计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200900146

基于用户延迟感知的移动边缘服务器放置方法

郭飞雁　唐　兵
湖南科技大学计算机科学与工程学院　湖南 湘潭 411201
(fyguo@mail.hnust.edu.cn)

摘　要　物联网和5G网络的快速发展产生了大量数据,通过将计算任务从移动设备卸载到具有足够计算资源的边缘服务器上,可有效减少网络拥塞和数据传播延迟等问题。边缘服务器放置是任务卸载的核心,高效的边缘服务器放置方法能有效满足移动用户访问低时延、高带宽等需求。为此,文中以最小化访问延迟和最小化负载差异为优化目标,建立边缘服务器放置优化模型;然后,提出了一种基于改进启发式算法的移动边缘服务器放置方法ESPHA(Edge Server Placement Based on Heuristic Algorithm),实现多目标优化。首先将K-means算法与蚁群算法相结合,通过效仿蚁群在觅食过程中共享信息素,将信息素反馈机制引入边缘服务器放置方法中,然后,通过设置禁忌表对蚁群算法进行改进,提高算法的收敛速度;最后,用改进的启发式算法求解模型的最优放置方案。使用上海电信真实数据集进行实验,结果表明提出的ESPHA方法在保证服务质量的前提下取得了低延迟和负载均衡之间的优化平衡,其效果优于现有的其他几种代表性的方法。

关键词:移动边缘计算;边缘服务器放置;启发式算法;访问延迟;负载均衡
中图法分类号　TP311.5

Mobile Edge Server Placement Method Based on User Latency-aware

GUO Fei-yan and TANG Bing
School of Computer Science and Engineering,Hunan University of Science and Technology,Xiangtan,Hunan 411201,China

Abstract　The rapid development of the Internet-of-Things and 5G networks generates a large amount of data. By offloading computing tasks from mobile devices to edge servers with sufficient computing resources,network congestion and data propagation delays can be effectively reduced. The placement of edge server is the core of task offloading,and efficient placement method can effectively satisfy the needs of mobile users to access services with low latency and high bandwidth. To this end,an optimization model of edge server placement is established through minimizing both access delay and load difference as the optimization goal. Then,based on the heuristic algorithm,a mobile edge server placement method called ESPHA (Edge Server Placement Method Based on Heuristic Algorithm) is proposed to achieve multi-objective optimization. Firstly,the K-means algorithm is combined with the ant colony algorithm,the pheromone feedback mechanism is introduced into the placement method by emulating the mechanism of ant colony sharing pheromone in the foraging process,and the ant colony algorithm is improved by setting the taboo table to improve the convergence speed. Finally,the improved heuristic algorithm is used to solve the optimal placement. Experiments using Shanghai Telecom's real datasets show that the proposed method achieves an optimal balance between low latency and load balancing under the premise of guaranteeing quality of service,and outperforms several existing representative methods.

Keywords　Mobile edge computing,Edge server placement,Heuristic algorithm,Access delay,Workload balancing

1　引言

万物互联时代,移动边缘计算(Mobile Edge Computing,MEC)应运而生。MEC的基本思想是将计算从核心网络下沉迁移到网络边缘,减少移动业务交付时延,抑制网络拥塞,MEC在接近移动用户终端的无线接入网内提供服务环境和计算功能。

一个典型的MEC系统如图1所示,其主要组件包括MEC服务器、移动终端设备,以及移动核心网络。MEC服务器通常是由电信运营商部署在基站附近,通常为服务器或者小型的数据中心。MEC服务器通过网关接入到移动核心网络,与无线接入点即基站进行协同合作。在靠近用户移动终端的基站侧部署MEC服务器,可以大大缩短响应用户请求的时延,从而极大地改善用户服务体验。

到稿日期:2020-09-20　返修日期:2020-11-18　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:湖南省教育厅重点项目(18A186);湖南省自然科学基金(2018JJ2135)
This work was supported by the Scientific Research Fund of Hunan Provincial Education Department(18A186) and Natural Science Foundation of Hunan Province(2018JJ2135).
通信作者:唐兵(btang@hnust.edu.cn)

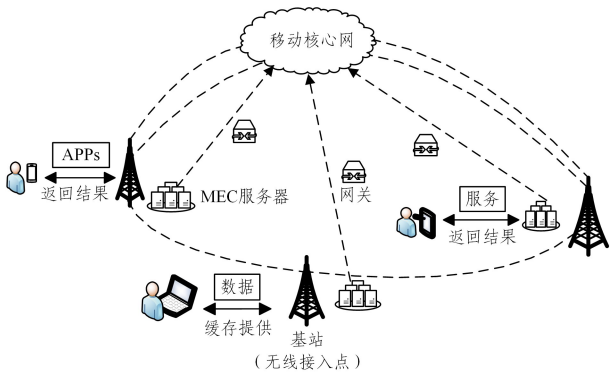


图 1 典型的 MEC 系统

Fig. 1 Typical MEC system

类似于移动云和微云,MEC 服务器在移动边缘环境中的部署也存在一些问题亟待解决[1]。根据具体的网络环境,MEC 服务器可以部署在移动网络的不同物理位置以灵活适应不同的业务需求。将计算任务从核心网迁移到网络边缘服务器,减少核心网拥塞和数据传播延迟是移动边缘计算的主要目标。但对于边缘服务器具体的放置位置并没有明确的方法。边缘服务器的放置就是在满足用户需求和目标的前提下,考虑用户和资源的约束限制,在一定的地理位置范围之内,设计一种特定的方法为边缘服务器选择恰当的地理位置来达到网络延时最小化、访问资源均衡化的最终目的。由于网络技术的快速发展,当前无线城域网覆盖的城市区域人口密度大,边缘服务器适合放置在无线城域网中。通过移动用户的访问,减少边缘服务器的闲置率,提高边缘服务器的成本效用及用户的访问满意度。但将移动边缘服务器放置在无线城域网中存在以下约束问题:

(1)边缘服务器的位置对于移动用户的访问延迟至关重要,尤其是在网络规模大的城市中,移动用户通过基站访问边缘服务器,效率低下的边缘服务器的放置将导致较长的访问延迟以及边缘服务器之间的工作负载严重不平衡。因此,战略边缘服务器的位置将大大改善边缘环境下各种移动应用程序的访问延迟。

(2)边缘服务器的放置位置对边缘服务器的资源利用率会产生较大影响。边缘服务器的放置位置决定了其服务的用户范围,由于无线城域网规模较大,易出现一些边缘服务器负载过量,而另一些边缘服务器使用不足,甚至闲置的问题。边缘服务器不恰当的放置会造成边缘服务器的负载不均衡现象。

目前,已有许多学者及技术人员对移动边缘服务问题展开了研究,但现有研究大多数集中在将移动用户的工作负载卸载到微云端或者远端云,使移动设备实现低时延,并且这种研究假定微云等设备已经被放置[2-5]。很少有研究者关注边缘服务器的放置问题及将移动用户的工作负载卸载到边缘服务器上,来提高移动用户访问性能[6-8]。随着物联网和 5G 网络的快速发展,为了解决低时延、高带宽等需求,移动边缘服务器的放置将是下一步的关键工作。现有的边缘计算相关研究中关于边缘服务器的放置存在以下几个问题:

(1)大多数研究已经假定边缘设备被放置在某个位置,进而对其访问延迟进行研究,并没有考虑负载相关问题;

(2)现有研究中关于移动边缘服务器放置模型方面的研究较少;

(3)现有的研究大多没有真实的移动边缘计算实验数据和环境。

考虑到上述 3 个问题,本文从问题定义、系统建模、模型分析等多个角度出发,提出了一种基于负载优先启发式算法的边缘服务器放置方法,从而满足了移动用户访问低时延、高带宽的需求。

本文第 2 节介绍了边缘云和服务器放置的相关研究;第 3 节介绍了本文的研究动机和边缘服务器放置的相关问题定义;第 4 节提出了边缘服务器的放置方法;第 5 节给出了实验结果并进行了相关讨论;最后总结了当前方法的特点并展望今后的工作。

2 相关工作

边缘计算可以为应用提供实时、高带宽、低延迟的访问,已实现的各种应用包括增强现实、虚拟现实、联网汽车、物联网应用等[9]。边缘计算使运营商可以托管靠近网络边缘的内容和应用程序,还为移动无线网络和有线网络带来了更高的性能和访问权限。在有关 5G 网络基础设施和网络功能虚拟化技术的研究中,经常涉及边缘计算[10-11]。

在移动边缘计算环境中,移动用户可以访问在基站范围内非常接近的边缘服务器。边缘服务器也可以被视为移动用户的卸载目标,目的是减少移动用户与远程云之间的访问延迟。这是通过将计算和存储容量从核心网络导入边缘服务器来实现的。许多研究集中在移动边缘计算的卸载上,例如文献[12-16],但是很少有研究集中在移动边缘计算环境中的边缘服务器放置上。大多数研究的研究依据是假定边缘设备设置在某个位置。

在研究移动边缘服务器的放置之前,一些学者已经研究了几个类似的放置问题。有些学者研究了 Web 服务器副本在内容分发中的位置,他们将 Web 服务器副本的放置描述为 K 中值问题,并建立了副本放置的图论方程[17]。还有部分学者采用了一种新的基于 K-means 聚类的媒体选择算法,以确定多媒体环境中客户端和媒体服务器之间的最佳匹配。他们还使用最新的网络协调技术来获取全球网络信息,并在客户端位置分布的限制下优化了服务延迟性能和部署成本之间的权衡[18]。

在内容分发网络中,副本服务器或缓存服务器是远程服务器的镜像,它可以向客户提供内容交付,同时保持有效且平衡的资源消耗。远程用户从缓存副本服务器获取服务。此模型减少了远程访问的带宽,共享了网络流量,并减少了服务器在原始网站上的负载。但是,移动边缘计算中的边缘服务器功能非常强大,可以为远程用户提供更多的计算资源。因此,不同网络环境中的不同服务器会导致不同的放置问题。

近年来,已经有一些关于边缘云放置[19-23]及边缘服务器放置的研究[24-27]。文献[24]提出了一种面向服务卸载的 ES 放置方法,以优化从传感器到 ES 的数据传输延迟以及 ES 之间的负载平衡。文献[26]将放置边缘服务器问题表述为多目

标约束优化问题并采用混合整数规划以找到最佳解决方案，以达到用户访问延迟最小化和负载均衡化。

尽管上述学者研究的方法是有效的，但他们既没有考虑移动边缘网络中边缘云或边缘服务器的工作负载平衡，也没有考虑移动用户的通信延迟。受此启发，本文在边缘服务器放置期间综合考虑了通信延迟和工作负载均衡这两个因素。

3 研究动机和问题定义

本文在移动边缘环境下，对边缘服务器放置系统模型进行设计，首先对系统模型中涉及到的符号变量进行设置和说明，如表 1 所列。

表 1 相关符号说明表

Table 1 Related symbols and their descriptions

符号	说明
S	边缘服务器集合
s	集合 S 中的边缘服务器
B	基站集合
B	集合 B 中的基站
E	基站和边缘服务器之间的关系
n	基站个数
k	边缘服务器个数
w	基站或边缘服务器的负载
P	基站或边缘服务器的位置
L	基站与边缘服务器之间的距离
A	边缘服务器放置方案集合
a	边缘服务器放置方案
C	基站分配方案
Es_i	分配到边缘服务器 s_i 的基站集合

3.1 研究动机

MEC 服务器可以部署在移动网络的不同物理位置上，包括宏基站处、多制式基站汇聚点或无线网络控制器处。由于基于宏基站的 MEC 服务器部署方式对于降低用户访问时延及缓解移动核心网拥塞效果最为明显，因此，本研究中采用 MEC 服务器与基站共享站址的部署方式进行 MEC 服务器放置。

