移动边缘计算中基于贡献度激励的端池化解决方案

阳　柳,章立群,林晓勇
(南京邮电大学 通信与信息工程学院,江苏 南京 210003)

摘　要:经典的移动边缘计算是将计算任务从云计算迁移到移动边缘,终端用户的计算任务请求可智能选择在云端和边缘处理,这都取决于强大的基站通信能力。在基站覆盖力不足,外节点无法接入基站,乃至基站通信缺失的场景下,计算任务无法通过云计算与边缘计算进行处理。因此,提出一种基于贡献度激励的端池化解决方案。该方案由终端提供自身可用计算力,通过端池化帮助基站完成任务计算,针对终端的管理,提出基于算力池最大化的动态分簇算法,该算法利用不同终端作为簇首的差异性聚簇,得到综合算力池最大时的分簇方案;针对基站覆盖力不足的情况,外围节点根据其历史贡献度指标,通过移动自组织网络与内部节点连接,该激励策略能提升内部节点的连接意愿,以此提高外节点的接入率,扩宽基站覆盖范围,解决基站弱覆盖的问题。仿真结果表明,对比其他方案,CETP 方案能在云计算与边缘计算无法实施的情况下,利用端池化过程得到最大算力池,能以最短的时延完成计算任务。

关键词:移动边缘计算;计算力池;端池化;分簇;激励
中图分类号:TP391　　**文献标识码**:A　　**文章编号**:1673-629X(2024)03-0083-06
doi:10.3969/j.issn.1673-629X.2024.03.013

A Solution of Terminal Pooling Based on Contribution Emulated in Mobile Edge Computing

YANG Liu,ZHANG Li-qun,LIN Xiao-yong
(School of Communication & Information Engineering,Nanjing University of Posts and Telecommunications,Nanjing 210003,China)

Abstract:The traditional mobile edge computing is to migrate computing tasks from cloud computing to mobile edge. The computing task requests of end users can be intelligently selected in the cloud and edge processing,depending on the strong communication capability of the base station. In the scenario where the coverage of the base station is insufficient,external nodes cannot access the base station,or even the communication of the base station is missing,computing tasks cannot be processed through cloud computing and edge computing. Therefore,a contribution based incentive end pooling solution is proposed,in which the terminal provides its own available computing power and helps the base station complete task calculations through end pooling. For terminal management,a dynamic clustering algorithm based on computing power pool maximization is proposed. This algorithm utilizes different terminals as cluster heads for differential clustering,and obtains the clustering scheme when the comprehensive computing power pool is maximum. In response to the insufficient coverage of base stations,external nodes are connected to internal nodes through mobile ad hoc networks based on their historical contribution indicators. This incentive strategy can enhance the connection willingness of internal nodes,thereby improving the access rate of external nodes,expanding the coverage range of base stations,and solving the problem of weak base station coverage. The simulation results show that compared with other schemes,CETP can use the end pooling process to obtain the maximum computing power pool when cloud computing and edge computing cannot be implemented,and can complete computing tasks with the shortest delay.

Key words:mobile edge computing;computation pool;terminal pooling;clustering;emulated

0　引　言

目前基于云计算(Cloud Computing,CC)[1]的研究日益成熟。云计算通过分布式计算技术,即将多台计算机的处理能力组合在一起,形成一个虚拟的计算

收稿日期:2023-05-18　　**修回日期**:2023-09-20
基金项目:国家自然科学基金(61801240);南京邮电大学科研项目(2022Y251)
作者简介:阳　柳(1999-),女,硕士研究生,通信作者,研究方向为边缘计算;林晓勇(1974-),男,博士,副教授,研究方向为信息与通信。

机,获取超高计算力[2],此过程又称为公共算力池[3-4](Public Computing Pool, PCP)。移动边缘计算(Mobile Edge Computing,MEC)通过将服务器下沉至数据源一侧为用户提供计算和网络等资源,具有更低的时延和能耗,以完成用户的各种业务请求[5-7]。边缘计算通过基站接收各用户的计算任务,通过边缘侧服务器进行处理,以完成用户的请求。

