计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.201000109

云边协同综述

陈玉平[1]　刘　波[1]　林伟伟[2]　程慧雯[1]
1 华南师范大学计算机学院　广州 510631
2 华南理工大学计算机科学与工程学院　广州 510640
(1127952401@qq.com)

摘　要　在物联网、大流量等场景下,传统的云计算具有强大的资源服务能力的优点和远距离传输的缺点,而新兴的边缘计算具有低传输时延的优点和资源受限的缺点,因此,结合了云计算与边缘计算优点的云边协同引起了研究者的广泛关注。在全面调查和分析云边协同相关文献的基础上,文中重点分析和讨论了资源协同、数据协同、智能协同、业务编排协同、应用管理协同和服务协同等协同技术的实现原理和研究思路与进展。然后分别从云端和边缘端深入分析了各种协同技术在协同中所起的作用以及具体的使用方法,并从时延、能耗以及其他性能指标方面对结果进行了对比分析。最后指出了云边协同目前存在的挑战和未来的发展方向。本综述有望为云边协同的研究提供有益的参考。

关键词:云计算;边缘计算;云边协同;资源协同;数据协同;智能协同
中图法分类号　TP399

Survey of Cloud-edge Collaboration

CHEN Yu-ping[1],LIU Bo[1],LIN Wei-wei[2] and CHENG Hui-wen[1]
1 School of Computer Science,South China Normal University,Guangzhou 510631,China
2 School of Computer Science and Engineering,South China University of Technology,Guangzhou 510640,China

Abstract　In the scenarios of Internet of things,large traffic and so on,traditional cloud computing has the advantages of strong resource service ability and the disadvantages of long-distance transmission,and the rising edge computing has the advantages of low transmission delay and the disadvantage of resource limitation. Therefore,cloud-edge collaboration,which combines the advantages of cloud computing and edge computing,has attracted much attention. Based on the comprehensive investigation and analysis of the relevant literature on cloud edge collaboration,this paper focuses on the in-depth analysis and discussion of the implementation principles,research ideas and progress of cloud collaboration technologies,such as resource collaboration,data collaboration,intelligent collaboration,business orchestration collaboration,application management collaboration and service collaboration. And then,it analyzes the role of various collaborative technologies in collaboration and the specific used methods,and compares the results from the aspects of delay,energy consumption and other performance indicators. Finally,the challenges and future development direction of cloud edge collaboration are pointed out. This review is expected to provide a useful reference for the research of cloud-edge collaboration.

Keywords　Cloud computing,Edge computing,Cloud-edge collaboration,Resource collaboration,Data collaboration,Intelligence collaboration

1　引言

云计算是一种计算范式,它可以根据用户的需求随时随地为最终用户提供无限的计算资源,用户只需为使用的服务付费。云中可以提供各种类型的服务,如资源池、弹性和灵活性、可扩展性(水平和垂直)、性能高可用性、托管服务等[1]。正是因为具有强大的服务能力,云计算成为了所有业务之首,为全世界提供就业机会,近十年来被学术界和工业界广泛研究。然而,根据数据机构 IDC 的预测,2020 年底将有超过 500 亿的终端与设备联网[2],从而产生海量的异构数据。此

到稿日期:2020-10-20　返修日期:2020-12-15
基金项目:国家自然科学基金(62072187,61872084);广东省基础与应用基础研究重大项目(2019B030302002);广州市科技计划项目(202007040002,201902010040)
This work was supported by the National Natural Science Foundation of China(62072187,61872084),Guangdong Major Project of Basic and Applied Basic Research(2019B030302002) and Guangzhou Science and Technology Plan Project(202007040002,201902010040).
通信作者:林伟伟(cslinw@scut.edu.cn)

时，传统的云计算已经不能满足一些对实时性比较敏感的应用，并且将全部数据都上传到云数据中心也会给网络带宽带来很大的压力。因此，以解决数据传输延迟、降低网络带宽为目标的边缘计算正迅速兴起。

边缘计算是指在网络边缘执行计算的一种新型计算模式。边缘计算中的下行数据表示云服务，上行数据表示万物互联服务，边缘计算的边缘是指从数据源到云计算中心的路径之间的任意计算和网络资源[2]。边缘计算架构中，用户数据不再需要全部上传到云数据中心，而是通过部署在网络边缘的边缘节点快速处理部分数据，从而大大减轻了网络带宽的压力，大幅降低了网络边缘端智能设备的能耗。为此，针对边缘计算的探索性研究已经广泛展开。随着其市场规模的逐渐扩大，边缘计算成为了与云计算同台竞技的解决方案。为了更好地结合云计算与边缘计算的优势，云边协同作为一种新型计算模式成为了新的研究趋势。

已有部分边缘计算产品被逐步推出，但云边协同的发展仍处于探索阶段。随着数据密集型应用与计算密集型应用的增加，需要利用云计算强大的计算能力以及通信资源与边缘计算短时传输的响应特性来实现并完成相应的应用请求。通过两者协同工作、各展所长，将边缘计算和云计算协作的价值最大化[2]，从而有效地提高应用程序的性能。目前，针对云边协同的研究大多数集中在物联网、工业互联网、智能交通、安全监控等诸多领域的应用场景上，主要目的是减少时延、降低能耗以及提高用户体验质量等。Ren 等[4]提出的云边缘协作方法能够有效地提高延迟性能。Ding 等[5]提出了一种云边缘协作框架，通过浅层卷积神经网络模型提供持续时间长、响应速度快的认知服务，给用户带来了良好的体验。Zhang 等[6]在工业互联网中提出了一个 Cloud-Edge 协作的工业设备管理服务系统，在一定程度上提高了工业现场系统的响应速度，减轻了数据传输带来的网络带宽负载压力，推动了工业物联网向智能化发展。然而，以上研究都忽略了云边协同的底层实现原理与架构平台的服务类型。因为云计算和边缘计算既不是单一的部件也不是单一的层次，而是涉及到 IaaS，PaaS，SaaS 的端到端的开放平台，每种服务模式下都有不同的服务形式，并进行着不同的协同计算。因此，云边协同中更深层次的服务模式值得我们探讨与研究。

本文将从云服务层次的角度出发，全面详细地分析资源协同、数据协同、智能协同、业务编排协同、应用管理协同、服务协同 6 种云边协同技术，并重点阐述协同技术的实现原理及目前的研究思路与进展，从而更好地为用户提供服务。本文的主要贡献如下：

(1)全面综述了云边协同技术，从 IaaS，PaaS，SaaS 3 种云服务层次详细讨论了云计算与边缘计算各个层次之间的协同技术原理和研究思路与进展，从而为云边协同技术的应用提供更加全面的保障。

