

EdgeFlow 移动边缘计算在物联网中的应用研究

丛书畅, 姚超, 王鹏飞, 郑子杰, 宋令阳

(北京大学现代通信研究所, 北京 100871)

摘 要: 为应对未来物联网更高计算速率、更低通信时延和更高接入密度的要求, 提出了移动边缘计算框架 EdgeFlow, 考虑了阻塞态和非阻塞态的任务卸载算法。通用软件无线电平台模拟实验表明, EdgeFlow 可以显著提高计算效率、降低通信时延, 为解决物联网更高计算需求的问题指明了新的发展方向, 奠定了理论与实践基础, 可以广泛应用于物联网场景中。

关键词: 物联网; 移动边缘计算; 任务卸载; 资源调度

中图分类号: TP391

文献标识码: A

doi: 10.11959/j.issn.2096-3750.2019.00098

Application of EdgeFlow mobile edge computing in Internet of things

CONG Shuchang, YAO Chao, WANG Pengfei, ZHENG Zijie, SONG Lingyang

Institute of Modern Communication, Peking University, Beijing 100871, China

Abstract: In order to meet the requirements of higher computing speed, lower communication delay and higher access density in the future Internet of things, a computational framework of EdgeFlow for mobile edge computing and a task unloading algorithm considering blocking and non-blocking state were proposed. The simulation experiments on the universal software radio platform show that the computational efficiency can be improved significantly and the communication delay can be reduced by EdgeFlow. It points out a new development direction, lays a theoretical and practical foundation for solving the problem of higher computational requirements of the Internet of things. And it will be widely used in the Internet of things scenarios.

Key words: Internet of things, mobile edge computing, task offloading, resource allocation

1 引言

随着物联网技术的发展, 用户对数据通信的需求逐渐增多^[1]。物联网技术一般被定义为依托于无线通信技术, 通过物理场景的实物与互联网的连接, 实现物与物的连接、物与人的连接, 最终实现万物互联^[2], 旨在搭建以终端设备和云计算中心为核心的分布式计算框架, 通过网络的智能感知和协作资源调度, 实现支持高密度、低能耗和低时延的通信处理网络^[3]。然而, 一些物联网应用需要较低的响应时延, 并且可能间歇性地生成大量数据, 将导致物联网网络的重负载。预测到 2020 年, 将有

500 亿个物联网设备连接到互联网^[4], 可能带来的问题包括: 1) 大量的原始数据和计算任务需要处理, 而每个边缘设备的计算能力有限。2) 为了满足实时任务的要求, 需要以较低时延通过网络传输大量数据, 而网络中的无线和有线传输资源有限。

为了解决第一个问题, 传统的云计算模型侧重于加强中心计算能力, 边缘设备一般仅负责数据收集, 复杂计算任务则由云计算中心完成。主要因为在传统的物联网场景中, 终端设备一般是功能较单一的数据采集传感器, 计算能力与续航能力有限, 这种情况会加剧第二个问题, 数据传输距离较长, 通信过程会导致时延较高。随着芯片技术的发展,

网络中的无线接入点如基站、Wi-Fi 等, 都拥有较强的计算能力。此外, 大多数物联网终端设备也拥有部分计算资源, 现有的大量计算设备都已运行 Linux 操作系统^[5]。因此, 可以考虑将部分计算任务卸载到计算资源比较富裕的网络节点, 移动边缘计算 (MEC, mobile edge computing) 的概念便被引入现代计算网络体系中, 可以应用于物联网领域, 通过虚拟化技术利用无线基站的空闲资源缓解计算系统中计算资源有限的问题, 并实现靠近数据源的低时延计算^[6]。

