

雾计算的任务卸载在移动终端的研究

侯严严, 陆贝妮, 薛欢欢

嘉兴南湖学院信息工程学院, 浙江 嘉兴

收稿日期: 2022年1月14日; 录用日期: 2022年2月10日; 发布日期: 2022年2月18日

摘要

针对移动云计算存在的传输延迟高和核心网络负载严重等问题, 主流的解决方案是通过计算下沉的方式在路侧单元上部署边缘服务器, 这种卸载到边缘服务器或路由器的计算模式称为雾计算。本文提出了两种雾计算环境下的任务卸载算法: 泛洪算法和基于阈值的切换算法, 并通过Matlab R2017a平台对这两种卸载算法进行分析, 并与本地执行和卸载到云服务器执行进行比较。结果表明雾计算环境下的任务卸载算法不仅能减轻云服务器的负载, 而且能够大大缩短任务的完成时间。雾计算环境下, 基于阈值的切换算法比泛洪算法更好地节约任务完成时间。

关键词

雾计算, 任务卸载, 泛洪算法, 计算切换

Research on Task Offloading of Fog Computing in Mobile Terminals

Yanyan Hou, Beini Lu, Huanhuan Xue

School of Information Engineering, Jiaxing Nanhu University, Jiaxing Zhejiang

Received: Jan. 14th, 2022; accepted: Feb. 10th, 2022; published: Feb. 18th, 2022

Abstract

In order to solve the problems of high transmission delay and high core network in mobile cloud computing, the main solution is to deploy edge servers on roadside units by calculating sinking. This computing mode of offloading to edge servers or routers is called fog computing. In this paper, two kinds of task offloading algorithms in fog computing environment, flooding algorithm and threshold-based switching algorithm, are proposed. The two kinds of offloading algorithms are analyzed on the platform of Matlab R2017a, and compared with local execution and offloading to

cloud server execution. The results show that the task offloading algorithm in fog computing environment can not only reduce the load of cloud servers, but also effectively shorten the task completion time. Threshold-based switching algorithm in fog computing environment can better reduce task completion time especially.

Keywords

Fog Computing, Task Offloading, Flooding Algorithm, Calculation Handoff

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

根据工信部发布的通知数据显示：2021年，全球物联网连接设备将超过180亿台，而我国的物联网设备数量在2020年就达到100亿台，预测将于2025年增长到220亿台，成为万亿级别的市场。随着移动互联网设备的爆炸式增长，如何处理资源受限设备的海量数据成为了该领域的研究热点，移动云计算(Mobile Cloud Computing, MCC)是过去十年解决设备资源受限主流方案[1]。

移动云计算通过对网络的计算、存储资源进行整合，实现随时随地按需获取资源的使用。计算卸载(Computing Offloading) [2]就是指把任务从资源受限的设备迁移到资源丰富的设备上进行处理。移动云计算主要是利用云服务器的计算资源和存储资源等优势，弥补移动设备的资源缺陷，给用户提供了更好的交互体验。随着与日俱增的密集型任务需要计算，移动云计算中云服务器流量过载、高传输延时等问题日益严重。

来自学术界和工业界的研究人员一直在积极推进运用计算下沉的方式在路侧单元上部署边缘服务器，使得处于网络边缘的路侧单元具有云计算能力。这种将计算能力、通信能力、存储能力推向网络系统边缘的模式称为雾计算(Fog Computing) [3]。相对于云计算模式，雾计算使得计算、通信等功能更加贴近用户设备，不仅可以减少数据传输的时延，还可以满足较高的移动性。

在具体计算中，我们要考虑到设备的移动性可能会影响到设备到代理节点的通信链路，进而影响到任务卸载的完成时间。为解决此问题，文中提出了基于泛洪[4]的任务卸载方案和基于阈值切换的[5]任务卸载方案，都能缩短任务完成时间，并且有效地提高边缘层的任务卸载效率。

