
杭州电子科技大学

硕士学位论文

题目: 基于云平台的生产管理系统设计及关键技术研究

研究生 王奕怡

专业 电子与通信工程

指导教师 陈科明 副教授

完成日期 2021 年 3 月

杭州电子科技大学硕士学位论文

基于云平台的生产管理系统设计及关键技术
研究

研 究 生： 王奕怡

指导教师： 陈科明 副教授

2021 年 3 月

**Dissertation Submitted to Hangzhou Dianzi University
for the Degree of Master**

**Production management system design
and key technology research based on
cloud platform**

Candidate: Wang YiYi

Supervisor: Associate Prof. Chen Keming

March, 2021

杭州电子科技大学 学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

论文作者签名：王奕怡

日期：2021 年 3 月 12 日

学位论文使用授权说明

本人完全了解杭州电子科技大学关于保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属杭州电子科技大学。本人保证毕业离校后，发表论文或使用论文工作成果时署名单位仍然为杭州电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密论文在解密后遵守此规定）

论文作者签名：王奕怡

日期：2021 年 3 月 12 日

指导教师签名：王奕怡

日期：2021 年 3 月 12 日

摘要

传统生产过程执行管理系统（Manufacturing Execution System，简称 MES 系统）需要独立服务器进行存储和处理数据，并且容易出现“数据孤岛”，基于云平台的 MES 系统以其服务便捷、成本低等特点，越来越受到市场的重视。但是目前云 MES 系统仍然存在一些问题需要优化：车间和工单等情况多变复杂，任务计算调度后的分配未能发挥出车间最大的工作效率；数据传输速度与安全性未满足某些企业的需求。

本文设计了基于云平台的生产管理系统，采用云原生技术、微服务架构打通工单排序调度、车间生产执行、生产数据采集、产品质量管控、设备维保管理、产品追溯管理等功能模块之间的数据流动，实现业务在线化、制作过程数字化，以此大幅度的提高车间生产率和产品质量，减少生产成本和生产周期，加强对于工厂生产线的控制，并将这套系统应用于表盘制造工厂。

对工单进行高效地排序调度是云 mes 系统的重要功能模块，现在普遍使用的粒子群优化算法（Particle swarm optimization，简称 PSO 算法）在使用过程中其性能过于依赖初始化粒子群的质量，同时由于收敛过快，粒子群可能难以逃逸局部，导致无法寻找到全局最优解。本文针对这些问题，通过初始化粒子群时进行筛选，使参数在迭代过程中动态调整以及混沌扰乱抑制早熟这三种方式，对其进行优化。经过多次测试，该算法迭代次数较之前降低了 34%，最终求得的最优解适应度值几乎无波动，且比原先的 PSO 算法更接近真正的最优解，大幅度稳定地提高了制造业车间的生产效率。

为了保证数据的安全性及传输效率，需要对数据进行加密传输、提高传输速率，目前使用的 IPSec VPN 算法在长时间的加密传输中，占用计算资源压力过大，无法满足工厂长期稳定的设备数据传输。本文通过国密算法对其进行优化，并加以硬件加密卡的辅助解放 CPU 压力。在协商层设计中通过加密、签名以及数字证书多种方式结合提高可靠度。同时通过硬件加密卡向服务器内核的 Linux 算法库添加了国密算法，便于数据包在传输过程中基于国密标准进行处理。此次优化提高了在传输过程中双方验证身份的权威性以及数据加密的安全性，并在数据传输速度与加密效率方面都有了很高的提升。经测试，该传输方式成功由国密算法加密，通过硬件加密卡的加入以及加密算法的更换使得传输速率较以往的 IPSec 快了 79%-118.6%，满足企业需求，且 48 小时连续传输不宕机，可应用于实地。

关键词：MES 系统，云平台，资源调度，PSO 算法，IPSec VPN

ABSTRACT

Traditional manufacturing execution system (MES) requires independent servers to store and process data, which is easy to generate "data islands". The MES system based on cloud platform has attracted more and more attention of the market due to its features of convenient service and low cost. However, there are still some problems in cloud MES system that need to be optimized: workshop situation and job order are changeable and complex, and the assignment after task calculation and scheduling fails to give full play to the maximization of workshop work efficiency; Data transmission speed and security cannot meet the needs of some enterprises.

This paper designed the production management system based on cloud platform, use native cloud technology, micro service architecture through the repair order scheduling, production workshop, production data acquisition, quality control, equipment maintenance and management, product traceability management functions such as data flow between modules, realize the production process of online digital business, and greatly improve plant productivity and product quality, reduce production cost and production cycle, to strengthen the control of factory production lines, and apply this system to dial out to the factory.

Factory order is an important function of effective scheduling module, cloud mes system is now widely used particle swarm optimization algorithm, particle swarm optimization, PSO algorithm) in the process of using performance rely too much on the quality of the initialization of particle swarm, and at the same time because of the rapid convergence, particle swarm is hard to escape from local, lead to can not find the global optimal solution. In order to solve these problems, three methods are proposed to optimize the dynamic adjustment of parameters during iteration and to suppress precocity by chaos by screening the initial particle swarm. After many tests, the number of iterations of the algorithm was reduced by 34% compared to the previous algorithm. Finally, the fitness value of the optimal solution has almost no fluctuation, which is closer to the real optimal solution than the original PSO algorithm, and greatly improves the production efficiency of the manufacturing workshop.

In order to ensure the security of the data, it is necessary to encrypt and transmit the data. In the long encrypted transmission, the IPSec VPN algorithm currently used occupies too much computing resources to meet the long-term and stable data transmission of factory equipment. In this paper, the state secret algorithm is used to optimize it, and hardware encryption card is used to relieve the CPU pressure. In the design of negotiation layer, a combination of encryption, signature

and digital certificate is adopted to improve the reliability. At the same time, the state secret algorithm is added to the Linux algorithm library of the server kernel through hardware encryption card, which is convenient for packet processing based on the state secret standard in the transmission process. This optimization improves the authentication authority of both parties and the security of data encryption in the transmission process, and greatly improves the data transmission speed and encryption efficiency. Through the test, the transmission method is successfully encrypted by the state secret algorithm. By adding hardware encryption card and changing the encryption algorithm, the transmission rate is 79%-118.6% faster than the previous IPSec, which meets the needs of enterprises. The 48-hour continuous transmission does not drop, and can be applied in the field.

Keywords: MES system, cloud platform, resource scheduling, PSO algorithm, IPSec VPN

摘要.....	I
ABSTRACT.....	II
第一章 绪论.....	1
1.1. 研究背景及意义.....	1
1.2. 国内外研究现状.....	4
1.3. 未来发展趋势.....	5
1.4. 论文结构安排.....	6
第二章 基于云 MES 的仪表生产管理系统设计.....	8
2.1. 系统设计.....	8
2.1.1. 需求分析.....	8
2.1.2. 整体框架.....	9
2.2. 云平台设计.....	10
2.2.1. 设备云设计.....	10
2.2.2. 业务云设计.....	11
2.3. 客户端设计.....	11
2.3.1. web 管理端.....	12
2.3.2. PDA 端.....	24
2.3.3. 小程序端.....	27
2.4. 传感器设计.....	29
2.5. 网关设计.....	34
2.6. 本章小结.....	36
第三章 基于早熟抑制的 PSO 算法任务调度设计.....	37
3.1. PSO 算法优化设计.....	38
3.1.1. 标准 PSO 算法概述.....	38
3.1.2. 标准 PSO 算法优缺点.....	39
3.1.3. 基于早熟抑制的 PSO 算法优化设计.....	39
3.1.4. PSO 算法优化流程.....	43
3.2. 测试分析.....	44
3.2.1. 仿真测试环境准备.....	45
3.2.2. 不同算法处理时间对比.....	45

3.2.3. PSO 算法优化性能对比.....	45
3.2.4. 任务分配调度实例测试.....	47
3.2.5. 测试分析.....	49
3.3. 本章小结.....	49
第四章 基于国密标准的 IPSEC VPN 加密优化设计.....	51
4.1. IPSEC 概述及缺陷.....	52
4.2. 基于国密标准的 IPSEC VPN 优化设计.....	53
4.2.1. 基于国密标准的 IKE 协商层设计.....	53
4.2.2. 基于国密算法的服务器内核设计.....	56
4.3. 测试分析.....	59
4.3.1. 传输速度对比.....	59
4.3.2. 解密速度对比.....	60
4.3.3. 测试分析.....	61
4.4. 本章小结.....	62
第五章 总结与展望.....	63
5.1. 本文总结工作.....	63
5.2. 研究展望.....	64
致谢.....	65
参考文献.....	66
附录.....	69

第一章 绪论

1.1. 研究背景及意义

当今世界随着科技的发展，人类的生活、生产过程已经离不开物联网的参与。近日，十九届五中全会召开，对于物联网做出了高度重视。会中指出数字化发展是现代产业体系建设的重要推动力。这个指示表明了当前社会物联网的发展已经不可抵挡^[1]。工业互联网的蓬勃发展，各种企业管理系统对企业数字化管理改革的落地使用率也越来越高，例如 ERP 企业资源配置管理、SCM 供应链管理、PLM 产品生命周期管理以及 MES 生产执行管理系统^[2]，不同系统相辅相成，其架构图如下所示。其中，MES 系统作为一套制造业生产过程系统化监控管理的信息系统^[3]，帮助了许多传统制造工厂实现数字化系统管理，适应当今社会的发展步伐。

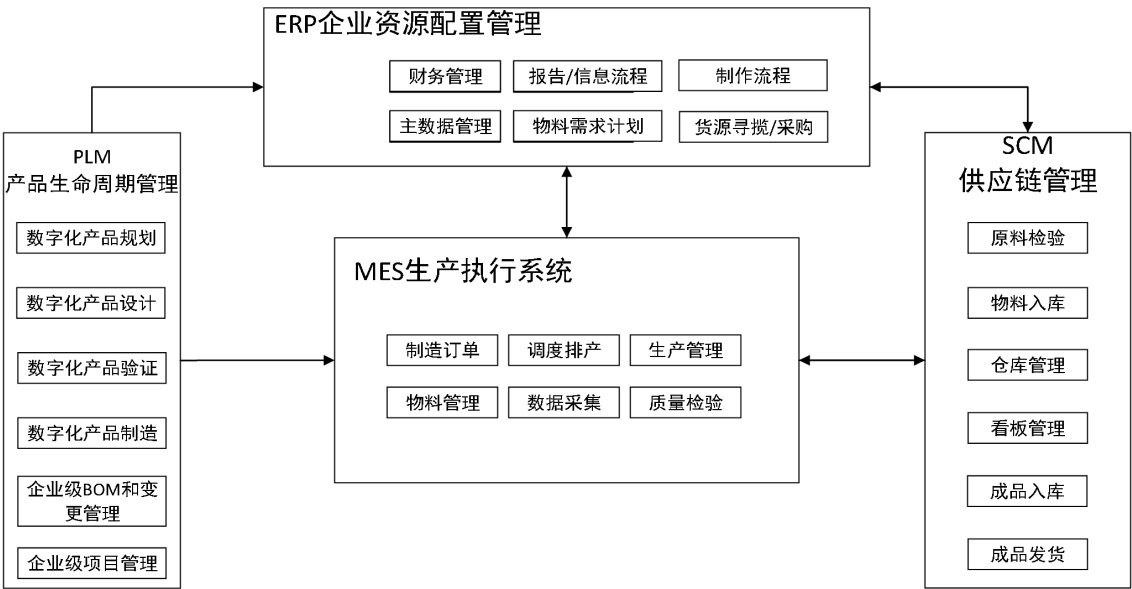


图 1.1 工业流程中管理系统关系图

MES 系统直击企业生产过程不透明、资源浪费难避免、生产异常难追溯、绩效分析靠人工等痛点，大幅度降低生产周期，减少质量损失，大力加强了对生产过程的控制^[4]。然而在实际应用时仍然存在着一些问题：

（1）本地 MES 系统无法在车间与企业的其余环节或者上下游企业间进行数据的互通共享，这无形中使之变成了一个封闭、孤立的“数据孤岛”，难以发挥数据灵活的特性^[5]。

（2）在我国，中小企业的数量占总体的绝大多数，且它们为我国贡献了一半多的 GDP 发展^[6]。然而，MES 系统在部署过程中，服务器高昂的成本使得众多中小企业形成了不小的压力，使之望尘莫及。

(3) 针对每个企业部署 MES 系统时根据其特性进行个性化开发, 导致开发成本以及后续的运维成本居高不下, 但其实许多产业链所需要的主要功能大多大同小异。

云 MES 系统是 MES 系统根据当今世界科技发展和云技术成熟的发展产物, 满足了以上企业转型所升级的需求。它由云端代替了独立服务器来进行储存和计算数据, 并通过 DevOps 流程管理、轻量级容器以及微服务架构实现云原生技术。

云原生技术较于传统应用更易于扩展, 具有较大的弹性与灵活性, 并可由 K8s 进行调度管理, 达到快速开发、持续交付项目的目的。作为一系列云计算技术体系和企业管理方法的集合, 它可以在开发过程中呈现 DevOps 模型, 将开发与运维工作紧密贴合, 缩短开发周期, 增加部署更新频率。并根据 K8s 进行调度管理, 来在运行时处理大量容器计算、存储, 编排负责容器集群的调度、服务发现和资源管理。

同时云原生技术通过微服务架构, 将一个应用拆分为多个独立的小应用, 每个小应用单独负责一个功能, 这样可以大幅度减少系统的复杂性, 便于单独对某个应用进行部署更新, 升级扩容。其中, 微服务架构分为侵入式架构和非侵入式架构。侵入式架构, 例如 Spring Cloud, 简化了分布式系统基础设施的开发, 将各公司成熟的服务框架经由 Spring Boot 风格封装, 实现易上手、易部署和易维护的分布式系统开发工具包。而非侵入式架构, 例如 Service Mesh 通过处理服务间请求/相应的可靠传递, 来实现服务治理、遗留系统的零侵入接入以及异构框架开发的微服务。云原生技术的基础架构关系图如下所示。

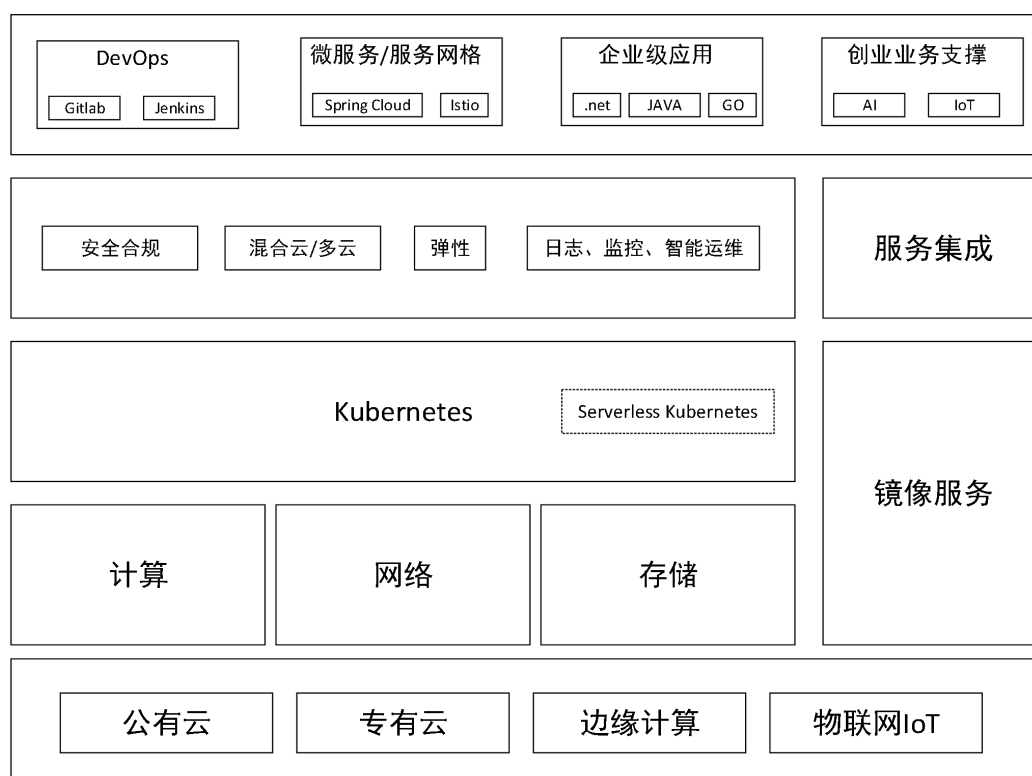


图 1.1 云原生技术基础架构，交付和应用之间的关系图

云原生技术更贴合基于 MES 系统的企业业务, 与云平台结合, 充分发挥云平台的扩展、

硬件解耦能力，具有更大的弹性、灵活性和跨云环境可移植性。云 MES 系统基于多端多角色协同，全方位地对生产过程实行去纸数字化，云 MES 的具体架构如图 1.1 所示。并主要通过以下三方面对车间进行优化：

（1）数据分析。通过不同数据聚合互通、图表多方式可视化呈现、多维度数据全面解读，从整体层面系统分析生产、设备、车间、人员的情况，挖掘出每组数据的内层含义以及数据间的深度联系，使管理层做出更有利的决策^[7]。

（2）生产管理。通过云端实时刷新数据、生产异常及时报警、远程监控产品质量，让管理层掌握最新生产进度和情况，及时恢复生产正常并快速追溯质量源头。

（3）计算预测。云平台计算出工单最优排期方案、物料最佳库存管理、预测设备最佳维护时间，以达到车间最大工作效率、仓库最少资源浪费和产线停线时间减少。

由此，工厂可以很大程度上减少用工成本、降低数据汇总和分析时间、减少生产周期、减少库存堆积、减少无效沟通时间、降低产品不良率以及提高人员生产率^[8]。

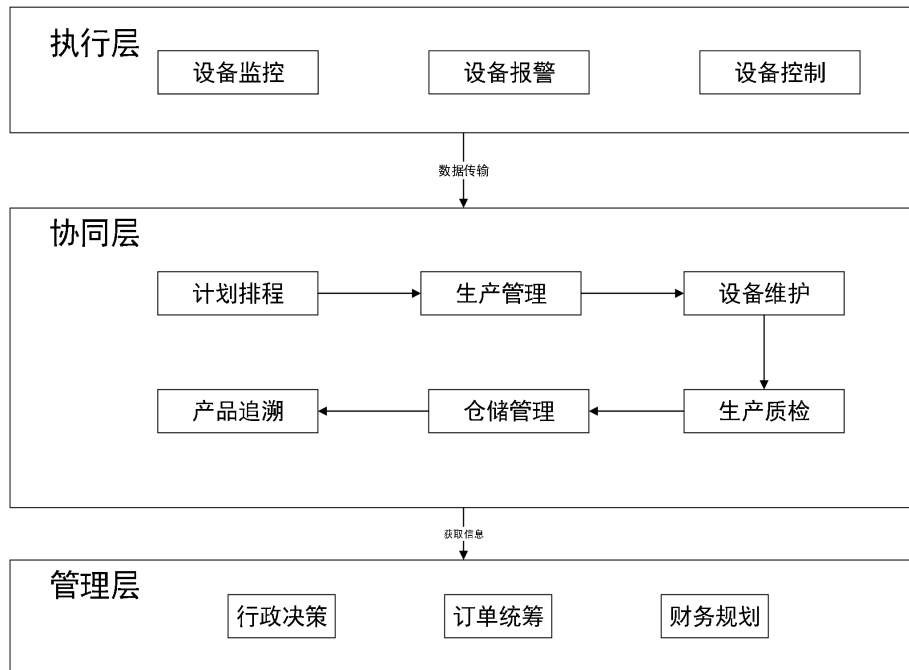


图 1.2 云 MES 系统架构图

云 MES 由于其价格便宜，性价比优惠的优点，推动了传统制造厂商中的下沉速度。不仅仅只给大型工厂提供转型优化服务，更为中小企业降低了使用云 MES 的门槛。我国作为人口大国、制造大国，大多制造厂商规模并不足以支撑本地 MES 高昂的开发成本和时间，云 MES 为这些制造厂商的数字化转型普及起到了很大的推动作用。

本文以仪表盘制造厂商为例，在当今世界，用户对于仪表的品种需求已经越来越多样，面板规格、电喷规格以及其余参数的个性化定制使得产品规格越发繁多，不同的产品使得原本困难的生产流水线管理更加复杂，且当前纸质化管理具有延时性、不透明性和难以追溯性

等缺陷。针对这种情况，本文通过云 MES 系统来优化对生产过程管理的数字化改革，基于云原生技术使工厂能够实现智能优化生产设备人员管理、实时掌握生产进度、快速追溯质量原因、产品质量严格把控、设备及时维护以及物料信息有效反馈等目的^[9]。

1.2. 国内外研究现状

当今社会自第三次科技革命以来，随着科学互联网技术的迅猛发展，目前 MES 系统大范围地应用于国外制造业之中。例如德国的 ADICOM MES 产品已经拥有了 500 多名企业客户，其软件使用安装处高达 2000 多个^[10]；温哥华的 Gemcom 公司致力于采矿产业的系统规划管理，其产品 Surpaca 已经被 90 多个国家所采用；美国的 Rockwell 对市面上工业自动化的知名品牌进行整合，在工业信息化领域处于领先地位，其分支机构分布在了 80 多个国家。但是国外的一些生产线由于其部件来自于不同的国家，难以做到整个产业区域的数据互通交流，且各自系统的相容性也较低^[11]。

表 1.1 国内外龙头 MES 企业对比

国外		国内	
公司	总市值（亿元）	公司	总市值（亿元）
Honeywell	8044.95	宝信软件	653.89
Siemens	6435.00	柏楚电子	232.31
GE	3864.99	鼎捷软件	84.25
ABB	3732.26	能科股份	56.01

由上表可得，目前 MES 服务在国内已经初见端倪，但是与发展较久的国外相比还有很大的差距。然而我国身为制造业大国，其中的数字化转型市场潜力不可言喻。同时今年的新冠疫情给我国制造业带来了极大影响，正所谓不破不立，很多制造商厂家也感受到了极大的挑战，更加清晰的了解到传统制造业转型的重要性。与此同时国内云原生技术的萌芽发展正回应了数字化管理转型的迫切需求。

目前已经有了不少的公司针对传统制造工厂数字化转型提出云 MES 方案。例如黑湖智造作为我国一套云端制造协同系统，帮助企业减少生产周期和库存堆积，降低用工成本，提高员工工作质量；极维度云 MES 通过工厂数字化管理，减少车间与管理层无效沟通，加强生产过程的掌控力度，提高车间工作效率。然而云 MES 运营公司质量良莠不齐，也存在很多公司在对制造厂商进行数字化改革部署时，并达到没有很好的规范与后续的维护，导致企业在应用过程中出现各种后遗问题。造成这个现象有以下几个原因。

(1) 国内对于传统制造业云 MES 的开发和应用环境并不成熟，例如其云原生技术中微服务产品开发能力不足，延缓了微服务产品开发进度，最终影响工业企业规模用云的效益；云原生技术推广程度不足，并未大范围的投入生产环境中，其社区经验并不丰富，不利于在其基础上的相应开发。

(2) 当系统某一功能产生故障时,在优化过程中将会对整体服务产生影响,其维护优化成本高,无法及时响应企业随时变换的需求以及发生故障的场景。

(3) 云 MES 系统在不同行业项目中,其需求也根据其特性各有不同。如何对各行业制定相应的标准,是一个需要经由该领域专家多次推敲制作的。同时云原生技术的标准建设不成体系,服务水平参差不齐。对以上标准盲目统一化,反而会使得一些制造厂商在数字化系统管理的过程中水土不服,拖了系统发展的后腿。

(4) 云 MES 系统的安全防护仍然存在安全隐患,云 MES 中所经由的数据皆为该厂商的商业机密,由于涉及到了财物,更应该以严谨的态度来对待。

我国的云 MES 技术较于国外还是一个新兴技术,我们应该理性的了解到我国与国外技术的差距,并且针对差距与问题寻求适合我国国情和当今制造业现状发展的方案,而不是盲目跟从照搬国外的生产模式,助力企业数字化转型。

1.3. 未来发展趋势

未来云 MES 系统必定随着 5G、大数据、工业互联网的飞速发展,逐步在制造业数字化转型的场景中进行普及。云原生技术在当前已经显示出广阔的前景,它可以为平台提供标准化基础架构,充分满足业务的弹性需求,实现跨平台资源横向打通,为用户提供一致的交付体验。云原生技术的快速发展和普及大力推动了当前传统制造厂商数字化的步伐。对于作为工业制造业数字化转型的有力途径,云 MES 不单是帮助企业管理层监控并管理车间产品生产进度,还将联合设备管理、员工管理等形成一套完整的系统管理。云 MES 在未来会向以下几个趋势进行发展。

(一) 云 MES 系统应用场景广阔,它普及的首要原因便是价格亲民、性价比高。为了减少云 MES 开发成本,云 MES 将呈现更加标准化、模块化的趋势^[12],并有中国信息通信研究院来推动云原生技术的标准化与规范化,以此减少开发时间和后期的运维难度。同时企业也可以按照自己的需求个性化进行添加附加功能。除此之外,再根据云原生平台的微产品服务,由企业根据自身情况自己把控个性化与开发成本的平衡,使每个不同规模的制造商企业都做到与云 MES 系统的适配。

(二) 云 MES 系统未来将更趋向以用户为中心,更快捷弹性地服务实地项目,通过云原生技术实现快速迭代、持续交付,以满足项目开发周期短,需求变化快的现状。丰富云原生技术社区中的经验和案例,形成稳定流程的云原生技术开发。并由云原生技术中的微服务架构,独立地对不同功能进行优化升级或者扩展等操作,同时不影响用户的使用体验,提高项目对实地项目的实时需求反馈。

