真实世界; 人、机器、物料之间相互操作, 各种数据通过MES系统采集、

存储、转换、互通; 数据反馈至模型在MES系统中仿真迭代, 人类智慧与人工智能相结合共同实现对制造过程的持续优化。

## 下一代MES系统

德国人工智能研究中心提出“工业4.0”的三大特征: 智能产品、智能机床、增强的操作员。美国GE公司提出工业互联网的三大特征: 智能设备、先进分析、与人的连接。中国航空工业提出智能制造的四大特征: 动态感知、实时分析、自主决策、精准执行。

综合各方观点, 在新的制造模式下, 下一代的MES系统应具备以下特征。

(1) 模型驱动。

随着设计、工艺、管理与信息化

的深度融合,生产的各个要素均实现了在虚拟世界的建模并开始整合,它们开始相互关联,快速迭代,进一步影响物理世界。流水线生产催生了产业工人,精益生产壮大了设计师规模,而批量定制将以模型构建人员为中心。设计、工艺人员构建产品模型,生产管理人员构建计划模型,检验人员构建质量模型,仓储人员构建资源模型,经营人员构建流程模型等,这些模型被整合为一个覆盖企业全领域、产品全生命周期的超级模型,以智能化为主线,驱动人员、设备、资源高效配置,更加精益、柔性、敏捷。

### (2) 数字孪生。

数据采集的完整性和精度是决定模型是否具备智能的关键,制造业大数据就是在某种程度上能真实反映实际生产制造过程的数据集,“Digital twin”是智能制造相关新技术的基础也是核心<sup>[1]</sup>,其主要对人、机器、物料、环境的大数据采集。随着工控网的建立,对机器的数据采集日趋成熟,物联网的发展使得对物料的数据采集逐步完善,而对人的数据采集尚需突破,音频、视频等流式数据将成为重点。

### (3) 智能规划。

批量定制模式下,产品小批量、混线生产常态化,研制周期大幅度缩减,加上对插单、各类型变更的快速响应,需要作业计划根据数据复现的仿真模型即时做出调整应对各种变化,赋予算法智能以寻求最优解<sup>[2]</sup>。就像导航软件,即时根据路况调整路线,代替了人脑寻路的这部分智能。

### (4) 自主保障。

设备、工装、工具、物料能自主与MES系统交互,实时反馈状态信息,通过“物-物”直接交互,消除人工环节,实现定期保养和故障处理自主化。同时让智能物流成为可能,为排程规划提供可靠的基础数据。结

合多维度数据进行自主分析决策,提前预警自身故障缺陷,确保资源的“7d×24h”的高可靠保障。

### (5) 人机增强。

鼠标和键盘成为人机交互的备选工具,音视频或者传感器成为主要交互途径,不同工种不同角色将配备专用界面和智能设备,并提供具有初级智能的助手,为一线员工提供培训、答疑、信息检索等服务<sup>[3]</sup>。

### (6) AI 认知。

构建可自主优化的智能算法,结合资深业务专家的经验,利用不断扩充的大数据进行训练,通过持续、高效的迭代让产品制造过程信息模型快速进化,促进模型与真实世界融合,在虚拟世界中快速地仿真推演,赋予信息系统真正的人工智能。

## MES 系统架构设计思路

为了应对颠覆式的业务变革对信息系统提出的挑战,MES系统需要重新定位,成为智能制造建设方案的中心,其总体设计和技术架构也将发生巨大变化,具体体现为以下几个方面。

### (1) 灰度平台。

MES与企业资源管理系统(ERP)、工艺设计系统(CAPP)的界线更加模糊,它变成一个开放的平台,整合各相关应用系统的服务,与众多新技术集成,为生产制造一线提供统一的入口和IT能力。

MES作为一个软件产品的定义也变得模糊,MES将变成众多软件产品和IT能力的综合体,由不同专业的供应商共同建设和维护。

软件研发与实施应用的阶段划分不再清晰,各制造企业的用户不仅是使用者,更是设计者。平台为用户提供了80%的软件,剩下的20%由用户使用二次开发工具完成,对基础服务进行组合、配置,提供自定义的个性化功能,如设计流程、设计界面、设计报表等。

### (2) 多态感知。

平台以服务的方式为用户提供丰富的数据感知接口,如DNC、音视频识别、流式数据计算、传感器接口适配、GPS数据采集、外部数据接入等,并且提供大数据存储、计算、展示等能力。

平台实现应用与数据的解耦,数据作为虚拟世界感知真实世界的结果提供给用户分析利用,数据的质量和数据分析能力成为评价企业能力的关键指标,但数据的安全保密管理还存在技术瓶颈。

### (3) 流式进化。

就像手机APP一样,平台为不同的用户提供个性化的功能组合,每一个小的功能模块成为独立生命周期的微应用,通过DevOps(Development operations)统一管理,支撑平台、研发团队、运行维护灵活配置,横向扩展。MES平台将进入流式进化阶段,成为一个微观的互联网生态。

