

基于雾计算的能耗最小化公平计算迁移研究

葛欣炜¹, 段聪颖¹, 陈思光^{1,2}

(1. 南京邮电大学 江苏省宽带无线通信和物联网重点实验室, 江苏 南京 210003;
2. 南京邮电大学 江苏省通信与网络工程技术研究中心, 江苏 南京 210003)

摘要:针对现有雾计算网络的迁移优化研究主要集中在降低任务计算时延及能量消耗上,缺乏融合考虑雾节点选择的公平性,该文提出了一种面向雾计算网络的能耗最小化公平计算迁移机制。具体地,构建了一个最小化所有任务完成总能耗的优化问题,充分考虑了任务迁移比、传输功率和雾节点选择的联合优化;基于上述优化问题,提出一种任务迁移候选目的节点集生成算法,通过二分法获得各个雾节点在相应延迟约束下的最低能耗及其对应的迁移比与传输功率;进一步,为了在低能耗与目的节点选择公平性之间取得平衡,基于公平调度指标,提出一种目的节点公平选择算法,以低能耗且公平的方式实现计算任务分配。最后,仿真结果表明,该机制可以在总能耗较低的情况下保证各个雾节点之间的公平性,较最大等效处理速率机制,平均雾节点存活率提升了10.9%。

关键词:雾计算;计算迁移;公平性;能耗;性能优化

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2022)03-0107-07

doi:10.3969/j.issn.1673-629X.2022.03.018

Fog Computing Based Energy Minimization and Fair Computation Offloading Mechanism

GE Xin-wei¹, DUAN Cong-ying¹, CHEN Si-guang^{1,2}

(1. Jiangsu Key Lab of Broadband Wireless Communication and Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;
2. Jiangsu Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: The existing research works on offloading optimization of fog computing networks mainly focus on reducing task computation latency and energy consumption, which ignore the joint consideration of the fairness selection of fog node. Therefore, we propose an energy minimization and fair computation offloading mechanism for fog computing networks. Specifically, an optimization problem with the minimization of total energy consumption for all tasks is formulated, in which the optimization allocation of task offloading ratio, transmission power and fog node selection are jointly considered. According to such optimization problem, a candidate set generation algorithm of destination node for task offloading is developed, and the lowest energy consumption of each fog node under the corresponding delay constraint, the corresponding offloading ratio and transmission power are obtained through the bisection method. Furthermore, in order to make a tradeoff between the low energy consumption and fair selection of destination node, based on fair scheduling indicator, a fair selection algorithm of destination node is proposed to realize the allocation of computation tasks in a low-energy and fair manner. Finally, the simulation results show that this mechanism can ensure the fair selection of the fog node under the condition of low total energy consumption, and the survival rate of fog node is enhanced by 10.9% on average as compared with the maximum equivalent processing rate mechanism.

Key words: fog computing; computation offloading; fairness; energy consumption; performance optimization

收稿日期:2021-04-05

修回日期:2021-08-11

基金项目:国家自然科学基金(61971235);江苏省第五期“333 高层次人才培养工程”(2018);江苏省“六大人才高峰”高层次人才项目(XYDXXJS-044);南京邮电大学“1311”人才计划(2016、2020);中国博士后科学基金(面上一等)(2018M630590);赛尔网络下一代互联网技术创新项目(NGII20190702)

作者简介:葛欣炜(1994-),男,硕士研究生,研究方向为雾计算;通讯作者:陈思光(1984-),男,博士,副教授,研究方向为雾/边缘计算、智能物联网。

1 概述

物联网 (Internet of Things, IoT) 和人工智能 (Artificial Intelligence, AI) 是实现智慧城市的两大基础技术^[1]。随着智慧城市概念的提出,爆发式增长的智能物联网设备产生了巨大的数据和计算工作量^[2]。尤其是,当智能物联网设备在运用深度学习之类的 AI 算法进行大数据分析时,有可能造成过高的延迟。移动云计算 (Mobile Cloud Computing, MCC) 通过将计算任务迁移到远程云数据中心来进行计算,以满足智能 IoT 移动网络中的计算需求^[3],是一种有效的计算模式。但是,由于一般这些远程数据中心距离智能 IoT 移动设备较远,并且它们与远程云数据中心之间的通信主要依赖于骨干网,因此数据中心很难为网络边缘的智能设备中的延迟敏感型应用提供高质量服务^[4]。

