计算机科学
COMPUTER SCIENCE

基于谱聚类的边缘服务器放置算法

郭迎亚，王丽娟，耿海军

引用本文

郭迎亚，王丽娟，耿海军. 基于谱聚类的边缘服务器放置算法[J]. 计算机科学，2023，50(10)：248-257.

GUO Yingya, WANG Lijuan, GENG Haijun. Edge Server Placement Algorithm Based on Spectral Clustering [J]. Computer Science, 2023, 50(10): 248-257.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220900211

基于谱聚类的边缘服务器放置算法

郭迎亚[1,2]　王丽娟[1,2]　耿海军[3]
1 福州大学计算机与大数据学院　福州 350108
2 福州大学网络计算和智能信息处理重点实验室　福州 350108
3 山西大学自动化与软件学院　太原 030006
(guoyy@fzu.edu.cn.)

摘　要　随着物联网(IoT)和5G技术的快速发展,移动边缘计算以其低访问延迟、低带宽成本和低能源消耗的优点引起了工业界和学术界的广泛关注。在移动边缘计算中,边缘服务器为移动端用户的请求提供服务,其放置位置对边缘计算性能和用户体验具有重要影响。目前边缘服务器的放置算法只考虑基站的地理位置,而缺乏对基站连接的用户数目因素的考虑。因此在实际用户分布不均的情况下,现有算法得到的服务器放置位置导致用户平均访问延迟较大。为了更好地解决上述问题,提出了基于谱聚类的延迟最小化边缘服务器放置算法LAMP。该算法在考虑边缘服务器放置位置时,不仅考虑了基站的地理位置,而且考虑了不同基站连接的用户数目这一重要参数,能够有效地降低用户的平均访问时延,同时实现边缘服务器的工作负载均衡。在仿真实验中,使用了上海电信的真实基站数据集来测试LAMP算法的性能。大量的实验结果表明,在用户访问延迟方面,LAMP算法的性能比传统的K-means算法提高了37.9%。在负载均衡方面,LAMP算法的性能与K-means算法相比最大可提高82.85%。LAMP算法在降低访问延迟和平衡边缘服务器工作负载方面均表现出了优越的性能。

关键词:移动边缘计算;边缘服务器放置;用户分布;谱聚类算法;访问时延;工作负载
中图法分类号　TP393

Edge Server Placement Algorithm Based on Spectral Clustering

GUO Yingya[1,2],WANG Lijuan[1,2] and GENG Haijun[3]
1 College of Computer and Data Science,Fuzhou University,Fuzhou 350108,China
2 Fujian Key Laboratory of Network Computing and Intelligent Information Processing,Fuzhou University,Fuzhou 350108,China
3 School of Automation and Software Engineering,Shanxi University,Taiyuan 030006,China

Abstract　With the rapid development of the Internet of Things(IoT) and 5G networks,mobile edge computing has attracted widespread attention from industry and academia for its low access latency,low bandwidth costs,and low energy consumption. In mobile edge computing,edge servers provide services for mobile user requests,and the placement of edge servers has an important impact on edge computing performance and user experience. At present,the placement algorithm of edge servers only considers the geographical location of server placement,and lacks the consideration of the number of users connected to the base station. Therefore,in the case of uneven distribution of actual users,the average user access delay caused by the server placement position obtained by the existing algorithm is large. In order to better solve the above problems,this paper proposes a latency minimization edge server placement algorithm based on spectral clustering. When solving the problem of edge server placement,the algorithm not only considers the geographical location of the base station,but also takes into account the important parameter of the number of users connected to different base stations,which can effectively reduce the average access latency of users and make the workload of each edge server more balanced at the same time. In the simulation experiment,this paper uses the real base station dataset of Shanghai Telecom to test the performance of the proposed server placement algorithm. Simulation experiment results show that the user-distributed access delay minimization edge server placement algorithm has significant advantages in solving the edge server placement problem. In terms of access latency,the performance of LAMP algorithm is increased by 37.9% compared with K-means algorithm. Compared with the K-means algorithm,the performance of the LAMP algorithm can be improved by up to 82.85% in terms of load balancing. The LAMP algorithm exhibits superior performance in reducing access latency and balancing

到稿日期:2022-09-22　返修日期:2022-11-28
基金项目:国家自然科学基金(62002064,62072109);福建省自然科学基金(2020J05110)
This work was supported by the National Natural Science Foundation of China(62002064,62072109) and Natural Science Foundation of Fujian Province,China(2020J05110).
通信作者:耿海军(ghj123025449@163.com)

edge server workloads.

Keywords Mobile edge computing, Edge server placement, User distribution, Spectral clustering, Access delay, Workload balancing

1 引言

随着万物互联和元宇宙时代的到来[1]，网络边缘设备的数量及其产生的数据量都呈爆炸式增长。根据思科公司的数据预测[2]，智能终端设备规模及其产生的数据规模均呈现出指数级增长趋势。到2021年，全球范围内将有超过500亿的终端设备，这些设备每年产生的数据量将达到847 ZB。随着移动边缘设备如VR/AR、智能车辆[3]和智能家居[4]等越来越普及和趋于智能化，产生的延迟敏感和计算密集型任务[5]不断增加，对服务时延的要求不断提高。传统的云计算模型[6]是将计算密集型任务卸载到远程云数据中心进行处理和存储。然而面对低传输时延的需求以及边缘应用程序的隐私保护需求，传统的云计算模式已经无法满足。为有效解决上述问题，移动边缘计算(Mobile Edge Computing，MEC)应运而生。MEC是一种为边缘设备提供所需服务和云计算资源的网络架构[7]。MEC作为云计算的延伸和补充，将计算资源下沉到网络边缘[8]，计算任务可以在靠近数据源的边缘服务器上访问资源并得到及时的响应和处理[9]，大大缩短了用户设备与云数据中心之间长距离数据传输所造成的长时延，同时减轻了数据传输的带宽压力，增强了对边缘设备应用程序中隐私数据的保护。因而近年来，移动边缘计算受到工业界和学术界的广泛关注。

在移动边缘计算中，边缘服务器为移动端用户的请求提供服务，其放置位置对移动边缘计算的性能和用户体验具有重要影响。在移动边缘计算网络系统中[5]，边缘服务器被部署在基站上，形成边缘节点，基站被分配给其邻近的边缘节点，并将服务请求转发到边缘节点。边缘节点对边缘设备以及应用程序的访问请求提供计算资源和存储资源，从而为用户提供高质量的网络服务。在移动边缘计算网络中，边缘设备访问服务器所产生的访问延迟主要受距离的影响。考虑到运营成本以及网络系统的实际部署情况，边缘服务器将被放置在已投入使用的基站上，并且需要确保放置的边缘服务器尽可能地靠近边缘设备，使得区域内所有基站上的用户可以快速地访问资源。其次，边缘设备访问服务器所产生的访问延迟还受到服务器工作负载的影响，服务器的工作负载与区域内各个基站上连接的用户数目密切相关。当多用户同时访问同一台服务器的资源时，服务的响应时间会有所延长，因此边缘服务器的放置必须考虑用户的分布情况，以尽可能地实现边缘服务器的工作负载均衡。在放置边缘服务器时考虑用户数目后，拥有大量用户数目的基站之间的访问距离会变大，使得它们之间聚为一类的概率变小，从而可以降低边缘服务器的工作负载，提高网络服务质量。综上所述，在移动边缘计算中，合理地部署边缘服务器以充分发挥移动边缘计算的优势是一项相当具有挑战性的工作，需要对边缘服务器的放置方案进行优化，以降低服务器的访问延迟和均衡服务器的工作负载。