随着技术的发展,终端节点上行链接接入基站,通过云计算和边缘计算完成计算需求。当基站由于外界因素导致基站覆盖范围缩小,基站计算服务器受损,原本处在基站覆盖范围内的节点无法接入,计算任务无法满足。终端设备都存在一定的计算能力,基于算力共享[4]和公共算力池化的理念,通过移动自组织网络(Mobile Ad hoc Network,MANET)进行连接,利用自身可用计算力实现算力共享,此过程即为端池化(Terminal Pooling)。将端池化后的系统计算合力定义为算力池,端池化以算力池作为评判端池化优劣的标准,在基站无法应付大量计算任务的请求时参与计算。

终端节点的数量影响着系统算力池的大小,目前对 MANET 网络的研究默认节点全都同意接入网络参与通信[8-9],其忽略了节点的自私性。文中通过对节点采取激励策略,提升节点的接入率,增强端计算网络的算力池。由于终端节点自主通信范围受限,通过分簇方式对终端节点进行管理[9-11],并选举簇首节点作为中间节点与基站进行连接,得到算力池最大的分簇方法,不同簇的算力池大小不一,基站可根据算力池下发计算任务。簇首的选举对簇中算力池也存在一定的影响,不同节点具有不同的连接度,作为簇首时组建的簇的特性也不一样。

针对聚簇算法,目前大多数研究都是通过考虑多种因素,通过加权分簇算法选举簇首[11-14]。文献[14]提出一种考虑节点度、节点移动性、节点剩余能量和节点与邻居节点距离和进行加权的方法,并通过带宽确定簇中节点个数,以此提高网络资源效率。该方法虽然考虑了各因素的影响,但忽略了每个因素影响的差异化,即使通过加权因子来平衡各因素的影响,没有从根本上得到最佳的簇首节点。文献[15]通过建立节点剩余能量和位置阈值模型选取簇头及分簇,考虑了节点剩余能量和位置,能够提高能耗效率和对网络生命周期进行分簇,但该方案没有考虑节点计算力对分簇的重要性。文献[16]通过修改 K-means 算法,根据点与计算的质心的距离进行迭代聚类,直到得到一个稳定的质心,并选取最接近质心的 D2D 用户作为该组的 D2D 的簇首,仅以距离作为簇首选举因素,虽然能保证簇的稳定性,却无法保证簇的计算能力。文献[17]通过等角度分簇算法,通过基站到簇首节点的通信半径,得到系统时延最低时的最佳簇首位置,以此选定簇首节点。该文献并没有考虑终端节点直接的通信距离,默认所有节点皆能接入簇中。

综上所述,在已有的分簇方案中,大多都未考虑终端节点通信能力和终端的计算力对簇性能的影响。基于此,该文提出一种基于贡献度激励的端池化(Contribution Emulated Terminal Pooling,CETP)解决方案,重点研究如何提升终端节点的接入率和算力池最大化要求下的簇首选举方式,聚成最佳的簇,通过端池化协助基站快速完成计算任务。

1 系统模型

处于基站覆盖范围内的节点定义为内部节点(Internal Nodes,INs),随着基站覆盖范围缩小,基站覆盖范围外的内部节点定义为外围节点(Peripheral Nodes,PNs),外围节点通过 MANET 网络与内部节点建立通信连接。设定所有 PNs 主动加入 MANET 网络与 INs 建立通信关系,INs 中存在自私节点,不愿与 PNs 通信。端池化过程中下发的计算任务分为两个阶段:第一阶段为基站通过蜂窝网络将计算任务传递给簇首;第二阶段为簇首通过 MANET 网络将计算任务传递给簇内节点,通过利用节点的剩余计算力进行端池化,实现计算力共享。

设定小区用户随机分布,外围节点的数量为 N,内部节点的数量为 M,分别用集合 $A=\{1,2,\cdots,n,\cdots,N\}$ 和 $G=\{1,2,\cdots,m,\cdots,M\}$ 表示,INs 的通信意愿定义为 willingness,用集合 W 表示。系统模型如图 1 所示。

