

边缘计算开源平台综述

顾笛儿¹, 卢华², 谢人超^{1,3}, 黄韬^{1,3}

(1. 北京邮电大学网络与交换国家重点实验室, 北京 100876;

2. 广东省新一代通信与网络创新研究院, 广东 广州 510700;

3. 紫金山实验室, 江苏 南京 211111)

摘要: 首先介绍了边缘计算 (EC) 的基本概念和研究现状, 并从多视角讨论了边缘计算平台的设计要求; 接着聚焦到 4 个典型的开源平台, 并从应用领域、部署方式等方面比较了它们之间的异同点; 随后选取了两个典型的应用案例, 分别针对它们的优势、搭建进行了概述和分析; 最后对边缘计算平台之间的合作、安全、标准化等问题进行了归纳和展望。

关键词: 边缘计算; 开源平台; 对比性研究; 应用案例

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2021020

Survey on open source edge computing platforms

GU Dier¹, LU Hua², XIE Renchao^{1,3}, HUANG Tao^{1,3}

1. State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing 100876, China

2. Guangdong Communications & Networks Institute, Guangzhou 510700, China

3. Purple Mountain Laboratories, Nanjing 211111, China

Abstract: Firstly, the basic concept of edge computing (EC) and its current state of research were introduced. Moreover, the design requirements of edge computing platforms from multiple perspectives were discussed. Then, four typical open source platforms were presented in detail, and their similarities and differences in terms of application areas, deployment methods were analyzed. Later, with regard to two typical use cases, their installations and advantages were summarized and analyzed. Finally, key challenges such as the cooperation among edge computing platforms, security and standardization were discussed.

Keywords: edge computing, open source platforms, comparative study, use case

收稿日期: 2020-06-02; 修回日期: 2020-10-03

通信作者: 卢华, luhua@gdcni.cn

基金项目: 国家重点研发计划基金 (2019YFB1804403)

Foundation Item: The National Key R&D Program of China (2019YFB1804403)

论文引用格式: 顾笛儿, 卢华, 谢人超, 等. 边缘计算开源平台综述[J]. 网络与信息安全学报, 2021, 7(2): 22-34.

GU D E, LU H, XIE R C, et al. Survey on open source edge computing platforms[J]. Chinese Journal of Network and Information Security, 2021, 7(2): 22-34.

1 引言

近年来,随着万物互联(IoE, internet of everything)时代的到来和5G网络的普及,传感器、智能手机、可穿戴物件等物联网设备产生的海量数据与边缘设备本身受限的计算、存储资源成为一对鲜明矛盾。针对终端设备资源受限问题的传统解决方法是将数据上传至云端做统一集中式处理。但云数据中心通常部署在远离用户的地方,这就带来以下两个问题:①终端和云端之间数据传输存在较大的时延;②大量数据交换给网络带宽和数据中心之间的链路带宽带来巨大压力。为了解决上述问题,将计算资源下沉到网络边缘的边缘计算(EC, edge computing)模型应运而生。

边缘是业务的汇聚点,是网络的自然延伸,边缘计算的实现过程可概括为联结能力下移、应用能力下移和端侧算力上移,可同时满足超低时延、业务连续、超大吞吐、内容本地化的需求。EC将计算密集型任务推向边缘并在用户附近对数据进行本地处理,移动网络运营商可以减少核心网和回程链路中的流量瓶颈,同时协助将繁重的计算任务从用户设备卸载到边缘,极大减少了网络时延^[1]。边缘计算的操作对象包括来自云服务的下行数据和来自万物互联服务的上行数据^[2],边缘的定义尚无统一标准,但多认可位于接近最后一公里网络的基础设施侧和设备侧。不同边缘

位置各自产生的时延如图1所示^[3]。

边缘计算这一技术的思想最早于2009年在“微云”^[4]和“移动云计算”^[5]这两个概念中有所体现,业界认可度最高的包含雾计算(fog computing)以及移动边缘计算(MEC, mobile edge computing)这两种方案。

思科于2012年针对物联网应用场景中海量终端分布在网络边缘以及传统集中式云计算架构很难满足物联网场景中时延敏感类业务的实时性要求等问题,首次提出雾计算的概念。雾计算是云计算的一种拓展,在物联网终端设备与传统的云计算中间为用户提供计算、存储以及网络服务。随着研究深入,雾节点的范围已从早期的分布式云计算,拓展至路由器、交换机等可以部署在终端周边的网络设备。

2014年,欧洲电信标准化协会(ETSI, European Telecommunications Standards Institute)首次提出了MEC(multi-access edge computing)的概念,并将其定义为在移动网络边缘提供IT服务环境和云计算能力^[6]。随着ETSI MEC标准化工作的推进,MEC概念已经从第一阶段针对第三代合作伙伴计划(3GPP, Third Generation Partnership Project)为目标,拓展至对非3GPP网络(Wi-Fi、有线网络等)以及3GPP后续演进网络(5G等)的支持,其名称也从第一阶段的移动边缘计算修改为多接入边缘计算。

雾计算针对的是广义网络,致力于解决如何

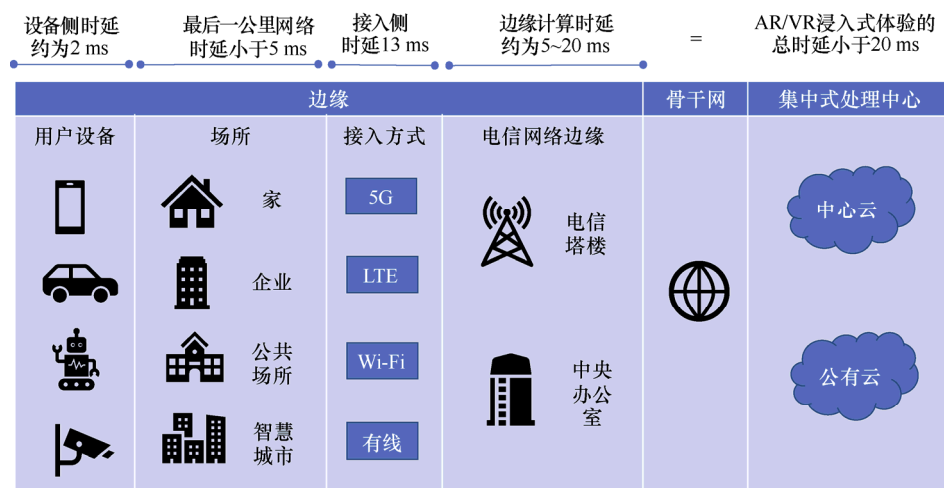


图1 不同边缘位置各自产生的时延
Figure 1 Different delays produced under different definitions of edge