移动边缘计算 (Mobile Edge Computing, MEC) 是解决上述问题的新兴计算模式^[5],通过在智能 IoT 设备附近部署边缘云服务器来降低计算延迟。雾计算是 MEC 模式的一个分支,它将计算任务迁移到附近的雾节点 (Fog Node, FN),进一步减少传输延迟并降低智能 IoT 设备的能耗。这是因为与 MCC 相比,雾节点更靠近网络边缘,可以提高智能设备和雾节点之间的响应速度^[6]。但与 MEC 不同,雾节点分布相对密集,难以与有利于迁移的智能设备区分开来^[7],且雾节点尺寸更小,资源更少,电池容量更低,这就使得对雾节点的相关资源进行合理分配以及对网络整体性能的优化变得尤为重要。

文献[8-13]通过联合优化网络资源分配与迁移决策,实现延迟与能耗的最小化,改善了整个雾计算网络的性能。文献[13]构建了含能耗与时间延迟约束的雾节点能耗最小化模型,基于对偶分解的加速梯度算法,快速求出最优迁移比,达到最小化雾节点能耗的目的,同时,极大地改善了传统求解方法的收敛速度,降低了计算任务延迟。文献[14]构建了一个系统能效最大化问题,通过联合优化计算频率、发射功率以及迁移时间,采用广义分式规划理论提出一种迭代算法进行求解,从而获得更高的系统计算能效。

文献[15-17]构建了从用户到 FN 的计算迁移模型,以降低能耗并减少延迟。文献[15]构建了一个混合整数非线性规划问题,提出了一种低复杂度三阶算法,通过平衡终端设备与雾节点的工作负载,实现了计算和传输延迟的最小化。Hu 等人在文献[16]中提出了一种基于雾计算的人脸识别机制,可以提高处理效率,有效地识别个人身份,同时减少网络传输的延迟。文献[17]提出了一种针对雾计算网络中的所有 FN、数据服务运营商和用户的联合优化机制,以实现最优

的分布式资源分配。虽然上述文献所做工作在一定程度上降低了能量消耗并减少了延迟,但其优化结果并不是太理想。尤其是,雾计算网络具有大量的雾节点,其中一些雾节点处于空闲状态,而另一些雾节点几乎一直处于满负荷工作的状态。因此, FN 之间的协作在雾计算网络中起着重要的作用。

文献[18]提出了一种众筹算法来整合网络中闲置资源的机制,并对资源提供者进行监督,有效提高了任务完成效率。文献[19]进一步扩展了上述工作,提出了更有效的激励措施来鼓励设备资源共享。文献[20]构建了一种与传统雾模型不同的雾计算模型,提出的基于博弈论的资源共享机制,提高了计算效率,但其并未考虑雾节点本身资源的有限性。文献[21]在多用户边缘计算场景下,考虑了云边资源的协同优化,提出一种基于次模理论的贪心算法,在用户资源受限的情况下仍能够稳定有效的降低系统能耗与时延。文献[22]将协同雾计算与资源分配优化问题结合,提出了一种基于变量拆分的交替方向乘子法的分布式计算迁移优化算法,以实现给定功率下的最优迁移机制。文献[23]构建了一个任务完成时延与能耗加权和的最小化问题,提出了一个基于深度强化学习的迁移算法,以较低的计算成本实现最优迁移策略的快速求解。

从上述分析可以发现,现有文献大都未考虑计算迁移过程中雾节点选择的公平性,而是侧重于最大程度地减少处理延迟与能耗,这可能会给终端节点 (Terminal Node, TN) 或 FN 带来极大的计算负担,导致 FN 过早失效。这是因为,部分雾节点可能具有强大处理能力和充足电源,而其他的则可能是电源有限甚至是电池供电的节点,显然,电池寿命对于这些 FN 至关重要。考虑到 FN 的不同处理效率和可持续性,在追求低能耗任务迁移的同时,可以通过在 FN 之间公平地进行计算任务迁移,从而防止 FN 由于任务过载而过早失效。

针对上述挑战,面向雾计算网络,该文提出一种能耗最小化公平计算迁移机制,主要贡献如下:

- 在雾计算网络场景下,构建了一个最小化所有任务完成总能耗的优化问题,通过对任务迁移比、传输功率和雾节点选择的联合优化,在一定延迟约束条件下,实现总能耗的最小化。