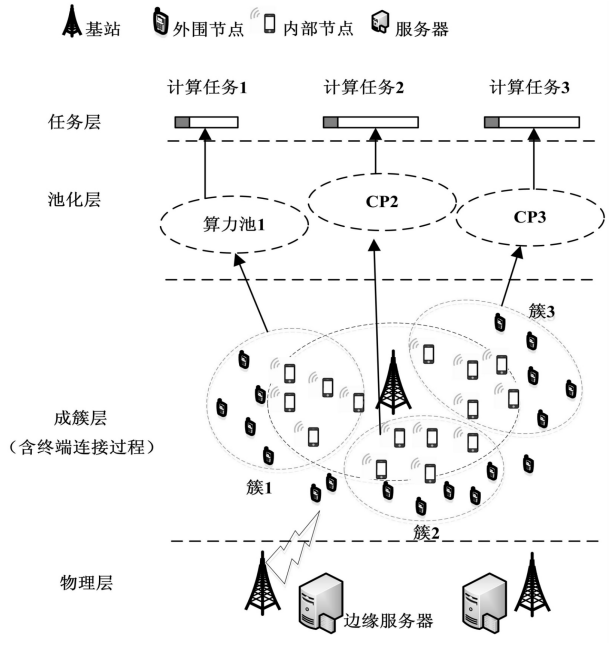

图 1 CETP 模型

2　端池化

端池化过程计算任务传输的第一阶段,基站将计算任务传递给簇首,所需传输时延定义为 $t_{\mathrm{bs,ch}}$,计算过程如公式 1 所示:

$$t_{\mathrm{bs,ch}} = \frac{C}{R_{\mathrm{bs,CH}}} \tag{1}$$

其中, C 为计算任务的大小, $R_{\mathrm{bs,CH}}$ 为基站到簇首的传输速率,可用香农公式表示,计算公式如公式 2 所示:

$$R_{\mathrm{bs,CH}} = \frac{B}{K}\log\left(1 + \frac{P_{\mathrm{bs}} d_{\mathrm{CH}}^{-l}}{n_0 \frac{B}{K}}\right) \tag{2}$$

其中, B 为基站总带宽, K 为簇的个数; P_{bs} 为基站的发射功率, d_{CH} 为簇首到基站的物理距离; l 为路径损耗因子; n_0 为噪声功率谱密度。

在端池化过程计算任务传输的第二阶段,定义簇首(CH)与簇内节点之间的连接关系为 v ,当通信条件时, $v=1$,反之 $v=0$ 。通信条件表示如公式 3 所示:

$$\begin{cases} \mathrm{willingness}(m) \geqslant w_{\mathrm{threshold}}, m \in G & \text{(a)} \\ d_{n,m} \leqslant r, n \in A, m \in G & \text{(b)} \end{cases} \tag{3}$$

通信条件(a)表示簇内中 IN 节点 m 的通信意愿,当 willingness 大于通信意愿阈值 $w_{\mathrm{threshold}}$,表示节点愿意加入与簇首建立通信连接;通信条件(b)表示双方节点的物理通信条件,当簇首与簇内节点之间的物理距离小于最大通信范围 r 时,双方能进行通信。

第二阶段完成计算任务的传输和计算需要的时延为 t_2 ,如公式 4 所示:

$$t_2 = \frac{\frac{C}{\mathrm{Num}_k}}{R_{\mathrm{CH},i}} + \frac{\frac{C\varphi}{\mathrm{Num}_k}}{f_i} \tag{4}$$

其中, Num_k 为第 k 个簇中可通信的簇内节点; f_i 为簇内节点 i 的剩余计算力; φ 为处理 1 bit 任务所需的 CPU 周期; $R_{\mathrm{CH},i}$ 为簇首与簇内节点的传输速率,计算过程如公式 5 所示:

$$R_{\mathrm{CH},i} = B_k \log\left(1 + \frac{P_t d_{\mathrm{CH},i}^{-l} v_{\mathrm{CH},i}}{n_0 B_k}\right) \tag{5}$$

其中, B_k 为第 k 个簇内节点可用带宽; P_t 为簇首的发射功率; $d_{\mathrm{CH},i}$ 为簇内节点 i 与簇首的距离; $v_{\mathrm{CH},i}$ 为簇内节点(CH)与簇首的连接关系。

