

边缘计算和感知融合在智能自助咖啡机中的应用研究

王树航,徐君,杨锴,邓庆绪

(东北大学 计算机科学与工程学院,沈阳 110167)

E-mail:773735050@qq.com

摘要:针对目前云端计算处理数据能力不足、效率不高等问题,在嵌入式开发迅猛发展的背景下,将新兴的边缘计算模式与新零售的思想结合起来,运用于智能自助咖啡机系统,重新构建和设计了咖啡机系统的嵌入式系统终端、服务器和用户界面等.该系统将咖啡机作为一个智能终端接入互联网,采用 EdgeX-Foundry 边缘计算框架,并将其部署在边缘网关;硬件采用 stm32 并利用边缘计算感知融合决策咖啡机状态;通过 SpringBoot 开源框架搭建完整的服务器;在用户 APP 方面,开发并使用了微信小程序作为客户端;在系统测试中,实验结果证明了边缘计算应用于智能咖啡机系统的优势.

关键词:边缘计算;感知融合;新零售;智能咖啡机系统;EdgeX-Foundry

中图分类号: TP393

文献标识码: A

文章编号: 1000-1220(2020)07-1451-07

Research on Application of Edge Computing and Perceptual Fusion Techniques in Smart Self-service Coffee Machine

WANG Shu-hang, XU Jun, YANG Kai, DENG Qing-xu

(Department of Computer Science and Engineering, Northeastern University, Shenyang 110167, China)

Abstract: Aiming at the problems of insufficient data processing capacity and low efficiency of current cloud computing, in the context of the rapid development of embedded studying, the emerging edge computing model is combined with the idea of new retailing, and applied to intelligent self-service coffee machine systems. The embedded system terminal, server and user interface of the coffee machine system were introduced. The system uses the coffee machine as a smart terminal to access the Internet, adopts the EdgeX-Foundry edge computing framework, and deploys it at the edge gateway. Hardware adopts stm32 and uses edge computing to perception fusion to make coffee machine status; build a complete server through the SpringBoot open source framework; in the user app, WeChat applet is developed and used as a client; in system testing, experimental results demonstrate the advantages of edge computing in smart coffee machine systems.

Key words: edge computing; perception fusion; new retailing; smart coffee machine system; EdgeX-Foundry

1 引言

1.1 咖啡机发展现状

随着时代的发展,人们的消费理念也在不断地进步,而咖啡作为一种可以帮助提神的日常饮品,正逐渐走进全世界越来越多人的生活.当前,中国正越来越与国际接轨,咖啡文化慢慢进入中国人的主流饮品文化之中.据有关的数据显示和预测,2020 年中国咖啡市场规模将达到 3000 亿元,到 2025 年将超过 1 万亿元,并且在未来的十年中,中国将会逐步超过美国,成为全世界咖啡消费最大的国家之一.但是与此同时,传统现磨咖啡业经营巨头如星巴克,Costa 等,正大举扩展中国市场份额,而传统咖啡店由于高昂地租及人力成本等缺点,注定了线下开店形态难以短时间迅速复制到城市各个角落,同时其单杯几十元价格也让许多咖啡爱好者难以长期承受.这时,更多的人就将目光投向了咖啡机.在小型咖啡机的帮助下,人们可以方便快捷地喝到咖啡.因此,中国的咖啡机市场

也在逐渐发展中.如今,不仅有德龙、西门子、飞利浦等国际品牌聚集在中国咖啡机市场,国产的小米心想、美依、九阳 One-Cup 等本土品牌也已经逐步活跃起来.一般来说,咖啡机分为半自动咖啡机、全自动咖啡机、胶囊咖啡机等.本文所介绍的智能自助咖啡机系统,是将原有的全自动咖啡机进行改装,在近年来大热的“新零售”商业模式的背景下,运用物联网的相关知识,对咖啡机进行远端控制,实现对咖啡的自助销售.所谓“新零售”,通俗来讲就是线上零售,线上支付,新零售模式下的智能自助咖啡机可以完成无人售货,消费者可以通过手机扫码的方式自助购买想要的咖啡种类.因此新零售咖啡与传统咖啡模式相比,最大的特点就是借助互联网打破了传统物理实体店铺对于销售地点的限制.

1.2 由云计算到边缘计算

随着互联网的高速发展,物联网产业也随之蓬勃兴起,万物互联的时代已经到来.对于已经可以实现全球互联的智能设备来说,需要处理的数据量也已达海量.据估计,随着接

入无线网的智能设备数量的快速增长,边缘数据量已达到泽字节(ZB)级别(1ZB=2⁷⁰B)。目前大部分小型智能物联网系统仍使用远端云计算来处理数据,海量的数据会导致服务器出现一系列问题,总结起来有以下四点:

- 1) 基于线性增长的云计算能力不足以处理海量数据;
- 2) 大量数据传输到云端会导致网络拥塞,进一步引起延迟^[1];
- 3) 将所有边缘数据都上传到云端处理会产生更多的安全和隐私问题;
- 4) 过多的数据向云端传输会引起能耗增加,造成能源浪费。