(2)考虑到协同技术中所用到的具体方法，使用了传统方法与新兴方法，如容器、深度学习、微服务等进行分析，并从时延、能耗以及其他性能指标方面对结果进行总结对比。

(3)通过对协同技术已有的研究进行探索，总结了目前在协同过程中设备异构、计算框架与安全性 3 个方面存在的挑战和未来的研究方向。

本文第 2 节介绍云边协同的参考框架；第 3 节对协同技术的研究现状进行分析；第 4 节指出云边协同存在的挑战和未来的研究方向；最后总结全文。

2 云边协同参考框架

在云边协同的过程中，边缘计算主要对需要实时处理的数据进行处理，并为云端提供高价值的数据；云计算则负责非实时、长周期数据的处理，并完成边缘应用的全生命周期管理[7]。云边协同的总体参考架构如图 1 所示，其中的边缘节点包括：

(1)边缘基础设施能力 EC-IaaS。一般由无线基站、小型数据中心(边缘服务器)、边缘节点和 EC-IaaS 接口组成[8]，主要提供计算、存储、网络以及虚拟化资源。

(2)边缘平台能力 EC-PaaS。将为边缘云环境提供 PaaS 功能，主要进行数据的预处理与分析，以及应用的部署与编排。

(3)边缘应用能力 EC-SaaS。将 SaaS 功能扩展到边缘上，最大程度地为应用程序提供服务。

云端包括：

(1)基础设施能力 IaaS。提供计算、存储、网络和虚拟机等基础设施。

(2)平台能力 PaaS。提供设备管理、资源管理、数据处理与分析、数据建模与分析、服务组件、边缘管理与业务编排等功能[7]。

(3)应用能力 SaaS。将应用作为服务提供给用户。

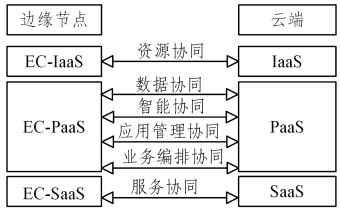

图 1 云边协同参考框架

Fig. 1 Reference framework of cloud edge collaboration

3 协同技术研究现状

边缘计算产业联盟认为云边协同技术涉及到边缘端与云端基础设施即服务(Infrastructure as a Service，IaaS)、平台即服务(Platform as a Service，PaaS)、软件即服务(Software as a Service，SaaS)各层面的协同[7]。边缘 IaaS 与云端 IaaS 之间可实现对网络、虚拟化资源、安全等的资源协同；边缘 PaaS 与云端 PaaS 之间可实现数据协同、智能协同、业务编排协同、应用管理协同；边缘 SaaS 与云端 SaaS 之间可实现服务协同。因此，云边协同主要包括 6 种协同技术：资源协同、数据协同、智能协同、业务编排协同、应用管理协同以及服务协同。

本节主要围绕以上 6 种协同技术的研究现状展开分析，重点阐述每种协同技术的原理和研究思路与进展。对于每种协同技术，我们将从边缘端和云端两端分别分析它们在协同

过程中所起的作用以及所用到的方法，目的是通过对两端底层原理的详细阐述更好地理解对应的协同技术。

3.1 资源协同

资源协同：边缘节点提供计算、存储、网络、虚拟化等基础设施资源，具有本地资源调度管理能力，也接收并执行云端的资源调度管理策略；云端则提供资源调度管理策略，包括边缘节点的设备管理、资源管理以及网络连接管理[7]。

3.1.1 计算卸载

近年来，人们提出了两种计算卸载方法：卸载到云端和卸载到边缘端。其中，卸载到云端允许用户将计算密集型任务卸载到资源强大的云服务器上进行处理；卸载到边缘端是在网络边缘部署云服务，以提供相邻的 IT 服务环境[4]。许多研究者利用这两种方法来研究时延最小化和能耗最小化问题。然而，卸载到云端会因传输距离远而无法很好地处理时延敏感性应用，因此目前大多数研究是把计算任务卸载到边缘端。

卸载到边缘端并非易事，需要考虑的因素很多，如计算/存储资源、能量消耗以及时延等。Zhang 等[9]为了降低移动设备的总时延，利用 Lyapunov 优化框架来研究任务卸载和数据缓存模型。Xu 等[10]为了降低边缘服务器的能耗，根据数据的流行程度进行缓存和卸载，以最大程度地减少由计算重复任务引起的数据传输延迟，同时保持较低的能耗。然而，目前的研究都建立在一个隐式的假设中，即边缘端能够处理所有类型的协作任务，忽略了边缘端实际的可行性。

据相关研究报道，云端与边缘端的协同有望弥补现有云计算传输距离远和边缘资源受限的不足，但这种卸载技术的协同作用还没有得到充分证实。Ren 等[11]制定了一个联合通信和计算资源分配计划来研究云边协同的计算卸载，以实现所有设备加权时延最小化。图 2 给出了云边协同的系统架构。该架构由一个集中式云服务器、多个基站（Base Station，BS）以及多个移动设备组成。每个 BS 与其 MEC 服务器的组合被认为是边缘节点。

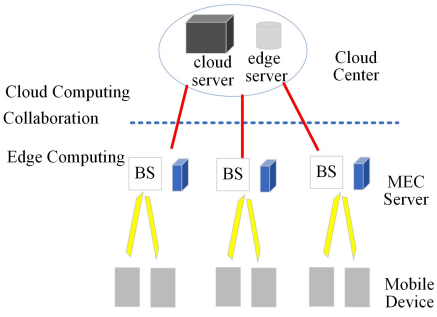

图 2 云边缘协作系统

Fig. 2 Cloud-edge collaboration system

将任务卸载到相应的边缘节点后，由边缘节点确定自行处理还是由边缘节点和云服务器协同处理。在云边缘协作的过程中，存在着以下 4 种时延。

（1）移动设备的传输延迟（第 i 个设备将计算任务卸载到所连接的第 j 个 BS 的平均传输延迟）

$$t_{j,i}^{tran,d}=\frac{L_{j,i}T}{R_{j,i}T_{j,i}} \tag{1}$$

其中，T 代表一个 TDMA 帧的长度，$R_{j,i}=E_h\{r_{j,i}\}$ 是期望的信道容量，$T_{j,i}$ 代表第 i 个设备传输数据的时间。

（2）边缘节点的计算延迟

$$t_{j,i}^{comp,e}=\frac{\lambda_{j,i}L_{j,i}C_{j,i}}{f_{j,i}^{e}} \tag{2}$$

其中，$\lambda_{j,i}$ 为任务分割率，$\lambda_{j,i}L_{j,i}C_{j,i}$ 为边缘节点处理数据所需的 CPU 周期数。

（3）边缘节点的传输延迟

$$t_{j,i}^{tran,c}=\frac{(1-\lambda_{j,i})L_{j,i}}{w_j} \tag{3}$$

其中，w_j 表示与第 j 个边缘节点关联的每个设备的回程通信容量。

（4）云服务器的计算延迟

$$t_{j,i}^{comp,c}=\frac{(1-\lambda_{j,i})L_{j,i}C_{j,i}}{f_{j,i}^{c}} \tag{4}$$

其中，$f_{j,i}^{c}$（以 CPU 周期为单位）表示云计算资源，该资源被分配给为第 j 个边缘节点服务的第 i 个设备，那么第 j 个边缘节点所服务的第 i 个设备的总延迟为：

$$T_{j,i}=t_{j,i}^{tran,d}+\max\{t_{j,i}^{comp,e},t_{j,i}^{tran,e}+t_{j,i}^{comp,c}\} \tag{5}$$

