

基于博弈论的多服务器多用户视频分析任务卸载算法*

陈 彧¹, 张 胜¹, 金熠波¹, 钱柱中¹, 陆桑璐¹

¹(南京大学 计算机软件新技术国家重点实验室, 江苏 南京 210023)

通讯作者: 张胜, E-mail: sheng@nju.edu.cn

摘要: 在过去的近十年中, 人工智能相关的服务和应用大规模出现, 它们要求高算力、高带宽和低时延. 边缘计算目前被认为是这些应用最适合的计算模式, 尤其是视频分析相关应用. 本文研究了多服务器多用户异构视频分析任务卸载问题, 其中用户选择合适的边缘服务器, 并将他们的原始视频数据上传至服务器进行视频分析. 为了有效处理众多用户对有限网络资源的竞争和共享, 并且能够获得稳定的网络资源分配局面, 即每个用户不会单方面地改变自己的任务卸载决策, 该多服务器多用户异构视频分析任务卸载问题被建模为一个多玩家的博弈问题. 基于最小化整体时延的优化目标, 先后研究了非分布式视频分析场景和分布式视频分析场景两种情形, 分别提出了基于博弈论的潜在最优服务器选择算法和视频单元分配算法. 通过严格的数学证明, 两种情形下提出的算法均可以达到纳什均衡, 同时保证较低的整体时延. 最后, 基于真实数据集的大量实验表明, 本文提出的方法比其他现有算法降低了平均 26.3% 的整体时延.

关键词: 博弈论; 纳什均衡; 任务卸载; 视频分析; 边缘计算

中图法分类号: TP311

中文引用格式: 陈彧, 张胜, 金熠波, 钱柱中, 陆桑璐. 基于博弈论的多服务器多用户视频分析任务卸载算法. 软件学报. <http://www.jos.org.cn/1000-9825/0000.htm>

英文引用格式: Chen Y, Zhang S, Jin YB, Qian ZZ, Lu SL. Game Theory-based Algorithm for Multi-server Multi-user Video Analytics Task Offloading. Ruan Jian Xue Bao/Journal of Software, 2016 (in Chinese). <http://www.jos.org.cn/1000-9825/0000.htm>

Game Theory-based Algorithm for Multi-server Multi-user Video Analytics Task Offloading

CHEN Yu¹, ZHANG Sheng¹, JIN Yi-Bo¹, QIAN Zhu-Zhong¹, LU Sang-Lu¹

¹(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China)

Abstract: In past years, artificial intelligence related services and applications have boomed, which require high computation, high bandwidth and low latency. Edge computing is regarded as an appropriate solution for them, especially video analytics. This paper investigates a multi-server multi-user heterogeneous video analytics offloading problem, where users select appropriate edge servers and then offload their raw video data to the servers for essential analytics. To deal with the cooperation and conflicts among users and get a stable network allocation situation where each user has no incentive to change the offloading decision unilaterally, the multi-server multi-user video analytics offloading problem is formulated as a multiplayer game. Based on the goal of minimizing the overall delay, the game theory-based potential optimal server selection and video unit allocation algorithms are proposed for the non-distributed and distributed scenarios, respectively. Via rigorous proof, Nash equilibrium can be reached in both of the two cases, and low overall delay is guaranteed. Finally, extensive trace-driven experiments show that our method improves the overall delay by 26.3% on average, compared with other algorithms.

Key words: game theory; Nash equilibrium; task offloading; video analytics; edge computing

* 基金项目: 国家自然科学基金(61872175, 61832008)

Foundation item: National Natural Science Foundation of China (61872175, 61832008)

收稿时间: 0000-00-00; 修改时间: 0000-00-00; 采用时间: 0000-00-00; jos 在线出版时间: 0000-00-00

CNKI 在线出版时间: 0000-00-00

随着智能设备(如智能手机、平板电脑和物联网设备)和大量新应用的出现,网络流量正在迅速增长.由于回程链路传输时延高、流量负载大,传统的集中式网络架构无法满足用户的需求.边缘计算^[1-4]是一种新兴的范式,它将存储和计算资源带到网络的边缘,比如基站或访问接入点.边缘计算允许终端设备产生的数据在网络边缘进行处理,而不是沿着长距离将其发送到云端或数据中心.在过去的近十年里,研究了许多与边缘计算相关的工作^[5,6],包括最优网络控制、多用户资源分配、服务缓存等.

同时,基于深度学习的人工智能(Artificial Intelligence,简称 AI)服务和应用在近些年蓬勃发展,比如视频监控、推荐系统和个人助理^[7,8].与边缘计算环境中的其他服务类似,用户可以将 AI 应用任务卸载到网络边缘进行计算.微软曾研究哪些具体的 AI 应用任务更适合移动到网络边缘^[9],包括交互式云游戏、语音命令识别、虚拟现实/增强现实、实时视频分析等等,其中的实时视频分析应用被认为是边缘计算的杀手级应用^[9,10,11].大多数运行在边缘服务器上的视频分析应用程序通过处理视频数据以检测一些特定对象,比如失踪儿童、遗弃行李、逃逸车辆等.通常来说,视频分析应用任务需要收集大量高清视频,并且保证要求高算力、高带宽和低时延.因此,边缘计算被认为是满足这些要求的合适解决方案.

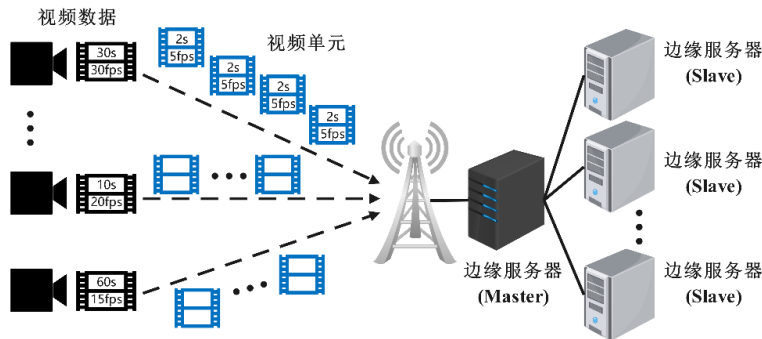


Fig.1 Multi-server multi-user video analytics task offloading

图 1 多服务器多用户视频分析任务卸载

在边缘计算环境中,部署着很多具有不同配置(如计算能力、内存)的边缘服务器,这些服务器分布在不同的物理位置,因而它们的数据传输时延也均不相同.大量用户将他们的视频数据传送到这些边缘服务器上进行分析,如图 1 所示.一些视频数据可以根据长度和配置(即帧率和分辨率)划分为更小的视频单元^[12,13],然后通过视频分析服务对它们分别进行分析.对用户而言,选择合适的边缘服务器并将他们的视频数据(或视频单元)合理地分配至不同服务器上具有一定的挑战性.边缘服务器的计算能力和传输带宽有限,不正确的卸载决策可能会导致部分边缘服务器过载或造成 CPU 周期的浪费.例如,如果将过多的视频数据分配至同一台边缘服务器,那么处理时间由于该服务器过载而显著延长;相反地,如果分配给边缘服务器的视频数据太少,则会导致该服务器的计算资源利用率低.此外,所有用户的视频数据具有异构性,即视频长度和配置各不相同,这使得多服务器多用户视频分析任务卸载问题更具挑战性.

对于边缘计算任务卸载问题,多数现有工作^[14,15,16]首先收集众多用户的任务请求,然后根据系统的统一决策确定每个用户的任务请求应该卸载到哪个服务器.但是该类方法做卸载决策前需要收集大量的用户信息,当在较大规模的分布式环境下,随着用户的信息量增大,该类方法无法取得较好的运算效率.因此,为了缓解上述较大规模分布式环境下出现的问题,可以考虑由用户分别选择服务器并卸载其任务.对用户来说,目标是 minimized 整体的计算和传输时延.每个用户需要确定将视频数据卸载到哪些边缘服务器,以及如何在边缘服务器之间分配视频单元.通常,边缘网络资源是有限的,众多用户对仅有的资源进行竞争和共享,从而尽可能达到最小化计算和传输时延的目标.对此可以使用博弈论(Game Theory,简称 GT)的方法来分析多个独立且自利的玩家之间的互动^[17,18],从而获得一个稳定的系统,其中没有玩家有单方面偏离的动机^[19,20].然而,现有的基于博弈论方法的工作^[21,22]很少结合视频分析应用的特点,没有考虑在边缘网络进行视频分析任务的划分、卸载的博弈问题.此外,也

没有相关工作^[23,24]全面地考虑非分布式和分布式视频分析场景下的博弈问题,并结合不同场景特点给出有性能保证的求解。

本文研究了多服务器多用户异构视频分析任务卸载问题,其中用户选择合适的边缘服务器并将他们的视频数据卸载到这些服务器进行视频分析.首先介绍了基于长度和帧率的将所有用户的异构视频数据划分为更小单元的方法,其次建立了计算和通信模型,并表示了用户将视频单元卸载到选定服务器进行计算的整体时延.为了达到最小化整体时延并获得稳定的网络资源分配局面,即没有用户有动机单方面改变其卸载决策,多服务器多用户异构视频分析任务卸载问题被建模为一个多人博弈问题.为了解决该博弈问题并达到纳什均衡(Nash Equilibrium,简称 NE),先研究了每个用户将其所有视频单元仅卸载到其选择的单独边缘服务器的非分布式情况,提出了基于 GT 的潜在最优服务器选择算法;之后研究了每个用户的视频单元可以卸载到多个服务器的分布式场景,基于少数者博弈的关键概念截断值,提出了基于 GT 的视频单元分配算法.通过严格的理论证明,分析了提出的算法接近最优的性能并证明其可以达到纳什均衡.此外,使用来自 AI City Datasets 2019^[25]的视频数据集和物体检测器 YOLOv3^[26]进行实验,验证了算法的有效性和高效性。

本文的创新贡献点包括:

- 结合视频分析应用的特点,考虑在边缘网络进行视频分析任务的划分和卸载,并建模为博弈问题.
- 全面地考虑了非分布式和分布式视频分析场景下的博弈问题,并结合不同场景的特点分别给出了有性能保证的求解.
- 基于真实世界数据的大量实验验证了在不同实验设置场景下,提出的基于博弈论的方法能取得相较于对比算法更好的效果,整体时延平均降低了 26.3%.

