

文章编号: 2095-2163(2020)02-0142-04

中图分类号: TP393.09

文献标志码: A

# 选择目标微云以最大程度地减少服务延迟

任子斌

(东华大学 计算机科学与技术学院, 上海 201620)

**摘要:** 移动边缘计算正在成为一种流行的计算范式,将云上的服务执行扩展到网络边缘。移动边缘计算可为资源受限的移动设备提供对网络边缘资源的低延迟访问。这些设备的移动性给目标微云分配带来了巨大挑战。为了向移动用户提供低延迟的服务,本文考虑了用户的移动性,研究了边缘计算中的目的地微云选择问题。针对该问题,本文提出了最低本地等待时间算法,该算法在目标小云选择过程中考虑了传输时间、计算时间、选择时间和用户位置。仿真结果表明,与现有算法相比,该算法为移动用户提供了最低的服务等待时间。

**关键词:** 目标微云选择; 服务延迟; 移动用户; 微云; 移动边缘计算

## Destination cloudlet selection to minimize service latency

REN Zibin

(School of Computer Science and Technology, Donghua University, Shanghai 201620, China)

**[Abstract]** Mobile edge computing is emerging as a popular computing paradigm, which extends the service execution on cloud to the edge of network. Mobile edge computing provides low latency access to resources at the edge of the network for resource-constrained mobile devices. The mobility of these devices brings great challenges in destination cloudlet allocation. In order to provide low latency service for mobile users, this paper studies the destination cloudlet selection problem in edge computing considering the users' mobility. This paper proposes lowest local latency algorithm for this problem, which considers transmission time, computing time, selection time and user's position in the process of destination cloudlet selection. Simulation results demonstrate that the proposed algorithm provides the lowest service latency for mobile users compared with existing algorithms.

**[Key words]** destination cloudlet selection; service latency; mobile user; cloudlet; mobile edge computing

## 0 引言

近年来,移动设备在人们的日常生活中无处不在<sup>[1]</sup>。这些移动设备受资源限制,并且通常使用云资源来满足其计算要求。许多移动应用程序对延迟敏感,就像增强现实和交互式游戏<sup>[2]</sup>一样,需要低服务延迟。然而,这些应用程序的延迟要求无法通过传统云的架构<sup>[3]</sup>来得到满足。这个问题引起了研究者和用户更多的关注与重视。

在文献[4-5]中提出了移动边缘计算,以满足移动用户这些需要超低延迟的应用程序。为了便于托管延迟敏感应用程序,边缘计算节点被认为实现了快速资源响应的优先级<sup>[6-7]</sup>。边缘计算旨在将云功能推送到网络边缘,此外还提供低延迟<sup>[8]</sup>。

在智能城市中,具有任意移动性的用户设备给选择边缘计算的目标微云带来了新的挑战<sup>[9-11]</sup>。为了提供低延迟服务,在考虑用户位置时尽可能选择为用户提供最低延迟的微云是一个颇具挑战性的难题<sup>[12]</sup>。为了解决这个问题,本文根据用户的位置提出了一种目标微云选择方案、即最低局部延迟算法。

该算法考虑了传输时间、计算时间、选择时间和用户在目标微云选择过程中的位置。对不同网络状况进行仿真,以评估本文提出的算法的性能。将本文提出的算法与现有算法进行了比较。仿真结果表明,该算法为移动用户提供了更低的服务延迟。

## 1 资源分配技术研究现状

随着用户的移动,现有的一些算法诸如:设备与AP接入点之间的最短距离算法、设备与微云之间的最短距离算法、最低延迟算法,虽然能够在一定程度上满足用户延迟需求,但总体用户的满意度仍不理想,因此,本文提出了局部最低延迟算法。

## 2 面向移动用户的最优微云选择机制与算法设计

在本节中,微云和AP接入点在网络中均匀分布。每个AP接入点都覆盖一个半径为 $R$ 的区域。位于AP接入点的覆盖区域圆圈中的用户可以直接与AP接入点通信。AP接入点通过相邻的AP接入点与不在其覆盖区域的微云进行通信。网络中有一个微云控制中心,记录着网络的一些重要信息,例如网络中的微云数量、每个微云的可用资源、每个用户