在本文其他部分，介绍了卸载场景，描述了实验算法，并引入泛洪和计算切换的概念，最后经过实验对不同方案的平均任务执行时间和平均任务完成时间进行了对比。

2. 雾计算卸载决策

2.1. 卸载场景与架构

如图1为雾计算的卸载场景，网络分为移动设备群-边缘层-中心云层三层计算架构。移动设备的任务请求不仅可以通过宏基站访问云服务器，还可以在贴近移动设备的边缘服务器上执行。

如图2为雾计算的卸载架构[6]。在移动设备收到任务卸载请求时，设备的资源发现模块就能通过目标选择模块根据目标选择策略搜寻周围可用代理资源，然后根据卸载分割决策做出相应的卸载决策。



Figure 1. Fog computing offloading scenario
图 1. 雾计算卸载场景

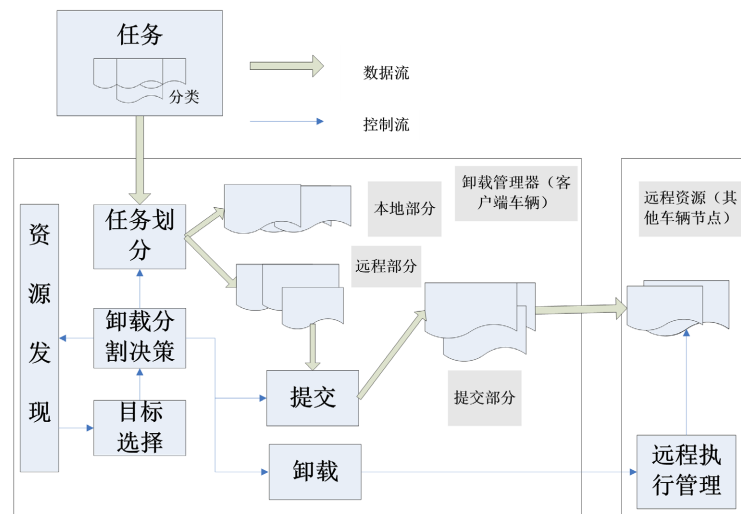


Figure 2. Offload architecture of fog computing
图 2. 雾计算的卸载架构

2.2. 节点运动模型

本文选用 Random Waypoint 模型[7]作为节点运动模型, 该模型是一种无记忆状态的模型, 即上一次的运动决策不会影响下一次的运动决策, 运动模型简单且易于分析, 适用于对普适性要求高的研究中。实现过程为:

- 1) 假设仿真区域二维运动空间 A , 在区域内随机选取起始点 S 和目的点 D ;
- 2) 在 $[V_{\min}, V_{\max}]$ 之间随机选择一个速度值 V , 节点按照速度 V 从 S 点匀速直线运动到 D 点;
- 3) 到达终点 D 后, 在 $[T_{\min}, T_{\max}]$ 之间随机选择一个暂停时间 T , T_{\min} 、 T_{\max} 分别为最小暂停时间和最大暂停时间;
- 4) 然后该节点继续运动, 并将 D 作为运动的新起点 S_{new} ;
- 5) 循环 1)~4) 步骤。

2.3. 数学模型

1) 网络模型

任务的卸载场景可用有向连通网络 $G(V, E)$ 表示, V 为该图顶点的集合, E 为边的集合, 对于每一条边 $(V_i, V_j) \in E$, 边的方向就是流量流动的方向。

2) 任务模型

任务表示为 $Task = \{D_{in}, D_{out}, C, T_{max}\}$, C 表示任务的计算量, T_{max} 代表任务的截止时间。

3) 节点模型

节点表示为 $Node = \{ID, P, R, X, Y\}$, 其中 ID 表示节点的编号, P 表示节点的计算能力, R 表示节点的通信半径, X, Y 表示节点的坐标位置。

4) 等效带宽

通信链路选择遵循最大等效带宽原则, 即在可选链路中寻找等效带宽最大的链路作为任务的传输链路。等效带宽 BW 由网络单跳带宽 $BW_i (1 \leq i \leq h)$ 和路由跳数 h 共同决定, 计算方法如公式(1)所示:

$$BW = \frac{BW_1 \times BW_2 \times \cdots \times BW_h}{BW_1 + BW_2 + \cdots + BW_h} \quad (1)$$