本文第 1 节总结相关工作进展.第 2 节给出计算和通信模型,并建模多人博弈问题.第 3 节研究非分布式视频分析任务卸载问题.第 4 节研究分布式视频分析任务卸载问题.第 5 节实验验证算法的有效性和高效性.第 6 节和第 7 节提出未来工作和结论.

1 相关工作

本文主要研究基于博弈论的多服务器多用户视频分析任务卸载问题,在本节中对边缘网络的任务卸载、基于边缘计算的视频分析和基于博弈论的边缘网络资源分配三个方面的一些工作进行总结.

边缘网络的任务卸载:Gao 等人^[27]为启用深度神经网络任务的移动边缘计算网络开发了任务分区和卸载的联合设计,该网络由单个服务器和多个移动设备组成.Yan 等人^[28]提出了在任务依赖模型下最小化能量消耗和任务执行时间的加权和最优化任务卸载策略和资源分配方法.Wu 等人^[29]研究了边缘计算和云计算合作实现安全任务卸载的区块链场景.Liu 等人^[30]充分考虑多用户请求对服务器资源的竞争关系,提出了一种面向多用户的串行任务动态卸载策略,解决边缘服务器资源受限下的任务卸载问题.Wu 等人^[31]提出一个基于停车协同的边缘计算任务卸载框架,解决网络边缘服务器超负荷的问题.这些工作侧重于边缘的任务卸载问题,但没有考虑视频分析应用程序.

基于边缘计算的视频分析:NoScope^[32]是基于差异检测子设计的,该检测子突出显示跨帧的时间差异,以加快边缘计算环境中的视频分析.Vantage^[33]作为一种直播上传解决方案被提出,它使用选择性质量增强重传而不是实时帧来提高时移观众的体验质量.Ren 等人^[34]提出了一种用于边缘计算中的多用户视频压缩和卸载的算法,该算法最大限度地减少了在本地、边缘和云进行压缩的时延.Li 等人^[35]依托移动边缘计算优势,提出基于神经网络的车载视频采集终端车牌号码识别算法.这些工作研究了边缘网络环境的视频分析配置适应和任务卸载,然而几乎没有工作应用基于博弈论的方法来保证一个稳定的网络资源分配局面.

基于博弈论的边缘网络资源分配:Li 等人^[18]提出了一种基于博弈论的算法,联合考虑卸载决策、传输功率和移动用户的 CPU 能力进行多用户边缘计算.Hu 等人^[21]将异构任务卸载问题表述为少数者博弈问题,最终进入少数者集合的玩家获胜.Zheng 等人^[22]提出了一种性能指标来评估用户的体验质量,并将面向 QoE 的资源分配问题建模为本地合作博弈.Zhan 等人^[23]设计了一个去中心化的卸载博弈,其中每个用户决定将其任务的一部

分卸载到边缘服务器.Lei 等人^[24]针对“中心云服务器+多个边缘服务器”构成的“云+边”混合环境中多任务卸载效率和效能不足的问题,提出了一种基于概率性能感知演化博弈策略的任务卸载方法.然而,这些工作没有针对视频分析相关的应用,考虑根据视频长度或帧速率将视频数据划分为更小单元的方法,以便在边缘网络进行视频分析任务的卸载.

2 系统模型及博弈定义

本节展示了系统模型的定义,并将多服务器多用户异构视频分析任务卸载问题建模为一个多人博弈问题.一些重要的符号见表 1.

2.1 系统模型

边缘网络:在网络边缘,考虑 N 个用户,记为 $\mathbf{N} = \{1, 2, \dots, N\}$.边缘服务器集合由一个 Master 边缘服务器和 M 个 Slave 边缘服务器构成, M 个 Slave 服务器记为 $\mathbf{M} = \{1, 2, \dots, M\}$.Master 边缘服务器是边缘用户卸载任务的入口点,而 Slave 边缘服务器负责处理视频分析任务.视频分析应用(如人脸识别、车辆跟踪和行人计数)部署在边缘服务器上,用户将视频分析任务卸载至它们进行计算,如图 1 所示.

Table 1 Important Notations in System

表 1 系统的重要符号

名称	描述
N, N	边缘用户集合,边缘用户数量
M, M	边缘服务器集合,边缘服务器数量
s_n	用户 n 的视频单元数量
C_u	每个视频单元的计算要求
Cap_m	边缘服务器 m 的计算能力
x_n	用户 n 的视频卸载决策
x_{-n}	用户 n 以外的用户的视频卸载决策
τ_m	边缘服务器 m 的计算时延
D_n^{comp}	用户 n 的计算时延
D_n^{comm}	用户 n 的传输时延
S_n	玩家 n 的策略空间
U_n	玩家 n 的效用函数

视频分段:根据视频长度和配置(如帧率和分辨率),可以将一些视频数据分成更小的单元,然后通过视频分析应用程序分析每个视频单元.例如,在目标检测应用中,任务是在卸载的视频数据中找到某个目标(如丢失的钱包).视频长度为 600 秒(second,简称 s),帧速率为 30 帧每秒(frame per second,简称 fps).将视频数据分成更小的单位,其长度和帧率分别为 2s 和 5fps.之后,将目标检测子(如 YOLOv3)并行应用于 $(600s * 30fps)/(2s * 5fps) = 1800$ 个视频单元.

一般地,所有用户持有异构的待分析视频数据,假设每个用户 n 都有一个视频分析任务 $T_n = \langle l_n, f_n, r_n \rangle$,其中 l_n 、 f_n 和 r_n 分别表示长度、帧率和分辨率.在一些视频分析应用中,视频数据是从监控摄像头^[12]捕获的,它们的视频分辨率是固定的.此外,输入到用于视频分析的卷积神经网络的视频帧尺寸通常设置为常数.因此在本文中,假设所有用户的视频数据的分辨率都是相同的常数 R (即 $\forall n \in \mathbf{N}, r_n = R$),根据长度和帧率将视频数据划分成更小的单位.然后可以计算用户 n 的视频单元数为 $s_n = (l_n f_n)/(L_u F_u)$,其中 L_u 设置为所有用户视频长度的公因数, F_u 为所有用户的视频帧率的公因数.值得一提的是,对于一组视频数据,总是有可行的 L_u 和 F_u (例如,可以将 L_u 设置为 1s,并将 F_u 设置为 1fps).因此,用户 n 需要将 s_n 个视频单元卸载到边缘服务器进行计算.

任务卸载:当用户 n 将其视频单元卸载到边缘服务器进行计算时,用户 n 的任务卸载决策表示为 $x_n = [x_{n,1}, x_{n,2}, \dots, x_{n,M}]$,其中 $x_{n,m}$ 表示用户 n 卸载到边缘服务器 m 的视频单元数量.对于每个服务器 $m \in \mathbf{M}$, $x_{n,m}$ 是一个非负整数,并且有 $\sum_{m \in \mathbf{M}} x_{n,m} = s_n$.

计算模型:与之前的研究工作^[12,13]类似,视频可以被划分为一组更小的单元,每个视频单元的计算需求表示为 C_u [CPU 周期].此外,可以使用虚拟并行处理^[36]来支持多个任务的处理.因此,边缘服务器 m 上的计算时延可以

计算为

$$\tau_m = \frac{\sum_{n \in N} x_{n,m} C_u}{Cap_m} \quad (1)$$

其中 Cap_m [CPU 周期每秒] 表示边缘服务器 m 的计算能力. 因此, 可以将用户 n 的整体计算时延表示为

$$D_n^{comp} = \max_{m \in M: x_{n,m} \neq 0} \tau_m \quad (2)$$

从上面的等式中观察到, 当用户 n 将其所有视频单元卸载到某一个边缘服务器 m 时, 整体计算时延仅仅取决于该服务器上的计算时延. 然而, 当视频单元被卸载到多个边缘服务器时, 整体计算时延取决于这些服务器的计算时延中最大的. 因此, 用户需要做出卸载决策, 并且根据其他用户的决策来调整他们的决策, 以最小化整体的计算时延, 具有一定的挑战性.