作者简介: 任子斌(1991-),男,硕士研究生,主要研究方向:边缘计算。

收稿日期: 2019-11-15

的位置以及每个 AP 接入点和微云的位置。所有微云和用户都可以通过 AP 接入点与微云控制中心通信,以更新或接收信息。每个微云可以实时监控其剩余可用的资源,并将这些信息发送到微云控制中心。每个用户都通过 GPS 确定其位置,并将位置信息发送到微云控制中心。

当移动用户生成需要微云服务的任务时,移动用户将连接到 AP 接入点并通过 AP 接入点将任务传输到微云。由于网络中的用户是移动的,微云控制中心应根据用户的位置为用户选择合适的 AP 接入点和服务微云,如此即可为用户提供低延迟的服务。根据用户位置为用户分配合适的 AP 接入点和服务微云从而最小化总服务延迟是本节的主要研究目标。

针对于目前三种算法,本文提出了局部最低延迟算法,局部最低延迟算法具体实现伪代码详见如下。

### 算法 1 局部最低延迟算法

```

mark destination cloudlet for task  $j$  as null;
get the nearest AP to the user, and denote the
nearest AP as  $AP_m$ ;
let  $AP_m$  be the new AP for the user;
let  $s_{AP_m}$  denote the set of cloudlets in the coverage
area of  $AP_m$ ;
for each cloudlet in  $s_{AP_m}$  do
update service latency  $\tau_{s_i,u}$ ,  $s_i \in s_{AP_m}, i = 1, 2, \dots,$ 
 $|s_{AP_m}|$ ;
end for
get the smallest  $\tau_{s_i,u}$  from  $\{\tau_{s_1,u}, \tau_{s_2,u}, \dots,$ 
 $\tau_{s_{|s_{AP_m}|},u}\}$ , and denote the cloudlet which has the
smallest  $\tau_{s_i,u}$  as  $s_m$ ;

```

let the cloudlet  $s_m$  as the destination cloudlet;

对上述算法进行剖析后详述如下:如果微云控制中心收到用户的资源分配请求,微云控制中心将运行迁移策略 4, 将给用户分配连接的 AP 和服务微云。研究时将用户表示为  $u$ , 每个 AP 接入点的覆盖半径为  $R$  的圆形区域。首先,获取最接近用户的 AP 接入点,并将此 AP 接入点表示为  $AP_m$  (请参阅第 2 行)。让  $AP_m$  成为用户的新接入的 AP (见第 3 行)。让  $s_{AP_m}$  表示  $AP_m$  覆盖区域中的微云的集合 (见第 4 行)。  $s_{AP_m}$  中的所有微云都可以与  $AP_m$  直接通信。然后,算法更新每个微云  $s_i$  ( $s_i \in s_{AP_m}, i = 1, 2, \dots, |s_{AP_m}|$ ) 和用户  $u$  之间的服务延迟  $\tau_{s_i,u}$  (请参阅第 6 ~ 9 行), 其中  $|s_{AP_m}|$  表示  $s_{AP_m}$  中的微云的数量。然后,从  $\{\tau_{s_1,u}, \tau_{s_2,u}, \dots, \tau_{s_{|s_{AP_m}|},u}\}$  集合中选择的最小的  $\tau_{s_i,u}$

并表示成  $s_m$  (请参见第 10 ~ 12 行)。最后,该算法将  $s_m$  作为任务  $j$  的目标微云 (见第 13 行)。

### 3 仿真实现

在本节中,将提出的最低局部延迟算法与现有算法进行比较,即设备与 AP 接入点之间的最短距离算法、设备与微云之间的最短距离算法以及最低延迟算法。所有仿真均使用 iFogSim<sup>[13]</sup> 在英特尔酷睿 i5 处理器 (2.7 GHz) 上执行。iFogSim 是一个模拟框架,用于建模和模拟云计算基础设施和服务。

