

面向 6G 的低时延高可靠边缘计算架构研究*

丁昱华, 陈力[†], 卫国

(中国科学技术大学 电子工程与信息科学系, 合肥 230022)

(2022 年 12 月 6 日收稿; 2023 年 4 月 3 日收修改稿)

丁昱华, 陈力, 卫国. 面向 6G 的低时延高可靠边缘计算架构研究[J]. *Journal of University of Chinese Academy of Sciences*, DOI:10. 7523/j.ucas. 2023. 029.

摘要 MEC(mobile edge computing)是 6G 移动通信网络打通通信与服务, 实现万物智联的支撑技术。针对 MEC 系统的计算时延优化, 提出了横向多主机架构, 设计了完整的信令流。针对 MEC 系统的传输时延优化和多主机并行计算的掉队者问题, 提出了多连接主从多主机架构, 设计了完整的信令流。针对 MEC 系统的性能评估, 搭建了基于开源库的多主机 MEC 仿真平台。实验表明, 提出的横向多主机 MEC 架构有效提高了计算时延性能; 提出的多连接主从多主机 MEC 架构有效缓解掉队者问题, 提高传输时延性能; 搭建的 MEC 仿真平台能够有效评估多主机架构的关键性能指标。

关键词 6G; 边缘计算; 低时延架构; 信令设计; 计算卸载

中图分类号: TN929.5

文献标志码: A

DOI:10. 7523/j.ucas. 2023. 029

Research on ultra-reliable low-latency edge computing architecture for 6G

DING Yuhua, CHEN Li, WEI Guo

(Department of Electronic Engineering and Information Science, University of Science and Technology of China,
Hefei, 230022, China)

Abstract MEC (Mobile Edge Computing) is the enabling technology for the 6G mobile communication network to connect communication and service and realize the smart connection of everything. For the computational delay optimization of the MEC system, a horizontal multi-host architecture is proposed and a complete signaling flow is designed. For the transmission delay optimization of the MEC system and the straggler problem of multi-host parallel computation, a master-slave architecture of multi-connectivity is proposed and a complete signaling flow is designed. For the evaluation of MEC system performance, a multi-host MEC simulation platform based on the open-source library is built. The experiments show that the horizontal multi-host MEC architecture effectively improves computational latency performance; the proposed master-slave MEC architecture of multi-connectivity effectively alleviates the straggler problem and improves transmission latency performance; the built MEC simulation platform can effectively evaluate the key performance indicators of the multi-host architecture.

Keywords 6G; edge computing; low latency architecture; signaling design; computation offloading

6G 移动通信网络^[1]将实现全新的范式转变, 有望 达成万物智联、支撑全社会深度数字化转型的愿景。

* 国家重点研发计划课题(2021YFB2900302), 国家自然科学基金面上基金(62071445)资助

[†] 通信作者, E-mail: chenli87@ustc.edu.cn

6G 将不仅仅追求传输速率和网络容量的突破,更要实现通信、计算、感知一体化,实现信息、通信和数据技术的深度融合^[2]。面对在网络边缘接入的海量设备产生的海量数据以及广泛的低时延高可靠需求,移动边缘计算是支撑 6G 网络建设的关键技术^[3]。MEC 将云计算能力分布式下沉至网络边缘,相比于传统的移动云计算,将极大提升数据的处理效率,降低由数据长距离路由和传输引入的时延。

MEC 领域的关键问题得到了广泛的研究。软件定义网络、网络功能虚拟化^[4]是支撑 MEC 系统与移动通信网络融合的关键技术。文献[5]着眼于 MEC 的移动性管理问题,为 MEC 的服务连续性提供了保障。MEC 系统的计算资源与网络资源分配的联合优化^[6]实现了系统的低时延性能与资源负载平衡。用户设备向 MEC 系统计算卸载策略的设计^[7]平衡了能量消耗与时延性能。MEC 的安全与隐私问题也越来越受到关注^[8]。