通信模型: 根据香农公式^[18], 计算用户 n 卸载视频数据到 Master 边缘服务器的速率 $r_{n,0}$ [比特每秒] 为

$$r_{n,0} = W \log_2 \left(1 + \frac{P_n H_{n,0}}{\sigma^2} \right) \quad (3)$$

其中 W 代表信道带宽, P_n 代表用户 n 的发射功率, 可以通过相关功率控制算法^[32] 来确定. 此外, $H_{n,0}$ 表示用户 n 与其连接的 Master 边缘服务器之间的信道增益, σ^2 表示背景噪声的方差.

与许多研究工作^[32-34] 类似, 边缘服务器返回计算结果的时间开销被忽略, 因为在许多视频分析应用程序 (如行人计数) 中, 计算结果数据大小远小于视频数据大小. 因此, 当用户 n 将 s_n 个视频单元通过 Master 服务器卸载到合适的 Slave 服务器时, 传输时延计算为

$$D_n^{comm} = \frac{s_n D_u}{r_{n,0}} \quad (4)$$

其中 D_u [比特] 是每个视频单元的数据大小.

2.2 博弈定义

将多服务器多用户异构视频分析任务卸载问题建模为多人博弈问题, 记为 $\mathbf{G} = \langle \mathbf{N}, \mathbf{S}, \mathbf{U} \rangle$, 其中用户集合 \mathbf{N} 被视为博弈玩家集合. 所有博弈玩家的策略空间集合表示为 $\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_N\}$, 其中 \mathbf{S}_n 表示玩家 n 的策略空间, 是用户 n 可以选择的所有策略的并集. 策略空间 \mathbf{S}_n 中的策略个数也称为策略空间的势, 取决于视频单元个数 s_n . 任务卸载决策 $\mathbf{x}_n = [x_{n,1}, x_{n,2}, \dots, x_{n,M}]$ 被视为每个玩家 n 的策略, 并且策略中每个方向上的量表示卸载到对应 Slave 边缘服务器的视频单元数量. 用 $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ 表示所有玩家的任务卸载策略, $\mathbf{x}_{-n} = (\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, \mathbf{x}_{n+1}, \dots, \mathbf{x}_N)$ 表示除玩家 n 之外所有其他玩家的卸载策略. 所有玩家的效用函数集合表示为 $\mathbf{U} = \{U_n(\mathbf{x}_n, \mathbf{x}_{-n})\}_{n \in \mathbf{N}}$. 基于计算和传输的整体时延, 博弈模型 \mathbf{G} 中每个博弈玩家 n 的效用函数构建为

$$U_n(\mathbf{x}_n, \mathbf{x}_{-n}) = 1 / (D_n^{comp} + \mathbb{E}_n[D_n^{comm}]) = 1 / \max_{m \in M: x_{n,m} \neq 0} \{\tau_m + \varphi\} \quad (5)$$

其中使用 $\varphi = \mathbb{E}_n[D_n^{comm}]$ 表示所有博弈玩家的期望传输时延^[32]. 对于每个博弈玩家 n 来说, 最大化其效用函数值是赢得博弈的最佳策略, 即

$$\begin{aligned} & \max U_n(\mathbf{x}_n, \mathbf{x}_{-n}) \\ & \text{s.t. } (1), (2), (3), (4), (5) \end{aligned}$$

值得注意的是, 如果每个用户最大化它的效用函数值, 该用户整体的计算和通信时延也同时被最小化.

然而, 由于每个博弈玩家的收益不仅仅取决于自己的决策, 也同时取决于其他玩家的策略, 因此玩家决策策略是具有挑战性的, 目标是设计算法以实现定义如下的纳什均衡:

定义 1 (Nash 均衡): 对于每个博弈玩家 n , 策略集 $\mathbf{x}^* = (\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_N^*)$ 在博弈模型 \mathbf{G} 中构成纳什均衡, 当且仅当任意博弈玩家不能通过单方面改变自己的任务卸载策略来提升自己的效用函数值, 即

$$\forall n \in \mathbf{N}, \forall \mathbf{x}_n \in \mathbf{S}_n, U_n(\mathbf{x}_n^*, \mathbf{x}_{-n}^*) \geq U_n(\mathbf{x}_n, \mathbf{x}_{-n}^*) \quad (6)$$

纳什均衡具有自稳定性, 使得处于均衡状态的用户可以获得各方都满意的决策方案. 通过实现纳什均衡, 可以有

效处理众多用户对有限网络资源的竞争和共享,并且能够获得一个稳定的网络资源分配局面,即每个用户都没有单方面改变其卸载决策的动机.

3 非分布式视频分析场景的任务卸载

本节首先研究一种特殊情况,即每个用户会选择一个边缘服务器,并将其所有视频单元仅卸载到自己选择的边缘服务器.在之后的第 4 节中,将问题扩展到一般情况,即每个用户可以将其视频单元卸载到多个边缘服务器.

在一些视频分析应用中,视频分析需要所有视频单元的全局信息,单独处理每个视频单元会影响分析结果的准确性.例如,在行人计数的应用中,包含相同行人目标的视频帧可能分布在不同的视频单元中.如果它们被卸载到多个边缘服务器,不同服务器之间的重复计算可能会导致行人数量的错误计数,所以应将所有视频单元卸载到同一服务器进行分析.因此,研究非分布式场景中的视频分析任务卸载是必要且有意义的,其中每个用户将其所有视频单元仅卸载到一个边缘服务器.在研究该非分布式视频分析场景之前,先提出两个概念:包和潜在最优服务器.

定义 2 (包):当每个用户将其所有视频单元卸载到它选择的单个边缘服务器时,将用户 n 的所有视频单元集合称为包 p_n .包 p_n 的大小定义为 s_n ,即用户 n 的视频单元数.

定义 3 (潜在最优服务器):在包 p_n 被卸载到某个边缘服务器之前,计算每个服务器 m 的潜在时延 $\Gamma_{n,m}$ 为

$$\Gamma_{n,m} = \frac{\lambda_m C_u}{Cap_m} + \varphi + \frac{s_n C_u}{Cap_m} \quad (7)$$

其中 λ_m 表示当前在服务器 m 上的视频单元数, s_n 是包 p_n 的大小.在式(7)中, $\frac{\lambda_m C_u}{Cap_m} + \varphi$ 部分表示服务器 m 的当前时延,它取决于 λ_m .此外, $\frac{s_n C_u}{Cap_m}$ 表示服务器 m 的潜在增加时延,它取决于 s_n .对于包 p_n ,边缘服务器 m 是潜在最优服务器,当且仅当服务器 m 的潜在时延 $\Gamma_{n,m}$ 在所有服务器中最小,即

$$m = \arg \min_{j \in M} \Gamma_{n,j} \quad (8)$$

注意到,如果多个服务器满足同时等式(8),可以选择下标 m 最小的服务器.

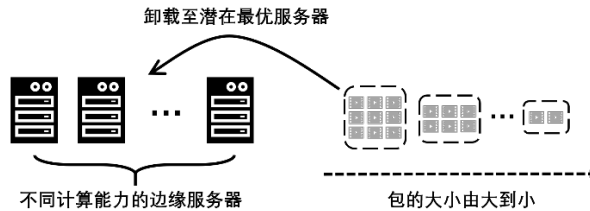


Fig.2 Offloading all packages one by one to the potential optimal servers in order of the size from large to small

图 2 按照包的大小从大到小顺序依次卸载到潜在最优服务器

基于以上的概念,有以下观察:

定理 1:按照包的大小从大到小顺序,将所有用户的包依次卸载到其对应的潜在最优服务器.当所有包卸载完毕后,达到了纳什均衡,即没有用户有单方面改变其卸载决策的动机.

证明:

用数学归纳法来证明.按照包大小从大到小的顺序,首先将所有包排序为 $L = [p_{\mu_1}, p_{\mu_2}, \dots, p_{\mu_N}]$,其中 p_{μ_i} 表示第 i 大的包,它是用户 μ_i 的包.令 m_i^* 表示 p_{μ_i} 的潜在最优服务器.令变量 ω 表示当前已卸载的包数量,由于所有包是被一个个地按照包大小顺序卸载的, ω 的值会随着每个包的卸载而不断更新.令表述 $P(\omega)$ 表示:当 ω 个包已被卸载到各自的潜在最优服务器,没有用户有单方面改变其卸载决策的动机.接下来通过对 ω 的归纳来证明 **定理 1**.

基础步骤:当 $\omega = 1$,最大的包 p_{μ_1} 被卸载到潜在最优服务器 m_1^* .根据定义 3,对于每个 $m \in M \setminus \{m_1^*\}$,满足

$$\frac{s_{\mu_1} C_u}{Cap_{m_1^*}} + \varphi m_1^* \leq \frac{s_{\mu_1} C_u}{Cap_m} + \varphi m \quad (9)$$

因此,包 p_{μ_1} 对应的用户 μ_1 单方面没有动机改变其任务卸载决策, $P(1)$ 得到验证.