在本文的模拟中,使用了  $200 \times 200$  公里区域。此区域中有一个移动用户。研究中为用户设置了 8 个移动方向。用户在实验开始时随机选择一个方向。用户在实验过程中不会更改其移动方向。用户的移动速度为 19 m/s。每个微云服务器的处理器速度是每秒 2 800 万条指令 (MIPS), 即  $v_{cloudlet} = 2\,800$  MIPS。网络中有 400 个微云和 930 个 AP 接入点。微云均匀分布,且相互间隔为 10 Km, 即  $d_{cloudlets} = 10$  Km, AP 接入点均匀分布,且相互间隔为 6.5 Km, 即  $d_{AP_s} = 6.5$  Km。每个 AP 接入点的覆盖范围为 5 Km, 即  $R = 5$  Km。模拟时间是 1 h。用户每隔 20 s 生成一个任务。任务的上传数据大小为 1 KB, 任务的下载数据大小为 1 KB。用户和 AP 接入点之间的上行带宽为每秒 100 Mbps, 用户和 AP 接入点之间的下行带宽为 2 Mbps。

研究中比较了 3 个不同网络环境中的 4 种算法,即:微云和 AP 接入点之间的带宽均匀分布在 10~20 Mbps 之间;微云和 AP 接入点之间的带宽均匀分布于 90~100 Mbps 之间;微云和 AP 接入点之间的带宽均匀地分布在 990~1 000 Mbps 之间。

### 4 实验结果与分析

本节以服务延迟为性能指标对不同的迁移策略进行仿真实验,不同微云和 AP 接入点之间带宽不同的仿真结果分别如图 1 ~ 图 3 所示。从图 1 中可以看到,迁移策略 4 的服务延迟 (即最低的本地延迟算法) 是最低的。尽管迁移策略 1 选择最接近用户的 AP 接入点并选择距离所选 AP 接入点最近的微云以向用户提供服务,但微云和用户可能不是距离最短的。因此,传输时间可能不是最短的。此外,迁移策略 1 在目标微云选择过程中不考虑微云的处理时间。因此,迁移策略 1 的服务延迟是最大的。为了最大程度地缩短微云和用户之间的传输时间,迁移策略 2 选择最接近用户的微云作为目标微云。因此,传输时间缩短。从图 1 也可以看出,迁移策略 2 的服务延迟小于迁移策略 1 的服务延迟。尽管迁移

策略2选择的目标微云是最接近用户的微云,但考虑到网络拥塞,传输时间可能不是最短的时间。此外,迁移策略1和迁移策略2在微云选择过程中不考虑微云选择时间,这可能会增加服务延迟。为了提供低服务延迟,迁移策略3选择服务延迟最短的微云作为目标微云。因此,可以减少迁移策略3的服务延迟。由图1可知,迁移策略3的服务延迟小于迁移策略1和迁移策略2的服务延迟。但是,要获得每个微云提供的服务延迟,迁移策略3应计算网络中所有微云的服务延迟,对于具有大量微云的网络,这尤其会增加选择时间。为了最小化选择时间,迁移策略4仅计算用户附近微云提供的服务延迟。综合前文分析后,并参见图1可知,迁移策略4的服务延迟最低。这是因为迁移策略4在目标微云选择过程中同时考虑传输时间、计算时间和选择时间。