端池化过程完成计算任务 C 所需花费的总时延包括计算时延和传输时延,计算时延受限于最小节点计算力,传输时延受限于最差节点信道质量,因此总时延为传输时延和计算时延和的最大值。定义总时延为 T ,计算过程如公式 6 所示:

$$T = \max(t_{\mathrm{bs,CH}} + t_2) = \max\left\{\frac{C}{R_{\mathrm{bs,CH}}} + \frac{\frac{C\varphi}{\mathrm{Num}_k}}{R_{\mathrm{CH},i}} + \frac{\frac{C\varphi}{\mathrm{Num}_k}}{f_i}\right\} =$$

$$\max\left\{C\left\{\frac{K}{B\log\left(1 + \frac{KP_{\mathrm{bs}} d_{\mathrm{CH}}^{-l}}{n_0 B}\right)} + \frac{\varphi}{\mathrm{Num}_k B_k \log\left(1 + \frac{P_t d_{\mathrm{CH},i}^{-l} v_{\mathrm{CH},i}}{n_0 B_k}\right)} + \frac{\varphi}{\mathrm{Num}_k f_i}\right\}\right\} \tag{6}$$

算力池为一个簇的可用计算力的合力大小,即一个簇综合处理单位计算任务的能力。第 k 个簇的算力池 F_k 为所需完成的计算任务大小与所花费时延的比值,计算过程如公式 7 所示:

$$F_k = \frac{C}{T} \tag{7}$$

基站与簇首之间的传输时延忽略不计,系统总合力(FF)定义为各簇算力池之和,计算过程如公式 8 所示:

$$\mathrm{FF} = \sum_{k=1}^{K} F_k \tag{8}$$

基于上述分析,可以得到以系统整体算力池最大为优化目标,对端池化过程的用户进行分簇,得到时延优化下分簇的个数以及簇首接入的节点数。以系统整体算力池最大为优化目标可以化作求以系统时延最小为优化目标,计算过程如公式 9 所示:

$$\begin{aligned} \min_{v_{\mathrm{CH},i}, K, \mathrm{Num}_k} T = \min \max\left\{C\left\{\frac{K}{B\log\left(1 + \frac{KP_{\mathrm{bs}} d_{\mathrm{CH}}^{-l}}{n_0 B}\right)} + \right.\right. \\ \left.\left. \frac{\varphi}{\mathrm{Num}_k B_k \log\left(1 + \frac{P_t d_{\mathrm{CH},i}^{-l} v_{\mathrm{CH},i}}{n_0 B_k}\right)} + \frac{\varphi}{\mathrm{Num}_k f_i}\right\}\right\} \\ \text{s.t } \mathrm{C1}: \mathrm{Num}_k \leqslant \mathrm{Nm}_k \\ \mathrm{C2}: \sum_{k=1}^{K} \mathrm{Num}_k \leqslant \mathrm{Sm} \\ \mathrm{C3}: B_k \leqslant B_{\max} \\ \mathrm{C4}: B_k \mathrm{Num}_k \leqslant B_{\mathrm{CH}} \\ \mathrm{C5}: \sum_{k=1}^{K} B_k \leqslant B \end{aligned} \tag{9}$$

其中,C1 是对簇内接入节点的约束,接入节点不能超过簇中节点个数 Sm ;C2 是对整个系统的簇内接入节点的约束,接入节点不能超过系统内节点个数;C3 是簇内节点带宽约束,信道带宽不能超过最大可接入带宽 $B_{\max}$;C4 是簇内的总带宽约束,簇内带宽和不能超过簇中总带宽 B_{CH} ;C5 是系统带宽约束,各簇带宽和不能超过系统总带宽 B 。

3　激励策略

由公式 6 可知,接入的 Num_k 影响算力池大小。为了提升 INs 的连接意愿,提出将贡献度激励 INs 与外围节点进行连接,以此增加通信的 PNs 数,从而增加 Num_k 。PNs 根据以往作为 INs 时,成功连接的次数 Ks