## 基于微服务的MES系统架构

### 1 现有MES系统架构存在的问题

(1)系统耦合度高,维护成本高,新业务交付周期长。

随着MES系统功能的不断增加,代码量也大幅度增加,开发人员越来越难以对全局有深入的了解,出现缺陷时分析、定位、修复的时间成本高。由于系统部署在同一进程内,每次新业务的推出都需要整体上线,不同功能之间存在干扰,为避免问题就要投入更多的回归测试,导致交付周期长。

(2)系统可扩展性差,优化困难。

单体应用架构业务逻辑运行在同一进程中,数据存放在同一个数据库中,水平扩展难。如MES系统中计划生成属于CPU密集型,而物料查询属于IO密集型,水平扩展需要服务器同时具备足够的CPU和内存来满足要求,且服务器资源不能按需分配。

(3) 难以满足复杂的系统内外部集成。

MES 系统必须快速与新技术、新供应商的业务进行集成对接,目前的集成往往通过企业服务总线(ESB)完成,ESB 实现了系统间解耦的目的,但是增加了交易链路长度,产生了新的瓶颈,降低了系统性能,同时集成逻辑与业务逻辑分离有可能降低业务交付效率。

## 2 微服务架构简介

微服务架构(Microservice architecture, MSA)<sup>[4]</sup>是一种架构模式,旨在通过将功能分解到各个离散的服务中以实现对解决方案的解耦。它将单体架构的系统划分成多组小的服务,每组服务运行在独立的进程

中,采用独立的数据库存储数据,可以独立地部署在生产环境中。每组服务围绕具体的业务进行构建,服务与服务之间采用轻量级的通讯模式,可以互相配合、互相协调。单体架构系统到微服务架构系统的演进方式如图 1 所示。

微服务架构本质是将应用系统在物理上、逻辑上同时拆分,优势在于:(1)单个微服务业务可以独立开发、部署,缩短了新业务上线时间;(2)单个微服务业务可以独立维护,降低单业务本身的维护难度;(3)独立分配资源,水平扩展能力增强。劣势在于:各服务之间的数据一致性差和服务联动的统一运维难度高两大问题,必须采用支撑微服务全生命

周期的平台来解决。

## 3 MES系统微服务架构(MES-MSA)设计

### 3.1 总体架构

MES-MSA 架构如图 2 所示,由交互层、决策层、数据层、感知层 4 个层次构成,由事件中心连接各个层次,由 IT 模型和支撑平台提供基础设施。

### 3.2 设计原则

MES-MSA 架构引入扩展立方体<sup>[5]</sup>的概念使应用系统具备良好的水平扩展能力,又避免出现数据不一致、可靠性低等问题。如图 3 所示,扩展立方体的起始点是一个单体应用,每一个轴线表示扩展的一个维度,每个维度的扩展不影响其他维度。

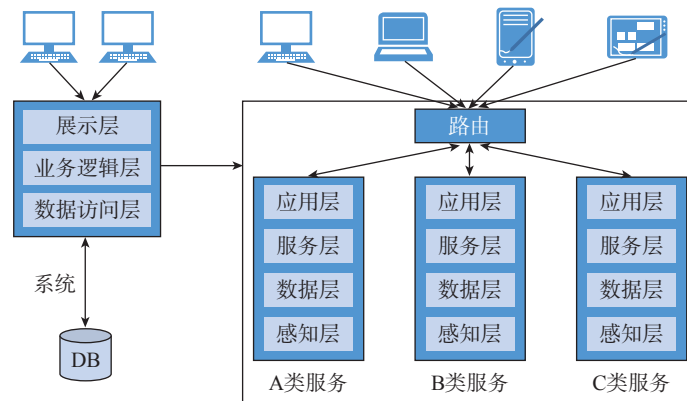


图 1 单体架构系统到微服务架构系统的演进方式

Fig.1 Evolution from single architecture system to microservice architecture system

