计算机科学
COMPUTER SCIENCE

基于端边协同的节点部署和资源分配联合优化方法

杨哲铭, 左路路, 纪雯

引用本文

杨哲铭, 左路路, 纪雯. 基于端边协同的节点部署和资源分配联合优化方法[J]. 计算机科学, 2024, 51(11A): 240200010-7.

YANG Zheming, ZUO Lulu, JI Wen. Joint Optimization Method for Node Deployment and Resource Allocation Based on End-EdgeCollaboration [J]. Computer Science, 2024, 51(11A): 240200010-7.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240200010

基于端边协同的节点部署和资源分配联合优化方法

杨哲铭[1,3]　左路路[1,2,3]　纪　雯[1,2]
1 中国科学院计算技术研究所　北京 100190
2 鹏城实验室　广东 深圳 518055
3 中国科学院大学计算机科学与技术学院　北京 100049
(yangzheming19b@ict.ac.cn)

摘　要　随着物联网技术的快速发展,边缘计算作为一种重要的数据处理方式,在多样化的应用场景中展现出其独特的优势。在这种背景下,边缘服务器的位置部署和资源分配成为了提高任务处理效率的关键因素。然而,这一过程面临着终端设备的广泛分布和边缘服务器的异构特性带来的显著挑战。为了有效解决这些问题,提出了一种基于端边协同的节点部署和资源分配联合优化方法,旨在全面提升边缘计算系统的整体性能。首先,利用分层聚类算法,根据终端设备在功能和地理位置上的相似性,将它们有效地划分为若干个区域。随后,根据边缘服务器的处理能力、存储空间和网络带宽等关键指标,决定每个区域内最适合的边缘节点。最后,通过联合优化节点部署和资源利用率来指导任务的分配情况。为了验证所提方法的有效性,在公开数据集上对不同方法进行了仿真实验。实验结果表明,相比现有方法,所提方法可以将负载均衡水平提升30%以上,并将任务处理的时延和能耗降低10%以上,这在提升边缘计算系统的可持续性和效率方面具有重要意义。

关键词:边缘计算;节点部署;资源分配;负载均衡;任务卸载
中图分类号　TP 311.5

Joint Optimization Method for Node Deployment and Resource Allocation Based on End-Edge Collaboration

YANG Zheming[1,3],ZUO Lulu[1,2,3] and JI Wen[1,2]
1 Institute of Computing Technology,Chinese Academy of Sciences,Beijing 100190,China
2 Pengcheng Laboratory,Shenzhen,Guangdong 518055,China
3 School of Computer Science and Technology,University of Chinese Academy of Sciences,Beijing 100049,China

Abstract　With the rapid development of IoT technology,edge computing shows its unique advantages in diverse application scenarios. The location deployment and resource allocation of edge servers become the key factors to improve the efficiency of task processing. However,this process faces significant challenges due to the wide distribution of end devices and the heterogeneous nature of edge servers. To effectively address these issues,this paper proposes a joint optimization method for node deployment and resource allocation based on end-edge collaboration,aiming to improve the overall performance of edge computing systems comprehensively. Our approach first utilizes a hierarchical clustering algorithm to effectively divide end devices into several regions based on their functional and geographic similarities. Subsequently,based on key metrics such as processing power,storage space,and network bandwidth of edge servers,the most suitable edge nodes in each region is decided. Finally,allocating tasks is guided by jointly optimizing node deployment and resource utilization. To validate the effectiveness of the proposed method,we conduct simulation experiments of different methods on public datasets. Experimental results show that our proposed method can improve the load balancing level by more than 30% and reduce task processing latency and energy consumption by more than 10%,compared to the existing methods.

Keywords　Edge computing,Node deployment,Resource allocation,Load balancing,Task offloading

1　引言

随着物联网的快速发展,部署在各类场景中的智能终端设备越来越多。据GSMA预测,世界上物联网设备的数量将在2025年达到246亿[1]。最近,视觉物联网在智慧城市中发挥着越来越重要的作用[2],提供了海量的实时视觉数据,这些数据对于提高城市运行效率和居民生活质量至关重要。然而,视觉数据的庞大体积和复杂性给数据处理带来了显著的挑战。最初的解决方案是通过云计算集中这些数据的处理。尽管云计算在处理大数据和复杂计算任务方面具有优势,但

基金项目:国家重点研发计划(2023YFB4502805);国家自然科学基金(62072440);北京市自然科学基金(L221004)
This work was supported by the National Key R&D Program of China(2023YFB4502805),National Natural Science Foundation of China(62072440) and Beijing Natural Science Foundation(L221004).
通信作者:纪雯(jiwen@ict.ac.cn)

由于数据中心与终端设备之间的物理距离较远，网络时延相对较大。边缘计算正在成为解决此问题的关键方法。边缘计算的概念在于将计算资源部署到接近终端设备的位置，以减轻云计算中心的负荷[3]，从而减少任务处理的时间，并提升用户体验。

边缘节点的位置部署和资源分配是边缘计算系统中的两个重要问题。但是近年来，许多研究人员认为已放置了边缘节点。他们只专注于资源分配，而忽略了节点位置的重要性[4]。实际上，边缘节点部署是资源分配的基础。边缘节点部署是指在靠近数据源或用户终端的位置放置服务器，以提高数据处理速度，如图 1 所示。良好的边缘节点部署方法可以有效地满足任务处理的低时延和高带宽等需求。合理的节点部署策略可以首先确保终端设备的访问时延。长时间的访问时延很难满足用户体验质量的需求[5]。其次，它可以平衡节点之间的负载并改善每个节点的资源利用率[6]。有时，某些节点的过载会导致整体任务失败。另外，它还可以减少节点的总能源消耗，并防止空闲节点导致的能源浪费。