(三) 云 MES 平台的使用必将是一个长期的过程,这势必要根据后续的稳定维护作为基础。完善的云 MES 服务应该不仅包括前期调研、中期多端开发设备升级,更重要的是应用于车间后的全面运营维护。针对如何长久地运行将可以如下图架构般使得后续运维更加专业化。

通过云平台提供方和制造商企业双方配合一起分担相应的运维负担^[13]。

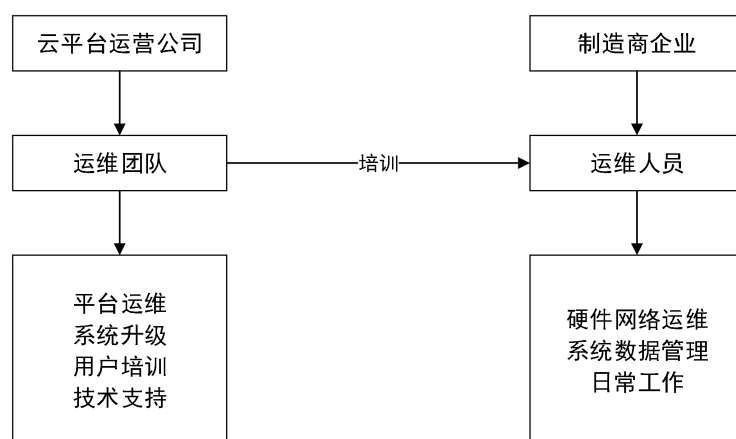


图 1.3 后续运维流程

（四）云 MES 服务可以在未来进行行业的整体性部署。通过云 MES 服务的数据互通功能，制造商工厂可联合上下游工厂合作，形成产业链系统数据化整体管理^[14]。尤其是在产品互补，产业上下游的产业园区，通过云 MES 系统，将各个部件制造厂商的加工生产数据整合起来，对多个企业进行统一管理。经此统一管理，打破数据孤岛的现象，使得合作的多个制造商厂商实现共赢，共同提高生产效率和加强管理控制。除此之外，整条生产链的数据展示更能直接反映出该行业的发展前景，在政府或者行业协会进行决策规划时提供有力的参考，清晰该行业未来的道路方向。同时，云 MES 在发展过程中可以由各个行业相关协会或者组织牵头，制定标准，帮助该行业部署适合该行业的云 MES，助力转型。

随着云平台、大数据、安全传输等相关技术的发展^[15]，云 MES 系统将更加完善。由此达到国内制造业厂商全面普遍地对传统车间生产模式的数字化升级管理改革的目标，响应国家《中国制造 2025》政策的号召，进一步提高国家生产力。

1.4. 论文结构安排

本文通过仪表设备生产管理系统的设计，为传统制造业进行基于云 MES 系统的数字化改革。本文主要通过整体系统搭建设计、云平台资源调度、数据传输加密三方面进行介绍。具体章节安排如下：

第一章 绪论。本章主要介绍了云 MES 系统的优势，并介绍了云原生技术，同时分析了国内外云 MES 系统应用的现状，对未来云 MES 系统的发展趋势做出了展望，为本文的设计提供了方向。

第二章 基于云 MES 的仪表生产管理系统设计。本章首先从整体介绍了需求以及框架。然后通过云平台、客户端、传感器和网关四个模块的设计，详细地展示了该系统的主要功能。

第三章 基于 MES 平台的资源调度的设计优化。本章首先通过当前任务调度的现状提出了 PSO 粒子群算法，在介绍了 PSO 算法的原理以及具体步骤后，对其优势与缺点进行分析。并从以下三方面来优化：在创建粒子群时通过熵进行筛选；将相关参数根据迭代次数进行动

态调整；在粒子群迭代过程中过早陷入局部时判断早熟，并通过混沌扰动跳出局部。测试表明优化后的 PSO 与 Min-Min 算法、Max-Min 算法、原始 PSO 算法比较都有很大的提高，具有很好的使用效果。

第四章 基于云平台的数据传输安全优化设计。本章首先对当前 MES 系统数据传输的现状进行介绍，并针对潜在的问题提出了加密优化的方案。在协商层设计中通过加密、签名以及数字证书多种方式结合提高可靠度。同时通过硬件加密卡向服务器内核的 Linux 算法库添加了国密算法，便于数据包在传输过程中基于国密标准进行处理。测试表明此种优化方式在传输过程已成功加密，并有很好的稳定性与传输速率，适合实地进行应用。

第五章 总结与展望。总结本文所设计的系统与相应的优化点，并且针对这个系统的一些可优化点提出简单的改进方案，对系统后续优化指出方向。

第二章 基于云 mes 的仪表生产管理系统设计

仪表盘在生产过程中，要经过多种工序、多台设备加工，才能完成一个仪表。本文进行制造业数字化的工厂负责电动车仪表盘和汽车仪表盘的加工组装，将仪表底盘、线路板、标度盘、指针、仪表面罩等零件通过螺丝、铆钉等工具进行组装。两种仪表盘功能相似，但在加工过程中所需要的部分零件不同。其仪表盘一般经由底盘模块简单组装、气压机注气加工、压针机安装指针、旋铆机钉铆钉进行连接、注油机往指针连接处添加润滑、四头绕线机自动绕线、自动螺丝机连接零件、吹净箱往仪表盘中吹净杂尘、高温老化箱对仪表盘进行老化测试这多个步骤才能实现。制作仪表盘虽然看似每个步骤都很简单，但是对精密仪表盘的灵敏度来说，都有着重要的关联。

当今采用传统生产流水线的仪表盘制造工厂，在生产管理中有以下几个痛点：1、订单的产量不多，但是其不同订单需求的产品种类繁多。人工分配工单易造成车间闲置的结果，无法发挥车间的最大效率 2、其工序集群呈孤岛式分布，生产设备、工作人员与生产任务等信息高度离散，难以进行系统管理 3、产品后续发现缺陷问题，难以对其原因进行及时追溯。4、设备在出现故障或者到寿命期限才进行维修，具有滞后性。

本文提出云 mes 系统来解决上述痛点。云 mes 系统采用云原生技术、微服务架构，并通过工单排序调度、车间生产执行、生产数据采集、产品质量管控、设备维保管理、产品追溯管理这几个模块打造仪表盘制造数字化工厂，实现业务在线化、制作过程数字化，并通过传感器设计、网关设计、云平台构建、客户端设计这四方面来实现。

2.1. 系统设计

2.1.1. 需求分析

根据当前市场分析，目前大多数传统制造业企业存在设备生产车间混乱、管理不到位、管理层对于车间进度不透明等问题，基于这些问题，对于需要设计的仪表设备生产管理系统，本文总结出了以下几点需求：

（1）工单排序调度。根据工单的信息以及当前车间的任务状况计算出效率最高的方案，凭借该最高效率的调度方案下发任务给车间。

（2）车间生产执行。车间接受下发的任务并在完成后进行上报，对任务进度进行实时监控。记录各个车间的工作效率以及工厂计划和实际情况的对比等信息，便于管理层对工厂后续发展的计划。

（3）生产数据采集。实现设备数字化，通过传感器和网关，远程实时监控状态和产量，并进行统计。

(4) 产品质量管控。在设备完成后通过质检员检查不良数并上报, 以提高产品质量。

(5) 设备维保管理。对于保养计划、保养情况和维修情况进行记录, 便于了解设备寿命状况。

(6) 产品追溯管理。对于生产工单的每个流程做到监控，能清楚及时地追溯设备的每一步的操作工、负责人以及设备，方便在产品出现问题时及时改进。

(7) 开发相对应的软件服务, 包括 web 端、PDA 端、小程序端。在遵循人性服务、科学生产的原则下, 多元化满足在不同身份或者不同情景下的软件需求。让该系统能在最大程度发挥它的作用。

2.1.2. 整体框架

本文将采用物联网技术，设计了一套基于云 mes 技术的仪表设备生产管理系统，在设计模块上，该系统设计分为云平台设计、客户端设计、传感器设计、网关设计，以此完成对于产品生产过程中的数字化管理改革。整体架构如图 2.1 所示。

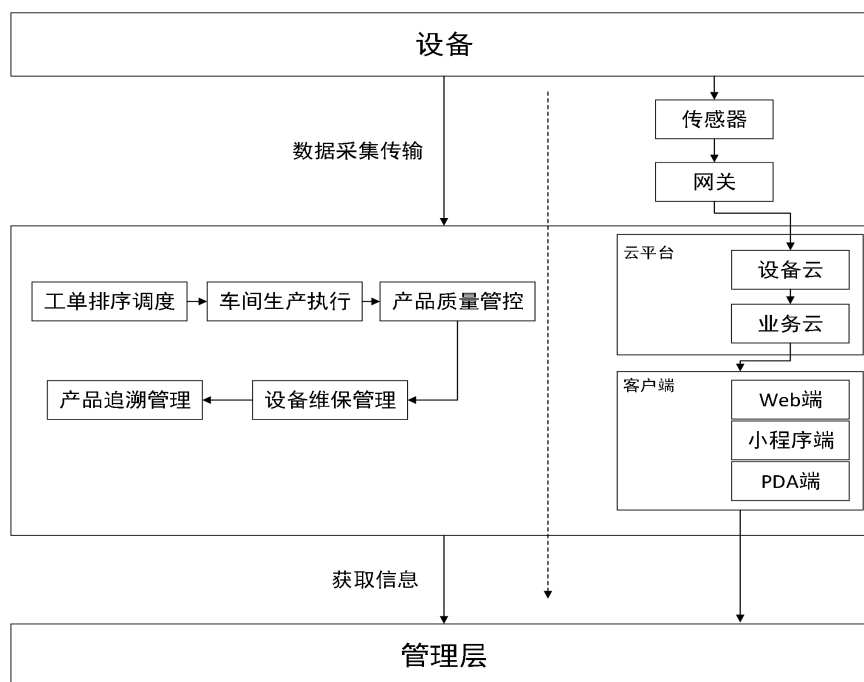


图 2.1 设备生产管理体系架构

设备首先通过传感器将产量和状态等数据进行采集，根据每个设备的使用方式采取了不同的传感器和收集数据方式，对仪表生产过程中的每一个设备和步骤都做到了数字化监控，并经由网关上报到云平台，实现对生产数据实时采集与传输。

云平台设计包括业务云平台和设备云平台。设备云与设备进行连接，设备采集完数据后将数据传输给云平台，实现数据采集和数据实时展示与统计。业务云则通过数据存储和管理，对客户端软件提供 API 调用功能，与客户端进行数据传输。

客户端为用户提供工单排序调度、车间生产执行、产品质量管控、设备维保管理、产品

追溯管理的操作以及相关信息展示。其中客户端设计包括 web 端、PDA 端和小程序三个平台。Web 端为企业提供设备管理、生产管理、维保管理、配置管理、数据统计等功能；PDA 端为操作工和质检工提供领单报工和质检上报功能；小程序则面向全员提供了统计数据的展示功能。

该系统实现仪表生产的智能化管理。通过实时监控设备、智能分配任务、展示任务进度、维保情况记录、生产效率计算、产品质量追溯等功能，科学智能化地管理仪表生产过程，多方面提高车间工作效率。

2.2. 云平台设计

本文云平台分为设备云和业务云，作为客户端和设备传输数据的桥梁，设备云和业务云各司其职：设备云与物联网设备对接，提供入网接口以及高适配的通信协议，实时收集设备数据和状态、并对设备实现远程控制；业务云则通过将数据进行整合与逻辑处理，为客户端提供用户所需求的产品、设备、人员等信息，由客户端对数据进行多元化展现。高效、安全、稳定的云平台为整个物联网系统提供高性能保证。

2.2.1. 设备云设计

本文设备云将采用中国移动 OneNet 云平台进行数据存储、通信和管理。OneNet 云平台可以与设备通过网关进行连接，为产品进行统筹管理记录，该平台支持多种类型的标准协议以及提供相应的 API 开发接口，以此承受大量设备与云平台的快速接入。OneNet 平台既对设备进行实时监控，以此来获得采集到的设备数据，又向 web 平台、移动端等提供设备数据，为智能工厂提供完善的物联网管理数字化优化方案。

网关通过 GPRS、WiFi、4G 多种方式将采集到的设备实时数据根据 MQTT 协议上报到 OneNet 云平台。OneNet 平台在获取数据后需要将数据进行存储、分析，并与业务云对接，开发者可以通过基于 RESTful 方案设计的 API 获取设备的实时数据，并对设备进行控制。如下图所示：

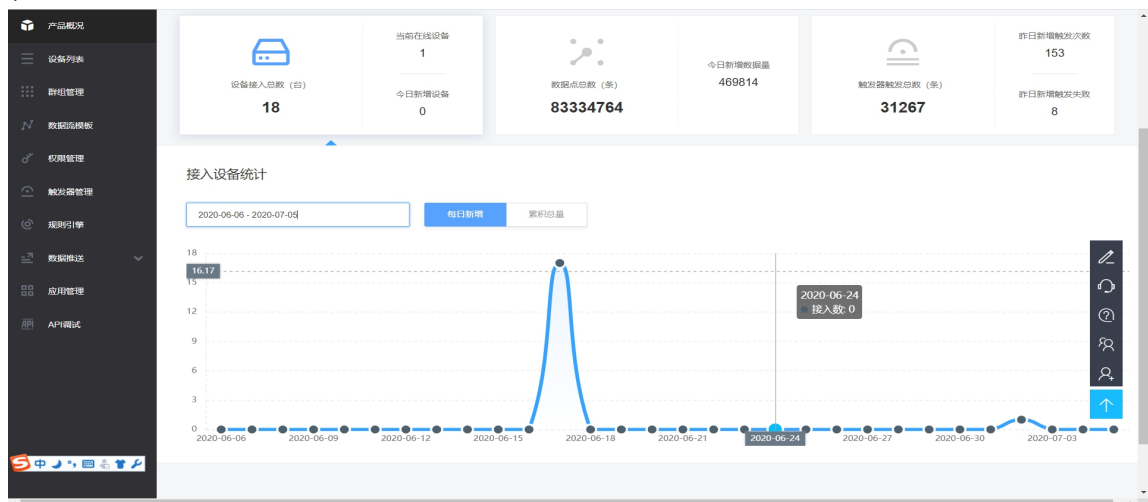


图 2.2 OneNet 设备云管理平台

2.2.2. 业务云设计

本文的业务云主要包括负载均衡、云服务器以及数据库三方面。首先由外网通过 Nginx 服务对服务器发起请求，以此来把客户端项目部署在 Nginx 服务器中，由此作为外部 SLB，接入外部流量到内部 K8s，K8s 相当于云端的操作系统，可以在上方部署云原生应用。Nginx 服务器具有处理静态请求的高性能、高扩展性、高可靠性以及低内存消耗的优势。后段项目则将单独部署于 tomcat 服务期内，令 tomcat 服务器能够更加专注于处理动态请求，提高服务器整体性能。

除此之外，为了减少服务器压力，客户端项目的一些图片资源将存放于 OSS 对象存储中，也方便后续对接 CDN，来优化网站的访问能力。在持久化层面上，本文采用 MySQL 主备的方式。当主机出现意外发生宕机，从机可以马上变为主机。同时在数据库前级加入 Redis 缓存，减轻数据库的负载，加速数据操作。Redis 缓存是按照主备模式设置的，由主库负责写入和读取，而从库只负责读取，在高并发下可以分担缓存的读取压力。并且本文设置了哨兵集群负责监控主备服务器，如果主服务器一旦宕机，哨兵可以切换主机，整个 Redis 哨兵负责对外接受请求。业务云整体架构如图 2.3 所示。

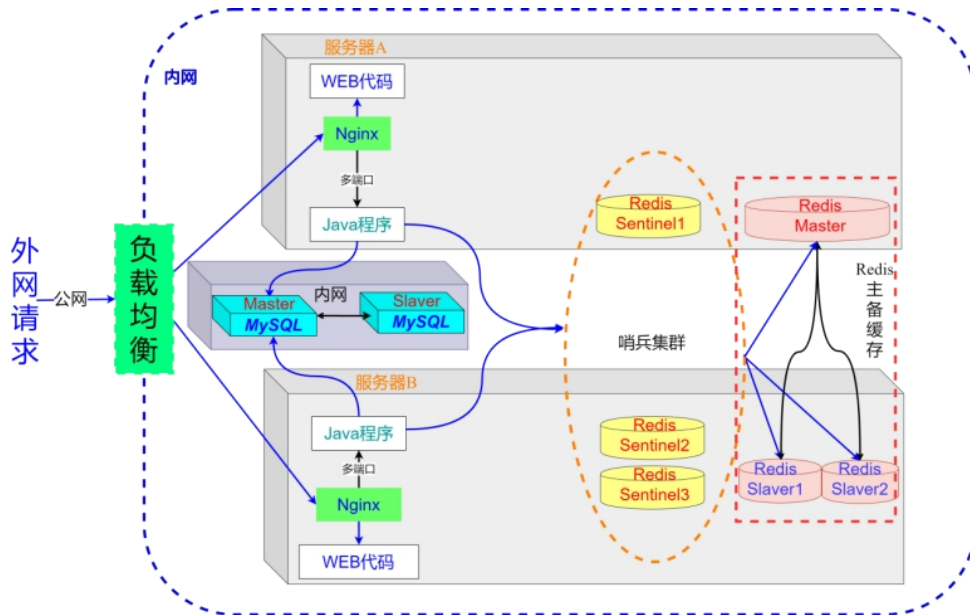


图 2.3 业务服务器整体架构

2.3. 客户端设计

为了更好地服务企业，web 端将根据账号密码判断职位与权限，对不同职位的员工根据他们的使用场景与需求展示相应的功能菜单栏。其职位角色主要分为：（1）管理层：对整个生产线进行全局布局管理（2）业务员：负责业务，对任务进行分配。（3）设备管理员：对仪表设备进行保养维修，并进行线上记录上报。（4）质检员和操作工：都是一线工人，质检员负责在仪表设备完成后进行检查，保证设备出厂不会出现质量问题。操作工负责仪表设备

的零件加工。其中管理层拥有最高权限，其余职位能就自己职责范围具有相应权限。

根据每个职位需求的不同，可将客户端分为 web 端、PDA 端、小程序端，以此进行更加多元化人性化地服务。

Web 端可以进行大屏展示设备监控、生产管理、维保管理、配置管理，不同职位的人登录显示其权限的功能。PDA 端由于其便捷性以及个人特性化，操作员和质检员可以登录进行领工、报工。小程序端可以就综合信息看板、设备运行监控、生产工单以及生产绩效这四个部分进行统计展示。

2.3.1. web 管理端

本系统 web 端将通过配置管理模块实现基础信息配置，通过设备管理和大屏展示实现生产数据采集后的实时展示，通过生产管理模块实现工单排序调度、产品质量管控、产品追溯管理的操作，通过维保管理模块实现设备维保管理的操作，

2.3.1.1. 技术架构

Web 端是基于 PC 浏览器采用 Vue.js 框架进行开发。Vue.js 框架通过 MVVM 的概念使数据实现数据层与视图层的双向绑定，使开发者不再花大量时间拘泥于 dom 对象。Vue.js 由于拥有成熟的组件库和中文社区，可以大幅度减小开发成本，也容易在项目后续进行维护。同时，Vue 也具有复杂单页应用的技术支持，让用户在跳转时只需要刷新局部数据，减少服务器响应，大幅度减轻了服务器压力。

前端整体技术架构将以 Vue.js 为核心，除此之外将通过以下几部分配合页面的搭建：1、路由管理部分：vue-router，管理路由跳转、路由状态；2、状态管理部分：vuex，管理全局下的数据状态。3、视图管理部分：element 和 echarts，通过表格和折线图、柱状图等将数据直观化。4、请求管理部分：axios，与后台进行交互，获取数据或者提交请求。

最后通过 webpack、npm、node.js 搭配实现工程搭建的完成。

整体框架如下所示：

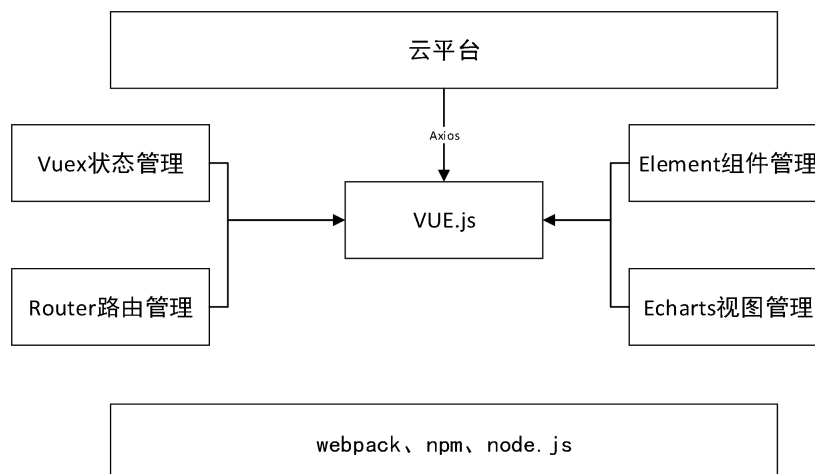


图 2.4 Web 端技术架构图

2.3.1.2. 功能设计

Web 端具有多种功能模块：大屏展示、设备管理、生产管理、维保管理、配置管理，各个功能部分互相衔接配合，因此在做好功能划分后应该做好整体规划。其各个功能模块流程如下所示：

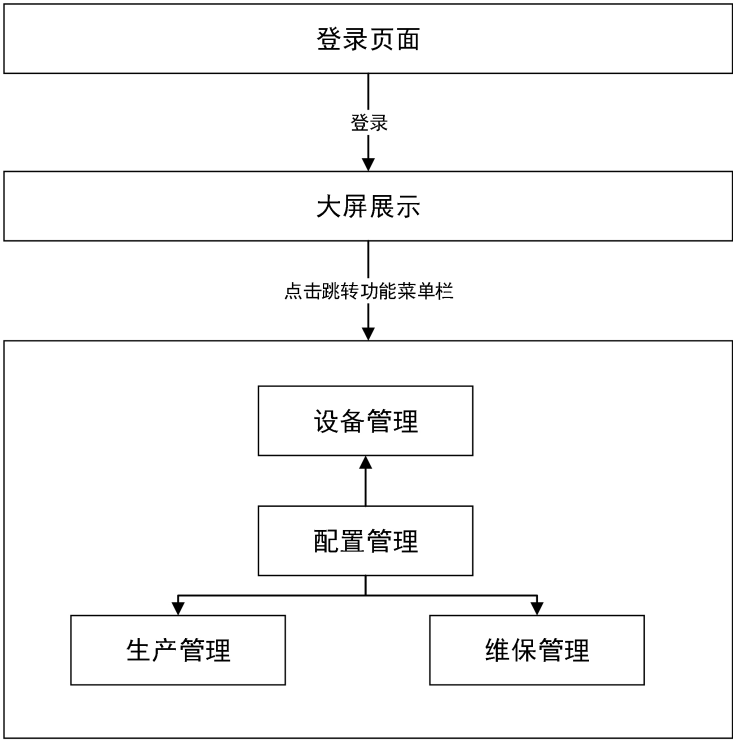


图 2.5 Web 端功能模块架构图

(一) 大屏展示

用户在登录之后首先进入大屏页面。如图 2.6 所示。在大屏页面中将会对设备运行情况、设备综合使用率、计划达成率、工单状态统计以多种方式进行展示，并可以实时展现某个设备的在线状态，可以通过电视机、大屏等投影方式进行展示。如图 2.7-2.8 所示。



图 2.6 web 端大屏展示



图 2.7 大屏显示车间图



图 2.8 大屏显示工厂图

(二) 设备管理设计

该模块将实时显示所有设备的状态、当日产量、累计产量、运行时长等信息进行展示具体如下图所示。

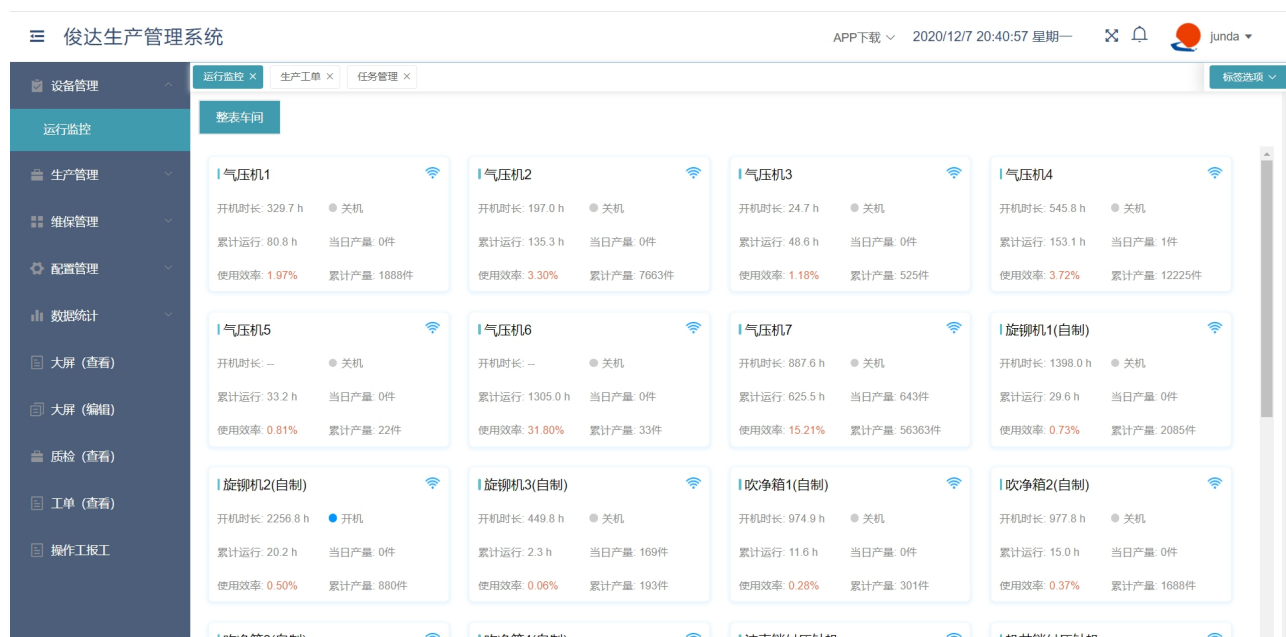


图 2.9 运行监控页面

为了实时获得设备的情况，定时向服务器发起请求获取数据这个方法虽然可行，但是频繁的发送 axios 请求会占用不必要的资源。为此，本文采用 WebSocket 的形式在设备发生状况时服务器主动发送设备信息给客户端，减少请求次数。