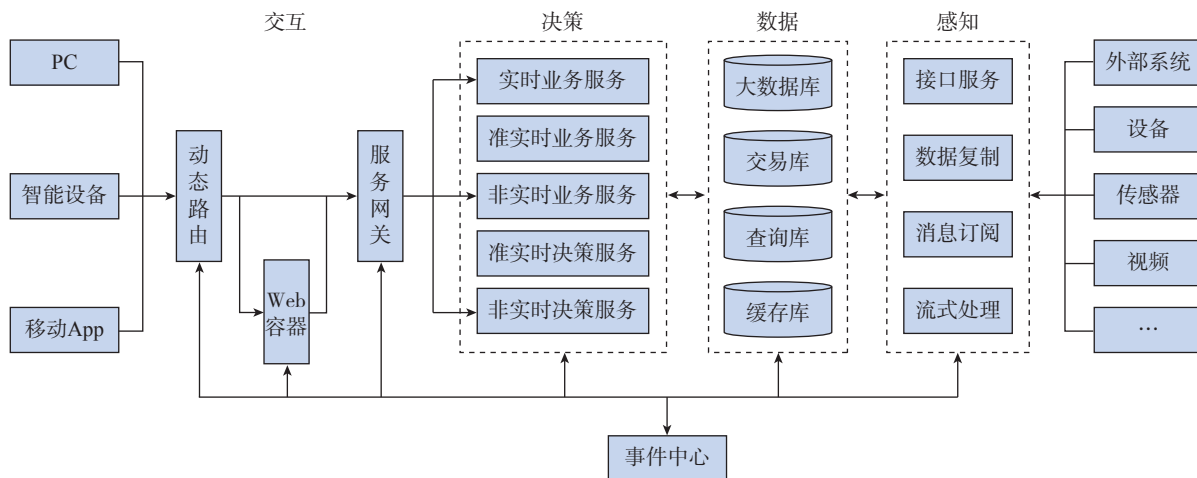


图 2 MES-MSA 架构

Fig.2 MES-MSA architecture

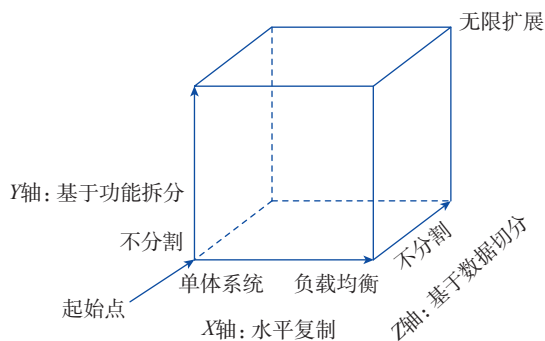


图3 扩展立方体  
Fig.3 Scale cube

X轴表示水平复制,即无差别地复制处理节点,例如进行并行的数值计算。X轴扩展方式通过负载均衡方式进行服务处理节点的定位。

Y轴表示按功能、职责拆分应用系统<sup>[6]</sup>,不同功能的业务逻辑运行在不同的进程中,具有独立的数据库,提高应用系统的水平扩展能力。例如即时规划属于非实时批量型,物料查询属于准实时交易型。Y轴的拆分方式通过服务路由进行服务处理节点的定位。

Z轴是按数据切分应用系统,在保证完整性的前提下根据业务相关性进行数据拆分。例如冷热数据分离<sup>[7]</sup>方式,频繁使用的数据和历史数据保存在不同数据库中。Z轴的切分方式通过数据路由进行服务处理节点的定位。

### 3.3 MES 四层架构设计

#### (1)交互层。

交互层是虚拟世界与物理世界的连接器,除了传统的鼠标、键盘、显示器以外,摄像头、扬声器、智能终端将成为主要的交互途径,不同工种不同角色将配备专用界面和智能设备。服务网关支持可扩展接口并提供集中式管控,包括访问控制、服务鉴权、服务路由、流量控制、访问日志、数据脱敏、内容缓存、服务编排等多种管控能力。

交互层主要采用X轴的水平复制模式进行扩展,通过对等复制、部

署多进程的方式提高处理能力。

#### (2)决策层。

决策层负责业务逻辑的实现,并通过人工智能辅助决策。根据技术特征,决策层服务主要分为实时业务服务、准实时业务服务、非实时业务服务、准实时决策服务、非实时决策服务5种类型,每种类型的服务部署在独立进程中,拥有独立的处理资源和数据库,实现进程与数据的隔离,防止服务间干扰。

决策层主要采用Y轴的功能划分进行扩展,按需分配硬件提高资源利用率。

#### (3)数据层。

数据层负责多态数据的存储和计算,实现数据与应用的解耦。采用冷热数据分离方式进行拆分,把历史大数据、读频繁数据、写频繁数据、流式数据独立存储,形成缓存、查询数据库、写数据库(交易数据)和大数据数据库四级的数据存储体系。

数据层主要采用Z轴的数据划分进行扩展,针对不同数据特点制定策略,兼顾了数据的可靠性和效率。

#### (4)感知层。

感知层负责连接、采集来自环境中的所有人、设备、传感器信息,实现在任意时段对任意信息的复刻,是虚拟世界描述物理世界的基本前提。感知层的数据采集包括接口服务、数据复制、消息订阅和流式处理4种方式。

### 3.4 IT 模型

IT模型是对MES-MSA系统的建模,通过对技术元信息的控制实现对复杂IT系统的管理,与之相对应的是业务模型(产品模型、资源模型、流程模型等)。IT模型是业务模型在软件系统实现时的载体,确保业务模型可以基于统一架构进行设计、开发、集成、交付、运行和维护。

IT模型描述的元信息包括静态与动态两部分,其中静态模型描述系统各层次中的组成部分、各部分之间的依赖与调用关系,如业务语义、通信协议、对外接口、处理流程、安全认证与授权方式。动态模型是系统运行时静态模型实例化后的描述,即静态模型产生的运行实例、实例之间的关系以及实例与物理资源之间的映射,如应用拓扑图、服务拓扑图、数据血缘关系图等。

### 3.5 支撑平台

采用MES-MSA架构后,MES系统从一个巨石型系统发展到分布式系统,支撑平台负责将需求、设计、开发、交付、运维的过程紧密协同与配合,以解决交付与运维复杂度呈指数级增加的问题。

支撑平台基于DevOps理念<sup>[8]</sup>,DevOps是一组过程、方法与系统的统称,用于促进应用开发、技术运营和质量保障各职能之间的沟通、协作与整合。它通过更好的优化开发(DEV)、测试(QA)、运维(OPS)的流程,使开发运维一体化,通过高度自动化的工具与流程使软件的构建、测试、发布更加快捷、频繁和可靠,可以说支撑平台是MES系统自身的生产线,其具备以下4点优势。