与被申请连接的次数 S 的比定义为节点的贡献度,表示为 Con,如公式 10 所示:

$$\mathrm{Con}(n)=\frac{\mathrm{Ks}}{S},n\in A \tag{10}$$

系统平均贡献度如公式 11 所示:

$$\overline{\mathrm{Con}}=\frac{\sum_{n=1}^{N}\mathrm{Con}(n)}{N} \tag{11}$$

当 PN 节点 n 所申请连接的 IN 节点 m 满足通信条件(b)时,不满足条件(a)时,判断该 PN 节点的贡献度与系统平均贡献度的关系,若大于等于系统平均贡献度,则将该意愿连接节点 m 的连接意愿进行提升,再判双方是否满足通信条件(a),最后得到 INs 的连接数以及 PNs 的接入数。意愿激励公式如下:

$$\mathrm{willingness}'(m)=\mathrm{willingness}(m)+\frac{\alpha\cdot\mathrm{PL}}{\overline{\mathrm{PL}}} \tag{12}$$

配对策略流程如图 2 所示。

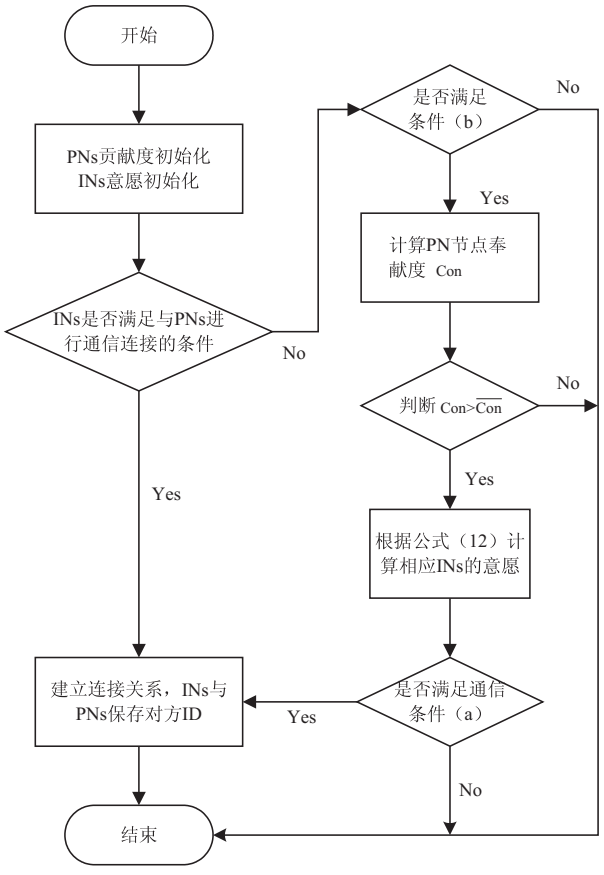

图 2 激励策略流程

4 聚簇算法

文中聚簇算法基于 k -means 算法将系统节点分割为 K 个簇,簇中包括内部节点和外围节点,内部节点可以根据意愿选择是否加入该簇,通过对比入簇的内部节点作为簇首时簇的算力池大小,选出算力池最大时具体簇的分布。根据公式 6 和公式 7 可知,在信道带宽和计算任务相同的情况下,算力池的大小受限于簇中接入数,簇的个数和信道质量,信道质量主要受两点之间的距离影响。

具体算法步骤如下:

输入:$B,B_k,P_{\mathrm{bs}},P_t,C,l,n_0,N,M,r,R,K_{\max}$

输出:v_{ij} , Num_k , K ,FF

步骤 1:初始化相关参数,得到系统所有节点横坐标 X ,纵坐标 Y ;内部节点横坐标 X_n 和纵坐标 Y_n ,内部节点意愿 willingness;外围节点横坐标 X_p 和纵坐标 Y_p ;

步骤 2:for $k\leftarrow K$ _min to K _max do

步骤 3: 通过 K -means 分簇得到初始簇 Cluster1;

步骤 4:For $i\leftarrow 1$ to Num_k do

步骤 5:For $j\leftarrow 1$ to Num_k do

步骤 6:If i ~ = j

步骤 7:计算各簇中的节点距离;

步骤 8:保存各点满足通信的节点坐标;