在移动边缘计算环境下，边缘服务器放置系统模型可以看作一个由移动终端设备、基站及即将放置的边缘服务器组成的无向网络 $G=(B\cup S,E)$，B 代表基站集合，S 代表边缘服务器集合，E 代表基站和边缘服务器之间的关系。MEC 服务器与基站共享站址，每一个基站都会被分配到某个边缘服务器管辖范围内。假设边缘服务器的个数为 K，则在 B 中需要寻找到 K 个边缘服务器的地址，并将所有的基站分配到这 K 个边缘服务器中，每一个边缘服务器负责它所管辖区域的所有基站转发出来的用户网络请求。由于每一个边缘服务器的计算资源容量有限且数据传输延迟与距离有关，因此，边缘服务器的位置及边缘服务器与其所管辖的基站距离非常重要。

基站可以通过链路直接访问其所属的边缘服务器，可以将经由其转发的任务卸载到其所属的边缘服务器上。但每个边缘服务器的计算资源容量有限，若使得每个边缘服务器的工作负载尽可能均衡，则可以平衡边缘服务器之间的工作负载，延长边缘服务器的使用寿命。因此，需要对基站卸载任务到边缘服务器设计最优的分配方法，来达到负载均衡。

为了更直观地描述边缘服务器放置系统模型，我们引入了一个例子，如图 2 所示。本文要解决的问题是如何从图中的 b_1 至 b_{12} 的 12 个基站中选择出放置 3 个边缘服务器 s_1，s_2 和 s_3 的位置，并将 12 个基站分别分配到边缘服务器 s_1，s_2 和 s_3 管辖范围内，每个基站通过访问这 3 个边缘服务器获取网络服务。以 s_1 边缘服务器为例，s_1 负责处理 b_1，b_2，b_3 和 b_4 4 个基站所转发的用户请求。模型中每一个边缘服务器的放置方案都尽量满足负载均衡化、延迟最小化两个目标。由于基站和基站间，以及基站和移动边缘设备之间的关系不是本文的研究重点，因此本文不做详细讨论。下面对本文中考虑的边缘服务器放置问题进行定义。

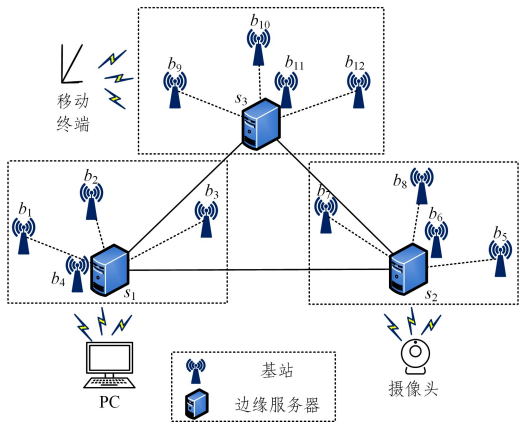


图 2 边缘服务器放置系统模型

Fig. 2 Edge server placement system model

3.2 问题定义

在移动边缘服务器放置方法中相关问题定义如下：假设在无向网络 G 中，$S=\{s_1,s_2,s_3,s_4,\cdots,s_k\}$ 为一组边缘服务器，每一个边缘服务器是同构的，计算和存储能力有限且一样；$B=\{b_1,b_2,b_3,b_4,\cdots,b_n\}$ 为所有基站集合，基站负责转发其覆盖范围内移动网络用户的所有服务请求。当边缘服务器放置完成后，每一个边缘服务器负责处理由其覆盖范围内所有基站所转发的移动网络用户服务请求，每一个边缘服务器的负载等于其覆盖范围内所有基站的负载之和。$d(P_b,P_s)$ 表示每一个基站到其所分配的边缘服务器间的距离，边缘服务器处理所有移动用户请求的访问延迟就等于其覆盖范围内所有基站到该边缘服务器的距离之和。

因此，我们需要解决的问题主要有两个：1）K 个边缘服务器的放置位置；2）N 个基站的分配方法。只要方案能够满足边缘服务器负载均衡化及各个基站上移动用户访问延迟最小化，则该方案是我们所寻找的最优放置方案。

根据前面对问题定义的描述，首先给出两个优化模型：延迟最小化模型和负载均衡化模型。设 w 表示基站处理数据量大小即负载，则第 j 个基站的负载大小为 w_{bj}，第 i 个边缘服务器上的负载大小为其管辖范围内基站负载大小之和，$w_{si}=w_{b1}+w_{b2}+\cdots+w_{bi}$。

延迟最小化模型主要考虑在移动边缘计算环境中，基站和其所分配的边缘服务器间的访问距离最小化，即：

$$L(a_i)=\min_{s\in S}\max_{b\in E_s} d(p_b,p_s) \tag{1}$$

其中，a_i 代表某一边缘服务器放置方案，$L(a_i)$ 代表该方案下各基站用户访问延迟，p_b 代表分配到某一边缘服务器中所有

基站集 E_s 中的某一个基站地址，p_s 代表边缘服务器 s 的地址。

负载均衡化模型主要考虑边缘服务器之间负载的差距最小化，为量化边缘服务器的负载均衡度，定义负载均衡指标的计算公式如下：

$$W(a_i)=\underset{s_i\in S}{\text{Min}}\ \underset{s_j\in S}{\text{Max}}\ (w_{s_i},w_{s_j}) \tag{2}$$