针对以上问题,也为了更好地适应未来万物互联时代中一些新兴技术如增强现实/虚拟现实(AR/VR)、智能家居、智慧城市等,学术界致力于研究新型的、更便捷更安全的计算模型,并提出了边缘计算的概念。在2004年举行的第20届IEEE会议中,H. H. Pang等人首次提到了边缘计算,并指出边缘计算会将应用程序逻辑以及底层数据推向网络边缘^[2]。美国韦恩州立大学的施巍松教授团队将边缘计算定义为:边缘计算是指在网络边缘执行计算的一种新型计算模式,边缘计算中边缘的下行数据表示云服务,上行数据表示万物互联服务,而边缘计算的边缘是指从数据源到云计算中心路径之间的任意计算和网络资源^[3]。随着近些年来物联网技术的不断飞跃,边缘计算也越来越被人们所重视。2017年11月,工业互联网产业联盟(AII)和边缘计算联盟(ECC)联合发布了边缘计算参考架构2.0白皮书^[4]。该参考架构包含边缘云、边缘计算节点、业务编排和智能服务等四个横向层次,以及服务管理、安全服务和数据全生命周期管理等纵向管理功能。

在边缘计算架构模型中,在靠近数据源处会部署若干个智能边缘节点,这些边缘节点具有存储能力并足以能够处理一些比较轻松的任务,通过在边缘端实现异常数据检测和逻辑控制,降低生产安全风险^[5]。不仅如此,由于这些边缘节点无论从网络层面还是物理层面上讲,都远远比云计算中心更靠近用户,因此对信息和数据的响应速度也是之前的云计算模式无法比拟的,这也势必会大大降低因数据传输占用网络带宽而带来的延迟,也会有效降低能耗。这样,之前提到的四个云计算数据处理模式的不足,通过新兴的边缘计算模式都能得到有效的缓解。

2 系统关键技术

2.1 EdgeX Foundry 边缘计算框架

边缘计算框架要在靠近用户一侧部署多个边缘网关节点,并要求这些边缘网关节点具备一定的存储能力和计算能力,用以处理一般的用户需求、提供一些简单的服务。由于边缘计算可以大大优化计算模型、提升服务质量,学术界已经将其列为物联网行业的重要课题,并已开发出一些边缘计算开源系统,如最早在2009年由美国卡内基梅隆大学的 Satyanarayanan M. 提出的 cloudlet^[6],代表“移动设备/IoT 设备—cloudlet—云”三层体系架构的中间层,是计算机或移动设备的集群,提供附近移动设备使用的计算能力,可视作“data center in a box”,表示该数据中心更靠近用户。全球各大云计

算和设备的巨头也已投入到商业性边缘计算系统的研发中,如阿里云的首个 IoT 边缘计算产品 Link-Edge、AWS 推出的边缘计算平台 Greengrass、Intel 最新一代的 Xeon D 处理器以及三大运营商移动、电信、联通推出的 MEC(移动边缘计算)一系列技术和产品。

EdgeX Foundry 是一个位于网络边缘的供应商中立开源软件平台,是由 Linux 基金会运营的厂商中立的开放源码项目,它与设备、传感器、执行器和其他物联网对象的物理、日常工作世界进行交互,目的是为工业物联网边缘计算建立一个通用框架。可以说 EdgeX Foundry 是一个边缘计算的开放互操作平台,由于该平台框架结构清晰,便于上手,因此采用 EdgeX Foundry 边缘计算框架作为智能自助咖啡机系统在边缘节点处的平台。

在 EdgeX Foundry 的边缘计算框架中,如图 1 所示,有“南侧”和“北侧”的概念,“南侧”是指物理领域内的所有物联网对象,以及与这些设备、传感器、执行器和其他物联网对象直接通信并从中收集数据的网络边缘,即更靠近用户的一侧,与用户不同种类的设备相连;“北侧”是指物理领域内的所有物联网对象,以及与这些设备、传感器、执行器和其他物联网对象直接通信并从中收集数据的网络边缘,即更靠近云端的一侧。EdgeX Foundry 边缘计算框架沟通南北两侧,相当于用户和云端的一座“桥梁”,将从“南侧”接受的数据和信息,经多层处理和筛选后传递给“北侧”。

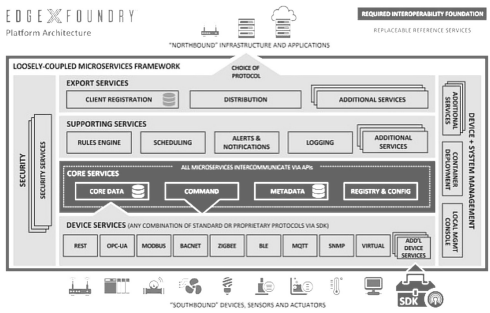


图 1 EdgeX Foundry 架构图
Fig. 1 Architecture of EdgeX Foundry

EdgeX Foundry 边缘计算框架的主体部分由开源的微服务集合构成,这些微服务层又分为 4 个服务层和 2 个全局系统服务层。4 个服务层从位于物理领域边缘的设备服务层遍历到位于信息领域边缘的导出服务层,其中核心服务层处于中心位置。

- 1) 核心服务层介于南侧与北侧之间,主要包括 4 个组成部分:①核心数据(Core Data):将从南侧收集到的数据进行持久性存储和管理;②指令(Command):是一种服务,用于促进和控制从北侧到南侧的驱使请求;③元数据(Metadata):关于连接到 EdgeX Foundry 的对象元数据的存储库和关联管理服务,提供配置新设备并将它们与其拥有的设备配对的功能;④注册表及配置(Configuration&Registration):为其他 EdgeX Foundry 微服务提供关于 EdgeX Foundry 和微服务配置属性(即初始化值存储库)中关联服务的信息。
- 2) 支持服务层包含广泛的微服务,为 EdgeX Foundry 边缘计算框架提供提示和通知、日志记录和调度等功能。

3) 导出服务层使得网关客户端能够注册来自边缘设备的所需数据,通知数据何时何地传送,通知要传递数据的格式和形状.该层在此情况下将边缘产生的数据收集并传送到云端;当网关在隔离的环境中运行,并且有时网络连接中断时,传感器和设备几乎没有外部监控或控制.因此,EdgeX Foundry 能够长时间运行和维护,而无需连接到云端系统.