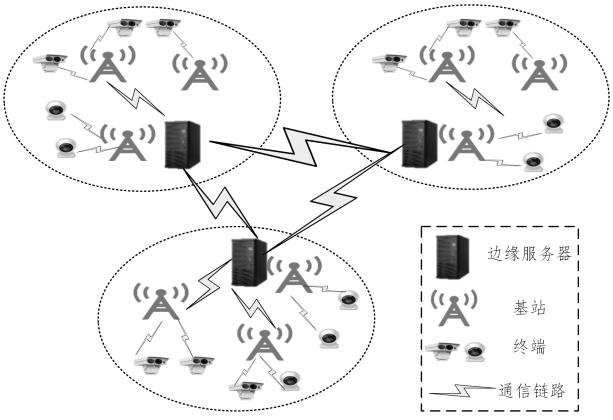

图 1 边缘节点部署示例

Fig. 1 Example of edge node deployment

但是，有效的边缘节点部署方案极具挑战性。边缘节点的位置需要从大量基站中选择。尽管可能有最佳的离散节点部署方案，但很难在如此大的搜索空间中找到它。为了寻找最优的边缘节点部署策略，众多研究者采用了混合整数规划[7]、启发式算法[8]、聚类[9]等方法，并在一定程度上取得了积极的成果。这些方法在处理边缘计算环境中的节点部署问题时，展现了其独特的优势和潜力。然而，尽管它们在多个方面表现出色，但其在处理异构边缘节点的部署问题时仍显示出一些不容忽视的局限性。这些现有方法在设计时往往没有充分考虑到边缘计算环境中的复杂性和多样性。特别是，它们通常忽视了不同区域内终端设备分布的不均匀性问题。在现实场景中，终端设备的分布往往是非均匀的，且受到多种因素的影响，如地理位置、用户行为模式以及设备类型等。这种分布的不均匀性可能导致资源分配不均衡，从而在节点部署过程中造成效率损失。例如，如果一个部署策略仅基于简单的地理或网络拓扑信息，而没有考虑到终端设备的实际使用模式和需求，可能会导致某些区域的边缘节点过载，而其他区域资源则闲置。这种资源分配的不均衡性不仅降低了整体网络的性能，也增加了任务处理的时延和能耗。

为了应对上述挑战，本文提出了一种基于端边协同的节点部署和资源分配联合优化方法。首先，通过分层聚类算法分析终端设备的特征相似性，并将其分为不同的区域。这种区域划分有助于更精确地理解和响应不同区域的特定需求，实现资源的有效分配。然后，通过考虑边缘服务器的资源情况，来确定每个区域的边缘服务器划分。最后，从联合优化的角度考虑，将边缘服务器的位置部署和资源分配结合在一起来实现大规模异构终端任务的协同处理，并提高整个系统的运行效率。本文的主要贡献可以总结如下。

(1)提出了一种基于端边协同的节点位置部署方法，它不仅考虑了终端设备的功能和地理位置的相似性，还考虑了边缘服务器的异构性，如处理能力、存储空间和网络带宽，从而决定每个区域最合适的边缘节点。

(2)开发了一个联合优化节点部署和资源利用率的方法，以指导任务的分配。这个方法结合了分层聚类算法和边缘节点的关键性能指标。它不仅关注单个节点的性能，而且还关注整个网络的资源分配和负载均衡。

(3)通过在公开数据集上进行仿真实验，验证了所提方法的有效性。实验结果显示，相比现有方法，本文方法在负载均衡、时延和能耗方面显著优于现有方法，能够将负载均衡水平提升 30%以上，同时将任务处理的时延和能耗降低 10%以上。

第 2 章讨论了相关工作；第 3 章详细给出了系统模型和问题描述；第 4 章介绍了提出的节点部署和资源分配联合优化方法；第 5 章介绍了仿真实验结果；最后总结本文并提出了未来展望。

2 相关工作

由于物联网设备的快速增长，边缘服务器部署问题已变得至关重要。最近，一些研究人员专注于通过优化边缘服务器部署来减少时延和能耗。文献[7]将其建模为多目标优化问题，然后通过混合整数优化来找到最佳的边缘服务器部署方案。文献[8]研究了 5G 超密集组网环境下边缘服务器的最优部署与分配策略，并基于排队论和矢量量化技术来优化边缘服务器的数量和位置，以及给定环境的移动用户分配。边缘服务器部署中的时延和能耗通常是矛盾的。为了平衡时延和能耗，它们被建模为一个总的目标函数，即任务开销。然后，提出了基于接入点评估的自适应聚类算法，以最大程度地减少任务开销[9]。在某些情况下，任务对处理时延需求较高，文献[10]将边缘服务器部署建模为容量约束的聚类问题，并通过启发式算法对其进行了优化。文献[11]提出了一个边缘服务器部署决策支持框架。该框架可以通过网络距离估计方法有效地找到所需的边缘服务器位置。为了提高边缘服务器提供商的利润，文献[12]研究了如何最大化边缘服务器的覆盖范围。他们假设边缘服务器的数量已固定，然后根据整数规划方法找到最佳解决方案。有时，某些边缘服务器会崩溃。为避免影响整体任务执行，有研究者提出了鲁棒服务器部署问题，以最大化预期的总体工作负载[13]。文献[14]还考虑了边缘服务器的优先位置和可靠性。他们开发了一种名为 PACK 的新算法，该算法将边缘服务器部署作为位置分配问题进行了制定。