通过雾节点来为移动终端或物联网终端提供分布式云计算服务,其通信链路可以基于有线网络、移动网络、Wi-Fi、传感网等任意一种连接方式。因此雾计算在节点设备选择上更加灵活,但设备算力和存储容量略逊于 MEC 服务器^[7]。上述边缘计算的概念虽然是不同研究与标准组织在不同背景下提出的,但它们都拥有一个共同目标——将云计算能力拓展至网络边缘,使终端用户可以快速、高效、安全地使用云计算服务。并且可以根据用户位置以及业务应用需求灵活部署边缘设备或者赋予相应的边缘设备网络与业务能力,以虚拟化网络与业务服务的形式,通过多种宽带接入方式,满足多租户服务的共享与隔离。

尽管雾计算和 MEC 存在很多相似之处,二者还是有一定区别。例如,EC 部署位置的差异直接决定谁会成为边缘云计算服务的提供者^[8]。在 MEC 场景下,运营商作为移动网络设施的投资建设和运营运维者,自然成为 MEC 边缘云功能的主要提供者,而雾节点可以由任意用户部署,用户也有机会成为服务提供者,甚至创建属于用户的私有云服务,成为边缘云服务生态链中的一环。本文重点讨论的边缘计算解决方案,既可以部署在运营商管控的电信大楼中,为第三方业务提供更靠近移动网络、用户的业务运营环境,实现移动网络上下文信息(位置、网络负荷等)的感知与开发;也可以部署在个人计算机中,直接实现对个人设备的操控。因此,本文统一称这些方案为边缘计算平台。

面对边缘数据的爆炸式增长,边缘计算作为一项新兴技术受到了学术界和产业界的高度关注。企业产生的数据大约有 10%是在传统集中式数据中心之外产生并处理的,据 IT 咨询机构 Gartner 估计,至 2025 年,这个数字将会上升至 75%^[9]。Grand View Research 预测,对比 2019 年的 35 亿美金,2020 年到 2027 年全球边缘计算市场规模的复合年增长率将超过 37%^[10],这意味着大数据时代的数据源也在经历转变——从大规模云数据中心转变为覆盖越来越广的边缘设备^[11]。为了充分利用通信链路上的计算、存储和网络资源处理这些数据,业内已有多家厂商与机构加入了边缘计算平台系统的开发中,以实现资源的统

一管理并方便开发者快速部署应用。

从用户视角来看,这些边缘计算平台必须满足以下体验要求。

(1) 互操作性:设备和服务器之间必须通过相同的通用通信协议进行连接。

(2) 可伸缩性:框架必须足够灵活,可以自适应容纳新添的用户和传感器。

(3) 可拓展性:框架必须支持新功能和设备的集成,而无须重新配置边缘网络。

(4) 安全性:框架必须防止数据泄露以及未经授权的内部用户访问网络。

从开发者视角来看,边缘平台开发必须满足以下设计要求。

(1) 同一个边缘计算平台多微服务框架可以互相通信,并且可基于多微服务框架进行相关的治理。

(2) 边缘计算平台可以管理多种 VIM,适配 x86、ARM 平台;支持虚拟机和容器的部署。

(3) 应用可以基于边缘计算平台的软件开发工具包(SDK, software development kit)或者 Rest API 使用 5G 网络能力。

(4) 平台拥有完善的生态应用系统,可以自动或手动打包镜像,并且能够基于平台测试提供对应应用程序接口(API, application program interface)能力。

目前已发布的边缘计算平台大致分为两类:开源框架和商业产品^[12]。Linux 基金会面对边缘计算市场的碎片化成立了一个开源项目——LF Edge,它包含 Akraino Edge Stack 和 EdgeX Foundry 等项目,该基金会希望通过这些项目,凝结用户、企业、运营商之间的合作,实现涵盖硬件、基础架构云、电信网络和应用程序的边缘生态系统。ONF 基金会和华为分别宣布了 CORD 和 KubeEdge 作为其开源边缘计算平台向广大开发者开放。商业产品包括亚马逊的 AWS、微软的 Azure 等。

边缘计算平台作为一种可同时解决数据异构性、计算资源可管理性、应用程序复杂性等问题的集成方案,已成为各界研究人员关注的重点之一。然而,尽管针对边缘计算平台的开发已经在企业内部如火如荼地展开,并取得了一些代表性成果,但学术界鲜有文献对边缘计算平台作针对性的研究。

因此,本文拟对边缘计算平台及其应用场景进行系统性的概述。

2 典型的边缘计算平台介绍

新兴的边缘计算部署方案中,用户可以通过边缘计算平台在边缘设备上安全、便捷地运行本地处理、消息转发、数据缓存及同步等操作,边缘计算平台正获得越来越多的业内关注和参与。本节就几个典型的边缘计算平台架构进行了介绍。因此,本文拟对边缘计算平台及其应用场景进行系统性的概述。

2.1 Akraino Edge Stack

Akraino Edge Stack 是 Linux 基金会于 2018 年 2 月创建的一个开源软件堆栈,支持包括 5G、LTE、Wireline 和 Wi-Fi 在内的接入方式,并提供针对边缘计算系统和应用程序优化的可用性云服务。它旨在改善企业边缘、OTT(over the top)边缘和运营商边缘网络的边缘云基础架构状态,为用户提供全新级别的灵活性,以便快速扩展边缘云服务,最大限度地提高每台服务器上支持的应用程序或用户数量,帮助确保系统运行时的可靠性^[13]。

与大多数开源项目只提供边缘计算所需的组件功能不同,Akraino Edge Stack 为构建完全集成

的边缘基础架构提供了一个涵盖基础设施、中间件及边缘计算应用的完整的解决方案,如图 2 所示。顶层是应用部署,主要负责部署边缘应用并创建 App/VNF(virtual network function)的边缘生态系统。中间层包含支持顶层应用的中间件,并通过开发、统一 API 的方式实现与第三方开源边缘计算项目的交互性操作。底层对接上层开源堆栈如 Kubernetes、OpenStack 等。

Akraino 开源项目以蓝图为组织架构,使用蓝图来实现基于边缘计算的多样化用例。每个蓝图包含上述 3 层堆栈的声明性配置,如云平台、API 和应用。具体来讲,它包括某一具体场景下的边缘计算应用或架构的配置声明、硬件、软件、管理整个堆栈的工具和交付点(PoD, pod of delivery),其中 PoD 详细描述了站点自动部署流程和脚本,覆盖了 CI/CD、集成以及测试工具等。理想情况下,每个发布的蓝图可以实现自动边缘基础架构构建、边缘堆栈站点部署、上下游项目集成和自动编排等,这意味着 Akraino 可以通过零接触配置、生命周期管理以及自动缩放功能支持大规模、低成本的边缘计算应用^[14],具有极高的商用价值。

