

基于5G技术的成型车间除尘设备远程监控系统设计

文/高 阔 段青娜 沙利波 邱 潇
红云红河集团昆明卷烟厂

为强化滤棒车间除尘设备的集控能力，结合工厂内5G网络的布局，建立了一套以5G网络为基础，使用Java语言结合实时数据库和MySQL数据管理工具的除尘设备远程自动化监控系统。相较于传统的声光报警系统，大大降低了巡检人员的工作强度，有效提高了维修人员对设备故障的应急处置能力。

5G技术以其大带宽、低延时和广连接的特性，广泛应用于各种工业场景。5G工业互联网正式被国家列入“新基建”的主要内容之一，如何让5G技术与烟草工业相结合，有力推动烟草工业向信息化、智能化方向发展，成为烟草行业内部从精益生产向智能制造转型的关键。基于5G信息技术在行业内的应用，余宝意等将5G CPE应用于烟草物流核心设备，建立了一套基于5G CPE WIN技术的终端组网形态，为同行业提供了借鉴。

除尘设备广泛用于烟草工业生产车间，分布式的布局特点，使其在远程操作和实时监控方面存在一定的弊端。以昆明卷烟厂成型车间为例，在A、B两作业区分别设有水幕除尘系统，且相互独立，由于二者间距离较远，给运维监控带来一定的困难。本文结合目前5G网络在厂区内的覆盖范围，对两套系统内部参数进行数据采集整合，并将数据集成至部门云端服务器，以便于远距离监控和预警。

除尘系统数据监控现状

昆明卷烟厂成型车间A、B作业区的水幕除尘设

备分布在联合工房U形槽的两侧，距离为500 m，采用自激式水幕除尘器，通过外接风机在除尘器内部闸板两侧形成压力差，用以激发水幕，同时在除尘管道内部产生负压，将管道内部的空气和杂质吸入除尘器内部。由于滤棒丝束的亲水特性，丝束杂质在水幕的作用下，沉积在除尘器内部，并被循环过滤，其余空气排向大气。该水幕除尘系统采用西门子315-2PN/DP PLC配合精致面板使用。在终端各传感器、电动球阀、风机变频器通过Profinet连接至PLC，并将信息存储于DB块中。当设备出现异常时，通过PLC系统内部的报警系统，将报警信息推送至精致面板显示，并在车间内部发出声光报警，提示相关人员进行处置。

在设备运行状态监控方面，由于其空间限制和对电气设备应用场景的要求，现场运行设备和控制系统不在同一空间内。目前，生产车间内仅凭声光报警信息无法准确定位故障，做出及时处理，需要巡检人员定时对除尘设备进行检查，给设备监控和维修保养带来一定的困难。同时，由于PLC设备的运行特点，无法对历史数据进行大量存储和分析

基金项目：2021年云南中烟工业有限公司科技项目“滤棒成型设备的区域性集控研究与实现”（2021xx03）

计算,以获得不同的控制模式应对不同生产场景的需求。

鉴于以上情况,本文结合工厂内5G网络的部署,构建一套以5G网络为基础,配合车间内部服务器进行计算分析的除尘设备远程监控系统,将A、B区的除尘监控系统进行有效整合,并集中显示,大大提高了设备的监控效率,避免了因故障未及时发现导致的大规模积水和电气设备停机等现象。