然而,以上的研究都没有关注复杂架构下的多主机 MEC 系统,在实际的计算密集型、多用户应用场景

下难以达成低时延高可靠的性能指标,相应的多主机 MEC 架构仿真工作因此也存在空缺。对多主机 MEC 系统讨论的不足限制了时延的优化空间,难以实现进一步的突破。基于以上讨论,为了实现复杂多主机架构下 MEC 系统时延优化,本文提出了横向多主机架构与多连接主从多主机架构,并设计了相应的完整信令流。搭建了多主机 MEC 架构的仿真平台验证性能表现。使用与真实部署环境的相符合的系统级 5G-MEC 仿真,通过将 UE 端的高维矩阵乘法运算通过 5G 移动网络卸载至 MEC 主机进行计算这样一个贴近现实场景的计算卸载任务,仿真验证了所提出的多主机架构显著降低了 MEC 系统的端到端时延。

1 研究现状

MEC 在网络边缘端提供了计算与存储能力,能够有效降低计算卸载的时延^[9]。MEC 将计算资源下沉至 RAN(Radio Access Network)内,如图 1(a)所示,用户设备的计算卸载任务可以在接入的边缘节点内完成计算。

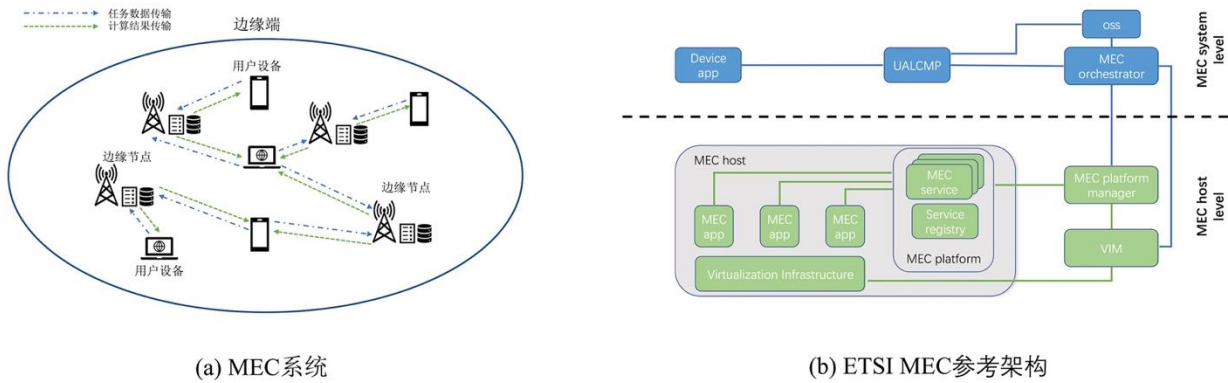


图 1 MEC 系统与 ETSI MEC 参考架构
Fig.1 MEC system and ETIS MEC reference architecture

用户设备与 MEC 系统的交互主要有以下三个步骤:

步骤 1 任务卸载: 数据上行传输过程,用户设备将计算卸载任务所必要的的数据上行传输至特定的 MEC 主机。

步骤 2 服务调用与计算: MEC 主机根据计算卸载任务的调用相应的 MEC 服务,计算产生用户设备

所需要的结果数据。

步骤 3 结果回传: 数据下行传输过程, MEC 系统将用户所需的计算结果下行传输至用户设备。

ETSI(european telecommunications standards institute)规范化了 MEC 系统,为其制定了参考架构^[10]。在这套参考架构下,各个功能实体的功能与交互都得到了良好的定义。ETSI 将 MEC 系统层次化为 MEC