针对边缘服务器的放置问题，现有的研究工作主要围绕基站的地理位置[10]、基站的集群规模[11-13]、边缘服务器的部署成本[14-15]、边缘服务器资源利用效率[16-19]等方面展开，并提出了一系列算法来实现用户时延、服务器负载均衡以及经济节能方面的优化。在MEC网络系统中，基站的位置具有分散程度大和分布密度不均的特性，同时，终端用户的分布具有高度的离散化，导致样本空间不规则，这对边缘服务器的放置方案提出了更高的要求。然而，现有的研究工作均未考虑到终端用户的分布情况，从而使用户体验受到影响。为解决这一问题，本文提出了基于用户分布的延迟最小化边缘服务器放置算法(LAtency Minimization edge server Placement algorithm based on user distribution，LAMP)。该算法采用谱聚类算法对所有的基站进行聚类，并在聚类中确定边缘服务器的部署位置，以解决边缘服务器放置问题。对于分布离散且密度不均的基站数据集样本空间，谱聚类可以充分发挥其优势，对样本进行有效的聚类，实现更合理化的基站分配，以满足实际部署中的要求。具体地，在进行谱聚类算法之前，首先需要对基站数据集进行预处理，将数据集中的基站分为两部分：中心集和非中心集。最后，本文采用上海电信的基站数据集[5,20-22]来进行算法性能的测试和验证。实验结果表明，本文提出的LAMP算法在降低访问延迟和平衡边缘服务器工作负载方面均表现出了优越的性能。

综上所述，本文的贡献包括以下3个方面：

(1)首次在边缘服务器放置问题中考虑了用户分布因素。基站除了地理位置所固有的属性之外，用户的分布情况较为离散且密度不均，导致基站上所连接的用户数目也不同。在边缘服务器放置时，基站连接的用户数目也应该加以考虑，以满足尽可能多的终端用户的网络需求，使边缘服务器尽可能地靠近用户侧。

(2)提出了基于谱聚类的用户延迟最小化边缘服务器放置算法。在MEC网络中，基站数据较多并且样本空间呈现不规则性，离散化程度较高。使用谱聚类算法进行聚类，可以有效地解决数据分布不均匀的问题，实现延迟最小化的边缘服务器放置。

(3)在真实的上海电信的基站数据集上进行了算法的评估。大量的实验结果表明，在访问延迟方面，LAMP算法的性能比K-means算法提高了32%。在负载均衡方面，LAMP算法的性能比K-means算法提高了83%。

本文第2章介绍移动边缘网络中边缘服务器放置问题相关的研究工作，并对提出的方法以及存在的问题进行介绍和一定的分析；第3章介绍研究场景并描述研究问题；第4章介绍基于用户分布延迟最小化的边缘服务器放置算法；第5章介绍实验所用的数据集以及实验参数设置和实验结果分析；最后总结全文并展望未来。

2 相关工作

随着边缘计算的研究越来越深入，边缘计算网络中相关

设备的放置问题也引起了越来越多的关注。为了实现在表达上的便捷性和一致性，路边单元(Roadside Units，RSUs)、微云(Cloudlets)以及移动边缘网络边缘服务器(MEC edge servers)等具有丰富的计算和存储资源的边缘设备都被统称为边缘服务器。对边缘服务器放置问题的相关工作调研，将从 RSUs、Cloudlets、边缘服务器 3 个方面展开。在边缘服务器放置算法方面，本文首次将谱聚类算法与边缘服务器放置问题相结合，本章将介绍谱聚类的相关知识。

2.1 RSUs 放置问题的研究

从目前的研究工作来看，关于 RSUs 的研究活动主要从通信范围和部署成本的角度[23-26]来部署网络。Liang 等[23]以网络覆盖和部署成本为优化指标，利用线性规划研究了二维车辆网络中的部署。Wang 等[27]将社交网络中中心性的概念引入 RSUs 的部署中，并提出了一种基于中心性的 RSUs 部署方法。首先，将 RSUs 部署问题抽象为线性规划问题，然后将 RSUs 问题转化为 0-1 背包问题。与随机部署方法相比，RSUs 的部署效率有了显著的提高。Zhang 等[28]提出了一种异步粒子群优化算法，从精准定位和部署成本的角度来部署 RSU，希望使用最少的 RSUs 来获得最佳的车辆精准定位。除此以外，还有少量研究是针对车辆应用[29]计算资源需求而进行的。Premsankar 等[30]从网络覆盖和计算资源两方面考虑了 RSUs 的放置问题，并使用混合整数规划的思想实现了 RSUs 的部署。

2.2 Cloudlets 放置问题的研究

现有的 MEC 网络中对于放置问题的研究工作都集中在 Cloudlets 放置[11,31-33]。有研究表明[21]，Cloudlets 和 MEC edge servers 的放置问题有很多相似之处。由于放置问题都是在一定的地理范围内进行研究的，因此 Zhang 等[9]应用覆盖算法(Covering Algorithm，CA)根据移动设备的地理位置自适应地聚类移动设备。Xiang 等[10]研究了自适应的微云放置问题，提出了基于移动应用地理位置信息大数据的自适应移动微云放置问题。在基站集群规模还不是很大时，Fajardo 等[12]提出了一种基于靠近远程单元放置处理和存储能力的解决方案，该方案特别适合部署小基站集群。随着基站集群规模的不断扩大，Ma 等[13]研究了大规模无线城域网中的 Cloudlets 放置问题，采用 K-medoids 算法来实现 Cloudlets 最佳部署。Jia 等[34]也研究了在大规模城域网中放置有限数量的微云和给微云分配用户，以最小化工作任务卸载的平均等待时间问题。在微云放置过程中，非常容易出现移动计算任务分布不均的情况。Chen 等[35]采用分布式博弈论方法，将移动设备用户之间的分布式计算卸载决策问题表述为多用户计算卸载博弈。Dashti 等[33]提出了修改粒子群优化算法，目的是在过载主机之间重新分配迁移的虚拟机，并动态集成负载不足的主机。Manasvi 等[36]利用社交网络信息放置边缘服务器，与 Mark 等[37]提出的一种带有需求预测器的进化最优虚拟机放置(Evolutionary Optimal Virtual Machine Placement，EOVMP)算法有异曲同工之妙，它们都是根据任务分布来优化放置算法。在决策边缘服务器的放置方案时，还应该考虑到服务提供商的成本问题。Fan 等[14]发现一些研究没有考虑部署成本这一要素，因此在考虑部署成本和访问延迟下建立了线性规划模型。Yang 等[15]研究微云放置时考虑了基站的位置，指出在每个基站上放置云服务中心的能源成本会非常昂贵。Zeng 等[16]专注于在不影响用户上网体验的情况下，尽量减少边缘服务器的总数，以降低放置成本，他们使用基于模拟退火的方法和基于贪婪的算法来解决这个问题。Santoyo-González 等[17]指出，边缘服务器放置方案对边缘资源的使用效率有很大的影响，他们提出了一系列参数来评估新出现的 5G 场景中边缘服务器的放置方案。Xu 等[38]考虑了具有不同计算资源的 Cloudlets 放置，并提出了一种快速有效的启发式算法以最小化访问延迟。Meng 等[39]考虑到用户的动态请求对 Cloudlets 放置的影响，采用近似和迭代算法来获得最佳放置方案。关于 Cloudlets 放置问题的研究，均是从基站地理位置、基站集群规模、计算任务分布、放置成本、资源使用效率等方面展开的。