步骤 9:Endif

步骤 10:Endfor

步骤 11:Endfor

步骤 12: $W_k\leftarrow$满足通信条件(a)的簇中内部节点集合,个数为 Num_k _1;

步骤 13:保存与簇中内部节点满足簇中通信条件的节点,连接关系为 v_{ij} ;

步骤 14:For $i\leftarrow 1$ to Num_k _1 do

步骤 15:根据公式 7 计算得到每个内部节点作为簇首的算力池 $F(i)$;

步骤 16:选举各簇中算力池最大的簇中内部节点作为簇首,保存其 F 值作为 F_k ,并保存该情况下的簇内个数 Num_k ;

步骤 17:根据公式 8 得到系统中总算力池;

步骤 18:Endfor

步骤 19:得到系统合力最大的分簇个数 K ,系统合力为 FF

5 仿真结果及分析

本节详细分析了激励策略下,研究在不同计算任务和小区内节点数的情况下,不同簇首选举算法对系统总时延的影响。为了评估文中算法的性能,将文献[17]的等角度分簇算法作为对比算法 1,加权分簇算法作为对比算法 2,对比算法 2 中影响因素为节点计算力、节点连接度以及节点与簇首的距离和。

对比算法 1 通过等角度分簇,限制了节点根据性能选择簇的权利,不能保证簇中节点加入最佳簇;对比算法 2 通过考虑不同的影响因素进行加权,在一定程度上保证了所分簇的性能,但权值最佳的簇首所聚的簇不一定是最佳性能簇,因此所分簇的性能存在不稳定性。文中算法是确定所分簇的最佳性来反向选举簇首,并通过端池化来确定系统的计算能力,以最短时延完成计算任务。

该文所采用的仿真参数如表 1 所示。

表 1　系统仿真参数设置

变量	仿真参数	仿真数值
p_t	簇首发射功率/dbm	20
p_{bs}	基站发射功率/dbm	30
n_0	热噪声功率/dbm	−144
l	路径损耗因子	4
f_i	本地计算能力/GHz	0.01 ~2
φ	处理 1 bit 任务所需的 CPU 周期	1 500
C	计算任务大小/bit	10^3 ~ 10^4
B	受损边缘服务器可提供的最大信道带宽总和/MHz	10
B_1	BTC 网络中所提供的最大信道带宽总和/MHz	10
B_k	终端节点最大分配带宽/kHz	500
R	基站弱覆盖范围/m	70
M	用户数	300 ~900

当用户数为 300 时,在节点接入的激励策略对比下,不同算法下的用户接入率随计算任务的变化情况如图 3 所示。由于组网过程存在自私节点,因此理论下的最大接入率无法到达 100%。对比算法 1 是通过等角度的分簇方式,限制了簇首节点选择簇中节点,无法最大限度接入满足通信条件节点,因此接入率最低;对比算法 2 虽然在一定程度上考虑了簇首节点的接入率,但在加权的影响下,需要综合考虑计算力与节点度的影响程度,无法确定能选出最大接入度的簇首,因此在接入率上低于文中算法;文中算法是基于算力池最大情况下进行聚簇,而簇中节点数影响着算力池的大小,因此文中算法充分考虑节点接入情况,但簇首节点的通信范围受限,无法达到理论最大值。仿真结果表明,相比其他几种算法,文中算法能够得到最大节点接入率。