2.2 EdgeX Foundry

EdgeX Foundry 是一个由 Linux 基金会主

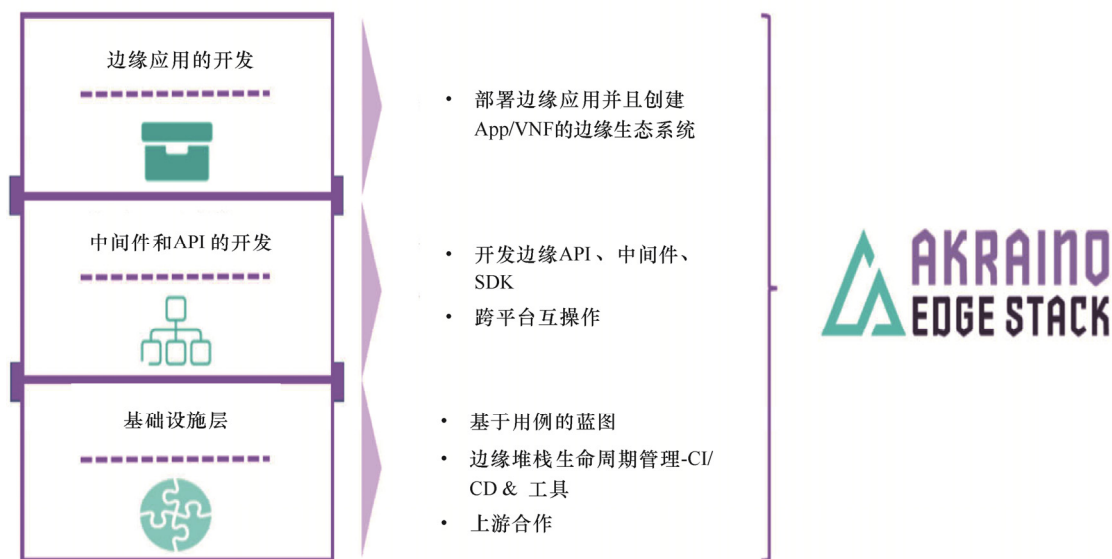


图2 Akraino Edge Stack 范围
Figure 2 The framework of Akraino Edge Stack

持,旨在为物联网边缘计算构建通用的开放框架的开源项目。该框架可以运行在包括网关、路由器、边缘服务器在内的任何边缘节点上,它独立于硬件、协议和操作系统,实现了即插即用的组件生态系统,解决了异构设备和应用程序之间的互操作性问题(如支持分析、数据编排、系统管理和服务、安全性等),进而统一了市场,推动了物联网解决方案的部署。

EdgeX Foundry 亦可视为一系列松耦合、开源的微服务集合。微服务架构允许开发人员将某一应用程序划分为一系列独立的服务,每个服务运行在独立的进程中,服务与服务采用轻量级的通信机制(如 Restful API)进行沟通。EdgeX Foundry 中所有微服务都被部署成彼此之间相互隔离的轻量级容器,支持动态增加或减少功能^[15],保证了整体框架的缩放能力和可维护性。

EdgeX Foundry 的整体架构如图 3 所示,框架下方的“南向设备和传感器”是数据产生源,指的是在物理领域内所有的物联网对象,以及

从中收集数据的网络边缘。框架上方的“北向基础设施和应用”指的是将南向传来的数据收集、存储、聚合、分析并转换为有用信息的云,以及与云通信的网络部分。EdgeX Foundry 由以下两部分组成:用于业务逻辑定义的 4 个水平子层,负责安全和管理功能的两个垂直子层(系统安全层和系统服务层),其中,第一部分的 4 个水平子层的描述如下。

(1) 设备服务层:负责将来自设备的原始数据格式转换后发送给核心服务层,并且翻译来自核心服务层的命令请求。EdgeX Foundry 提供了消息队列遥测传输(MQTT, message queuing telemetry transport)协议、ModBus 串行通信协议和低功耗蓝牙(BLE, bluetooth low energy)协议等接入方式。对于非官方提供的协议可通过库开发集成此类协议的微服务。

(2) 核心服务层:包含负责服务注册与发现的注册表配置和注册微服务、采集和存储南向设备侧数据的核心数据微服务、描述设备自身能力的元数据微服务和向南向设备发送指令的命令微服务。

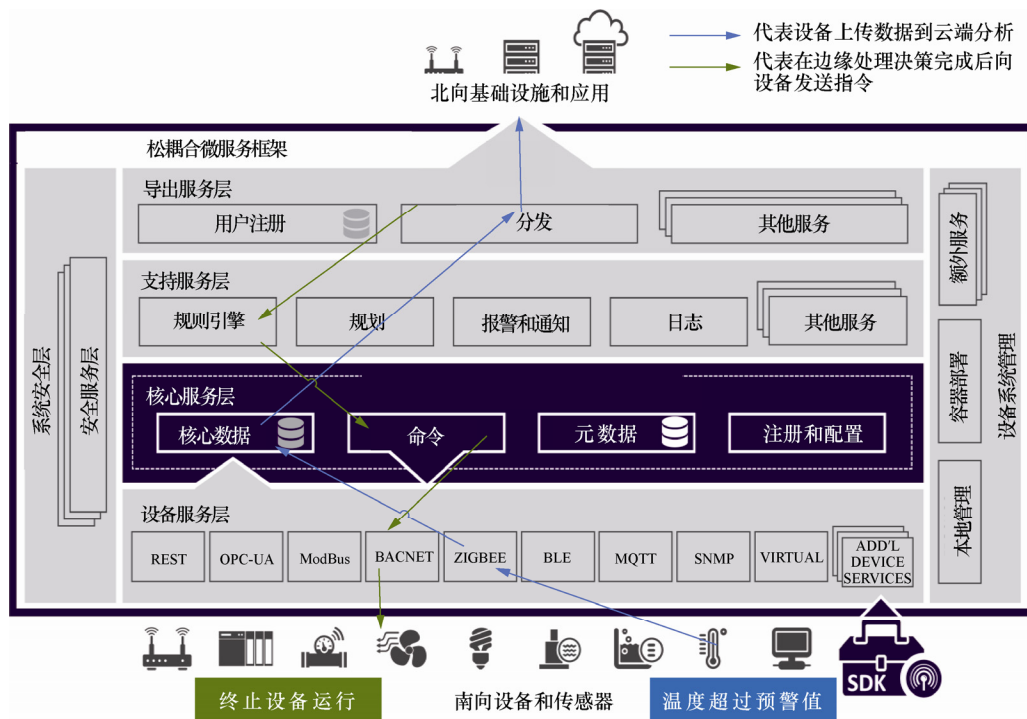


图 3 EdgeX Foundry 的整体架构
Figure 3 The framework of EdgeX Foundry

(3) 支持服务层: 提供边缘分析和智能服务。目前已安装了规则引擎、报警和通知、规划、日志、记录等微服务。规则引擎微服务允许用户设定一些规则, 当监测到数据达到规则要求时可以触发某一特定操作。报警和通知微服务可以在紧急情况或服务故障发生时通过邮件、REST callback 回调等方式告知另一系统或管理员。调度模块可以设置计时器以定期清除陈旧数据。日志记录用于记录 EdgeX 的运行信息。与设备服务层的协议类似, 用户也可以通过 API 开发新的功能模块^[16]。