因此所有设备的总延迟为：

$$P_1:\min_{\{T_{j,i},\lambda_{j,i},f_{j,i}^{c},f_{j,i}^{e}\}}\sum_{j=1}^{J}\sum_{i=1}^{I_j}\beta_{j,i}T_{j,i} \tag{6}$$

其中，$\beta_{j,i}$ 是权重因子，代表移动设备之间的公平性，且 $\sum_{j=1}^{J}\sum_{i=1}^{I_j}\beta_{j,i}=1$。将 P_1 等效地分解为两个子问题 P_2（每个移动设备和所连接的 BS 的加权和延迟最小化）和 P_3（每个边缘节点和云服务器的加权求和延迟最小化），然后求解 P_2，得出最优的通信资源分配解决方案，通过 Karush-Kuhn-Tucker（KKT）条件间接求解 P_3，得出最佳计算资源分配策略。这种新型的云边协同卸载方法有效地利用了云端与边缘端的资源，从而减少了所有移动设备的总延迟。

3.1.2 资源管理策略

资源管理策略一直都是学术界和产业界的研究重点，包括边缘端的本地资源管理与云端的资源管理。Li 等[12]基于自回归移动平均和 BP 神经网络预测边缘端的资源负载情况，并根据负载情况灵活进行资源管理以提供弹性资源，最大程度地降低边缘云集群的成本。此外，他们考虑了资源缩减可能带来的数据丢失问题，弥补了当前资源管理的空缺。尽管边缘节点能够利用本地资源管理策略并对降低时延与提高资源利用率产生一定的效果，但相比边缘端有限的资源，云端强大的计算能力使得其同样有市场需求。例如在智能城市中，云服务的系统资源需要一个统一的资源分配系统来传达每个原始独立资源子系统的实时信息，并实现多种类型资源的最佳配置。

边缘计算与云计算协同的资源管理策略对于数据密集型计算非常重要。例如，智能交通、智慧城市以及无人驾驶汽车等都需要减少服务器延迟并提高 Qos。Li 等[13]采取云边协同的资源调度方法来弥补边缘端资源的不足。将云数据中心的资源调度到每个边缘服务器的内存或磁盘中以满足边缘对任务所需资源的分配，一定程度上提高了云端资源利用率，而

与此同时，降低能耗开销也是至关重要的。Li 等[14]根据预测的工作量与边缘节点的价格来计算租用节点的财务成本，然后通过惩罚函数进行资源的约束，以达到降低远程云计算和存储资源成本的效果。

3.2 数据协同

边缘节点负责终端设备数据的收集，并对数据进行初步处理与分析，然后将处理后的数据发送至云端；云端对海量的数据进行存储、分析与价值挖掘[7]。

3.2.1 数据收集与处理

在早期，主要的数据收集方法是多跳收集方法和设备辅助数据收集方法。然而，前者会导致能耗不平衡，如果中继节点距离接收节点较近，则会消耗更多的能量；后者则会导致严重的数据收集延迟。最近，研究者开始采用两者结合的方式进行数据采集。Luo 等[15]先通过聚类算法选择聚类节点，再由聚类节点传感器收集其负责的各分节点的数据，避免遍历所有节点，从而加快了数据的采集效率。Cai 等[16]则在文献[15]的基础上采用磁感应(Magnetic Induction，MI)与声学通信相结合的方式收集数据，使得数据收集的速度更快。

数据处理大多选择直接在网络边缘的边缘端进行，将收集到的数据进行加工、整理，形成适合数据分析的样式。它主要包括数据清洗、数据转化、数据抽取、数据合并、数据计算等处理方法。通常数据都需要进行一定的处理才能用于后续的数据分析工作。Carrizales 等[17]为每个数据建立哈希索引，以标示重复数据并将其删除，这不仅减少了重复数据上传的成本，还提高了数据准备的效率。Lopez 等[18]使用特征选择算法来优化筛选特征子集并进行归一化，然后对数据进行快速分类，从而提高数据处理的精度并减少处理时间。而对于需要实时处理的数据，Zhao 等[19]对传感器数据到达的速率变化进行监控和预测，并根据变化进行批间隔压缩调整，减少了端到端时延。

3.2.2 数据存储与分析

边缘服务器收集与处理完数据之后，通常还需要将数据发送至云端，由云端存储与分析。这样既能快速响应设备的请求，又能为数据提供大量的云存储。Tao 等[20]允许边缘服务器实时处理 IoT 数据，将其存储在云服务器上并对存储数据进行加密，生成对称密钥密文，以便在保证数据安全的同时降低用户的通信开销和计算成本。

深度学习是目前云端较为流行的数据分析技术，旨在从数据集中挖掘隐藏信息，并将此信息转换为已知结构[21]。Google 利用深度学习算法来实现 Google 的翻译器、图像和视频搜索以及 Android 的语音识别；微软和 IBM 等公司也正在利用深度学习技术进行数据分析[22]。Chatterjee 等[23]利用深度学习来进行数据的分析与挖掘，并获得了最佳性能。然而深度学习在有着强大分析能力的同时，或多或少也存在着不足。Chen 等[24]提出了一种基于深度学习的云工作负载预测算法，借助稀疏自动编码器从高维的数据中提取特征，并将稀疏自动编码器和门控循环单元集成到循环神经网络(Recurrent Neural Network，RNN)中，以弥补 RNN 对内存的长期依赖性的不足。Tou 等[25]在工业机床场景下提出了一种云和边缘之间数据协作的方法，其协作框架如图 3 所示。

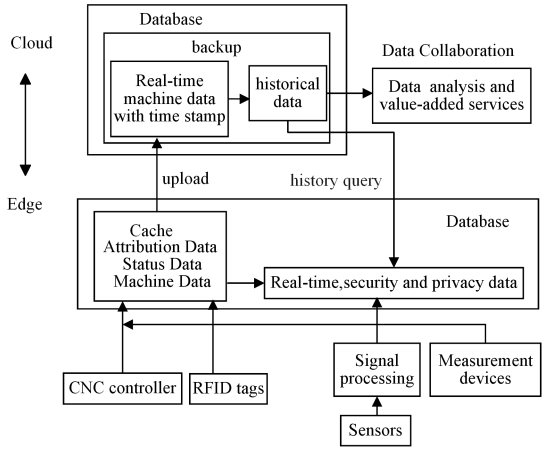

图 3 云边数据协同原理

Fig. 3 Collaboration principle of cloud edge data

该方法通过 CNC 控制器、RFID、传感器和测量设备来获取数据，获取后的数据全部由边缘进行处理和缓存。然后将机床的所有数据用于构建数据库，且定期传输到云端进行存储。云接收从边缘上传的数据，将其存储在数据库中，并将带有时间戳的处理数据存储为历史数据，将部分数据实时传输到云中以进行模型训练。而实时、安全性和隐私性数据类的部分数据存储在边缘上，方便及时处理并迅速作出响应。该数据协作方案减轻了网络传输的负担和边缘存储的压力，缩短了实时数据的响应时间，同时提高了数控机床的智能化程度。