4) 设备服务层负责与设备服务交互,设备服务是与设备或物联网对象交互的边缘连接器,即通常所说的各种用户设备,在本文所介绍的系统中即为智能咖啡机中的传感器等部件,这些设备将该台咖啡机的状态和信息上传至设备服务层,并由其他微服务进行下一步的处理.设备服务可以同时服务一个或多个设备.

两个全局性的系统服务层分别为安全基础设施和系统管理,其中:

1) 安全基础设施:由此框架管理的设备、传感器、和其他对象的数据和命令来确保框架内外部的安全;

2) 系统管理:提供了对于 EdgeX Foundry 对微服务,BIOS 固件,操作系统和其他网关相关软件的管理.

在智能自助咖啡机系统中,与主控制板相连的 GPRS/4G 通信模块就充当了咖啡机与 EdgeX Foundry 框架联系的“跳板”,它作为框架的设备服务,将特定咖啡机的工作状态和传感器相关数据通过无线通信的方式传递给边缘节点,再由 EdgeX-Foundry 边缘计算框架的多个微服务层进行处理,最后上传至云端.边缘节点由一个边缘网关承担,在整个系统之中主要承担着信息存储转发与咖啡机异常检测的作用.当咖啡机发生故障时,边缘节点能够及时检测出来并在第一时间通知咖啡机停止工作,确保用户不会喝到品质不佳的咖啡,同时将这些故障信息上传给后台,通知管理员进行故障维护.一个边缘网关可以同时管理多台咖啡机上传的数据.云端服务器面向网关,与网关进行信息交互,负责处理所有来网关的咖啡机信息,云端服务器还与客户端进行远程,承担起购买支付、意见反馈的作用.智能自助咖啡机系统模型如图 2 所示.

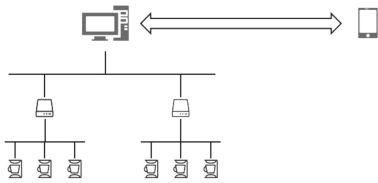


图 2 基于边缘计算的智能自助咖啡机系统

Fig. 2 Intelligent self-service coffee machine system based on edge computing

2.2 基于边缘计算的多模态感知融合异常检测算法

为了能够有效的监测咖啡机是否发生故障,利用边缘计算的思想,可以在边缘节点处,利用传感器网络结合感知融合技术来研究有效的数据异常检测算法.据 Vielzeuf 等人总结,无线传感器网络中多模态数据感知融合有以下几点好处^[7]:

1) 多模态数据从不同角度反映检测对象,经融合后的信息可以更为全面准确地描述检测对象的状态;

2) 由于融合感知数据中包含有对同一对象的多角度检测信息,噪声等干扰因素对结果的影响会显著减小;

3) 经融合感知后的数据信息能有减少原有观测值的不确定性和模糊性.

在对来自咖啡机传感器的数据进行异常检测之前,必须要对数据进行预处理,因为来自于不同传感器度的数据计数单位不同,数据格式也不同,若不能统一不同传感器的数据的数据格式与计数单位,则不能进行不同传感器数据之间的交叉运算,同样也就不能利用不同传感器的数据进行对咖啡机异常情况进行判断.研究中常用的数据规范方法有“最小-最大规范化”、“Z-score 规范化”和“按小数定标规范化”等.这里采用的数据规范处理方法是“Z-score 规范化”^[8].

对于一个咖啡机而言,其传感器的数据是以数据流的形式传递给边缘网关的,故定义其数据结构为:

$$DT_m = \{S_1, S_2 \cdots S_i \cdots S_m\}$$
$$\begin{bmatrix} r_1(t_1) & r_1(t_2) & \cdots & r_1(t_m) \\ r_2(t_1) & r_2(t_2) & \cdots & r_2(t_m) \\ \vdots & \vdots & \ddots & \vdots \\ r_n(t_1) & r_n(t_2) & \cdots & r_n(t_m) \end{bmatrix}$$

(1)

DT_m 表示在 m 个周期内接受到的 n 个传感器数据集合, S_i 表示在第 i 个周期内接收到的 n 个传感器数据集合,即 $S_i = \{r_1, r_2, r_3 \cdots r_n\}$,而 $r_i(t_j)$ 则表示在 t_j 时刻传感器 r_i 向边缘网关传送的数据.

由于边缘网关的存储数据的能力有限,而且随着传感器上传的数据不断累积而增多,边缘网关的存储器容量最终会达到饱和,故采用滑动窗口来管理传感器上传的数据.

滑动窗口模型是从传感器上传的数据流中截取窗口大小为 $|W|$ 数据,将其分成 n 个数据块,即 ($block_1, block_2, block_3, \cdots block_n$),每一个数据块的长度为 m ,当传感器上传新的数据块 $block_{new}$ 时,在滑动窗口中,最早接收的数据块 $block_{old}$ 会被替换出去.同一咖啡机上不同传感器内部均使用滑动窗口来处理数据流,如图 3 所示.