2.3 边缘服务器放置问题的研究

研究者们通常将边缘服务器放置问题形式化为整数线性规划问题[11,20-21,31]。为了更好地求解整数规划问题，Guo 等[21]提出了一种基于 K-means 和混合整数二次规划的近似算法。Wang 等[20]提出了一种 MIP(Mixed Integer Programming)放置算法，其主要目标是实现边缘服务器的负载平衡，实验结果表明，与经典算法 K-means[40-41]和 Random[20]相比，MIP 算法表现出了一定的优势。Cui 等[42]考虑了分布式 MEC 环境的网络鲁棒性，提出了一种基于整数规划的方法来放置边缘服务器，从而提升用户的体验，达到访问网络时延小、服务器负载均衡的目标。Fan 等[43]用拉格朗日启发式算法最小化云服务中心的计算代价和端到端的延迟代价。在目前的研究中，很多工作使用启发式算法[5,38,44-45]来解决服务器放置问题，采取先确定服务器放置的位置再分配其所属基站的策略思想。为实现对移动用户服务请求的实时响应，Guo 等[21]提出了一种基于动态编程的离线微服务协调算法，使用马尔可夫决策过程定义了微服务协调问题，并加入了强化学习以实现最优策略的选择。他们在网络服务的过程中还注意到了移动终端设备的隐私信息保护问题。Zhang 等[46]基于联邦学习的思想，在动态的网络系统中为用户分配边缘服务器，优化边缘区域的服务缓存，有效保护了用户的隐私信息。上述工作大多考虑了访问延迟，但忽略了现实中基站位置分布的离散性和不均匀性，这将大大降低算法在现实场景中的性能。

2.4 谱聚类算法

谱聚类[47-48]是一种从图论中演化出来的聚类方法，其主要思想是把样本空间中所有的数据看作空间中的点，这些点之间可以用线连接起来，线称为两点间的连接边，两个点间的距离越远，其连接边的权重值越小，而距离较近的两个点间的连接边权重值较大。对所有数据点组成的图进行切图，实现切图后不同的子图间的连接边权重之和尽可能地小，而子图内的连接边权重之和尽可能地大，从而达到聚类的目的。该算法建立在图论基础上，与 K-means 算法相比，具有能在任意形状的样本空间上聚类且收敛于全局最优解的优势，尤其是在分布离散且密度不均的基站数据集样本空间中，谱聚类可以充分发挥其优势，对样本进行有效的聚类，实现更合理化的基站分配，以满足实际部署中的要求。目前的研究工作中，鲜有谱聚类算法在 MEC 环境中解决放置边缘服务器放置

问题的相关研究。因此，本文提出在考虑用户分布的情况下，首次利用谱聚类进行边缘服务器放置方案的设计，实现更低的访问时延和更均衡的边缘服务器工作负载。

3 研究场景及问题描述

3.1 研究场景

在 MEC 网络系统中[48]，基站是进行网络通信的结点，边缘服务器是进行网络计算的结点。国内通信运营商考虑到边缘服务器的放置成本问题，在实际的网络部署中，边缘服务器往往被部署在某一个基站。边缘智能设备可以通过基站将计算任务发送至边缘服务器，边缘服务器也是通过基站为边缘智能设备提供网络服务和资源共享。但由于边缘移动设备的分布较为分散且密度不均，因此每个基站上连接的用户数目并不相同，用户数目越多的基站将拥有较高的优先级放置边缘服务器。本文的实验场景是在已经部署好的基站中，寻找边缘服务器的部署位置，实现用户平均访问延迟和边缘服务器平均负载的最小化。

3.2 问题描述

本节首先对边缘服务器放置问题进行数学建模，将移动边缘网络建模为无向图 $G=(V,E)$，其中 V 是样本空间中所有基站的集合，E 是基站之间的连接边。$B=\{b\}$ 和 $S=\{s\}$ 分别代表基站集合和边缘服务器集合，l_b 和 l_s 分别代表基站的位置和边缘服务器的位置，d 表示基站和边缘服务器之间的距离。在移动边缘网络中，访问时延与距离成正比，因此可以将基站到边缘服务器的距离等效表示为用户通过基站访问边缘服务器资源时的访问时延。B_s 表示访问边缘服务器 $s\in S$ 资源的基站集合，W_b 表示基站 $b\in B$ 的工作负载。访问边缘服务器 s 的基站工作负载之和表示该边缘服务器的工作负载，记为 W_s。本文的研究目标是通过不断优化放置方案来降低访问时延（见式(1)）和均衡服务器的工作负载（见式(2)）。

$$D(L)=\min\max d(l_b,l_s),b\in B,s\in S \tag{1}$$

$$W(L)=\min\max(W_i-W_j),i\in S,j\in S \tag{2}$$

其中，L 是边缘服务器放置方案，$D(L)$ 是该放置方案下的访问时延；$d(l_b,l_s)$ 是基站 $d(l_b,l_s)$ 与边缘服务器 s 的距离；$W(L)$ 是该放置方案下任意两台边缘服务器的工作负载之差，W_i 和 W_j 分别表示边缘服务器 i 与边缘服务器 j 的工作负载。

在移动边缘网络系统中，每个基站能且仅能被分配给一个边缘服务器，也就是说，访问边缘服务器 i 和边缘服务器 j 的两组基站之间不可以有交集，如式(3)所示。其中，E_i 和 E_j 是访问边缘服务器 i 和边缘服务器 j 的基站集合。同时，边缘服务器可以为网络系统中所有的基站提供计算和存储资源，即边缘服务器服务的所有基站集合为 B，如式(4)所示。

$$E_i\bigcap E_j=\emptyset \tag{3}$$

$$\bigcup E_s=B,s\in S \tag{4}$$

基站的工作负载是其连接的所有用户的上网时间之和，基站 b 的工作负载计算如式(5)所示，其中 t_i 为智能终端设备 i 的上网时长。边缘服务器的工作负载是连接该服务器的所有基站的工作负载之和，如式(6)所示。

$$W_b=\sum_{i=1}^{m}t_i \tag{5}$$

$$W_s=\sum_{b\in E_s}W_b \tag{6}$$

实验的优化目标是将访问时延和工作负载最小化，根据式(1)－式(6)，将本次实验的优化目标表述为式(7)。可以看出，该问题是一个多目标优化的边缘服务器放置问题[19-21]。

$$\begin{aligned}&\min(D(L)+W(L))\\&\text{s.t. }(1)-(6)\end{aligned} \tag{7}$$

4 基于用户分布的边缘服务器放置算法

针对多目标优化的边缘服务器放置问题，本文提出了基于用户分布的延迟最小化边缘服务器放置算法 LAMP，算法流程图如图 1 所示。该算法的输入是一个基站信息的三元组 (x_b,y_b,u_b)，其中 x_b 是基站 b 的经度，y_b 是基站 b 的纬度，u_b 是基站 b 的用户数目。首先进行数据集的预处理，将数据集划分为两类。接着对不同基站集进行不同的聚类处理，将预处理之后的数据集分别进行单独聚类和谱聚类（Spectral Clustering，SC），具体细节将在 4.2 节中展开介绍。在得到聚类结果之后在聚类集群内部计算基站间的距离，进而确定聚类中心。聚类中心就是边缘服务器的放置位置，最终得到边缘服务器的放置方案。LAMP 算法的具体步骤如算法 1 所示。