尽管上述研究工作取得了一些有效的结果，但仍然有很大的改进空间。为了降低边缘服务器部署的时延和能耗，文献[15]提出了一种基于知识的节点部署方法。文献[16]认为

用户知道首次计算访问时延，然后他们根据权重提出了一种粒子群优化算法，用于平衡用户访问时延和服务器能源消耗。文献[17]提出了基于负载感知和移动性感知的边缘服务器部署方法，他们使用决策树和非主导分类遗传算法来获得平衡的部署策略。文献[18]提出了基于用户偏好的边缘服务器部署方法，他们建模了多目标优化问题并设计了两种渐进方法来加以解决。文献[19]提出了改进的 TOP-K 算法和改进的多目标非主体分类遗传算法，以提高用户体验质量。文献[20]提出了一种可以通过几何图像来计算边缘服务器的覆盖范围的方法，然后使用动态规划算法来最大化覆盖面积。但是，这些研究将边缘服务器视为相同的，并且不考虑边缘服务器的异构性。这些研究假定边缘服务器具有相同的计算能力和存储容量。边缘服务器在实际应用中通常是异构的。针对这个问题，文献[21]提出了用于部署异构边缘服务器的方法。他们结合了离线策略和在线策略，以完成异构边缘服务器的最佳部署。受到深入强化学习在芯片设计中的成功应用的启发[22]，一些研究人员试图使用强化学习来解决边缘服务器部署问题。文献[23]将边缘服务器部署问题建模为围棋游戏问题。基于深度强化学习，他们设计了一个自组织网络的移动节点部署模型。然后提出了一个新的强化学习框架，以解决边缘服务器部署和基站分配的联合问题。然而，尽管这种方法在理论上具有吸引力，但其应用在实际场景中并未达到预期的效果。首先，边缘服务器的部署问题本质上是高度复杂且多变的，难以简单地被建模为标准的顺序决策问题。这是因为边缘计算环境涉及到多种动态变化的因素，如网络流量、用户需求以及各种环境因素，这些都需要实时考虑和应对。因此，要将这种复杂性精确地映射到一个强化学习模型中，是一项极具挑战性的任务。其次，确定在边缘服务器部署后的收益函数同样具有挑战性。收益函数是强化学习中的关键组成部分，指导学习过程中的决策。然而，在边缘服务器部署的场景中，收益往往与多个因素相关，如服务质量、响应时间、资源利用率等。这些因素之间可能存在相互依赖和矛盾，使得定义一个全面且精确的收益函数变得复杂和困难。

3 系统模型和问题描述

3.1 系统模型

本文考虑了一个支持分散终端设备和异构边缘服务器的节点部署问题。节点部署系统模型可以视为由基站、边缘服务器和终端设备组成的无向图 $G=(C\cup S,X)$。其中，$B=\{b_1,b_2,b_3,\cdots,b_i,\cdots,b_m\}$ 代表基站的集合，$S=\{s_1,s_2,s_3,\cdots,s_j,\cdots,s_n\}$ 代表边缘服务器集合，并且 $S\in B$；$C=\{c_1,c_2,c_3,\cdots,c_q,\cdots,c_k\}$ 代表终端设备的集合；$X=\{x_{ij}=\{0,1\},\forall b_i\in B,\forall s_j\in S\}$ 表示边缘服务器和基站之间的连接集合。x_{ij} 是一个二进制变量，表示边缘服务器 s_j 是否放置在基站 b_i 的位置。边缘服务器是异构的，具有不同的计算能力和存储容量。每个边缘服务器的计算资源容量有限，数据传输时延与距离有关。因此，边缘服务器与覆盖的终端设备之间的距离非常重要。当前的研究仅根据终端设备的位置确定边缘服务器的位置。

本文专注于研究在终端设备的位置相对固定的情况下，如何优化边缘服务器的部署。我们的方法不仅考虑了终端设备的数量，还细致地分析了这些设备的类型和产生的数据量大小。这种综合考虑确保了部署策略能够针对不同类型的设备和数据需求提供最佳的服务。显然，边缘服务器的不同部署方案，对系统性能的关键指标，包括时延、能耗以及负载均衡，有着显著的影响。为了全面评估这些影响，我们在系统模型中引入了 3 个子模型：时延模型、能耗模型和负载均衡模型。时延模型旨在量化数据从终端设备到边缘服务器的传输时间以及服务器处理这些数据所需的时间。能耗模型则关注边缘服务器在处理任务过程中的能量消耗，这对于构建可持续的边缘计算系统至关重要。最后，负载均衡模型用于确保所有边缘服务器的工作负载均衡，避免某些服务器过载而其他服务器资源闲置。为了更好地设计和理解这些模型，表 1 详细列出了本文使用的一些重要符号。