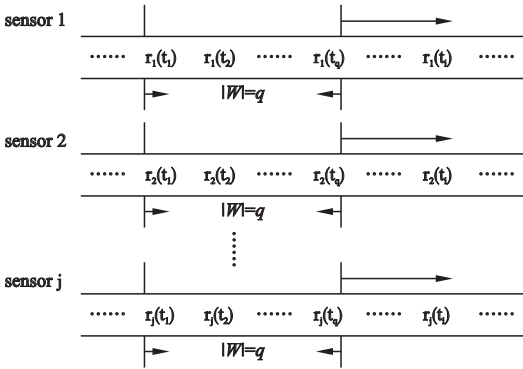


图 3 多维数据流图的滑动窗口模型

Fig. 3 Sliding window model of multidimensional data flow graph

定义 1. 假设在时间 t_q 之前将 q 个来自于某个传感器的数据载入其对应的窗口大小 $|W| = q$ 的滑动窗口内,即 ($r_j(t_1), r_j(t_2), r_j(t_3) \cdots r_j(t_q)$) 则该组处于滑动窗口内数据的方差是:

$$\delta^2 = \frac{\sum_{i=1}^q (r_j(t_i) - \bar{r}_j)^2}{q - 1}$$

(2)

\bar{r}_j 为当前滑动窗口内的数据的均值. 当滑动窗口向前滑动, 即新数据 $r_j(t_{q+1})$ 被边缘网关接收并替换 $r_j(t_i)$ 时, 则:

$$\delta^2 = \frac{\sum_{i=2}^{q+1} (r_j(t_i) - \bar{r}_j)^2}{q-1}$$

(3)

定义 2. 相干性系数:

$$\rho_{ik} = \frac{C(X_i, Y_k)}{\sqrt{C(X_i, Y_i)} \sqrt{C(X_k, Y_k)}}$$
$$= \frac{\sum_{i=1}^n (x_{ji} - \bar{X}_i)(y_{jk} - \bar{Y}_i)}{\sqrt{\sum_{i=1}^n (x_{ji} - \bar{X}_i)^2} \sqrt{\sum_{i=1}^n (y_{jk} - \bar{Y}_i)^2}}$$

(4)

其中: x_{ji}, y_{jk} 分别是 X_i, Y_k 两条数据流中的第 k 个数. 当 ρ 大于 0 时, 表示两条数据流是正相关, 当 ρ 小于 0 时, 表示两条数据流是负相关, 当 ρ 等于 0 时, 表示两条数据流是不存在相关关系.

定义 3. 在正常情况下, 下一个接收到数据与当前数据窗口内的数据变化波动较小, 而当数据波动较大时, 则说明该数据可能是异常数据.

$$\left| r_j(t_i) - \frac{E_{ej}(t) + E_{nj}(t)}{2} \right| > \delta^2$$

(5)

$r_j(t_i)$ 是第 j 个传感器在 t_i 时刻接收到传感器数据, $E_{ej}(t)$ 表示正常工作的传感器在时间区域内测量值的数学期望, $E_{nj}(t)$ 表示传感器在时间区域内测量值的数学期望, δ^2 表示方差. 当满足上式时, 表示当前上传的传感器的数据可能是异常数据, 不满足则表示是正常数据.

定义 4. 当某个传感器自身出现故障时, 则传感器向上传递的数据在一段时间可能相同, 即:

$$r_j(t_i) = r_j(t_{i+1})$$

(6)

所以将此作为判断是否传感器自身上传数据异常问题的依据. 定义单个传感器数据异常的概率为:

$$P_j(t_{i+1}) = P_j(t_i) + c \cdot k^2$$

(7)

初始是 $P_j(t_0)$ 设为 0, 当发生 $r_j(t_i) = r_j(t_{i+1})$ 时, 则 $P_j(t_{i+1})$ 按照式(7)计算, 当 $r_j(t_i) \neq r_j(t_{i+1})$ 时, $P_j(t_{i+1})$ 清零.

根据式(4)可知, 传感器之间都是存在相关性的, 若有异常事件发生的话, 其他传感器必定也会有响应, 但是若只是某个传感器出现了异常数据的话, 则其他传感器不会也发生响应, 故而计算多个传感器之间的异常概率是非常有必要的. 由单个传感器数据异常概率计算多个传感器数据异常概率, 可以根据下面的式子进行计算:

$$P_T(t_i) = \sum_{j=1}^m \lambda_j \cdot P_j(t_i)$$

(8)

其中 λ_j 为权重系数, 有:

$$\sum_{j=1}^m \lambda_j = 1$$

(9)

考虑到 λ_j 与数据的波动幅度有关, 可以由每个数据流的方差得到:

$$\lambda_1 : \lambda_2 : \lambda_3 \cdots \lambda_j = \sigma_1 : \sigma_2 : \sigma_3 \cdots \sigma_j$$

(10)

定义 5. 在传感器内部, 每个时刻都可以根据式(8)得到 $P_T(t_j)$, 可以根据下面的公式判断是否传感器检测到的数异常数据, 还是有事件发生导致传感器数据异常:

$$|P_T(t_i) - \mu| < \tau$$

(11)

μ 是滑动窗口内数据的均值, τ 是滑动窗口内数据的方差, 当满足式(11)时, 说明是发生了事件而导致数据异常, 即咖啡机有故障发生^[9].