其中，$W(a_i)$代表该边缘服务器放置方案中负载均衡量大小，w_{s_i} 和 w_{s_j} 分别代表边缘服务器 s_i 和 s_j 间的负载大小，其差值越小，边缘服务器间的负载越均衡。结合负载均衡化和延迟最小化模型，建立如下加权目标优化函数公式：

$$PW(a_i)=\lambda L_{nor}(a_i)+(1-\lambda)W_{nor}(a_i) \tag{3}$$

其中，λ 为加权系数，λ 值位于[0,1]区间；$L_{nor}(a_i)$和 $W_{nor}(a_i)$则是 $L(a_i)$和 $W(a_i)$归一化处理后的值，为了便于比较，我们采用 min-max 标准化方法将数据统一映射到[0,1]区间上，如式(4)和式(5)所示：

$$L_{nor}(a_i)=\frac{L(a_i)-\min\limits_{1\leqslant j\leqslant n}\{L(a_j)\}}{\max\limits_{1\leqslant j\leqslant n}\{L(a_j)\}-\min\limits_{1\leqslant j\leqslant n}\{L(a_j)\}} \tag{4}$$

$$W_{nor}(a_i)=\frac{W(a_i)-\min\limits_{1\leqslant j\leqslant n}\{W(a_j)\}}{\max\limits_{1\leqslant j\leqslant n}\{W(a_j)\}-\min\limits_{1\leqslant j\leqslant n}\{W(a_j)\}} \tag{5}$$

在寻求边缘服务器放置方法的目标最优解时，需考虑以下两个约束条件：

(1)每一个基站都会被分配到一个边缘服务器上，而且只能分配到一个边缘服务器上，任意两个边缘服务器之间没有基站交叉，所有边缘服务器所覆盖的基站和就等于总的基站数。约束条件的形式化表示如下：

$$E_{s_i}\cap E_{s_j}=\varnothing \tag{6}$$

$$B=\sum_{i\leqslant i\leqslant k}E_{s_i} \tag{7}$$

其中，E_{s_i}为分配到 s_i 边缘服务器下的基站集，k 为边缘服务器数量，B 为所有基站的集合。

(2)各边缘服务器的访问负载等于各边缘服务器所覆盖基站的负载之和。约束条件的形式化表示如下：

$$\sum_{s_i\in S}W_{s_i}=\sum_{b_i\in B}W_{b_i} \tag{8}$$

其中，w_{s_i} 表示边缘服务器 s_i 的负载大小，w_{b_i} 表示基站 b_i 的负载大小。

4 优化方法

由于用户具有移动性，访问基站会发生变化，因此各个基站的负载大小也呈现一定的动态性。移动边缘服务器的放置也是一个动态过程，基站负载一旦发生变化，移动边缘服务器的放置位置及基站分配结果也会随之变化。本文采用一种负载优先的启发式算法(ESPHA)来解决边缘服务器的放置问题。

4.1 ESPHA 方法

ESPHA 算法基于 K-means 算法和启发式算法中的蚁群算法来解决边缘服务器的放置问题。

ESPHA 方法采用典型的基于距离的非层次聚类算法完成边缘服务器的位置选择过程，在最小化误差函数的基础上将数据划分为预定的类数 K(K 即边缘服务器的个数)，同时采用距离作为相似度评价指标，距离越近，相似度越大，相似度大的则归属到同一类中。在实践中，为了得到较好的结果，每次运行聚类算法时，在基站密集区选择不同的初始聚类中心，多次运行聚类算法。在所有对象分配完成后，重新计算 K 个聚类的中心，聚类中心取该簇的均值。当质心不发生变化时停止并输出聚类结果。边缘服务器放置方法将质心所在的基站位置设置为边缘服务器的位置。由于本文所采用的聚类方法是根据距离进行相似性聚类，因此满足边缘服务器放置问题中各基站访问延迟最小化的要求。

确定边缘服务器位置后，需要将各个基站分配到 K 个边缘服务器中，尽量满足各个边缘服务器负载均衡化。在基站分配过程中，我们引入蚁群算法的信息素矩阵 $\boldsymbol{T}_t=(\tau_{ij})$及禁忌表。信息素矩阵 $\boldsymbol{T}_t$ 用于保留蚂蚁在遍历各个边缘服务器过程中获得的历史经验，让所有基站可以共享在搜索边缘服务器过程中积累的边缘服务器信息，从而找到距离最近且负载未超过平均负载量的边缘服务器；禁忌表用来记录蚂蚁已访问过且负载已超过平均负载量的边缘服务器，其最初为空，在遍历的过程中逐步添加，直到所有边缘服务器添加完毕为止，禁忌表所存放的是负载量已达平均负载量的边缘服务器，表中所有的边缘服务器不再参与基站分配。

信息素矩阵 $\boldsymbol{T}_t$ 中元素 τ_{ij} 为基站 b_i 放置在边缘服务器 s_j 上积累的信息素，为该边缘服务器上服务的基站负载大小。在每一次遍历中，信息素矩阵的更新主要依据边缘服务器当前所承担的负载是否已达边缘服务器负载上限值，如果已达上限值，则 τ_{ij} 为 0，且将该边缘服务器添加到禁忌表；如果未达到上限值且属于离基站最近的边缘服务器，则将 τ_{ij} 设置为该基站的负载值，禁忌表不变。在下一个基站分配过程中，通过浏览信息素及禁忌表来选择即将分配到的边缘服务器，达到访问延迟最小化、负载均衡化的最优选择。

4.2 算法推导

ESPHA 算法主要用于解决边缘服务器的位置选择及基站分配两个问题。

$$B=\{b_1,b_2,b_3,\cdots,b_n\} \tag{9}$$

$$d(b,b_i)=2R\arctan\left(\frac{\sqrt{hav(\theta)}}{\sqrt{1-hav(\theta)}}\right) \tag{10}$$

$$hav(\theta)=\sin 2\left(\frac{\sigma_1-\sigma_2}{2}\right)+\cos\sigma_1\cos\sigma_2\sin 2\left(\frac{\alpha_1-\alpha_2}{2}\right) \tag{11}$$