表 1 文中涉及到的符号列表

Table 1 Notations involved in this paper

符号名称	符合含义	符号名称	符号含义
m	基站数量	v_j	节点处理速度
n	节点数量	L_j	节点位置
k	终端数量	L_q	终端位置
T	总时延	R	地球半径
T_{tran}	传输时延	p_j	节点功率
T_{comp}	计算时延	Z_j	终端区域
E	能耗	t_j	计算时间
H	负载均衡	η_j	节点负载率
d_{qj}	终端到节点距离	w_j	节点最大负载
x_{ij}	二进制变量	$\widetilde{\omega}_T$	时延权重
u_{qj}	任务大小	$\widetilde{\omega}_E$	能耗权重
φ	网络带宽	$\widetilde{\omega}_H$	负载均衡权重

文中系统模型细致地考虑了两种主要的时延类型，即传输时延和计算时延，这两者在边缘计算环境中起着至关重要的作用。传输时延是由数据从终端设备传输到边缘服务器所需的时间构成的。在我们的模型中，这种时延主要取决于两个关键因素：数据传输的物理距离和网络带宽的大小。尽管所有的边缘服务器和终端设备都处于统一的网络环境中，但是传输时延仍然主要由边缘服务器与终端设备之间的距离决定。这是因为距离的增加会导致数据传输所需时间的增加，从而增加整体的传输时延。另一方面，计算时延是指边缘服务器处理终端设备发送的任务所需的时间。这种时延主要取决于终端设备的任务大小以及边缘服务器的计算处理能力。任务大小代表了需要处理的数据量，而服务器的计算能力则决定了处理这些数据所需的时间。因此，计算时延可以被视为终端设备任务复杂性和边缘服务器处理能力之间的交互作用的结果。时延的计算模型如下所示：

$$T=T_{\text{tran}}+T_{\text{comp}} \tag{1}$$

$$T_{\text{tran}}=\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{q=1}^{k}\left(\frac{d_{qj}\cdot x_{ij}}{\varphi}\right) \tag{2}$$

$$T_{\text{comp}}=\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{q=1}^{k}\left(\frac{u_{qj}\cdot x_{ij}}{v_j}\right) \tag{3}$$

其中，$x_{ij}=\{0,1\}$ 表示边缘服务器 s_j 是否放置在基站 b_i 的位置。在考虑边缘服务器和终端设备的位置时，由于它们的位置坐标是以经度和纬度表示的，因此计算它们之间的距离需要使用适合球面坐标的距离公式。最常用的方法之一是半正矢(haversine)公式，它可以精确地计算地球表面上两点之间的最短距离。根据半正矢公式，终端设备 q 和边缘服务器 j 之间的距离可以表示为：

$$d_{qj}=2\arcsin\sqrt{\left[\sin^2\frac{a}{2}+\cos(L_q^{\text{lat}})\cdot\cos(L_j^{\text{lat}})\cdot\sin^2\frac{b}{2}\right]}R \tag{4}$$

$$a = L_q^{\text{lat}} - L_j^{\text{lat}} \tag{5}$$

$$b = L_q^{\text{lon}} - L_j^{\text{lon}} \tag{6}$$

在构建系统能耗模型时，有众多因素可能影响能耗，但为了简化分析，决定仅在边缘服务器处理任务时考虑能耗。在这个简化的模型中，能耗主要取决于两个关键因素：边缘服务器的功率以及任务处理的时间。能耗的计算公式可以表述为：

$$E = \sum_{j=1}^{n} p_j \cdot t_j \tag{7}$$

$$t_j = \sum_{q \in Z_j} \frac{u_{qj}}{v_j} \tag{8}$$

其中，功率是指边缘服务器在执行计算任务时的平均功率消耗，任务处理时间则是指服务器处理特定任务所需的时间。在负载均衡模型中，实现负载均衡的目的是确保每个边缘服务器的任务负载率保持在相似的水平。这一策略有助于避免单个服务器过载而引起的任务拥堵，同时也防止其他服务器资源的浪费。为了量化负载均衡，采用一种基于边缘服务器之间最大负载差的计算方法，可以表述为：

$$H = \max \eta_j - \min \eta_j \tag{9}$$

$$\eta_j = \frac{\sum_{q \in Z_j} u_{qj}}{w_j} \tag{10}$$

其中，η_j 为节点负载率，w_j 为节点最大负载，u_{qj} 为任务大小。具体来说，该方法计算所有边缘服务器的负载情况，然后确定负载最高的服务器与负载最低的服务器之间的差异值。

3.2 问题描述

不同的终端设备也可以生产许多不同类型和大小的数据流。这些数据流将上传到边缘服务器进行处理。为了最大程度地减少任务处理时延和能耗，有效的边缘服务器部署是必要的。对于此问题，给出以下假设：(1)边缘服务器位于与基站相同的位置，即需要从当前基站的位置选择边缘服务器的位置；(2)每个终端设备只能分配给由当前边缘服务器控制的区域；(3)放置的这些边缘服务器应覆盖所有终端设备。因此，边缘服务器部署问题可以描述为：如何将 n 个边缘服务器放置在 m 个基站中，以最大化 k 个终端设备的总任务处理效率。这个问题的核心目标是在终端设备与边缘服务器之间实现最优化的时延、能耗和负载均衡。为了达成这一目标，我们结合了前述的时延模型、能耗模型和负载均衡模型，以构建一个综合性的评估框架。在这个框架中，我们提出了一个总加权目标函数，它是对时延、能耗和负载均衡这 3 个关键指标的综合评估。其中，决策变量包括两个主要方面：边缘节点的选择和资源分配情况。边缘节点的选择指的是从所有可用节点中挑选出最适合承载当前工作负载的节点，以确保高效的数据处理与传输。资源分配情况则涉及对计算资源、带宽和存储容量等的合理分配，以满足不同应用和用户的特定需求。这个目标函数可以被表达为：

$$\begin{aligned}
&\min(\tilde{\omega}_t \cdot T + \tilde{\omega}_e \cdot E + \tilde{\omega}_h \cdot H)\\
&\text{s.t. } C1: \sum_{i=1}^{m} x_{ij} = 1 (\forall 1 \leqslant j \leqslant n)\\
&\quad C2: Z_i \cap Z_j = \varnothing (\forall i, j, 1 \leqslant i, j \leqslant n)\\
&\quad C3: \bigcup_{j \in S} Z_j = C\\
&\quad C4: \sum_{q=1}^{k} U_{qj} \leqslant U
\end{aligned} \tag{11}$$