(4) 导出服务层: 用于将数据传输至北向云计算中心, 由客户端注册和分发等微服务组件组成。前者使某一特定云端或本地应用可以注册为核心数据模块中的数据接收者, 后者将对应数据从核心服务层导出至指定客户端。

EdgeX 数据流首先利用设备服务层从设备中收集数据, 接着传至核心服务层做本地持久化, 最后由导出服务层转换、格式化、过滤之后传向北侧云端做进一步处理。传至导出服务层的数据亦可在规则引擎模块作边缘分析, 再通过命令模块向南侧设备发出相关指令。

代码层面上, EdgeX Foundry 的 golang 版本 edgex-go 共有 cmd、Api、Docker、Internal、pkg、snap、bin 这 7 个子目录。cmd 文件夹中包含项目程序入口 main.go, 并描述了调用的 EdgeX 微服务 IP 和端口等关键信息; Api 文件夹包含了整个框架中每个微服务以 http 形式呈现的 Api; Docker 文件夹用于构建镜像, 文本内容是构建镜像所需的指令和说明; Internal 文件夹负责实现设备微服务驱动初始化及对北向命令请求进行处理等功能, pkg 是 IDE 自动生成的打包文件, 方便应用自动化部署, snap 包含所有基于 EdgeX Go 的微服务; bin 包含可执行的 exe 文件。

综上, EdgeX Foundry 提供了一个双重转换引擎: 一是将来自不同传感器和设备、有着不同通信协议和格式的数据统一转换成常见的 EdgeX 数据结构; 二是通过基于 TCP/IP 的协议以客户指定格式向应用程序、企业、云系统提供数据^[17]。

2.3 CORD

CORD 是 ONF 基金会使用软件定义网络

(SDN, software defined network)、网络功能虚拟化(NFV, network functions virtualization)、云计算等技术, 为网络运营商创建的一个开放的、云原生、可编程的数据中心开源软件平台。它参考云原生的设计原则, 借助云计算的敏捷性和通用硬件的规模性构建更加灵活和经济的未来网络基础设施, 并通过业务协同和编排层实现应用的快速响应和灵活部署。

CORD 是一个集成了商用硬件和开源软件的通用平台, 包括以下 4 个面向不同部署场景的解决方案: 面向 5G 移动边缘基站的分布式解决方案 M-CORD、面向有线接入家庭用户的 R-CORD、面向城域网和广域网中企业用户的 E-CORD 和面向性能探测和度量的 A-CORD。CORD 体系结构中通用的软硬件架构如图 4 所示^[18]。

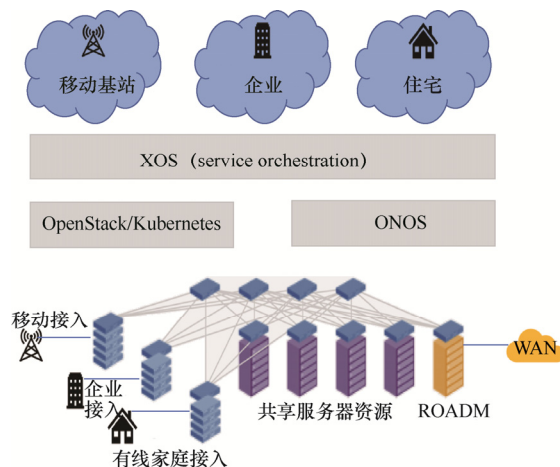


图 4 CORD 软硬件架构
Figure 4 The hardware and software architecture of CORD

CORD 的软件架构包含 OpenStack、ONOS 和 XOS 3 部分。其中, 控制层面的 XOS 是 CORD 的 Web 管理控制台, 调用数据层面的 ONOS 和 OpenStack 提供 CORD 相关服务, 负责服务的诊断监控、分配资源、性能隔离以及分发部署和迁移。OpenStack/Kubernetes 提供了 IaaS(infrastructure as a service) 服务, 管理计算存储网络资源和运行虚拟化网络功能的虚拟机和容器, 它允许用户在 CORD 平台上实现自定义功能。ONOS 是控制底层白盒(white box)硬件的网络操作系统, 同时为终端用户提供通信服务^[19]。

CORD 硬件主要包括白盒交换机以及在其基

础上互联形成的商用服务器。白盒交换机是 SDN 交换机的一个组件,负责根据 SDN 控制器调节数据流。商用服务器提供计算存储和网络资源,并将网络构建为叶脊拓扑结构以支持横向网络的通信带宽需求^[20]。商用服务器上部署了包括 Kubernetes、OpenStack 等在内的功能组件,而 ONOS 则运行在白盒交换机上。

2.4 KubeEdge

云边协同边缘计算框架 KubeEdge 是开源的云原生生态中第一个边缘计算框架,它由华为设计研发,旨在利用 Kubernetes 为边和云之间的网络通信、应用部署和元数据同步提供基础架构支持^[21]。用户既能利用 KubeEdge 在云上统一管理边缘节点上资源受限的应用和设备,又能实现云边应用和设备数据之间的同步^[22]。其整体框架共分为 3 层,分别是云端、边缘层和设备层,如图 5 所示。

(1) 云端组件

为了兼容 Kubernetes 生态,云端组件采用标

准的 Kubernetes 架构模式进行构建,这意味着 KubeEdge 的云端不负责应用的调度、管理,只是将调度到边缘节点的应用、设备元数据下发到边缘^[23]。KubeEdge 在云端的核心组件是 CloudCore,包含 EdgeController、DeviceController 和 CloudHub 3 个模块。其中 CloudHub 是云端通信接口模块,它实质是 Web 套接字服务器,负责监视云端更改、缓存以及向 EdgeHub 发送消息。EdgeController、DeviceController 负责将应用、设备元数据下发到边缘,同时将来自边缘的节点状态、应用状态、设备状态刷新到 Kubernetes 集群中。