传统的 HTML 协议通信只能被动地由 web 端发起，服务器无法主动地将信息推送给 web 端。而在这方面 WebSocket 不仅可以从客户端向服务器发送请求，服务端也能自己向客户端发送警告数据^[13]。由于是通过 TCP 协议，服务器端在部署过程十分方便，并且达到了真正的双向平等对话。在 WebSocket API 中客户端与服务器双方在完成第一次握手后，不需要继续请求，双方已经搭建了平等的持久的桥梁，并可以进行双向数据传输^[14]。

因此在设备管理中，可以通过 WebSocket 协议来呈现数据的实时刷新，由服务器进行定时推送。具体操作如下所示：当用户想进行 WebSocket 交互时客户端会对服务器进行相关请求，并在请求中声明对升级为 WebSocket 的需求。服务器接收到 HTTP 请求后会对此进行解析，并且对客户端回复相关信息，至此，双方持久性的连接就建立好了。服务器可以定时往客户端发送设备的各种实时数据，客户端接收到后刷新数据，将其展示在设备管理页面上。这个连接将持续到其中一方主动关闭连接。如下图所示。

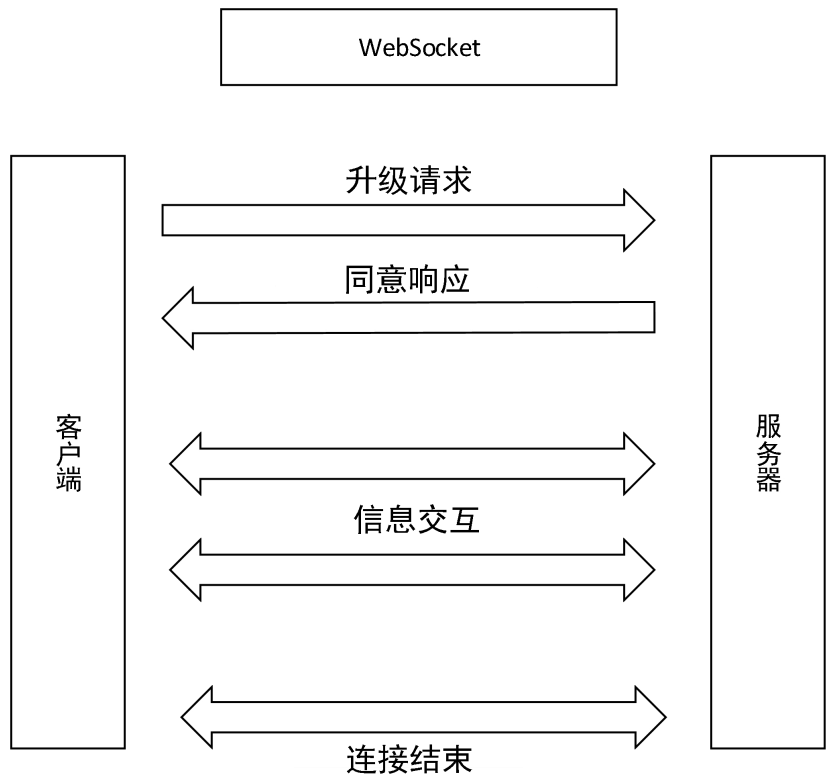


图 2.10 WebSocket 交互过程图

(三) 生产管理设计

在该系统中，工单处理流程具体如下所示：

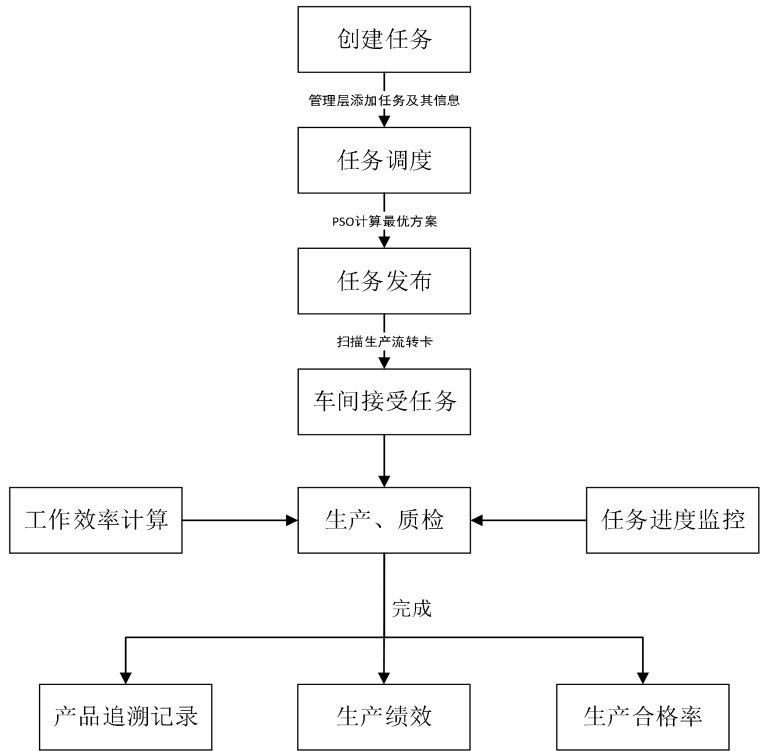


图 2.11 工单处理流程图

工单流程首先由管理层发布任务及其人物信息，并由系统通过任务调度计算自动派发工单给相应车间。**Web** 端登录后在生产管理-任务管理界面会对各任务的具体信息通过表格的方式进行展示，车间负责人可以通过打印按钮将某任务的生产流转卡打印出来并张贴在车间内。其生产流转卡中附有一个二维码，车间工人可以通过扫描该二维码进行领单、上报、质检等操作。

生产流转卡			
计划单号	RW_202012002		
产品信息	产品名称	QJ125F电喷	
	产品规格		
计划内容	计划生产数量	1000	
	开工时间	2020-12-02	
	完工时间	2020-12-05	

图 2.12 生产流转卡

在 **web** 端的生产管理-生产工单页面实时显示该任务进度，管理层可以随时了解任务的完成情况。若生产完成且质检合格，那么该工单标记为已完工。如下图所示。

俊达生产管理系统										2020/12/4 00:35:50 星期五				junda	
运行监控 × 任务管理 × 生产工单 × 质检管理 ×										标签选择					
生产工单										开始日期 至 结束日期					
生产工单										状态					
生产工单										产品名称					
生产工单										Q 搜索					
生产工单										重置					
生产工单										序号					
生产工单										工单号					
生产工单										车间班组					
生产工单										产品名称					
生产工单										产品编码					
生产工单										规格型号					
生产工单										计划生产量					
生产工单										开始时间					
生产工单										结束时间					
生产工单										生产人					
生产工单										状态					
生产工单										进度					
生产工单										用时					
生产工单										操作					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					
生产工单										报工记录					
生产工单										重新派单					

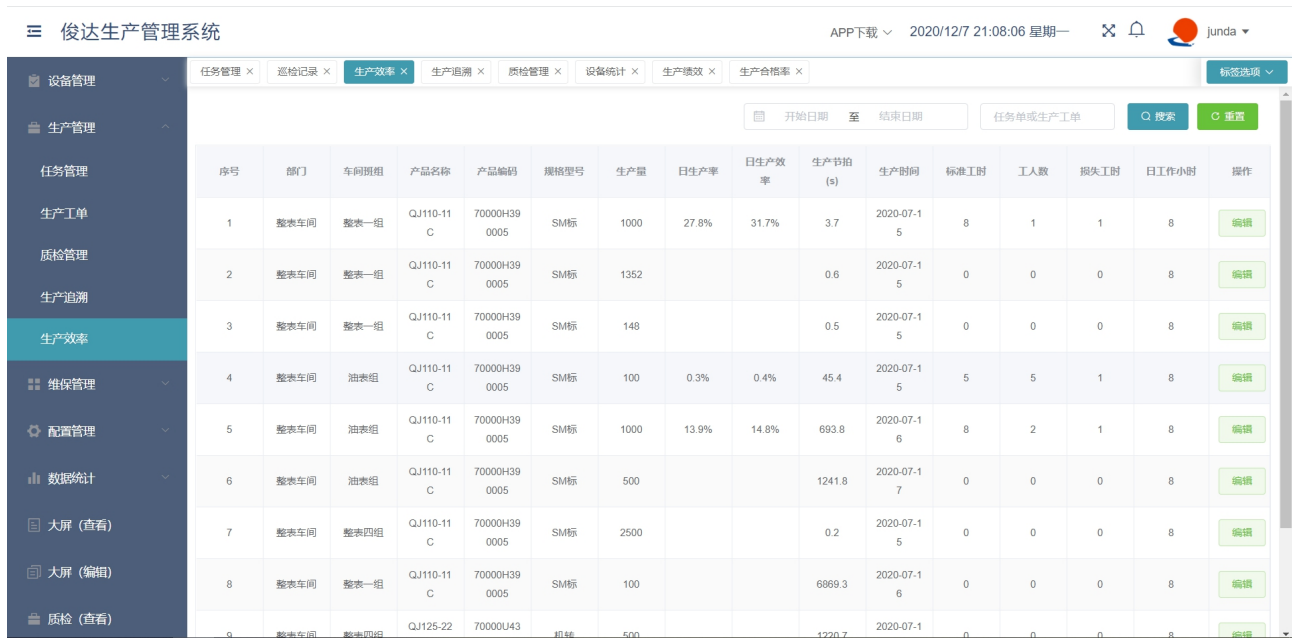


图 2.14 生产效率页面

该任务完成后，每个批次的任务都会在生产管理-生产追溯对质检员与操作工进行记录，方便总结回溯。除此之外，该批次的生产合格率会在数据统计-生产合格率页面计入所在车间的历史合格率中，同时，该批产品的计划时间和实际完成时间作为生产绩效会在数据统计-生产绩效中计入所在车间的生产绩效中，合格率与绩效可通过图形或者表格的形式进行展示。如图所示。

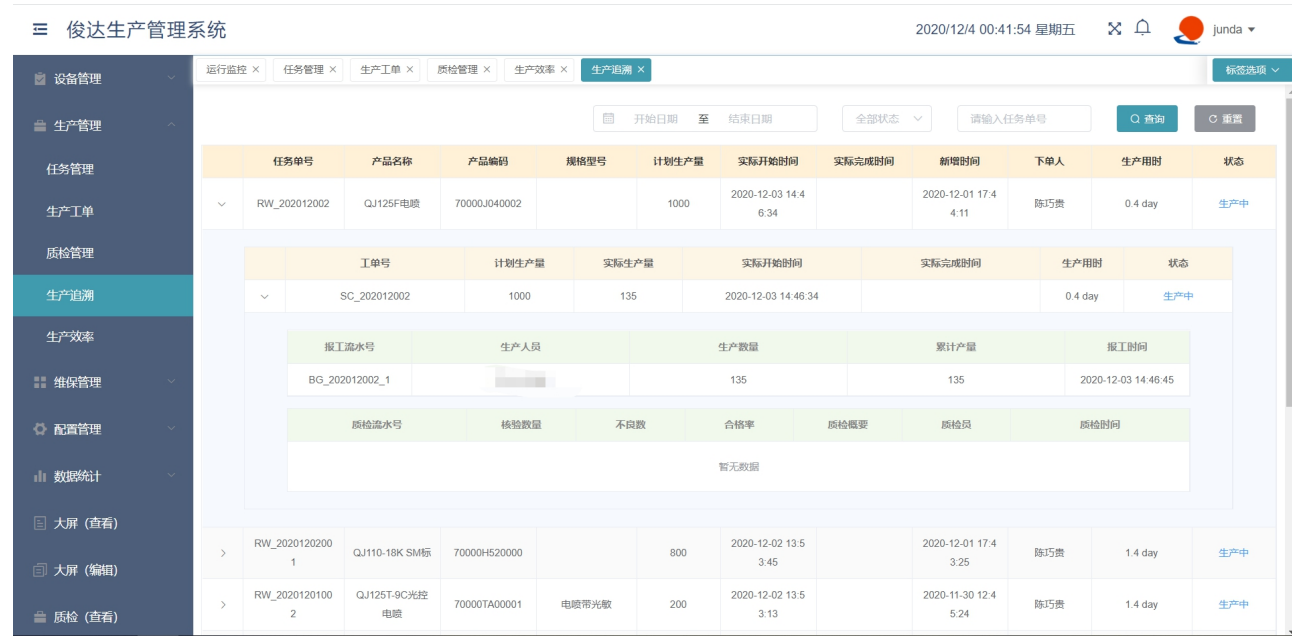


图 2.15 生产追溯页面



图 2.16 生产绩效页面



图 2.17 生产合格率页面

(四) 设备维保设计

为了设备能持久稳定的工作，本文将定期设备保养，并在发现故障后及时上报维修。其具体流程如下所示：

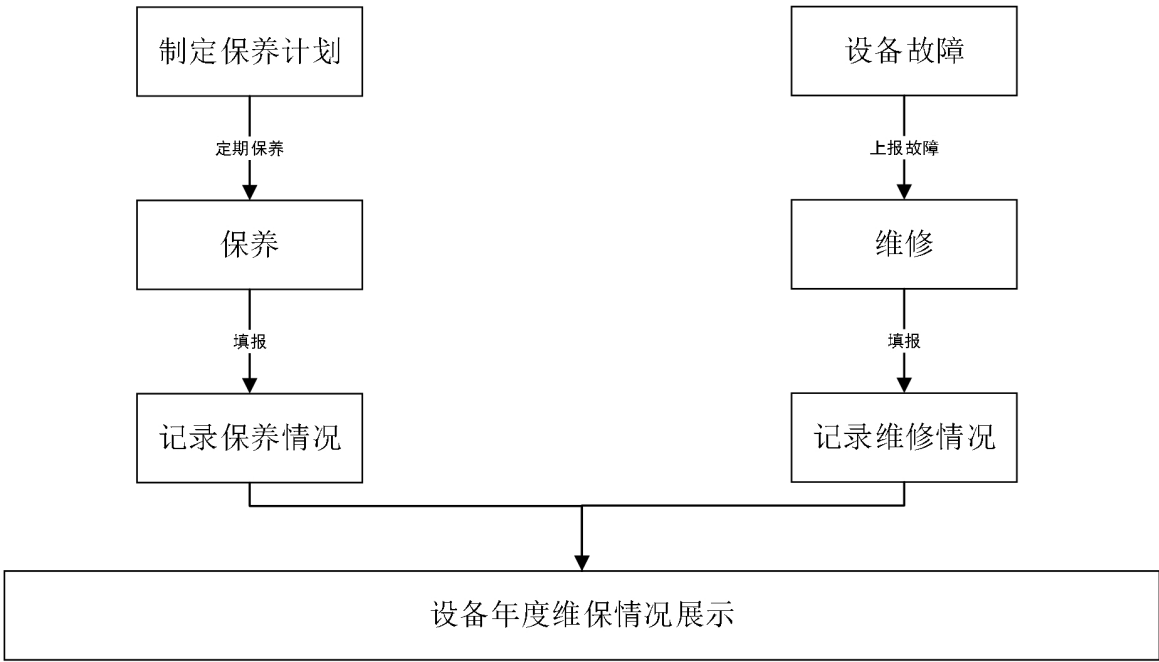


图 2.18 设备维保流程图

首先对不同的设备进行相应的保养计划，在 web 端维保管理-保养计划页面中以表格的方式展示保养的周期和日期，并可根据实际情况对保养计划进行增删改操作。如下图所示。

俊达生产管理系统2020/12/4 00:52:22 星期五junda

序号	设备名称	设备编号	规格型号	设备类型	保养类别	保养周期	保养时间	担当人	责任人	创建时间	备注	操作
1	搅拌机1(自制)	JDYB/SB-03-015	自制	整表设备	A型	月	15 19:00			2020-07-15		修改删除
2	气压机1	JDYB/SB-03-001	自制	整表设备	A型	月	20 10:00			2020-06-20		修改删除
3	高温老化箱	JDYB/SB-03-024		整表设备	A型	月	19 15:00			2020-06-19		修改删除
4	高温老化箱	JDYB/SB-03-024		整表设备	A型	月	19 15:00			2020-06-19		修改删除
5	高温老化箱	JDYB/SB-03-024		整表设备	A型	月	19 12:00			2020-06-19		修改删除
6	手扳压机011	JDYB/SB-03-011		整表设备	A型	月	19 12:00			2020-06-19		修改删除
7	气压机1	JDYB/SB-03-001	自制	整表设备	A型	月	19 12:00			2020-06-19		修改删除

共 7 条1

图 2.19 保养计划页面

在计划时间内对设备进行保养后，通过维保管理-保养记录进行上报，对设备保养的情况进行及时的汇报，以便发现设备故障的苗头并扼杀于摇篮中。

俊达生产管理系统

2020/12/4 00:58:08 星期五

生产效率 × 生产追溯 × 维保计划 × 保养记录 × 维修记录 × 设备统计 × 设备维修 × 设备保养 ×

生产计划 生产记录 维保管理 保养计划 保养记录 维修记录 配置管理 数据统计 大屏(查看) 大屏(编辑) 质检(查看) 工单(查看)

开始日期 至 结束日期 全部 请输入名称或编号 查询 重置

序号	设备名称	设备编号	规格型号	设备类型	保养类别	保养周期	保养时间	担当者	责任人	新增时间	备注	状态	操作
1	气压机1	JDYB/SB-03-001	自制	整表设备	A型	月	20 10:00			2020-11-20		未保养	填报
2	高温老化箱	JDYB/SB-03-024		整表设备	A型	月	19 15:00			2020-11-19		未保养	填报
3	高温老化箱	JDYB/SB-03-024		整表设备	A型	月	19 15:00			2020-11-19		未保养	填报
4	手板压机011	JDYB/SB-03-011		整表设备	A型	月	19 12:00			2020-11-19		未保养	填报
5	高温老化箱	JDYB/SB-03-024		整表设备	A型	月	19 12:00			2020-11-19		未保养	填报
6	气压机1	JDYB/SB-03-001	自制	整表设备	A型	月	19 12:00			2020-11-19		未保养	填报
7	旋压机1(自制)	JDYB/SB-03-015	自制	整表设备	A型	月	15 19:00	吴林红	吴林红	2020-11-15		未保养	填报
8	气压机1	JDYB/SB-03-001	自制	整表设备	A型	月	20 10:00			2020-10-20		未保养	填报

共 41 条 1 2 3 4 5

图 2.20 保养记录页面

同时，在设备上出现故障时，通过维保管理-维修记录进行上报，当维修工解决好问题后将解决办法通过该页面进行上报，以随时了解维修情况。

俊达生产管理系统

2020/12/4 01:00:02 星期五

生产效率 × 生产追溯 × 新增 ×

生产计划 生产记录 维保管理 保养计划 保养记录 维修记录 配置管理 数据统计 大屏(查看) 大屏(编辑) 质检(查看) 工单(查看)

设备名称或编号 搜索 重置

查看

故障设备: 高温老化箱 规格型号:

设备类型: 整表设备 设备编号: JDYB/SB-03-024

故障时间: 2020-06-17

故障原因: 17号机以油过少

解决办法: 加注95#

维修人: 验收人:

当前状态: 已完成

取消 确定

验收人 维修状态 操作

已完成 修改 查看

已完成 修改 查看

已完成 修改 查看

待检修 修改 删除

待检修 修改 删除

共 5 条 1

图 2.21 维修页面记录

最后，可以通过数据统计设备保养和数据统计-设备维修两个页面以图形或者列表的形式了解维保的年度情况，对于设备寿命更加熟悉。



图 2.22 设备维保页面图形方式

(五) 配置管理设计

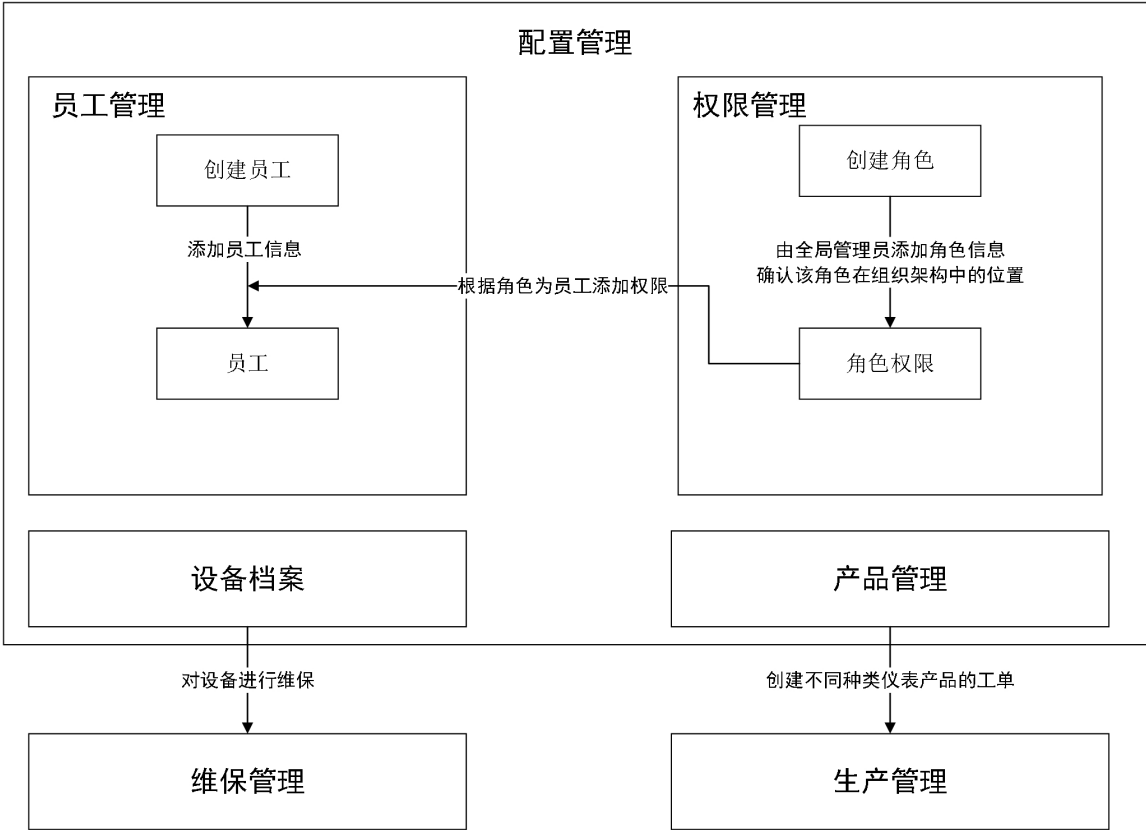


图 2.23 配置管理框架图

全局管理员可以通过权限管理页面新增角色，该角色享有相应的功能权限。例如创建的业务员身份登录后只有生产管理模块的页面展示。其中质检员和操作员由于其身份特殊性，

系统默认存在该角色。



图 2.24 权限管理页面

全局管理员可以通过员工管理页面添加新员工，并在信息中赋予新员工相关身份。



图 2.25 员工管理页面

在产品管理页面可以对产品进行批量导入，并对产品具体进行禁用、修改、删改操作。其中当某个产品被禁用后，那么在生产模块中新增任务不能选取被禁用的产品，禁用后可通过启用来恢复正常。