根据上面的叙述, 我们可以得出该算法的流程图, 如图 4 所示.

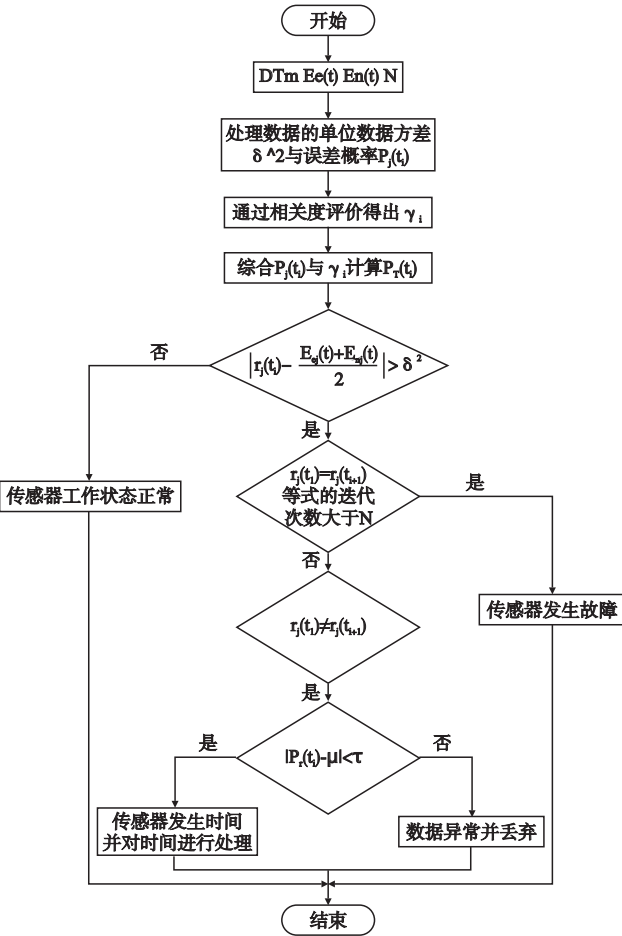


图 4 基于多模态的智能感知融合异常检测算法流程图
Fig. 4 Flow chart of anomaly detection algorithm based on multi-modal intelligent perception fusion

基于以上算法, 为了判断由于软件故障, 部分难以检测的机械故障以及其他不可知的故障导致咖啡机产生的咖啡品质降低, 这里提供了几个判定变量用于咖啡机故障的检测, 如表 1 所示.

表 1 咖啡机故障检测常用变量

变量	温度	PH 值	容量	颗粒细度
参考值	80 ~ 90.5℃	6.5 ~ 7	400mL	因咖啡种类而异

3 智能自助咖啡机系统设计

3.1 咖啡机硬件设计

在咖啡机硬件方面, 主要针对主控制板、传感器、继电器及咖啡制作和 GPRS/4G 通信模块 4 部分进行了设计和改良,

咖啡机硬件系统设计框架如图5所示。

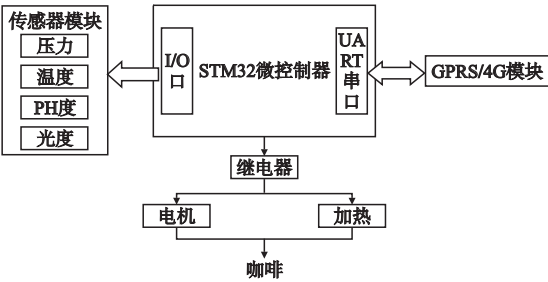


图5 咖啡机硬件系统框架图

Fig.5 Architecture of the coffee machine hardware system

3.1.1 主控制器选择

系统采用STM32F103RBT6芯片作为主控制板。STM32是ARM系列32位处理器芯片,采用高性能Cortex-M3内核。此芯片片上资源和接口丰富,具有成本低、功耗小、框架成熟等优点,在智能家居、智能管理等核心控制领域应用广泛。

3.1.2 传感器模块

传感器模块与STM32的I/O接口直接相连。在原有全自动咖啡机系统的基础上,对其中的传感器进行了改良和更换,使其能够满足对咖啡机工作状态的实时监测,当咖啡机工作状态的参数超过了初始设置得阈值,传感器就会产生中断使主控制板响应,之后再通过GPRS/4G模块及时向边缘节点上报。传感器具体使用了以下三种:1)HX711压力传感器;2)DS18B20温度传感器;3)CS1237pH传感器。

3.1.3 继电器及咖啡制作模块

通过STM32控制板与继电器连接,进而控制电机和加热电路工作,用来完成咖啡最后的冲泡工作。咖啡机内部可以设置不同的咖啡粉,通过主控制板程序来决定原料的使用比例等,这样可以准备不同风格和口味的咖啡供用户选择。此模块只涉及单纯的机械控制,在改装咖啡机系统过程中基本未作改动。

3.1.4 GPRS/4G通信模块

与STM32主控制板UART串口相连的是用于通信的GPRS/4G通信模块。在以往的物联网系统中,GPRS/4G通信模块与主控制板串口相连后另一端通常与云端服务器进行通讯,由于本系统采用边缘计算模式,在网关处部署了边缘节点,因此GPRS/4G模块通讯的对象就变成了边缘网关节点。前文已提到过,本系统在网关节点处搭建了EdgeX-Foundry边缘计算框架,其中靠近“南侧”的设备服务层与设备服务进行交互,在咖啡机系统中,GPRS/4G模块就充当了设备服务的角色,它将特定咖啡机的工作状态和异常检测结果(基于传感器上传至主控制板的数据)通过无线通信的方式传递给边缘网关节点,再由EdgeX-Foundry边缘计算框架的多个微服务层进行处理,最后上传至云端。系统采用深圳合方圆的GU900D GSM/GPRS模块,该模块的9引脚和10引脚分别是串口通信数据发送和接受引脚,分别连接主控制板STM32的RXD4和TXD4端,通过AT指令集通信。