5) 目标函数

最小执行时间如公式(2)所示:

$$\min(T_{computing}) \quad (2)$$

最小完成时间如公式(3)所示:

$$\min(T_{up} + T_{computing} + T_{down}) \quad (3)$$

6) 约束条件

任务上传、下载、计算时间需满足公式(4):

$$(T_{up} + T_{computing} + T_{down}) \leq T_{max} \quad (4)$$

2.4. 卸载算法

本小节主要介绍实验涉及的两种任务卸载算法: 泛洪算法和基于阈值切换算法。

1) 泛洪算法

在泛洪算法中, 任务可以卸载至所有部署边缘服务器的小基站上, 并不需要根据任务属性、代理节点的初始状态和节点的初始位置做出当前最优决策。

假定一共有 w 个边缘服务器, 遵循此算法, 任务的完成时间如公式(5)所示:

$$T = \min(T_{m1}, T_{m2}, \cdots, T_{mw}) \quad (5)$$

如果边缘服务器 $w' \in (1, \cdots, w)$, 则任务在边缘服务器 w' 上的完成时间如公式(6)所示:

$$T_{w'} = \frac{D_{in}}{BW_{uw'}} + \frac{C}{F_{w'}} + \frac{D_{out}}{BW_{dw'}} \quad (6)$$

$BW_{uw'}$ 表示上传时代理节点至移动设备的等效带宽, $BW_{dw'}$ 表示下载时代理节点 w' 至移动设备的等效带宽, $F_{w'}$ 表示代理节点 w' 的计算能力。

2) 基于阈值切换的算法

当任务在执行阶段, 资源发现模块需要根据每一抽样时刻的任务完成状态、代理节点状态、移动设

备的位置做出切换决策。

切换原则如公式(7)所示:

$$T_1 > T_2 + T_{switch} + I_m \quad (7)$$

其中, T_1 表示没有切换的任务完成剩余时间, T_2 表示发生计算切换的任务完成剩余时间, T_{switch} 代表切换费用。 I_m 为改善因子, 即阈值。为了防止频繁切换计算引起的乒乓效应, 在此次试验中 I_m 值的大小根据切换效果进行合理的设置。

T_1 的计算方法如公式(8)所示:

$$T_1 = \frac{C_{res}}{F_{old}} + \frac{D_{out}}{BW_{old}} \quad (8)$$

其中, C_{res} 表示任务未计算量, F_{old} 表示切换前代理节点的计算能力, BW_{old} 表示切换前代理节点到移动设备的等效带宽。

T_2 的计算方法如公式(9)所示:

$$T_2 = \frac{C_{res}}{F_{new}} + \frac{D_{out}}{BW_{new}} \quad (9)$$

其中, F_{new} 表示切换后代理节点的计算能力, BW_{new} 表示切换后代理节点至移动设备的等效带宽。

T_{switch} 的计算方法如公式(10)、(11)、(12)、(13)所示:

$$T_{switch} = T_{package} + T_{a \rightarrow b} + T_{reboot} \quad (10)$$

$$T_{package} = \frac{C_b}{C} \times T_{package}^C \quad (11)$$

$$T_{a \rightarrow b} = hops \times d \quad (12)$$

$$T_{reboot} = \frac{C_b}{C} \times T_{reboot}^C \quad (13)$$

其中, $T_{package}$ 表示任务未计算的打包时间, $T_{a \rightarrow b}$ 表示边缘服务器 S_a 到 S_b 的时延, T_{reboot} 表示边缘服务器 S_b 的重启时间。 $T_{package}^C$ 表示总计算量的任务打包时间, T_{reboot}^C 表示总计算量的任务重启时间, d 表示路侧单元间的单跳时延。