3.3 智能协同

边缘端负责深度学习模型的推理，实现分布式智能；云端负责深度学习模型的集中式训练，然后将训练好的模型下发至边缘端[7]。

3.3.1 推理优化技术

以往的研究都是在云端进行人工智能模型的训练、优化和推理，原因是模型的训练和优化需要大量的资源，云则是良配。然而，模型的推理需要的资源比训练、优化少得多，并且有新数据需要实时推理。因此，在边缘侧进行模型的推理已逐渐成为新的研究热点。考虑到边缘节点的计算存储资源有限，如何减小模型大小和优化模型在边缘推理中显得尤为重要。

模型压缩是目前主流的模型推理优化技术，相关的研究有很多。一些研究学者通过提升网络剪枝的剪枝率来压缩模型；也有一些研究通过减少网络参数量来压缩深度神经网络。Huang 等[26]采用多目标优化思想，用卷积神经网络进行压缩学习，在优化错误率和压缩率以获得更高压缩率的同时，通过将压缩结果再训练来保证原始网络的准确性。实验结果显示，该方法不但减少了传统网络压缩中的巨大计算量，还降低了模型的推理时延。

模型分割技术与提前退出机制也被广泛用于优化模型和加快模型的推理。Kum 等[27]设计了具有统一接口可访问 AI 模型的基本容器，从而有利于将 AI 应用程序转换为微服务，随后按顺序分割模型，以加快模型的推理速度，达到降低时延的目的。Teerapittayanon 等[28]利用提前退出机制，避免了对

所有层进行逐层处理，缩短了大多数样本推理的运行时间并减少了能量使用。考虑到模型分割技术与提前退出机制的优点，一些研究将两者结合起来，以达到更好的模型优化效果。Li 等[29]对任意神经网络结构的每层延迟和能量消耗进行建模以决定 DNN 的划分，通过从模型中提前退出来加速 DNN 推断，以进一步减少计算延迟。实验结果表明，该方法在处理一些关键任务应用时能在边缘侧实现低延迟和高效能的 DNN 推理。

3.3.2 模型训练优化

在云端进行模型的训练一直是普遍采用的方法。深度神经网络模型的训练需要大量的训练数据且耗费大量的时间和计算资源，而云端丰富的计算存储资源为其提供了优质的保障。但如何在训练过程中使模型更准确、训练速度更快也一直是研究人员研究的重点。

提高模型精确度主要集中在提高模型的训练效率和误差最小化两方面。大多数深度学习模型的训练依赖于反向传播算法来迭代更新权重或偏置。为此，研究反向传播中梯度下降算法的优化或者变体来加速大型深度神经网络的优化是一个研究热点。为了提高神经网络模型的训练效率，Yan 等[30]用 CNN 提取数据特征，用 LSTM 构建编解码网络，并利用基于 Map-Reduce 的批量梯度下降算法进行模型的训练优化，更新权重，使得模型训练的效果更好。为了使训练误差最小化，且考虑到仅基于误差最小化可能会产生泛化性能不足的问题，Senhaji 等[31]将多层神经网络的训练看作一个优化问题，利用 Pareto 实现多层感知器神经网络的多目标优化，使得模型训练在准确度和复杂度之间取得平衡。

联邦学习是一种新兴的机器学习技术，通过在边缘设备上进行分布式训练来提高模型训练的性能，该技术为移动设备提供了数据的隐私保护。现有的大多数研究都集中于设计高级学习算法以优化模型的训练效果。Kang 等[32]利用激励机制来激励具有高质量数据的移动设备来进行可靠的联邦学习训练，有效地提高了模型的准确度。Ding 等[5]先利用历史数据在云端对模型进行训练，然后将较小的网络层 EdgeCNN 共享至边缘，由边缘利用新收集的实时数据进行模型的优化，以达到减少传输时延、提高用户体验的效果。EdgeCNN 初始化的损失函数为：

$$f_e(W_{e\text{-}mc})=\frac{1}{N}\sum_{i=1}^{N}H(y_i,\sigma(f(x_i;W_{e\text{-}mc})))$$

其中，W_e 表示 EdgeCNN 的权重参数，$W_{e\text{-}mc}$ 表示除去 m 层剩余的 EdgeCNN 权重参数。

EdgeCNN 更新阶段的损失函数为：

$$f_e(w'_{e\text{-}mc})=\frac{1}{N+M}\sum_{i=1}^{N+M}H(y_i,\sigma(f(x_i;w'_{e\text{-}mc})))$$

其中，$w'_{e\text{-}mc}$ 代表更新后的边缘服务器自己独有的网络结构的权重参数值，M 为新的标记对象。

3.4 业务编排协同

边缘节点提供模块化、微服务化的应用实例；云端提供按照客户需求实现业务的编排能力[7]。

3.4.1 应用实例

与传统的集中式处理的大型应用程序不同，微服务可以实现功能模块之间的完全解耦。边缘节点通过模块化分解，将一个对象分解为具有特定接口的单个模块。每个模块服务于一个特定的功能，通过指定的接口进行连接，而每个功能都有多个应用实例。用户通过在服务器调用应用实例来访问服务实现请求，具有相同功能的多个服务实例可以部署在不同的服务器上，使得边缘服务器提供应用实例成为快速响应用户请求与时延敏感性应用程序的不二之选。

通常情况下，应用程序供应商租用大量边缘服务器部署微服务实例，以提供更好的用户体验。在 MEC 体系结构中，邻近的服务器作为一个平台来协同工作，以集成它们的计算和存储能力。随着移动设备的增加，大量设备会产生大量的数据流量。对于数据密集型应用，若将所有内容都传到远程云数据中心，将消耗大量的网络带宽。为了弥补这缺点，学术界和行业的研究人员都希望将内容和基础设施推向移动用户附近，以减轻网络压力。为此，通过在附近的分布式边缘服务器上部署大量微服务实例，可以大大减少数据传输延迟。Deng 等[33]根据应用程序的预期响应时间和成本选择应用实例的部署方案。然而，该方案忽略了上下文和服务器的负载情况，导致云服务提供商难以根据用户的喜好在边缘处部署合适的微服务实例。此外，同样忽略了上下文和服务器负载情况的还有 Spring Cloud 和 Dubbo 两大主流框架中的负载均衡算法。Shao 等[34]基于上述经验和不足，动态获取了物理、服务和用户 3 种上下文数据库的权重信息，以此平衡主观和客观因素对微服务实例的影响，进而提高微服务实例选择的准确性并缩短响应时间。

3.4.2 业务编排

业务编排是在多个功能模块中提供业务流程，并按照不同的业务流程对应用进行编排、连接、监控和管理。企业通过业务流程使用 API 接口或其他通信协议/通道来编排软件应用程序。目前，业务流程由一系列微服务实现，微服务通过远程 API 调用形成微服务链，从而提高软件更新和服务交付的灵活性、可靠性和速度[35]。通过微服务之间的协作实现一个完整的业务流程。随着业务流程迁移到云中，开发人员不仅需要协调将代码部署到云资源的过程，还需要协调代码在云平台上的分布[36]。