本文在 MEC 框架下, 基于流式数据任务提出边缘计算节点和云计算中心协同工作的边缘计算框架 EdgeFlow, 并解决了相应的卸载问题使得任务时延最优化。在流式数据任务中, 考虑任务时延受任务执行时间和任务阻塞状态的共同影响, 因此, 本文提出了基于阻塞判断的可调度任务卸载算法。在阻塞状态下, 该算法优化系统吞吐量以缓解系统阻塞情况, 从而减小任务时延; 在非阻塞状态下, 该算法通过调度任务卸载策略和资源分配策略从而降低任务时延。仿真实验基于 Linux 开源系统和通用软件无线电平台 (USRP, universal software radio peripheral) 设计并实现实验平台, 该系统可以模拟实现多层网络、不同的通信和计算场景。基于上述实验平台, 在人脸识别场景中论证基于流式数据的多层网络任务卸载策略可明显降低任务时延, 提高系统吞吐量, 通过网络中的通信与计算资源平衡以避免系统进入阻塞状态, 从而保证系统稳定运行。

2 EdgeFlow 整体架构

根据欧洲电信标准协会 (ETSI, European telecommunication standard institute) 对于 MEC 的定义, 即通过部署于蜂窝基站的 MEC 服务器, 提供靠近移动用户的通信与计算服务, 减少网络传输时延并提高服务体验。MEC 框架如图 1 所示, 通过将服务器部署于靠近数据源的无线基站, 以近距离向用户提供计算服务, 降低任务时延。同时通过边缘计算节点对数据的预处理, 使得系统只需将少量计算结果传输到云计算中心。因此, MEC 架构不仅可以提升运算速率、降低时延, 还可以缓解核心网的传输压力^[7-8]。

按照功能性划分, 将系统分为虚拟层、平台层和应用层, MEC 功能划分如图 2 所示。其中, 虚拟层利用虚拟化技术, 通过将设备虚拟为逻辑计算单

元使得现有的计算、存储和通信等设备能够在系统中相互兼容, 克服了由于传统计算设备的硬件隔离导致的计算网络难以突破硬件资源限制的问题, 实现了资源兼容。平台层在虚拟层之上, 针对系统应用需求进行虚拟设备的资源管理, 调用实际硬件资源以提供对上层系统兼容稳定的虚拟计算或通信资源, 并开放 API 以供上层应用。应用层则利用虚拟层开放的 API, 实现虚拟资源池的资源按需分配和调度, 从而满足计算和通信需求, 包括数据分流、任务卸载和资源调度等。另外, 基于边缘计算系统的平台层接口, 边缘计算应用可以实现任务卸载, 将部分计算复杂型任务交给上层富裕的计算节点计算, 实现网络中异构设备的协同计算, 以突破物理设备的硬件隔离, 从而实现网络资源的协作共享。

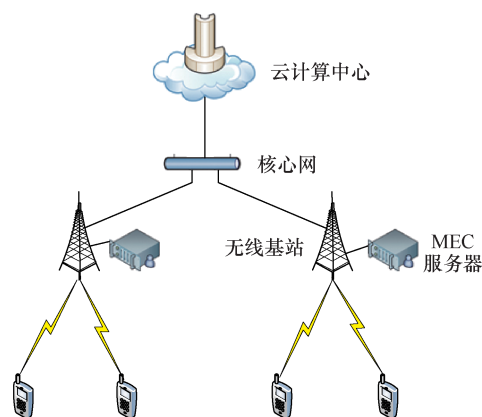


图 1 MEC 框架

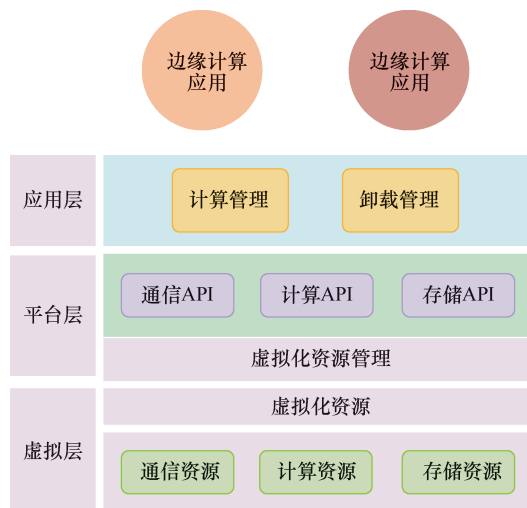


图 2 MEC 功能划分

图 3 为本文所提的 EdgeFlow 框架的系统架构, 底层为终端设备 (以下简称“ED”), 如无线传感器; 中间层是无线接入点 (以下简称“AP”), 如 Wi-Fi