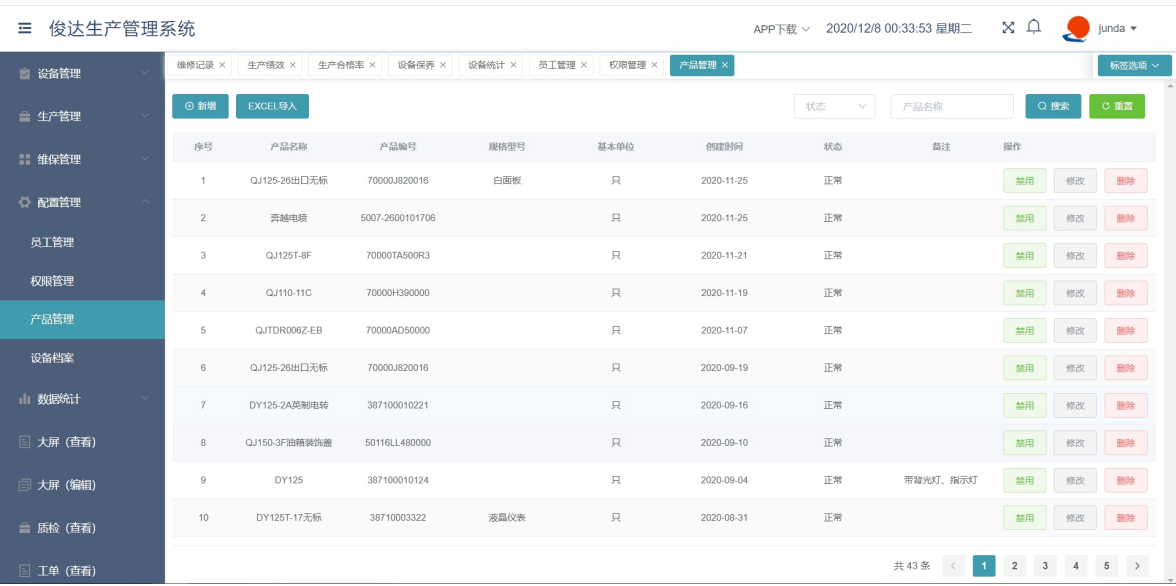


图 2.26 产品管理页面

在设备档案可以看到每个设备的描述，并可对设备进行报废、修改、删除操作。若对某台设备做报废处理后，维保管理中不能选取报废的设备，报废后可通过启用来恢复正常。

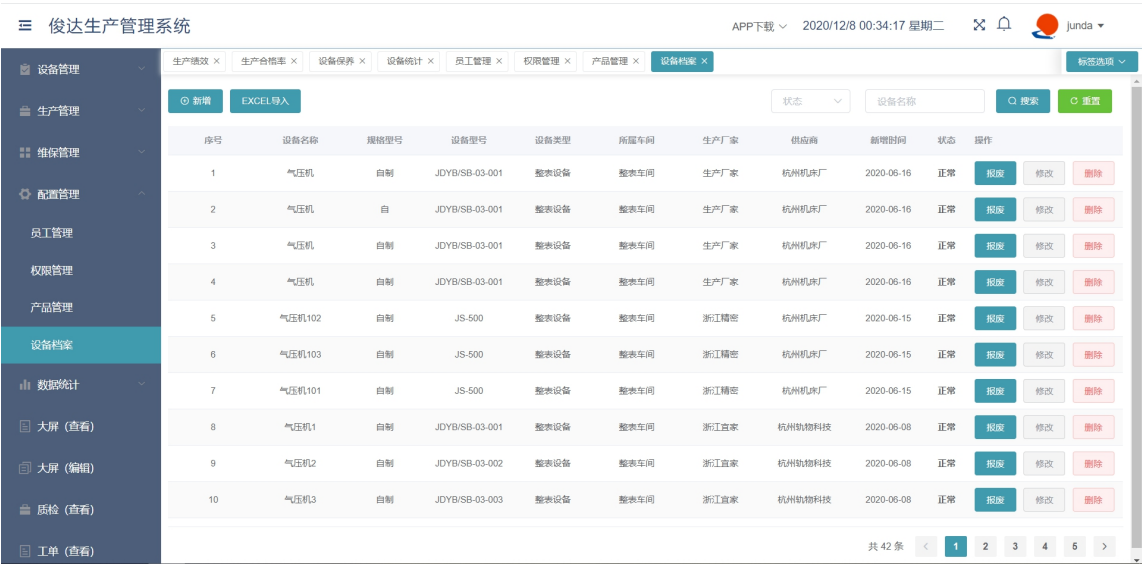


图 2.27 设备档案页面

2.3.2. PDA 端

本系统 PDA 端将通过操作工和质检员的领单、上报操作，实现车间生产执行中员工工作进度的监控。

2.3.2.1. 功能设计

PDA 端是基于 Html5 开发的网页，HTML5 是根据如今迅猛发展的网络世界，由 W3C 和 WHATWG 合作指定的关于 HTML、XHTML 以及 HTML DOM 的新标准^[15]。

PDA 端以一线工人操作方便为宗旨，只需在 PDA 端操作即可，扫描车间打印下来的生

产流转卡上的二维码即可进行相应操作。具体设计如下图所示。页面如图 2.14 所示。

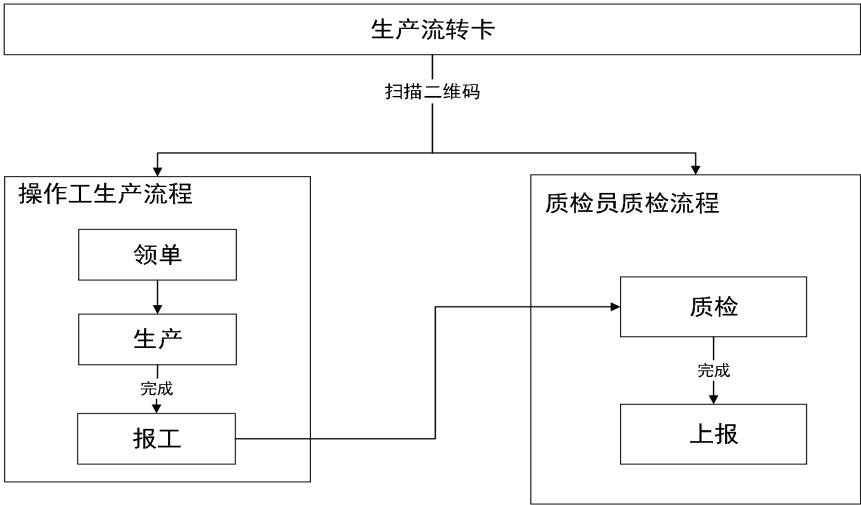


图 2.28 生产质检流程

车间中的操作工扫描二维码打开 PDA 端，在生产工单页面可以查看当前任务的具体信息，点击下方领单按钮进行领单。在完成任务后填写生产数量，点击下方报工按钮进行报工。如下图所示。

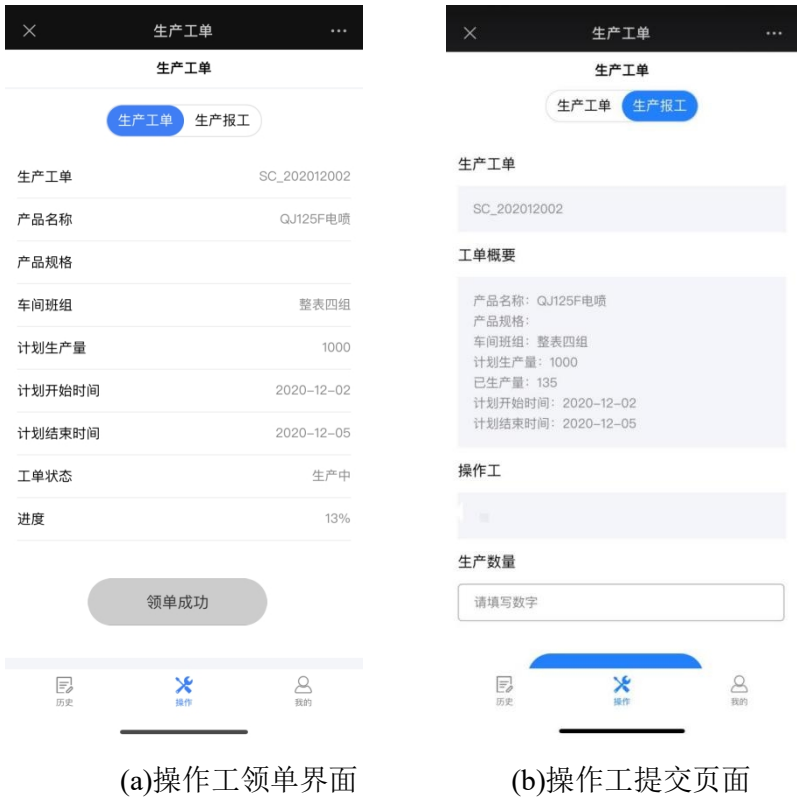


图 2.29 操作工 PDA 端

操作工提交后质检员可以通过 PDA 端扫描二维码登录，在任务工单页面查看当前任务的具体信息。点击 tab 按钮，跳转到质检页面，在线下对产品质量进行检查，检查完毕通过 PDA

端将结果汇报，其中汇报内容包括操作工，不良数，及质检概要（两项均为必填项）。



图 2.30 质检员 PDA 端

2.3.2.2. 系统展示

操作工和质检工有相应的 PDA 进行操作。可通过上方扫描口扫描二维码获取网站 url 进行后续操作。下图为 PDA 实物图。



图 2.31 手持 PDA 实物图

2.3.3. 小程序端

本系统小程序端将通过各界面的展示，实现生产数据采集时信息实时且便捷的获取功能以及车间生产执行时展示工单、车间、设备信息的功能。

2.3.3.1. 技术架构

本文小程序基于 uni-app 进行开发，uni-app 是由 Vue.js 进行跨平台开发的框架，具有开发成本低，扩展灵活度高、跨端抹平度高的特点。一份代码可以在浏览器、小程序、APP 平台多平台运行，并且开发者在开发中可以为专属平台调用它特有的代码，且不会波及到别的平台的开发效果。同时，uni-app 将大多数组件和 API 进行跨平台封装，满足普通的业务需求。

本文小程序中包含了多个图表，echarts 虽然是大多数开发者面对图表的选择。然而由于小程序无法对 DOM 直接进行操作，因此 echarts 在跨端开发远比在 PC 端开发复杂且不易维护，甚至还存在着 iOS 端图表数字或者柱状体发生错位或者消失等现象，对此官方表示只能通过引用页面来进行优化。因此本文将选择使用 ucharts。Ucharts 支持跨端开发，性能及用户体验反应很好，虽然较 echarts 图表种类功能较为简陋，但已满足本文的开发需求。

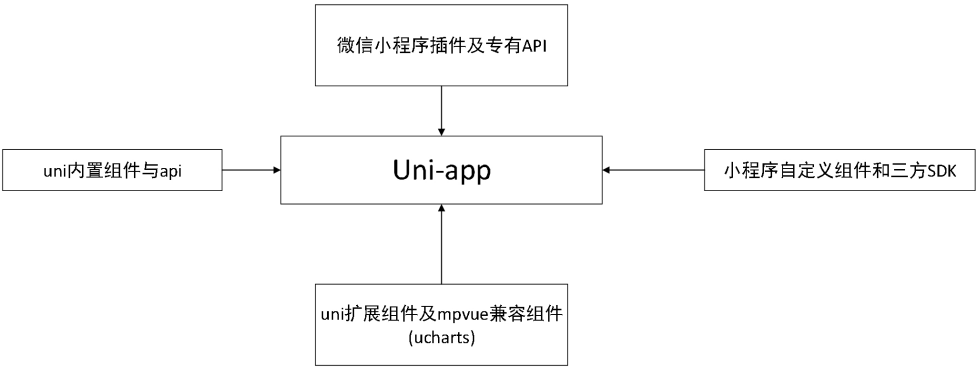


图 2.32 小程序开发架构图

2.3.3.2. 功能设计

小程序旨向全体人员，展示了设备数据、任务数据等统计，通过折线图、饼图等方式进行展示。具体功能如下表。

表 2.1 小程序端各页面功能

模块名称	功能概述
综合信息看板	展示设备运行状态饼图、设备本周运行时间柱状图、设备综合利用率柱状图、工单状态统计饼图信息
设备运行监控	实时显示每个设备的状态以及累积的开机时间和产量等信息。
生产工单	展示所有生产工单信息，可以通过时间、产品名称、产品编号、状态等进行筛选

生产绩效

通过柱状图的形式展示不同时间的车间的计划时间和完成时间，可以通过车间名、起始截止时间进行筛选。



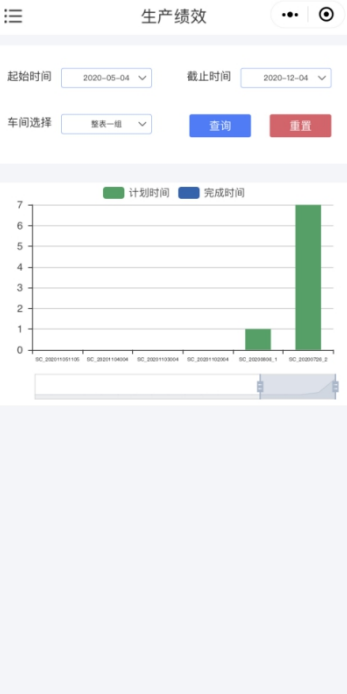
(a)综合信息看板



(b)设备运行监控



(c)生产工单



(d)生产绩效

图 2.33 小程序端页面展示

当用户登录综合信息看板模块，根据展示需求，小程序将直接加载两个柱状图和两个饼图，同时渲染多个 `canvas`，对于一些低端机来说，可能会导致卡顿、页面不流畅等现象。因此针对这个问题，本文具体通过以下举措进行页面的优化：

1、逐步加载：对于用户来说，在刚进入该页面不滑动时仅出现综合信息总览的数据直接展示和设备运行状态的柱状图展示。因此在初始化该页面时，客户端首先仅向服务器发送关于综合信息总览、设备运行状态这两块数据的请求，并在拿到数据后对于柱状图所在的 `canvas` 进行渲染。渲染完成后再调用 `ready` 方法对后续三个图表进行渲染。通过逐步加载来降低对于用户手机性能的要求。

2、适当缓存：缓存一些有效的数据，例如在用户切换菜单时，从别的功能模块切换回综合信息看板模块，无需反复向服务器发起请求，减少服务器响应时间，直接将缓存的数据渲染在小程序上即可。

3、优化代码：当前小程序的视图层与逻辑层在进行数据交互时，双方需要通过之前协议的 `evaluateJavascript` 所实现。由于是单线程，频繁调用 `serData` 将会导致 `js` 处理队列产生堵塞现象，界面渲染不及时，产生卡顿，因此要减少不必要的 `setData` 的使用。并且要控制服务器的回包大小，减少用户的等待时间。

2.4. 传感器设计

本系统将对于每一台加工的设备进行数字化监控，通过传感器的方式收集设备的状态和产量，并对其进行相对应的判断，实现生产数据采集。

本系统主要采用气缸磁感应传感器和电流互感器两种方式进行数据采集。根据传感器自身特性，每种设备依据自身特性选择不同的传感器，并以不同的方式进行数据实时收集。

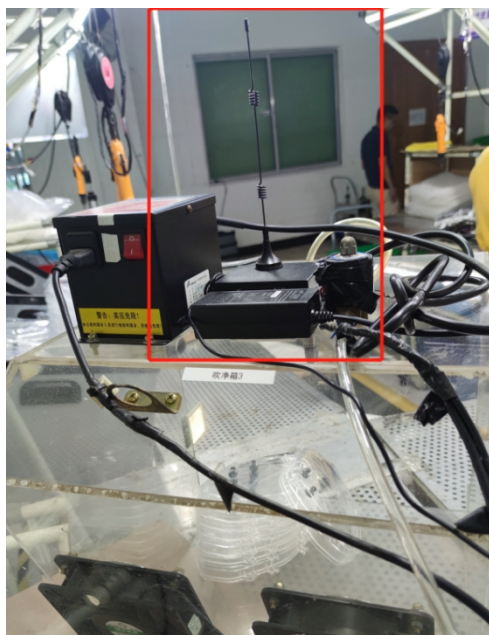


图 2.34 吹净器传感器实物图

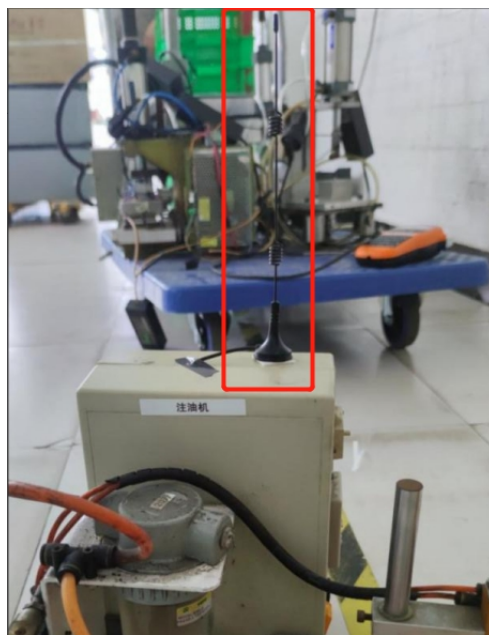


图 2.35 注油机传感器实物图

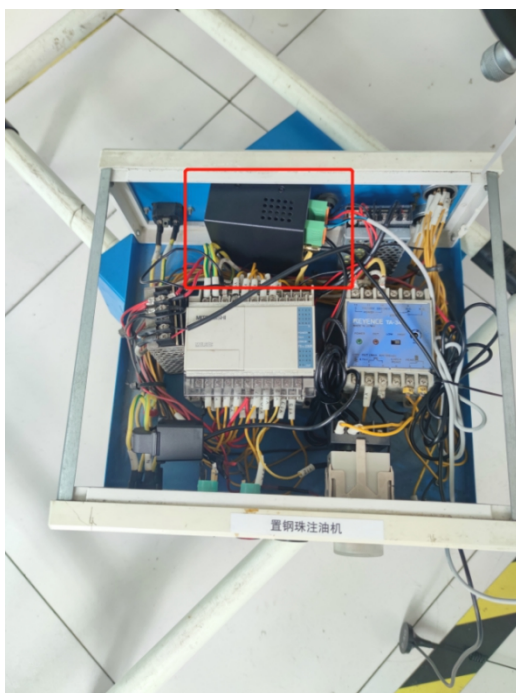


图 2.36 置钢珠注油机传感器实物图

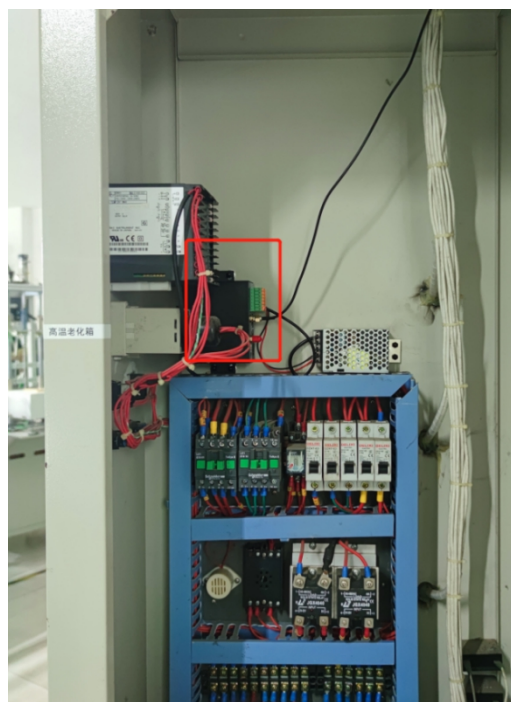


图 2.37 高温老化箱传感器实物图

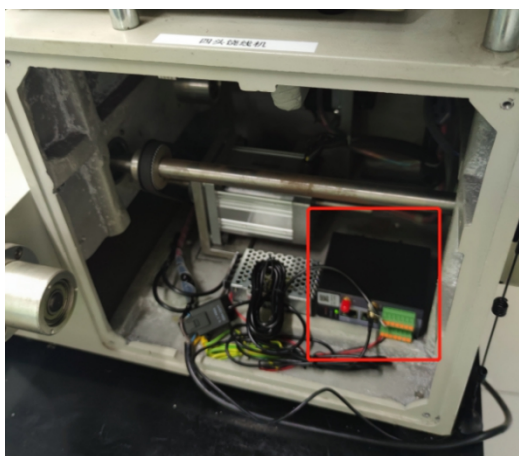


图 2.38 四头绕线机传感器实物图

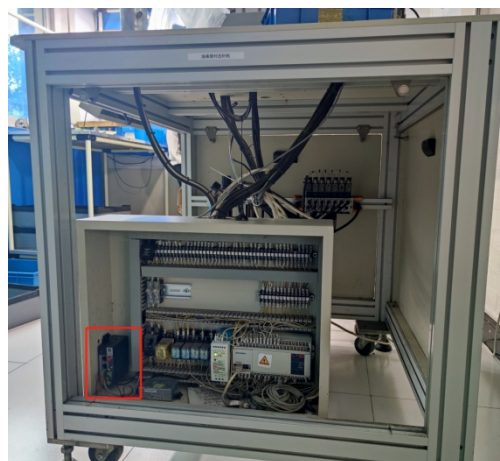


图 2.39 油表锁附压针机传感器实物图

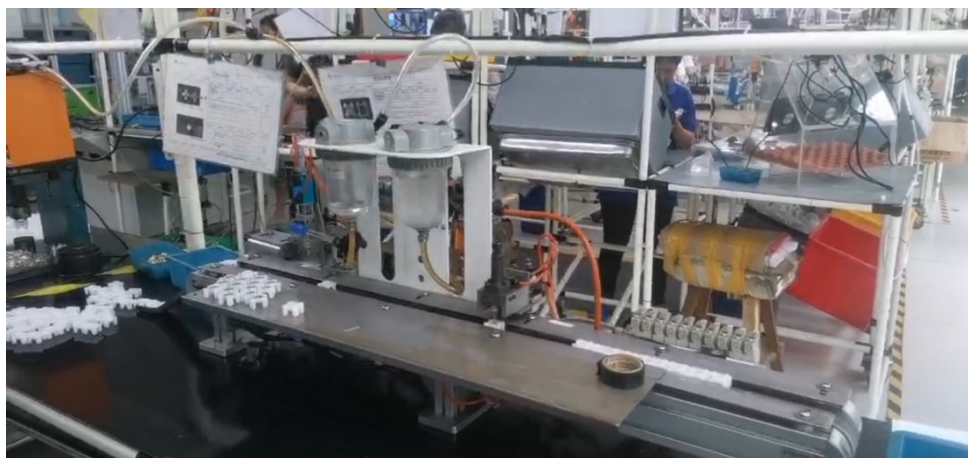


图 2.40 投入使用生产线图

本系统主要通过电流互感器和气缸接近开关两种传感器采集数据。

气缸磁性开关是通过安装在缸筒外方的磁性开关来进行感应，非磁性的活塞上方安装永久磁铁的磁环。通过活塞移动带动磁环，若磁环靠近开关，两条簧片会由于磁感而吸引接触，导致触点合并，相反，若磁环离开磁性开关，两条簧片的吸引力消失，将会恢复原有的状态，原本接触的触点断开。当触点发生接触或者发生断开时，电信号会突然产生或者消失，以此来让电磁阀实现相应的动作^[16]。

本文电流互感器采用的是 V9260 芯片，平均电流采样时间为 320ms，电流获取的间隔时间为 500ms^[17]。当设备在生产产品时，电流互感器会出现电流升高的现象，假设先后采样的为 A 点电流和 B 点电流，当 A 点电流小于设定的工作电流阈值，同时 B 点大于设定的工作电流阈值的情况下，则判定设备处于工作状态。反之，如果出现电流一直小于工作电流阈值，且超过了设定的时限，那么就判定设备并未活动。电流不仅可以记录设备的状态，还可以通过脉冲次数记录设备使用次数。

以自动螺丝机为例，该设备添加了电流互感器和气缸接近开关两种传感器，并且通过网关进行数据处理，其网关连接图如图 2.43 所示，实物连接图如图 2.44 所示。

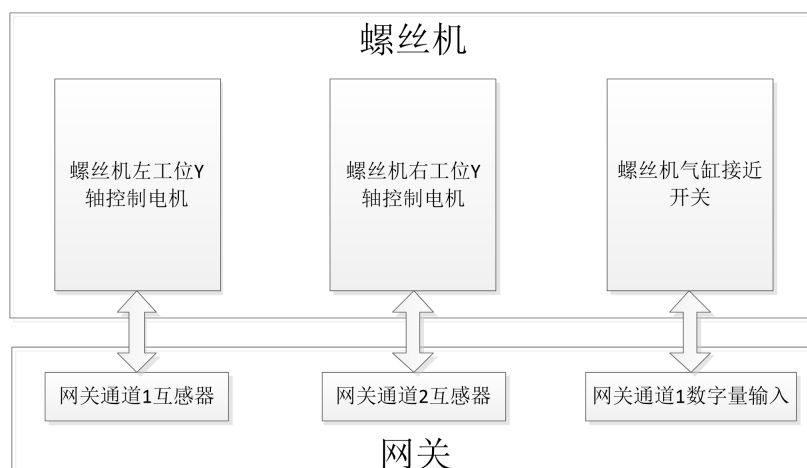


图 2.41 网关连接方式示意图



图 2.42 添加电流互感器和气缸接近开关后的自动螺丝机实物图

电流互感器和气缸磁性开关采集到的工作数据将通过网关上报给 OneNet 平台,如下图所示,可以观察到螺丝机左工位 Y 轴间隔为 7 秒。

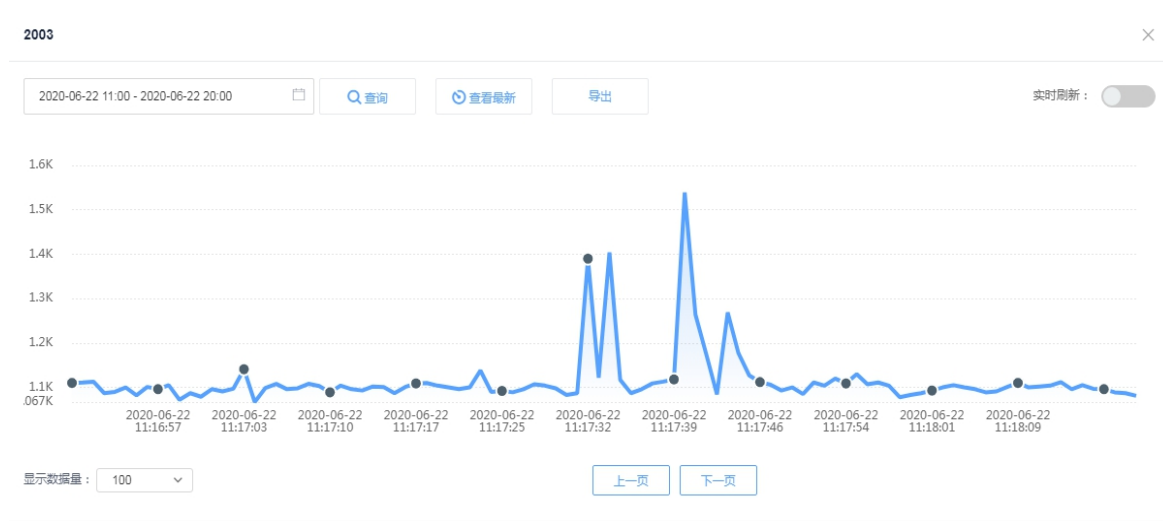


图 2.43 螺丝机左工位 Y 轴控制电机电流图

同时, OneNet 平台收到的螺丝机右工位 Y 轴控制电机电流波动如下图所示。可以看出当螺丝机右工位 Y 轴控制电机在工作时,会出现多次的电流脉冲,且脉冲电流都是大于 1300mA。其持续时间约为 9 秒,两个尖峰之间最大间隔为 6 秒。