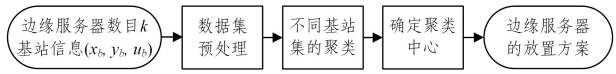


图 1 边缘服务器放置算法流程图

Fig. 1 Flowchart of edge server placement algorithm

算法 1 LAMP 算法

输入：边缘服务器数目 k，基站信息(x_b，y_b，u_b)，b∈B

输出：L(l_{b_1}，l_{b_2}，l_{b_3}，…，l_{b_k})/ * 放置方案 * /

1. 对数据集中的基站进行预处理：对比 SC(k，B)和 K-means 的聚类结果，将基站数据集 B 分为 CPS 和 NCPS，NCPS 集合的大小为 n，n＜k；
2. 对 NCPS 中的每个基站 b，都单独放置一台边缘服务器，得到边缘服务器的放置方案 L_1(l_{b_1}，l_{b_2}，l_{b_3}，…，l_{b_n})；
3. For iteration＝1，2，3，…，DO
4. SC(k－n，CPS)；
5. End for
6. 对每一个集群，根据 4.3 节中的式(8)和式(9)计算基于用户分布的基站间的距离，得到边缘服务器的放置方案 L_2($l_{b_{n+1}}$，$l_{b_{n+2}}$，$l_{b_{n+3}}$，…，l_{b_k})；
7. L＝$L_1\cup L_2$；
8. 返回 L(l_{b_1}，l_{b_2}，l_{b_7}，…，l_{b_k})。

4.1 数据集预处理

在聚类之前，首先需要对基站数据集进行预处理，将其分为中心基站集（Center Point Set，CPS）和非中心基站集（Non-Center Point Set，NCPS），如图 2 所示。使用谱聚类算法对原始数据集进行聚类，将得到的聚类结果与 K-means 的聚类结果进行对比，将类内距离较大的聚类标号记录下来，拥有这些聚类标号的基站被标记为非中心基站，剩余基站被标记为中心基站。具体地，将数据集中的各个基站表示为二维空间中的点，对原始基站数据集的样本空间使用谱聚类和 K-means 算法分别进行聚类。得到聚类结果后，计算两种算法下每个集群的类内距离。通过对比可知，使用谱聚类算法得到的聚类结果中存在某些集群的类内距离超过了 K-means 算法聚类结果中的最大类内距离的情况，这些集群一般由非中心

基站组成，我们将这些集群包含的所有基站标记为非中心基站。经过多次实验和反复筛选，将那些被高频标记为非中心基站的基站划分为非中心基站集，其他基站划分为中心基站集(算法1第1行)。例如，在图2中，通过聚类和筛选，聚类标号为1—8的集群是谱聚类算法中聚类结果大于K-means算法聚类结果的最大类内距离的被标记集群。聚类标号为1—8的基站被标记为非中心基站，其他集群中的基站被标记为中心基站。

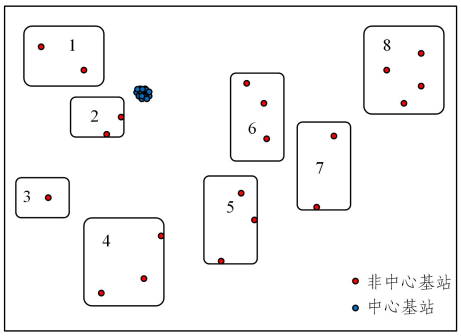


图2 数据集预处理示意图

Fig. 2 Schematic diagram of dataset preprocessing

4.2 不同基站集的聚类

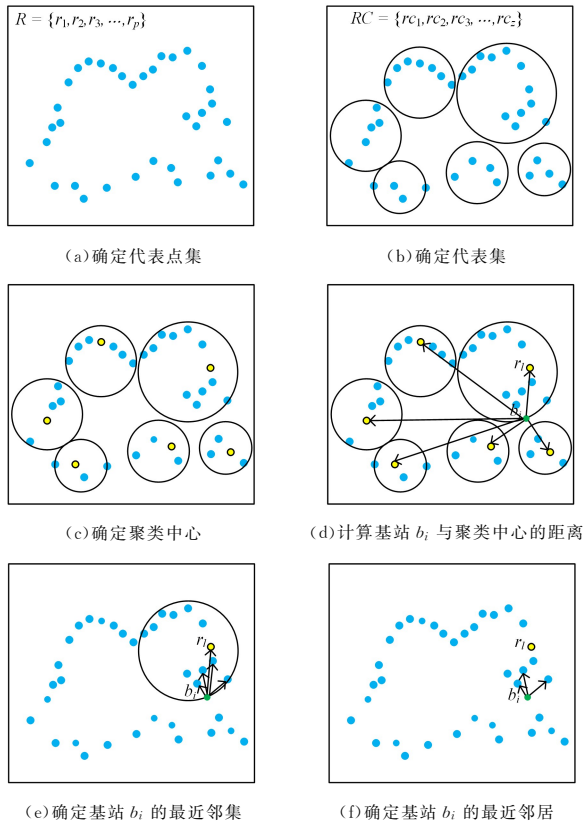
对预处理后的两个基站集进行分别处理。对非中心基站集，采用单独放置的方式部署边缘服务器，每一个基站上都放置一个边缘服务器(算法1第2行)。非中心基站集中的每个基站都作为一个单独的集群，显然，该基站就是该集群的聚类中心，边缘服务器的部署位置就是该基站的位置。由于边缘服务器的总数是k，非中心基站集的大小为n，因此中心基站集将放置$k-n$个边缘服务器(算法1第4行)。使用谱聚类算法对中心基站集进行聚类。为了提高初始中心点的选择有效性，谱聚类[12]算法首先采用随机选择和K-means混合选择策略来选择聚类的p'个候选中心点(算法2第1行)。其次，采用K-means算法获得p个聚类，并将这p个集群的聚类中心作为初始中心点。p个初始中心点构成了代表点集$R(r_1, r_2, r_3, \cdots, r_p)$(算法2第2行)，如图3(a)所示，通过混合选择策略在样本空间中得到了初始中心点。再次，使用K-means算法将代表点集R再次划分为z个代表集$RC(rc_1, rc_2, rc_3, \cdots, rc_z)$(算法2第3行)，如图3(b)所示，调用$K$-means算法将代表点集$R$中的初始中心点再次划分为$z$个集群，这$z$个集群共同组成了代表集$RC$。然后，计算代表集$RC$中每个集群的集群中心(算法2第4行)，如图3(c)所示，被标记为黄色的样本点为所在集群的中心点。然后，计算每个基站样本点与集群中心点的距离，如图3(d)所示，选择具有最小距离的集群中心点所在的集群作为基站b_i的最近邻集RCB_i(算法2第5行)，如图3(e)所示。其中，最近邻集RCB_i的集群中心为r_l。根据r_l，我们可以得到基站b_i与r_l的最近邻之间的距离。通过比较距离的大小，最终找到基站b_i的K-最近邻居(算法2第6行)，如图3(f)所示。随后，构造基站数据集CPS与代表点集R的稀疏亲和子矩阵$\boldsymbol{A}$(算法2第7行)，并将该矩阵表示为二部图$G(CPS, R, \boldsymbol{A})$(算法2第8行)，如图4所示。其中，$CPS$和$R$是二部图$G$的顶点集，稀疏亲和子矩阵$\boldsymbol{A}$中所有的非零项构成二部图的边集。从图3(f)可以看到，每个样本点有K个最近邻居。因此在图4中，CPS中每个基站b_i与代表点集R中的K个初始中心点相连。最后，利用transfer cut技术[49]切割二部图G(算法2第9行)。其中，每个基站只能被分配给一个初始中心点处的边缘服务器。从而最终得到中心基站集的聚类结果(算法2第10行)。