接入点；顶层为云计算中心（以下简称“CC”）。ED 与 AP 之间运用无线通信技术，AP 与 CC 之间运用有线通信技术，并且该架构考虑了实际情况与未来发展趋势，可以假设节点均具有一定的计算和通信能力。当系统发起计算任务时，CC 利用 ED 和 AP 的空闲计算和通信能力，将部分任务分配给 ED 和 AP 进行计算，利用网络整体计算资源协同完成任务。针对如何合理分配资源并将任务卸载到各层节点的问题，是本文研究的关键。

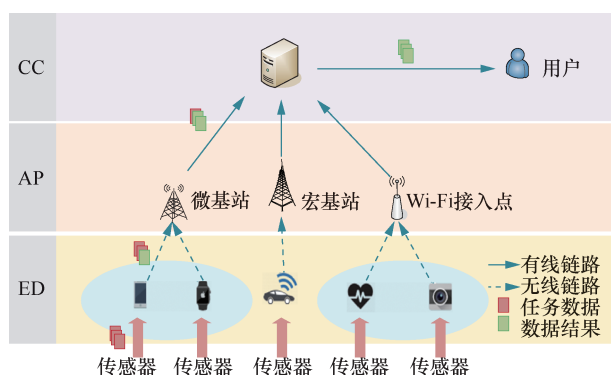


图 3 EdgeFlow 框架的系统架构

3 任务卸载问题

任务卸载问题在传统的移动云计算场景中已被提出，考虑终端节点的计算能力有限，因此把全部任务卸载到 CC，以满足终端设备的计算需求。但是，在当前以及未来的物联网背景下，终端节点的计算能力越来越强，而且将计算任务全部卸载到 CC 会导致较大的通信时延，影响用户体验，因此 MEC 框架被引入计算系统中。利用部署于网络边缘的计算中心，移动用户可以享受优质、低时延的计算服务。但是由于边缘计算节点的资源受限，当数据任务量巨大时，仅利用边缘计算节点无法为用户维持稳定、优质的计算服务。因此，在 MEC 场景中，任务卸载问题的关键是如何运用策略将计算任务合理分配卸载到边缘计算服务中心以及应该选择多少通信资源进行卸载才能保证应用服务质量。

在 MEC 场景中，数据处理或传输分为以下 5 个阶段，即分配给每个 ED 的计算任务 T_{ED} 、由 ED 向 AP 的数据传输任务 t_{ED} 、分配给每个 AP 的计算任务 T_{AP} 、由 AP 向 CC 的数据传输任务 t_{AP} 以及分配给每个 CC 的计算任务 T_{CC} 。

在 EdgeFlow 框架中，5 个阶段并行进行（如当 ED 向 AP 传输数据时，AP 可迅速开始处理数据）。

为了满足低任务时延的需求，系统的优化目标为通过合理分配任务使得时延最小化。与传统场景的任务时延目标不同，由于未来数据量巨大，所以数据流的任务时延不仅受任务执行时间影响，还受任务等待时间影响，因此需要考虑阻塞状态。传统方案关注资源充沛的非阻塞场景，而忽略了阻塞情况下的任务等待时间，该假设在物联网场景下将不再适用。基于系统阻塞状态进行分析，在阻塞状态下，系统受阻塞状态影响，为了使任务时延最小化，其优化目标应该是如何减小系统的阻塞情况。因此，在阻塞情况下，应使系统吞吐量最大化以分担瓶颈节点的任务负载，从而减小系统的阻塞情况和数据分组的等待时间。当系统在非阻塞状态下，系统中数据分组无阻塞，任务时延取决于任务执行时间。因此，系统在非阻塞情况下，需要最小化任务执行时间，任务执行时延为数据的实际计算和传输时间间隔，其组成包括数据分组所有途经节点的计算时间和传输时间。所以，为了更好地解决优化问题，本文将针对边缘系统的阻塞状态与非阻塞状态进行分析讨论。