通过这种方式，能够在单一的优化框架内同时考虑时延、能耗和负载均衡，从而提供一个全面且灵活的解决方案。这个总加权目标函数不仅为我们提供了一个量化的优化目标，还允许我们根据不同的应用需求和约束条件调整优化策略。其中，加权因子用于调整各子目标在总目标函数中的相对重要性。时延、能耗和负载均衡的权重和为 1，且初始参数设置为 1/3，在优化过程中可以根据结果动态调整。这种设置是基于对这 3 个因素同等重要性的假设，从而保证在运行初期各指标能够平衡地影响整体性能。在优化过程中，将根据监测到的实际运行数据，动态调整这些权重。这些系数的选择反映了不同应用场景对时延、能耗和负载均衡的不同需求。为了确保部署策略的有效性和实用性，我们设定了一些约束条件，定义了边缘服务器的放置规则、覆盖范围和服务保障等。其中，C1 表示每个边缘服务器只能部署在一个独立的基站上，且不允许在同一基站上重复部署多个边缘服务器。这样的规定旨在确保基站资源的有效利用，避免不必要的资源浪费和部署冲突。C2 是为了确保没有两个边缘服务器会覆盖相同的终端设备。也就是说，每个终端设备在任何给定时间内只能连接到一个边缘服务器。这个条件的设置是为了避免资源的冗余使用和潜在的服务冲突，确保每个终端设备都能获得最优化的服务和资源分配。C3 表示所有的终端设备必须被至少一个边缘服务器覆盖。这意味着部署策略需要确保网络中的每个终端设备都能接入至少一个边缘服务器，从而保证无一设备处于服务盲区。约束 C4 表示所有终端分配的资源小于或等于可用资源。通过在这些约束条件下寻找解决方案，找到一种既高效又实用的边缘服务器部署策略，从而最大化整个网络的任务处理效率。

4 联合优化方法

由于式(11)中的约束条件包括整数向量和连续向量，所以目标函数是一个组合优化问题。整数向量表示的变量通常代表资源的离散选择，而连续向量表示的变量则可以采取连续值，导致解空间既包含离散元素，又有连续范围。将时延、能耗和负载均衡等多项指标综合考虑并在此组合优化问题中求解，使得问题复杂度显著增加[24]。为了有效解决上面定义的问题，首先提出了基于端边协同的节点部署方法。针对不均匀的终端设备地理分布问题，采用了一个先进的分层聚类算法，通过这种算法，能够将地理分布不均的终端设备有效地划分为不同的区域。在这个过程中，那些地理位置相近或任务产生速度高度相似的终端设备被分配到同一个区域。我们的方法采用了一种自下而上的聚合策略，这种策略在聚类分析中非常有效。具体来说，我们的算法最初将每个终端设备视为一个独立的类。接着，算法计算这些终端设备之间的距离以及它们任务产生速度的相似性，这些计算基于精确的数学模型来量化设备之间的相似度。随后，算法将地理位置最接近或任务产生速度最相似的终端设备组合成一个类。通过将相似或相邻的终端设备分组，能够更准确地识别出哪些区域需要更多的资源，哪些区域则可能不需要那么多资源。这样，就可以根据每个区域的实际需求来部署边缘服务器，从而在整个网络中实现更高效和平衡的资源分配。

在成功地通过分层聚类算法划分出不同的终端设备区域

之后，下一步便是确定每个区域内最适合的边缘节点。这一决策过程涉及到对边缘服务器的关键性能指标的细致考量，包括处理能力、存储空间和网络带宽。这些指标不仅直接影响边缘服务器的工作效率，也决定了其能够提供的服务水平。考虑到边缘服务器可能存在的异构性以及多种资源的协调难题[25]，我们开发了一个综合的异构资源度量模型：$U=U^{comp}\cdot U^{cach}\cdot U^{comm}$。其中，$U^{comp}$，$U^{cach}$和$U^{comm}$分别表示边缘服务器的计算水平、存储水平和通信水平。该模型旨在全面评估边缘服务器的不同方面。这些不同层面的评估帮助我们精确地了解每台边缘服务器的能力和潜力，使得资源分配过程更为科学和合理。基于这个模型，能够根据每个区域内终端设备的具体资源需求来分配相应的边缘服务器。这意味着那些具有更高资源容量的边缘服务器会被优先分配给资源需求更为密集的区域，如图 2 所示。