在边缘服务器位置选择过程中，首先计算基站间的距离。本文使用地理距离算法计算两个基站间的距离 $d(b_i,b_j)$。式(9)中，B 为基站集合，n 表示基站顺序。R 为地球半径，式(11)中我们将 R 假设为 6 371 km。θ 为两个基站地理位置间圆心角。(α_1,σ_1)和(α_2,σ_2)为两个基站间纬度、经度坐标。接着根据距离对所有基站进行相似度聚类，任取 K 个基站位置作为聚类质心，以最小距离为迭代标准进行迭代，直到质心不变，得出 k 个类簇，即 k 个 A_i。式(12)中，A_i 表示第 i 个聚类中的基站集合，式(13)中 b_j 表示基站集合 B 中第 i 个聚类中的各个基站。

$$A_i=\{\cdots b_j\cdots\} \tag{12}$$

$$(1\leqslant i\leqslant k)\bigcup(b_j\in B)\bigcup\text{Min}(d(b,b_i)) \tag{13}$$

ESPHA 算法采用的聚类方法是将边缘服务器设置于一个类簇的中心，类簇中心基站的位置 p_{b_i} 则是边缘服务器的位置 p_{s_j}。例如，选定 b_i 基站为 s_j 边缘服务器地址，则 b_i 基站地

址就是 s_j 边缘服务器地址，如式(14)所示：

$$p_{s_j}=p_{b_i} \tag{14}$$

通过 ESPHA 算法得出的边缘服务器位置可以满足延迟最小化要求，得出最优的边缘服务器放置方案。

在边缘服务器最优放置方案 a 中得出 K 个边缘服务器位置，随后进行基站分配。首先计算边缘服务器的平均负载，如式(15)所示：

$$\bar{w}_s=\frac{\sum\limits_{b_i\in B} w_{b_i}}{K} \tag{15}$$

其中，w_{b_i} 为各基站负载量，b_i 为基站集合 B 中的某个基站，K 为边缘服务器数量，$\bar{w}_s$ 为边缘服务器的平均负载量。基站分配过程遵守以下两个原则：1)基站分配遵循负载优先分配原则，所有基站在分配前按负载大小进行降序排列，负载大的基站优先分配，以减少各个边缘服务器负载不均衡的现象；2)若某一个边缘服务器的负载大小 w_{s_j} 超过平均负载量 $\bar{w}_s$，则不再进行基站分配。

在基站分配过程中，引入蚁群算法中的信息素矩阵 $\boldsymbol{T}_t$ 及禁忌表 F。信息素矩阵 $\boldsymbol{T}_t$ 的定义如下：

$$\boldsymbol{T}_t=\begin{bmatrix}\tau_{11} & \cdots & \cdots & \cdots & \tau_{1k}\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ \cdots & \cdots & \tau_{ij} & \cdots & \cdots\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ \tau_{n1} & \cdots & \cdots & \cdots & \tau_{nk}\end{bmatrix} \tag{16}$$

其中，τ_{ij} 记录基站 b_i 与边缘服务器 s_j 间的分配关系，n 为全部基站数量，k 为全部边缘服务器数量。τ_{ij} 值的定义如下：

$$\tau_{ij}=\begin{cases}0, & b_i\notin E_{s_j}\\ w_{b_i}, & b_i\in s_j\end{cases} \tag{17}$$

当基站 b_i 不分配在 s_j 上时，τ_{ij} 值为 0；当基站 b_i 分配在 s_j 上时，τ_{ij} 值为基站 b_i 的负载大小。基站分配结束后，信息素矩阵第 j 列上的所有元素之和则为边缘服务器 s_j 的负载大小，如式(18)所示：

$$w_{s_j}=\sum\tau_{ij},i\in\{1,2,3,\cdots,n\} \tag{18}$$

禁忌表 F 为负载已满的边缘服务器列表，当基站进行分配时，在禁忌表以外可分配的边缘服务器列表中寻找最近的边缘服务器进行分配。进入禁忌表 F 的边缘服务器需满足如下条件：

$$w_{s_j}\geqslant\bar{w}_s \tag{19}$$

基站分配过程中，采用负载优先分配原则，负载大的基站 b_i 优先分配，即当 $\sum\limits_{i=1}^{n}\tau_{ij}+w_{b_i}<\bar{w}_s$ 时，$\tau_{ij}=w_{b_i}$ 且 $b_i\in E_{s_j}$；当 $\sum\limits_{i=1}^{n}\tau_{ij}+w_{b_i}\geqslant\bar{w}_s$ 时，$\tau_{ij}=0$。

基站 b_i 在禁忌表外的边缘服务器列表中进行遍历时，找到最近的边缘服务器进行判断：如果信息素矩阵中当前边缘服务器 s_j 的信息素之和加上当前基站负载后超过平均负载，则不进行分配，继续寻找下一个最近的边缘服务器；如果没有超过平均负载，则将该基站分配到当前边缘服务器上，更新信息素表，并将基站 b_i 的分配方案记录到边缘服务器方案 E_s 中。每完成一个基站分配，都需要检测信息素表中是否有超过负载均衡值的边缘服务器，如果有，则将该边缘服务器加入禁忌表，不再对其遍历。禁忌表更新规则如下：当 $\sum\limits_{i=1}^{n}\tau_{ij}>\bar{w}_s$ 时，$s_j\in F$；当 $\sum\limits_{i=0}^{N}\tau_{ij}<\bar{w}_s$ 时，F 不变。

完成基站分配后，输出分配方案 $C(b_i,s_j)(b_i\in B\cup s_j\in S)$，以及该方案下的负载标准差 w 和平均延迟 $\bar{d}$，公式如下：

$$w=\sqrt{\frac{(w_{s_j}-\bar{w})^2}{k}} \tag{20}$$

$$\bar{d}=\frac{\sum\limits_{s_j\in S}\sum\limits_{b_i\in B} l(p_{b_i},p_{s_j})}{k} \tag{21}$$

4.3 算法流程

第 1 步　初始化相关数据。基站相关数据、聚类个数 K(边缘服务器数量)、聚类最大循环次数 n、信息素矩阵 $\boldsymbol{T}_t$、禁忌表 F、边缘服务器平均负载 w_s。