归纳步骤:接下来证明,对于任何正整数 $k < N$,如果 $P(1), P(2), \dots$ 和 $P(k)$ 成立,那么 $P(k+1)$ 也成立,即当 $k+1$ 个包已经被卸载到各自的潜在最优服务器后,任何用户都没有动机单方面地改变其卸载决策.令 $\lambda_{m,k+1}$ 表示在包 $p_{\mu_{k+1}}$ 被卸载之前服务器 m 上的视频单元数量.将在用户 μ_{k+1} 之前卸载包的用户分成 2 个集合: $\mathbf{A} = \{\mu_i | m_i^* = m_{k+1}^*\}$ 和 $\mathbf{B} = \{\mu_i | m_i^* \neq m_{k+1}^*\}$,集合 \mathbf{A} 中用户的包对应的潜在最优服务器和用户 μ_{k+1} 相同,而集合 \mathbf{B} 中用户的包对应的潜在最优服务器和用户 μ_{k+1} 不同.接下来分别证明这两个集合中的用户满足 $P(k+1)$.

对于集合 \mathbf{A} 中的任意用户 μ_i :因为用户 μ_{k+1} 选择潜在最优服务器 m_{k+1}^* ,根据最优服务器的定义,满足

$$\forall m \in M, \frac{\lambda_m C_u}{Cap_m} + \varphi + \frac{s_{\mu_{k+1}} C_u}{Cap_m} \geq \frac{\lambda_{m_{k+1}^*} C_u}{Cap_{m_{k+1}^*}} + \varphi + \frac{s_{\mu_{k+1}} C_u}{Cap_{m_{k+1}^*}} \quad (10)$$

同时,因为所有用户的包按照大小从大到小顺序依次卸载到其对应的潜在最优服务器,所以满足

$$\forall m \in M, \forall \mu_i \in \mathbf{A}, \frac{\lambda_m C_u}{Cap_m} + \varphi + \frac{s_{\mu_i} C_u}{Cap_m} \geq \frac{\lambda_m C_u}{Cap_m} + \varphi + \frac{s_{\mu_{k+1}} C_u}{Cap_m} \quad (11)$$

综上有

$$\forall m \in M, \forall \mu_i \in \mathbf{A}, \frac{\lambda_m C_u}{Cap_m} + \varphi + \frac{s_{\mu_i} C_u}{Cap_m} \geq \frac{\lambda_{m_{k+1}^*} C_u}{Cap_{m_{k+1}^*}} + \varphi + \frac{s_{\mu_{k+1}} C_u}{Cap_{m_{k+1}^*}} \quad (12)$$

即,集合 \mathbf{A} 中的用户如果单方面改变自己的决策会使得自己的效用函数值降低,因此他们没有动机单方面改变自己的决策.

对于集合 \mathbf{B} 中的任意用户 μ_i :为了便于证明,将集合 \mathbf{B} 进一步分为集合 $\mathbf{B}_1 = \{\mu_i | \exists m \in M, \mu_i \text{ 是在用户 } \mu_{k+1} \text{ 之前卸载包的用户中,最后一个向边缘服务器 } m \text{ 卸载包的用户}\}$ 和集合 $\mathbf{B}_2 = \mathbf{B} \setminus \mathbf{B}_1$, \mathbf{B}_2 是 \mathbf{B}_1 在 \mathbf{B} 中的补集.对于集合 \mathbf{B}_1 的用户 μ_i 而言:因为用户 μ_i 选择潜在最优服务器 m_i^* ,且在用户 μ_i 卸载了包之后其他服务器上可能会接收新的包,所以有当 $\forall m \in M \setminus \{m_{k+1}^*\}$ 时,满足

$$\forall \mu_i \in \mathbf{B}_1, \frac{\lambda_m C_u}{Cap_m} + \varphi + \frac{s_{\mu_i} C_u}{Cap_m} \geq \frac{\lambda_{m_i^*} C_u}{Cap_{m_i^*}} + \varphi \quad (13)$$

或当 $m = m_{k+1}^*$ 时,满足

$$\forall \mu_i \in \mathbf{B}_1, \frac{\lambda_m C_u}{Cap_m} + \varphi + \frac{s_{\mu_{k+1}} C_u}{Cap_m} + \frac{s_{\mu_i} C_u}{Cap_m} \geq \frac{\lambda_{m_i^*} C_u}{Cap_{m_i^*}} + \varphi \quad (14)$$

即,集合 \mathbf{B}_1 中的用户如果单方面改变自己的决策会使得自己的效用函数值降低,因此他们没有动机单方面改变自己的决策.对于集合 \mathbf{B}_2 的用户 μ_i' 而言:因为所有用户的包按照大小从大到小顺序依次卸载到其对应的潜在最优服务器,所以在集合 \mathbf{B}_1 中存在一个用户 μ_i' 对应的潜在最优服务器和 μ_i' 一致(均为 m_i^*),但是包大小比 μ_i' 更小.因此结合上式,当 $\forall m \in M \setminus \{m_{k+1}^*\}$ 时,满足

$$\forall \mu_i' \in \mathbf{B}_2, \frac{\lambda_m C_u}{Cap_m} + \varphi + \frac{s_{\mu_i'} C_u}{Cap_m} \geq \frac{\lambda_{m_i^*} C_u}{Cap_{m_i^*}} + \varphi \quad (15)$$

或当 $m = m_{k+1}^*$ 时,满足

$$\forall \mu_i' \in \mathbf{B}_2, \frac{\lambda_m C_u}{Cap_m} + \varphi + \frac{s_{\mu_{k+1}} C_u}{Cap_m} + \frac{s_{\mu_i'} C_u}{Cap_m} \geq \frac{\lambda_{m_i^*} C_u}{Cap_{m_i^*}} + \varphi \quad (16)$$

即,集合 \mathbf{B}_2 中的用户如果单方面改变自己的决策会使得自己的效用函数值降低,因此他们没有动机单方面改变自己的决策.综上有,当 $\forall m \in M \setminus \{m_{k+1}^*\}$ 时,满足

$$\forall m \in M, \forall \mu_i \in \mathbf{B}, \frac{\lambda_m C_u}{Cap_m} + \varphi + \frac{s_{\mu_i} C_u}{Cap_m} \geq \frac{\lambda_{m_{k+1}^*} C_u}{Cap_{m_{k+1}^*}} + \varphi + \frac{s_{\mu_{k+1}} C_u}{Cap_{m_{k+1}^*}} \quad (17)$$

或当 $m = m_{k+1}^*$ 时,满足

$$\forall \mu_i \in \mathbf{B}, \frac{\lambda_m C_u}{Cap_m} + \varphi + \frac{s_{\mu_{k+1}} C_u}{Cap_m} + \frac{s_{\mu_i} C_u}{Cap_m} \geq \frac{\lambda_{m_i^*} C_u}{Cap_{m_i^*}} + \varphi \quad (18)$$

即,集合 \mathbf{B} 中的用户如果单方面改变自己的决策会使得自己的效用函数值降低,因此他们没有动机单方面改变自己的决策。

综上,所有用户没有动机单方面改变自己的决策,否则会使得自己的效用函数值降低,即 $P(k+1)$ 成立。

结论:通过数学归纳法,当 $\omega = N$ 时,命题 $P(N)$ 也成立.因此,定理 1 被证明。

在算法 1 中,提出了一种基于 GT 的潜在最优服务器选择算法,它的原理基于定理 1,如图 2 所示.每个用户首先发布自己的信息,即包大小.通常,用户将他们的信息发送到某个指定的共享存储区域,然后他们从其他用户那里收集信息.基于收集到的信息,每个用户根据式(7)和(8)计算并记录自己的潜在最佳服务器.同时,当前视频单元数量在第 10 行更新.最后,每个用户将其包卸载到潜在最优服务器.运行算法 1 后,可以达到纳什均衡,即没有用户有单方面改变其任务卸载决策的动机。

算法 1. 基于博弈论的潜在最优服务器选择算法

输入:包 p_n 的大小 s_n ,边缘服务器 m 的算力 Cap_m ,所有用户的期望传输时延 φ

- 1: **for** each user $n \in N$ **do**
- 2: Publish its information, i.e., package size s_n ;
- 3: **for** each user $n \in N$ **do**
- 4: Collect the packet sizes of other users;
- 5: Sort packages into $L \leftarrow [p_{\mu_1}, p_{\mu_2}, \dots, p_{\mu_N}]$ in non-increasing order of size;
- 6: **for** each package $p_{\mu_i} \in L$ **do**
- 7: Calculate potential optimal server m_i^* following Eqs. (7) and (8);
- 8: **if** $\mu_i = n$ **then**
- 9: Record potential optimal server for user n ;
- 10: $\lambda_{m_i^*} \leftarrow \lambda_{m_i^*} + s_{\mu_i}$;
- 11: Offload p_n to potential optimal server for user n ;

4 分布式视频分析场景的任务卸载

在本节中,将重点关注分布式视频分析场景,即每个用户的视频单元可以卸载到多个服务器的情况.在一些视频分析应用程序中,并行分析所有视频单元可以提高分析效率,而不会影响结果的准确性.例如,在追踪丢失儿童的应用中,在任意一个视频单元中找到丢失的儿童就足够了.因此,可以将所有视频单元分别卸载到多个服务器.由于每个用户可以选择多个服务器,并且分配到不同服务器的视频单元数量取决于服务器的计算能力,所以实现纳什均衡更具挑战性。