#### (1)互联网生态。

项目不再采用瀑布式的模式,而是被切分成多个子项目,各个子项目的成果都经过测试,具备可视、可集成及可运行的特征。MES-MSA架构让交互层、决策层、数据层、感知层的各种应用都可以独立发布、部署、运

行,使MES系统被分为多个相互联系但可以独立运行的APP。系统的升级优化无时无刻不在发生,而在此过程中系统一直处于可使用的状态。

#### (2) 即时集成。

高质量的软件项目要求每个IT研发人员每天至少集成一次,也就意味着每天会发生多次集成,因此支撑平台必须具备自动化构建能力(编译、发布、自动化测试)。即时集成有利于自动化检查缺陷,实时掌握软件健康状况,减少代码编译、数据库集成、测试、审查、部署及反馈中的重复劳动,支撑快速迭代、高可靠性要求的MES系统研发。

#### (3) 一键式部署与持续交付。

复杂IT系统研发过程中,往往存在多个环境,包括开发环境、测试环境、准生产环境、性能测试环境、生产环境等,研发人员需要将代码、配置、类库等部署到多个环境中,遇到问题需要回退到前一个状态,手工操作繁琐,且效率低下。

支撑平台提供自定义部署过程,

一键部署、一键供应、一键创建环境。一键式部署让持续交付成为可能,通过更频繁的自动化部署,让新上线的功能可以尽快地呈现在用户面前,并能在一定的时间里从用户处获得尽可能多的反馈。根据反馈更快速地对新业务功能进行调整,从而加快交付速度,适应裂变式的业务需求变化。

#### (4) 智能监控。

智能制造下的MES系统已经与生产过程密不可分、融为一体,IT系统需要为业务提供毫秒级不间断服务,IT系统的故障率成为制造缺陷率的关键制约因素。支撑平台必须提供秒级的修复能力,基于IT运维大数据结合人工智能即时产生问题处理策略或修复方案,实现弹性伸缩与故障自愈,实现MES系统的高可用性。

### 4 MES系统微服务架构(MES-MSA)实践

#### 4.1 系统微服务技术架构

如图4所示,整个微服务体系基

于云平台<sup>[9]</sup>,基础设施层提供IAAS(Infrastructure as a service),容器平台、基础服务、研发效能平台、中台层共同构成了PAAS(Platform as a service)层,其中基础服务的数据库与缓存模块,构成MES-MSA架构的数据层,其余PAAS层的模块组成一套完整优化开发、测试、运维的流程体系,实现支撑平台的功能。用户层与API网关功能构建系统的交互层。基于PAAS提供的服务进行开发,通过API网关层与基础服务层,完成感知层的构建,同时以上述功能为基础,基于研发效能平台完成决策层的应用开发,实现业务逻辑。

#### 4.2 系统部署模式

MES-MSA系统架构基于容器平台进行部署<sup>[10]</sup>,如图5所示,“容器实例”与“微服务治理中间件”已经全面“Docker化”;基于Kubernetes进行容器管理,基于Jenkins、Nexus、Harbor、SonarQube、Gitlab共同实现CI/CD功能,快速实现基于应用能力的横向扩展,基于服务能力的纵向升级。