一些现有的业务流程使用逻辑集中的业务流程协调器来管理应用程序的变化。然而随着服务器数量的增加，管理应用程序变得困难。Kiss 等[37]利用基于容器的开源云技术创建了一个基于微服务应用的动态编排框架 MiCADO。在动态编排时，首先通过 API 将成本和性能优化机制构建到应用程序代码中，然后不断收集应用程序指标并传递给编排组件，进而完成应用运行的编排，同时也为应用程序提供动态和自动化的资源供应。图 4 给出了 MiCADO 框架的编排层。

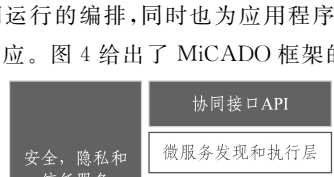

图 4 MiCADO 框架的编排层

Fig. 4 Choreography layer of MiCADO framework

其中，协调应用程序接口（Application Programming Interface，API）是提供对业务控制流程的访问，并通过这些接口来调用应用程序；微服务发现和执行层管理微服务的执行，并跟踪服务的运行情况；微服务协调逻辑层负责应用实例的关闭和开启；云接口 API 用于抽象云访问，执行更改操作；安全、隐私和信任服务层进行安全策略管理，以在业务流程上实施安全策略。实验结果证明了该框架能够扩展和缩减应用程序集群，在一定程度上弥补了传统软件设计将所有模块作为一个整体进行打包和部署的缺陷，从而更好地进行应用编排。

3.5 应用管理协同

边缘节点提供应用部署与运行环境，并对本节点多个应用的生命周期进行管理调度；云端提供应用开发和能力测试，实现对边缘节点应用的生命周期管理[7]。

3.5.1 系统与应用部署

最近有些研究关注边缘计算应用的部署与运行环境，这是因为边缘计算提供分布式云计算功能和高性能、低延迟、高带宽的服务环境。然而，应用类型和所处边缘位置的不同，会导致边缘计算硬件的选择部署和提供的环境不同。例如，边缘用户端节点设备采用低成本、低功耗的 ARM 处理器或者英特尔 Atom 处理器；边缘基站服务器采用英特尔至强系列处理器。而微服务、容器及虚拟化技术等的云原生软件架构能够减少因硬件基础设施差异化带来的部署及运维问题。

如今，越来越多有影响力的 IT 公司或应用程序供应商开始借助微服务技术（如 Kubernetes）来开发复杂的应用程序。此外，借助基于容器的技术，这些基于微服务的应用程序可以轻松地部署在边缘服务器上[33]。考虑到物联网应用程序是分布式的，且通常包含多个计算密集型任务，因此需要在每个单独的计算节点上完成应用程序的部署与管理。研究者一直在探索如何找到一个合适的、低成本的应用程序部署方案。Xiong 等[38]介绍了边缘计算环境中的基础架构 KubeEdge。KubeEdge 通过 Kubernetes 管理远程边缘节点并使用相同的 API 将应用程序部署到边缘并进行管理，给应用程序带来了更好的性能，同时提升了用户体验。针对边缘中的动态环境，Zhang 等[39]提出了一种动态规划的编程算法，用于在运行时按需将应用程序部署在边缘环境中。Deng 等[33]将应用程序部署问题转化为部署异构微服务实例的问题，并将微服务实例服务器建模为排队节点，以微服务的计算成本和存储成本作为约束条件，达到在满足移动服务的平均响应时间的前提下减少部署成本的效果。

3.5.2 应用生命周期管理

如今，Google Play 商店中管理着超过 260 万个应用程序，其中大多数应用程序都是软件开发人员使用 PaaS 云平台服务来开发、测试和运行的。Dcoker 因能简化分布式应用程序的开发、部署和执行成为了最受欢迎的应用程序容器之一。Ozcan 等[40]通过创建 Docker，分区放置应用程序的可执行文件安全副本，避免了牵一发而动全身的麻烦，一定程度上节省了开发人员的开发时间，并为工业互联网在边缘设备远程调试应用程序提供了新的思路。

简单的应用程序很容易开发，但复杂的应用程序可包含数百个微服务，使得开发应用程序变得困难且不易扩展。微服务设计则是为了应对这种挑战而开发的一种新范例，其具有细粒度的方法和松散耦合的服务。使用微服务设计开发的应用程序可以实现更好的扩展，并以最小的成本为开发人员提供扩展的灵活性。Bao 等[41]对微服务性能建模和任务调度进行了深度的研究探索，以减少开发过程中微服务带来的性能开销和资源消耗。Sampaio 等[36]根据微服务之间的关联性和资源使用情况提出了一种 REMaP 适应机制，用于自动管理微服务在应用程序中的放置，最终可以节省多达 80% 主机的使用，从而减少资源使用并提高应用程序的性能。

3.6 服务协同

服务协同主要是指边缘计算的边缘服务与云计算的云服务进行协同。边缘节点按照云端策略实现部分 ECSaaS 服务，通过 ECSaaS 与云端 SaaS 的协同实现面向客户的按需服务；云端主要提供 SaaS 服务在云端和边缘节点的服务分布策略，以及云端承担的 SaaS 服务能力[7]。

3.6.1 边缘服务

目前，移动计算应用程序不断增长的需求已触发了从集中式移动云计算向边缘计算的转变[42]。许多云服务提供商都十分看重边缘计算，并且提出了专门针对这种新兴范例的服务。例如，阿里巴巴在中国部署了超过 500 台边缘服务器，通过分散式的部署使用户随时随地都可获得低延迟的边缘服务[43]；Google 在全球范围内部署了 1400 多个边缘服务器，并提出了针对边缘应用的解决方案；Amzaon CloudFront 提供了边缘计算应用程序所需的低延迟的内容交付网络服务。这意味着云计算提供商已准备好满足边缘计算的服务要求。

随着移动边缘基础架构和边缘设备的快速增长，边缘服务器的数量呈指数级增长，所提供的边缘服务也越来越多。Bhattacharjee 等[44]基于卷积神经网络以监控摄像头作为边缘端，提供了目标检测、识别和跟踪等服务。然而，边缘节点资源受限的问题依旧阻碍着更多服务的实现。Chen 等[45]认为，资源有限的边缘服务器之间的协作是增强边缘服务功能的有效解决方案，为此，他们将更多边缘服务功能的实现归因于边缘计算服务器（Edge Computing Service，ECS）的放置和服务功能的设置，并通过二者之间的协作来最小化部署 ECS 的数量，以达到增强服务功能的效果。Chen 等采用最小比率增加算法 MinRI 来确定 ECS 的最小放置数量，同时实现了 ECS 之间资源的补充协作，提高了资源利用率，然后通过搜索和交换算法 SeSw 来增加 ECS 的任务数量。仿真结果表明，通过在 ECS 之间建立协作，可以在保证服务质量（QoS）的前提下大大减少所需 ECS 的数量并为智能应用提供更多边缘服务。