GPRS/4G模块与主控制板的连接与数据交换是双向的,用户通过微信小程序产生的订单消息也通过GPRS/4G模块传递给主控制板,进而控制咖啡机进行相应的咖啡制作。

3.2 边缘网关设计

3.2.1 硬件设计

网关作为边缘端的核心部件,它主要由核心的处理器和外围的功能模块组成。本系统的网关不仅具有信息汇聚的协议转化功能,还具有快速的运算处理性能和丰富的外设资源。系统采用PNCORE-I.MX6嵌入式卡板,该卡板是由恩智浦公司开发,并且采用核心板加底板的设计模式,核心板采用i.MX6核心板。通过这种方式可以充分的将所有接口资源利用起来,进行硬件设计与软件开发^[10]。

目前市场上的嵌入式处理器种类繁多,常见的型号有Power PC, Am186/88, SC-400, ARM/StrongARM, Max 系列。综合成本,计算能力,接口类型等方面,本系统的网关核心处理器选择i.MX 6UltraLite。I.MX 6UltraLite是一款高性能、超高效率的处理器,采用飞思卡尔先进的单核ARM Cortex-A7,运行时钟最高频率达到了528 MHz。I.MX 6UltraLite拥有着一个集成的电源管理模块,能够有效地降低外接电源的复杂性,并简化了上电时序。同时,这个系列的每个处理器提供多种存储器接口以及提供各种接口用于连接外围设备。I.MX 6UltraLite处理器尤其适用于ePOS设备, IoT网关、门禁控制面板等应用。

3.2.2 软件设计

对于一个高效稳定的网关,软件架构也显得尤为重要。靠近用户侧的边缘网关与下面的咖啡机相互沟通,对上又与云端服务器相连,是系统中边缘计算模式中的核心部件,在本系统中搭建的EdgeX Foundry边缘计算框架,主要起到了数据接收、数据转发、适配以及数据处理的作用。

作为边缘节点,网关起到了为云端服务器分担部分处理信息的功能,有效地减小云端服务器的负荷,同时,由于为了避免咖啡机出现故障后仍然生产不合格的咖啡,咖啡机停止工作必须具有及时性,边缘网关必须要承担咖啡机的故障检测和应急处理的功能。因而,数据处理是边缘网关的最重要的功能。咖啡机可能发生的故障主要分为以下4种故障:

- 1) 缺料类型故障:主要包括咖啡机缺咖啡粉、缺水、缺牛奶、缺糖,这一类型故障较容易判断,当发生故障后,网关立即通知咖啡机停止工作,向服务器发送咖啡机缺料故障并等待维护;
- 2) 咖啡机网络类型故障:主要是咖啡机掉线,咖啡机定时向边缘网关发送keepalive报文,当超时未收到报文,即向服务器报送咖啡机掉线信息并等待维修;
- 3) 咖啡机机械故障:包括粉渣满、蒸汽开关错误、托盘位置错误、滤芯未定时更换等机械故障,其中上述错误能够很容易判断出来,这类错误由咖啡机通过GPRS/4G模块传至边缘网关,边缘网关处的EdgeX Foundry微服务层上报给服务器,通知咖啡机停止工作等待维修;
- 4) 咖啡机软件故障:主要影响咖啡的质量,这一类故障不能够通过单个传感器或者利用简单的方法来判断,必须用到用到多个传感器并结合咖啡机异常检测算法来判断,即前面所提到“基于边缘计算的多模态感知融合异常检测算法”,该算法同样也适用于部分硬件所导致的错误。

数据接收模块采用多路复用的思想,会为下面所属的每一个咖啡机设置一个channel,每当有咖啡机向网关发送请求时,网关才会接受咖啡机传来的数据,这样做提高了网关的处

理效率. 适配模块设计的主要是针对系统扩展性考虑而设置的,程序的每个模块都编译为动态库. 程序在运行的时候,会根据配置文件加载不同的库,之后会调用适配模块的某个函数,然后该函数负责管理其他的模块^[11].

3.3 服务器平台设计

服务器采用 Spring Boot 提供后端框架. Spring Boot^[12]是由 Pivotal 团队提供的全新框架,其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程. Spring Boot 框架遵循“约定优于配置”的思想,既清除了原先使用 Spring 框架的那些样板化的配置,又同时继承了原有 Spring 框架的优秀基因.

管理平台主要由 4 个模块组成,对咖啡机进行管理维护,当咖啡机发生故障时,向管理员发送故障信息,对所有购买咖啡的用户进行统一管理,以及维护整个智能咖啡机系统的功能.