众所周知,少数者博弈^[37],也称为 El Faro Bar 问题,是一个具有代表性的资源分配问题.首先引入少数博弈的关键概念截断值,它解释了玩家如何确定策略的依据.例如,在 El Faro Bar 问题的简化版本中,100 人决定去酒吧消费,他们有两个相同的酒吧 A 和 B 作为候选.每个酒吧的容量是有限的,例如有限的房间空间.因此,对每个人而言,选择人少的酒吧会是一个更明智的决定,因为可以享受到更高质量的服务和更优雅的环境.那么该问题中酒吧 A 或 B 的截断值可以表示为 $100/2 = 50$.如果酒吧 A 中的人数小于截断值,那么酒吧 A 中的人在该少数者博弈中获胜.否则,酒吧 B 中的人获胜.接下来定义在视频分析任务卸载问题中的截断值。

定义 4 (截断值):截断值 ϕ_m 表示卸载到每个边缘服务器 m 的视频单元数量的阈值,基于此可以最小化所有边缘服务器中最大的计算和通信延迟和.所有边缘服务器的截断值构成集合 $\Phi = \{\phi_1, \phi_2, \dots, \phi_M\}$,其中满足

$$\sum_{m \in M} \phi_m = \sum_{n \in N} s_n \quad (19)$$

定理 2:为了最小化所有边缘服务器中最大的计算和通信延迟和,对于每个边缘服务器 $m \in M$,满足

$$\phi_m = (Cap_m \sum_{n \in N} s_n) / \sum_{j \in M} Cap_j \quad (20)$$

证明:

与 El Faro Bar 问题类似,当所有服务器共享相同的计算和通信延迟和时,所有用户之间的最大总延迟最小,即

$$\tau_1 + \varphi = \tau_2 + \varphi = \dots = \tau_M + \varphi = D^* \quad (21)$$

其中常量 D^* 是理论最优的整体时延.根据公式(1),对于每个截断值 ϕ_m ,有

$$\frac{\phi_m C_u}{Cap_m} + \varphi = D^* \quad (22)$$

移动上式,得

$$\phi_m = \frac{Cap_m}{C_u} (D^* - \varphi) \quad (23)$$

进一步地,在上式两边同时加上 $\sum_{m \in M}$,并基于定义 4,得

$$D^* = (\sum_{n \in N} s_n C_u + \sum_{m \in M} \varphi Cap_m) / \sum_{m \in M} Cap_m \quad (24)$$

将公式(24)代入(23),得

$$\phi_m = (Cap_m \sum_{n \in N} s_n) / \sum_{j \in M} Cap_j \quad (25)$$

定理 2 得证.

根据定义 4,当卸载到边缘服务器 m 的视频单元数量小于 ϕ_m 时,那么一定存在一个其他的边缘服务器,它的计算和通信延迟较高.相反,当卸载到 m 的视频单元数量明显大于 ϕ_m 时,可能会提高所有服务器的整体延迟(早完成的服务器可能会等待晚完成的服务器).因此,为了最小化所有边缘服务器中最大的计算和通信延迟和,卸载到每个边缘服务器的视频单元数量 m 应该接近截断值 ϕ_m .

但是,注意到公式(20)中计算的截断值可能不是整数,因而公式(24)中的常数 D^* 只是理论上的最优整体延迟.因此,不能直接利用截断值作为卸载到边缘服务器的视频单元数量.为了确定每个用户卸载到服务器的视频单元数量,提出了一种基于博弈论的视频单元分配算法.通过该算法,可以实现纳什均衡,即用户没有动机单方面改变自己的任务卸载决策.

基于博弈论的视频单元分配算法如算法 2 所示,以下为三个主要的步骤(如图 3 所示):

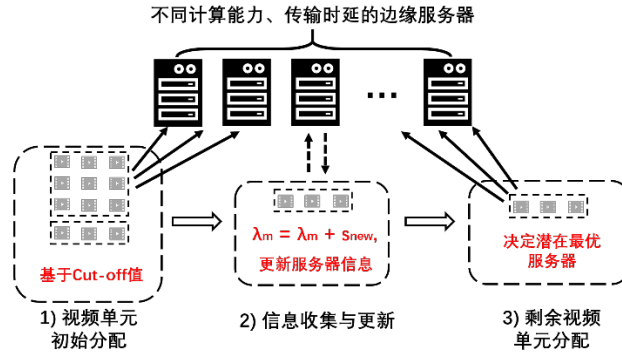


Fig.3 Three main procedures of Alg. 2

图 3 算法 2 的三个子步骤

- 1) **初始视频单元分配:**基于一组截断值 $\Phi = \{\phi_1, \phi_2, \dots, \phi_M\}$,计算用户 n 卸载到边缘服务器 m 的视频单元初始分配数为

$$x'_{n,m} = \left\lfloor \frac{s_n \phi_m}{\sum_{j \in M} \phi_j} \right\rfloor \quad (26)$$

其中使用舍入整数方法来近似初始视频单元分配 $x'_{n,m}$.不难看出,最初会将更多的视频单元分配给具

有较大截断值的服务器.

- 2) **信息收集与更新:**在此过程中,用户收集其他用户关于视频单元数量的卸载决策信息,并了解边缘服务器的资源利用率.对于边缘服务器 m ,其视频单元数量更新为

$$\lambda_m = \lambda_m + s_{new} \quad (27)$$

其中 s_{new} 表示新卸载到服务器 m 的视频单元数量,它取决于用户的视频单元分配决策.通过信息收集和更新,用户为剩余的视频单元分配做准备.

- 3) **剩余视频单元分配:**由于用舍入整数方法来近似初始视频单元分配,因此初始视频单元分配与理论上的最优分配之间存在差距.所以对于用户 n ,有剩余视频单元的数量:

$$r_n = s_n - \sum_{m \in M} x'_{n,m} \quad (28)$$

对于剩余的视频单元,用户将它们每一个卸载到**定义 3**中的潜在最佳服务器.由于每个视频单元的大小相同(即 C_u).根据**定理 1**,剩余视频单元卸载到潜在最佳服务器的顺序无关紧要.

算法 2. 基于博弈论的视频单元分配算法

输入:用户 n 的视频单元数量 s_n ,边缘服务器 m 的算力 Cap_m ,所有用户的期望传输时延 φ

- 1: **for** each user $n \in N$ **do**
- 2: Publish its number of video unit s_n ;
- 3: Collect number of video units of each other user;
- 4: Calculate cut-off value set $\Phi = \{\phi_1, \phi_2, \dots, \phi_M\}$ following Eq. (20);
- 5: Calculate initial allocation $x'_{n,m}$ following Eq. (26);
- 6: **for** each server $m \in M$ **do**
- 7: Offload $x'_{n,m}$ video units to server m ;
- 8: **for** each user $n \in N$ **do**
- 9: Collect information of others' video unit allocation;
- 10: $\lambda_m = \lambda_m + s_{new}$;
- 11: Calculate the number r_n of remaining video units following Eq. (28);
- 12: **for** each video unit u in remaining video units **do**
- 13: Calculate potential optimal server following Eqs. (7) and (8);
- 14: Offload u to its potential optimal server;
- 15: Publish information of video unit allocation;

定理 3:在所有用户的视频单元通过算法 2 卸载到边缘服务器进行视频分析后,可以实现纳什均衡,即没有用户有动机单方面改变其任务卸载决策.

证明:

由于每个视频单元的大小相同(即 C_u),剩余视频单元卸载到潜在最佳服务器的顺序无关紧要.类似**定理 1**中遵循的想法,可以证明在所有剩余的视频单元都卸载到各自潜在的最优服务器后,达到了纳什均衡,即所有用户没有动机单方面改变他们的卸载决定.得证.

为了从理论上验证算法 2 的性能,推导出算法 2 的实际总延迟 D 与公式 2 中获得的理论最优总延迟 D^* 之间的差距.通过算法 2,所有用户的视频单元被卸载到适当的边缘服务器进行视频分析,有实际的整体延迟为

$$D = \max_{m \in M} \{\tau_m + \varphi\} \quad (29)$$

实际总延迟 D 是所有边缘服务器中最大的计算和通信延迟和,有以下定理成立.

定理 4:实际总延迟 D 与理论最优总延迟 D^* 之间的差距不大于 $C_u / \min_{m \in M} \{Cap_m\}$,其中 C_u 是每个视频单元的计算需求,

$\min_{m \in M} \{Cap_m\}$ 是所有边缘服务器中的最小计算能力.

证明:

令 m_1 表示延迟为 D 的服务器,令 m_2 表示计算和通信延迟和最小的服务器,并且设

$$D' = \tau_{m_2} + \varphi \quad (30)$$

由于公式(20)中计算的截断值可能不是整数,它不能直接用作卸载到服务器的视频单元的数量.由此可以确定,卸载到服务器 m_1 的视频单元数量不小于截断值 ϕ_{m_1} ,并且卸载到服务器 m_2 的视频单元数量不超过 ϕ_{m_2} .然后有 $D' \leq D^* \leq D$.

基于定理 2 和算法 1 中的想法,对于边缘服务器 m_1 上的每个视频单元,如果将其转移到另一个服务器 m_2 ,计算和传输延迟不会减少,即 $D < D' + C_u/Cap_{m_2}$.因此满足

$$D - D^* \leq D - D' \leq C_u/Cap_{m_2} \leq C_u / \min_{m \in M} \{Cap_m\}$$

因此定理 4 得证.