3.6.2 服务分布策略

云计算环境中的服务分布策略是一个非常热门的话题，对此的研究有很多。然而，云服务离用户较远，服务请求时延和用户体验质量得不到保证。新兴的时延敏感性服务需要借助边缘服务低延迟的特性来降低时延，提高用户体验，因此仅仅分配云服务是不够的。Lai 等[46]基于之前的启发式方法的不足，提出了一种新的启发式算法 QoEUA 来解决边缘服务

的分配问题。但遗憾的是，QoEUA 没能找到最优的服务分配策略。Bonadio 等[42]则重点关注在有限区域内为用户服务提供计算能力的 MEC 系统，并通过马尔可夫求解最佳服务分配策略，利用最少的资源达到节省能耗的效果。

对于多种多样的应用服务，边缘节点与云服务协同可作为一种有效的新型计算模型，以创建更好的服务价值。Huang 等[47]对应用程序的计算消耗、通信消耗以及等待时间进行建模，根据整个任务的负载结果对具有不同资源需求和时延敏感性的应用实施不同的服务分布处理。将时延敏感性应用分布到边缘，将计算型应用分布到云端，以达到减少等待时间和能耗的效果，满足了不同应用的需求。考虑到云边混合环境，Chen 等[48]将模拟退火算法引入蚁群算法以更新信息素矩阵，有效地避免了局部最优的问题，求出了最优的策略，从而为密集型服务组件分布最佳服务，同时也减小了数据传输时延。Chen 等[49]综合考虑边缘服务器容量的不同、请求数量的变化、应用程序的复杂结构以及不断变化的地理环境等，提出了多缓冲区深度确定性策略梯度（MB_DDPG），利用强化学习和神经网络来学习服务分布策略，达到减少设备平均等待时间的效果。

3.7 协同技术的对比

本节对资源协同、数据协同、智能协同、业务编排协同、应用管理协同和服务协同 6 种技术进行归纳总结，然后根据每种协同技术的目的和所用的方法从能耗、时延以及其他性能指标进行对比分析。表 1 列出了 6 种云边协同技术的对比。

表 1 云边协同技术的对比

Table 1 Comparison of cloud-edge collaborative technologies

协同技术	文献	云边协同目的	云边协同实现方法	特点
资源协同	[11]	解决边缘节点资源受限问题	将联合通信和计算资源分配问题转化为凸优化问题，根据 KKT 条件和拉格朗日乘数法求出最优资源分配策略	通过云和边缘之间的有效协作减少了所有移动设备的总体延迟，提升了系统性能，但忽略了每个单一用户对网络的延迟要求
	[12]	解决边缘云资源的扩张与收缩问题	利用自回归移动平均（ARIMA）和 BP 神经网络预测集群负载，避免资源浪费	能够在云边缘环境中降低边缘云架构的成本，并且解决了资源缩减导致的数据可靠性问题
	[14]	最大程度地减少中央云集群的开销	提出了基于禁忌算法（Tabu Search）的资源管理策略和基于 TOPSIS 方法的动态副本分配方法	能够为边缘云环境中突变的工作负载租用云资源，在保证服务质量的同时有效地减少了数据传输的时延和存储开销，但是没有考虑副本一致性问题对资源分配的影响
数据协同	[21]	从大量异构数据中有效地提取数据感知智能功能，以便向最终用户提供实时数据分析和反馈	提出了一种新颖的云边协同框架，用于处理无线物联网网络中的实时数据分析	加快了对实时数据的分析速度，且保证了能耗和时延之间的平衡。但是在边缘侧处理信息时没有考虑通过获取上下文信息来优化数据处理
	[25]	改善数控机床在状态感知、数据处理和实时反馈方面的缺陷	提出了一种基于云边协作的智能机床新架构	减小了云端的网络堵塞压力和传输时延，但没有考虑到数据种类繁多对标准协议转换的困扰
智能协同	[5]	弥补先前的移动应用程序在边缘云计算环境中通信延迟和自动升级方面的不足	提出了一种基于深度学习浅层卷积神经网络的边缘云协作框架，通过云端共享其较小的层来协助边缘端	能够缩短认知服务的平均响应时间并提高网络模型的准确性，但是忽略了环境等干扰因素对重新训练时资源分配的影响
	[27]	更好地将 AI 应用程序部署在边缘上	将 AI 应用程序转变为微服务的容器结构，提出了一种新的 DNN 部署方法	通过云和边缘的协作部署使得应用程序部署的准确度没有损失，但由于云边需要中间数据的传输，导致端到端的服务延迟增加
业务编排协同	[37]	解决应用实例的管理随着服务器数量的增加而变困难的问题	创建一个基于微服务应用在云环境中动态编排的 MiCADO 框架	灵活进行应用的编排
应用管理协同	[31]	减少微服务带来的昂贵的性能开销	对基于微服务的应用程序性能进行建模和预测，然后针对微服务应用程序的工作流调度问题提出了一种基于启发式算法的微服务调度算法	提高了基于微服务的应用程序的性能，减少了端到端延迟。但是对于每个微服务实例仅创建或调用一个实例，没有考虑实例复制所带来的安全性问题
	[33]	在资源受限的边缘节点部署基于微服务的应用	针对微服务的组合对应用进行建模，将应用部署问题转化为异构微服务实例的部署问题，提出了一种基于容器技术优化应用部署成本的方法	缩短了移动服务的平均响应时间，但是忽略了应用部署所需的成本
服务协同	[47]	为具有不同资源需求和时延敏感性的任务分配服务	对能耗和等待时间进行建模优化并估算成本，然后基于 SDN 技术的编排数据即服务机制，直接在边缘层对 MEC 服务器收集到的元数据进行编排处理	减小了网络带宽的压力并降低了时延
	[48]	为资源受限的边缘节点提供更多的服务	将服务部署问题转化为约束满足问题，利用最小资源比率增加算法来最小化服务部署，减少所需边缘服务器的数量	同时部署边缘计算服务器和服务功能，以减少能耗和提高资源利用率。但是，目前仅考虑了 CPU 和内存的消耗情况，忽略了系统延迟和用户服务质量的影响

4 未来发展

目前，围绕云边协同应用展开研究的文献有很多，它们也给出了相应的算法、技术和架构，展示了不同的效果与性能提升。然而，在云边协同过程中，仍然存在着一些急需解决的问题，如设备异构、计算框架与安全性。下面通过对已有研究的分析与总结，指出了一些需要进一步改进的地方和未来的研究方向。

(1)设备异构。目前针对移动设备的研究大多没有考虑设备异构的特性，而通常异构设备之间的计算能力和通信资源存在巨大的差异，导致将相同的方法应用于不同的设备会获得不同的结果。为此，调节异构设备之间的兼容性、协调性以及合理地管理和分配资源至关重要。Xu 等[50]通过屏蔽神经元的数量来确保模型训练所需的计算量，并通过异步联邦学习来调节训练速度，有效地解决了边缘设备异构导致的计算能力不同。Singh 等[51]基于 Kruskal 的聚类算法来管理异构的电池储能设备，保证了资源的平均分配和能量的负载平衡。此外，考虑到移动设备自身能力的不断增强，在大量的异构设备之间建立灵活的协调机制以保证异构设备之间的相互通信和协作也是值得我们探索的一条路径，与此同时，统一的 API 接口也是研究协同机制需要考虑的问题。