(2) 边缘层

KubeEdge 在边缘层的核心组件是 EdgeCore,包含 EdgeHub、MetaManager 等组件。与 CloudHub 相对应,EdgeHub 是边缘通信接口模块,它实质是 Web 套接字客户端,负责同步云端资源到边缘以及上报云端边缘主机和设备的状态。MetaManager 管理边缘节点上的元数据,DeviceTwin 是处理元数据的设备软件镜像。Edged

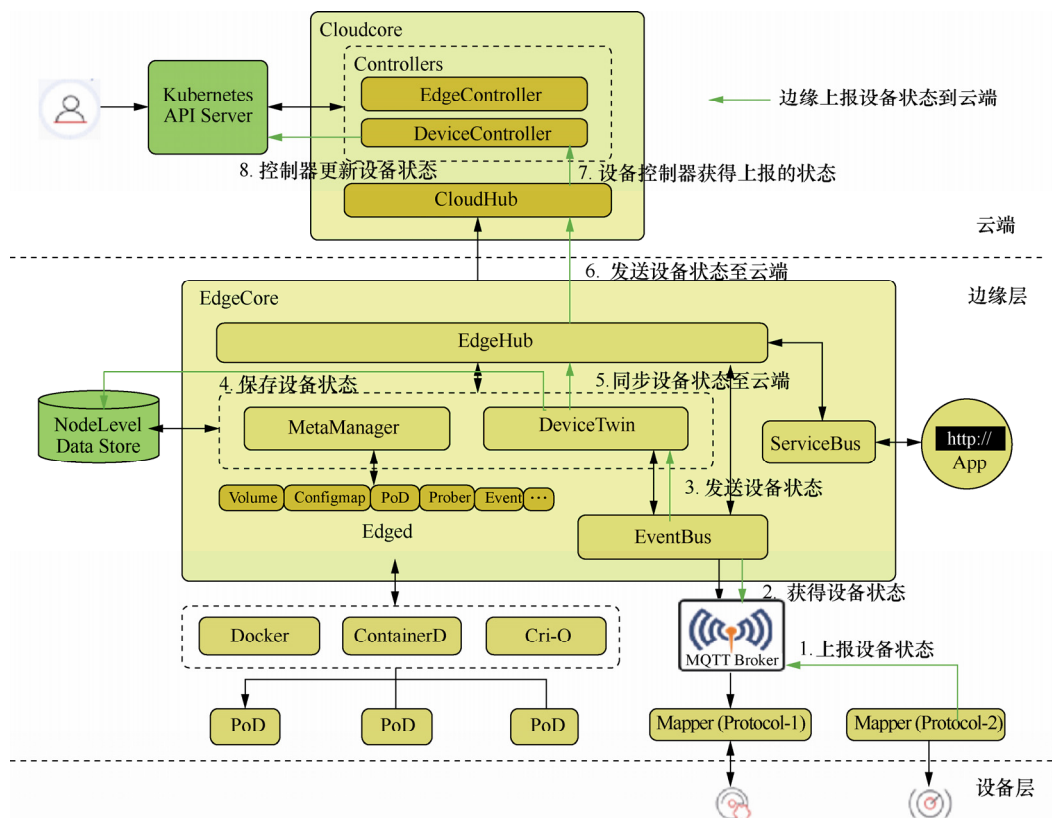


图 5 KubeEdge 框架
Figure 5 The architecture of KubeEdge

是运行在边缘节点上的 agent 程序，管理边缘容器化应用程序。

(3) 设备层

KubeEdge 利用 MQTT Broker 将设备状态变化同步到边缘节点上传到边缘节点再到云端。支持 MQTT 协议的设备可以直接接入 KubeEdge。使用专有协议（非 MQTT）的设备，通过协议转换器 Mapper 将数据处理转为 MQTT 接入 KubeEdge。目前 KubeEdge 针对工业设备场景在 DeviceAPI 中内置了 Bluetooth、Modbus、OPC-UA 3 种常见通信协议的设备支持。

以上介绍的是各模块的基本功能，下面以边缘上报设备状态到云端为例，简要说明 KubeEdge 的工作流程：实时设备状态信息在依次到达 Mapper、MQTT Broker 和 EventBus 之后，最终到达 Device Twin。此时 KubeEdge 将设备状态保存在与 Device Twin 相连的轻量级数据库 SQLite 中，接着通过 EdgeHub 同步状态信息到云端组件 CloudHub。DeviceController 在接收到 CloudHub 发送来的状态信息后完成实时更新，最后状态信息可以通过 Kubernetes API server 执行 Kubernetes 中针对各类资源对象的增删改查功能。

代码层面上，KubeEdge 除了 CloudCore 和 EdgeCore 部分，还包括横框云端和边缘层的部分。① EdgeMesh：基于 Istio 的横跨云和边缘的服务网格解决方案；②EdgeSite：为满足在边缘需要完整集群功能的场景，定制的在边缘搭建既能管理、编排又能运行负责的完整集群解决方案^[24]。这 4 个组件的源码中都使用了命令行框架 cobra，而组件中各个功能模块是通过 Beehive 统一组织和管理的，因此 KubeEdge 的代码相对而言逻辑清晰、可读性较强。

KubeEdge 充分利用 Kubernetes 的优势：良好的扩展性、跨越底层基础设施提供一致的体验、

容器的轻量化和可移植性等，同时解决了原生 Kubernetes 在边缘场景下离线自治的问题，并且支持海量异构边缘设备的接入，因此具有广阔的发展前景。

3 比较性研究

在 4 个开源边缘平台单独详细介绍的基础上，本节着重总结对比了上述平台在应用领域、部署方式等方面的不同，并分析了它们内在的联系和合作，如表 1 所示。

(1) 应用领域

Akraino Edge Stack 旨在改善企业边缘、OTT 边缘和运营商边缘的边缘云基础架构状态，它更专注于将边缘服务引入电信和网络系统。Akraino 的所有蓝图都来自实际的商业需求，涵盖了电信、5G、虚拟无线接入网（vRAN，virtual radio access network）、工业物联网（IIoT，industrial internet of things）等广泛的应用场景。

EdgeX Foundry 收集和分析来自传感器、设备或其他物联网器件的数据，并将它们导出至边缘计算应用或云计算中心作处理，保障了各类异构设备接口与应用程序之间的互操作性，因此是统一物联网边缘计算平台的一大利器。

CORD 核心是通过软硬件解耦、控制与转发分离等技术，将传统运营商网络转变成类似于云服务商的数据中心，从而达到降低网络成本、提升网络效益的目标。目前 CORD 已将 R-CORD、M-CORD、E-CORD 三者打造成统一的通用平台，向多边缘接入演进。

