

引用格式:李波,黄鑫,牛力,等. 车载边缘计算环境中的任务卸载决策和优化 [J]. 微电子学与计算机, 2019, 36(2): 78-82.
[LI B, HUANG X, NIU L, et al. Task offloading decision in vehicle edge computing environment [J]. Microelectronics & Computer, 2019, 36(2): 78-82.]

车载边缘计算环境中的任务卸载决策和优化

李 波,黄 鑫,牛 力,薛 端,白晨青
(云南大学 信息学院,云南 昆明 650500)

摘 要: 车载云计算卸载存在高网络延迟、核心网负载严重等问题,本文提出了一种计算切换策略来降低车载边缘计算环境下的任务完成时间. 基于 Matlab R2016a 平台对最小执行时间算法和最小完成时间算法进行了 2 组仿真实验,结果表明:通过本文提出的计算切换策略能改良任务卸载效率,缩短任务的完成时间.
关键词: 计算卸载;最小执行时间算法;最小完成时间算法;车载边缘计算;计算切换
中图分类号: TP316.4 **文献标识码:** A **文章编号:** 1000-7180(2019)02-0078-05

Task Offloading Decision in Vehicle Edge Computing Environment

LI Bo, HUANG Xin, NIU Li, XUE Duan, BAI Chen-qing
(School of Information Science and Engineering, Yunnan University, Kunming 650500, China)

Abstract: Vehicle cloud computing offloading has high network latency and serious core network load. This paper proposes a handoff strategy to reduce the task completion time under the vehicle edge computing environment. Based on the Matlab 2016a platform, two sets of simulation experiments were performed. The results show that the handoff strategy proposed in this paper can improve the task offloading efficiency and shorten the task completion time.
Key words: computational offloading; minimum execution time algorithm; minimum completion time algorithm; vehicle edge computing; handoff

1 引言

随着智能交通系统的快速发展,如何处理车辆的需求信息、路况信息等海量数据成为了该领域重要的研究课题之一,车联网(Internet of Vehicles, IoV)在该方面发挥着关键作用. 目前,车辆可以与其他车辆(Vehicle to Vehicle, V2V)、路侧单元(Vehicle to Road Side Unit, V2R)以及中央云(Vehicle to Cloud, V2C)等基础设施进行信息交互.
车辆自身尺寸限制及结构限制等情况会导致计算资源受限等问题,传统的解决方案是将任务上传

至远程云端进行处理. 这种把任务从资源受限的设备迁移到其他资源更加丰富的设备上处理的方式被称为计算卸载^[1],以达到增强设备能力、降低能耗等目的. 车辆通过路测单元访问 Internet,并通过 Internet 在远程云端获取更加丰富的计算资源,形成车载云计算^[2]环境(Vehicular Cloud Computing, VCC).
随着计算密集型任务的数目和比例不断增加,车辆-云端两层计算结构将造成云端的流量过载、高回程带宽消耗等问题. 在文献[3]中,作者利用车载接入网络中的认知无线电和软数据融合技术,设计

了认知云车辆网络的分布式流量卸载方案。

云端具有覆盖范围广的特点,任务由云端完成前,车辆往往处于同一个云端的覆盖范围,所以车辆的移动不会改变车与云端的通信链路。车载边缘计算不同于车载云计算,由于路侧单元覆盖范围小,车辆的移动性会导致车辆到所选边缘服务器的链路发生变化,导致通信时间随着车辆运动发生变化,从而影响任务的完成时间。为了解决该问题,本文提出了一种计算切换策略来降低任务完成时间,提高任务在边缘层的卸载效率。

2 车载边缘计算卸载决策

2.1 调度算法

在计算卸载中,任务调度至关重要。任务调度是指将任务合理分配至可选资源,选择合理有效的调度算法,可使计算卸载效果更优,提高核心云和边缘层的资源利用效率,本文涉及到的调度算法有:

(1)最小执行时间算法(Minimum Execution Time, MET):它将每个任务分配给能够以最少的执行时间运行任务的代理资源上。在异构计算环境中,最小执行时间算法可能会导致任务集中到计算能力较强的少数资源上,导致严重的负载不均衡和较长的任务等待时间。