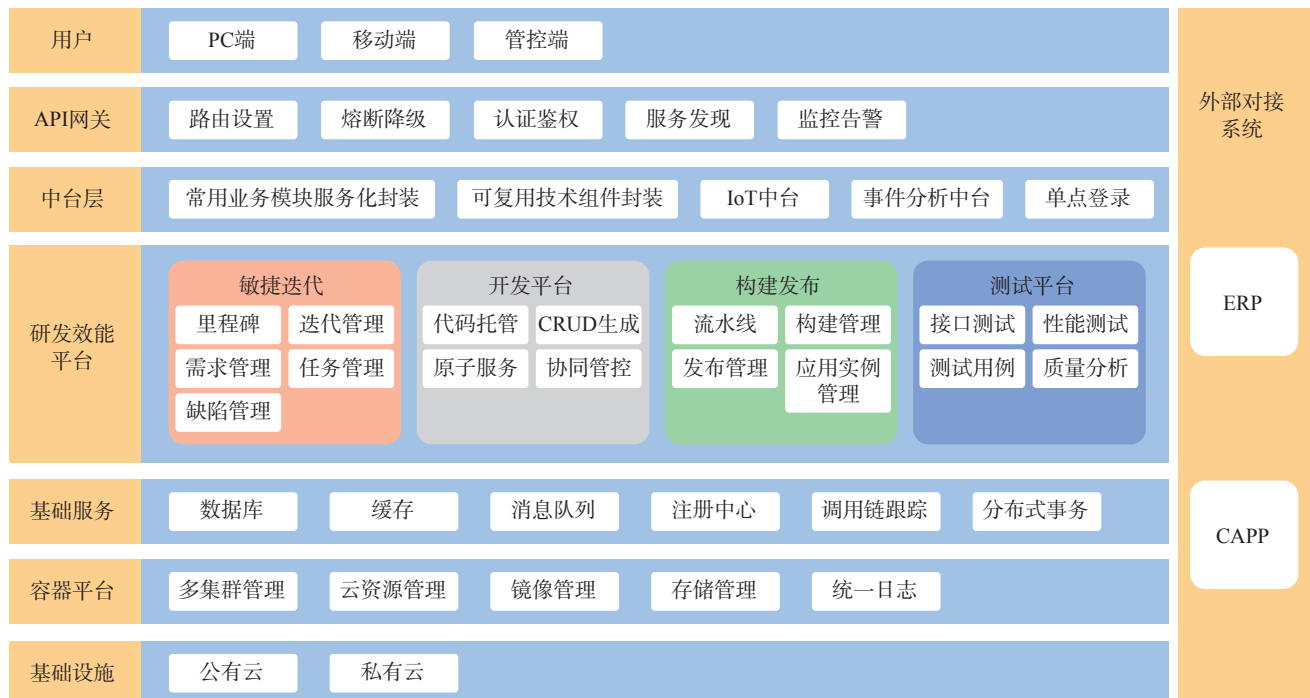


图4 系统微服务技术架构图

Fig.4 Technical architecture diagram of system microservices

4.3 系统应用拓扑

如图6所示, MES-MSA 架构系统中的各个模块以“Rest”风格对

外提供服务, 以前后端分离、应用与数据解耦的模式部署。每个模块对应一个数据库实例, 将数据进一步解

耦, 实现更细化的服务拆分。

4.4 系统模块简介

基于模块化、组件化的思想, 如

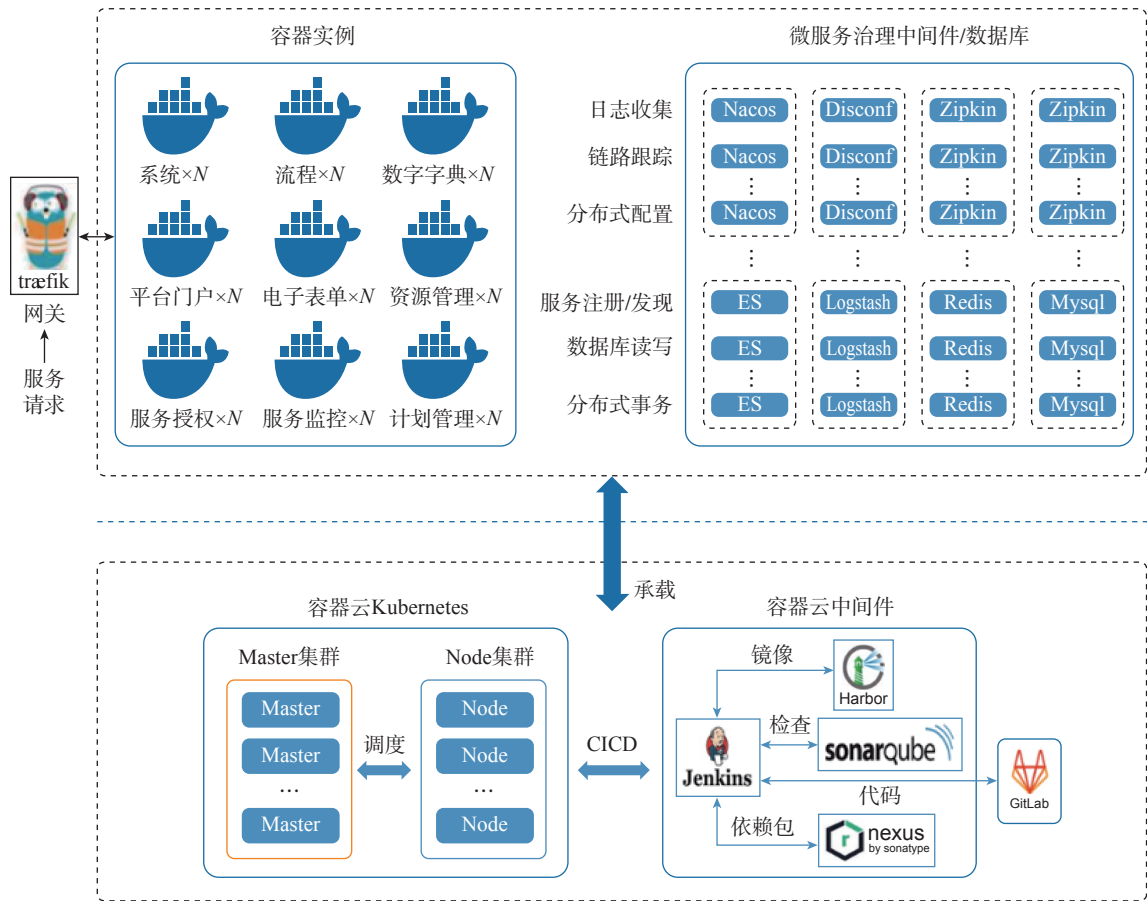


图5 系统部署模式图

Fig.5 System deployment mode diagram

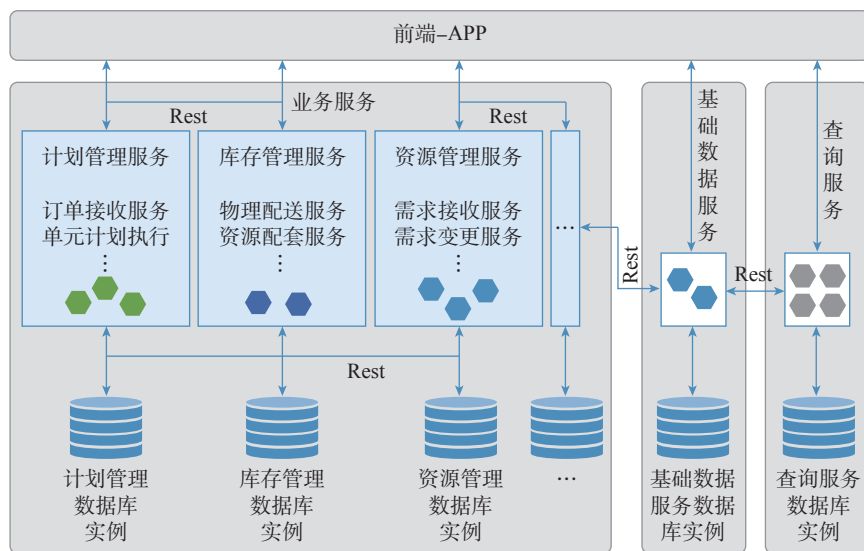


图6 系统应用拓扑图

Fig.6 System application topology diagram

图 7 所示,将整个系统划分为资源管理、用户行为、计划/库存管理、数据字典、平台问题协同解决、执行管理、文件服务管理 7 个模块,通过数据字典设计方法将各个模块下的组件串联起来,提高数据可用性,形成一个完整的系统。

#### 4.5 系统—执行管理微服务实现

如图 8 所示,执行管理服务,采用微服务架构的设计理念,将执行管理业务与整个的 MES 业务进行剥离,通过“Rest”风格的服务接口与

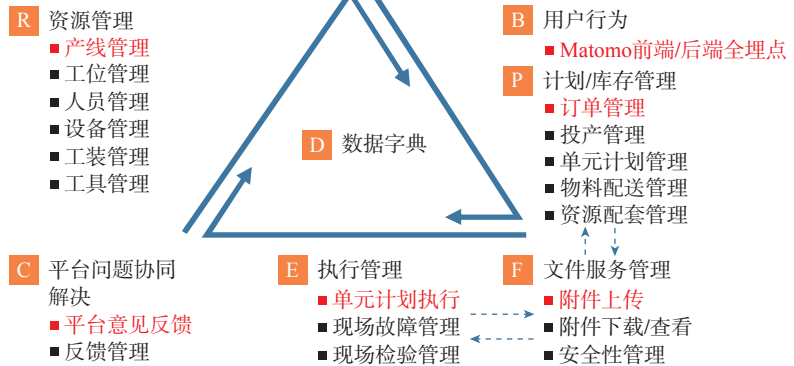


图 7 系统模块图

Fig.7 System module diagram

```

import { mapGetters } from 'vuex'
import VueScroll from 'vue-scroll/dist/vue-scroll-native'
import { addDrawer } from '@components/BasicType/CACComponents/AdvancedDrawer.vue'
import TableSearch from '@components/BasicType/CACComponents/AdvancedSearch.vue'
import AutoBadge from '@components/BasicType/AutoBadge'
import launch from './components/launch'
import mtrlDeliver from './components/mtrlDeliver'
import { getAllStepsById, getAllCellplansBySteps, getAllStepsByCellplan, getCellplanOprInfo, cellplanBegin, cellplanEnd } from '@api/execute'
import { addMeasureResult, delMeasureResultByIds, updateMeasureResult, getMeasureResult } from '@api/execute/measure'

export default {
  components: {
    // ...
  },
  data() {
    return {
      stepDesc: '描述加载中-',
      stepStatusColorArr: ['gold', 'blue', 'lime'],
      cellplanData: [],
      cellplanDataShow: [], // 用于过滤后展示
      sketchArr: [], // 草图信息数据
      standardOprInfo: [], // 标准作业规程信息
      cellplanInfoSpinning: false,
      oprInfoSpinning: false,
      selectedDeptPlanId: '',
      columns: [
        // ...
      ],
      columnsMeasured: [
        // ...
      ],
      selectedRowKeys: [],
      pagination: {
        // ...
      },
      formItemList: [
        // ...
      ],
      measuredArr: [
        // { measuredItem: '-', measuredValue: '-', standardValueLower: '-', standardValueUpper: '-' }, createTime: '-' }
        // ...
      ],
      measuredArr: []
    }
  }
}
    
```