在实际场景中,当边缘服务器的计算能力为 10GHz 左右, C_u 设置为 10^6 个 CPU 周期左右时, D 和 D^* 之间的差距可以限定在 1 ms 内,这显示了算法 2 的优越性能.

5 实验验证

在本节中,通过各种设置的实验来评估提出算法的性能,并将设计的算法与其他一些现有方法进行比较.

5.1 实验设置

与之前的研究^[32,38]类似,考虑一个包含 24 个用户和 6 个边缘服务器的边缘计算系统.根据现有工作^[39]中的实验测量,服务器计算能力 Cap_m 服从高斯分布 $N(5,4)$ GHz,在实验中计算能力的方差设置为不同值.根据现有工作^[38],信道带宽 W 设置为 20MHz,背景噪声设置为 50dBm,每个用户保持传输功率 $P_n \sim N(1,0.1)W$.设置信道增益 $H_{n,0} = (dist_{n,0})^\mu$,其中 $dist_{n,0}$ 是用户 n 和主边缘服务器之间的距离,路径损耗因子 μ 设置为 4.

对于任务卸载,使用源自 AI City Datasets 2019^[25]的视频在目标检测子 YOLOv3 上进行视频分析.视频长度 l_n 从 10 秒到 5 分钟不等,视频帧率 f_n 从集合 $\{2,3,5,10,30\}$ fps 中分配^[12].在实验中,将 L_u 设置为 1s,并将 F_u 设置为 1fps.为了合理设置 C_u 的值,使用 YOLOv3 在 NVIDIA Jetson TX2 上处理大小为 1920×1080 的图像.该图像在评估中可以视为一个视频单元,根据实验测量结果, C_u 设置为 2.25×10^6 个 CPU 周期.

将非分布式场景中的算法 1 称为非分布式任务卸载方案(Non-distributed Task Offloading Scheme,简称 NTOS),将分布式场景中的算法 2 称为分布式任务卸载方案(Distributed Task Offloading Scheme,简称 DTOS).将设计的算法与以下方案进行比较:

- 1) 贪心方案(Greedy Scheme,简称 GS):与现有相关工作^[36]类似,当每个用户卸载视频分析任务时,基于最优反应函数^[40]选择上一轮其他用户决策之后的最优卸载决策.
- 2) 计算能力优先方案(Computation Capacity Prior Scheme,简称 CCPS):用户在任务卸载博弈中做出决策时优先考虑计算能力,作为综合考虑服务器计算能力和传输延迟的 NTOS 和 DTOS 的对比基准方法.
- 3) 传输延迟优先方案(Transmission Delay Prior Scheme,简称 TDPS):用户在任务卸载博弈中做出决策时优先考虑传输延迟,作为综合考虑服务器计算能力和传输延迟的 NTOS 和 DTOS 的对比基准方法.
- 4) 随机方案(Random Scheme,简称 RS):用户在任务卸载博弈中随机选择边缘服务器进行任务卸载^[41],作为 NTOS 和 DTOS 的对照方法.
- 5) 理论上 OPTimal 方案(Theoretically OPTimal scheme,简称 TOPT):TOPT 的计算基于定理 2 中截断值的计算结果^[37].值得注意的是,由于截断值可能不是整数,因此虽然 TOPT 达到理论上的最优解,但是实践中可能无法实现,以此作为方法 DTOS 的对照方法.

5.2 实验结果

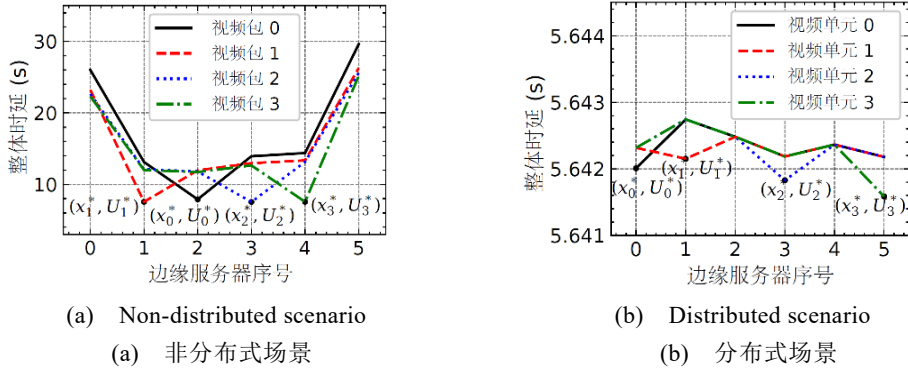
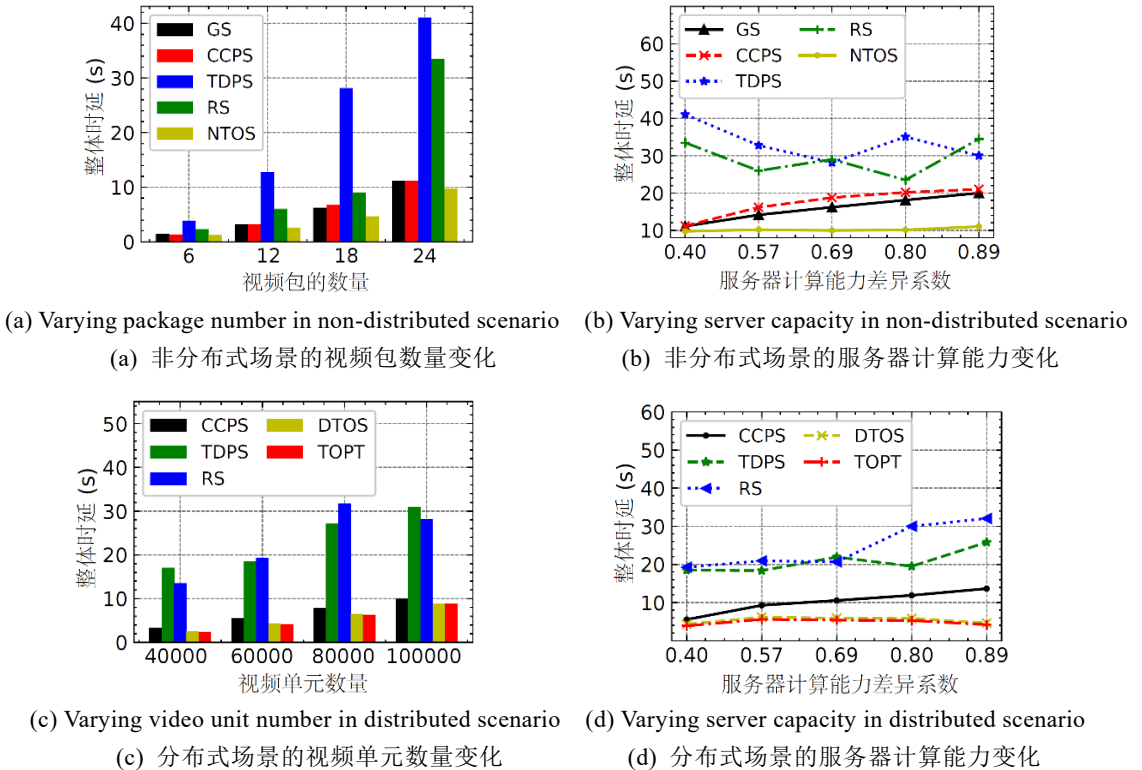


Fig.4 NE achieved in non-distributed and distributed scenarios after running Algs. 1 and 2, respectively
图 4 分别运行算法 1 和 2 在非分布式和分布式场景中实现了纳什均衡

1) 纳什均衡的结果:首先研究运行算法 1 和算法 2 后的纳什均衡结果.如图 4(a)所示,在所有用户中任意选择四个包,并显示将包卸载到的边缘服务器后各个服务器的整体延迟.对于来自用户 0 的包 0,观察到它在服务器 2 上时,整体延迟约为 8 秒.但是,当包 0 被卸载到其他服务器时,整体延迟会变高,如图 4(a)所示,其他包也是如此.与算法 1 中的包类似,将从四个用户中任意选择的四个视频单元从其潜在最优服务器转移到其他服务器,如图 4(b)所示,所有这些视频单元的整体延迟都变得更高.因此,可以图 4 验证了在运行设计的算法 1 和算法 2 之后,所有用户都没有动机单方面改变他们的决定,即实现了纳什均衡.