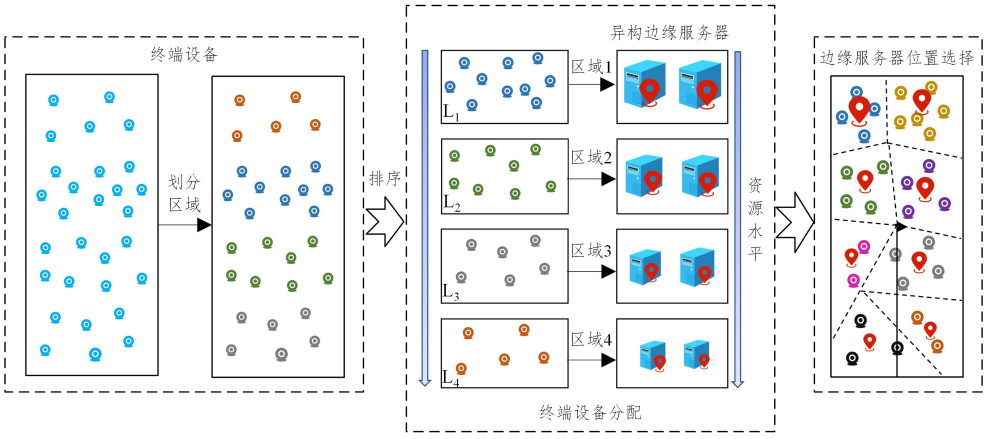

图 2　终端设备区域划分和服务器分配

Fig. 2　End device zoning and server allocation

在完成边缘服务器的初步分配之后，进一步采用了一种节点部署和资源分配联合优化算法，以精确确定在每个区域内边缘服务器的最佳部署位置和资源分配方案，如算法 1 所示。这种联合优化方法的核心目的是在大规模异构终端任务处理中实现更高的协同效率，从而提升整个系统的运行效率。该联合优化算法考虑了多个关键因素，包括但不限于每个区域的任务负载、边缘服务器的处理能力、存储容量以及网络带宽。通过综合这些因素，算法能够为每个区域内的边缘服务器确定最优的物理位置和资源配置。这一步骤至关重要，因为它直接影响到任务处理的速度、系统的可靠性以及用户体验的质量。

算法 1　节点部署和资源分配联合优化算法

输入：$C=\{c_1,c_2,c_3,\cdots,c_q,\cdots,c_k\}$

　　$S=\{s_1,s_2,s_3,\cdots,s_j,\cdots,s_n\}$

输出：节点部署方案和资源分配方案

1. for 每个边缘服务器 in 所有边缘服务器：
2.　　计算基于当前的最佳部署位置
3.　　更新边缘服务器的部署位置
4. end for
5. for 每个边缘服务器 in 所有边缘服务器：
6.　　根据终端设备需求和服务器能力动态分配资源
7.　　更新服务器的资源分配方案
8. end for
9. 计算当前解决方案的性能指标
10. if 性能满足预定目标
11.　　输出结果
12. else 性能为满足预定目标
13.　　更新参数
14. endif
15. return result

具体地，首先分析网络拓扑和用户分布，以确定边缘服务器的最佳部署位置。然后，它根据每个服务器的计算能力、存储容量和网络带宽，以及各终端设备的需求，来动态分配资源。这个过程通过多次迭代，使用启发式方法来不断调整节点位置和资源分配，直到达到预定的性能目标。算法在考虑约束条件的同时，确保所有解决方案均满足这些条件。此外，我们的算法特别强调了大规模异构环境下的任务协同处理。在现实世界中，边缘计算系统经常需要处理来自各种不同类型和功能的终端设备的任务，这些任务在性质、大小和处理要求上可能大相径庭。我们的优化算法通过智能地分配和调整资源，确保了即使在这种复杂多变的环境中，每个任务也能被有效地处理，从而最大化整个系统的运行效率。

5　仿真实验

5.1　实验设置

在本节中，我们验证了所提方案的有效性。使用 EUA 数据集[26]来模拟实验。EUA 数据集是由澳大利亚通信和媒体管理局颁发的无线电通信许可证的数据集。EUA 数据集包含澳大利亚的基站和用户的位置。在模拟研究中，我们精心选择了墨尔本中央商务区作为实验场景，以其作为一个典型的环境进行案例分析。在这个区域中，收集了 816 个用户位置数据，这些数据代表了不同位置的终端设备。此外，还使用了 125 个基站位置数据，以模拟现实世界中的网络基础设施布局。为了更贴近实际情况，每个终端设备的任务产生速度被设定在 10 Mb/s 到 50 Mb/s 之间的随机值。总带宽被统一设置为 500 Mbps，这代表了整个网络环境的数据传输上限。对于边缘服务器，我们考虑了计算能力和存储能力的异构性。每个服务器的计算和存储能力在 100 Mb/s 至 300 Mb/s 之间随机生成，以模拟不同服务器的性能差异。同时，设定每个服务器的功率为 100 W，这有助于在模拟中评估能耗和环境影响。最后，为了全面评估所提方案的有效性，将其与其他几种基准

方法进行了对比。这些比较主要集中在负载平衡、时延和能耗等关键性能指标上。用于比较的基准方法如下。

(1)随机方案:首先从基站的位置集随机选择以放置边缘服务器,然后将所有终端设备分给最近的边缘服务器。这个过程不依赖于任何先验知识或特定策略,确保了服务器位置的选择是完全随机的。

(2)TOP-K 方案:首先,将所有终端设备分给最近的基站。然后,计算每个基站的工作量,在评估了所有基站的工作量之后,选择工作量最大的前 K 个基站作为边缘服务器的部署点。最后,将所有终端设备分给最近的边缘服务器。