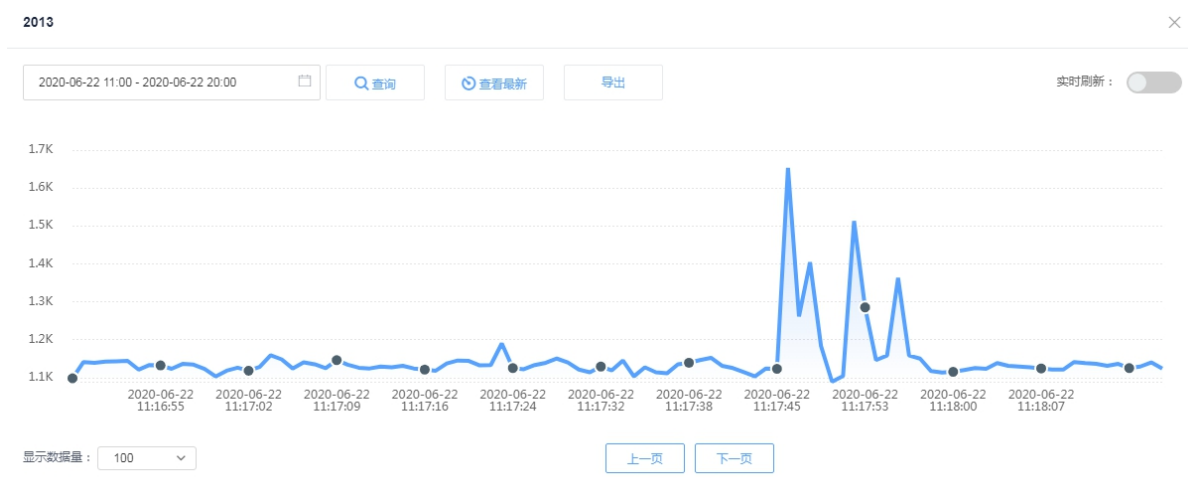


图 2.44 螺丝机右工位 Y 轴控制电机电流图

通过 OneNet 平台可以得知电流互感器采集到设备的数据并正常上报给云平台,根据电流数据计算自动螺丝机的产量。其产量判断方式如下图所示。

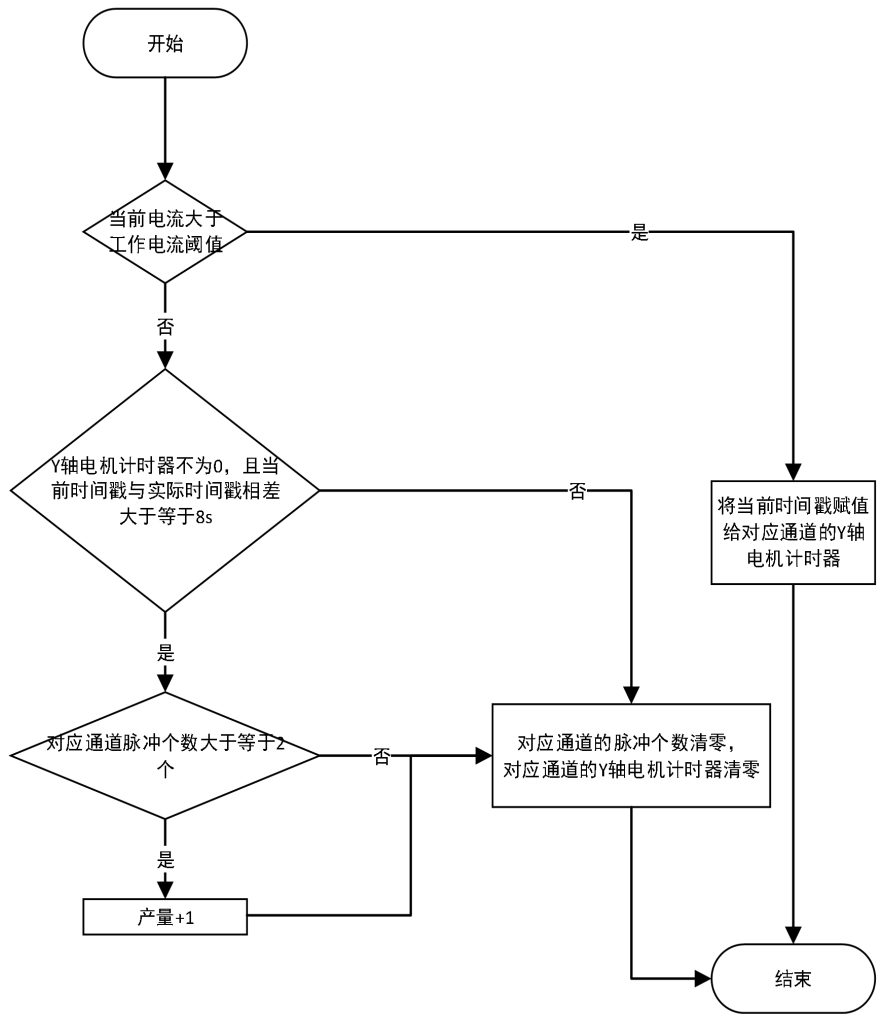


图 2.45 电流判断图

接近开关在螺丝机打螺丝时，会出现脉冲信号，通过脉冲信号可以感知，此时螺丝机打了多少颗螺丝。当接近开关出现有效脉冲时，判断此时设备处于工作状态。如果接近开关没有出现有效脉冲，超过 5 分钟以后，则判断此时设备处于待机状态。同时通过脉冲信号可以感知，此时螺丝机打了多少颗螺丝。具体如图 2.48 所示：

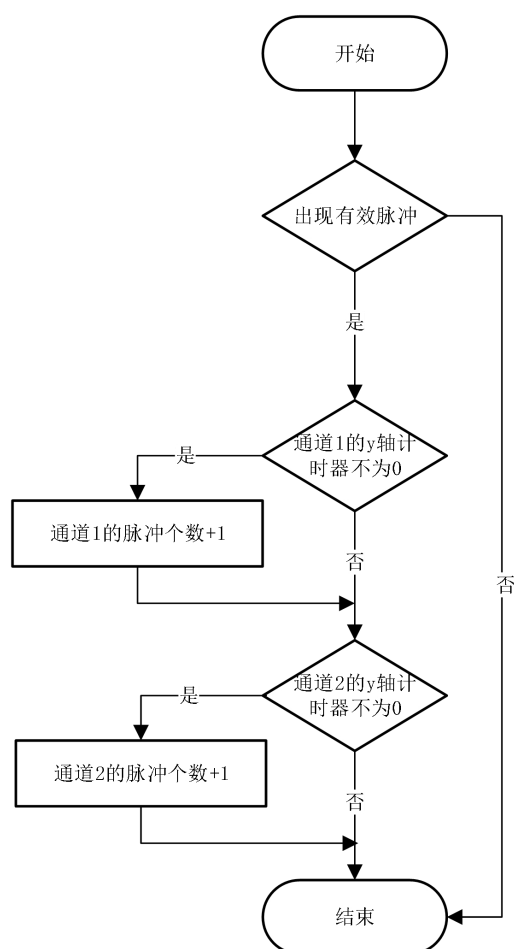


图 2.46 螺丝机脉冲判断图

2.5. 网关设计

本文将在每个加工设备上添加传感器进行监测，并通过网关对数据进行传输。本文针对网关设计具体做了以下设计，主要包括以下几个模块：MCU 处理模块、JS9331 处理模块、4G 通信模块、LED 模块、按键复位模块、RS232 模块、RS485 通信模块、串口隔离模块、电源模块、传感器。

MCU 处理模块为硬件系统核心，本文使用了包含主芯片 STM32F103RCT6 和晶振电路的 MCU。STM32F103RCT6 具有高达 72M 的处理速度，使得该集中器拥有过人的计算优势；以及具备 512KB 的 Flash ROM 和 64KB 的 SRAM，为整个系统提供更大的程序存储空间和运行空间。另外 STM32F103RCT6 还提供了 5 个串行通讯口 USART 和 3 个 SPI 通信口。该系统中用了 3 路串口，一路串口与 AR9331 进行交互，另外两个串口分别应用于 485 通信电路和 232 的通信电路中。

JS9331 处理模块主要包括 JS9331 核心板模块和以太网口电路。JS9331 处理模块通过串口与 MCU 处理模块相连，处理由 MCU 处理模块转发过来的数据，还可以通过 WIFI 或者以

太网与服务器平台建立连接,进行数据通信。同时连接两个 HR911102A 的水晶头接口,支持可通过局域网连接本地设备的有线 lan 口和通过广域网连接外部网络 wan 口。

电源模块包含两个 24V 转 5V 电源模块电源模块,24V 转 5V 电源模块 A 与 5V 转 3.3V 模块相连,该 5V 转 3.3V 模块提供 3.3V 输出电压给 MCU 处理模块和 4G 通信模块供电;24V 转 5V 电源模块 B 与 5V 转 3.3V 模块与 5V 转 3.3V 模块相连,该 5V 转 3.3V 模块提供 3.3V 输出电压给 JS9331 处理模块供电,该 5V 转 5V 模块提供 5V 输出电压给 RS232 通信模块和 RS485 通信模块供电。

其中网关结构图如下图所示:

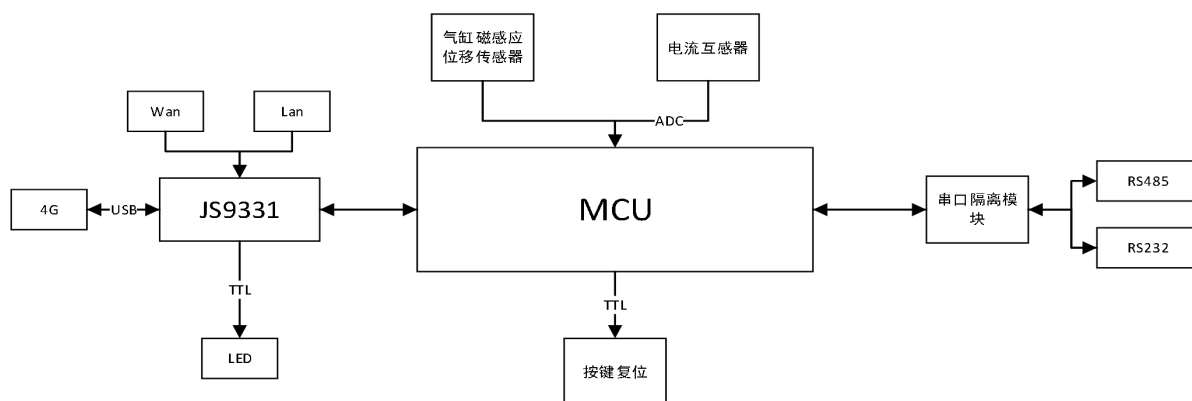


图 2.47 网关结构图

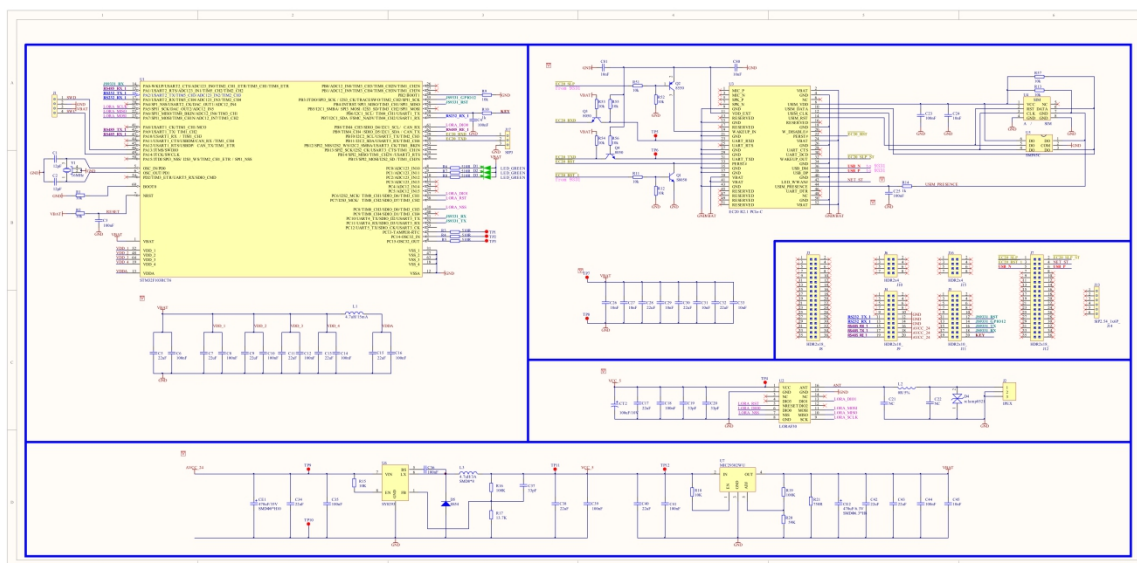


图 2.48 网关电路图



图 2.49 物联网网关实物图

2.6. 本章小结

本章通过企业当前痛点入手，对于目前传统制造业工厂生产设备管理进行数字化改革进行全面的需求分析，结合第一章提出的问题，介绍了基于云 MES 的仪表设备生产管理系统，通过整体架构的框架介绍和云平台、客户端、传感器、网关这四方面进行详细解析。本系统全面地考虑了不同职业身份和场景应用的使用需求，提供不同平台的多样化服务，帮助仪表盘工厂在制造仪表的过程中更方便、智能地进行管理。

第三章 基于早熟抑制的 PSO 算法任务调度设计

在当今制造业中，频繁出现的流水线堵塞、任务分配不科学等问题严重制约了我国传统制造业的生产效率。在当前仪表生产中，由负责人来人工分配任务。这对于目前正在快速扩张的工厂车间来说已经越来越力不从心，他们往往由于小小的决策失误、分配错误，会导致连锁性的反应。假设在仪表生产过程中，当负责人分配 A 车间完成压气操作，B、C、D 等车间负责后续不同操作。而此时 A 车间还在忙于上一个任务，那么将会直接导致 B、C、D 等多个车间闲置，整条生产链停滞，造成人工浪费、设备闲置以及产品延迟交工。

然而如何能够正确的在尽快短的时间内提供方案，使得多项产品的多道工序在有限车间内能以最快速度与效率完成。学者们提出了通过任务调度算法计算最优的方案。当确定需要完成几组工单中的任务次序以及不同车间在完成不同任务的时间这些参数时，那么就一定存在一个最优方案，可以通过该方案使车间在最短的时间内按次序完成这些任务。此时对车间的闲置时间最少。

如今已有许多国内外对于任务调度的研究，就例如 Min-min 算法^[18]和 Max-min 算法^[19]的传统研究算法，同时也提出了类似于模拟退火算法^[20]、轮询调度算法^[21]和蚁群调度算法^[22]的智能调度算法。其中 Min-min 算法在分配任务时从完成时间最快的任务开始，寻找完成此任务最快的设备。虽然实现简单，但是在具体应用过程中可能会导致效率不高的设备一直处于闲置状态。而 Max-min 算法在进行任务调度时与 Min-min 算法相似，但是从需要完成时间最多的任务开始。这两种方式都仅从当前出发，并未考虑全局情况下最高效的方案

在众多任务调度算法中，粒子群算法凭借着其卓越的表现脱颖而出。粒子群算法又名 PSO 算法，它通过鱼群、鸟群等群居生物的社会习性进行模拟^[23]，较为妥善地解决了连续解空间问题。粒子群算法首先将创建出一堆无序粒子，求出各个粒子的个体极值和全局极值，然后通过这两个值刷新粒子的位置和速度，再根据刷新的粒子计算当前情况下的个体极值和全局极值，依次循环，循环至次数达标或者满足精度要求，输出最终方案。该方案从全局考虑，并未局限于完成眼前任务的最优方案，一定程度上提高了任务调度算法的性能。

虽然 PSO 算法对于任务资源调度有很好的引导作用，但 PSO 算法仍然有一些不足之处：粒子群算法在循环计算最优解时并没有对速度按实时情况进行变动，这样静态的算法使得粒子群的收敛速度过快，若粒子群聚集在某个区域，并未平均分布在解空间时，容易被局部最优解所困住而无法挣脱，往往在迭代了许多次，也无法找寻到全局的最优解。这使得企业通过 PSO 算法来进行任务调度的时候，若初始化的粒子群质量不高，那么在迭代了很多次之后，企业所求的方案与算法求出的最优方案仍然有一定的差距。若使用该方案对任务进行调度，车间工作效率将受到限制。

本章通过三方面对 PSO 算法进行优化，来避免 PSO 算法提前陷入局部的现象。首先在初始化粒子群的时候，对每个创建的粒子通过熵进行筛选，以确保粒子群不会聚集于某一个区域，来获得一个质量较高的粒子群。其次在迭代过程中，对学习因子和惯性权重进行动态调整，使得在每次循环中，参数适合当时场景。最后会在粒子群适应度方差为 0 时判断此时粒子群有无陷入早熟情况，若过早陷入局部最优解，那么将通过混沌扰动的方式使粒子群跳出该区域，从全局中寻找最优解。

3.1. PSO 算法优化设计

3.1.1. 标准 PSO 算法概述

粒子群优化算法（PSO 算法）通过将鱼群、鸟群中的个体都视为只有位置和速度的粒子，每个粒子都没有重量和大小。然后模拟出每个群体中的粒子根据自己移动速度和所在的空间位置，判断出对于自己与集体最优的值作为个体极值。将该值与粒子群中的其余粒子进行分享，把最优方案中拥有极值的粒子视为全局最优解，以此实现出自适应的优化^[24]。这个思路是从鸟群和鱼群在集体行动的时候，会产生认为无法预料的突然分散、突然转弯的行为，但这个行为却由整个集体有序地一起完成，最终通过所有个体的一致同步，并未产生碰撞等现象，最终群体到达它们的最终目的地^[25]。这种现象是由于鸟群、鱼群中存在信息共享机制。

根据其原理，粒子群优化 PSO 算法通过以下几步进行资源调度优化：

- (1) 首先初始化每个粒子，形成随机粒子群。并将粒子群设为 N ，同时设定粒子的位置与速度为 X_i 、 V_i 。
- (2) 对每个粒子进行评估，求出每个粒子在粒子群里的适应度值 $F_{fitness}[i]$ 。
- (3) 将第二步获得的粒子适应度值 $F_{fitness}[i]$ 与该粒子针对个体与粒子群的个体最优解 $Pbest[i]$ 进行比较，如果 $F_{fitness}[i]$ 比 $Pbest[i]$ 更适合，那么将 $F_{fitness}[i]$ 的值赋予 $Pbest[i]$ 。
- (4) 将第二步获得的 $F_{fitness}[i]$ 与最新全局最优解 $Gbest$ 来比对，若 $F_{fitness}[i]$ 比全局极值 $Gbest$ 更合适，那么将 $F_{fitness}[i]$ 的值赋予 $Gbest$ 。
- (5) 确定好粒子极值 $Pbest[i]$ 与全局极值 $Gbest$ ，通过以下公式，将该粒子的速度与位置进行刷新重新赋值^[26]：

$$V_i^{t+1} = \omega V_i^t + c_1 * rand() * (pbest_i - x_i^t) + c_2 * rand() * (Gbest - x_i^t) \quad (3.1)$$

$$x_i^{t+1} = x_i^t + v_i^t \quad (3.2)$$

其中 ω 为惯性权重，若 ω 越大，则代表着该粒子的惯性越大，对于速度的改变就越缓慢；当 ω 越小，则代表着该粒子的惯性越小，此时粒子的速度更容易产生变化。 ω 的大小与适宜搜索区域的范围呈正比，且 ω 的范围在 0.2-1.2 之中。

c_1 为粒子的自我认知学习因子， c_2 为粒子的社会认知学习因子。其学习因子一般为 2^[27]。

$rand()$ 则通过随机数增强粒子群的随机性，通过 0 到 1 的随机均匀分布因素，使其在更新时提高种群多样性。

第一个式子中粒子通过惯性、自我学习、社会学习，更新自己的速度，进行下一步运动。第二个式子如图 3.1，表达了粒子在解空间中最优解位置的变换过程。

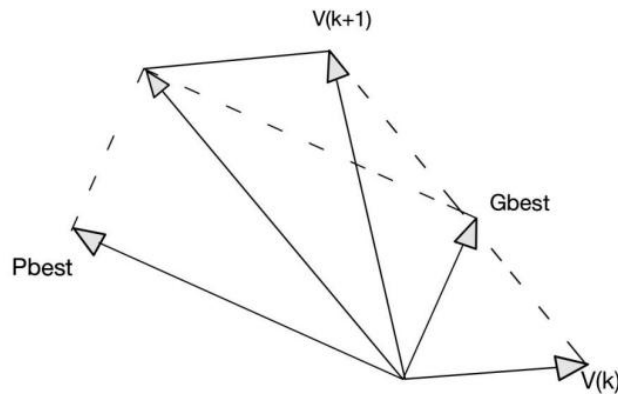


图 3.1 粒子最优解位置变换图

(6) 重复步骤二，直至达到算法设定的最大循环次数。

3.1.2. 标准 PSO 算法优缺点

粒子群优化算法对于研究大量任务在云平台进行资源任务调度有很多优势：

(1) 粒子群优化算法具有记忆性，其粒子的最优解可以通过粒子群的分享迭代传递给其他粒子。当其中一个粒子计算出群体最优解，那么所有粒子将会跟随群体的最优解进行调动，以此提高算法的效率。

(2) 算法原理通俗易懂，需要调整和涉及的参数较少，能够在算法实现的过程中进行操控，并具有很好的扩展性。

但是同时，我们也注意到了粒子群优化算法 PSO 在运用当中也存在着一些不足之处：

(1) 粒子群优化算法在计算迭代中缺乏速度的动态调节，由于收敛速度过快，容易只能获得局部的最优解，而忽略了搜索过程中潜在的更优解。

(2) 粒子群优化算法的效率依赖于随机出来的初始种群质量。若初始种群质量差，那么该算法的计算就需要更长的时间，且计算出来的最优解有可能仅是局部的最优解。初始种群质量差包括初始出来的粒子种群并未均匀散在解空间中，而是聚集在某一区域，或者这些粒子离最终结果的最优解较远时，粒子群通常获取不到真正的最优解。

3.1.3. 基于早熟抑制的 PSO 算法优化设计

3.1.3.1. 初始化粒子群优化

对于粒子群优化算法是否能快速寻找到全局最优解这个问题，关键依赖于初始种群的分布情况，若初始出来的粒子群过于密集，那么计算出来的最优解极大可能仅仅是粒子群聚集区的区域最优解。针对如何寻找全局最优解而不陷入局部这个问题，我们将在对粒子群初始化时进行优化^[28]，通过粒子熵对初始化的粒子进行判断，使粒子群初始化后尽量平均分散在

解空间中。具体操作如下所示：

(1) 首先分别对临界熵 S_0 、粒子群规模 M 与粒子数目 k 进行赋值，其中 k 初始值为 1。

(2) 生成第 k 个粒子，根据下方表达式计算粒子 n 与粒子 k 在第 i 维相同的概率和第 k 个粒子的第 i 维熵。其表达式如下所示，其中假设已经初始化了 $k-1$ 个满足条件的粒子，计算出来的 S_i 是第 k 个粒子的第 i 维熵， P_{nk}^i 代表了粒子 n 与粒子 k 在第 i 维相同的概率。 $D_{\max}^i$ 代表者粒子在第 i 维的上限， $D_{\min}^i$ 代表者粒子在第 i 维的下限。 y_n^i 代表了粒子 n 在第 i 维的值， y_k^i 代表了粒子 k 在第 i 维的值^[29]。

$$P_{nk}^i = 1 - \frac{|y_n^i - y_k^i|}{|D_{\max}^i - D_{\min}^i|} \quad (3.3)$$

$$S_i = \frac{1}{k-1} \sum_{n=1}^{k-1} (-\ln P_{nk}^i) \quad (3.4)$$

(3) 若 $S_i > S_0$ ，则将第 k 个粒子的第 i 维保留，反之，则重新对第 k 个粒子的第 i 维进行初始化。

(4) 重复步骤 3，直到初始化出粒子 k 的所有分量。

(5) 重复步骤 2、步骤 3、步骤 4，直至达到粒子群规模 M 。

通过对粒子分量维度的熵作为依据进行评判，使初始化出来的粒子群尽可能地分布，而不是聚集在某一区域，使后续进行计算的时候有更大的可能以及更快的效率寻找到全局最优解。

3.1.3.2. 学习因子与惯性权重的自适应动态调整

研究指出惯性权重 ω 与学习因子 c_1 、 c_2 对算法的性能有着至关重要的作用^[30]。由上文可知惯性权重 ω 在 PSO 的运算过程中对于搜索到什么范围的最优解至关重要，在计算初始， ω 越大则越适合应用于全局搜索场景，在计算中后期， ω 较低可以完善局部的精细搜索。然而在一般粒子群优化算法中， ω 从未发生过变化，并未随着迭代实时对场景进行适应，致使粒子群多样性丧失，算法由于忽略了全局而陷入了局部寻优。

因此为了更好地平衡从初始化到最终值这个过程，协调这个过程中局部精确与全局寻优两者的平衡，可以将惯性权重 ω 与迭代次数和粒子群的多样性进行关联。在计算过程中，实时地根据迭代次数和搜索时间来线性动态地对惯性权重 ω 进行优化，让惯性权重 ω 通过算法往最优解的位置偏移。关于惯性权重的具体算法如下式所示：

$$\omega = \omega_{\text{start}} + (\omega_{\text{end}} - \omega_{\text{start}}) * \frac{t}{T_{\max}} \quad (3.5)$$