算法2 谱聚类算法SC

输入：聚类数目p，基站数据集B

输出：label /* 类标号 */

1. 采用随机选择和K-means混合选择策略来选择聚类的p′个候选中心点；
2. 使用K-means算法将p′个候选中心点分为p个集群，集群中心点构成代表点集$R(r_1, r_2, r_3, \cdots, r_p)$，$p'=10p$；
3. 将包含p个中心点的代表点集R划分为z个代表集$RC(rc_1, rc_2, rc_3, \cdots, rc_z)$，$z=\lfloor\sqrt{p}\rfloor$；
4. 计算代表集RC中每个集群的集群中心；
5. 根据式(8)和式(9)计算每一个基站与集群中心的距离，距离最小的集群作为基站b_i的最近邻集$RCB_i=rc_j$，$b_i \in B$；
6. 最近邻集$RCB_i=rc_j$的集群中心记为r_l，根据r_l的最近邻居找到基站b_i的K-最近邻居；
7. 基站集B与代表点集R之间的关系表示为稀疏亲和子矩阵$\boldsymbol{A}$；
8. 将稀疏亲和子矩阵$\boldsymbol{A}$表示为二部图G(B,R,A)；
9. 利用transfer cut技术有效地划分二部图G；
10. B中的每个基站只能与代表点集R中的一个顶点相连，从而得到基站的类标号label；
11. 返回类标号label。



(a)确定代表点集 (b)确定代表集

(c)确定聚类中心 (d)计算基站b_i与聚类中心的距离

(e)确定基站b_i的最近邻集 (f)确定基站b_i的最近邻居

图3 确定基站的K-最近邻居

Fig. 3 Determine the K-nearest neighbors of base station

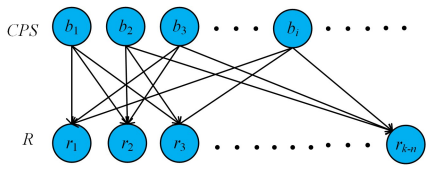


图 4 二部图

Fig. 4 Bipartite

4.3 确定聚类中心

从实际出发，边缘服务器放置问题主要受到基站的地理位置这一属性的影响，但为了实现边缘服务器尽可能靠近终端设备这一目标，还应该考虑到用户分布情况对边缘服务器放置问题的影响。用户的分布情况主要通过基站连接的用户数目来体现，因此加入基站的用户数目这一参数可以使得边缘服务器放置更加合理。故我们在式(8)中根据基站的地理位置和基站连接的用户分布重新定义基站的距离，然后根据新定义的距离确定每个集群的聚类中心。具体地，对于非中心基站集，每个集群只有一个基站，该基站就是该集群的聚类中心，聚类中心即为该集群中边缘服务器的放置位置，非中心基站集中的每个基站都是一个聚类中心，从而可以得知非中心基站集的放置方案即为非中心基站集本身(算法 1 第 2 行)；对于中心基站集，得到聚类结果之后，根据式(8)中我们对基站距离的新定义，选择距离之和最小的基站(见式(9))作为所在集群的聚类中心，该基站为该集群的中心点基站，即边缘服务器的放置位置，从而得到中心基站集中边缘服务器的放置方案(算法 1 第 6 行)。对于基站数据集来说，边缘服务器的放置方案由非中心基站集的放置方案和中心基站集的放置方案两部分组成(算法 1 第 7 行)。

$$d(l_{b_n},l_{b_m})=W_1*dist(l_{b_n},l_{b_m})+W_2*u_{b_n} \tag{8}$$

$$l_{s_i}=\{l_{b_m}\mid b_m=\operatorname{argmin}\sum d(l_{b_n},l_{b_m})\},b_n\in E_{s_i},b_m\in E_{s_i} \tag{9}$$

其中，$dist(l_{b_n},l_{b_m})$是基站 b_n和 b_m基于地理位置在地球上的实际距离，u_{b_n}是基站 b_n的用户数目，$d(l_{b_n},l_{b_m})$表示基站 b_n连接基站 b_m的访问距离。在基站距离的新定义中，加入用户数目 u_{b_n}后，可降低用户数目大的基站聚为一类的概率，从而可实现边缘服务器的负载均衡。$\sum d(l_{b_n},l_{b_m})$是该集群E_{s_i}中其他基站连接基站 b_m的总访问距离，我们以总访问距离最小的基站作为本集群边缘服务器的部署位置，显然，该基站的平均访问距离也是本集群中最小的，该基站的平均访问距离即为本集群的访问距离，边缘服务器的放置方案就是由每个集群中平均访问距离最小的基站位置组成的。

5 实验与结果分析

5.1 数据集描述

本文进行实验的数据集来自于上海电信的基站数据集[5,20-22]，数据集主要包括基站的分布情况以及用户接入基站的相关信息，数据集由上海电信提供，包含 9 481 台边缘设备通过 3 233 个基站在 2014 年 6 月至 2014 年 12 月期间访问互联网的超过 720 万条记录，记录着用户 ID、上网时间和下网时间以及访问基站的位置信息。

5.2 实验设置

本文实验在具有以下配置的电脑上运行：Intel(R) Core (TM) i5-9500 CPU @3.00 GHz 3.00 GHz，16 GB RAM 和 Windows 10 OS。所有的实验环境均使用 Python 3.7 和 MATLAB 2016b 实现。

实验将所提方法与当前主流算法在访问延迟和工作负载方面进行了比较。针对边缘服务器的放置问题，目前主流的算法有 MIP 算法[20]、传统 K-means 算法[40-41]和 Random[20]算法。具体地，MIP 算法通过求解混合整数线性规划问题得到放置方案。K-means 算法的主要思想是在初始聚类中随机选择聚类中心，然后不断细化聚类以调整聚类中心，最后输出基于当前初始聚类中心的最佳聚类结果。Random 算法则随机选择基站作为边缘服务器在网络系统中的部署位置。

5.3 参数设置

本实验考虑了用户分布对边缘服务器放置产生的影响，但是基站的地理位置以及用户数目对边缘服务器的放置问题产生的影响力不同，因此加入了权重来平衡两者的影响力。将 W_1作为基站地理位置的权重，将 W_2作为基站用户数目的权重。因此，在边缘服务器放置算法评估之前，首先需要确定式(8)中 W_1和 W_2的值，实验结果如图 5 所示。

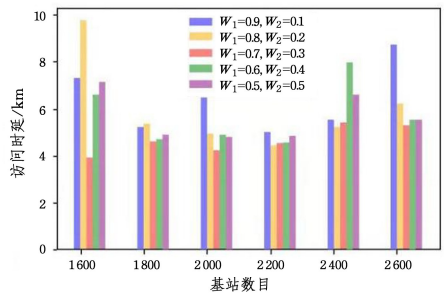


图 5 不同权重下 LAMP 算法的访问时延

Fig. 5 Access delay of LAMP algorithm with different weights

从图 7 可以看到，在不同的基站数目下，当 $W_1=0.7$，$W_2=0.3$ 时，基于用户分布的延迟最小化边缘服务器放置算法的访问时延明显低于其他权重时的时延。因此，在后面的实验中，都将使用 $W_1=0.7$，$W_2=0.3$ 这一权重参数。

5.4 结果分析

本文中实验所用的数据为 2014 年 6 月上海电信用户的上网信息，包含 7 692 台边缘设备通过 3 042 个基站产生的超过 113 万条有效上网记录，以测试各放置算法在真实网络场景下的性能。实验使用 2014 年 6 月 1 日至 6 月 30 日的用户上网记录进行算法性能测试。