(3)K-means 方案:首先,初始化 K 个聚类中心,并计算每个终端设备到聚类中心的距离,再将其分给最近的聚类中心。然后,重新计算每个类的聚类中心,重复直到聚类中心不变。计算从每个基站到 K 个聚类中心的距离,取最近的作为边缘服务器的位置。最后,将所有终端设备分到最近的边缘服务器。

(4)MCT 方案:综合考虑了边缘节点的部署与资源的联合优化。这是一种基于排队论和矢量量化技术的方案。排队论主要用于建模移动用户任务的处理过程,帮助合理分配边缘服务器的资源。矢量量化技术则用于高效确定边缘服务器的最佳部署位置,同时为给定环境中的移动用户提供分配策略。

5.2 结果分析

首先对任务处理过程中各个节点的负载情况进行了分析。为了具体说明,设定边缘节点的总数为 50,并对每个节点的负载情况进行了详细的统计,结果如图 3 所示。可以直观地观察到,与其他方案相比,我们所提出的方案在负载均衡方面展现出了显著的优势。在我们的方案中,各边缘节点之间的负载方差相对较小,这表明了一个更加均衡的资源分配。这种稳定性的增加不仅减小了过载节点的可能性,还意味着资源在整个网络中得到了更加合理和高效的利用。这种平衡的资源分配是提升任务处理效率的关键因素。

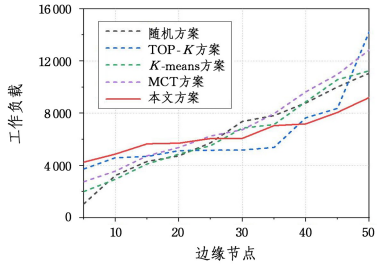

图 3 每个节点的工作负载情况

Fig. 3 Workload of each edge node

进一步地,图 4 量化地展示了各方案负载平衡性能指标的对比结果。这些结果显示,相比于其他方法,我们的方案在负载平衡方面具有显著优势,平均能够将负载平衡提升超过 30%。这一显著的性能提升主要源于我们方案在放置边缘服务器时所采用的综合策略。我们的方法不仅考虑了端边协同的区域划分策略,而且还综合了资源分配联合优化策略。这种综合考虑确保了在整个网络中资源被最有效地利用,同时每个节点都能够根据实际需求得到适当的资源支持。与其他方法相比,我们的方案更加精细地考虑了节点之间的协作和资源分配,从而实现了更优的负载均衡性能。

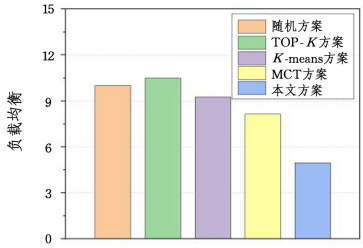

图 4 负载均衡的对比结果

Fig. 4 Comparative results of load balancing

最后,为了全面评估所提方案的性能,进一步测试了不同数量的边缘节点(从 10 个增加到 50 个)对任务处理时延和能耗的影响。这些实验结果被详细地展示在图 5 和图 6 中。结果显示,随着边缘节点数量的增加,我们的方案在时延方面表现出了明显的优势。即使在节点数量较少时,我们的方法也能维持较低的时延。在能耗方面,我们的方案同样展示了一定的优势。随着节点数量的增加,所提方案的能耗增长率较低,显示出更好的能效性能。这些实验结果表明,无论是在时延还是能耗方面,我们的方案都略优于其他方法。总的来说,对比其他方法,我们的方案可以将平均时延和平均能耗降低 10%以上。这主要归因于我们方案在边缘节点部署和资源分配上的综合考虑和优化。实验还显示了我们的方案具有良好的可扩展性,即使在边缘节点数量增加的情况下也能保持较好的性能表现。整体上,这些实验不仅验证了所提出方案的有效性,还展示了其在不同规模和条件下的鲁棒性和优越性。这些结果为未来的边缘节点部署、网络设计和资源优化提供了有力的支持和参考。

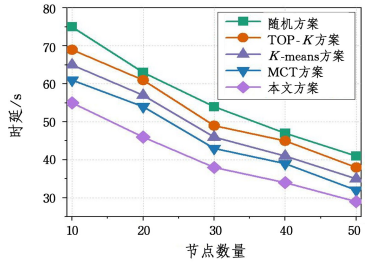

图 5 时延的对比结果

Fig. 5 Comparative results of delay

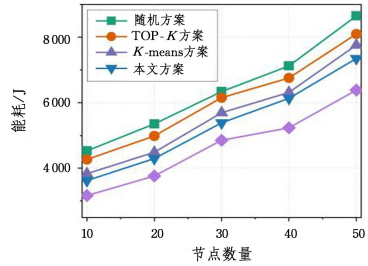

图 6 能耗的对比结果

Fig. 6 Comparative results of energy consumption

结束语 针对物联网快速发展背景下的边缘节点位置部署问题,本文提出了一种基于端边协同的节点部署和资源分配联合优化方法。通过对物联网环境中终端设备的广泛分布和边缘服务器的异构特性进行深入分析,我们的方法成功地

解决了边缘计算系统面临的关键挑战。最终，通过联合优化节点部署和资源利用率，我们的方法指导了有效的任务分配，从而提升了整个系统的运行效率。在未来的工作中，我们计划将方法应用于更多样化的场景，并研究如何实时调整边缘服务器的部署和资源分配策略，以适应网络条件和终端需求的动态变化。

参 考 文 献

[1] The Mobile Economy[EB/OL]. [2023-07-16]. https://www.gsma.com/mobileeconomy/.