1) 咖啡机管理模块:咖啡机管理模块主要起到了增删咖啡机,咖啡机状态查询,控制咖啡机生产咖啡,异常咖啡机标记等功能. 当想要创建新的咖啡机时,需要提供咖啡机编号,咖啡机种类,咖啡机的位置,能够制作的咖啡种类等相关信息. 当用户向咖啡机发出购买请求并且完成支付过程后,咖啡机管理模块向相应的咖啡机发出支付报文,并确认交易是否正确完成. 当管理员想要查看某个咖啡机状态时,只要登录后台管理平台,即可查询到自己想要的信息;

2) 咖啡机故障维护模块:咖啡机故障管理模块主要由添加故障咖啡机,删除故障咖啡机,编辑故障咖啡机三个模块构成. 当咖啡机发生故障时,咖啡机所在的网关会向服务器发送咖啡机故障信息,这是就会在咖啡机故障维护模块中添加发生故障的咖啡机,并向管理员通知. 管理员根据发生故障的咖啡机信息即时去维护相应的机器并编辑需要处理的设备. 故障排除后,将设备删除即可;

3) 用户管理模块:用户管理模块主要起到了用户注册,注销,状态查询,用户支付,收集用户反馈等功能. 当用户注册时,需要提供用户名,微信号,手机号,省份,城市,密码等基本信息. 当用户在购买咖啡过程中,用户模块负责记录下订单信息,并将其导入数据库中,同时更新用户的数据库,若用户购买咖啡失败,则将用户购买咖啡的金额退回给微信账户,终止操作;

4) 系统管理模块:系统管理模块主要包括登录管理模块,页面管理模块,退出登录模块. 主要是面向管理员而开发设计的,管理员可以使用 web 服务器登录后台管理平台对咖啡机,用户进行注册,注销,状态查询等功能,同时可以及时了解故障咖啡机的状态并对其做到及时的维护.

云端的后台管理平台可以被视作该整个咖啡机系统的“大脑”,尽管采用边缘计算极大地减少了云服务器的处理任务,但是毋庸置疑的是,后台管理平台仍然是整个系统的核心. 后台管理平台主要承担了两个最重要的任务,即:完成用户购买咖啡的交易和向管理员提供系统的状态信息. 根据上面面对管理平台不同模块的描述,能够得到该云端服务器在完成用户交易时工作的流程图如图 6 所示.

3.4 用户端设计

本系统采用微信小程序来设计整个咖啡机系统的客户终端.

微信现在是人人手机上必备的应用软件,拥有着广大的用户体系. 微信小程序因其无需安装的特点满足了人们对简约的追求,提升了客户对整体咖啡机系统的好感. 再者,小程序处于微信的生态环境之下,经过微信的审核才能发布,相较于传统的 Andriod 应用程序,提高了其安全性^[13].

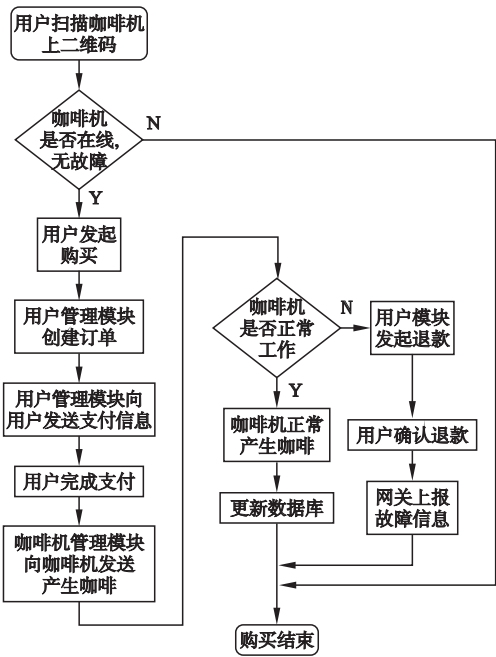


图 6 用户购买咖啡流程图

Fig. 6 User purchase coffee flow chart

该微信小程序主要面向用户提供购买支付和问题反馈两个主要功能. 首先用户使用微信去扫描智能咖啡机上的二维码,如果用户之前没有进行过注册,则会先要求用户先进行用户注册并绑定该用户微信;如果注册过,则判断该咖啡机是否发生故障通知用户,没有任何问题将会直接登录小程序. 用户在“商城”界面选择想要的咖啡,并进行付款操作. 如果购买过程中出现故障,可以在“我的”界面通过“联系电话”进行反应. 如果用户有任何建议或意见,也可通过“意见反馈”向服务器反馈.

4 系统测试

在搭建完成 Edgex Foundry 边缘计算框架和智能自助咖啡机系统后,对该系统的网络性能进行测试,测试主要针对物联网系统两个重要性能指标:丢包率和实时性,并与传统云计算模式进行比较. 通过将模拟设备发送的数据包与接收到的数据包进行统计,得到丢包率;用户给咖啡机传感器发送采集指令,传感器采集后上传咖啡机主控制板,再送给设备服务层,延时定义为从采集指令下发到设备服务层或云平台的服务层收到数据为止,由此测试此框架的数据传输性能. 本文所介绍的咖啡机系统原始数据是由传感器采集产生,经 GPRS/4G 模块发送给边缘网关,再由 EdgeX Foundry 设备微服务进行下一步处理. 将咖啡机部署于边缘网关下的一个子网中,运用 Docker 虚拟化容器技术,先安装 Docker 及镜像虚拟设备,再在 Ubuntu 上运行 EdgeX,将其作为咖啡机系统的边缘网

关,最后在另一台中央虚拟机上部署云平台。对于 Edgex Foundry 框架,具体采用 Golang 语言编写,使用 device-modbus 的设备微服务。通过 Ping 测试的方法得到系统实际丢包率和平均时延如表 2 和表 3 统计。