EdgeFlow 任务卸载过程如图 4 所示，EdgeFlow 框架根据功能性可以划分为虚拟化资源层、系统 API 层和应用层。虚拟化资源层的主要功能是将硬件的通信资源和计算资源进行虚拟化，提供虚拟化的计算与通信资源接口给上层系统。系统 API 层的主要功能是对系统虚拟化资源进行管理，并整合为计算 API 和通信 API，以方便边缘计算系统调用。应用层包括三大功能块，即资源评估模块、计算管理模块和卸载管理模块。资源评估模块对系统各个节点的空闲资源进行评估并汇总于 CC，由 CC 进行任务卸载算法分析，获得适宜的任务卸载策略。应用层通过卸载管理模块管理调度任务卸载策略，通过计算管理模块调用任务的执行逻辑。任务卸载策略决定任务卸载的上行比例，实际工作逻辑可简单概括为 ED 和 AP 向 CC 汇报计算资源条件和通信资源条件，由 CC 进行资源分析。CC 根据系统资源条件进行阻塞判断，从而选择合适的任务卸载算法，以更新 ED 和 AP 的卸载策略。对数据流的每一个计算数据分组，系统基于卸载概率判断决定本地计算或上行卸载，实现基于数据的任务部分卸载。具体实现过程为根据任务卸载概率进行卸载，对单一任务数据分组，通过随机概率判断是否上行卸载或以固定的卸载序列实现部分卸载，如 1/3 的

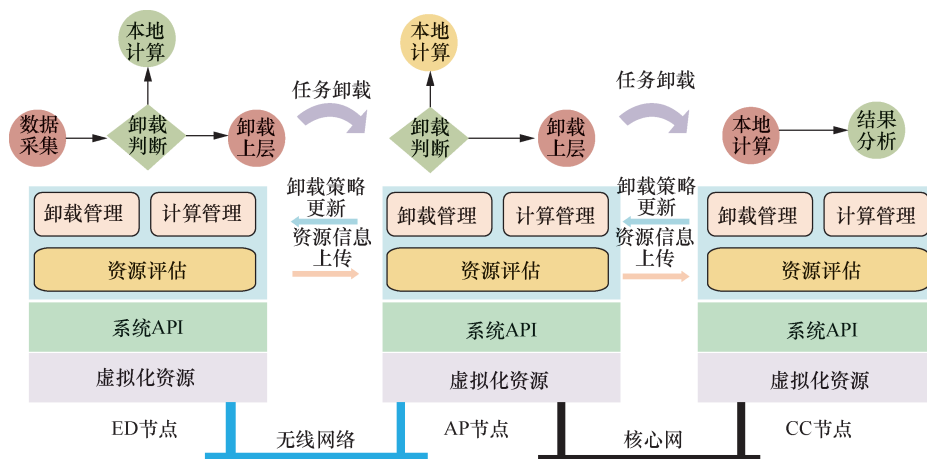


图 4 EdgeFlow 任务卸载过程

卸载策略即以 3 个任务为周期，第一个和第二个数据分组在本地计算，第三个数据分组上行进行周期卸载策略。

3.1 阻塞态

当吞吐量小于任务数据产生速率时，系统将无法及时完成计算任务，数据将在系统部分节点出现堆积，可能会造成部分节点或者整个系统阻塞，即系统进入阻塞态。在阻塞情况下，节点队列是不断堆积、增长的，仅当系统调整调度策略、重新分配资源或者任务数据减小时，才可能恢复非阻塞状态，队列堆积情况才会开始减少直至归零。此时，从任务时延的角度看，任务执行时间的影响远小于任务阻塞时间的影响。所以，系统需要考虑的资源调配和任务卸载方案应减小瓶颈节点的任务负载，以最大化系统吞吐量，从而缓解甚至消除系统阻塞情况。