另外, 本次实验采用基于 Docker 的容器迁移技术来减少计算迁移的时间[8], 图 3 为计算切换示意图。

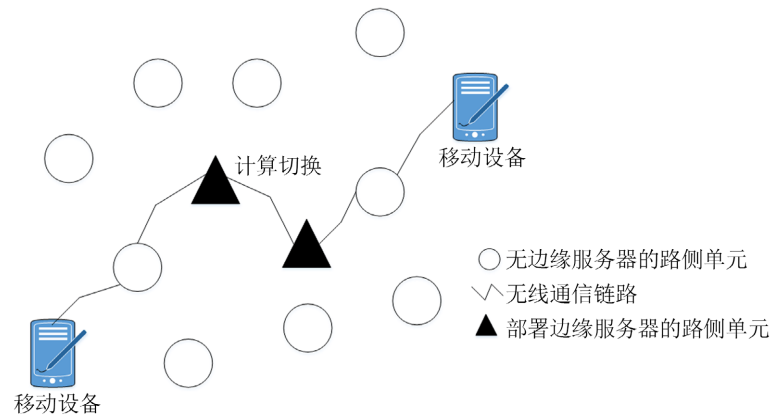


Figure 3. Calculation handoff diagram

图 3. 计算切换示意图

3. 实验仿真与结果分析

本实验基于 Matlab R2017a 仿真平台进行，并对实验结果进行了详细的分析：

- 1) 根据实验结果，对比本地执行、卸载到云服务器计算和卸载到边缘服务器计算的平均任务执行时间；
- 2) 根据平均任务完成时间，对比本地执行、卸载到云服务器、洪泛算法和基于阈值切换的。

3.1. 环境设定

本次实验假设其它物体对通信没有影响，时间抽样间隔为 0.1 秒，设备移动速度满足道路标准。卸载模式为 MD2MEC 和 MDC2 两种。

任务为可间断执行任务，如视频流任务，即任务未完成部分不受到已完成部分影响。

3.2. 参数设置

仿真次数设为 1000 次，其它参数设置如表 1 所示。

Table 1. Parameter settings

表 1. 参数设置

参数名	参数值	参数名	参数值
BS 数目(个)	1	MD2RUS 带宽(Mbps)	10
C 数目(个)	1	MD2BS 带宽(Mbps)	2
RSU 数目(个)	16	RUS2RUS 带宽(Mbps)	5
MEC 数目(个)	3	RUS2C 带宽(Mbps)	1.28
BS 半径(m)	1000	MD2RUS 延时(s)	0.01
RSU 半径(m)	[100, 150]	RUS2RUS 延时(s)	0.1
任务数据量(M)	[40, 60]	MD2C 延时(s)	0.2
结果数据量(M)	[40, 60]	MD 计算能力(MIPS)	82335×0.75
任务计算量(MIPS)	$1.8 \times 5 \times 10^6$	MEC 计算能力(MIPS)	$82335 \times [1.251.75]$

3.3. 性能参数

本文采用的性能指标为如下，其中任务完成时间最为重要。

- 1) 平均任务执行时间

$$\text{平均任务执行时间} = \text{任务执行总时间} / \text{任务个数}$$

其中，任务执行时间表示该任务在节点上的计算时间，即任务的计算量与资源计算能力的比值。

- 2) 平均任务完成时间

$$\text{平均任务完成时间} = \text{任务完成总时间} / \text{任务个数}$$

其中，任务的完成时间 = 任务的上传时间 + 计算时间 + 下载时间 + 通信时延。

3.4. 结果分析

实验的统计结果保留 2 位小数，对平均任务执行时间和平均任务完成时间情况下的任务卸载方案结果进行对比分析。图 4 为不同方案的平均任务执行时间对比图，图 5 为不同方案的平均任务完成时间对比图。

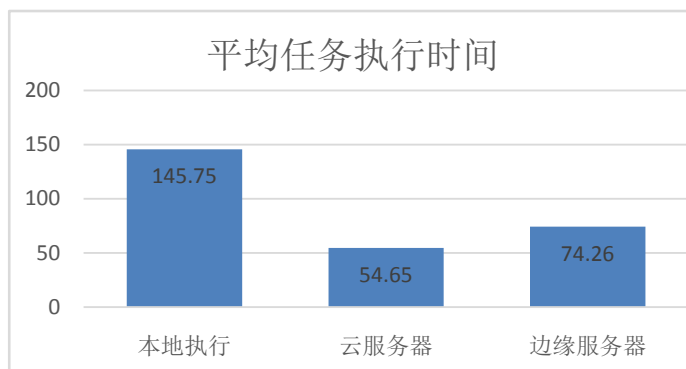


Figure 4. Average task execution time result graph
图 4. 平均任务执行时间结果图

由图 4 可知，任务交给云端服务器执行，时间最短，为 54.65 秒。按照平均任务执行时间的大小依次排序为：云服务器 > 边缘服务器 > 本地执行。

如果根据平均任务执行时间的结果进行任务卸载，那么卸载方案将导致所有任务都卸载至云服务器上，从而造成云服务器负载过大等问题。因此，我们可以考虑把任务卸载到边缘服务器运行。