KubeEdge 作为全球首个 Kubernetes 原生的开源边缘计算平台，依托 Kubernetes 的容器编排和调度能力管理用户的边缘节点，提供将云上应用延伸到边缘的能力，联动边缘层和云端的数据，满足客户对边缘计算资源的远程管控、数据处理、

表 1 开源平台特性比较
Table 1 Comparison of open edge system characteristics

平台	用户类型	接入类型	OS 支持	虚拟化方式
Akraino	企业边缘、OTT 边缘、运营商边缘	支持任意类型接入，如 4G/LTE、5G、有线、Wi-Fi 等	Linux	虚拟机/容器
EdgeX	个人、企业		任意操作系统	容器
CORD	蜂窝网用户、住宅用户、企业用户	多接入（有线、无线）	Ubuntu	虚拟机/容器
KubeEdge	个人、企业		Ubuntu	容器

分析决策、智能化的诉求。同时,在云端提供统一的设备或应用监控、日志采集等运维能力,为企业提供完整的边云协同一体化服务的边缘计算解决方案^[25]。目前 KubeEdge 最典型的商业应用是摄像头类场景,如园区人脸识别、车牌识别等。

(2) 部署方式

边缘计算平台普遍依赖虚拟机和容器这两种虚拟化技术。虚拟机隔离度高,适合重量级应用或对安全性有极高要求的应用,但可能会带来资源的重复使用、启动极慢等问题。而容器的轻量级意味着任意数量的微服务可以快速打包并部署到边缘计算平台上,并且容器的 API 支持生命周期管理^[26]。

Akraino Edge Stack 支持在通用架构中运行的基于虚拟机(VM)或容器化的应用,并利用 Operator Framework 对这些应用程序进行生命周期管理。Kubernetes Machine API 允许用户在任意地点部署边缘应用。例如,针对容器部署 vRAN 和 MEC 的应用场景的 Provider Access Edge 蓝图,使用 Kubernetes 提供的插件对虚拟机和容器进行集中管理^[27]。

在 EdgeX Foundry 中,所有的微服务都以各类编程语言写进轻量级容器中,微程序之间彼此隔离保障了整个 EdgeX Foundry 的可伸缩性^[28]。用户可以在路由器、网关、交换机等边缘设备中动态增减微服务,运行自定义的边缘应用。

CORD 可以通过物理部署和虚拟化部署两种方式实现^[29]。在物理部署方案中,CORD 集成了相关的开源组件(无线电和光接入网、具有可编程转发平面的白盒交换机、微服务基础设施和端到端服务网格),打包成易于配置、部署和运行的独立 PoD,配合若干物理服务器和交换机实现物理部署。在虚拟化部署方案中,PoD 又称为 CiaB(CORD-in-a-box),服务器和交换机通常以虚拟机和 Open vSwitch(OvS)的方式运行在单个物理主机中。

KubeEdge 的基本运行环境要求:一个运行了 Kubernetes 控制面的云端节点和一个可以运行 EdgeCore 的边缘节点(256 MB 内存即可)。云端部署完 Kubernetes 的控制面后,将 CloudCore 以容器的方式部署在云端,将 EdgeCore 以进程的方

式部署到边缘层即可。用户可以直接通过 Kubernetes 中的 kubectl 命令行在云端管理边缘节点、设备和应用。

(3) 内在联系和区别

Akraino Edge Stack 和 EdgeX Foundry 在 IIoT 领域的解决方案是互补的,用户可以获取 EdgeX 代码并创建一个蓝图作为端到端堆栈保存在 Akraino Edge Stack 中。EdgeX 专注于网关、传感器等物联网设备间的通信,而 Akraino 侧重于云连接。因此,增强 EdgeX 应用层上 API 与 Akraino 边缘基础设施相关的 API 之间的上下文感知,可以让基础架构感知上下文变化动态优化自身,以满足任意可兼容 EdgeX 的微服务的需求。

Akraino 和 CORD 之间存在紧密合作。例如,可以在承载网边缘支撑大量虚拟化访问技术的 Akraino 蓝图 SEBA,它是基于 R-CORD 变体的一个轻量级平台,支持用户访问及无线回传,确保流量可以通过快速通道直接进入骨干网^[30]。

此外,KubeEdge 作为 Akraino 上游社区中的一个开源堆栈,参与了 Akraino 多个蓝图的设计。例如,在 KubeEdge Edge Service 蓝图中,借助 KubeEdge,应用程序供应商可以在边缘推动各种应用程序。相比于 EdgeX Foundry 偏重端侧设备的管理,KubeEdge 还具有云边协同等智能边缘系统的能力。

4 典型应用

5G 背景下的服务场景对网络的时延、带宽、计算和存储等方面提出了很高的要求。本节着重分析两个经典案例,并分别介绍了 Akraino 和 KubeEdge 在这些场景中的应用价值。

4.1 Akraino 在车联网中的应用

在万物互联的背景下,车联网领域内激增的感知信息数据给核心网带来了巨大压力,同时实际视频传输也会受到时变信道状态的影响。针对上述问题,利用 Akraino 的 CVB(connected vehicle blueprint)蓝图在道路边缘侧实时处理地图、交通、车辆信息业务,可以在回程带宽受限的条件下,实时地为车辆提供事故预警、辅助驾驶等服务。

5G 场景下基于 Akraino 的车联网路障监测工



图6 基于 Akraino 的车联网路障监测流程
Figure 6 The process of roadblock monitoring in internet of vehicles based on Akraino

作流程如图6所示。首先探头获取到路况信息，并将相关数据及时发送到相关的MEC平台（注意分发策略可以自行配置），随后在边缘层和云端进行智能分析和决策，最后将操作指令反馈给车辆。其中，位于MEC平台的CVB蓝图共分为4层，从下到上的分布如下。①硬件层：支持Arm和x86框架；②IaaS层：支持虚拟机、容器及其他IaaS主流软件如OpenStack、Kubernetes等；③PaaS层：Tars是CVB蓝图的微服务框架，它可以提供高性能的RPC调用，在大规模场景下高效部署微服务；④SaaS层：运行最新V2X应用程序。

为减少硬件资源成本，官方提供了如下CVB蓝图部署方式^[31]：在AWS（Amazon web service）上创建3台CPU为8核，内存为16GB的虚拟机服务器，分别用来部署持续集成工具Jenkins、Tars Master节点、Tars Slave节点和车联网相关应用。用户可先在AWS（腾讯云亦可）上申请创建3个新实例，依次配置Jenkins、Tarsframework、Jenkins slave以及Tars微服务框架相连的Tars节点。最后构建CI jobs持续、自动地测试CVB蓝图。