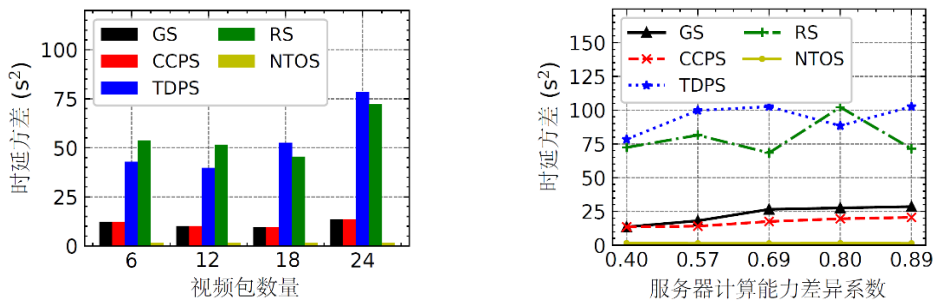


(a) Varying package number in non-distributed scenario (a) 非分布式场景的视频包数量变化
(b) Varying server capacity in non-distributed scenario (b) 非分布式场景的服务器计算能力变化
(c) Varying video unit number in distributed scenario (c) 分布式场景的视频单元数量变化
(d) Varying server capacity in distributed scenario (d) 分布式场景的服务器计算能力变化

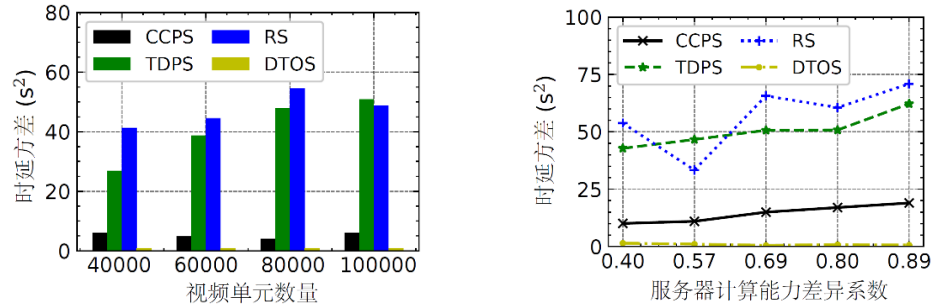
Fig.5 Overall delay with varying video amount and server capacity in non-distributed and distributed scenarios

图 5 在非分布式和分布式场景中随着视频量和服务器容量的变化而产生的总体延迟

2) 总体延迟比较:如图 5 所示,将设计的算法与其他方法在传输和计算的总体延迟方面进行比较.当非分布式视频分析场景中的包数量增加时,服务器运行计算任务的时间会增加.因此,如图 5(a)所示,整体延迟变高.可以观察到,设计的 NTOS 在整体延迟方面比其他方法具有更好的性能.此外,定义差异系数为服务器计算能力的标准差和期望值的比值,可以发现当差异系数从 0.40 变化到 0.89 时,GS 和 CCPS 的总体延迟在图 5(b)中呈现增加的趋势,这是由于更多的视频数据将被卸载到具有高计算能力的服务器上,导致整体延迟较高.相比之下,设计的 NTOS 可以在不同的设置下获得稳定的整体延迟.同样,在分布式场景中改变视频单元数和服务器容量的差异系数.在图 5(c)和 5(d)中,将 DTOS 与理论最优方案 TOPT 进行比较,发现 DTOS 的整体延迟略高于 TOPT.因此,可以看出设计的算法 2 可以获得接近最优解决方案的性能结果.



(a) Varying package number in non-distributed scenario (b) Varying server capacity in non-distributed scenario
 (a) 非分布式场景的视频包数量变化 (b) 非分布式场景的服务器计算能力变化



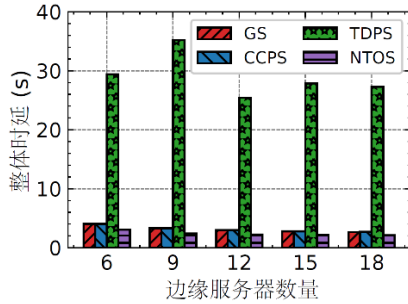
(c) Varying video unit number in distributed scenario (d) Varying server capacity in distributed scenario
 (c) 分布式场景的视频单元数量变化 (d) 分布式场景的服务器计算能力变化

Fig.6 Delay variance with varying video amount and server capacity in non-distributed and distributed scenarios

图 6 在非分布式和分布式场景中随着视频量和服务器容量的变化而产生的延迟方差

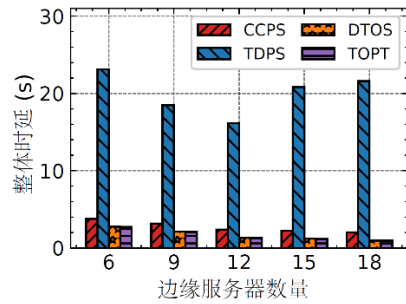
3) 延迟方差比较:如图 6 所示,研究了所有边缘服务器的延迟差异,并计算了不同视频数据量和服务器容量的延迟方差. CCPS 方法在做卸载决策时优先考虑边缘服务器的计算能力,因此可以有效地规避过多的用户向同一个边缘服务器卸载视频分析任务,从而超出服务器负载能力的问题;而 TDPS 方法仅仅优先考虑传输延迟,而非边缘服务器的计算能力,因此有的服务器的处理时间由于任务过载而显著延长,而有的服务器被分配视频数据太少,导致该服务器的计算资源利用率低.因此,就延迟方差而言,CCPS 的性能比 TDPS 平均提升了 79%.但是提出的算法 1 和算法 2 兼顾了边缘服务器的计算能力和传输延迟,在各种设置下得到的延迟方差接近于 0,因而可以得知所有服务器的最大和最小延迟之间的差距非常小,它相对于 CCPS 降低了 58%的平均时延方差,所以设计的算法 1 和算法 2 可以更进一步地降低用

户的整体延迟。



(a) Varying server number in non-distributed scenario

(a) 非分布式场景的边缘服务器数量变化



(b) Varying server number in distributed scenario

(b) 分布式场景的边缘服务器数量变化

Fig.7 Overall delay with varying edge server number in non-distributed and distributed scenarios

图 7 在非分布式和分布式场景中随着边缘服务器数量的变化而产生的总体延迟

- 4) 边缘服务器数量的变化:最后,针对非分布式场景和分布式场景不断改变边缘服务器的数量,验证提出的算法 1 和算法 2 具有较好的扩展性,在服务器数量变化的不同场景下,相比其他对比算法能取得更好的效果.如图 7(a)所示的非分布式场景中,随着服务器数量由 6 增加至 18,对比方法 GS、CCPS 和设计的方法 NTOS 取得的整体时延均有一个下降的趋势,原因是随着更多边缘服务器可供选择,视频分析任务会被分配到更多的服务器上,降低了计算时延,从而减少了整体时延.同时,相对于 GS 和 CCPS,NTOS 平均降低了 24.7%的整体时延.然而对比方法 TDPS 由于优先考虑传输时延,因此虽然边缘服务器数量在不断增加,TDPS 方法倾向于将视频分析任务卸载至传输时延较低的边缘服务器上,所以 TDPS 方法取得的整体时延总是在 30s 左右波动.如图 7(b)所示的分布式场景中,对比方法 CCPS 和 TDPS 有类似于非分布式场景中的变化趋势,同时设计的方法 DTOS 在服务器数量变化的不同场景下,总是可以获得接近最优解决方案 TOPT 的性能结果.

6 讨论

细粒度的视频单元:在本文中,最小的不可分割视频单元是单帧.打算考虑更细粒度的视频单元,即将一个视频帧分成多个切片,每个切片形成一个视频单元.然而,由于输入之间的依赖性,深度神经网络(例如卷积神经网络)在视频帧的切片上执行具有挑战性,这是由以邻域值作为输入的神经元操作的结果.为了解决这个问题,可以利用依赖感知推理^[42]来分析切片的依赖关系,并推断具有足够多邻值的区域.

信息不完整的任务卸载游戏:在分布式环境中,完整的信息并不总是可用的,用户通常不知道其他人做出的决策.此外,当多服务器多用户视频分析任务卸载问题被表述为多人游戏时,每个玩家的策略空间对其他人是未知的,并且所有用户的视频单元数量不同.因此,实现纳什均衡的难度大大增加,这将是未来的工作.

使用强化学习的任务卸载博弈:在现实世界中,用户可能对边缘服务器的计算能力和传输成本一无所知.在视频分析任务卸载场景中,可以使用马尔可夫决策过程来对决策进行建模,并利用强化学习让用户确定适当的行动.此外,可以根据效用函数计算每个用户的奖励.强化学习的利用有助于提高在不同情况下做出视频分析任务卸载决策的灵活性,并且经过长时间的运行,最终可以在没有完整或未来信息的情况下获得接近最优的卸载决策,这可能具有挑战性和意义.

7 总结

在本文中,研究了多服务器多用户视频分析任务卸载问题,用户选择合适的边缘服务器并将其视频数据卸载到服务器进行分析.为了最大限度地减少整体延迟并获得用户没有动机单方面改变卸载决策的稳定情况,该

问题被建模为多人博弈.为了达到纳什均衡,针对非分布式和分布式两种不同的场景分别提出了基于博弈论的算法,并且证明了在这两种情况下都可以达到纳什均衡,同时保证较低的整体时延.最后,通过可扩展实验展示算法的良好性能.