EdgeFlow 任务处理过程如图 5 所示,分析可知在阻塞状态下,系统的等待时间取决于上述提到的 5 个并行阶段中最长的时间。当数据处理阶段的时

间与最长时间相等时,说明计算吞吐量是系统的瓶颈;当数据传输阶段的时间与最长时间相等时,说明传输速度为系统的瓶颈。因此在阻塞状态下,系统任务卸载策略的目的是最小化上述 5 个耗费阶段中的最长时间,具体表达式为

$$\min \max \{T_{\text{ED}}, T_{\text{AP}}, T_{\text{CC}}, t_{\text{ED}}, t_{\text{AP}}\} \quad (1)$$

由上述可知,阻塞态任务卸载方案可以分为 3 步。1) 在 ED 上卸载,根据 ED 上通信资源和计算资源的使用情况重新分配卸载任务;2) 在 AP 上卸载,根据 AP 上通信资源和计算资源的使用情况重新分配卸载任务;3) 将全部剩余任务交到 CC,完成计算任务。

3.2 非阻塞态

当系统的吞吐量大于系统的数据产生速率时,说明系统理论上能处理这些数据,则并行计算流程不会发生阻塞,即系统处于非阻塞态,此时系统优化目标为如何通过最佳的任务卸载策略和资源配置方案以最小化任务执行时间。最优化系统执行时延的核心思想在于使用优先资源处理任务、使用优先资源原则,

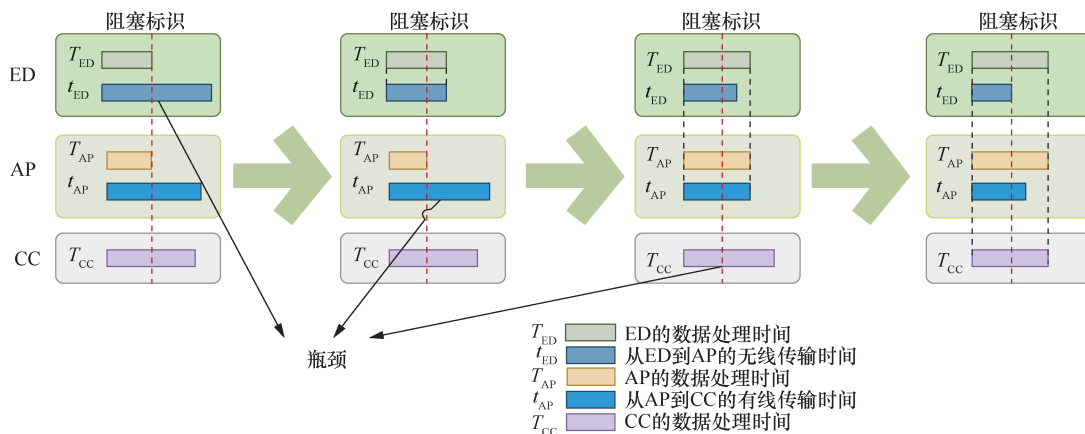


图 5 EdgeFlow 任务处理过程

系统节点在当前系统条件下有其优先资源,所谓优先资源是指边缘节点本地计算资源优先代表本地计算时延小于卸载任务到上层节点后的任务时延,而本地通信资源优先代表本地计算时延大于卸载任务到上层节点后的任务时延,以保证任务执行时延最小。

设 λ 表示进入该节点的原始数据量, s 表示原始数据在该节点的计算比例, ρ 表示计算后数据的压缩比例, θ 和 ϕ 分别表示在单位时间内该节点的计算能力和传输能力。对于 CC 层,没有传输任务,不发生阻塞的条件只需要计算资源不发生阻塞。EdgeFlow 系统根据任务卸载策略 s 、计算资源 θ 和通信资源 ϕ 判断系统的设备节点是否发生阻塞。当数据量为 λ 的数据传入 ED 时,系统可以分别估计任务在 ED 计算到数据结果传递到 CC 的任务执行时间 L_{ED} 、任务在 AP 计算的任务执行时间 L_{AP} 和任务在 CC 执行的任务执行时间 L_{CC} 。

$$L_{ED} = \frac{\lambda}{\theta_{ED}} + \frac{\rho\lambda}{\phi_{ED}} + \frac{\rho\lambda}{\phi_{AP}} \quad (2)$$

$$L_{AP} = \frac{\lambda}{\phi_{ED}} + \frac{\lambda}{\theta_{AP}} + \frac{\rho\lambda}{\phi_{AP}} \quad (3)$$

$$L_{CC} = \frac{\lambda}{\phi_{ED}} + \frac{\lambda}{\phi_{AP}} + \frac{\lambda}{\theta_{CC}} \quad (4)$$