- 基于上述优化问题,提出一个任务迁移候选目的节点集生成算法,该算法通过二分法,以较低的复杂度获得各个雾节点在相应延迟约束下的最低能耗及其对应的迁移比和传输功率,生成任务迁移候选目的节点集。进一步,为了在低能耗与目的节点选择的公平性之间取得平衡,基于公平调度指标,提出一个目的节点公平选择算法,以公平、节能的方式实现计算任务分

配,防止雾节点过早失效。

· 仿真结果表明,该机制可以在低能耗的情况下保证各雾节点之间的公平性,较最大等效处理速率机制,平均雾节点存活率提升了 10.9%。

2 系统模型

2.1 网络模型

构建的计算任务迁移模型如图 1 所示,考虑一个由 1 个终端节点 TN 和 N 个雾节点 FN 组成的雾簇,该雾簇中的终端节点可以是手机、电脑等智能设备, N 个雾节点(路由设备、运营商提供的各种服务器等)随机分布在终端节点周围。该雾簇中的雾节点分为主动雾节点和被动雾节点,主动雾节点由电网供电,对能量消耗不太敏感,而被动雾节点由电池供电,电池寿命有限,对能耗敏感。

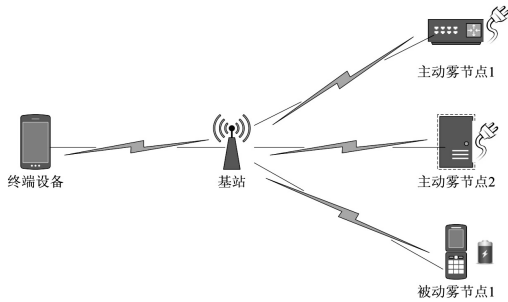


图 1 计算任务迁移模型

当终端生成计算任务时,可以在终端设备本地进行处理,也可以分为多个子任务,将一部分或全部任务迁移到雾节点,再将结果发送回终端节点。

2.2 计算模型

定义处理 1 bit 数据所需的 CPU 周期为 c ,雾节点 i 的 CPU 频率 f_i 和终端节点的 CPU 频率 f_{loc} 。一旦在 TN 上生成了具有延迟约束 T_{max} 的计算任务,除了在本地图理任务之外,TN 会根据已知的 FN 信息和任务的约束条件,要求控制器为此任务的迁移服务分配一个选定的目的节点 FN_i 。然后,根据迁移比 a_i 将任务分为两个子任务,分别在本地和 FN_i 上处理。最后,结果从 FN_i 返回 TN。

(1) 计算延迟。

根据上述迁移过程,任务延迟包括三个部分,即计算任务迁移到雾节点 i 的传输延迟 t_{tra} ,雾节点 i 的计算延迟 t_i ,以及终端节点本地的计算延迟 t_{loc} 。

子任务迁移到雾节点 i 的传输延迟可以表示为:

$$t_{tra} = \frac{a_i w}{R_i} \quad (1)$$

其中, w 是总需要处理的任务大小, R_i 是终端节点到雾节点 i 的数据传输速率,可以用香农公式表示为:

$$R_i = B \log_2 \left(1 + \frac{P_i G^2}{nB} \right) \quad (2)$$

其中, B 是上行链路带宽, P_i 是终端到雾节点 i 的数据传输速率, G 是上行链路的无线信道增益, n 是上行链路的噪声功率谱密度。

雾节点 i 的计算延迟可以表示为:

$$t_i = \frac{a_i w c}{f_i} \quad (3)$$

在终端节点本地计算的延迟可以表示为:

$$t_{loc} = \frac{(1 - a_i) w c}{f_{loc}} \quad (4)$$

终端节点只有在接收到所有子任务的处理结果后才能做出下一步决策,因此,总任务延迟定义为所有子任务延迟的最大值,即总延迟表示为:

$$T = \max [(t_{tra} + t_i), t_{loc}] \quad (5)$$

(2) 能量消耗。

根据前面所描述的计算任务迁移到雾节点 i 的子任务的传输延迟 t_{tra} ,终端节点的传输能耗表示为:

$$e_{tra} = P_i t_{tra} = P_i \frac{a_i w}{R_i} \quad (6)$$

雾节点 i 的能耗可以表示为:

$$e_i = \kappa a_i w c f_i^2 \quad (7)$$

本地处理的能耗可以表示为:

$$e_{loc} = \kappa (1 - a_i) w c f_{loc}^2 \quad (8)$$

其中, κ 表示有效电容系数。

一般地,雾节点处理完成后的数据大小远小于源数据,所以结果返回给终端节点的传输能耗可以忽略不计。因此,整个模型处理任务的总能耗为:

$$E = e_{tra} + e_i + e_{loc} \quad (9)$$

3 优化问题描述

任务延迟和能耗由任务迁移比 a_i 和传输功率 P_i 决定。在满足延迟约束的条件下,恰当地分配 a_i 和 P_i ,进而使总能耗最小,从而将优化问题表述为:

$$\begin{aligned} \text{P1} \quad & \min_{a_i, P_i} E \\ \text{s. t.} \quad & \text{C1} \quad 0 \leq a_i \leq 1 \\ & \text{C2} \quad 0 \leq P_i \leq P_{\max} \\ & \text{C3} \quad T \leq T_{\max} \end{aligned} \quad (10)$$

C1 是任务迁移比约束, $a_i = 0$ 表示终端生成的任务将全部在本地计算, $a_i = 1$ 表示任务将全部迁移到雾节点 i 上进行计算。C2 是传输功率约束,确保终端设备到雾节点 i 的传输功率不超过 P_{\max} 。C3 是延迟约束,确保总任务延迟不超过 T_{\max} 。

4 联合任务迁移比、传输功率与目的节点选择的公平计算迁移算法

4.1 任务迁移候选目的节点集生成算法

由于 P1 是非凸优化问题,所以将其转换成关于终

端传输功率 P_i 的单变量问题。通过二分法,以较低的复杂度获得各个雾节点在相应延迟约束下的最低能耗及其对应的迁移比和传输功率,生成任务迁移候选目的节点集。

首先,在满足 $T \leq T_{\max}$ 的条件下,总能耗 E 最小时,迁移总时延满足 $t_{\text{tra}} + t_i = T_{\max}$,即:

$$t_{\text{tra}} + t_i = \frac{a_i w}{R_i} + \frac{a_i w c}{f_i} = T_{\max} \quad (11)$$

则有:

$$a_i = T_{\max} \left[\frac{w}{B \log_2 \left(1 + \frac{P_i G^2}{nB} \right)} + \frac{wc}{f_i} \right]^{-1} \quad (12)$$

为了保证优化问题 P1 可解,必须满足 $t_{\text{loc}} \leq T_{\max}$ 的条件,结合公式(12),可知当且仅当满足以下两个条件时,总任务时延 T 可以小于上界 T_{\max} ,即:

$$\frac{wc}{T_{\max}} \leq \frac{f_{\text{loc}} + f_i}{c} \quad (13)$$

$$P_i \leq P_{\max} \quad (14)$$

其中,

$$P_i = \left[2^{\left(\frac{T_{\max}}{wc - T_{\max} f_{\text{loc}}} - \frac{1}{f_i} \right)^{-1} c^{-1} B^{-1}} - 1 \right] \frac{NB}{G^2} \quad (15)$$

优化问题 P1 在这两个条件下是可解的。则当 $t_{\text{loc}} \leq T_{\max}$ 时,传输功率 P_i 的下界为 $\max(0, P_i^{\vee})$ 。

如果不满足上述条件,则无论如何从 a_i 和 P_i 的取值范围中选取哪一个值,总任务延迟 T 都会超过其上限 T_{\max} ,这意味着将雾节点 FN_i 用于计算任务迁移服务是不可行的。对于可行的 FN_i ,当 $P_i = P_i^{\vee}$,且 $P_i^{\vee} \geq 0$ 时,本地处理延迟 $t_{\text{loc}} = T_{\max}$ 。如果存在 $P_i^{\vee} \leq 0$,则意味着在延迟约束下在所有的计算任务都将在本地设备上进行处理,其中传输功率 $P_i = 0$ 。

由于 P1 的非凸性,原问题 P1 转化为:

$$\text{P2} \quad \min_{P_i} E = \kappa w c f_{\text{loc}}^2 + T_{\max} f_i \frac{P_i + \kappa c (f_i^2 - f_{\text{loc}}^2) R_i}{R_i c + f_i} \quad (16)$$

$$\text{s. t.} \quad \max(0, P_i^{\vee}) \leq p_i \leq \min(P_{\max}, P_i^{\wedge})$$