基于5G的除尘设备远程监控系统设计

1. 5G网络的部署

昆明卷烟厂的5G基站架构是基于3GPP的R15版

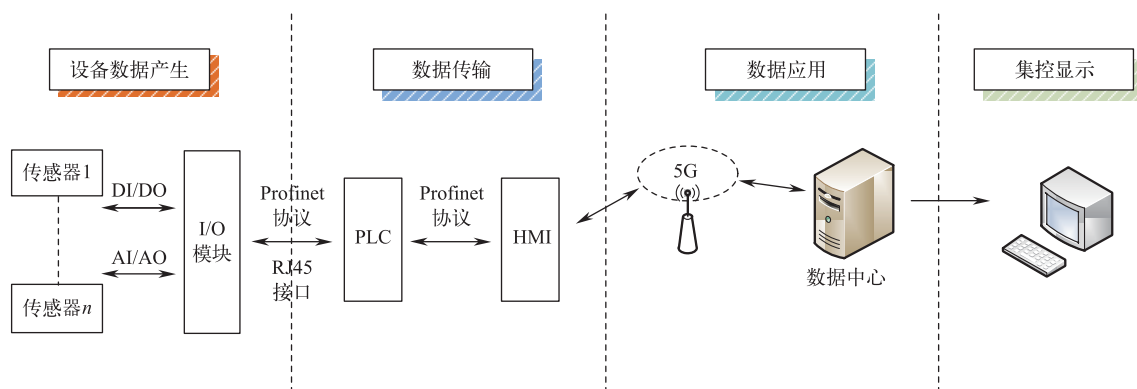


图1 5G网络数据流结构

2. 数据采集系统设计

(1) 数据采集

数据采集系统的硬件主要包含一个5G CPE模块。其工作原理是将现场传感器收集到的数据通过RS 485发送至PLC,PLC通过CP模块与精致面板和5G CPE模块相连,将DB块存储的数据通过无线信号转发至服务器端。通信协议采用西门子的S7协议,是PLC内部集成的、运行在传输层之上且经过特殊优化的通信协议,其信息传输可以基于MPI网络、Profibus网络或以太网。本文通过Java语言对采集的数据进行解析,编译机制如图2所示。即在服务器端建立基于C/S的单边通信,在pom中添加S7Connector依赖项并定义PLC的地址,依据DB块数据点表将数据取出并解析。同时Java语言在Web前端开发具有强大的开源框架支持,为系统界面设计提供基础。

本中的第一个阶段发布的产品(即NSA),是对现有的4G网络的改造和升级,并增加了5G设备,使用户能够感受到使用5G超高网速带来体验的同时,又不影响现有4G网络的配置架构。由于采用NSA,不支持MEC,在数据采集和远程控制方面,是基于存储服务器和控制服务器的超融合服务器进行操作,计算和存储均在超融合服务器内进行,因此,本项目中5G网络主要应用在eMBB场景中。通过在PLC通信模块附近添加5G CPE模块,使除尘系统的控制系统接入5G网络,由5G网络将数据转发至109网段的178服务器上,实现数据采集与控制,其结构原理如图1所示。

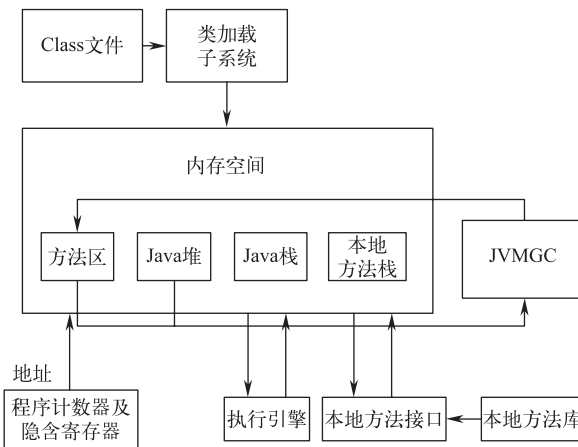


图2 Java编译运行机制

在建立连接时,定义了一个多线程的TrasmitterRemovalMethod,同时定义了多个与实时数据库相关的私有变量,当通信正确连接后,通过循环语句依据数据点表取出,程序如图3所示。

```
public class TransmitterRemovalMethod extends Thread {  
    private String machineIp;  
    private RedisTemplate redisTemplate;  
    private String machineName;  
    public TransmitterRemovalMethod(String machineIp, RedisTemplate redisTemplate, String machineName){  
        this.machineIp = machineIp;  
        this.redisTemplate = redisTemplate;  
        this.machineName = machineName;  
    }  
    @Override  
    public void run() {  
        SiemensS7Net siemensTopNet = new SiemensS7Net(SiemensPLCS.S300, machineIp);  
        OperateResult connect = siemensTopNet.ConnectServer();  
        if (!connect.IsSuccess){  
            System.out.println("除尘房=>" + machineIp + "链接失败:" + connect.Message + "");  
            redisTemplate.opsForValue().set(machineName + ":linkStatus", 0, 60 * 60 * 12, TimeUnit.SECONDS);  
            siemensTopNet.ConnectClose();  
            return;  
        }  
        redisTemplate.opsForValue().set(machineName + ":linkStatus", 1, 60 * 60 * 12, TimeUnit.SECONDS);  
        UtilCollectionIF collection = new UtilCollectionVF();  
        List<ResultSetFloat> list1Bool = collection.clearRemovalDbx1YF(siemensTopNet);  
        for (ResultSetFloat resultBool : list1Bool){  
            redisTemplate.delete(machineName + ":" + resultBool.getDesc());  
            redisTemplate.opsForValue().set(machineName + ":" + resultBool.getDesc(), resultBool.getValue(), 60 * 60 * 12, TimeUnit.SECONDS);  
        }  
        List<ResultSeBoolean> list2Bool = collection.clearRemovalDbx2YF(siemensTopNet);  
        for (ResultSeBoolean resultBool : list2Bool){  
            redisTemplate.delete(machineName + ":" + resultBool.getDesc());  
            redisTemplate.opsForValue().set(machineName + ":" + resultBool.getDesc(), resultBool.getValue(), 60 * 60 * 12, TimeUnit.SECONDS);  
        }  
        siemensTopNet.ConnectClose();  
    }  
}
```