其中 ω_{start} 代表着惯性权重 ω 最开始的值， ω_{end} 则代表着惯性权重 ω 的最终值， t 为目前为

止的计算时间， $T_{\max}$ 则代表着算法迭代的最多次数，根据计算研究，当 ω 最开始为 0.9，最终值为 0.4 时，采用了这种线性动态变化的惯性权重 ω ，比原先的标准粒子群优化算法在优化精度上有了明显的提升。

但这种惯性权重线性变化也具有一定的缺陷，它也会过早地陷入局部最优，为了继续进行优化，针对惯性权重 ω ，提出了更适合这种场景的惯性权重非线性动态变化策略^[31]：

$$\omega(t) = \omega_{\text{end}} + (\omega_{\text{end}} - \omega_{\text{start}})^{\exp(-k * (\frac{t}{t_{\max}})^2)} \quad (3.6)$$

其中 k 是控制因数，通过 k 的大小控制 ω 和 t 两者曲线的平滑度，通过计算研究得出，当 k 为 3 时权重变化较适合，惯性权重呈现先凸后凹的曲线，使粒子群算法在前期可以使 ω 加速度降低，从而较快进行区域搜寻，在后期惯性权重的加速度增加，令粒子群不会提前陷入局部最优解无法获得全局极值。

学习因子的变化需求与惯性权重相似，都是在算法计算过程的前期期望粒子群能够在解空间中尽可能大地探索，在计算过程的后期尽量避免提前拘于区域的状态。其中 c_1 、 c_2 分别代表了粒子的自我认知和社会认知的程度。对于不同的学习因子，本文提出了不同的处理方式： c_1 先增加后减少， c_2 先减少后增加。这两种处理方式意味着粒子在计算过程中在前期需要更注重自身本身的历史信息，同时在后期则更需要注重全局的粒子信息。学习因子的变化方式具体如下面两个式子所示：

$$c_1 = c_{1k} + (c_{1r} - c_{1k}) * \frac{t}{t_{\max}} \quad (3.7)$$

$$c_2 = c_{2k} + (c_{2r} - c_{2k}) * \frac{t}{t_{\max}} \quad (3.8)$$

其中， c_{1k} 为粒子自我认知 c_1 的初始值， c_{1r} 为粒子自我认知 c_1 的最终值。 c_{2k} 为粒子社会认知 c_2 的初始值， c_{2r} 为粒子社会认知 c_2 的最终值， t 为目前为止的计算时间。根据迭代次数对 c_1 、 c_2 与 ω 进行更新，可以保证在计算过程中能够尽量满足每一个时期不同的要求，保证粒子群算法更大可能性地计算出整个解空间内的最优解。

3.1.3.3. 早熟抑制策略

通过学习因子和惯性权重的动态变化可以使粒子群优化算法在初始化出质量不好的粒子群时减少提前被局部解所拘泥的可能性，从而提高快速寻找到真正最优解的性能。粒子群在计算过程中是粒子群中的最优粒子会对其他粒子进行带领引导，致使所有粒子朝着寻找到的最优解进行聚集。若全局的最优解处于粒子群周围，易被粒子寻找到，那么在计算过程中就有更大可能聚集在全局最优位置周围。但如果粒子群初始化偏聚集化某一区域，但是全局的最优解却处于另一个区域，那么在计算过程中，粒子群寻找到的最优解仅仅会在某一区域的

最优解，并且难以逃出这个区域，这便是粒子群早熟的原因^[32]。

对于粒子群在迭代过程中过早陷入局部最优解的早熟情况，本文将通过混沌扰动打破粒子群陷入僵局的情况。

首先会对粒子群是否过早进行局部早熟收敛进行判断，若停滞于某个局部极值陷入了局部最优，那么将对其进行早熟处理。本文将通过种群的粒子分布范围进行判断，若种群全部粒子都聚集在某一区域，那么便可以判断过早收敛。因为粒子群中的个体的位置可以体现了它的适应度，对粒子群内的粒子们的适应度值进行全程的监控，可以判断它们的相对位置是否过早发生了聚集，以此判断有无提前陷入了局部最优。具体判断方式如下所示^[33]：

$$f = \begin{cases} \max|f_i - \bar{f}| & \max|f_i - \bar{f}| > 1 \\ 1 & \text{其他} \end{cases} \quad (3.9)$$

设粒子群规模为 M ， F_i 为第 i 个粒子的适应度值， $\bar{f}$ 为粒子群全体粒子的平均适应度， σ^2 为粒子群的适应度值方差。通过归一化定标因子 f 对粒子群适应度方差 σ^2 进行大小取值约束。具体如下式所示^[34]：

$$\sigma^2 = \sum_{i=1}^M \frac{f_i - \bar{f}}{f} \quad (3.10)$$

σ^2 的大小体现了粒子群的分布情况， σ^2 越大，说明该粒子群此刻越分散，若 σ^2 越小，说明该粒子群越聚集，所有粒子的适应度越接近。当 σ^2 为 0 时，则意味着粒子群提前进入早熟状态或者达到了全局收敛。因此仅凭借为了粒子群适应度方差 σ^2 是否为 0，无法进行准确判断，因此为了区分两者，将会通过进一步计算，判断算法此时寻找到的值是否为粒子群理论上的最优适应度，如果该值并不是全局最优值，那么便可判断当前粒子群已经陷入了早熟状态^[35]。

当粒子群陷入了早熟局部最优，那么仅仅凭借粒子本身的计算和运动是无法跳出该区域的，此时就必须依靠外界的力量使粒子群跳出该区域，进入其他区域继续进行寻找。面对该情况，通过基于立方映射的混沌扰动机制使陷入了早熟的粒子群跳出局部最优值。具体式子如下所示，对于 $y(k)$ 有着 $[-1, 1]$ 的约束：

$$y(k + 1) = 4y(k)^3 - 3y(k) \quad (3.11)$$

该混沌机制在实际应用中，如果初始值不为 0，那么就会一直产生混沌。该混沌抑制早熟机制具体思路如下所示：

(1) 随机生成一个 d 维向量，并将其赋值给粒子群的第一个粒子，其中该向量的每个分量在 -1 与 1 之间取值。

(2) 为后续替换粒子群中的粒子提前设定一个数值，本文将取值为粒子群中粒子个数的百分之六十。

(3) 通过上式对粒子进行 $r-1$ 迭代, 并获得 r 个新粒子, 其中所生成的新粒子向量的每个分量同样将介于 -1 与 1 之间。新粒子将通过下式映射到粒子群的搜索解空间中, 其中 $x_{\max}$ 代表了粒子在每个维度的上界, $x_{\min}$ 代表了粒子在每个维度的下界, $y_i(d)$ 代表通过立方映射的混沌机制生成的第 i 个新粒子的第 d 维度, x_{id} 代表了在解空间中第 i 个粒子的第 d 维度^[36]。其表达式如下所示:

$$x_{id} = x_{\min} + (1 + y_i(d)) * \frac{x_{\max} - x_{\min}}{2} \quad (3.12)$$

(4) 对已经陷入早熟的粒子群中的所有粒子凭借适应度值进行排序, 并且用上一步产生的 r 个新粒子代替排序后的最后 r 个粒子。

(5) 此时种群获得了新的粒子, 新粒子有能力指引粒子群跳出局限的区域, 在这过程中, 学习因子和惯性权重都应根据迭代进行动态化变动。

以上步骤可以帮助粒子群在陷入早熟收敛时, 及时通过混沌机制生成新的粒子, 并且凭借那些新粒子跳出局限的区域, 使粒子群算法寻找到全局最优解。

3.1.4. PSO 算法优化流程

为了优化粒子群的资源调度, 本文通过粒子群初始化的优化、学习因子和惯性权重^[37]的实时变动以及早熟抑制策略三方面来优化粒子群在计算过程中能更快地寻找到正确的全局最优解。具体流程如下方流程图所示:

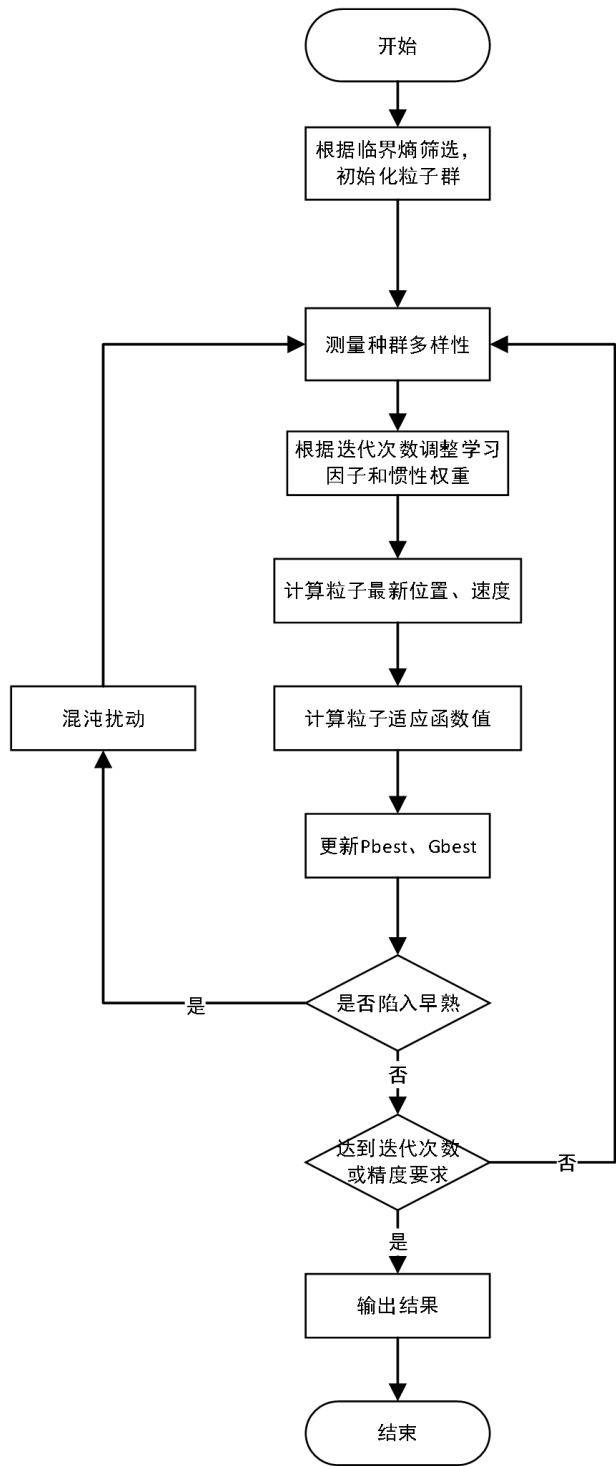


图 3.2 PSO 算法优化流程图

3.2. 测试分析

本文将根据上述 PSO 优化流程对任务资源调度进行优化，为了验证 PSO 的性能，将车间数视为资源数，并对此进行不同处理，并从收敛程度、处理速度等方面进行对比，最后将通过实例进行测试。

3.2.1. 仿真测试环境准备

为了更清楚地了解粒子群算法优化，我们将通过 CloudSim 仿真平台进行测试数据的采集。本文操作系统为 win10，处理器为 Inter(R)Core(TM)i7-6700 2.6GHz。通过 eclipse 运行 CloudSim，进行对粒子群算法优化的测试。

3.2.2. 不同算法处理时间对比

当资源数设定为 5 确定不变时，对 Min-Min 算法、Max-Min 算法和本文优化后的 PSO 算法在不同任务数的情况下处理的时间进行对比。

	Min-Min 算法	Max-Min 算法	优化后的 PSO 算法
任务数目为 100	36112.76s	34975.16s	34202.12s
任务数目为 200	71471.92s	69932.37s	68024.42s
任务数目为 300	102349.58s	104743.63s	100535.74s
任务数目为 400	138769.43s	137314.21s	135831.34s
任务数目为 500	172531.38s	171536.83s	168392.01s

表 3.1 各算法在不同任务数目下处理时间

从上表可以看出优化后的 PSO 算法与其余两个传统调度算法在使用过程中的对比。在资源数不多的情况下，任务数目逐步递增，优化后的 PSO 算法在处理时间上一直优于 Min-Min 算法、Max-Min 算法。

3.2.3. PSO 算法优化性能对比

下表为 PSO 的相关参数设定：

粒子群规模 M	100
维度 d	20
惯性权重 ω	优化前设定为 0.9，优化后根据迭代次数进行调整
次数	10
$X_{\max}$	30
$X_{\min}$	-30
迭代次数	1200
临界熵 S_0	0.3
学习因子	优化前设定为 2，优化后根据迭代次数进行调整
适应度函数	$f = \sum_{i=1}^{30} (x_i^2 - 10\cos(2\pi x_i) + 10)$

表 3.2 PSO 参数设定

根据上方的参数，对优化前后的 PSO 算法进行最优解性能对比。

算法	优化前 PSO 算法	优化后 PSO 算法
最优值	7.1254	4.2158
最差值	33.2156	21.5412
平均值	21.1423	11.3451
迭代次数	684	452

表 3.3 PSO 优化前后对比

由上方数据可得知在优化后 PSO 算法迭代次数明显少于原先的 PSO 算法，对迭代次数以及最优值的性能都大大优化。其中随着迭代次数的增加，其适应度根据上方函数变化如图 3.3 所示。

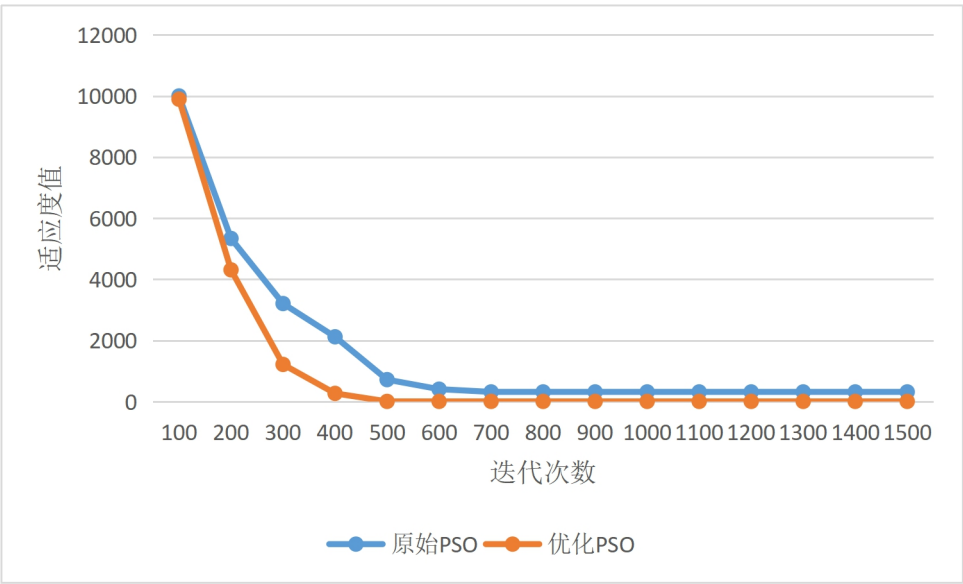


图 3.3 PSO 算法优化前后适应度对比 PSO 算法优化前后适应度对比

由上图可得。优化后粒子群算法收敛的程度与效率都比原先的粒子群算法更完善，同时优化后寻找到的最优值也更优于原先寻找到的最优值，在更少的迭代次数内找到更优解。

通过控制变量的方式，当查找资源数与总体任务数分别产生变化时，观察两种算法对于资源调度处理速度的影响。设定平台资源数为 10，并保持不变时，任务数量对性能也有着不同的变化。具体如下图所示，其中横坐标为任务数量，纵坐标为处理时间。

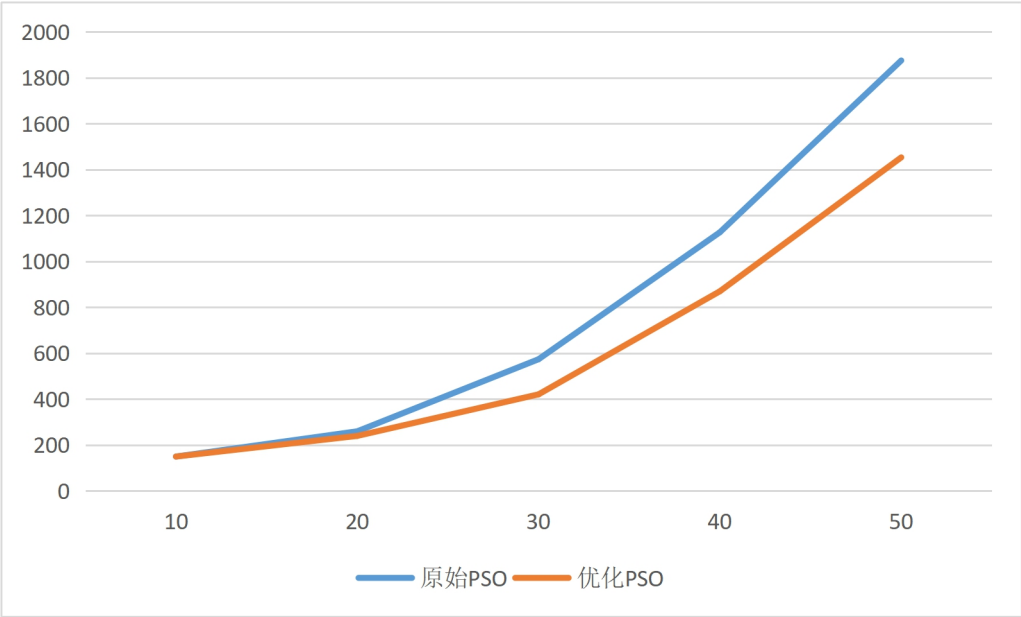


图 3.4 任务数量不同时 PSO 优化前后的处理时间（ms）对比

当总体任务数为 100 并保持不变时，资源数量对性能也有着不同的变化。具体如下图所示，其中横坐标为资源数量，纵坐标为处理时间。

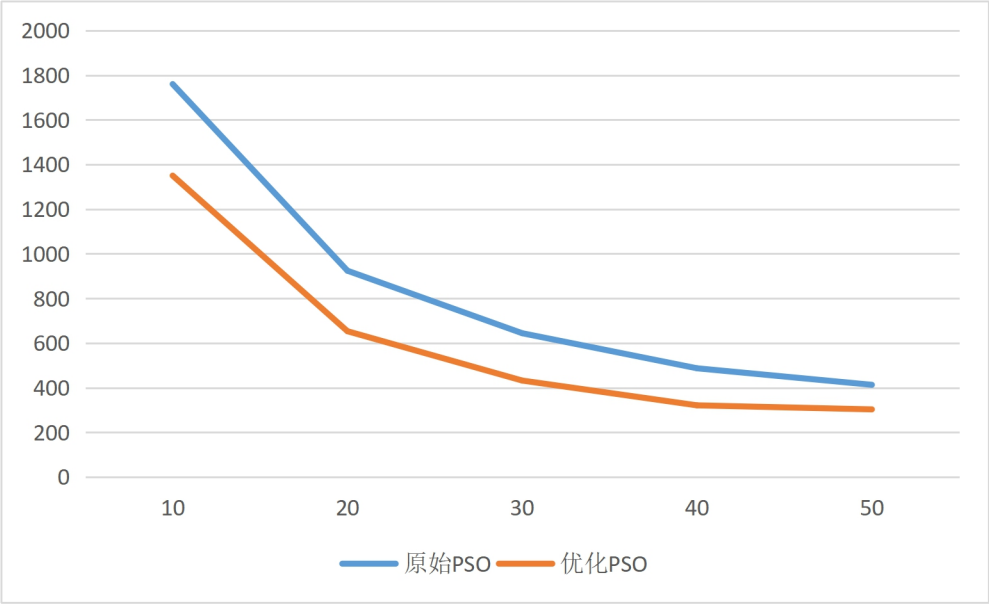


图 3.5 资源数量不同时 PSO 优化前后的处理时间（ms）对比

3.2.4. 任务分配调度实例测试

为了更贴切本文的仪表产品生产过程任务调度优化，本文还通过实例进行测试，观察调度结果。下表为各个仪表在不同车间在处理不同工序所需要的时间。

种类	工序	车间 1	车间 2	车间 3	车间 4	车间 5	车间 6
仪表 A	压气 A1	2	3	4	-	-	-

	压针 A2	-	3	-	2	4	-
	注油 A3	1	4	5	-	-	-
仪表 B	压气 B1	3	-	5	-	2	-
	压针 B2	4	3	-	-	6	-
	注油 B3	-	-	4	-	7	11
仪表 C	压气 C1	5	6	-	-	-	-
	压针 C2	-	4	-	3	5	-
	注油 C3	-	-	13	-	9	12
仪表 D	压气 D1	9	-	7	9	-	-
	压针 D2	-	6	-	4	-	5
	注油 D3	1	-	3	-	-	3

表 3.4 实际车间调度实例

根据上表的车间与仪表的处理时间，其最优解如下图所示：

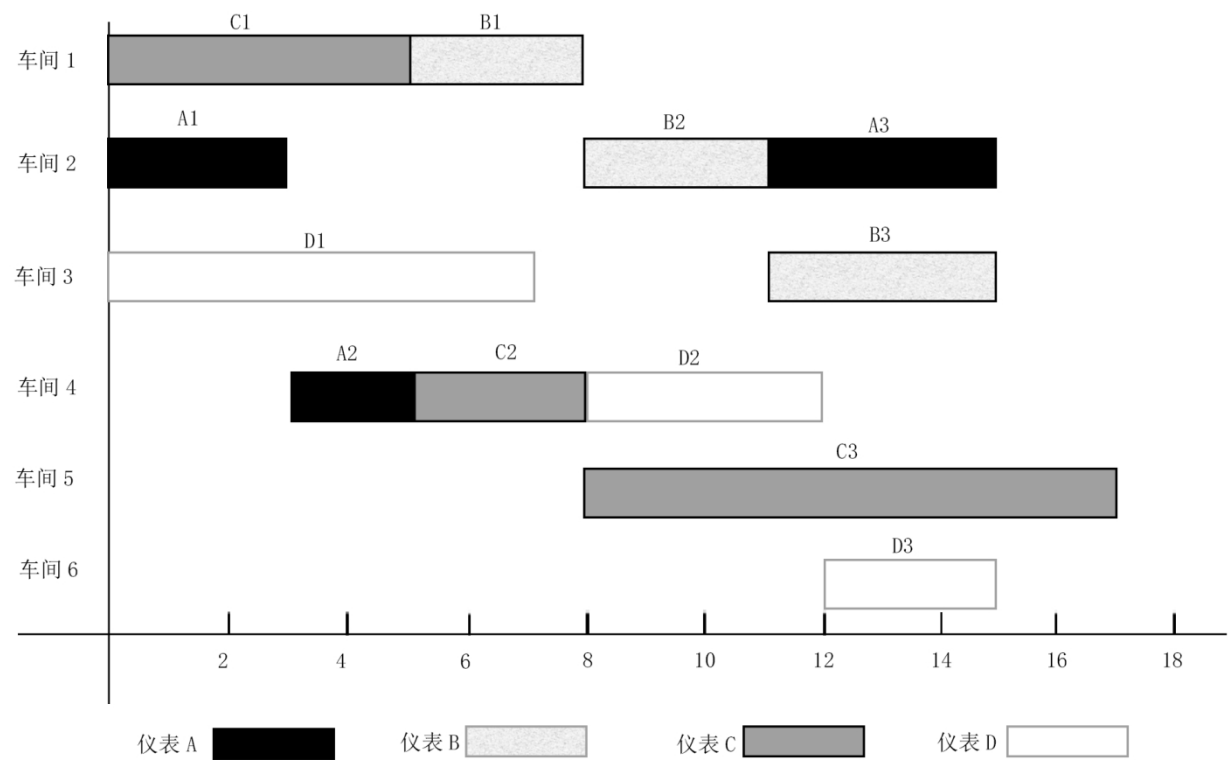


图 3.6 最优解甘特图

对上方任务通过本章优化后的 PSO 算法进行调度，最大迭代次数设定为 100，粒子群规模设定为 100。那么测试得出其适应度随着迭代次数如下图所变化。

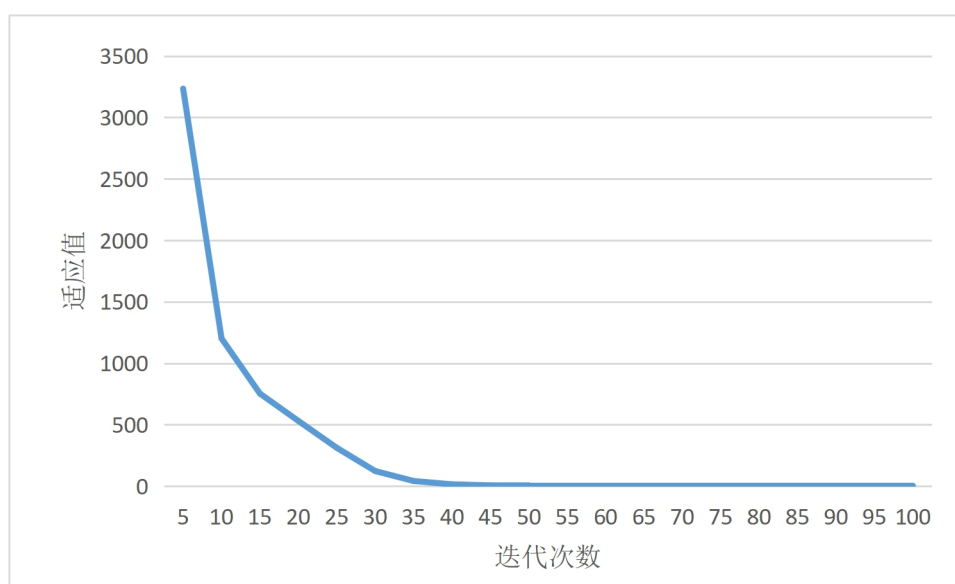


图 3.7 适应度趋势

根据测试, PSO 优化算法在第 50 次左右趋于收敛获得全局最优解, 且其计算出来的最优解与该问题最优解接近。可以很好地应用于实际之中, 对任务进行分配调度。经过多次试验, 求出最优解的适应度于 3-5 之间上下浮动, 证明该算法求出的最优解性能稳定。