第 2 步　采用 ESPHA 算法根据距离对基站进行相似度聚类，选择位于每一个聚类质心位置的基站，作为 K 个边缘服务器的位置。

第 3 步　基于负载优先分配原则，将所有基站按负载从大到小进行排序，负载大的基站优先分配。基站 b_i 根据信息素矩阵对禁忌表以外的边缘服务器进行搜索。如果搜索到最近且加入当前基站负载没有超过平均负载 w_s 的边缘服务器，则将基站 b_i 分配在该边缘服务器上，并更新信息素矩阵；如果超过 w_s，则不分配，并寻找下一个最近的边缘服务器进行分配，将分配结果记录到基站分配方案 $C(s_j,b_i)$ 中。

第 4 步　每一个基站完成分配后，检查信息素矩阵 $\boldsymbol{T}_t$，判断该边缘服务器负载是否超过平均负载 w_s，如果超过，则将其添加到禁忌表中，其不参与下一次基站分配。

第 5 步　基站循环数量加 1，下一个基站继续在禁忌表外的边缘服务器进行分配。

第 6 步　基站分配完成后，输出最优分配方案 $C(s_j,b_i)$，及该方案下的负载标准差和平均延迟。

5 实验与分析

5.1 数据集说明

本实验采用上海电信的一个真实基站数据集，其中包括 6 262 位移动用户在 2 768 个基站上访问的 562 914 条记录，访问记录中详细记载了每一个用户每一次访问基站的开始时间和结束时间及基站的经纬度。通过 Google 地图工具可得图 3 所示的上海市基站分布图。

图 3　上海市基站分布图

Fig. 3　Shanghai base station distribution map

通过放大地图可以发现，基站位置分布并不均衡，如图 4 和图 5 所示，其中各基站所处位置的红色数字代表基站访问

负载大小。我们对6262位移动用户在2768个基站上15天的访问负载数据集进行了初步统计汇总，并随机选择10位用户的数据进行展示，如表2所列，从中可以发现基站负载严重不均衡。

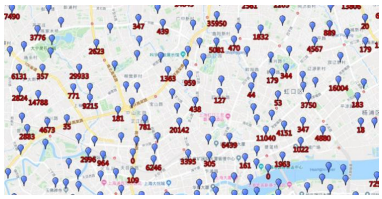


图4 上海市基站分布密集区

Fig. 4 Densely distributed area of base stations in Shanghai

图5 上海市基站分布稀疏区

Fig. 5 Sparsely distributed area of in base stations Shanghai area

表2 上海各个基站访问负载量

Table 2 Workload of each base station in Shanghai

基站 ID	用户 ID	负载/min
12	354	24958
23	147	10655
345	28	824
400	1242	61972
543	311	12566
1126	67	1331
2227	1	89
3428	3	78
4139	440	9639
6023	261	14026

5.2 实验设置

本文对如下3个实验场景进行了实验对比。

(1)将基站数量从300增加到3000，以300为步长进行变化，以$R=0.1$的边缘服务器放置比例(例如300个基站配置30个边缘服务器)，采用不同方法测试各个边缘服务器的负载标准差及平均延迟。

(2)将基站数量设置为3000，边缘服务器数量从100变化到500，以100为步长进行变化，即测试不同边缘服务器的放置比例下采用不同方法的负载标准差及平均延迟的变化。

(3)采用ESPHA方法测试不同边缘服务器放置比例对负载标准差及平均延迟的影响。

5.3 实验对比方法

本节主要将本文的ESPHA方法与Top-K方法、Random方法、K-means方法这3种常用方法进行实验对比。

(1)Top-K方法。取负载量最大的前K个基站位置作为边缘服务器位置，在基站分配过程中，以最近边缘服务器原则进行分配。

(2)Random方法。随机选取K个基站地址设置为边缘服务器地址，每一个基站选择离自己最近的边缘服务器进行分配。

(3)K-means方法。将基站按K-means方法聚成K类，取其质心作为边缘服务器地址，每一个基站选择离自己最近的边缘服务器进行分配。

5.4 基站数量变化时的对比结果

将边缘服务器与基站放置比例R设为0.1，基站数量从300变化到3000时的实验结果如图6所示。可以看出，K-means方法和ESPHA方法的基站访问平均距离是最小的，由于基站数的增加，边缘服务器放置比例不变，边缘服务器数量增加，基站可选的服务器范围增加，因此，基站访问平均距离都呈现下降趋势。其中，K-means方法是距离优先的边缘服务器放置方法，在基站访问平均距离上可以得到最优解，而ESPHA方法、Top-K方法、Random方法在总平均距离上分别超过最优解的百分比为4%，205%，251%，ESPHA方法接近最优解。

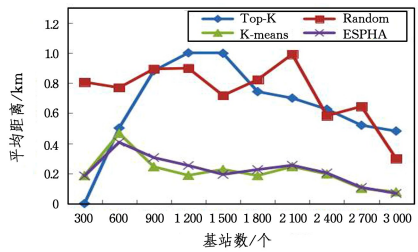


图6 $R=0.1$时不同放置方法下的基站访问平均距离

Fig. 6 Base station access average distance with different placement niethods when $R=0.1$

当基站数逐渐变大，边缘服务器个数逐渐增多时，各边缘服务器访问的负载标准差逐渐减少。由于Top-K是采用负载量最大的前K位基站作为边缘服务器地址，因此负载标准差最小，ESPHA方法、Random方法、K-means方法分别超过Top-K方法平均负载标准差的百分比为54.2%，57.4%，68.43%，如图7所示。

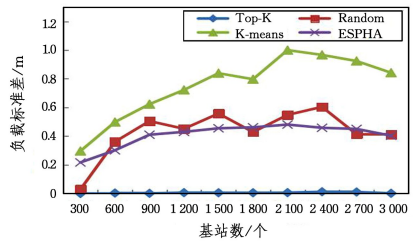


图7 $R=0.1$时不同放置方法下的负载标准差

Fig. 7 Load standard deviation with different placement method when $R=0.1$

综合访问平均距离及负载标准差两个属性，取每个属性所占权重比例为0.5，得到各方法综合性能结果如图8所示，其中ESPHA方法在基站数大于800时，基站访问平均距离及负载标准差综合值为最小，即效果最好。依据式(3)，取$\lambda=0.5$，4种放置方法的$pw(a)$值如表3所列。