(2)最小完成时间算法(Minimum Complete Time, MCT):它将每个任务分配给具有最短预计完成时间的代理资源上,与最小执行时间算法相比,此算法克服了最小执行时间算法中的负载不均衡等问题。

2.2 车载边缘计算

车载边缘计算卸载场景如图1所示。

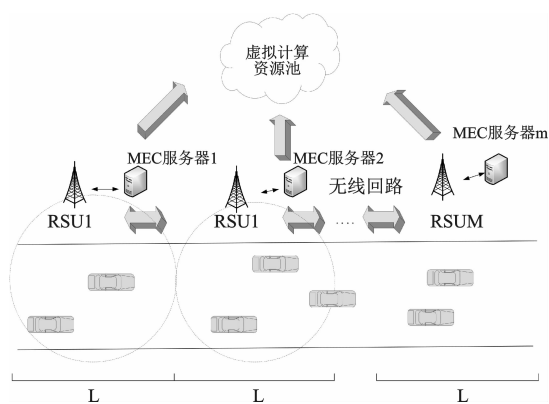


图1 车载边缘计算卸载场景

车载边缘计算卸载场景包括三层:云计算层、边缘计算层和车载计算层。

云计算层:由高性能服务器集群组成,具有强大的计算和存储能力。网关负责与上层链路通信,并将下层数据发送给上层网络。

边缘计算层:主要由具有一定计算能力和存储容量的边缘服务器组成。对接收数据的快速处理和结果的即时转发,为调度提供了方便。同时边缘层还可以记录车辆位置,暂存路侧单元采集到的交通信息等,然后将任务上传至远程云端,实现信息共享。

车载计算层:由路侧单元和车载终端组成。路侧单元以单跳方式与车辆终端、移动设备等连接通信,从而收集诸如车辆位置、速度等信息,并将其发送给可通信的边缘设备。这样车载终端接入网络后,就能够快速获得边缘计算层提供的服务,减少了时延。

2.3 计算卸载决策

云端的计算能力用 f_c 表示,边缘服务器的计算能力用 f_e 表示,车辆的计算能力用 f_l 表示。车辆产生的任务表示为 $task = \{D_{in}, D_{out}, T_{max}\}$,其中 D_{in} 表示任务的数据量, D_{out} 表示结果的数据量, C 表示任务的计算量, T_{max} 表示任务死亡时间。

在车载边缘计算环境下,因为车辆的运动,车辆通常会经过多个路侧单元所在的覆盖范围。假设有 N 个边缘服务器,车辆需与边缘服务器 $S_i (i \leq N)$ 进行通信,当车辆移动到边缘服务器 S_i 的覆盖范围之外时,会因为车辆无法直接与边缘服务器 S_i 进行通信而导致任务的上传和下载过程发生变化。因为通信链路会随着车辆运动发生变化,所以任务的通信时间也是动态变化的。为保证任务的最短完成时间,本文通过最大等效带宽的方式找到通信开销最短的链路。

因为端与端之间的通信时延与它们之间的距离呈正相关,所以本文考虑车辆和边缘服务器之间的物理距离,以评估实验中的通信时延。虽然云端的计算能力强,计算时间短,但车辆到云端的带宽 B_c 小于车辆到边缘服务器的带宽 B_e ,所以卸载过程中到云端的通信开销 $(D_{in} + D_{out})/B_c$ 要大于到边缘层的通信开销 $(D_{in} + D_{out})/B_e$,同理到云端的时延也比单跳下到边缘的时延大。

2.4 计算切换决策

本次实验采用基于 Docker 的容器迁移技术^[4]来减少服务迁移的时间,任务迁移时间是由代理资源之间的带宽和打包的数据量大小所决定,图2为计算切换示意图。

(1)上传阶段

BW 表示等效带宽,等效带宽由网络单跳带宽

2.5 车载边缘计算卸载中的移动模型

常见的车辆运动模型有两种: Random Way-point 模型和 Random Direction 模型^[6]. 综合考虑这两种运动模型的优点和缺点,本文提出了一种适用于车载边缘计算环境下的运动模型,该模型主要有以下 5 个步骤:

- (1) 在仿真区域内选取起始点 S_i 和终点 D_e , 终点设置在离中心区域的一定范围外;
- (2) 在 $[v_{\min}, v_{\max}]$ 中随机选取速度 v , 节点按照这个速度匀速的从 S_i 沿直线移动到 D_e ;
- (3) 到达终点 D 后在 $[t_{\min}, t_{\max}]$ 之间随机选择一个暂停时间, 设定节点不仅能在终点停留, 也可以在运动的中途做短暂停留;
- (4) 暂停时间后, 节点开始下一次运动, 将本次移动的终点 D_e 作为下一过程的起始点;
- (5) 循环(1)到(4)步骤, 直至仿真结束.

3 实验仿真与分析

- 实验仿真过程为:
- (1)生成仿真环境, 设定环境参数;
 - (2)设置任务和资源的属性参数;
 - (3)车辆节点按照运动模型进行运动;

- (4)根据所选算法, 进行任务卸载;
- (5)生成性能参数的结果;
- (6)多次实验, 重复(2)到(5)过程直至仿真次数结束.

基于 Matlab R2016a 的仿真平台, 本文进行 2 组仿真实验:

- (1)根据最小执行时间算法, 对比任务自身计算、卸载至云端、卸载至边缘层的任务执行时间;
- (2)根据最小完成时间算法, 对比任务自身计算、卸载至云端、卸载至边缘层、卸载至边缘层并引入计算切换的任务完成时间.

3.1 环境设定

本次实验不考虑周边障碍物对通信的影响, 时间抽样间隔为 0.1 s , 速度满足道路限速标准下的速度范围. 计算卸载的模式有 V2R 和 V2C 两种, 不考虑 V2V 的模式. 任务为可间断式执行的任务, 即任务未完成部分不受到任务已完成部分的影响.

3.2 参数设置

实验以 iPhoneX 的 Apple A11 GPU 的跑分 14 970 为基准, 根据 Geek2MIPS 准则^[7], 设定代理节点的计算能力为 $82\ 335 \times \lambda$, 云端、边缘服务器、车辆本身的 λ 不同, 参数设置如表 1 所示.

表 1 参数设置

参数名	参数值	参数名	参数值
BS 端覆盖范围(m^2)	$1\ 000 \times 1\ 000$	仿真次数(次)	1 000
RSU 覆盖半径(m)	$[100\ 150]$	边缘端计算能力(MIPS)	$82\ 335 \times [1.25, 1.5, 1.75]$
BS 数目(个)	1	云端计算能力(MIPS)	$82\ 335 \times 2$
边缘端数目(个)	3	车辆计算能力(MIPS)	$82\ 335 \times 0.75$
RSU 数目(个)	16	R2R 基础带宽(Mbps)	5
V2R 基础带宽(Mbps)	10	V2C 基础带宽(Mbps)	2
无线端对端时延(s)	0.01	RSU 端到端时延(s)	0.1
路由跳数范围(跳)	$[1, 2, 3]$	V2C 时延(s)	0.2
任务的输入数据量	50 M	CCR	$[0.1\ 0.9]$
任务的输出数据量	30 M	任务的计算量	$1.8 \times 5 \times 10^6$

3.3 性能参数

本文使用以下性能参数对卸载效果进行评价, 其中平均任务完成时间最为重要.

- (1)平均任务执行时间
平均任务执行时间表示任务在资源上的计算时间.
- (2)平均任务完成时间
平均任务完成时间表示成功卸载的任务的平均完成时间. 单个任务的完成时间定义为该任务的上

传时间、下载时间、通信时延以及计算时间的总和.

3.4 结果分析

本次实验的统计数据是基于 2 组仿真实验所得, 对比了不同的任务执行方案. 图 3 为平均任务执行时间的实验结果, 图 4 为平均任务完成时间的实验结果.