图3 变量定义连接

(2) 数据存储

将应用层获取到Java解析的数据分为历史数据和实时数据，分别采用MySQL和实时数据库实现存储，以满足不同生产场景的需求。键值对型数据是基于非关系型的开源存储系统，可以将内存中的数据保存在软盘中，以便重启时加载使用，其高速的数据读写能力可以为前端提供实时准确的数据支撑。MySQL集群是一个无共享区、分布式节点架构的存储方案，广泛应用于Web开发应用中，具有很强的通用性。但这类数据库的灵活性不高，当遇到大量数据进行读写操作时，性能较差，在处理高并发读写时，硬盘I/O具有一定的瓶颈，因此，MySQL更适合作为数据管理。除尘房的数据采集系统既要保证高速的读写能力，又能对大量数据进行存储，结合MySQL和实时库的特点，除尘系统采用内存数据库作为实时数据中转，使用MySQL搭建数据库分布式集群，以确保数据的安全稳定。

(3) 前端设计

前端采用Vue.js框架，是构建用户界面的渐进式结构，自底向下增量开发，它的本质是JavaScript MVVM库，可将视图和模型进行动态绑定，解决了Web开发中操作DOM、渲染数据等难题，因此其前台显示的数据状态和视图是实时同步的，对于视图的维护和管理只需对数据的变化进行持续监控即可。数据请求方式可以使用Vue的常用插件axios，工作流程是首先从浏览器中创建XMLHttpRequest，从node.js中发出，通过基于Promise API接口

对请求进行处理，将数据封装成JSON后给界面提供数据来源，编程如图4所示。

```
/*  
 * @RestController  
 * @RequestMapping("/removal")  
 *  
 */  
public class RemovalController {  
    @Autowired  
    private RemovalService removalService;  
    @GetMapping(value = "/getPlanInfo")  
    public Result getBoardList(ProductionBoardQueryCriteria criteria) {  
        return removalService.getPlanInfo(criteria);  
    }  
    @GetMapping(value = "/getEachInfo")  
    public Result getTransmitterBoardList(ProductionBoardQueryCriteria criteria) {  
        return removalService.getEachInfo(criteria);  
    }  
}
```

```
// Double linkStatus = redisUtil.getRemoval(machineName, "linkStatus");  
JSONObject json = new JSONObject();  
try {  
    json.put("fhf1", redisUtil.getRemoval(machineName, key: "dbox1_1_fhf"));  
    json.put("fhf2", redisUtil.getRemoval(machineName, key: "dbox1_2_fhf"));  
    json.put("fhf3", redisUtil.getRemoval(machineName, key: "dbox1_3_fhf"));  
    json.put("fhf4", redisUtil.getRemoval(machineName, key: "dbox1_4_fhf"));  
    json.put("fhf5", redisUtil.getRemoval(machineName, key: "dbox1_5_fhf"));  
    json.put("fhf6", redisUtil.getRemoval(machineName, key: "dbox1_6_fhf"));  
    json.put("fhf7", redisUtil.getRemoval(machineName, key: "dbox1_7_fhf"));  
    json.put("gbl1", redisUtil.getRemoval(machineName, key: "dbox1_cgbl1"));  
    json.put("fns", redisUtil.getRemoval(machineName, key: "dbox1_fm_fns"));  
    json.put("fax", redisUtil.getRemoval(machineName, key: "dbox1_cfax"));  
    json.put("yct", redisUtil.getRemoval(machineName, key: "dbox1_cqyct"));  
    json.put("pls", redisUtil.getRemoval(machineName, key: "dbox1_pls"));  
    json.put("fhf21", redisUtil.getRemoval(machineName, key: "dbox2_1_fhf"));  
    json.put("fhf22", redisUtil.getRemoval(machineName, key: "dbox2_2_fhf"));  
    json.put("fhf23", redisUtil.getRemoval(machineName, key: "dbox2_3_fhf"));  
    json.put("fhf24", redisUtil.getRemoval(machineName, key: "dbox2_4_fhf"));  
    json.put("fhf25", redisUtil.getRemoval(machineName, key: "dbox2_5_fhf"));  
    json.put("fhf26", redisUtil.getRemoval(machineName, key: "dbox2_6_fhf"));  
}
```