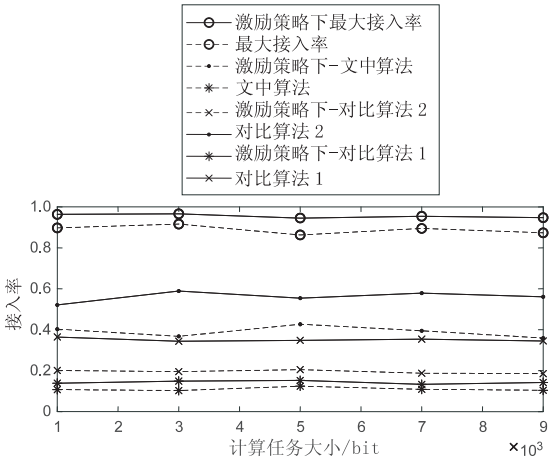

图 3　接入率与计算任务的关系

当用户数为 300 时,在节点接入的激励策略对比下,不同算法下的系统时延随计算任务的变化趋势如图 4 所示。随着计算任务的增大,各种算法时延都有明显的增加。对比算法 1 通过等角度分簇,节点随机分布,无法保证每次分簇的稳定性,总体计算力无法保证;对比算法 2 通过加权多个因素选举簇首,在一定程度上保证了簇的性能,但簇首权值最大不代表整个簇的性能最强。由图 3 可知,文中算法的节点接入率对比其他两种算法更高,因此系统算力池更大,完成计算任务所花费的时延也更低。仿真结果表明,相比其他几种算法,文中算法能够获得最低的时延。

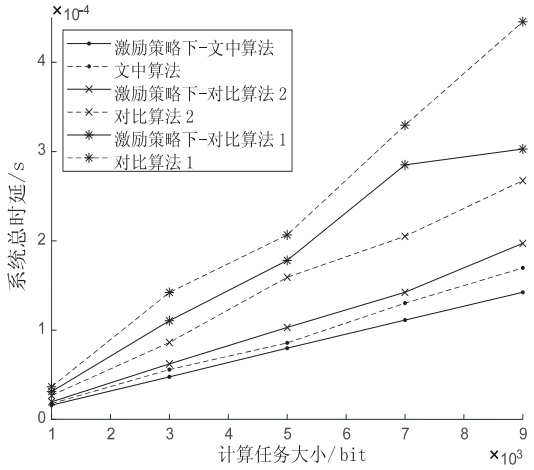

图 4　系统总时延与计算任务的关系

当计算任务为 10^3 bit 时,不同算法下的用户接入率随用户数的变化情况如图 5 所示。小区覆盖范围不变的情况下,当用户数增加时,单位范围内的用户密度会增加,各算法中的簇首所能接入的用户数也会增加,接入端计算网络的节点越多,用户数的增加并不影响接入率的变化,但数量越多,接入率越稳定。

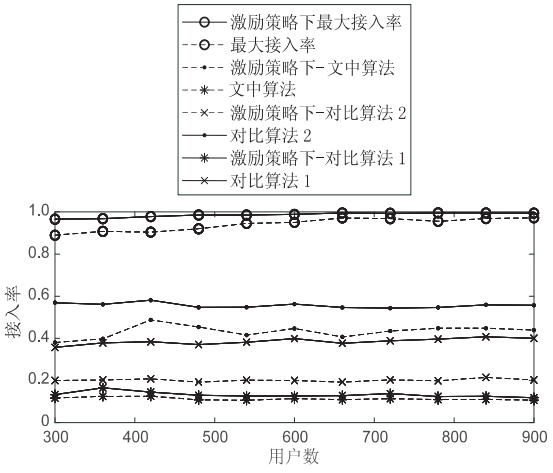

图 5　接入率与用户数的关系

当计算任务为 10^3 bit 时,在节点接入的激励策略对比下,不同算法下的系统时延随用户数的变化趋势如图 6 所示。小区用户数越多,接入端计算网络的节点数越大,因此各节点所分配的计算任务越小,但随着用户密度增加,算力池受到簇中最差信道和节点最小计算力的约束,最后在用户数为 8 000 时趋于稳定。