由图 3 可知, 在平均任务执行时间方面, 任务在云端的执行时间最短, 为 54.65 秒. 云端的执行时间和边缘层的执行时间分别比车辆自身的执行时间缩

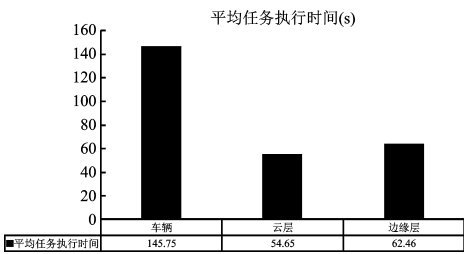


图 3 平均任务执行时间

短了 62.5%和 57.15%.

这是因为按照计算能力的大小排序:云端>边缘层>车辆自身. 根据最小执行时间算法,任务都会卸载至云端,这势必会造成云端的负载严重.

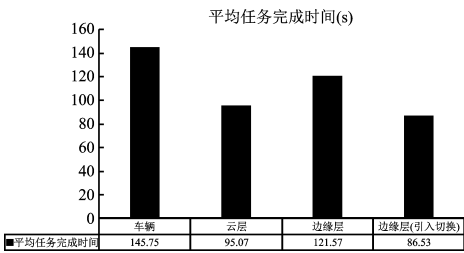


图 4 平均任务执行时间

由图 4 可知,在平均任务完成时间方面,在边缘层引入计算切换的平均任务完成时间最短,为 86.53 秒. 对比不引入计算切换的边缘层的平均任务执行时间缩短了 28.82%,对比自身计算的平均任务执行时间缩短了 40.63%,对比卸载至云端的平均任务完成时间缩短了 8.98%.

这是因为不同网络层次的代理资源带宽不同,处于核心网的云端的带宽要小于处于感知层的边缘服务器,所以任务对代理资源的选择也会不同. 因为通信链路的变化会增加任务的通信开销,所以在未引入计算切换时,边缘层的平均任务完成时间高于云端的平均任务完成时间. 引入计算切换后,任务会时刻处于最优的代理资源上,能有效的降低任务的完成时间.

4 结束语

本文针对车载边缘计算环境中的任务卸载决策进行了研究,对比了云端、边缘层、自身的平均任务执行时间和平均任务完成时间. 考虑到车辆的移动

性会导致通信链路的变化,进而影响到通信时间,所以本文在传统调度算法上,提出了运用计算切换的方式来保证任务完成时间最短. 结果表明:通过本文提出的切换决策,能为任务找到当前时刻的最优卸载位置,能达到降低任务的完成时间的目的.

在未来的研究中,将重点研究车载边缘计算环境中计算切换的切换准则,在保证任务完成时间的同时,降低移动端的能耗开销和任务的执行成本,以达到切换效果最优的目的.

参考文献:

[1] Jiang Z, Mao S. Energy Delay Tradeoff in Cloud Offloading for Multi-Core Mobile Devices[J]. IEEE Access, 2017, 3:2306-2316.

[2] Bitam S, Mellouk A, Zeadally S. VANET-cloud: a Generic Cloud Computing Model for Vehicular Ad Hoc Networks[J]. IEEE Wireless Communications, 2015, 22(1):96-102.

[3] Hou X, Li Y, Chen M, et al. Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures [J]. IEEE Transactions on Vehicular Technology, 2016, 65(6):3860-3873.

[4] Bernstein D. Containers and Cloud: From LXC to Docker to Kubernetes[J]. IEEE Cloud Computing, 2015, 1(3):81-84.

[5] 李波, 薛端, 黄鑫. 移动云计算中的网络切换管理 [J]. 微电子学与计算机, 2018,35(2):37-42.

[6] 于冲, 赵海, 司帅宗, 等. 车联网运动模型的复杂网络解析方法[J]. 电子学报, 2017, 45(6):1449-1455.

[7] Li B, Pei Y, Wu H, et al. Heuristics to Allocate High-Performance Cloudlets for Computation Offloading in Mobile ad hoc Clouds[J]. Journal of Supercomputing, 2015, 71(8):3009-3036.

作者简介:

李 波 男,(1977-),博士,副教授. 研究方向为移动云计算中资源共享技术.

黄 鑫(通讯作者) 男,(1994-),硕士研究生. 研究方向为物联网、车载边缘计算中的卸载技术.

E-mail:454765958@qq. com.