(a) 前端

```

// 按条件不分页查询
// @param queryReqBean 查询条件
// @return ResponseMsg<TFOallInfoDTO>
//
@GetMapping("/search/v1")
@ApiOperation(value = "按条件不分页查询")
@ApiRestMethodAuth(desc = "按条件不分页查询")
public ResponseMsg<TFOallInfoDTO> search(@RequestParam(value = "查询条件", name = "queryReqBean") @RequestBody QueryReqBean<TFOallInfoDTO> queryReqBean) {
    ResponseMsg<TFOallInfoDTO> responseMsg = new ResponseMsg<>();
    TFOallInfoDTO result = new TFOallInfoDTO();
    String planId = queryReqBean.getSearchParams().getPlanId();

    QueryReqBean<TFOBasicDTO> tFOBasicDTOQueryReqBean = new QueryReqBean<>();
    TFOBasicDTO tFOBasicDTO = new TFOBasicDTO();
    tFOBasicDTO.setPlanId(planId);
    tFOBasicDTO.setUserReqBean.setSearchParams(tFOBasicDTO);
    result.setTFOBasicDTOS(tFOBasicService.searchTFOBasic(tFOBasicDTOQueryReqBean));

    QueryReqBean<TFOpInfoDTO> tFOpInfoDTOQueryReqBean = new QueryReqBean<>();
    TFOpInfoDTO tFOpInfoDTO = new TFOpInfoDTO();
    tFOpInfoDTO.setPlanId(planId);
    tFOpInfoDTOQueryReqBean.setSearchParams(tFOpInfoDTO);
    result.setTFOpInfoDTOS(tFOpInfoService.searchTFOpInfo(tFOpInfoDTOQueryReqBean));

    // 原旧
    QueryReqBean<TFODrawingDTO> tFODrawingDTOQueryReqBean = new QueryReqBean<>();
    TFODrawingDTO tFODrawingDTO = new TFODrawingDTO();
    tFODrawingDTO.setPlanId(planId);
    tFODrawingDTOQueryReqBean.setSearchParams(tFODrawingDTO);
    result.setTFODrawingDTOS(tFODrawingService.searchTFODrawing(tFODrawingDTOQueryReqBean));

    // 新增NEW
    QueryReqBean<TFODrawingFileDTO> tFODrawingFileDTOQueryReqBean = new QueryReqBean<>();
}
    
```

(b) 后端

图 8 前后端开发

Fig.8 Development of front-end and back-end

计划、资源、库存、数据字典等服务进行数据交互。

执行管理服务实现采用前后端分离的开发模式。如图 8 (a) 所示, 前端基于 Vue-element-admin, 实现执行管理服务前端页面的快速开发。后端采用 Spring cloud 技术体系, 如图 8 (b) 所示, 将计划执行所需的制造大纲工序、草图等信息的查询, 单元计划查询, 单元计划执行开工、完工、实测记录等业务操作, 通过服务的方式注册到网关中, 并统一开放给前端提供服务。

执行服务基于容器平台进行部署, 实现应用的自动构建、发布及上线。

#### 4.6 系统功能页面展示

如图 9 所示, 系统以每个执行单元为最小单位进行管理, 通过各类微服务的组合, 形成部件、系统组件两大专业化 MES 应用 APP。各个模块分为:

(1) 单元计划管理。单元计划管理分为订单管理、投产管理、计划

管理。

(2) 单元资源管理。单元资源管理模块由产线配置、工位管理、人员管理、设备管理、工装管理、工具管理组成。

(3) 单元库存管理。单元库存管理模块由物料主数据、物料配送、资源配套功能组成。

(4) 单元执行管理。单元执行管理模块由现场执行、故障处理、制造大纲功能组成。

### 结论

在智能制造趋势下, 现有 MES 系统在技术和业务上难以满足企业需求, 下一代 MES 系统需要从各方面适应新变化。本文通过分析现有 MES 系统架构的问题, 提出了基于微服务的 MES 系统架构 (MES-MSA), 并给出了架构的设计原则和功能概述, 在有效解决了现有系统架构耦合度高、扩展性不强、集成困

难等技术问题的同时, 满足了智能制造对 MES 系统业务上的新要求。该体系架构的实现方法和实施模式将在后续的实践中得到进一步完善和验证。

### 参考文献

- [1] 陶飞, 张贺, 戚庆林, 等. 数字孪生模型构建理论及应用[J]. 计算机集成制造系统, 2021, 27(1): 1-15.
- TAO Fei, ZHANG He, QI Qinglin, et al. Digital twin model construction theory and application[J]. Computer Integrated Manufacturing Systems, 2021, 27(1): 1-15.
- [2] ZWOLINSKA B, TUBIS A A, CHAMIER N, et al. Personalization of the MES System to the needs of highly variable production[J]. Sensors, 2020, 20(22): 6484-6485.
- [3] BLAGA A, MILITARU C, MEZEI A D, et al. Augmented reality integration into MES for connected workers[J]. Robotics and Computer Integrated Manufacturing, 2021, 68: 102057.
- [4] MCLARTY M, AMUNDSEN M.