(2)计算框架。随着边缘计算和云计算技术的发展，云边协同框架作为一种有效的新型计算架构被提出，并在许多领域得到应用。例如，Gang 等[52]介绍了计算框架在电力负荷监控领域的应用，提出了一种在边缘进行处理、在云端进行优化的云边协同框架，有效地解决边缘端计算资源弱和云端通信压力大的矛盾；Ding 等[53]介绍了计算框架在医疗保健领域的应用，设计了一个在边缘进行病变检测、在云端进行高性能建模的云边缘协同框架，能够在离线状态和在线状态两种情况下准确检测胃肠病变，从而节省了胃镜医生的时间。然而，以上 2 种应用架构都忽略了计算复杂度高和通信成本高的问题。因此，未来需要研究计算架构中的多目标优化模型与算法来提高应用的整体性能。此外，结合本地、边缘和云计算三者构建一个新的计算架构以进一步提升应用的效率和性能仍是一个开放问题，值得进一步的探讨和研究。

(3)安全性。安全性一直以来都是云计算与边缘计算的重点探索方向。Radovici 等[54]回顾了近年来与物联网相关的恶意软件和安全威胁，并提出了有效的解决方案。然而，开放的环境和新软件技术的不断引进，带来了新的威胁和安全风险。Bousselham 等[3]提出了一种新的安全机制，其主要使用诱饵技术生成虚假信息来干扰入侵者对数据的区分，从而防止入侵者的有害攻击，降低安全风险。然而，目前对云边协同计算安全性的研究远未令人满意，未来的研究需要填补现有的漏洞并排除隐患。一方面，需要更强大的安全防御解决方案，尤其是预防机制，以减少个人攻击；另一方面，结合安全机制以更统一的方式保护整个安全防御系统值得探索。虽然云边协同安全性的研究和开发仍处于起步阶段，但在新兴应用程序以及现代加密技术的推动下，确保云边协同安全的创新设计和实现将在可预见的未来蓬勃发展。

结束语 云边协同是一种新型计算范式，其将云计算强大的资源能力与边缘计算超低的时延特性结合起来，实现了边缘支撑云端应用，云端助力边缘本地化需求的协同优化目标。目前的研究集中在应用的落地与行业的发展上，忽略了云边协同过程中的技术实现原理。为了全面而深刻地了解云边协同的底层实现原理，本文首先对云计算与边缘计算各个层次之间的协同技术进行分析，随后深入探讨每种协同技术的具体实现方法，并从时延、能耗以及其他性能指标方面对结果进行了对比分析，指出了云边协同目前存在的挑战与未来发展方向。我们相信，随着“新基建”的不断推进，云边协同会逐步走向成熟，为产业创造越来越多的价值。

参 考 文 献

[1] KUMAR M, SHARMA S C, GOEL A, et al. A comprehensive survey for scheduling techniques in cloud computing[J]. Journal of Network and Computer Applications, 2019, 143: 1-33.

[2] SHI W S, SUN H, CAO J, et al. Edge computing: a new computing model for the Internet era [J]. Journal of Computer Research and Development, 2017, 54(5): 907-924.

[3] BOUSSELHAM M, BENAMAR N, ADDAIM A. A new Security Mechanism for Vehicular Cloud Computing Using Fog Computing System[C]// 2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS). IEEE, 2019: 1-4

[4] REN J, HE Y, YU G, et al. Joint communication and computation resource allocation for cloud-edge collaborative system [C]// 2019 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2019: 1-6.

[5] DING C, ZHOU A, LIU Y, et al. A Cloud-Edge Collaboration Framework for Cognitive Service[J/OL]. IEEE Transactions on Cloud Computing, 2020. https://ieeexplore.ieee.org/abstract/document/8895891.

[6] ZHANG H, CHEN S, ZOU P, et al. Research and Application of Industrial Equipment Management Service System Based on Cloud-Edge Collaboration[C]// 2019 Chinese Automation Congress (CAC). IEEE, 2019: 5451-5456.

[7] Edge computing Consortium and Alliance of Industrial Internet: White Paper on Edge Computing and Cloud Computing Collaboration[EB/OL]. http://www.ecconsortium.org/Uploads/file/20190221/1550718911180625.pdf.

[8] YAMANAKA H, KAWAI E, TERANISHI Y, et al. Proximity-Aware IaaS in an Edge Computing Environment With User Dynamics[J]. IEEE Transactions on Network and Service Management, 2019, 16(3): 1282-1296.

[9] ZHANG N, GUO S, DONG Y, et al. Joint task offloading and data caching in mobile edge computing networks[J]. Computer Networks, 2020, 182: 107446.

[10] XU X, LI Y, HUANG T, et al. An energy-aware computation offloading method for smart edge computing in wireless metro-

politan area networks[J]. Journal of Network and Computer Applications，2019，133：75-85.

[11] REN J，YU G，HE Y，et al. Collaborative cloud and edge computing for latency minimization[J]. IEEE Transactions on Vehicular Technology，2019，68(5)：5031-5044.

[12] LI C，SUN H，CHEN Y，et al. Edge cloud resource expansion and shrinkage based on workload for minimizing the cost[J]. Future Generation Computer Systems，2019，101：327-340.

[13] LI J. Resource optimization scheduling and allocation for hierarchical distributed cloud service system in smart city[J]. Future Generation Computer Systems，2020，107：247-256.

[14] LI C，BAI J，CHEN Y，et al. Resource and replica management strategy for optimizing financial cost and user experience in edge cloud computing system[J]. Information Sciences，2020，516：33-55.

[15] LUO Y，ZHU X，LONG J. Data Collection Through Mobile Vehicles in Edge Network of Smart City[J]. IEEE Access，2019，7：168467-168483.

[16] CAI S，ZHU Y，WANG T，et al. Data collection in underwater sensor networks based on mobile edge computing[J]. IEEE Access，2019，7：65357-65367.

[17] CARRIZALES D，SÁNCHEZ-GALLEGOS D D，REYES H，et al. A Data Preparation Approach for Cloud Storage Based on Containerized Parallel Patterns[C]// International Conference on Internet and Distributed Computing Systems. Springer，Cham，2019：478-490.

[18] LOPEZ M A，MATTOS D M F，DUARTE O C M B，et al. A fast unsupervised preprocessing method for network monitoring[J]. Annals of Telecommunications，2019，74(3/4)：139-155.

[19] ZHAO H，YAO L B，ZENG Z X，et al. An edge streaming data processing framework for autonomous driving[J/OL]. Connection Science. https://www.tandfonline.com/doi/abs/10.1080/09540091.2020.1782840.

[20] TAO Y，XU P，JIN H. Secure Data Sharing and Search for Cloud-Edge-Collaborative Storage[J]. IEEE Access，2019，8：15963-15972.

[21] SHARMA S K，WANG X. Live data analytics with collaborative edge and cloud processing in wireless IoT networks[J]. IEEE Access，2017，5：4621-4635.