CVB蓝图已实现如下功能^[32]。

精准定位：定位准确率比现在的GPS系统定位高出10倍。GPS系统的定位精准范围为5~10m，

但CVB蓝图可以把这个范围缩至不到1m。

便捷导航：实现实时路况信息更新，为司机提供最高效的路径。

减少违规：CVB蓝图可以让驾驶员了解某些特定区域的交通规则，如在狭窄街道前更改路线等。

4.2 KubeEdge在AI识别中的应用

随着5G通信的进一步推广，网络边缘的设备数量、产生的数据爆发式增长使集中式的数据中心面临着实时性、带宽、能耗和数据隐私的挑战，越来越多的场景需要应用边缘计算。KubeEdge作为一种云原生边缘计算解决方案，为应用从云端向云边端的迁移提供了很好的解决思路。

图7展示了一个汽车保养门店车位绩效AI识别的应用案例^[33]。在该案例中，每个门店的多个车位均需记录车辆维护数量、时长，以评估每个工位、每个门店的业务绩效。

部署上，KubeEdge中CloudCore可作为应用管理和业务运营平台部署在云端，此外云端还需配置编译环境Golang、集群管理工具Kubernetes和封装工具Docker。每个线下门店作为一个边缘侧节点，只需额外配置一台arm服务器，与云端类似，除了基础配置（如Docker），在其上安装KubeEdge边缘核心模块EdgeCore和

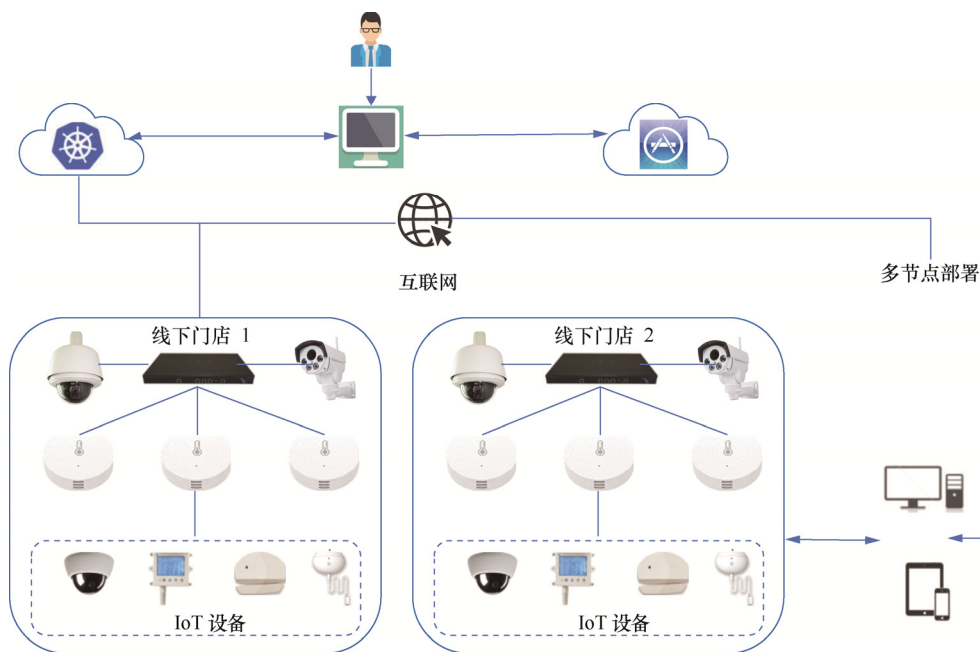


图 7 基于 KubeEdge 的 AI 识别
Figure 7 AI recognition based on KubeEdge

负责与终端设备通信的 Mosquitto, 即可将其加入云端的集群进行统一管理。用户同样可在边缘 arm 服务器上部署 AI 计算类应用, 实现高效识别。

相比传统做法中额外购置智能摄像头或者使用云识别服务, 利用 KubeEdge 可以继续采用原有系统的 IP 摄像头抓取车辆进出的照片, 实时识别车辆信息、计算车辆停留保养时间。最后汇总报表推送至云端的运营中心, 实现门店的业绩效率统计、KPI 考核以及门店所在地域的汽车维修热点统计分析。

除了减少运营成本, KubeEdge 还有助于提高识别效率。借助 KubeEdge, 单张图片识别时间不超过 50 ms, 且在网络质量不佳时可以离线使用。同时, 配合 IoT 感知技术, 可以实时监控轮胎等可燃物料存储区域的温度状态, 如果店面出现明火及烟雾时, 及时推送报警短视频到门店管理人员及运营平台, 达到自动报警的目的。

5 问题和挑战

为了充分发挥边缘计算平台的优势, 还有一些潜在的开放性研究问题与挑战需要进一步探讨, 总结如下。

1) 目前市面上的边缘计算平台的面向范围

包括物联网端、网络运营商、云边融合等, 如何提高平台之间合作的安全性是值得思考的问题。例如, 当数据从一个平台迁移到另一个平台时, 应采用何种数据迁移算法保证数据完整性^[34]。此外, 改进接口设计、提高多平台之间的互操作性也需要进一步研究。

2) 边缘设备之间的通信有时难免会涉及用户位置等个人信息, 如何加强对用户的隐私保护, 避免数据恶意泄露还需进一步研究。区块链技术作为一种限制未授权访问、防止数据被篡改的新兴技术, 可以为数据安全性提供一定的解决思路。

3) 边缘计算平台的生命周期安全管理涉及平台开发方、平台运营方、应用开发者、软硬件组件供应商等多方, 任何一环出现安全问题都将影响整个系统的正常运转。在边缘计算平台的全生命周期管理中设立关键检查点, 通过安全开发管理、代码审计、脆弱性评估、渗透测试、安全运维和事件响应与恢复等安全措施可以保障平台的顺利运行^[35]。

4) 边缘计算平台同标准化组织之间的合作有待加强。作为边缘计算标准化的先行者, ETSI 虽然在全球范围内最先给出边缘计算参考架构,

但产业内对于标准的落实情况却极为不佳。虽然一些设备厂商、互联网企业参考了部分 ETSI 提出的边缘计算框架, 但大多接口和平台建设均由企业自行定义, 离标准中所希望达到的平台互通和解耦开放等愿景尚有距离。各边缘计算平台的功能模块与 ETSI 参考架构的功能实体之间的映射关系、如何在现有的边缘计算平台中使用 ETSI MEC 标准规定的 API 接口等, 这些都是后期构建一个标准化边缘开放环境需要关注的方向。