最短时延表达式如式(5)所示

$$\min\{s_{ED} \cdot L_{ED} + s_{AP} \cdot L_{AP} + s_{CC} \cdot L_{CC}\} \quad (5)$$

由此可知,在非阻塞态的策略为结合具体的任务情况,部分节点优先考虑通信资源,部分节点优先考虑计算资源,即可获得最优解。

4 实验过程

实验实物如图 6 所示,为了基于系统实验分析 EdgeFlow 的系统性能,本实验模拟搭建无线监控场景,ED 收集图像数据,系统根据收集的图像信息进行人脸识别,并从原始图像中切割有效人脸部分传输到 CC。CC 根据获得的人脸数据进行分析,并执行相应处理。在该任务中,人脸数据识别功能基于 OpenCV 视频开源库实现,OpenCV 是一个开源编程功能库,主要针对实时计算机视觉。本系统利用 OpenCV 的人脸识别功能,通过任务识别接口实现从图像中检索人脸并抽取相关人脸图像数据发送到云端。

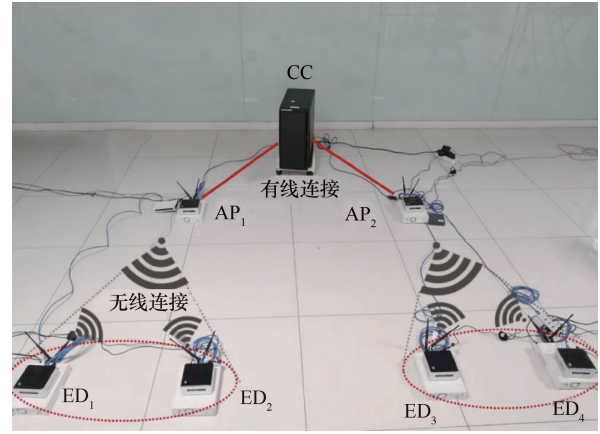


图 6 实验实物

本文选择了 5 种对比场景,对 EdgeFlow 方案的性能进行综合分析。在云计算场景中,ED 节点将任务数据流直接发送给 CC,任务计算工作在 CC 完成;在本地计算场景中,ED 节点会在本地计算处理任务数据,并将结果数据汇总到云端;在 Cloudlet 场景中,ED 将计算任务卸载到相邻的 Cloudlet 服务器,Cloudlet 服务器就近部署在小区的 AP 中。当 Cloudlet 服务器完成计算任务时,则将数据结果汇总到 CC。为对比本文任务卸载算法与现有任务卸载工作之间的异同,以两项最新的任务卸载策略工作作为分析对象。Liu 等^[10]提出基于 Markov 模型的任务卸载问题以优化任务时延,算法简称为“LMARKOV”。Tang 等^[9]提出基于博弈理论的多用户场景下的任务选择卸载策略,解决如何选择部分任务完全卸载的问题以最优优化任务时延,算法简称为“TGAME”。

5 实验结果

在本实验中,研究系统任务时延受任务数据到达速率的影响。数据到达速率对平均任务时延的影响如图 7 所示,EdgeFlow 在大多数情况下表现最好。系统实验结果和模拟仿真场景中的结果基本相符,而在实际系统中,由于系统任务到达时间等不确定性和额外数据打包等消耗,导致实际系统的任务时延场景相对于理论模拟场景有一定的额外支出。本文提出的多层网络任务卸载方案在低阻塞情况下,保证系统选择优先资源使得系统任务时延最小,并保证系统非阻塞。当任务到达速率增大时,通过合理调度计算资源和通信资源,实现多层节点之间的负载均衡,尽可能降低系统阻塞,保证在阻塞条件下的系统任务时延最小。