[2] YANG Z M, HU D, GUO Q, et al. Visual E^2C: AI-driven visual end-edge-cloud architecture for 6G in low-carbon smart cities [J]. IEEE Wireless Communications, 2023, 30(3): 204-210.

[3] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile edge computing: A survey [J]. IEEE Internet of Things Journal, 2017, 5(1): 450-465.

[4] MACH P, BECVAR Z. Mobile edge computing: A survey on architecture and computation offloading [J]. IEEE Communications Surveys& Tutorials, 2017, 19(3): 1628-1656.

[5] CRUZ P, ACHIR N, VIANA A C. On the edge of the deployment: A survey on multi-access edge computing [J]. ACM Computing Surveys, 2022, 55(5): 1-34.

[6] YANG Z M, JI W, GUO Q, et al. JAVP: Joint-aware video processing with edge-cloud collaboration for DNN inference[C]// Proceedings of the 31st ACM International Conference on Multimedia. ACM, 2023: 9152-9160.

[7] WANG S, ZHAO Y, XU J, et al. Edge server placement in mobile edge computing [J]. Journal of Parallel and Distributed Computing, 2019, 127(5): 160-168.

[8] LI B, HOU P, WU H, et al. Optimal edge server deployment and allocation strategy in 5G ultra-dense networking environments [J]. Pervasive and Mobile Computing, 2021, 72: 101312.

[9] LI B, HOU P, WU H, et al. Placement of edge server based on taskoverhead in mobile edge computing environment [J]. Transactions on Emerging Telecommunications Technologies, 2021, 32(9): e4196.

[10] LEE S, SHIN M K. Low cost mec server placement and association in 5g networks[C]// 2019 International Conference on Informationand Communication Technology Convergence. IEEE, 2019: 879-882.

[11] YIN H, ZHANG X, LIU H H, et al. Edge provisioning with flexible server placement [J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 28(4): 1031-1045.

[12] CUI G, HE Q, CHEN F, et al. Trading off between user coverage and network robustness for edge server placement [J]. IEEE Transactions on Cloud Computing, 2020, 10(2): 2178-2189.

[13] LU D, QU Y, WU F, et al. Robust server placement for edge computing[C]// 2020 IEEE International Parallel and Distributed Processing Symposium. IEEE, 2020: 285-294.

[14] AHDERANTA T L, LTPPANEN T, RUHA L, et al. Edge computing server placement with capacitated location allocation [J]. Journal of Parallel and Distributed Computing, 2021, 153(2): 130-149.

[15] LI Y, ZHOU A, MA X, et al. Profit-aware edge server placement [J]. IEEE Internet of Things Journal, 2021, 9(1): 55-67.

[16] GUO Y, WANG S, ZHOU A, et al. User allocation-aware edge cloud placement in mobile edge computing[J]. Software: Practice and Experience, 2020, 50(5): 489-502.

[17] XU X, XUE Y, QI L, et al. Load-aware edge server placement for mobile edge computing in 5g networks[C]// International Conference on Service-Oriented Computing. Springer, 2019: 494-507.

[18] CHEN Y, LIN Y, ZHENG Z, et al. Preference-aware edge server placement in the internet of things [J]. IEEE Internet of Things Journal, 2021, 9(2): 1289-1299.

[19] QIN Z, XU F, XIE Y, et al. An improved top-k algorithm for edge servers deployment in smart city [J]. Transactions on Emerging Telecommunications Technologies, 2021, 32(8): e4249.

[20] WANG F, HUANG X, NIAN H, et al. Cost-effective edge server placement in edge computing[C]// Proceedings of the 5th International Conference on Systems, Control and Communications. ACM, 2019: 6-10.

[21] CAO K, LI L, CUI Y, et al. Exploring placement ofheterogeneous edge servers for response time minimization in mobile edge-cloud computing [J]. IEEE Transactions on Industrial Informatics, 2020, 17(1): 494-503.

[22] MIRHOSEINI A, GOLDIE A, YAZGAN M, et al. A graph placement methodology for fast chip design [J]. Nature, 2021, 594(7862): 207-212.

[23] WANG H, YANG R, YIN C, et al. Research on the difficulty of mobile node deployment's self-play in wireless ad hoc networks based on deep reinforcement learning [J]. Wireless Communications and Mobile Computing, 2021, 2021(1): 1-13.

[24] YANG Z M, JI W, WANG Z. Adaptive joint configuration optimization for collaborative inference in edge-cloud systems[J]. Science China Information Sciences, 2024, 67(4): 1-2.

[25] YANG Z M, LIANG B, JI W. An intelligent end-edge-cloud architecture for visual iot assisted healthcare systems [J]. IEEE Internet of Things Journal, 2021, 8(23): 16779-16786.

[26] LAI P, HE Q, ABDELEAZEK M, et al. Optimal edge user allocation in edge computing with variable sized vector bin packing [C]// International Conference on Service-Oriented Computing. Springer, 2018: 230-245.

YANG Zheming, born in 1996, doctoral student. His main research interests include multimedia systems, edge intelligence, network optimization, and visual IoT.

JI Wen, born in 1976, Ph.D, professor, Ph.D supervisor. Her main research interests include vision processing units, end-edge-cloud computing systems, vision coding and transmission, intelligent multimedia computing, low-carbon computing, and optimization theory methods.