[22] JAN B，FARMAN H，KHAN M，et al. Deep learning in big data Analytics：A comparative study[J]. Computers & Electrical Engineering，2019，75：275-287.

[23] CHATTERJEE A，GUPTA U，CHINNAKOTLA M K，et al. Understanding emotions in text using deep learning and big data[J]. Computers in Human Behavior，2019，93：309-317.

[24] CHEN Z，HU J，MIN G，et al. Towards accurate prediction for high-dimensional and highly-variable cloud workloads with deep learning[J]. IEEE Transactions on Parallel and Distributed Systems，2019，31(4)：923-934.

[25] LOU P，LIU S，HU J，et al. Intelligent Machine Tool Based on Edge-Cloud Collaboration[J]. IEEE Access，2020，8：139953-139963.

[26] HUANG J H，SUN W Z，HUANG L. Deep neural networks compression learning based on multiobjective evolutionary algorithms[J]. Neurocomputing，2020，378：260-269.

[27] KUM S，KIM Y，MOON J. Deploying Deep Neural Network on Edge-Cloud environment[C]// 2019 International Conference on Information and Communication Technology Convergence (ICTC). IEEE，2019：242-244.

[28] TEERAPITTAYANON S，MCDANEL B，KUNG H T. Branchynet：Fast inference via early exiting from deep neural networks[C]// 2016 23rd International Conference on Pattern Recognition (ICPR). IEEE，2016：2464-2469.

[29] LI E，ZHOU Z，CHEN X. Edge intelligence：On-demand deep learning model co-inference with device-edge synergy[C]// Proceedings of the 2018 Workshop on Mobile Edge Communications. 2018：31-36.

[30] YAN H，YU P，LONG D. Study on deep unsupervised learning optimization algorithm based on cloud computing[C]// 2019 international conference on intelligent transportation，Big data & smart city (ICITBS). IEEE，2019：679-681.

[31] SENHAJI K，RAMCHOUN H，ETTAOUIL M. Training feedforward neural network via multiobjective optimization model using non-smooth L1/2 regularization[J]. Neurocomputing，2020，410(10)：1-11.

[32] KANG J W，XIONG Z H，NIYATO D，et al. Incentive mechanism for reliable federated learning：A joint optimization approach to combining reputation and contract theory[J]. IEEE Internet of Things Journal，2019，6(6)：10700-10714.

[33] DENG S，XIANG Z，TAHERI J，et al. Optimal application deployment in resource constrained distributed edges[J/OL]. IEEE Transactions on Mobile Computing. https://ieeexplore.ieee.org/abstract/document/8975987

[34] SHAO J X，ZHANG X G，CAO Z Y. Research on context-based instances selection of microservice[C]// Proceedings of the 2nd International Conference on Computer Science and Application Engineering. 2018：1-5.

[35] AHMAD S，KIM D H. A multi-device multi-tasks management and orchestration architecture for the design of enterprise IoT applications[J]. Future Generation Computer Systems，2020，106：482-500.

[36] SAMPAIO A R，RUBIN J，BESCHASTNIKH I，et al. Improving microservice-based applications with runtime placement adaptation[J]. Journal of Internet Services and Applications，2019，10(1)：1-30.

[37] KISS T，KACSUK P，KOVACS J，et al. MiCADO—Microservice-based cloud application-level dynamic orchestrator[J]. Future Generation Computer Systems，2019，94：937-946.

[38] XIONG Y，SUN Y，XING L，et al. Extend cloud to edge with KubeEdge[C]// 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE，2018：373-377.

[39] ZHANG J，MA M，HE W，et al. On-Demand Deployment for IoT Applications[J]. Journal of Systems Architecture，2020：101794.

[40] OZCAN M O, ODACI F, ARI I. Remote Debugging for Containerized Applications in Edge Computing Environments[C]// 2019 IEEE International Conference on Edge Computing (EDGE). IEEE, 2019: 30-32.

[41] BAO L, WU C, BU X, et al. Performance modeling and workflow scheduling of microservice-based applications in clouds[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(9): 2114-2129.

[42] BONADIO A, CHITI F, FANTACCI R. Performance Analysis of an Edge Computing SaaS System for Mobile Users[J]. IEEE Transactions on Vehicular Technology, 2019, 69(2): 2049-2057.

[43] LIANG Y, GE J, ZHANG S, et al. A Utility-Based Optimization Framework for Edge Service Entity Caching[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(11): 2384-2395.

[44] BHATTACHARJEE A, BARVE Y, KHARE S, et al. Stratum: A bigdata-as-a-service for lifecycle management of iot analytics applications[C]// 2019 IEEE International Conference on Big Data (Big Data). IEEE, 2019: 1607-1612.

[45] CHEN Y, SUN Y, FENG T, et al. A Collaborative Service Deployment and Application Assignment Method for Regional Edge Computing Enabled IoT[J]. IEEE ACCESS, 2020, 8: 112659-112673.

[46] LAI P, HE Q, CUI G, et al. QoE-aware user allocation in edge computing systems with dynamic QoS[J]. Future Generation Computer Systems, 2020, 112: 684-694.

[47] HUANG M, LIU W, WANG T, et al. A cloud-MEC collaborative task offloading scheme with service orchestration[J]. IEEE Internet of Things Journal, 2019, 7(7): 5792-5805.

[48] CHEN X, TANG S, LU Z, et al. iDiSC: A new approach to IoT-data-intensive service components deployment in edge-cloud-hybrid system[J]. IEEE Access, 2019, 7: 59172-59184.

[49] CHEN L, XU Y, LU Z, et al. IoT Microservice Deployment in Edge-cloud Hybrid Environment Using Reinforcement Learning [J]. IEEE Internet of Things Journal, 2020(99): 1-1.

[50] XU Z, YANG Z, XIONG J, et al. Elfish: Resource-aware federated learning on heterogeneous edge devices[J]. arXiv: 1912.01684, 2019.

[51] SINGH V, PEDDOJU S K. Container-based microservice architecture for cloud applications[C]// 2017 International Conference on Computing, Communication and Automation (ICCCA). IEEE, 2017: 847-852.

[52] GANGY N, LIU X S, TONG D H, et al. Non-invasive Power Load Monitoring Method Based on Cloud Edge Collaboration [C]// IOP Conference Series: Earth and Environmental Science. IOP Publishing, 2020: 012115.

[53] DING S, LI L, LI Z, et al. Smart electronic gastroscope system using a cloud-edge collaborative framework[J]. Future Generation Computer Systems, 2019, 100: 395-407.

[54] RADOVICI A, CRISTIAN R, ŞERBAN R. A survey of iot security threats and solutions[C]// 17th RoEduNet Conference: Networking in Education and Research (RoEduNet). IEEE, 2018: 1-5.

CHEN Yu-ping, born in 1995, postgraduate. Her main research interests include cloud computing and edge computing.

LIN Wei-wei, born in 1980, Ph. D, professor, is a member of China Computer Federation. His main research interests include cloud computing, big data technology and AI application technology.