图4 数据封装

应用效果

依据现场设备的布局情况，做出显示界面图示如图5所示。在除尘设备的监控过程中主要针对循环水箱和七台除尘器的运行频率、风速、压差等关键参数进行实时监控，以在设备故障初期能够及时发现。对比设备端HMI显示数据可知，该套数据采集系统可以准确反映设备的实时运行状况，数据与

HMI显示一致。本系统中,将整个除尘系统的状态分为风机运行状态和循环过滤状态两类,依据不同的分类,将设备的运行日志进行保存,在后期可以

针对不同状态的故障信息和时间,提出有针对性的维保提示,更加方便维修和巡检人员对整套除尘设备的使用和管理。

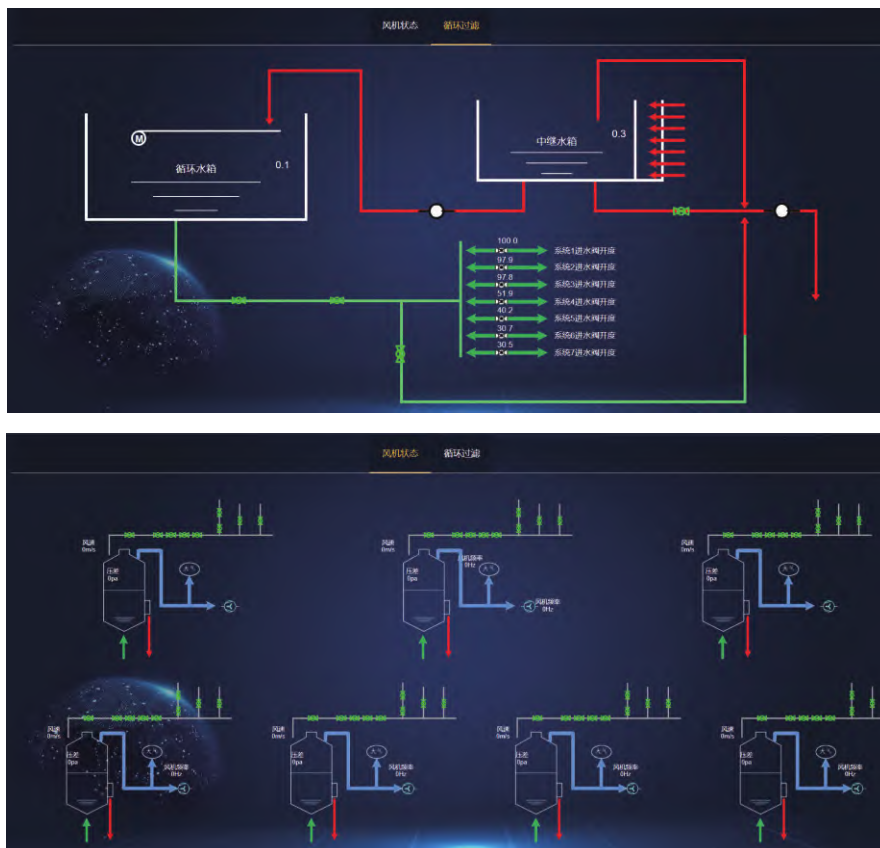


图5 VUE监控界面

结束语

本文设计了一套滤棒车间除尘设备的远程监控系统,将厂区内布设的5G信号进行合理利用,大大降低了网线敷设和相关通信设备的安装成本,同时5G网络的低时延也给除尘设备远程监控系统的实时运行提供了强大的网络通道。通过结果验证,该系统真实、动态地反映了除尘设备的运行状态,为除尘设备的故障预知、精准定位和故障诊断提供了数据基础,大大降低了人工检修工作量,有效实现了“数据流”触发“工作流”的新型管理模式,为管理人员的决策部署提供了数据基础。EA

(收稿日期:2023-02-19)

参考文献

[1] 余宝意. 5G CPE在烟草物流核心设备中的应

用研究[J]. 物流技术与应用, 2022(5): 117-119.

[2] 陈力, 张海涛, 闻磊. 基于5G的工业数据采集模式[J]. 通信企业管理, 2021(2): 50-53.

[3] 胡邦金. 基于Java的高等教育国际化调查数据处理系统[D]. 西安: 西安电子科技大学, 2021.

[4] 柏继华. 基于OPC UA协议的数据采集系统设计与实现[D]. 杭州: 浙江工业大学, 2020.

[5] 徐小辉, 刘江涛, 高涵, 等. 基于Spring-Boot+Vue 框架的采气方案系统开发[J]. 计算机仿真, 2021, 38(6): 248-250, 382.

[6] 徐晓春, 章立宗, 李奔, 等. 一种基于5G通信分布式监测终端的智能重合闸方法[J]. 电气应用, 2022, 41(11): 85-90.