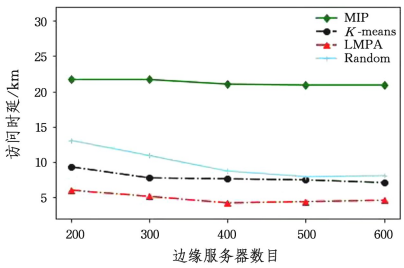
在实验中，将边缘服务器与所在集群中其他基站的基于用户分布的访问距离等效为基站的访问时延，并将同一集群中所有基站上连接的边缘设备的上网时长之和记为边缘服务器的工作负载。对比和分析不同算法的边缘服务器放置方案在工作负载方面的性能时，采用计算工作负载的标准差来评估边缘服务器的负载是否均衡，如式(10)和式(11)所示。在计算 LAMP 算法在负载均衡方面的提高率时，本文采用提高率的基础计算公式，如式(12)所示，其中 X_1为 LAMP 算法的实验数据，X_2为对比算法的实验数据，代入数据可得 LAMP 算法的性能与对比算法相比的提高率。

$$\bar{w}=\frac{\sum W_{s_i}}{k}, i=1,2,3,\cdots,k \tag{10}$$

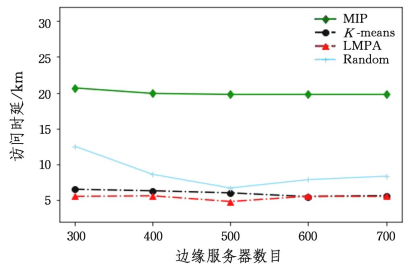
$$\sigma_W=\sqrt{\frac{\sum(\bar{w}-W_{s_i})^2}{k}} \tag{11}$$

$$\beta=\frac{X_1-X_2}{X_2}*100\% \tag{12}$$

为了证明本文提出的 LAMP 算法在优化终端用户访问时延和均衡服务器负载方面的良好性能，首先在固定基站数量的移动边缘网络中放置不同数量的边缘服务器来进行测试。具体来说，在 2000 个基站的移动边缘网络中分别放置 200，300，400，500，600 台边缘服务器，并记录用户 2014 年6 月 1 日至 6 月 15 日的平均访问延迟和服务器的平均负载，实验结果如图 6(a)和表 2 所示。从图6(a)可以看出，随着边缘服务器数目的增加，不同算法的访问延迟都在下降，且 LAMP 算法的曲线始终位于其他曲线的下方。这表明在放置不同服务器数目的情况下，本文提出的方法 LAMP 得到的边缘服务器放置方案与其他算法相比具有更小的访问时延。表 1 列出了图 6(a)实验中对应的放置方案下边缘服务器的负载情况，可以看出，随着放置服务器数目的增加，边缘服务器的平均负载在下降。由此可得，放置更多的边缘服务器可以使服务器的工作负载变得越来越均衡。根据表 1 中第 3 列和第 4 列数据可计算得出 K-means 算法和 LAMP 算法的负载标准差的平均值，代入式(12)计算可知，在均衡负载方面，LAMP 算法的性能比 K-means 算法平均提高了大约82.85%。本文基于 2014 年 6 月 16 日至 6 月 30 日的用户上网记录进行实验，在 2000 个基站的移动边缘网络中分别放置 300，400，500，600，700 台边缘服务器，并记录用户的平均访问延迟和服务器的平均负载，实验结果如图 6(b)和表 2 所示。在图 6(b)中可以看到，LAMP 算法和 K-means 算法两者的曲线非常接近，但 LAMP 算法的曲线始终低于 K-means 算法的曲线。可以看出 LAMP 算法在不同边缘服务器数目的情况下，仍然可以对服务器进行最佳部署，以实现最低的访问时延。因此可以得出，LAMP 算法在降低访问延迟方面表现出了卓越的性能优势。



(a)2014-06-01－2014-06-15 基站数据集



(b)2014-06-16－2014-06-30 基站数据集

图 6 不同边缘服务器数目下边缘服务器的访问时延

Fig. 6 Access latency of edge servers with different numbers of edge servers

表 1 不同边缘服务器数目下边缘服务器的负载标准差

(2014-06-01－2014-06-15)

Table 1 Workload standard deviation of edge servers with different number of edge servers(2014-06-01－2014-06-15)

Algorithm	MIP	Random	K-means	LAMP
$N=2000,k=200$	14.55	16.02	90.47	16.92
$N=2000,k=300$	14.55	12.17	71.57	12.62
$N=2000,k=400$	9.38	9.71	60.67	9.48
$N=2000,k=500$	8.56	8.47	53.40	8.55
$N=2000,k=600$	8.56	7.57	45.47	7.58

表 2 是图 6(b)实验中对应的放置方案下边缘服务器的负载情况。

表 2 不同边缘服务器数目下边缘服务器的负载标准差

(2014-06-16－2014-06-30)

Table 2 Workload standard deviation of edge servers with different number of edge servers(2014-06-16－2014-06-30)

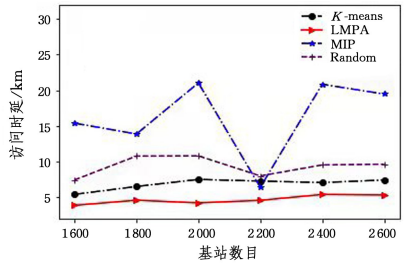
Algorithm	MIP	Random	K-means	LAMP
$N=2000,k=300$	13.05	11.33	54.93	11.67
$N=2000,k=400$	8.82	9.44	46.28	9.46
$N=2000,k=500$	8.05	8.06	42.12	7.79
$N=2000,k=600$	8.05	7.45	34.22	7.31
$N=2000,k=700$	8.05	6.71	35.21	6.69

从表 2 中可以看出，LAMP 算法与 MIP 算法和 Random 算法在负载均衡方面的性能相差较小，根据表 2 中第 3 列和第 4 列数据可算得 K-means 算法和 LAMP 算法的负载标准差的平均值，代入式(12)计算可知，在工作负载方面，LAMP 算法的性能比 K-means 算法平均提高了 79.83%。

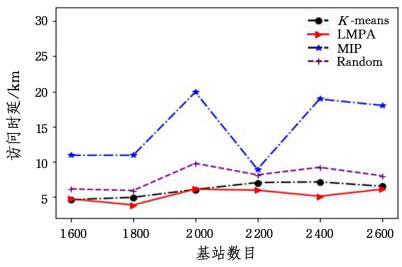
另外，我们在基站数量逐增的移动边缘网络系统中放置固定数量的边缘服务器来进行算法性能的测试。具体地，将 400 个边缘服务器分别放置在 1600，1800，2000，2200，2400，2600 个基站数目的移动边缘网络中，用户 2014 年 6 月 1 日至 6 月 15 日的平均访问延迟和服务器的平均负载实验结果如图 7(a)和表 3 所示。在图 7(a)中，LAMP 算法的访问时延与其他算法相比始终是最低的，且波动幅度较小。由此可以得出，在不同的基站数目下，LAMP 算法得到的边缘服务器放置方案具有更低的访问时延，且算法性能较为稳定。表 3 列出了图 7(a)实验中边缘服务放置方案的服务器负载情况，从表中可以看到不同算法的工作负载情况，随着基站数目的增加，其他放置算法的负载标准差呈明显的增大趋势，但是 LAMP 算法的负载标准差的变化一直较小。根据表 3 中第 3 列和第 4 列数据可算得 K-means 算法和 LAMP 算法的负载标准差的平均值，代入式(12)计算可知，在工作负载方面 LAMP 算法的性能比 K-means 算法平均提高了大约 71.25%，尽可能地实现了边缘服务器的负载均衡。图 7(b)

给出了基于2014年6月16日至6月30日的用户上网记录进行实验的结果，在1600，1800，2000，2200，2400，2600个基站数目的移动边缘网络中分别放置400个边缘服务器，以测试不同放置算法的性能。从图7(b)中可以看到，MIP算法和Random算法的实验曲线变化幅度较明显，在基站数目为1800和2400的移动边缘网络中，LAMP算法的访问延迟明显低于其他放置算法。从整体来看，LAMP算法的曲线始终位于其他放置算法的下方，由此可知，LAMP算法可在不同基站数目的场景中做到访问时延最低。