仿真结果表明,相比其他几种算法,CETP 方案能够获得最低的时延。

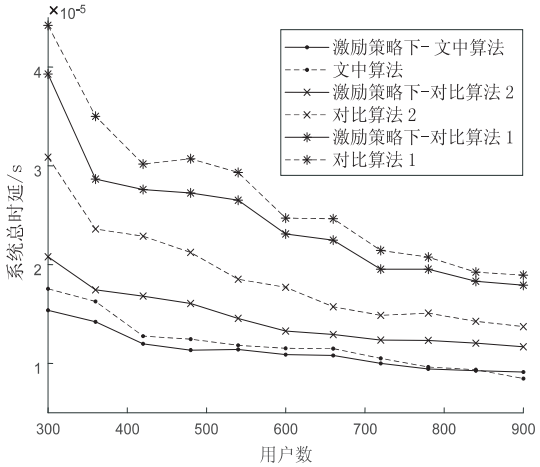

图 6 系统总时延与用户数的关系

6 结束语

该文综合了云计算和移动边缘计算技术,在特定的场景下,提出了移动智能终端构建端池化过程的 CETP 解决方案,并对不同贡献度大小的端节点进行分类,由 MANET 技术构建通信分簇,根据算力池最大化来选举簇首,通过贡献度激励获取更大的用户接入,最终评估出 CETP 可获得各种算力池方案结果。仿真结果表明,CETP 方案可获得 90% 以上的接入率,50% 以上的参与率,较低的系统时延,以及不同用户数变化下相对稳定的计算时延特性。CETP 方案为运营商在雾计算领域中提供了新的算力池化解决思路。

参考文献:

[1] AGEED Z S,ZEEBAREE S R M,SADEEQ M A M,et al. Comprehensive Study of moving from grid and cloud computing through fog and edge computing towards dew computing [C]//2021 4th international Iraqi conference on engineering technology and their applications. Najaf:IEEE,2021:68-74.

[2] KUMAR M,DUBEY K,PANDEY R. Evolution of emerging computing paradigm cloud to fog: applications, limitations and research challenges[C]//2021 11th international conference on cloud computing, data science & engineering (confluence). Noida:IEEE,2021:257-261.

[3] 何凤婕. 数据集群中心云资源池建设方案探讨[J]. 长江信息通信,2022,35(12):149-151.

[4] 陈星延,张雪松,谢志龙,等. 面向“云-边-端”算力系统的计算和传输联合优化方法[J]. 计算机研究与发展,2023,60(4):719-734.

[5] LI X,WAN J,DAI H N,et al. A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing[J]. IEEE Transactions on Industrial Informatics,2019,15(7):4225-4234.

[6] 赵兴兵,李 波,杨志军,等. 边缘计算中的边缘服务器放置策略[J]. 计算机工程与设计,2022,43(11):3008-3014.

[7] SALEEM U,LIU Y,JANGSHER S,et al. Mobility-aware joint task scheduling and resource allocation for cooperative mobile edge computing[J]. IEEE Transactions on Wireless Communications,2021,20(1):360-374.

[8] 邓小明. 基于小规模 MANET 网络的安全通信设计与实现[J]. 数字技术与应用,2022,40(8):232-236.

[9] 胡光明,蒋 杰,龚正虎. 移动自组网络分簇算法综述[J]. 计算机工程与科学,2005,27(1):48-50.

[10] 刘玉涛,张春晖. MANET 网络分簇组网协议比较与分析[J]. 无线电工程,2018,48(2):101-105.

[11] ZHANG J,WANG B,ZHANG F. A distributed approach of WCA in Ad-Hoc network[C]//2006 international conference on wireless communications, networking and mobile computing. Wuhan:IEEE,2006:1-5.

[12] CHATTERJEE M,DAS S K,TURGUT D. An on-demand weighted clustering algorithm (WCA) for ad hoc networks [C]//Proc. of the IEEE global telecommunications conference. San Francisco:IEEE,2000:1697-1701.

[13] SHI F,SHI Y,LAI L. A clustering algorithm of Ad-Hoc network based on honeycomb division[C]//2011 IEEE international conference on granular computing. Kaohsiung:IEEE,2011:863-866.

[14] ZHENG Sihai,ZHENG Changrui. A weight-based clustering routing algorithm for Ad Hoc networks[C]//2021 20th international symposium on distributed computing and applications for business engineering and science. Nanning:IEEE,2021:123-126.

[15] 苗长云,郭 芮,李 杰. 基于节点剩余能量和位置的无线传感网络分簇方法[J]. 天津工业大学学报,2023,42(2):61-66.

[16] SOOR S,CHALLA A,DANDA S,et al. Iterated watersheds, a connected variation of k-means for clustering GIS data [J]. IEEE Transactions on Emerging Topics in Computing,2021,9(2):626-636.

[17] 李旭杰,刘春燕,孙 颖. 面向蜂窝网络的 D2D 多播通信的分簇和中继选择方法[J]. 电子与信息学报,2023,45(2):488-496.