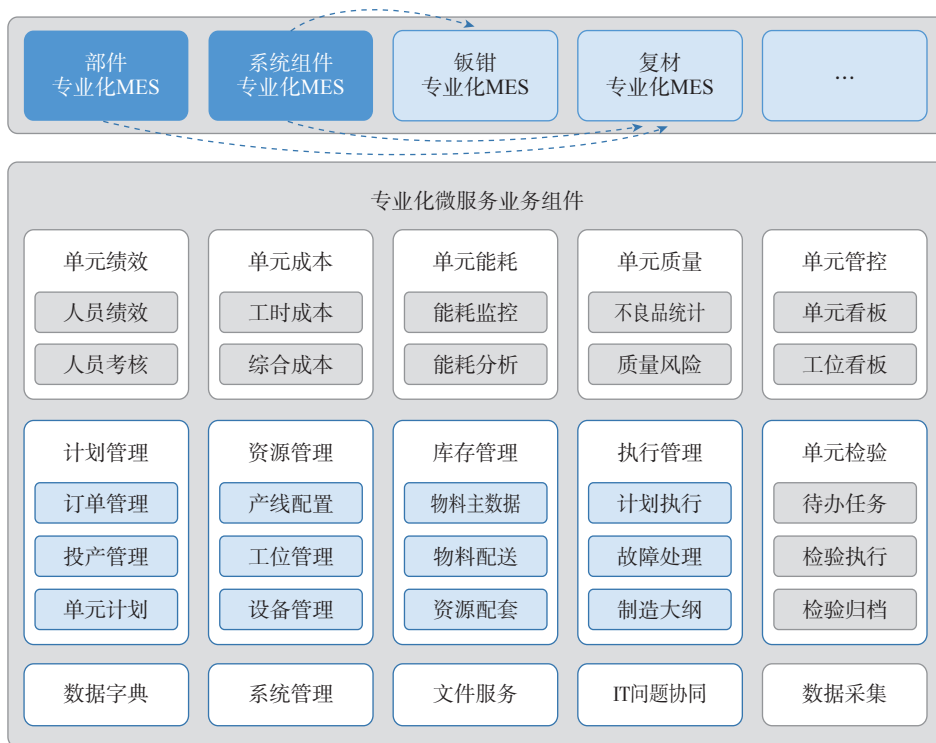


图 9 系统应用架构图

Fig.9 System application architecture diagram

- Aligning principles, practices, and culture[M]. New York: The Haworth Press, 2016.
- [5] 马丁·阿伯特. 架构即未来: 现代企业可扩展的 Web 架构、流程和组织 [M]. 陈斌, 译. 北京: 机械工业出版社, 2016.
- ABBOTT M. Architecture is the future: Scalable web architectures, processes, and organizations for modern enterprises[M]. CHEN Bin, trans. Beijing: China Machine Press, 2016.
- [6] 丁丹, 彭鑫, 郭晓峰, 等. 场景驱动且自底向上的单体系统微服务拆分方法 [J]. 软件学报, 2020(11): 3461–3480.
- DING Dan, PENG Xin, GUO Xiaofeng, et al. Scenario-driven and bottom-up single system microservice split method[J]. Journal of Software, 2020(11): 3461–3480.
- [7] 魏学才, 官庆媛, 沈佳杰, 等. 适应冷热数据存储的多编码架构的设计与实证 [J]. 计算机应用与软件, 2017(2): 35–41.
- WEI Xuecai, GONG Qingyuan, SHEN Jiajie, et al. Design and demonstration of multi coding architecture for hot and cold data storage [J]. Computer Applications and Software, 2017(2): 35–41.
- [8] BASS L, WEBER I, ZHU L M, et al. DevOps: A software architect's perspective[M]. New York: Addison-Wesley Professional, 2015.
- [9] 方舟, 郝昕, 李锐, 等. 基于微服务架构的云制造执行系统 [J]. 科学技术创新, 2020(8): 69–70.
- FANG Zhou, HAO Xin, LI Rui, et al. Cloud manufacturing execution system based on microservice architecture[J]. Scientific and Technological Innovation, 2020(8): 69–70.
- [10] 相铮, 石春鹏, 韩立新. Docker 容器技术在 MES 系统部署中的应用 [J]. 制造业自动化, 2020, 42(4): 116–119.
- XIANG Zheng, SHI Chunpeng, HAN Lixin. The application of Docker container technology in MES system deployment[J]. Manufacturing Automation, 2020, 42(4): 116–119.

## Research on MES–MSA Architecture Based on Microservices

OUYANG Senshan, LEI Hao, CHEN Chen, PENG Chun

(AVIC Chengdu Aircraft Industrial (Group) Co., Ltd., Chendu 610092, China)

**[ABSTRACT]** With the rise of intelligent manufacturing, the MES system will become an enabler for the manufacturing industry to integrate physical space and virtual space. The existing MES system is insufficient in its architecture design to meet explosive and disruptive business needs. This paper proposes a MES system architecture based on microservice, which is composed of interaction layer, decision layer, data layer and perception layer. Each layer is connected by event center, and infrastructure is provided by IT model and supporting platform. This paper expounds the key technologies of MES–MSA architecture, and gives some application scenarios of this architecture. It effectively solves the technical problems of the existing system architecture, such as high coupling degree, weak expansibility and difficulty in integration, and meets the new requirements of intelligent manufacturing for MES system business.

**Keywords:** Manufacturing execution system (MES); Microservice architecture (MSA); System architecture;

Intelligent manufacturing; Containerized deployment

(责编 七七)