表4列出了图7(b)实验中边缘服务器放置方案的服务器负载情况，LAMP算法的服务器负载标准差与MIP算法和Random算法始终保持较小的差距。



(a)2014-06-01－2014-06-15 基站数据集



(b)2014-06-16－2014-06-30 基站数据集

图7 不同基站数目下边缘服务器的访问时延

Fig. 7 Access latency of edge servers with different numbers of base stations

根据表4中第3列和第4列数据可计算得出*K*-means算法和LAMP算法的负载标准差的平均值，代入式(12)计算可知，在工作负载方面LAMP算法的性能比*K*-means算法平均提高了82.29%。与其他算法相比，LAMP算法不仅实现了边缘设备的最低访问时延，并且较好地实现了边缘服务器的工作负载均衡，其可以在基站不断增加的移动边缘网络中将服务器部署在最佳的位置，更适用于实际生活中变化的网络场景。

表3 不同基站数目下边缘服务器的负载标准差(2014-06-01－2014-06-15)

Table 3 Workload standard deviation of edge servers with different number of base stations(2014-06-01－2014-06-15)

Parameters	MIP	Random	*K*-means	LAMP
$N=1600,k=400$	5.09	6.86	51.19	30.59
$N=1800,k=400$	6.68	9.38	56.33	15.98
$N=2000,k=400$	9.38	9.85	60.01	15.54
$N=2200,k=400$	9.71	10.38	61.48	15.11
$N=2400,k=400$	10.33	11.45	68.03	15.39
$N=2600,k=400$	10.48	11.69	79.07	15.52

表4 不同基站数目下边缘服务器的负载标准差(2014-06-16－2014-06-30)

Table 4 Workload standard deviation of edge servers with different number of base stations(2014-06-16－2014-06-30)

Algorithm	MIP	Random	*K*-means	LAMP
$N=1600,k=400$	4.81	6.58	35.94	4.86
$N=1800,k=400$	4.81	6.63	42.62	5.17
$N=2000,k=400$	8.82	9.42	44.80	9.43
$N=2200,k=400$	9.65	9.94	52.38	10.97
$N=2400,k=400$	9.99	10.15	54.67	10.22
$N=2600,k=400$	9.78	10.98	60.50	10.87

为了更好地比较不同算法在访问时延方面的性能，在表5中对实验结果进行了量化分析，记录了在不同边缘服务器数目和不同基站数目时，不同算法得到的服务器放置方案的平均用户访问延迟。可以看到，在2014年6月1日至6月15日时间段内，不同边缘服务器数目下，LAMP算法得到的用户平均访问延迟比MIP算法降低了77.1%，比Random算法降低了49.9%，比*K*-means算法降低了37.9%。在2014年6月1日至6月15日时间段内，不同基站数目下，LAMP算法得到的用户平均访问延迟比MIP算法降低了71.1%，比Random降低了50.1%，比*K*-means降低了32.1%。在2014年6月16日至6月30日时间段内，不同边缘服务器数目下，LAMP算法得到的用户平均访问延迟比MIP算法降低了72.9%，比Random算法降低了38.6%，比*K*-means算法降低了9.8%。在2014年6月16日至6月30日时间段内，不同基站数目下，LAMP算法得到的用户平均访问延迟比MIP算法降低了63.7%，比Random降低了32.5%，比*K*-means算法降低了14.2%。

表5 不同边缘服务器数目下算法的平均访问延迟

Table 5 Average access delay of algorithms with different number of edge servers

Algorithm	(2014-06-01－2014-06-15)		(2014-06-16－2014-06-30)		(2014-06-01－2014-06-15)		(2014-06-16－2014-06-30)	
	平均访问时延	时延降低比率/%	平均访问时延	时延降低比率/%	平均访问时延	时延降低比率/%	平均访问时延	时延降低比率/%
MIP	21.23	77.1	16.16	71.1	19.96	72.9	14.59	63.7
Random	9.74	49.9	9.36	50.1	8.81	38.6	7.85	32.5
K-means	7.86	37.9	6.87	32.1	6.00	9.8	6.05	14.2
LAMP	4.88	—	4.66	—	5.41	—	5.30	—

结束语 本文针对移动边缘网络系统中边缘服务器的放置问题，提出了基于用户分布的延迟最小化边缘服务器放置算法LAMP来解决实际问题。我们在对基站的聚类中考虑了基站连接的用户数目以及基站的地理位置，定义了新的基站距离，并使用谱聚类来确定聚类方案和服务器放置位置，实现用户访问延迟的降低和服务器工作负载的均衡。最后，使用上海电信提供的基站数据集进行了大量的实验来验证算法的性能。实验结果表明，基于用户分布的延迟最小化边缘

服务器放置算法在降低访问延迟和均衡边缘服务器工作负载方面具有突出的性能优势。

在未来的工作中，我们会从服务提供商的角度，专注于边缘服务器的能耗问题和服务利润等相关问题的研究。因此，如何设计一个低成本高利润的移动边缘计算网络是我们未来工作的重点。

参 考 文 献

[1] SHI W S, ZHANG X Z, WANG Y F, et al. Edge Computing: State-of-the-Art and Future Directions [J]. Journal of Computer Research and Development, 2019, 56(1): 69-89.

[2] QU Z, YE B, CHEN G, et al. State-of-the-art Survey on Resource Optimization in Edge Computing[J]. Big Data Research, 2019, 5(2): 17-33.

[3] LI Z Y, WANG Q, CHEN Y F, et al. A Survey of Task Offloading Research in Vehicle Edge Computing Environment [J]. Chinese Journal of Computers, 2021, 44(5): 963-982.

[4] LI S Y, LIU H, ZHANG Z Y, et al. A Smart Home-oriented Edge Computing Bandwidth Resource Allocation Method and System, China, CN113507519A[P]. 2021-10-15.

[5] LI Y, ZHOU A, MA X, et al. Profit-aware Edge Server Placement [J]. IEEE Internet of Things Journal, 2023, 9(1): 55-67.

[6] ZHOU Y Z, ZHANG D. Near-end Cloud Computing: Opportunities and Challenges in the Post-cloud Computing Era [J]. Chinese Journal of Computers, 2019, 42(4): 677-700.

[7] LIU Y, WANG T, PENG S L, et al. Edge-based Federated Learning Model Cleaning and Device Clustering Methods [J]. Chinese Journal of Computers, 2021, 44(12): 2514-2528.

[8] SHI W S, SUN H, CAO J, et al. Edge Computing: An emerging Computing Model for the Internet of Everything Era [J]. Journal of Computer Research and Development, 2017, 54(5): 907-924.

[9] ZHANG Y L, LIANG Y Z, YIN M J, et al. A Review of Research on Computing Offload Schemes in Mobile Edge Computing [J]. Chinese Journal of Computers, 2021, 44(12): 2406-2430.

[10] XIANG H, XU X, ZHENG H, et al. An Adaptive Cloudlet Placement Method for Mobile Applications over GPS Big Data [C]// 2016 IEEE Global Communications Conference (GLOBECOM). 2016: 1-6.

[11] ZHANG Y, WANG K, ZHOU Y, et al. Enhanced Adaptive Cloudlet Placement Approach for Mobile Application on Spark [C]// Security and Communication Networks. 2018: 1-12.

[12] FAJARDO J O, LIBERAL F, GIANNOULAKIS I, et al. Introducing Mobile Edge Computing Capabilities through Distributed 5G Cloud Enabled Small Cells [J]. Mobile Networks & Applications, 2016, 21(4): 564-574.

[13] MA L, WU J, CHEN L, et al. DOTA: Delay Bounded Optimal Cloudlet Deployment and User Association in WMANs[C]// 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). 2017: 196-203.

[14] FAN Q, ANSARI N. Cost Aware Cloudlet Placement for Big Data Processing at the Edge[C]// Proceedings of the 2017 IEEE International Conference on Communications (ICC). 2017: 1-6.