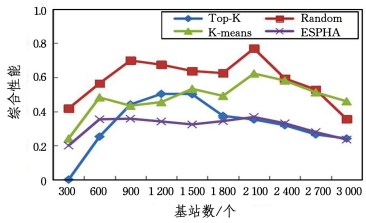


图 8 $R=0.1$ 时不同放置方法下的综合性能值变化情况

Fig. 8 Comprehensive performance values with different placement methods when $R=0.1$

表 3 $R=0.1$ 时不同放置方法下的综合性能值对比

Table 3 Comparison of comprehensive performance values with different placement methods when $R=0.1$

N	Top-K	Random	K-means	ESPHA
300	0	0.4178	0.2408	0.1999
600	0.2522	0.4101	0.4824	0.3540
900	0.4409	0.4824	0.4337	0.3575
1200	0.5030	0.4890	0.4549	0.34120
1500	0.5009	0.3897	0.5326	0.3239
1800	0.3739	0.4396	0.4913	0.3435
2100	0.3529	0.5220	0.6225	0.3674
2400	0.3191	0.3149	0.5813	0.3307
2700	0.2649	0.3430	0.5121	0.2788
3000	0.2413	0.1694	0.4598	0.2350

5.5 边缘服务器数量变化时的对比结果

边缘服务器的数量从 100 变化到 500 的实验结果如图 9 和图 10 所示。

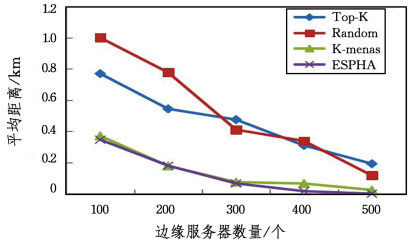


图 9 K 为不同值时不同放置方法下的基站访问平均距离

Fig. 9 Base station access average distance with different placement methods and different K values

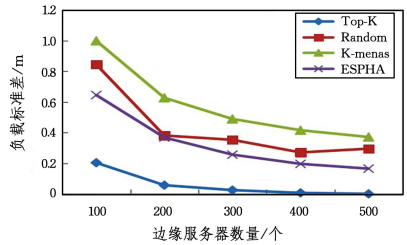


图 10 K 为不同值时不同放置方法下的边缘服务器访问负载标准差

Fig. 10 Workload standard deviation with different placement methods and different K values

可以看出，基站访问平均距离和边缘服务器访问负载标准差均随着边缘服务器数量的增加而逐步减小。在图 9 中，ESPHA 方法效果最好。在图 10 中，Top-K 方法负载标准差最小，依次为 ESPHA 方法、Random 方法和 K-means 方法。依据式(3)，4 种放置方法的 $pw(a)$ 值即综合性能参数如图 11 所示。

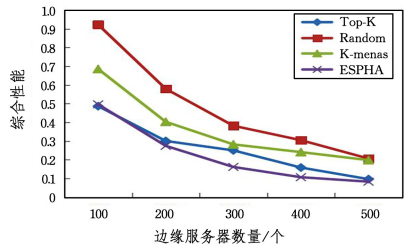


图 11 K 为不同值时不同放置方法下的综合性能值变化情况

Fig. 11 Comprehensive performance value with different placement methods changes with K values

从图 11 可以看出，4 种方法中边缘服务器放置方案最好的依次为 ESPHA 方法、Top-K 方法、K-means 方法和 Random 方法。K 取不同值时 4 种方法的 $pw(a)$ 值如表 4 所列。

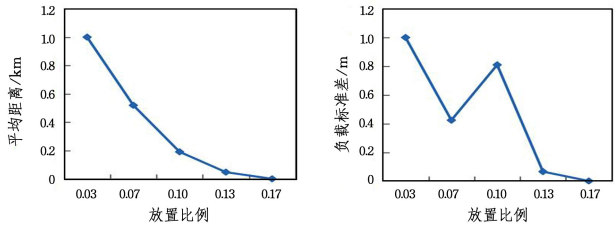
表 4 K 取不同值时不同放置方法下的综合性能值对比

Table 4 Comparison of comprehensive performance values with different placement methods and different K values

K	Top-K	random	K-means	ESPHA
100	0.4864	0.5778	0.6853	0.4960
200	0.3002	0.4239	0.4028	0.2740
300	0.2497	0.2264	0.2819	0.1612
400	0.1579	0.1837	0.2406	0.1062
500	0.0959	0.0681	0.1977	0.0823

5.6 边缘服务器放置比例对结果的影响

当基站数为 3000 时，设置边缘服务器放置比 R 为 0.03，0.07，0.10，0.13，0.17，采用 ESPHA 方法得出各基站访问延迟及负载标准差如图 12 所示。在不考虑经济成本时，边缘服务器放置比例越大，即边缘服务器数量越多，布置越密集，各基站访问边缘服务器的平均延迟越小，各边缘服务器的负载标准差也逐渐变小。



(a)当 $n=3000$ 时，R 为不同值时 ESPHA 放置方法的基站访问平均距离

(b)当 $n=3000$ 时，R 为不同值时 ESPHA 放置方法的负载标准差

图 12 当基站数 $n=3000$ 时，边缘服务器放置比例 R 为不同值时 ESPHA 方法的性能值比较

Fig. 12 Comparison of performance values with ESPHA placement method when R changes and $n=3000$

结束语 移动边缘计算是将云端业务下沉到移动边缘网络的一项重要技术。通过边缘服务器的放置来提高移动用户的访问质量，因此，本文研究了边缘服务器的放置问题。本文在移动边缘计算中，把边缘服务器的放置问题作为一个多目标优化问题，主要采用一种改进的启发式算法得出边缘服务器放置方案。我们共设计了 3 个基于上海电信实际基站数据集的对比实验，以综合评估不同方法下边缘服务器工作负载均衡及通信延迟方面的性能，实验结果表明本文方法在一定程度上优于其他方法。

虽然本文方法的整体性能是可以接受的，但其在负载平衡方面对比其他方法没有达到最优。由于负载平衡和通信延迟是两个相互制约的因素，因此，边缘服务器的放置问题是一个较为复杂的问题。如何设计一种既能保证工作负载平衡最优又能保证通信延迟最小化的方法是我们未来研究工作的重点。

参 考 文 献

[1] ZHAO Z, LIU F, CAI Z, et al. Edge Computing: Platforms, Applications and Challenges[J]. Journal of Computer Research and Development, 2018, 55(2): 327-337.