其中,

$$P_i^{\wedge} = \begin{cases} \left[2^{\left(\frac{T_{\max}}{wc - T_{\max} f_{\text{loc}}} - \frac{1}{f_i} \right)^{-1} c^{-1} B^{-1}} - 1 \right] \frac{nB}{G^2}, & \frac{wc}{T_{\max}} \leq \frac{f_i}{c}, \\ +\infty, & \frac{wc}{T_{\max}} > \frac{f_i}{c} \end{cases} \quad (17)$$

若存在 $P_i = P_i^{\wedge}$,则在延迟约束下所有的计算任务都将迁移到 FN_i 上进行处理,传输功率 $P_i = P_i^{\wedge}$ 。

现在,针对 a_i 和 P_i 的优化问题 P1 被转换为针对

TN 传输功率 P_i 的能量最小化问题 P2。

通过对传输功率 P_i 求二阶导数可知,总能耗 E 随传输功率 P_i 单调递增,则最优的传输功率 P_i^* 将直接由 P_i 定义域中的最小值给出,那么,在问题 P2 的取值范围中,若满足如下条件:

$$1 + \frac{\kappa c (f_i^2 - f_{\text{loc}}^2) G^2}{n \ln 2} \geq 0 \quad (18)$$

总能耗 E 是关于传输功率 P_i 的严格单调递增函数。因此, P_i 最优值等于 P_i 取值范围内的最小值,即:

$$P_i^* = \max(0, P_i^{\vee}) \quad (19)$$

对于系统参数不满足上述条件的情况,当 $P_i \in [0, +\infty)$ 时,若满足如下条件:

$$1 + \frac{\kappa c (f_i^2 - f_{\text{loc}}^2) G^2}{n \ln 2} < 0 \quad (20)$$

则当 $P_i \in [0, P_i^{\vee})$ 时,总能耗 E 是关于传输功率 P_i 的严格单调递减函数,当 $P_i \in (P_i^{\vee}, +\infty)$ 时,总能耗 E 是关于传输功率 P_i 的严格单调递增函数,则当传输功率 $P_i = P_i^{\vee}$ 时,整体能耗 E 最小。其中, P_i^{\vee} 的值是以下方程唯一的正解,可以通过二分法求得。

$$R_i c + f_i + f_i \kappa c B (f_i^2 - f_{\text{loc}}^2) \frac{\partial R_i}{\partial P_i} - P_i c B \frac{\partial R_i}{\partial P_i} = 0 \quad (21)$$

所以,

$$P_i^* = \min[\max(P_i^{\vee}, P_i^{\vee}), \min(P_{\max}, P_i^{\wedge})] \quad (22)$$

再根据公式(12)可以求出最优迁移比 a_i^* 。

基于上述推导过程,引入如下算法 1。对于空闲的 FN ,该算法验证了 FN 进行计算任务迁移的可行性。此外,对于可行的 FN ,结果提供了最优传输功率 P_i^* 和最优迁移比 a_i^* ,从而实现能耗最小化。

算法 1:任务迁移候选目的节点集生成算法。

1. 输入:任务大小 w ,雾节点 i 的 CPU 频率 f_i
2. 输出:候选目的节点集 \mathbb{N} with (a_i^*, p_i^*)
3. BEGIN
4. $\mathbb{N} = \emptyset$;
5. for each i
6. 基于公式(15),计算 P_i^{\vee} 的值;
7. if $\frac{wc}{T_{\max}} > \frac{f_{\text{loc}} + f_i}{c}$ or $P_i^{\vee} > P_{\max}$, then
8. $\mathbb{N} = \mathbb{N} \cup \emptyset$;
9. else
10. if 条件(18)满足, then
11. 基于公式(19),计算 p_i^* 的值;
12. else
13. 基于公式(17),(21),(22),分别计算 P_i^{\wedge} 、 P_i^{\vee} 、 P_i^* 的值;
14. end if
15. 输入 p_i^* 至公式(12),计算 a_i^* 的值;
16. $\mathbb{N} = \mathbb{N} \cup \{i\}$;


```

17. end if
18. end for
19. return  $\mathbb{N}$  with  $(a_i^*, p_i^*)$ .
20. END

```

4.2 目的节点公平性选择

各个雾节点 FN 之间的能耗均衡对于雾计算网络的可持续性和稳定性至关重要。在这一部分,基于上述任务迁移候选目的节点集生成算法,提出一种目的节点的公平性选择算法。因此,在延迟约束下,总能耗和计算任务公平分配之间实现了平衡。