表 2 系统实际丢包率测试

Table 2 System actual packet loss rate test

计算模型	数据包/个				
	100	200	300	400	500
云计算丢包数/个	0	1	1	1	2
边缘计算数据包/个	0	0	1	1	0

计算模型	数据包/个				
	600	700	800	900	1000
云计算丢包数/个	1	2	2	1	3
边缘计算丢包数/个	1	0	2	1	2

表 3 系统平均时延测试

Table 3 System average delay test

计算模型	数据包/个				
	100	200	300	400	500
云计算平均时延/ms	202	205	211	208	217
边缘计算平均时延/ms	82	91	84	86	93

计算模型	数据包/个				
	600	700	800	900	1000
云计算平均时延/ms	214	223	228	221	224
边缘计算平均时延/ms	96	95	99	89	93

由表 2、表 3 中数据可以得出,随着发送数据包数量的增加,两种计算模式中该系统丢包数和时延都略有上升,传统云计算模式下,平均丢包率和时延分别为 0.26% 和 215ms,而在边缘计算模式中,这两项数据分别为 0.13% 和 91ms,平均丢包率降低了 50%,平均时延降低了约 58%,这说明边缘计算模式对于该系统的丢包率和实时性优化明显,提升了系统的物联网性能,与实验预期结果相同。

5 结 论

在万物互联的时代浪潮下,云计算模型面对爆发式增长的数据量而逐渐无法适应物联网行业新技术的发展,在此背景下,边缘计算模型应运而生。本文将边缘计算模型应用于智能咖啡机系统,在边缘网关节点处搭建了 EdgeX Foundry 边缘计算框架,用来处理来自咖啡机的状态信息,在网关的软件设计中使用了基于边缘计算的多模态感知融合异常检测算法处理数据,并从云服务器搭建、用户 APP 设计、硬件设计等几个方面介绍了咖啡机系统的设计过程,最后给出了对系统性能测试。边缘计算使该智能自助咖啡机系统运行更加安全可靠,针对用户需求响应更加迅速,感知融合技术使系统针对状态的判断更加准确,决策更加智能,云端协作的方式在智能自助咖啡机系统得到了验证,论文工作证明了边缘计算和感知融合对提升类似于智能自助咖啡机等新零售业务的有效性。

References:

[1] Ren L, Zhang Q, Shi W, et al. Edge-based personal computing services:fall detection as a pilot study[J]. Computing,2019,101(8):1199-1223.

[2] Pang H H,Tan K L. Authenticating query results in edge computing[C]//International Conference on Data Engineering(ICDE), IEEE,2004:560-571.

[3] Shi Wei-song, Sun Hui, Cao Jie, et al. Edge computing: a new computing model in the era of everything[J]. Journal of Computer Research and Development,2017,54(5):907-924.

[4] Edge computing reference architecture 2.0 white paper[R]. Edge Computing Consortium, Industrial Internet Industry Alliance,2017.

[5] Yu W, Liang F, He X, et al. A survey on the edge computing for the internet of things[J]. IEEE Access,2018,6:6900-6919.

[6] Satyanarayanan M, Bahl P, Caceres R, et al. The case for VM-Based cloudlets in mobile computing[J]. IEEE Pervasive Computing,2009,8(4):14-23.

[7] Vielzeuf V, Lechervy A, Pateux S, et al. Multilevel sensor fusion with deep learning[J]. IEEE Sensors Letters,2019,3(1):1-4.

[8] Chen Li-wan, Yang Zhen, Li Hong-bing, et al. Node scheduling problem based on improved genetic algorithm for wireless sensor networks[J]. Journal of Chongqing University of Posts and Telecommunications(Natural Science Edition), 2019,31(6):305-312.

[9] Fei Huan, Xiao Fu, Li Guang-hui, et al. Anomaly detection method of wireless sensor networks based on multi-modal data streams[J]. Acta Computica Sinica,2017,40(8):1829-1842.

[10] Zexi C, Xuan Y. Design of interconnection gateway in elevator remote monitoring system[C]//6th International Conference on Measuring Technology & Mechatronics Automation, IEEE, 2014: 247-250.

[11] Qi Xuan, Li Yi-min, Long Hua, et al. Serial-network port communication scheme design based on TCP/IP protocol stack[J]. Electronic Measurement Technology,2018,41(8):123-127.

[12] Chen Guang-jian. Spring boot development[M]. Beijing: Machinery Industry Press,2018.

[13] Zhou Wen-jie. Wechat mini program development combat[M]. Beijing: Tsinghua University Press,2019.

附中文参考文献:

[3] 施巍松,孙辉,曹杰,等.边缘计算:万物互联时代新型计算模型[J]. 计算机研究与发展,2017,54(5):907-924.

[8] 陈立万,杨震,李洪兵,等.无线传感器网络基于改进遗传算法的节点调度[J]. 重庆邮电大学学报(自然科学版),2019,31(6):305-312.

[9] 费欢,肖甫,李光辉,等.基于多模态数据流的无线传感器网络异常检测方法[J]. 计算机学报,2017,40(8):1829-1842.

[11] 齐宣,李一民,龙华,等.基于 TCP/IP 协议栈的串口-网口通信方案设计[J]. 电子测量技术,2018,41(8):123-127.

[12] 陈光剑. Spring boot 开发实战[M]. 北京:机械工业出版社,2018.

[13] 周文洁. 微信小程序开发实战[M]. 北京:清华大学出版社,2019.