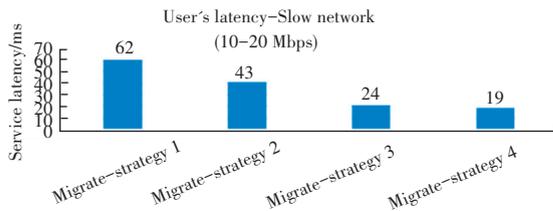


图1 当带宽较小时各种迁移策略的服务延迟

Fig. 1 Service delay for various migration strategies when bandwidth is small

从图2和图3中,还可以看到迁移策略4的服务延迟最低。与其他三种算法相比,迁移策略是目标微云选择的最佳算法,该算法在考虑用户的移动的同时,提供了最低的服务延迟。

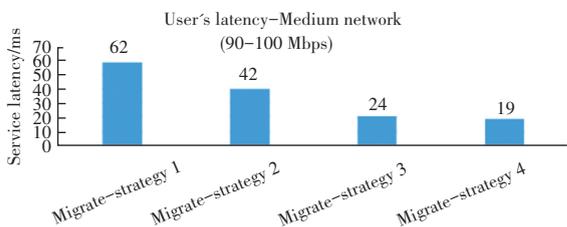


图2 当带宽较中等时各种迁移策略的服务延迟

Fig. 2 Service delay for various migration strategies when bandwidth is moderate

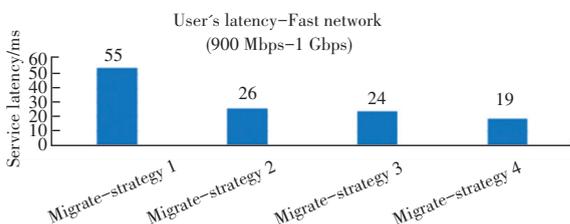


图3 当带宽较大时各种迁移策略的服务延迟

Fig. 3 Service delay for various migration strategies when bandwidth is large

## 5 结束语

本文研究了目标微云的选择问题。提出了最低局部延迟算法,为移动用户提供低服务延迟。在此算法中,在目标微云选择过程中考虑了传输时间、计算时间、微云选择时间和用户位置。研究中还进行了广泛的模拟,以证明最低局部延迟算法的有效性。对于未来的工作,将重点讨论移动用户的一组云子选择和VM选择问题。

## 参考文献

- [1] Cisco, San Jose. CVN index: Global mobile data traffic forecast update White Paper, 2015-2020. [EB/OL]. [2016]. <https://www.scrip.org/reference/ReferencesPapers.aspx?ReferenceID=2307087>.
- [2] PATEL M, NAUGHTON B, CHAN C, et al. Mobile edge computing-introductory technical white paper[R]. USA: Mobile-Edge Computing(MEC) Industry Initiative, 2014.
- [3] DASTJERDI A V, BUYYA R. Fog computing: Helping the internet of things realize its potential[J]. Computer, 2016, 49(8), 112.
- [4] AHMEDA, AHMED E. A survey on mobile edge computing [C]// 10<sup>th</sup> IEEE International Conference on Intelligent Systems and Control. Coimbatore, India; IEEE, 2016: 1.
- [5] HU Y C, PATEL M, SABELLA D, et al. Mobile edge computing -A key technology towards 5G[J]. ETSI White Paper, 2015, 11(11): 1.
- [6] XU Xiaolong, ZHAO Xuan, RUAN Feng, et al. Data placement for privacy-aware applications over big data in hybrid clouds[J]. Security and Communication Networks, 2017, 2017: 2376484.
- [7] XU Xiaolong, DOU Wanchun, ZHANG Xuyun, et al. A traffic hotline discovery method over cloud of things using big taxi GPS data [J]. Software: Practice and Experience, 2016, 47(3): 361.
- [8] XENAKIS D, PASSAS N, MERAKOS L, et al. Mobility management for femtocells in LTE-advanced: Key aspects and survey of handover decision algorithms[J]. IEEE Communications Surveys & Tutorials, 2014, 16(1): 64.
- [9] BITTENCOURT L F, LOPES M M, PETRI I, et al. Towards virtual machine migration in fog computing [C]// 2015 10<sup>th</sup> International Conference on P2P, Parallel Grid Cloud and Internet Computing. Krakow, Poland :IEEE, 2015: 1.
- [10] VELASQUEZ K, ABREU D P, GONCALVES D, et al. Service orchestration in fog environments [C]// International Conference on Future Internet of Things and Cloud. Prague; IEEE, 2017: 329.
- [11] BATISTA D M, GOLDMAN A, HIRATA R, et al. IntersCity: Addressing future Internet research challenges for smart cities [C]// 7<sup>th</sup> IEEE International Conference on the Network of the Future. [S.l.]; IEEE, 2016: 1.
- [12] PULIAFITO C, MINGOZZI E, ANASTASI G. Fog computing for the internet of mobile things: Issues and challenges [J]. IEEE International Conference on Smart Computing. Hong Kong, China; IEEE, 2017: 1.
- [13] GUPTA H, DASTJERDI A V, GHOST S K, et al. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments [J]. arXiv preprint arXiv:1606.02007, 2016.