3.2.5. 测试分析

根据测试将 PSO 算法与 Min-Min 算法、Max-min 算法进行对比, 在资源数一定, 任务数目不变的情况下, 优化后的 PSO 算法处理时间皆比另外两种算法要快, 三者在不同任务数目下的处理时间的比例相近。

确定相关参数, 将 PSO 算法优化前后算法性能进行对比, 根据测试数据可得, 优化后的 PSO 算法可在更少的迭代次数中求得更为接近的最优解, 其迭代次数较优化前减少了 34%。这有利于在车间有限的情况下更快速度下求出效率更高的任务调度方案, 为物联网云 MES 系统任务调度提供更好的服务。

本章还根据实例进行任务调度测试, 在四种仪表生产的过程中, 存在一个最优解使得车间的工作效率最高。PSO 优化算法经过第 50 次左右的迭代获得一个极为接近的最优解, 且经过多次实验, 其最优解变动幅度不大, 具有稳定性, 使车间运作时减少出现工作效率忽高忽低的不稳定情况。

根据本章多次测试可得出, 优化后的 PSO 算法使用更少的迭代次数计算出更接近于实际最优解的方案, 证明了该算法已经具有早熟抑制的性能, 且不再依赖初始化粒子群的质量, 具有稳定性。

3.3. 本章小结

本章介绍了在 MES 系统中任务调度的紧迫性以及当前的现状, 并针对该现状提出了 PSO 粒子群算法, 介绍了 PSO 算法的原理以及具体步骤, 分析 PSO 算法的优势与缺点。根据 PSO

算法的不足之处从以下三方面进行优化：在创建粒子群时根据熵进行筛选；学习因子与惯性权重动态调整；在粒子群迭代过程中过早陷入局部时判断早熟，并通过混沌扰动跳出局部。测试表明优化后的 PSO 与 Min-Min 算法、Max-Min 算法、原始 PSO 算法比较都有很大的提高，优化后迭代次数约减少了 34%，具有很好的使用效果，并可用于实地。

第四章 基于国密标准的 IPSec VPN 加密优化设计

随着互联网在制造业结构的逐步深度使用，设备、人工等与数据的结合让网络连接尤为重要，通过数据信息的传递，为国家、企业和个人带来了全新的发展前景。云 MES 通过生产过程管理优化使得企业生产效率大幅度提高。但同时也由于物联网较互联网与人类的生活交互更加密切，实实在在地关联着我们的财产，MES 系统更是能直接获取与我国制造业生产商业机密数据与整个企业核心制造流程，因此我国物联网对于数据传输安全，有了较之前更高的标准。如何部署一个成本亲民但是安全性高的网络安全架构，是云 MES 系统生产的尤为重要的一个部分。

目前云平台在远程数据包的传输中，存在以下几个安全问题：

（一）中间人攻击

中间人攻击是通过 DNS 欺骗、秘钥泄露等方式进行来拦截原本正常的传输通信数据包，并对其数据进行篡改，冒充通话，而通信双方却无法发现，以此达到实现所谓的 MITM 攻击。现在 IPSec 协议往往采用 RFC 标准下的 IKE 协议，该协议采取预共享密钥的方式对数据包进行认证，该认证方式易遭受攻击，安全保证不高，易被中间人盗取密钥冒充。

（二）算法非国内技术

当今国内使用的大部分加密算法、平台、软件等都是国外引进的核心创新技术。这对于国内安全来说相当于被外国扼住喉咙。所以推广我国开发的相关技术，掌握真正的技术核心是当今国内的技术方向。国家密码局因此提出了其自主研发的相关加密算法，为国内数据传输安全提供了保障^[38]。

（三）数据包处理的灵活性不足

当前云平台在传输时对于网络数据包的处理并未加以区分，全部进行加密，并未考虑使用中是否需要，提供差异性服务，这无疑提高了服务器系统的成本压力。

MES 系统中传统的传输形式不仅加密效果差，易被攻击，其传输与加密解密的速率都不如人意。因此本章将会针对以上问题对于当前的云平台的数据传输安全与效率进行优化。本章采用 IPSec 协议进行数据的传输。IPSec 协议是由 IETF 发布的保护数据传输过程安全的一种三层隧道安全加密协议技术^[39]。通过对数据完整性的验证和身份验证，在一定程度上保证了传输过程中的安全性^[40]。但是长时间的加密传输，会导致 CPU 和服务器资源压力过大，其效率逐渐下降，持久性弱^[41]。同时，将 IPSec 体系中所采用的主流算法与国密算法进行比较，国密算法不仅更符合我国国情，其加密性能也更胜一筹。

据此，本章提出基于国密算法的 IPsec 框架的实现。通过将 IPsec 协议内的主流算法更换为国密标准下的 SM2、SM3、SM4 算法，再加以硬件加密卡的辅助，解放服务器压力，来达

到更深层次的安全保障以及更高的加密解密效率。

为了更好地实现 IPsec VPN 与国密算法的结合，本章从 IKE 协商层设计与服务器内核设计两方面进行优化。在 IKE 协商过程中通过对传输的数据包进行加密、签名以及数字证书认证多种方式结合，使其能更安全地验证双方身份，提供安全保障。服务器内核中添加硬件加密卡，使 Linux 内核算法库可以通过硬件加密卡调用加密算法，以便后续数据传输时根据安全策略库和安全联盟库对数据包进行对应的加密解密。让 MES 系统为传统制造业数字化管理转型提供更加安全快速和稳定的传输途径。

4.1. IPsec 概述及缺陷

IPsec 协议通过对 IP 层的数据拆解并进行加密最后重新封装的方式在网络层发挥作用^[42]。它是通过网络层的主机与网关实现点到点的安全通信隧道，在此基础上实现 VPN，搭建双方物理子网构建的虚拟局域网。

IPsec 协议是由因特网密钥交换协议 IKE^[42]以及包含 ESP 协议和 AH 协议的保护分组流协议组成。ESP 协议负责对分组流进行加密，IKE 协议负责保障密钥交换，AH 协议负责对数据进行认证^[43]。具体架构体系如图 4.1 所示：

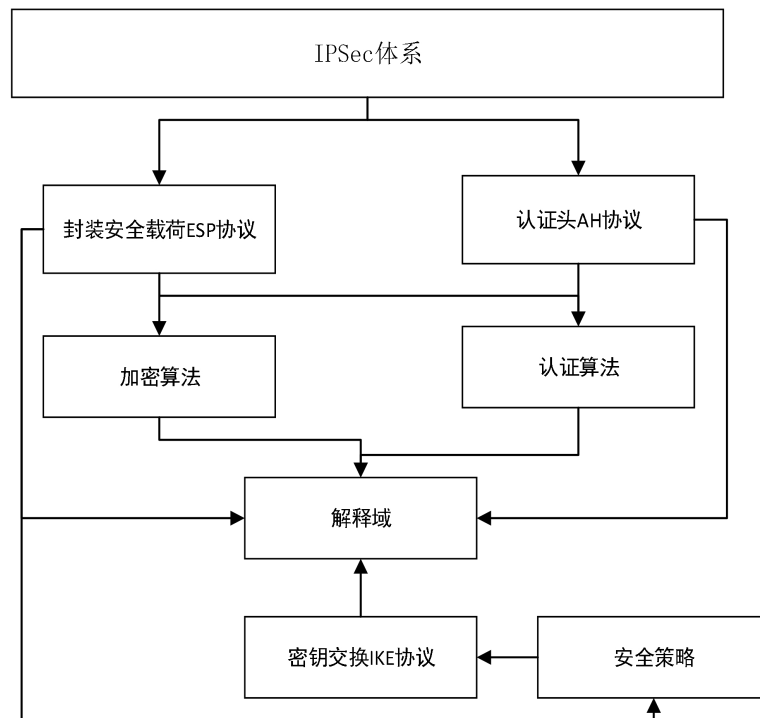


图 4.1 IPsec 体系架构图

但是 IPsec VPN 在长时间的加密传输中，会导致 CPU 和服务器资源压力过大，其效率逐渐下降，难以长久保持高效率的数据传输。而工厂需要长时间监控设备，无法满足工厂长期稳定的设备数据传输。以此针对如何减轻服务器压力，加强传输的稳定性，提高传输过程中的安全性这些问题，本章采用基于国密标准的 IPsec VPN，并添加硬件加密卡的方式调用国密

算法的方式来解决。

4.2. 基于国密标准的 IPSec VPN 优化设计

4.2.1. 基于国密标准的 IKE 协商层设计

IPSec 默认采用 RFC 标准下的 IKE 协议^[44]，随着时代发展，IPSec 的缺陷却逐日凸显出来，由此将根据 IPSec 于 IKE 协商层的缺陷^[45]进行优化。

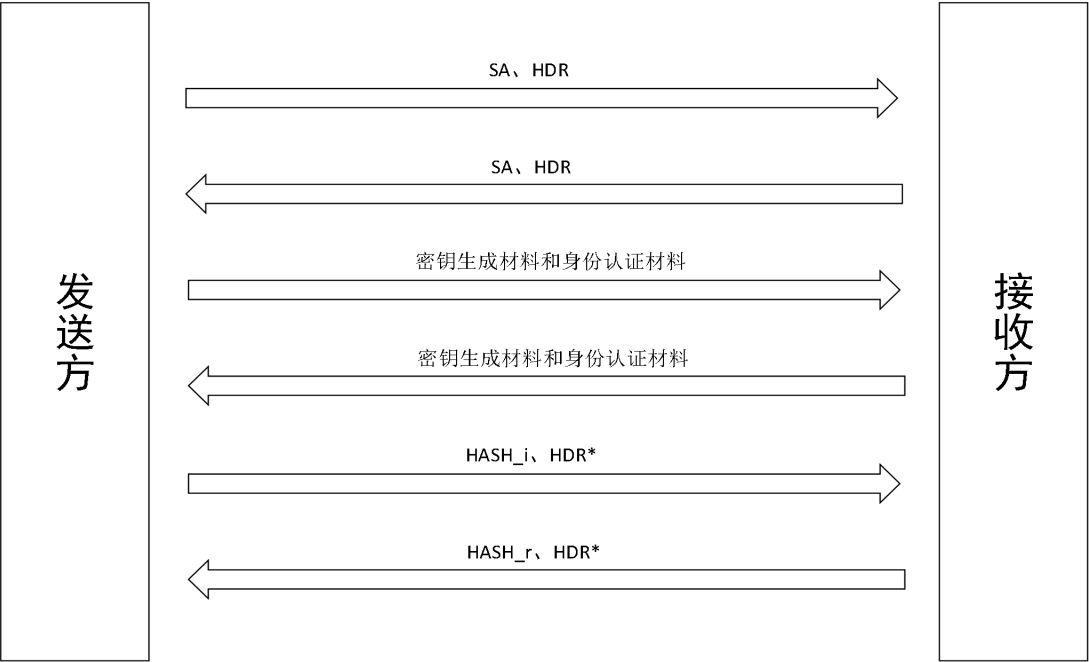


图 4.2 RFC 标准的 IKE 协商流程图

基于 RFC 标准的 IKE 协商流程具体如上图所示，通过三对数据包的交流，来进行通信加密解密^[46]。

（一）首先发送方和接收方将会互相交换第一对数据包，其中包含了 SA 和 HDR 载荷头。

（二）其次进行交互的第二对数据包包括基于 DH 算法的密钥生成材料和身份认证材料。DH 算法可确保通讯双方在公开不安全网络共享密钥。发送方与接收方通过 DH 算法的公开信息生成相同的密钥，再通过预共享密钥与两个 Nonce（只使用一次的随机数值）获得 skeyID，以用于后续的 skeyID_e 等计算。

（三）最后通过交互的第三对数据包来对身份进行认证，通过上一步的 skeyID_e 生成密钥，并通过预共享密钥的方式对数据包进行认证。然而该身份认证方式由于扩展性差，同时密钥的安全性也不高，所以易遭受攻击，安全保证不高。

国密算法^[47]将针对上述安全问题对 IKE 协商层协议进行优化。使用权威部门颁发的数字证书来保障权威性和安全性。它通过对传输的数据包进行加密、数字签名以及双证身份认证来确保其数据包的数据完整性和传输安全性^[48]。在本文的研究中，国密标准采用了使用 X.509 证书格式的国密 SM2^[49] 算法签名的数字证书，通信双方都将用数字证书对对方的身份进行

验证，以确保不是受到中间人攻击。

4.2.1.1. 基于国密标准的 IKE 载荷设计

在国密标准下的 IKE 协商层中所传输的数据包载荷都有着严格的标准要求，下文将对协商过程中传输的主要载荷格式进行分析^[50]。

1、消息头载荷

在消息传输中，消息载荷中包含一个定长的消息头载荷和多个不定长的其余载荷。消息头载荷中需要包含连接状态以及其他对于载荷处理所依赖的信息。

表 4.1 消息头载荷格式表

发送方 cookie			
接收方 cookie			
下一个载荷	版本号	交换类型	标志
消息 ID			
长度			

2、安全联盟 SA 载荷

安全联盟载荷是用于协商时确定接下来所需的安全服务类型。

表 4.2 SA 载荷格式表

下一个载荷	保留	载荷长度
解释域（DOI）		
情形		

3、建议载荷

建议载荷用于双方在密钥交换过程中，标识发送方选择的安全协议和 SA 的相关安全机制。

表 4.3 建议载荷格式表

下一个载荷	保留	载荷长度	
建议号	协议 ID	SPI 长度	变换数
变长的 SPI			

4、变换载荷

接收方通过变换载荷获取发送方对于指定的协议所提出的不同的安全机制。

表 4.4 变换载荷格式表

下一个载荷	保留	载荷长度
建议号	协议号	保留 2
SA 属性		

4.2.1.2. 基于国密标准的 IKE 流程设计

在国密标准下，身份认证主要分为两个阶段^[51]：第一阶段的主模式和第二阶段的快速模式。在第一阶段，双方互相交换彼此的密钥，并进行身份认证。第二阶段，双方在协商过程中使用第一阶段的密钥对网络数据包进行加密保护。

第一阶段的主模式具体操作如图 4.3 所示。



图 4.3 国密标准 IKE 协商主模式流程图

在国密标准下的 IKE 协商过程双方要互相交换三对消息，其消息的具体内容和作用如下所示：

（一）发送方首先向接受方发送安全联盟 SA 载荷，SA 载荷中包含了建议载荷，向接收方提出安全联盟 SA 的候选项建议。接受方收到消息后，将 SA 载荷、相关证书载荷一同发送给发送方。表明自己接受发送方的建议。

（二）其次双方开始交换自己的信息，信息包括身份标识、Nonce 等，在传输时其信息数据由密钥 Sk 进行加密，同时密钥 Sk 由加密证书进行加密，并由签名证书对数据包发送方的身份进行签名，以保证接收方能确定发送方的身份。这两个证书也会一起传输。在交换完信息后，通过接收到的数据计算生成 skeyID，作为基本密钥参数。并根据 skeyID 计算获得后续参数 skeyID_e。

（三）最后双方将对之前的协商过程进行验证，通过之前协商的密钥为 skeyID_e 的对称加密算法对这两条消息进行加密。

其中该过程中使用的数字证书是由 SM2 算法进行加密，通过权威部门的颁发来提高身份验证安全性。SM2 椭圆曲线算法是由国密局提出的一个基于 ECC 椭圆曲线密码学扩展的商用非对称加密算法，具有自主知识产权^[52]。

SM2 算法较于 RSA 加密算法更为安全。RSA 核心掌握在美国方，具有在算法中加入后门的危险，在当今贸易战的世界形势下，使用国家密码局自主研发的 SM2 算法对于我国传输更加安全。除此之外，SM2 算法可凭借国密局自主研发的硬件加密卡来加密数据，通过这种方式来增加我国的加密算法安全性。不仅站在国家战略安全高度来考虑，目前 RSA 非对称算法已经并不是无懈可击，在算法可行性上，已可通过最新发明的秀尔算法进行暴力破解，由此找到 RSA 算法的密码^[53]。

第二阶段的快速模式具体如图 4.4 所示：

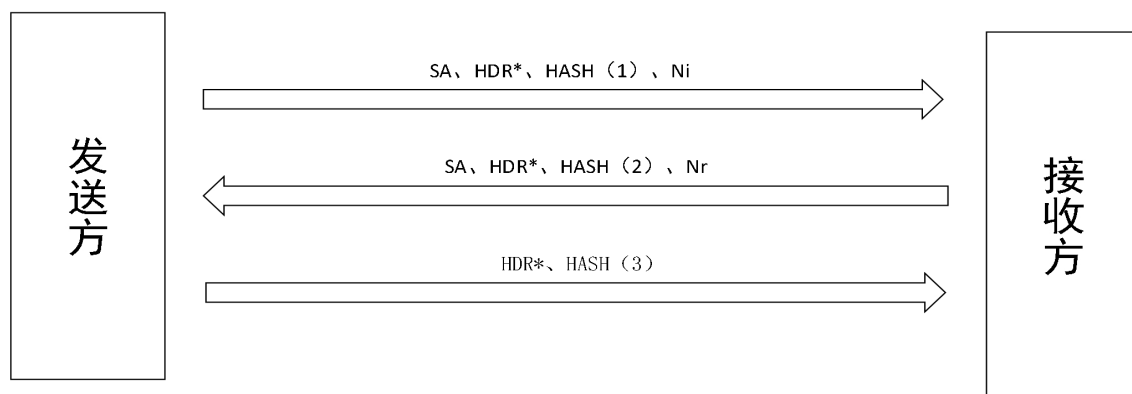


图 4.4 国密标准 IKE 协商快速模式流程图

在国密标准下的 IKE 协商快速过程中，双方要互相发送三条数据包，其数据包的具体内容和作用如下所示：

- （一）发送方向接收方发送一个数据包，数据包中包括 SA 载荷、Nonce 和标识载荷。
- （二）接收方在接收到发送方的数据包对数据包进行验证，确认信息后，同样向发送方发送一个数据包，数据包中包括 SA 载荷、Nonce 和标识载荷。
- （三）最后由发送方发送 Hash 载荷来判断前两步互相发送的数据。

4.2.2. 基于国密算法的服务器内核设计

基于主流 RFC 标准的 IPsec 框架使用的算法默认为 AES 算法、SHA-1 算法等，因此仅需配合 Linux 内核的算法库便可以方便地进行调用。然而基于国密标准下的 SM3 算法和 SM4 算法虽然并不在 Linux 的算法库中，但是通过硬件加密卡的加入添加相应算法，来解放软件 CPU 与内存的压力，在一定程度上提高了传输速度。

除此之外，SM3 算法与 SHA-1 相比，它的输出长度更长，且更能保证杂凑算法在使用中的安全性。SM3 算法是一种根据信息的长度位数产生相应的摘要值的哈希算法，其长度可高达 256 位。其算法通过信息的数字签名、消息验证码、随机数生成摘要等方式来进行验证。同时 SM3 的计算方式较严谨，例如在压缩函数的时候，每次都使用了两个消息字。。

同样，SM4 较于 AES 的加密效率更高。SM4 分组密码算法是我国由国密局首次公布的商用密码算法。其算法其中包括了非线性 S 盒与 32 位循环移位^[54]。AES 对称算法属于美国

国家标准技术研究所发布的主流加密算法，在国际 RFC 标准下，大众往往通过 AES 对称加密算法进行加密。但 SM4 密码算法较于 AES 对称加密算法在生成密钥过程来说更为简洁，同时解密时也无需另外进行代码编写。并且由于 SM4 源码并未公布，密码算法的安全性有所保障。

根据以上原因，本文对服务器内核进行优化设计，使得 IPSec 框架能便利地调用加密算法，进行适配。服务器内核的具体设计如图 4.5 所示：

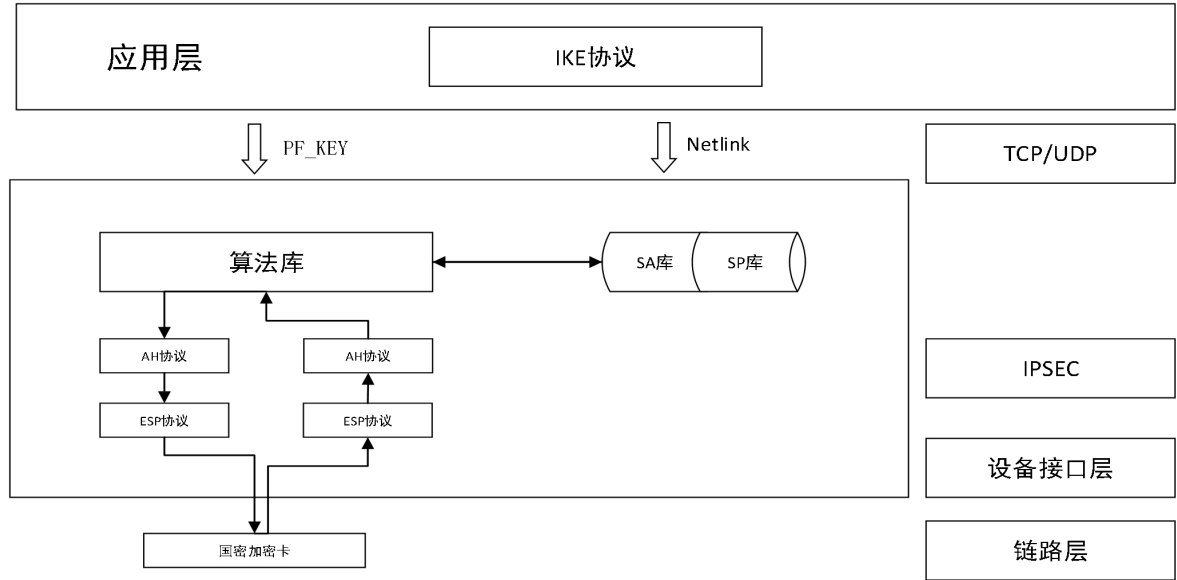


图 4.5 IPSec 内核整体架构图

下文将对数据加密解密模块、安全策略管理模块进行简要介绍，并对 IPSec 数据包的接收发送流程进行解析。

4.2.2.1. 算法库加密解密模块

通过 IPSec 协议进行数据传输，需要对数据进行加解密。但是目前的 Linux 内核只有大众常用的加密算法例如 RSA、AES、SHA-1 等，并不包含国密算法。所以在操作之前，我们应先在算法库中加入国密算法，使其在后续可以调用国密算法进行加密、解密操作。本文将通过硬件加密卡的方式添加。通过调用硬件加密卡中的加密算法，以此达到释放 CPU 资源压力，大幅度提高传输与运算效率的目的。由于 Linux 内核无法直接调用硬件加密卡的加密解密算法，因此需要添加一个加密框架来进行接口的转换。有了加密硬件的参与，数据加密解密模块可以为 IPSec 协议提供了 Linux 内核算法库中包含国密算法在内的所有加密算法和认证算法。

在应用层双方协商后，接收到传输的数据包，解析出相关信息，按照协商好的安全策略和安全联盟，调用加密算法库内的相应算法来完成传输过程中的加密解密和认证。

4.2.2.2. 安全策略管理模块

该模块由安全策略 SP 数据库与安全联盟 SA 数据库组成。在 IKE 协商后，通过协商好的

内容, 将这些配置信息通过 XFRM 框架对安全策略 SP 数据库与安全联盟 SA 数据库进行增加或者删除。并且其对应的安全策略和安全联盟将在乱开 VPN 连接后自动删除。

XFRM 是一种基于策略的处理 IP 数据包的网络框架, XFRM 从 Linux2.6 起就和协议栈结合, 形成一个高扩展性的网络安全架构。IPSec 核心处理于 Linux2.6 版本起, 就化为 Linux 内核中的 NETKEY 模块。由于 IPSec 与 Linux 的高度结合, 使得运算速率大幅度提高。而在 NETKEY 模块中, XFRM 框架是 IPSec 协议的核心。开发者既可以通过 XFRM 框架对 SP 数据库和 SA 数据库进行增删改查的操作, 也可以接受到 XFRM 在触发某些特定条件时发出的报警信息。

XFRM 管理着 SP 数据库与 SA 数据库, 其数据库对如何处理数据包的信息息息相关。根据安全策略确定是否要对数据包经由 IPSec 进行传输, 或者进行传输的协议、模块等信息。根据安全联盟确定加密算法的各种信息, 例如 AH 协议和 ESP 协议的算法种类、密钥信息等。

在做好上述数据库的配置处理后, 后续无论是接收数据包还是发送数据包, 都要通过解析源地址、目标地址与端口号来从数据库中获得对应的安全策略和安全联盟, 并根据其策略和算法进行加密或者解密。

4.2.2.3. IPSec 数据处理过程

通过以上步骤, 通讯双方通过 NETKEY 通讯接口获取 IKE 协商过程中确定的安全联盟 SA 和安全策略 SP 并将其各自保管在相应的数据库中, 并通过 XFRM 对其进行管理。同时, 加密算法库中添加了国密算法, 可以调用相应的算法。下面从发送 IP 数据包和接收 IP 数据包两方面进行介绍。

(一) 发送 IP 数据包

IP 数据包在传输前, 会根据安全策略 SP 和安全联盟 SA 进行判断, 从 SA 中确认该数据包是否要进行 IPSec 传输, 若需要进行 IPSec 传输, 那么根据 SP 搜寻相应的加密算法和密钥等信息。在获取算法后对数据包进行处理。