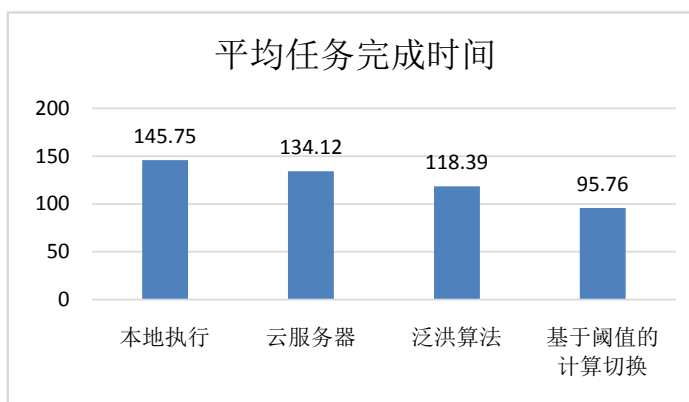


Figure 5. Average task completion time result graph
图 5. 平均任务完成时间结果图

由图 5 可知，基于阈值切换的平均任务完成时间最短，为 95.76 秒。按照平均任务完成时间的大小依次排序是：本地执行 > 云服务器 > 泛洪算法 > 基于阈值切换的。

由仿真结果可得，泛洪算法和基于阈值切换的任务卸载方案均可降低任务完成时间，但基于阈值切换的卸载方案效果更优。

4. 结论与总结

本文针对雾计算环境下的任务卸载方法进行了研究，对比分析任务在本地执行、云服务器和边缘服务器的执行时间和完成时间。另外，文中又提出了雾计算环境下的两种卸载方案，即基于泛洪和基于阈值切换的。实验结果可得：本文提出的两种任务卸载算法，都可以减少任务的完成时间，而且基于阈值切换的卸载算法性能较好。

在以后的研究中，可重点研究雾计算下的多任务多节点的卸载算法，在节点可用的情况下，找到最佳卸载节点。

参考文献

- [1] 田广东, 葛东玉. Ad Hoc 云中基于移动预测的多准则任务卸载算法[J]. 计算机应用研究, 2020, 37(1): 283-286.
- [2] Liu, J.H. and Zhang, Q. (2018) Offloading Schemes in Mobile Edge Computing for Ultra-Reliable Low Latency Communications. *IEEE Access*, **6**, 12825-12837. <https://doi.org/10.1109/ACCESS.2018.2800032>
- [3] 叶卓映. 雾计算技术探讨[J]. 数据通信, 2018(5): 43-44.
- [4] Yan, H., Xu, Z., He, J., *et al.* (2017) Striped-Flooding: Improve Scalability and Energy Efficiency of Flooding Algorithm in Wireless Sensor and Actor Networks. 2016 *IEEE 84th Vehicular Technology Conference (VTC-Fall)*, Montreal, 18-21 September 2016, 1-5. <https://doi.org/10.1109/VTCFall.2016.7881240>
- [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(2): 37-42.
- [6] 李波, 黄鑫, 薛端, 侯严严, 裴以建. 基于 DTN 的车载云计算卸载算法[J]. 云南大学学报(自然科学版), 2018, 40(2): 215-221.
- [7] Silva, R.T., Colletti, R.R., Pimentel, C., *et al.* (2016) BETA Random Waypoint Mobility Model for Wireless Network Simulation. *Ad Hoc Networks*, **48**, 93-100. <https://doi.org/10.1016/j.adhoc.2016.06.001>
- [8] 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.