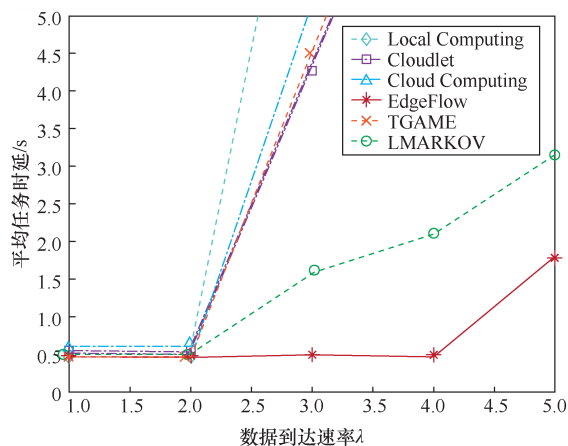


图7 数据到达速率对平均任务时延的影响

图8为不同数据到达速率对系统处理速度的影响,从图8可以看出,在低数据到达速率情况下,由于资源出现空闲,因此系统处理速度无法达到模拟实验中系统的计算吞吐量水平。在该场景下,EdgeFlow的系统处理速度相较于其他方案的任务处理速度优势不明显,由于系统资源空闲,EdgeFlow系统通过牺牲部分系统吞吐量、选择优先资源以使得任务时延最优化。在高数据到达速率情况下,即 $\lambda \geq 3$ 时,其他方案由于系统受单一节点的资源瓶颈影响,进入阻塞状态,任务处理达到上限。EdgeFlow通过动态调整计算资源和通信资源的平衡,减小了瓶颈节点的资源负载,提高了系统吞吐量,尽量保证系统不发生阻塞。

数据到达速率对系统阻塞队列的影响如图9所示,阻塞队列由系统中所有节点的等待数据分组相加得到,在该实验中代表系统执行400个处理任务后,系统阻塞最严重情况时的阻塞队列大小。在实际场景中,本文可以发现AP的通信总带宽约为300 kbit/s,实验中的单任务分组文件大小约为61 kbit,当任务到达速率达到每秒5个数据分组时,系统资源已无法保证系统的非阻塞运行。如图9所示,在 $\lambda=5$ 的场景下,相较于其他方案,EdgeFlow通过平衡多层网络节点的通信资源和计算资源消耗、多层节点分担系统的任务负载,在降低系统瓶颈节点负载的同时,提高了系统的计算吞吐量,减少了系统中阻塞数据分组的堆积,因此数据分组的阻塞情况最轻。所以,EdgeFlow方案可以实现基于任务条件和系统资源条件的可调度任务卸载策略,尽可能地保证系统稳定工作。