首先,为每个雾节点定义公平调度指标,如下:

$$M_i = \frac{\lambda_i}{\bar{E}_i E^*} \quad (23)$$

其中, \bar{E}_i 表示雾节点 i 的历史平均能耗, E^* 表示通过雾节点 i 进行计算迁移服务的最低总能耗, λ_i 表示雾节点 i 的优先级指数。

历史平均功耗更新如下:

$$\bar{E}_i' = (1 - \mu) \bar{E}_i + \mu E_i \quad (24)$$

其中, μ 是遗忘因子, $0 < \mu < 1$ 。

引入 Jain's 公平指数:

$$F = \frac{(\sum_{i \in \mathbb{N}} \bar{E}_i)^2}{N \sum_{i \in \mathbb{N}} \bar{E}_i^2} \quad (25)$$

其中, Jain's 公平指数 F 在 $1/N$ 到 1 之间, F 越大,表示计算任务迁移机制的公平性越高。

基于算法 1 得到的候选目的节点集和公平性调度指标,提出如下算法 2。

算法 2: 目的节点公平选择算法。

```

1. 输入: 候选目的节点集  $\mathbb{N}$ 
2. 输出: 最优目的节点  $i^*$ 
3. BEGIN
4. if  $\mathbb{N} = \emptyset$ , then
5.  $i^*$  is null;
6. else
7. for each  $i \in \mathbb{N}$  do
8. 基于公式(23), 计算公平调度指标  $M_i$ ;
9. end for
10. 选择公平性指数最高的雾节点进行迁移,
    即  $\text{FN } i^* = \arg \max_{i \in \mathbb{N}} M_i$ ;
11. return  $i^*$ 
12. end if
13. END

```

5 仿真结果

在本节中,将进行仿真以验证提出的能耗最小化公平迁移算法的性能。在多种网络场景下,对最小化的迁移能耗、迁移服务的可行性以及雾节点选择的公

平性进行了验证分析。

在本仿真环境中,设定每个雾节点的带宽 $B = 10$ Mb/s, 上行链路的噪声功率谱密度 $n = -100$ dBm/Hz, 总的雾节点个数为 10 个, 任务总数为 20 个, 终端节点的计算速率 $f_{\text{loc}} = 2$ Mbps。

图 2 是总能耗与雾节点的任务处理速率关系图。可以看出,随着 f_i 的增加,雾节点的处理速率上升,雾节点处理的能效比本地处理的能效高,通过将计算任务迁移到雾节点进行处理,减少了总能量消耗。随着 FN_i 的 CPU 频率的进一步提高,该任务将完全迁移到 FN_i 上进行处理,这导致恒定的总能耗。因此,当 f_i 足够大时,总能量消耗趋于收敛。从图中还可以看出,当生成的计算任务量越大,进行计算迁移需要任务处理速率越快。

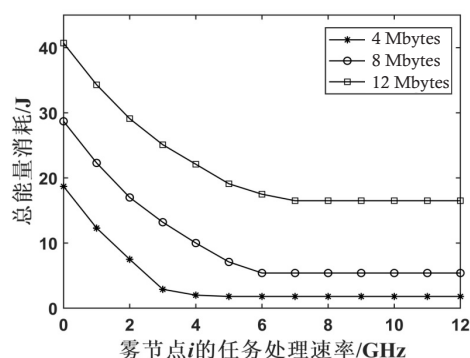


图2 总能耗与雾节点的任务处理速率关系图

图 3 是不同雾节点分布半径下三种任务迁移机制的 Jain's 公平性指标图。可以看出,随着雾节点分布半径增大,最大等效处理速率机制和完全相等任务迁移机制的公平性指标均大幅下降,远低于该文所提机制的公平性指标。这是因为半径越大雾节点分布越稀疏,选择远程雾节点进行任务迁移的概率越小。该文提出的公平迁移机制在雾节点半径变化时始终保持较高的公平性指数,这是由于它在所有备选雾节点之间采取公平的迁移策略。