若需要通过 AH 协议进行处理, 那么数据包的 IP 头与数据内容分别经由 AH 协议处理后, 将其结果拼凑在一起, 生成输出数据包的 AH 协议头, 其中含有保留字段, 其值为该 IP 头距离报文头部的偏移量。在对数据加密处理过后, 通过 SM3 密码杂凑算法对数据进行签名处理, 便于接收方在收到数据包时通过签名来检查数据包在传输过程中是否完整。

若要经由 ESP 协议进行处理, 那么数据包 IP 头在处理生成输出数据包的协议头, 并在协议头后方添加 16 位字节的验证数据, 其值为末端距离报文头部的偏移量, 数据内容在处理生成输出数据包的用户数据区域。

(二) 接收 IP 数据包

IP 数据包的接收与发送的过程相反。当接收方收到了 IP 数据包, 那么首先会对其进行解析判断是否经由 IPSec 协议处理。如果经过处理, 则进行后续解析。

若解析出该数据包是经由 AH 协议处理，那么首先会对其进行检查，通过校验值的计算判断该数据包是否完整，或经过了恶意篡改。若发现异常，那么对此数据包将不做后续操作，直接丢弃。

若解析出该数据包是经由 ESP 协议处理，那么将会通过解析出来的信息在 SA 数据库中有无相符的安全联盟。如果找到符合的安全联盟，那么就通过匹配的安全联盟，获取对应的密钥、算法等信息，最终将密文解密，得出明文。反之，如果没有找到相对应的安全联盟，那么就不对该数据包进行后续操作。

4.3. 测试分析

本文通过安装 Ubuntu10.04 操作系统的服务器来进行测试，并把改装后的符合国密算法的 Open WRT 作为 IPSec 协议的软件。在双方服务器中都插着华正天网商用 PCI-E 加密卡 SJK0817-B，并对其 Linux 内核进行重新编译。

在搭建好环境后，双方服务器根据上方步骤进行连接，可从信息中可以看出传输信息明文已基于国密算法进行加密，并由 SM3 密码杂凑算法进行验证。整个传输过程中安全性能大范围提高。若在传输过程中被恶意第三方获取信息，也只会获取加密信息。经过 48 小时的连接，数据传输稳定，双方服务器传输正常，且未出现内存泄漏现象。已满足实地使用标准。

下文将从传输速度与解密速度两个方面与主流 IPSec 进行对比。

4.3.1. 传输速度对比

MES 系统在实地应用时，传输速度是最为基础和重要的。因此本文将通过 iperf 对不同环境下的数据传输速度进行测试。在命令行中输入 `sudo iperf -c 192.168.7.11 -i 2` 开始进行测试。首先对于主流 IPSec 进行测试：

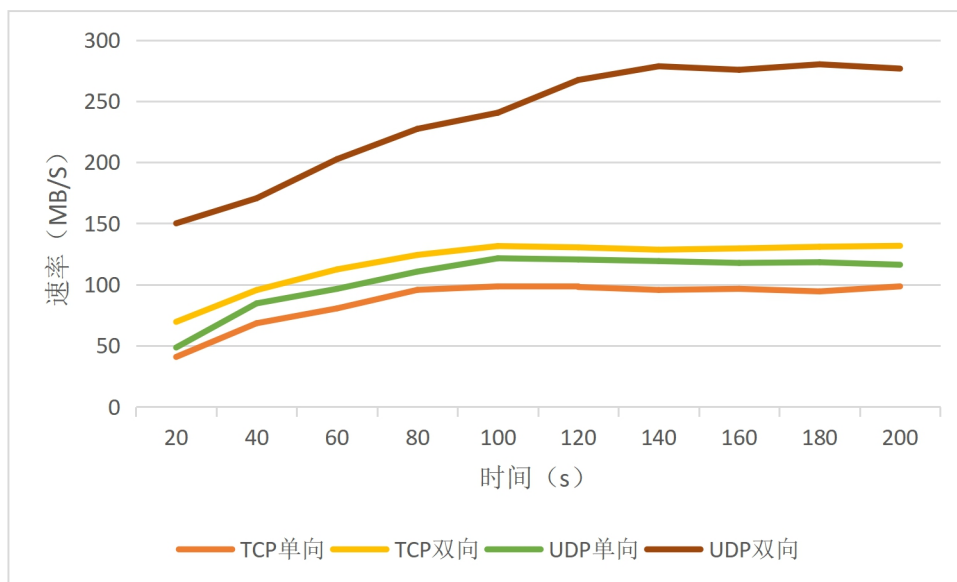


图 4.7 主流 ipsec 下不同环境传输速率测试图

通过多次测试计算出方差其平均值与方差，如下表所示：

表 4.5 主流 ipsec 下不同环境传输速率表

	平均值	方差
TCP 单向	86.582MB/S	17.897
TCP 双向	118.327MB/S	19.653
UDP 单向	105.256MB/S	22.107
UDP 双向	236.89MB/S	45.654

然后对基于国密的 IPSec 协议传输的数据进行测试，其不同环境下的测试结果如下图所示：

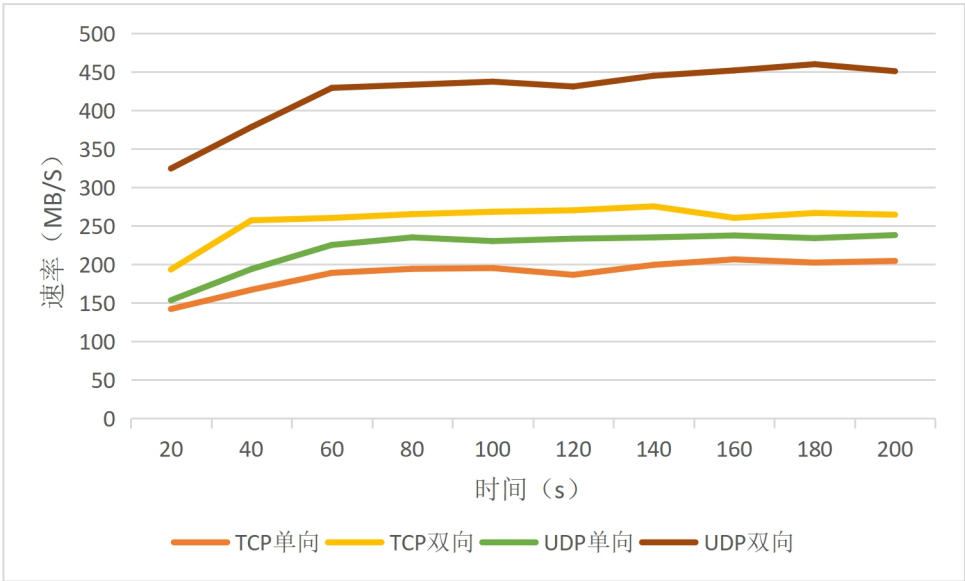


图 4.8 国密 ipsec 下不同环境传输速率测试图

表 4.6 国密 ipsec 下不同环境传输速率表

	平均值	方差
TCP 单向	188.67MB/S	18.97
TCP 双向	258.241MB/S	22.19
UDP 单向	221.637MB/S	25.88
UDP 双向	424.33MB/S	39.48

由于 UDP 无需进行三次握手，因此较 TCP 速率更快，从上述数据中可以看出，基于国密的 IPSec 在数据传输的速率较之前提高了 80%-110%不等。经过多次测试，其方差与主流 IPSec 相近，其速率的提高并未牺牲速率的稳定性。

4.3.2. 解密速度对比

本节将在操作系统为 win10，处理器为 Inter(R)Core(TM)i7-6700 2.6GHz 的环境下对主流对称加密算法进行测试。其测试结果如下：

表 4.7 对称加密算法性能对比表

算法	加密速率	解密速率	运行内存
DES	3.14MB/S	3.24MB/S	314KB
AES	42.61MB/S	62.14MB/S	308KB
IDEA	34.15MB/S	34.84MB/S	319KB
SM4	62.75MB/S	67.14MB/S	317KB

由此可以看出在主流对称算法中，SM4 的加密解密速率最高，同时运行内存与其余三个算法接近。在同样环境下对非对称算法进行测试，计算其速率。具体如下表所示。

表 4.8 非对称算法性能对比表

算法	加密速率	解密速率
RSA	432 次/s	15638 次/s
SM2	4214 次/s	896 次/s

从上表可看出 RSA 算法虽然在解密上比 SM2 算法快，但是在加密时 SM2 更胜一筹。

通过以上算法的对比，接着从总体框架的格局上，对主流 IPSec 协议与国密 IPSec 两种传输方式在速率这一性能进行对比。首先在 LoadRunner 上虚构 30 位用户，重复不间断的进行登录操作，将账号密码通过两种传输方式进行传输，计算在一定时间内回复结果的次数。并通过不同算法进行加密解密，通过测试结果对这些算法进行比较。

表 4.9 主流 IPSec 框架与国密 IPSec 框架对比表

	1min 回复次数	10min 回复次数	60min 回复成功比率
主流 IPSec 协议	1754605	17506489	99.997%
国密 IPSec 协议	2314986	22976425	99.999%

测试显示在国密基础上的 IPSec 框架较之前的传输速率更快，恢复成功的比例也更高。

4.3.3. 测试分析

经测试，本文的 IPSec VPN 已经过国密算法加密，且经过 48 小时的长时间连接，数据包传输正常未出现宕机，稳定性较高，可以很好地投入实地进行使用。

根据上方主流 IPSec 与基于国密标准的 IPSec 在不同环境传输速度对比，可以清晰看出硬件加密卡的加入大大提高了数据传输的速度。其中在 UDP 双向通信环境下较主流 IPSec 提高了 79% 的速度，提高的速度最多。而在 TCP 单向通信环境中，较主流 IPSec 提高了 117.9% 的速度，但是起伏最小。总体提高了 79%-118.6% 不等，为设备传输数据提供更快捷优质的服务。

此外，将 SM2 算法、SM4 算法分别于主流非对称算法和主流对称算法进行对比。根据数据易得 SM4 算法、DES 算法、AES 算法、IDEA 算法中 SM4 算法在加密与解密的综合速度

上最快，且 SM4 与其余算法所占内存相近，并未依靠牺牲内存来获得速度的快捷。而 SM2 算法在加密速度上远比 RSA 算法要快，但在解密速度上比不过 RSA 算法。为了更好地求出国密算法的综合性能，本文对主流 IPSec 与基于国密标准的 IPSec 整体加解密的速度进行对比。可看出国密 IPSec 协议回复次数约比主流协议少了 31.9%，且回复成功率也比之前高了 0.002%，其丢包率更低，更好地满足企业传输数据的需求。

根据本章测试可证明，经过硬件加密卡加入的基于国密标准的 IPSec VPN 稳定性满足企业数据传输需求，提高了数据传输的速度，并且在加密解密时速度更快、丢包率更低。

4.4. 本章小结

本章介绍了当前 MES 系统数据传输的现状，并针对传统 IPSec 的缺陷提出了加密优化的方法——基于国密标准对 IPSec VPN 进行优化。本章在协商层设计中通过加密、签名以及数字证书多种方式结合提高传输安全行。同时凭借硬件加密卡向服务器内核的 Linux 算法库添加了国密算法。使数据包在传输中基于国密标准进行处理。测试表明此种优化方式在传输过程已成功加密，传输加密速度较之前提高了 79%-118.6%不等，并有很好的稳定性与传输速率，适合实地进行应用。

第五章 总结与展望

5.1. 本文总结工作

本文针对传统仪表生产制造业的数字化转型作为契机入手,通过 MES 系统与云平台的结合,来优化仪表设备生产的制造过程,并通过派单排序、规整流程、工单进程透明等方式对工厂一线制作和企业高层管理直接结合起来,为企业在进行后续计划和排班时有了明确的目标和方向。同时在 MES 系统与云平台结合的过程中,针对云平台安全传输以及资源调度的问题进行了优化,使 MES 系统面对工厂更多设备数据以及数据实时传输的更高需求时能够安全并快速地进行数据传输、处理。本文主要做了以下工作:

(一) 对当前传统仪表设备生产过程进行调研,整理归纳目前企业在设备生产中的问题,以及改进的方向。根据该改进方向,设计出基于云 mes 的仪表生产管理系统,该系统总共包括硬件、客户端和云平台三部分。其中硬件设计中通过气缸磁感应传感器和电流互感器两种方式进行设备数据和状态的采集;云平台设计中设备云负责设备数据的传输与存储,业务云负责业务逻辑,并与客户端传输数据与信号。客户端设计包括 PDA 端、web 端、小程序端。PDA 端是基于 h5 页面的操作页面,由车间工人直接对自己的任务单进行线上反馈。Web 端是基于 PC 端采用 Vue.js 框架进行开发的,它具有完善的系统功能以及健全的权限机制,针对不同权限的用户登录,展现个性化的不同菜单栏功能。小程序端是面向全体用户,便于直观以及便捷地查看所需要的设备与工单信息。通过三个平台为所有不同的场景需求设计贴心个性化的全方位服务。

(二) 本文通过粒子群优化 PSO 算法根据不同车间和任务的需求进行工单排序。并针对当前 PSO 在计算过程中易寻找到局部最优解,陷入早熟无法逃逸的问题提出了三方面的解决方案:首先是在初始化粒子群时通过熵对粒子进行筛选,初始化出质量较好、尽量平均分布的粒子群;其次是通过学习因子和惯性权重在计算过程中的变化使能尽可能的减少局部最优解的可能性;最后是通过早熟抑制策略,对粒子群进行判断,若已经陷入早熟,那么通过混沌扰动给粒子群外力使其跳出这个区域。

(三) 由于添加了云平台这一环节,数据的传输更需要安全的保障,本文通过 IPSec 协议进行数据包的传输,并根据其不足之处,作出相应优化:在 IKE 协商过程中通过加密、签名以及数字证书多种方式提高传输安全性;同时凭借硬件加密卡向服务器内核的 Linux 算法库添加了国密算法,使数据包在传输中基于国密标准进行处理。实现基于国密算法的 IPSec VPN,是数据在传输速度和加密性能上都有了很高的优化。

5.2. 研究展望

本文针对当前传统制造业关于排班混乱、工作效率低下等痛点，设计了一套基于云 mes 的仪表设备生产系统，并针对任务科学调度和数据传输安全这两个问题进行了优化，为企业提供了完善的制造业数字化方案。但是由于时代的迅速发展，以及企业对于生产要求的提高，该系统仍可以针对以下几方面进行优化，具体如下：

（一）对于设备生产过程中，当出现设备老化、产品质量变差、操作耗能增加等问题时，会对设备生产造成极大的效率延误，因此我们可以在监控设备数据、人力效率等情况时，警惕数据异常，对设备的老化趋势及时预警，将设备异常遏制在苗头。

（二）在设备生产过程中，不同操作工的工作效率不同，本文以车间为单位对生产绩效进行展示，在此基础上，我们可以对员工的生产效率和生产质量等数据进行录入，以此作为一项凭据，并形成系统的员工薪资管理，提高员工的生产效率。

（三）该系统对工单进行分配排序，然而在实际应用中车间可能处于动态的状况，例如设备出问题，员工请假导致生产力降低等，静态的对车间的判断往往会被拘泥住。因此可以与第一个优化点结合，并且及时记录员工情况，确定车间实时的生产效率，以此更好地进行任务调度。

致谢

论文即将完成之际，我借此机会向我的老师、同学以及朋友们致以深切的感谢！

首先，我要感谢我的论文导师陈科明教授，整论文写作过程中，他给予了我最大的指导与帮助，尽管他每天的工作很繁忙，但他仍抽出宝贵的时间对我辅导，对我在论文写作过程中提出的疑问进行解答，提供思路与方法，使我有明确的写作方向，正是这样的严格负责才让我能一直朝着目标前进，让我在学习上收获颇丰。三年的师生情谊让我耳濡目染，陈老师让我看到了一名专业人士应该具备的责任与态度。在此我要向他表示最衷心的感谢！

其次，我要感谢我的同门俞锋、檀宇、安建伟、王彪、吴新虎、郑杭明、司端杨、杜坤，在三年的学习生涯里，我们一起努力一起进步，使我的思想与知识得到成长与积累，为我的论文写作打下了坚实的基础。同时他们也在我需要帮助的时候给予了我关心与鼓励，生活上给予了我极大的照顾，大家之间互帮互助，给我留下了深刻的印象，使我有了一个良好的生活环境与学习氛围，为了学校生涯的学习提供了良好的条件。在此，我一并向他们表示感谢！

感谢论文评阅和答辩委员会的老师，感谢你们的耐心，我一定会悉心听取你们宝贵的意见并积极改正。

最后，我还要借此机会感谢我的朋友和家人们，正是有他们在背后默默的支持，我才能得以顺利毕业，在此我向你们表示深深的感谢！

参考文献

- [1]陈腾. 基于 MES 平台的中小玩具企业动态管理系统设计与实现[D].电子科技大学,2011.
- [2]智造进阶, 云启未来——蒲惠云 MES 解码中小企业数字化转型路径[J].中国中小企业,2019(11):52-53.
- [3]Pedro Daniel Urbina Coronado,Roby Lynn,Wafa Louhichi,Mahmoud Parto,Ethan Wescoat,Thomas Kurfess. Part data integration in the Shop Floor Digital Twin: Mobile and cloud technologies to enable a manufacturing execution system[J]. Journal of Manufacturing Systems,2018.
- [4]林融.过程自动化技术在石化工业中的应用趋势与战略思考[J].石油化工自动化,2011,47(04):1-9+13.
- [5]闫亦农. 中小型服装企业物料管理系统的研究与开发[D].天津工业大学,2004.
- [6]黄思明.我国中小企业融资担保运作模式分析[J].科技广场,2012(07):214-216.
- [7]Eyelit Inc.; Eyelit's Integrated Manufacturing Execution System MES and Automation Solution Selected by Another Fortune 100 Company[J]. Medical Letter on the CDC & FDA,2020.
- [8]胡前科. 基于价值流分析的 S 公司生产线改善研究[D].苏州大学,2015.
- [9]蓝鹏程. 某制药企业 ERP 系统应用研究[D].山东建筑大学,2019.
- [10]侯晓慧. 生产执行系统 (MES) 在包钢焦化厂的实现[D].内蒙古大学,2010.
- [11]谢峰,吕玉琳,雷小宝,耿林.面向机械制造业的工业 4.0 技术实验平台的开发[J].实验技术与管理,2020,37(06):207-210.
- [12]李海平. 某造纸公司的 MES&ERP 的信息集成研究[D].华南理工大学,2011.
- [13]张建超,张崇,张帆.面向产业园区的云 MES 建设探讨[J].制造业自动化,2018,40(08):94-97.
- [14]刘娟,刘力,杨文娟.浅谈云 MES 助推中小型企业迈向 2025[J].智能制造,2018(11):43-44.
- [15]黄永慧,陈程凯.HTML5 在移动应用开发上的应用前景[J].计算机技术与发展,2013,23(07):207-210.
- [16]孙嘉豪. 开关磁阻电动机驱动系统的设计及其噪声抑制控制的研究[D].杭州电子科技大学,2018.
- [17]毛润华. 基于 PLC 的气缸主体自动加工系统研究[D].天津科技大学,2019.
- [18]张福刚. 基于混沌映射的粒子群优化算法改进研究[D].广西大学,2012.
- [19]常民,仇磊.一种改进的网格任务调度算法[J].微型电脑应用,2017,33(11):30-31+35.
- [20]张文越. 基于 Min-Min 和 Max-Min 算法改进的网格调度算法的研究[D].北京工商大学,2010.

- [21] T. Yokoi, Y. Kondo, K. Ikawa, A. Nakamura, K. Matsunaga. Stable and metastable structures and their energetics of asymmetric tilt grain boundaries in MgO: a simulated annealing approach[J]. Journal of Materials Science, 2021, 56(publish).
- [22] 周松泉. 一种改进的集群动态负载均衡算法[J]. 计算机与现代化, 2012(01):135-139.
- [23] 吴皓. 云环境下任务调度算法研究[D]. 南京邮电大学, 2013.
- [24] 宋寒冰. 云计算环境下基于改进 PSO 算法的任务调度研究[D]. 吉林大学, 2017.
- [25] 涂娟娟. PSO 优化神经网络算法的研究及其应用[D]. 江苏大学, 2013.
- [26] K. Raja Kumari, P. Sengottuvelan, J. Shanthini. A Hybrid Approach of Genetic Algorithm and Multi Objective PSO Task Scheduling in Cloud Computing[J]. Asian Journal of Research in Social Sciences and Humanities, 2017, 7(3).
- [27] 喻明让, 陈云, 张志刚. 离散粒子群优化算法求解多目标柔性作业车间调度问题[J]. 制造技术与机床, 2019(01):159-165.
- [28] 沙熠, 邱晓燕, 宁雪姣, 韩轩. 协调储能与柔性负荷的主动配电网多目标优化调度[J]. 电网技术, 2016, 40(05):1394-1399.
- [29] 袁欣雨. 基于改进粒子群算法的中压配电网规划研究[J]. 科技通报, 2020, 36(09):53-55+60.
- [30] 杨华芬, 杨有, 杨丽华, 董德春. 筛选和记忆相结合的粒子群算法[J]. 计算机应用研究, 2016, 33(04):1039-1043.
- [31] 程鹏飞. 基于系统协调粒子群算法的电力系统多目标优化调度研究[D]. 广西大学, 2012.
- [32][1] 高玉娇. 基于改进粒子群算法的虚拟机调度策略研究[J]. 现代商贸工业, 2019, 40(04):176-178.
- [33] 余庆. 粒子群改进算法及在水库优化调度中的应用研究[D]. 南昌工程学院, 2015.
- [34] 田东平, 徐成虎. 改进的粒子群优化算法的研究和分析[J]. 计算机工程与应用, 2008, 44(34):56-60.
- [35] 王铁君, 邬月春. 基于混沌自适应变异粒子群算法的铁路空车调配[J]. 计算机应用研究, 2011, 28(04):1276-1278+1282.
- [36] 邬开俊, 鲁怀伟, 王铁君. 基于自适应变异粒子群算法的铁路空车调配[J]. 兰州理工大学学报, 2011, 37(02):102-105.
- [37] 黄海. 基于改进粒子群算法的低碳型土地利用结构优化——以重庆市为例[J]. 土壤通报, 2014, 45(02):303-306.
- [38] 王东风, 孟丽. 粒子群优化算法的性能分析和参数选择[J]. 自动化学报, 2016, 42(10):1552-1561.
- [39] 王妍. 基于 IPSec 的 VPN 系统设计与实现[D]. 电子科技大学, 2013.
- [40] 张越. 国密 IPSec VPN 安全机制研究与实现[D]. 西安电子科技大学, 2018.
- [41] Anil Rajput, Manmohan Singh, Naveen Kher, Shumali Kankane, Pawan Meena. VPN and IPSEC

Server With IDS[J]. Oriental Journal of Computer Science and Technology,2010,3(1).

[42]瞿霞,华建祥.IPSec 中密钥交换协议 IKE 的安全性分析与改进[J].武夷学院学报,2015,34(12):65-68.

[43]何韦伟,季新生,刘彩霞.基于数字签名认证的 IKE 协议安全性分析及改进[J].计算机应用,2008(07):1807-1809+1815.

[44]Saad Arslan,Hayotjan Aliev,HyungWon Kim. Parallel Architecture of Concurrent ESP Engines for High-speed IPSec Routers[J]. 대한전자공학회 학술대회,2018.

[45]Cisco Technology Inc.; Patent Issued for System And Method For Traversing A NAT Device With IPSec AH Authentication (USPTO 10,333,919)[J]. Internet Business Newsweekly,2019.

[46]Muzaffar Rao,Thomas Newe,Ian Grout,Avijit Mathur. An FPGA - based reconfigurable IPSec AH core with efficient implementation of SHA - 3 for high speed IoT applications[J]. Security and Communication Networks,2016,9(16).

[47]李兆斌,刘丹丹,黄鑫,曹浩.基于国密算法的安全接入设备设计与实现[J].信息网络安全,2016(11):19-27.

[48]骆钊,谢吉华,顾伟,严童,王志贺,林英俊.SM2 加密体系在智能变电站站内通信中的应用[J].电力系统自动化,2015,39(13):116-123.

[49]张永建. RSA 算法和 SM2 算法的研究[D].江西理工大学,2015.

[50]李志鹏. 嵌入式可信签名终端软件栈关键技术研究[D].西安电子科技大学,2011.吕述望,苏波展,王鹏,毛颖颖,霍利利.SM4 分组密码算法综述[J].信息安全研究,2016,2(11):995-1007.

[51]蒋华,李康康,胡荣磊.一种基于 strongSwan 的 IPSec VPN 网关的实现[J].计算机应用与软件,2017,34(07):79-84.

[52]郭彦涛. 面向国密新标准的 IPsec VPN 服务器软件研究[D].西安电子科技大学,2017.

[53]J Zhou. Further analysis of the Internet key exchange protocol[J]. Computer Communications,2000,23(17).

[54]Pablo Neira - Ayuso,Rafael M. Gasca,Laurent Lefevre. Communicating between the kernel and user - space in Linux using Netlink sockets[J]. Software: Practice and Experience,2010,40(9).

[55]Vanessa de Campos Machado,Ivandro Cecconello. Development of an open source-based manufacturing execution system (MES): industry 4.0 enabling technology for small and medium-sized enterprises[J]. Scientia cum Industria,2019,6(2).

附录

作者在读期间发表的学术论文及参加的科研项目

参加的科研项目：

- [1] **工厂数字化项目平台开发，横向课题
- [2] 电工产品智能云平台建设技术服务，横向课题

专利：

- [1] 一种基于 MES 系统用户登录信息多重加密的传输方法