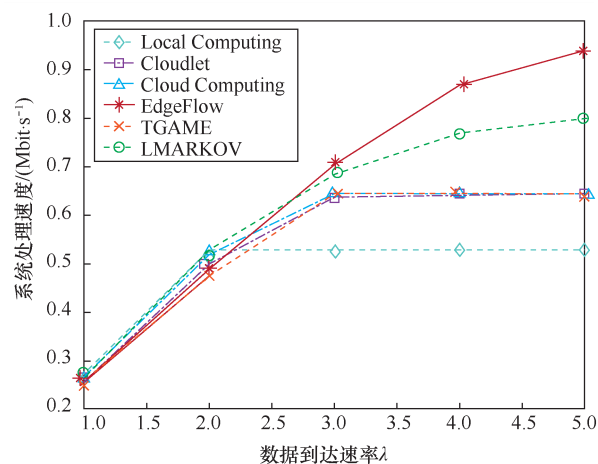


图8 不同数据到达速率对系统处理速度的影响

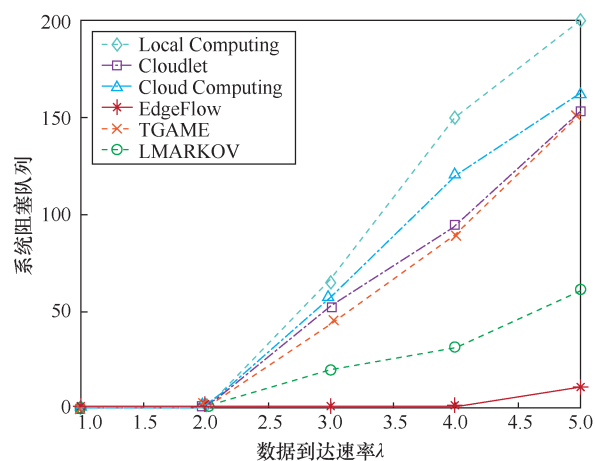


图9 数据到达速率对系统阻塞队列的影响

6 结束语

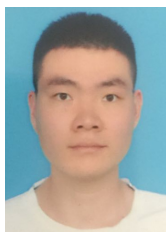
本文在MEC框架下,基于流式数据任务提出边缘计算节点和云计算中心协同工作的边缘计算框架EdgeFlow,并构建了相应的卸载问题使得任务时延实现最优化。实验表明,EdgeFlow显著降低了平均任务时延,提升了系统处理速度,缩短了系统等待队列。EdgeFlow为应对由物联网普及导致的海量计算任务与低时延问题提供了新的思路,奠定了理论与实践基础,可以广泛地应用于人脸识别、智慧城市等物联网场景中。随着该框架的进一步完善与优化,高速率、低时延计算将不再是问题,将直接推动物联网飞速发展,使生活更加轻松、便捷。

参考文献:

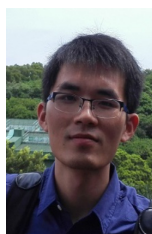
- [1] MUHAMMAD A N, MUHAMMAD A S, IMRAN A K. A survey on current repertoire for 5G[J]. Journal of Industrial Information Integra-

- tion, 2017, 40(2): 18-29.
- [2] MOHAMMAD A R, MARIJA M J, ANDREI P, et al. Middleware for Internet of things: a survey[J]. IEEE Internet of Things Journal, 2017, 3(1): 70-95.
- [3] JEFFREY B. Cisco: SDN, collaboration, Internet of things to drive future trends[J]. eWEEK, 2013(12): 5.
- [4] EVANS D. The Internet of things: how the next evolution of the Internet is changing everything[J]. CISCO White Paper, 2011(1): 11.
- [5] ALEKSANDRA C, HENRIK L C, YING Y, et al. Cloud RAN for mobile networks—a technology overview[J]. IEEE Communications Surveys and Tutorials, 2015, 17(1): 405-426.
- [6] WANG S, ZHANG X, ZHANG Y, et al. A survey on mobile edge networks: convergence of computing, caching and communications[J]. IEEE Access, 2017: 1.
- [7] YI S H, LI C, LI Q. A survey of fog computing[C]//The 2015 Workshop on Mobile Big Data. ACM, 2015: 9-16.
- [8] SHI W S, CAO J, ZHANG Q, et al. Edge computing: vision and challenges[J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646.
- [9] TANG L, HE S B. Multi-user computation offloading in mobile edge computing: a behavioral perspective[J]. IEEE Network, 2018, 32(1): 48-53.
- [10] LIU J, MAO Y Y, ZHANG J, et al. Delay-optimal computation task scheduling for mobile-edge computing systems[C]//IEEE International Conference on Distributed Computing Systems. IEEE, 2016: 1451-1455.

[作者简介]



丛书畅 (1995—)，男，黑龙江齐齐哈尔人，北京大学现代通信研究所硕士生，主要研究方向为无线通信、边缘计算、数据挖掘等。



姚超 (1993—)，男，浙江衢州人，北京大学现代通信研究所硕士生，主要研究方向为边缘计算、分布式计算等。



王鹏飞 (1994—)，男，河北保定人，北京大学现代通信研究所硕士生，主要研究方向为无线通信、边缘计算和车载通信。



郑子杰 (1991—)，男，山东济南人，北京大学现代通信研究所博士生，主要研究方向为信息与信号处理、无线通信中的大数据算法等。



宋令阳 (1979—)，男，辽宁抚顺人，北京大学现代通信研究所教授，主要研究方向为无线通信、编码技术、MIMO、OFDMA、认知无线电以及协同通信等。