6 结束语

本文主要从研究背景、平台架构、应用场景以及存在的问题和挑战等方面对边缘计算平台进行了系统性介绍。首先介绍了边缘计算的基本概念和本文研究的背景; 然后, 对几个典型边缘计算平台做了细节的阐述, 并从应用领域、部署方式、联系和区别这 3 个角度比较了它们之间的异同点; 随后, 详细介绍了以 Akraino 和 KubeEdge 为代表的典型案例; 最后, 对边缘计算平台中几个值得研究的方向做了进一步归纳。

参考文献:

- [1] YOUSEFPOUR A, FUNG C, NGUYEN T, et al. All one needs to know about fog computing and related edge computing paradigms: A complete survey[J]. Journal of Systems Architecture, 2019, 98: 289-330.
- [2] 张星洲, 鲁思迪, 施巍松. 边缘智能中的协同计算技术研究[J]. 人工智能, 2019 (5): 7.
ZHANG X Z, LU S D, SHI W S. Research on collaborative computing technology in edge intelligence[J]. Artificial Intelligence, 2019 (5): 7.
- [3] Akraino Edge Stack[EB]. 2020.
- [4] SATYANARAYANAN M, BAHL P, RAMÓN C, et al. The case for VM-based cloudlets in mobile computing[J]. Pervasive Computing IEEE, 2009, 8(4): 14-23.
- [5] DINH H T, LEE C, NIYATO D, et al. A survey of mobile cloud computing: architecture, applications, and approaches[J]. Wireless Communications and Mobile Computing, 2013, 13(18): 1587-1611.
- [6] MACH P, BECVAR Z. Mobile edge computing: a survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.
- [7] DOLUI K, DATTA S K. Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing[C]//2017 Global Internet of Things Summit (GIoTS). 2017: 1-6.
- [8] 张建敏, 杨峰义, 武洲云, 等. 多接入边缘计算及关键技术[M]. 北京: 人民邮电出版社, 2019.
- [9] What edge computing means for infrastructure and operations leaders[EB]. 2020.
- [10] Four edge computing market predictions[EB]. 2020.
- [11] WANG X, HAN Y, LEUNG V C M, et al. Convergence of edge computing and deep learning: A comprehensive survey[J]. IEEE Communications Surveys & Tutorials, 2020, 22(2): 869-904.
- [12] ZHAO Y, WANG W, LI Y, et al. Edge computing and networking: a survey on infrastructures and applications[J]. IEEE Access, 2019, 7: 101213-101230.
- [13] SABELLA D, REZNIK A, FRAZAO R. Multi-access edge computing in action[M]. CRC Press, 2019.
- [14] Personalized intrusion detection for internet of things gateway[EB]. 2020.
- [15] LIU F, TANG G, LI Y, et al. A survey on edge computing systems and tools[J]. Proceedings of the IEEE, 2019, 107(8): 1537-1562.
- [16] ANTONINI M, VECCHIO M, ANTONELLI F. Fog computing architectures: a reference for practitioners[J]. IEEE Internet of Things Magazine, 2019, 2(3): 19-25.
- [17] EdgeX Foundry homepage[EB]. 2020.
- [18] WANG A, ZHA Z, GUO Y, et al. Software-defined networking enhanced edge computing: a network-centric survey[J]. Proceedings of the IEEE, 2019, 107(8): 1500-1519.
- [19] HONG J, KIM W, YOO J H, et al. Design and implementation of container-based M-CORD monitoring system[C]//20th Asia-Pacific Network Operations and Management Symposium (APNOMS). 2019: 1-4.
- [20] 梁家越, 刘斌, 刘芳. 边缘计算开源平台现状分析[J]. 中兴通讯技术, 2019(3):8-14.
LIANG J Y, LIU B, LIU F. The present situation of open source platforms for edge computing [J]. ZTE Technology Journal, 2019(3): 8-14.
- [21] KubeEdge homepage[EB]. 2020.
- [22] TAO Z, XIA Q, HAO Z, et al. A survey of virtual machine management in edge computing[J]. Proceedings of the IEEE, 2019, 107(8): 1482-1499.
- [23] 张琦, 贾玄, 张森, 等. 云原生边缘计算架构分析[J]. 电信科学, 2019, 35(S2): 98-109.
ZHANG Q, JIA X, ZHANG S, et al. Analysis of cloud native edge computing architecture[J]. Telecommunications Science, 2019, 35(S2): 98-109.
- [24] KubeEdge 源码分析系列之整体架构[EB]. 2020.
KubeEdge source code analysis: overall architecture[EB]. 2020.
- [25] 智能边缘平台 IEF[EB]. 2020.
Intelligent edge platform IEF[EB]. 2020.
- [26] TALEB T, SAMDANIS K, MADA B, et al. On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1657-1681.
- [27] Kubernetes native infrastructure provider access edge[EB]. 2020.
- [28] XU R, JIN W, KIM D. Microservice security agent based on API gateway in edge computing[J]. Sensors, 2019, 19(22): 4905.
- [29] CANELLAS F, BONJORN N, KENTIS A M, et al. Build-ship-run

approach for a CORD-in-a-Box deployment[C]//9th International Conference on Cloud Computing and Services Science. IEEE, 2019.

[30] SEBA Homepage[EB]. 2020.

[31] CVB Installation Doc for R2[EB]. 2020.

[32] BRUCE L, ROBERT Q, HECHUN Z, et al. An Akraino developer use case[EB]. 2020.

[33] 智能计算、边缘计算环境下的云原生进化之路[EB]. 2020. Cloud-native evolution path under intelligent computing and edge computing environment [EB]. 2020.

[34] NING H, LI Y, SHI F, et al. Heterogeneous edge computing open platforms and tools for internet of things[J]. Future Generation Computer Systems, 2020, 106: 67-76.

[35] 陶耀东, 徐伟, 纪胜龙. 边缘计算安全综述与展望[J]. 计算机集成制造系统, 2019, 25(12): 3043-3051.

TAO Y D, XU W, JI S L. Summary and prospect of edge computing security [J]. Computer Integrated Manufacturing Systems, 2019, 25 (12): 3043-3051.

[作者简介]



顾笛儿(1996-), 女, 江苏盐城人, 北京邮电大学硕士生, 主要研究方向为边缘计算等。



卢华(1976-), 男, 江苏南京人, 广东省新一代通信与网络创新研究院副研究员, 主要研究方向为新型网络架构体系、软件定义网络、P4 可编程、虚拟化等。



谢人超(1984-), 男, 福建南平人, 博士, 北京邮电大学副教授, 主要研究方向为信息中心网络、移动网络内容分发技术、工业互联网、标识解析技术和移动边缘计算等。



黄韬(1980-), 男, 重庆人, 博士, 北京邮电大学教授、博士生导师, 主要研究方向为路由与交换、软件定义网络、内容分发网络等。