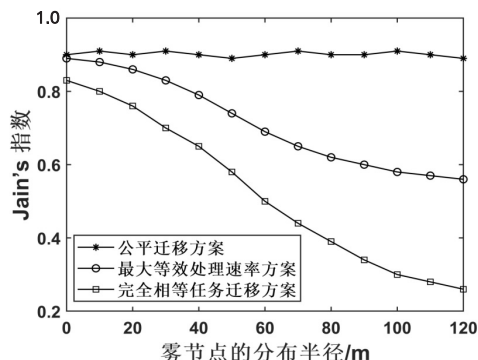


图3 雾节点分布半径与 Jain's 指数关系图

图 4 是任务迁移概率与最大传输功率的关系图。可以看出,随着终端发射功率上限的增大,任务迁移的

概率从 0 增加到 1, 并且增加的幅度近似线性。图 5 是任务迁移概率与最大容忍延迟的关系图。可以看出, 随着延迟上限的增加, 计算任务迁移概率从 0 增加到 1。当最大容忍延迟为 2 s 时, 4 MB 大小的任务可以全部在本地处理以满足延迟上限, 因此该情况下任务迁移概率始终为 0。结果表明, 更严格的延迟和传输功率约束导致能够为 TN 提供可行的任务迁移服务的 FN 减少。

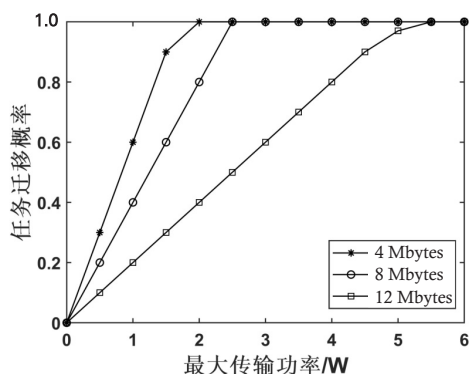


图 4 任务迁移概率与最大传输功率关系图

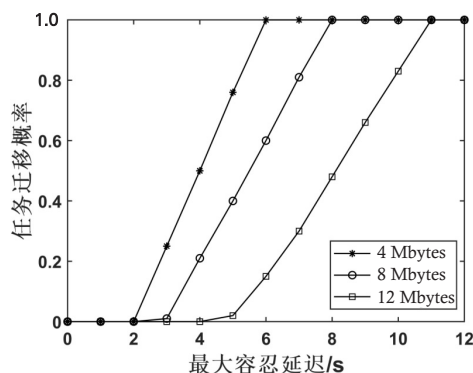


图 5 任务迁移概率与最大容忍延迟关系图

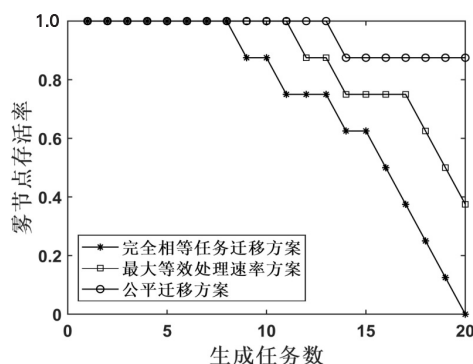


图 6 雾节点存活率与生成任务数关系图

图 6 是雾节点存活率与生成任务数的关系图, 可以看出, 随着生成任务数的不断增加, 三种机制的雾节点存活率都呈下降趋势, 这是因为随着任务数的不断增加, 能量有限的雾节点能量不断消耗殆尽, 最终当能量耗尽时雾节点也就随之死亡。而该文的公平迁移机制雾节点的存活率始终更高, 比完全相等任务迁移机

制和最大等效处理速率机制的平均雾节点存活率分别高 31.9% 和 10.9%。这是因为该文考虑了雾节点之间迁移的公平性, 终端节点始终倾向于将任务迁移给剩余能量较高且历史平均能耗较低的雾节点, 保证了雾节点之间的公平性, 从而保证了雾节点不会过早耗尽能量而失效。

6 结束语

该文提出了一个基于雾计算的能耗最小化公平计算迁移机制, 通过联合优化迁移比、传输功率和雾节点选择达到公平且低能耗的计算迁移。提出任务迁移候选目的节点集生成算法, 通过二分法获得雾节点在相应延迟约束下的最低能耗及其对应的迁移比和传输功率; 基于公平调度指标, 通过目的节点公平选择算法, 以低能耗且公平的方式进行计算任务分配, 实现低能耗与目的节点选择公平性之间的平衡。仿真结果表明, 该机制可以在总能耗较低的情况下保证各个雾节点之间的公平性。

参考文献:

- [1] SHARIF A, LI J, KHALIL M, et al. Internet of Things-smart traffic management system for smart cities using big data analytics [C]//Proceedings of international computer conference on wavelet active media technology and information processing. Chengdu, China: IEEE, 2017: 281-284.
- [2] SUN X, ANSARI N. Dynamic resource caching in the IoT application layer for smart cities [J]. IEEE Internet of Things Journal, 2018, 5(2): 606-613.
- [3] KHAN A, OTHMAN M, MADANI S, et al. A survey of mobile cloud computing application models [J]. IEEE Communications Surveys & Tutorials, 2014, 16(1): 393-413.
- [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] JIA G, HAN G. Edge computing-based intelligent manhole cover management system for smart cities [J]. IEEE Internet of Things Journal, 2018, 5(3): 1648-1656.
- [6] WU Y, WU J, CHEN L, et al. Fog computing model and efficient algorithms for directional vehicle mobility in vehicular network [J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(5): 2599-2614.
- [7] STOJIMENOVIC I, WEN S. The fog computing paradigm: scenarios and security issues [C]//Proceedings of federated conference on computer science and information systems. Warsaw, Poland: IEEE, 2014: 1-8.
- [8] CHEN S, ZHU X, ZHANG H, et al. Efficient privacy preserving data collection and computation offloading for fog-assisted IoT [J]. IEEE Transactions on Sustainable Computing, 2020, 5(4): 526-540.

- [9] LI S, TAO Y, QIN X, et al. Energy-aware mobile edge computation offloading for IoT over heterogenous networks[J]. IEEE Access, 2019, 7: 13092–13105.
- [10] WANG J, FENG D, ZHANG S, et al. Computation offloading for mobile edge computing enabled vehicular networks[J]. IEEE Access, 2019, 7: 62624–62632.
- [11] GUO K, SHENG M, QUEK T, et al. Task offloading and scheduling in fog RAN: a parallel communication and computation perspective[J]. IEEE Wireless Communications Letters, 2020, 9(2): 215–218.
- [12] MUKHERJEE M, KUMAR S, SHOJAFAR M, et al. Joint Task offloading and resource allocation for delay-sensitive fog networks[C]//Proceedings of IEEE international conference on communications. Shanghai, China: IEEE, 2019: 1–7.
- [13] CHEN S, ZHENG Y, WANG K, et al. Delay guaranteed energy-efficient computation offloading for industrial IoT in fog computing[C]//Proceedings of IEEE international conference on communications. Shanghai, China: IEEE, 2019: 8–13.
- [14] 施丽琴, 叶迎晖, 卢光跃. 无线供能边缘计算网络中系统计算能效最大化资源分配方案[J]. 通信学报, 2020, 41(10): 59–69.
- [15] ZENG D, GU L, GUO S, et al. Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system[J]. IEEE Transactions on Computers, 2016, 65(12): 3702–3712.
- [16] HU P, NING N, QIU T, et al. Fog computing based face identification and resolution scheme in Internet of Things[J]. IEEE Transactions on Industrial Informatics, 2017, 13(4): 1910–1920.
- [17] ZHANG H, XIAO Y. Computing resource allocation in three-tier IoT fog networks: a joint optimization approach combining stackelberg game and matching[J]. IEEE Internet of Things Journal, 2017, 4(5): 1204–1215.
- [18] SUN Y, ZHANG N. A resource-sharing model based on a repeated game in fog computing[J]. Saudi Journal of Biological Sciences, 2017, 24(3): 687–694.
- [19] SUN Y, LIN F. Non-cooperative differential game for incentive to contribute resource-based crowd funding in fog computing[J]. Boletín Técnico/Technical Bulletin, 2017, 55(8): 69–77.
- [20] KAPSALIS A, KASNESIS P, VENIERIS I, et al. A cooperative fog approach for effective workload balancing[J]. IEEE Cloud Computing, 2017, 4(2): 36–45.
- [21] 梁冰, 纪雯. 基于次模优化的边云协同多用户计算任务迁移方法[J]. 通信学报, 2020, 41(10): 25–36.
- [22] XIAO Y, KRUNZ M. QoE and power efficiency tradeoff for fog computing networks with fog node cooperation[C]//Proceedings of IEEE conference on computer communications. Atlanta, USA: IEEE, 2017: 1–9.
- [23] 陈思光, 陈佳民, 赵传信. 基于深度强化学习的云边协同计算迁移研究[J]. 电子学报, 2021, 49(1): 157–166.