[15] YANG S, LI F, SHEN M, et al. Cloudlet Placement and Task Allocation in Mobile Edge Computing [J]. IEEE Internet of Things Journal, 2019, 6(3): 5853-5863.

[16] ZENG F, REN Y, DENG X, et al. Cost-Effective Edge Server Placement in Wireless Metropolitan Area Networks [J]. IEEE Internet of Things Journal, 2019, 19(1): 5853-5863.

[17] SANTOYO-GONZÁLEZ A, CERVELLÓ-PASTOR C. Edge Nodes Infrastructure Placement Parameters for 5G Networks [C]// 2018 IEEE Conference on Standards for Communications and Networking (CSCN), 2018: 1-6.

[18] DAYARATHNA M, WEN Y, FAN R. Data Center Energy Consumption Modeling: A Survey [J]. IEEE Communications Surveys Tutorials, 2016, 18(1): 732-794.

[19] WANG S, LIU Z, ZHENG Z, et al. Particle Swarm Optimization for Energy-aware Virtual Machine Placement Optimization in Virtualized Data Centers[C]// 2013 International Conference on Parallel and Distributed Systems. 2013: 102-109.

[20] WANG S, ZHAO Y, XU J, et al. Edge Server Placement in Mobile Edge Computing [J]. Journal of Parallel and Distributed Computing, 2019, 127: 160-168.

[21] 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.

[22] WANG S, GUO Y, ZHANG N, et al. Delay-Aware Microservice Coordination in Mobile Edge Computing: A Reinforcement Learning Approach [J]. IEEE Transactions on Mobile Computing, 2021, 20(3): 939-953.

[23] LIANG Y, LIU H, DINESH R. Optimal Placement and Configuration of Roadside Units in Vehicular Networks[C]// Proceedings of the 2012 IEEE 75th Vehicular Technology Conference (VTC Spring). 2012: 1-6.

[24] ASLAM B, AMJAD F, ZOU C C. Optimal roadside units placement in urban areas for vehicular networks[C]// 2012 IEEE Symposium on Computers and Communications. 2012: 423-429.

[25] TRULLOLS O, FIORE M, CASETTI C, et al. Planning Roadside Infrastructure for Information Dissemination in Intelligent Transportation Systems [J]. Computer Communications, 2010, 33(4): 432-442.

[26] BALOUCHZAHI N M, FATHY M, AKBARI A. Optimal Road Side Units Placement Model Based on Binary Integer Programming for Efficient Traffic Information Advertisement and Discovery in Vehicular Environment[C]// IET Intelligent Transport Systems. 2015: 851-861.

[27] WANG Z H, ZHENG J, WU Y, et al. A Centrality-based RSU Deployment Approach for Vehicular Ad Hoc Networks[C]// Proceedings of the 2017 IEEE International Conference on Communications (ICC). 2017: 1-5.

[28] ZHANG R, YAN F, XIA W, et al. An Optimal Roadside Unit Placement Method for Vanet Localization [C]// 2017 IEEE Global Communications Conference (GLOBECOM 2017). 2017: 1-6.

[29] XU X L, FANG Z J, QI L Y, et al. Distributed Service Offload Method Based on Deep Reinforcement Learning in the Edge

Computing Environment of the Internet of Vehicles [J]. Chinese Journal of Computers,2021,44(12):2382-2405.
[30] PREMSANKAR G,GHADDAR B,FRANCESCO M,et al. Efficient Placement of Edge Computing Devices for Vehicular Applications in Smart Cities[C]// 2018 IEEE/IFIP Network Operations and Management Symposium(NOMS 2018). 2018:1-9.
[31] CESELLI A,PREMOLI M,SECCI S. Cloudlet Network Design Optimization[C]// 2015 IFIP Networking Conference. 2015:1-9.
[32] SHI W,CAO J,ZHANG Q,et al. Edge Computing:Vision and Challenges [J]. IEEE Internet of Things Journal,2016,3(5):637-646.
[33] DASHTI S E,RAHMANI A M. Dynamic VMs placement for energy efficiency by PSO in cloud computing [J]. Journal of Experimental & Theoretical Artificial Intelligence,2016,28:97-112.
[34] JIA M,CAO J,LIANG W. Optimal Cloudlet Placement and User to Cloudlet Allocation in Wireless Metropolitan Area Networks [J]. IEEE Transactions on Cloud Computing,2017,5(4):725-737.
[35] CHEN X,JIAO L,LI W,et al. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing [J]. IEEE/ACM Transactions on Networking,2016,24(5):2795-2808.
[36] MANASVI G,CHAKRABORTY A,MANOJ B S. Social Network Aware Dynamic Edge Server Placement for Next-Generation Cellular Networks[C]// 2020 International Conference on COMmunication Systems & NETworkS(COMSNETS). 2020:499-502.
[37] MARK C,NIYATO D,CHEN-KHONG T. Evolutionary Optimal Virtual Machine Placement and Demand Forecaster for Cloud Computing[C]// 2011 IEEE International Conference on Advanced Information Networking and Applications. 2011:348-355.
[38] XU Z,LIANG W,XU W,et al. Capacitated Cloudlet Placements in Wireless Metropolitan Area Networks[C]// Proceedings of the 2015 IEEE 40th Conference on Local Computer Networks (LCN). 2015:570-578.
[39] MENG J,SHI W,TAN H,et al. Cloudlet Placement and Minimum-delay Routing in Cloudlet Computing[C]// Proceedings of the 2017 3rd International Conference on Big Data Computing and Communications(BIGCOM). 2017:297-304.
[40] WAGSTAFF K L,CLAIRE C,SETH R,et al. Constrained K-means Clustering with Background Knowledge[C]// ICML. 2001.
[41] WANG G,ZHAO Y,HUANG J,et al. A *K*-means-based Network Partition Algorithm for Controller Placement in Software Defined Network[C]// 2016 IEEE International Conference on Communications(ICC). 2016:1-6.
[42] CUI G,HE Q,XIA X,et al. Robustness-oriented k Edge Server Placement[C]// 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing(CCGRID). 2020:81-90.
[43] FAN Q,ANSARI N. On Cost Aware Cloudlet Placement for Mobile Edge Computing [J]. IEEE/CAA Journal of Automatica Sinica,2019,6(4):926-937.
[44] GUO F,TANG B. Mobile Edge Server Placement Method Based on User Latency-aware [J]. Computer Science,2021,48(1):103-110.
[45] ZHAO X B,ZHAO Y F,LI B,et al. Edge Server Placement Method Based on Latency and Energy Perception [J]. Computer Engineering,2021,47(11):37-43.
[46] ZHANG P C,WEI X M,JIN H Y. Dynamic QoS Optimization Based on Federated Learning Under Mobile Edge Computing [J]. Chinese Journal of Computers,2021,44(12):2431-2446.
[47] HUANG D,WANG C,WU J,et al. Ultra-Scalable Spectral Clustering and Ensemble Clustering [J]. IEEE Transactions on Knowledge and Data Engineering,2020,32(6):1212-1226.
[48] CHEN W,SONG Y,BAI H,et al. Parallel Spectral Clustering in Distributed Systems [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2011,33(3):568-586.
[49] LI Z,WU X,CHANG S. Segmentation Using Superpixels:A Bipartite Graph Partitioning Approach[C]// 2012 IEEE Conference on Computer Vision and Pattern Recognition. 2012:789-796.

GUO Yingya, born in 1990, Ph.D, associate professor, master supervisor, is a member of China Computer Federation. Her main research interests include computer network, traffic engineering and routing optimization.

GENG Haijun, born in 1983, Ph.D, associate professor, master supervisor, is a member of China Computer Federation. His main research interests include computer network and multi-path routing.

(责任编辑:何杨)