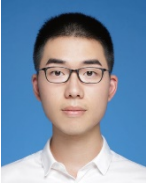
References:

- [1] Li ZY, Qi W, Chen YF, Xie GQ, Li RF. A survey on task offloading research in vehicular edge computing. *Chinese Journal of Computers*, 2021, 44(5): 963-982.
- [2] Zhou YZ, Zhang D. Near-end cloud computing: opportunities and challenges in the post-cloud computing era. *Chinese Journal of Computers*, 2019, 42(4): 677-700.
- [3] Shi WS, Zhang XZ, Wang YF, Zhang QY. Edge Computing: State-of-the-Art and Future Directions. *Journal of Computer Research and Development*, 2019, 56(1): 69-89.
- [4] Xie RC, Lian XF, Jia QM, Huang T, Liu YJ. Survey on computation offloading in mobile edge computing. *Journal on Communications*, 2018, 39(11): 138-155.
- [5] Wang S, Zhang X, Zhang Y, et al. A survey on mobile edge networks: Convergence of computing, caching and communications. *IEEE Access*, 2017, 5: 6757-6779.
- [6] Caprolu M, Di Pietro R, Lombardi F, et al. Edge computing perspectives: architectures, technologies, and open security issues. *IEEE EDGE*, 2019: 116-123.
- [7] Zhou Z, Chen X, Li E, et al. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 2019, 107(8): 1738-1762.
- [8] Hsieh K, Ananthanarayanan G, Bodik P, et al. Focus: Querying large video datasets with low latency and low cost. *USENIX OSDI*, 2018: 269-286.
- [9] Satyanarayanan M, Bahl P, Caceres R, et al. The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 2009, 8(4): 14-23.
- [10] Ananthanarayanan G, Bahl P, Bodik P, et al. Real-time video analytics: The killer app for edge computing. *Computer*, 2017, 50(10): 58-67.
- [11] Hung CC, Ananthanarayanan G, Bodik P, et al. Videoedge: Processing camera streams using hierarchical clusters. *IEEE/ACM Symposium on Edge Computing*, 2018: 115-131.
- [12] Wang C, Zhang S, Chen Y, et al. Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics. *IEEE Conference on Computer Communications*, 2020: 257-266.
- [13] Jiang JC, Ananthanarayanan G, Bodik P, et al. Chameleon: scalable adaptation of video analytics. *ACM SIGCOMM*, 2018: 253-266.
- [14] Wu TT, Dou WC, Ni Q, Yu S, Chen GH. Mobile live video streaming optimization via crowdsourcing brokerage. *IEEE Transactions on Multimedia*, 2017, vol. 19, no. 10, pp. 2267-2281.
- [15] Sun YX, Zhou S, Xu J, EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks. *IEEE JSAC*, 2017, vol. 35, no. 11, pp. 2637-2646.
- [16] Zhou YP, Chen L, Jing M, Ming Z, Xu YD. Performance analysis of thunder crystal: A crowdsourcing-based video distribution platform. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018, vol. 28, no. 4, pp. 997-1008.
- [17] Dutta PK. *Strategies and games: theory and practice*. MIT press, 1999.
- [18] Li N, Martinez-Ortega JF, Rubio G. Distributed joint offloading decision and resource allocation for multi-user mobile edge computing: A game theory approach. *arXiv preprint arXiv:1805.02182*, 2018.
- [19] Wang WB, Kwasinski A, Niyato D, et al. A survey on applications of model-free strategy learning in cognitive wireless networks. *IEEE Communications Surveys & Tutorials*, 2016, 18(3): 1717-1757.
- [20] Abdalzaher MS, Seddik K, Elsabrouty M, et al. Game theory meets wireless sensor networks security requirements and threats mitigation: A survey. *Sensors*, 2016, 16(7): 1003.
- [21] Hu M, Xie ZX, Wu D, et al. Heterogeneous edge offloading with incomplete information: A minority game approach. *IEEE TPDS*, 2020, 31(9): 2139-2154.
- [22] Zheng JC, Cai YM, Liu YK, et al. Optimal power allocation and user scheduling in multicell networks: Base station cooperation using a game-theoretic approach. *IEEE TWC*, 2014, 13(12): 6928-6942.

- [23] Zhan YF, Guo S, Li P, et al. A deep reinforcement learning based offloading game in edge computing. IEEE TC, 2020, 69(6): 883-893.
- [24] Lei Y, Zheng WB, Wei W, Xia YN, Li XB, Liu CW, Xie H. Task offloading method based on probabilistic performance awareness and evolutionary game strategy in “cloud + edge” hybrid environment. Journal of Computer Applications, 2021, 41(11): 3302-3308.
- [25] Naphade M, Tang Z, Chang M C, et al. The 2019 AI City Challenge. IEEE CVPR Workshops, 2019, 8: 452-460.
- [26] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection. IEEE CVPR, 2016: 779-788.
- [27] Gao MJ, Shen RJ, Shi L, et al. Task Partitioning and Offloading in DNN-Task Enabled Mobile Edge Computing Networks. IEEE TMC, 2021: 1-12.
- [28] Yan J, Bi SZ, Zhang YJ, et al. Optimal task offloading and resource allocation in mobile-edge computing with inter-user task dependency. IEEE TWC, 2019, 19(1): 235-250.
- [29] Wu HM, Wolter K, Jiao PF, et al. EEDTO: an energy-efficient dynamic task offloading algorithm for blockchain-enabled IoT-edge-cloud orchestrated computing. IEEE IoTJ, 2020, 8(4): 2163-2176.
- [30] Liu W, Huang YC, Du W, Wang W. Resource-Constrained Serial Task Offload Strategy in Mobile Edge Computing. Journal of Software, 2020, 31(6): 1889-1908.
- [31] Wu ZQ, Ye DD, Yu R, Zhou WH, He ZS. Edge Computing Offloading with Parked Vehicular Collaboration in Internet of Vehicles. Journal of Beijing University of Posts and Telecommunications, 2019, 42(2): 108-113.
- [32] Kang D, Emmons J, Abuzaid F, et al. Noscope: optimizing neural network queries over video at scale. arXiv preprint arXiv: 1703.02529, 2017.
- [33] Ray D, Kosaian J, Rashmi K V, et al. Vantage: optimizing video upload for time-shifted viewing of social live streams. ACM SIGCOMM, 2019: 380-393.
- [34] Ren JK, Yu GD, Cai YL, et al. Latency optimization for resource allocation in mobile-edge computation offloading. IEEE TWC, 2018, 17(8): 5506-5519.
- [35] Li MK, Ke ZX, Yu DR, Zhang JL, Jia JY, Liu LF. Vehicle license plate number recognition algorithm based on mobile edge calculation. Computer Engineering and Design, 2021, 42(11): 3151-3157.
- [36] Ranadheera S, Maghsudi S, Hossain E. Minority games with applications to distributed decision making and control in wireless networks. IEEE Wireless Communications, 2017, 24(5): 184-192.
- [37] Wang YJ, Yuan JT, Yu GD, et al. Minority game for distributed user association in unlicensed heterogenous networks[J]. IEEE TWC, 2020, 19(6): 4220-4233.
- [38] Zou JF, Hao TB, Yu C, et al. A3C-DO: A regional resource scheduling framework based on deep reinforcement learning in edge scenario. IEEE TC, 2020, 70(2): 228-239.
- [39] Li N, Martinez-Ortega JF, Diaz VH. Distributed power control for interference-aware multi-user mobile edge computing: a game theory approach. IEEE Access, 2018, 6: 36105-36114.
- [40] Mark V. Best-response potential games. Economics letters 66.3, 2000: 289-295.
- [41] Javad G. Randomized algorithms for scheduling VMs in the cloud. IEEE INFOCOM, 2016: 1-9.
- [42] Liu LY, Li HY, Gruteser M, Edge assisted real-time object detection for mobile augmented reality. ACM MobiCom, 2019: 1-16.

附中文参考文献:

- [1] 李智勇,王琦,陈一凡,谢国琪,李仁发.车辆边缘计算环境下任务卸载研究综述.计算机学报,2021,44(5):963-982
- [2] 周悦芝,张迪.近端云计算:后云计算时代的机遇与挑战.计算机学报,2019,42(4):677-700
- [3] 施巍松,张星洲,王一帆,张庆阳.边缘计算:现状与展望.计算机研究与发展,2019,56(1):69-89
- [4] 谢人超,廉晓飞,贾庆民,黄韬,刘韵洁.移动边缘计算卸载技术综述.通信学报,2018,39(11):138-155
- [24] 雷鹰,郑万波,魏嵬,等.基于概率性能感知演化博弈策略的“云+边”混合环境中任务卸载方法.计算机应用,2021,41(11):3302-3308
- [30] 刘伟,黄宇成,杜薇,等.移动边缘计算中资源受限的串行任务卸载策略.软件学报,2020,31(6):1889-1908
- [31] 吴振铨,叶东东,余荣,周文辉,何昭水.车联网中基于停车协同的边缘计算卸载方法.北京邮电大学学报,2019,42(2):108-113
- [35] 李猛坤,柯正轩,于定荣,张建林,贾军营,刘利峰.基于移动边缘计算的车联网车牌号码识别算法.计算机工程与设计,2021,42(11): 3151-3157



陈彧(1997-),男,博士研究生,主要研究领域为边缘计算、视频分析、博弈论等.



张胜(1986-),男,博士,副教授,博士生导师,CCF 会员,主要研究领域为分布式计算、边缘计算和视频分析等.



金熠波(1994-),男,博士研究生,CCF 会员,主要研究领域为大数据分析、边缘智能、视频分析等.



钱柱中(1980-),男,博士,教授,博士生导师,CCF 会员,主要研究领域为分布式系统、云计算、边缘计算等.



陆桑璐(1970-),女,博士,教授,博士生导师,CCF 会员,主要研究领域为是分布计算、普适计算、无线网络等.