[2] ZENG J, ZHANG J, LIN B, et al. Micro cloud load balancing algorithm based on wireless metropolitan area network[J]. Computer Science, 2019, 46(8): 163-170.

[3] XIA Q, LIANG W, XU W. Throughput maximization for online request admissions in mobile cloudlets[C]// IEEE Conference on Local Computer Networks. IEEE, 2014.

[4] VERBELEN T, SIMOENS P, TURCK F D, et al. Cloudlets: bringing the cloud to the mobile user[C]// ACM Workshop on Mobile Cloud Computing & Services. 2012.

[5] CHUN B, IHM S, MANIATIS P, et al. Clonecloud: elastic execution between mobile device and cloud[C]// The Sixth Conference on Computer Systems. 2011.

[6] XU Z, LIANG W, XU W. Capacitated cloudlet placements in wireless metropolitan area networks[C]// IEEE 40th Conference on Local Computer Networks. 2015.

[7] XU Z, LIANG W, XU W, et al. Efficient Algorithms for Capacitated Cloudlet Placements[J]. IEEE Transactions on Parallel & Distributed Systems, 2016, 27(10): 2866-2880.

[8] ZHANG J, LIN B, LU Y, et al. Cloudlet Placement and User Task Scheduling Based on Wireless Metropolitan Area Networks[J]. Computer Science, 2019, 46 (6): 128-134.

[9] SHI W, CAO J, ZHANG Q, et al. Edge Computing: Vision and Challenges[J]. Internet of Things Journal, IEEE, 2016, 3(5): 637-646.

[10] SARRIGIANNIS I, KARTSAKLI E, RAMANTAS K, et al. Application and Network VNF migration in a MEC-enabled 5G Architecture[C]// IEEE CAMAD. IEEE, 2018.

[11] HSIEH H C, CHEN J L, BENSLIMANE A. 5G Virtualized Multi-access Edge Computing Platform for IoT Applications [J]. Journal of Network and Computer Applications, 2018, 115(8): 94-102.

[12] WANG C, LIANG C, YU F R, et al. Computation Offloading and Resource Allocation in Wireless Cellular Networks With Mobile Edge Computing[J]. IEEE Transactions on Wireless Communications, 2017, 16(8): 4924-4938.

[13] DAI Y, XU D, MAHARJAN S, et al. Joint Computation Offloading and User Association in Multi-Task Mobile Edge Computing[J]. IEEE Trans. Vehicular Technology, 2018, 67(12): 12313-12325.

[14] DINH T Q, LA Q D, QUEK T Q S, et al. Learning for Computation Offloading in Mobile Edge Computing[J]. IEEE Transactions on Communications, 2018, 66(12): 6353-6367.

[15] XIAO M, CHUANG L, HAN Z, et al. Energy-Aware Computation Offloading of IoT Sensors in Cloudlet-Based Mobile Edge Computing[J]. Sensors, 2018, 18(6): 1945.

[16] 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.

[17] QIU L, PADMANABHAN V N, VOELKER G M. On the Placement of Web Server Replicas[C]// IEEE INFOCOM 2001. Anchorage, Alaska, USA, 2001: 1587-1596.

[18] YIN H, ZHANG X, ZHAN T, et al. NetClust: A Framework for Scalable and Pareto-Optimal Media Server Placement[J]. IEEE Transactions on Multimedia, 2013, 15(8): 2114-2124.

[19] 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.

[20] XU Z, LIANG W, XU W, et al. Efficient Algorithms for Capacitated Cloudlet Placements[J]. IEEE Transactions on Parallel & Distributed Systems, 2016, 27(10): 2866-2880.

[21] ZHAO J, OU S, HU L, et al. A heuristic placement selection approach of partitions of mobile applications in mobile cloud computing model based on community collaboration[J]. Cluster Computing, 2017, 20(4): 3131-3146.

[22] LIANG T, LI Y. A Location-Aware Service Deployment Algorithm Based on K-Means for Cloudlets[J]. Mobile Information Systems, 2017, 8342859: 1-8342859: 10.

[23] YAO H, BAI C, XIONG M, et al. Heterogeneous cloudlet deployment and user-cloudlet association toward cost effective fog computing[J]. Concurrency and Computation: Practice and Experience, 2017, 29(17): e3975.

[24] ZHANG J, LI X, ZHANG X, et al. Service offloading oriented edge server placement in smart farming[J/OL]. Software Practice and Experience. https://doi.org/10.1002/spe.2847.

[25] XU X, XUE Y, QI L, et al. Load-aware Edge Server Placement for Mobile Edge Computing in 5G networks[C]// The 17th International Conference on Service-oriented Computing. 2019.

[26] WANG S, ZHAO Y, XU J, et al. Edge server placement in mobile edge computing[J]. Journal of Parallel and Distributed Computing, 2018, 127(MAY): 160-168.

[27] REN Y, ZENG F, LI W, et al. A Low-Cost Edge Server Placement Strategy in Wireless Metropolitan Area Networks[C]// 2018 27th International Conference on Computer Communication and Networks (ICCCN). 2018.

GUO Fei-yan, born in 1982, Ph. D student. Her main research interests include service computing and edge computing.

TANG Bing, born in 1982, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include parallel and distributed computing, cloud computing, etc.