系统层与 MEC 主机层。

MEC 系统层：主要负责响应终端用户请求，维护 MEC 系统的全局信息。

MEC 主机层：负责为 MEC 系统提供软、硬件资源并维护软、硬件资源。

ETSI MEC 已经成为了当下较为通行的重要 MEC

标准。然而如图 2(a)所示的参考架构中的单主机架构资源有限，将成为优化低时延 MEC 的瓶颈。

文献[11]利用 ETSI MEC 参考架构的标准化接口在边缘端部署了基于 LoRa(long range radio)协议的 IoT(internet of things)应用，相比于在中心云上的部署，减少了数据获取与处理的延迟。文献[12]提出了一种基

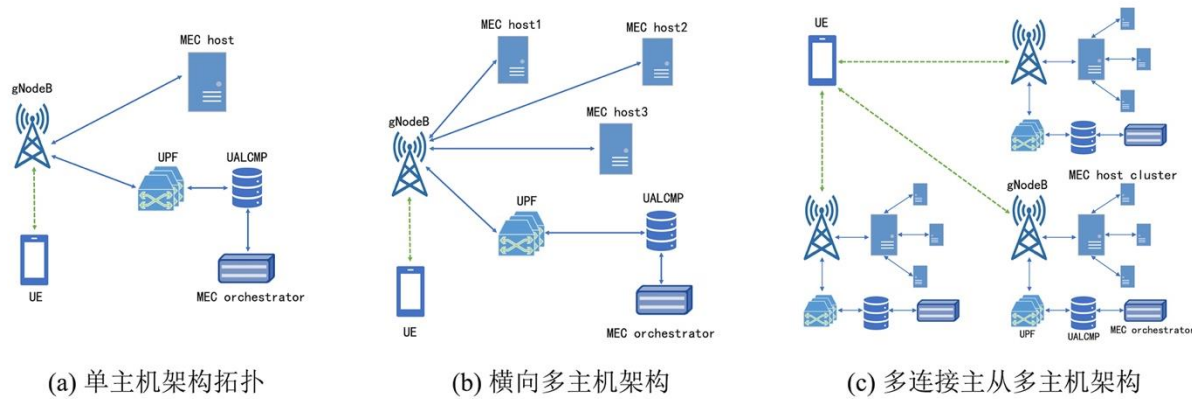


图 2 单主机架构与多主机架构拓扑
Fig.2 Single-host vs. multi-host architecture topologies

于 ETSI MEC 参考架构的车辆碰撞检测算法及其测试平台实现，MEC 的低时延特性为算法带来很好的有效性。文献[13]提出了一种在支持 ETSI MEC 的蜂窝网络中进行 MEC 流量与互联网流量识别与优先级排序算法，避免 MEC 流量拥塞从而进一步降低时延。文献[14]对比了在 LTE(long term evolution)、5G NSA(non-standalone)、5G SA(standalone)部署环境下集成 ETSI MEC 系统的性能表现，仿真说明 MEC 系统在 5G SA 部署环境下拥有更加优异的时延性能表现。但是以上工作均在单主机架构下进行讨论，忽略了多主机复杂架构可能带来的时延优化潜力。

2 多主机架构与信令流

卸载至边缘端的计算任务通常是计算复杂度极高的任务，需要充足的计算资源。使用 MEC 的代价是要付出相应的通信时延，如果计算任务在本地设备执行的时延与计算卸载的通信时延相比处于相同的量级，那么将计算任务卸载至边缘是不合适的选择。部署在基站单元内的 MEC 系统是面向多用户的，一旦基站单

元所服务的用户量增大，用户将对边缘端有限的资源进行竞争，会造成排队时延。根据计算卸载对计算资源的大量需求和潜在的多用户对边缘资源的竞争，部署在边缘端的单 MEC 主机的资源有限，单主机架构将不能满足越来越多用户设备的接入与计算密集型低时延应用需求。

然而多主机的并行计算会引入掉队者问题^[15]，这是一个分布式计算场景下的重要问题。由于计算节点的工作负载不均衡、计算节点本身就存在不可控的异质性问题，各计算节点完成各自任务所耗费的时间不尽相同。但是完整任务的结果依赖所有参与计算节点生成的子结果，也就导致了任务计算总时延取决于所有子节点中计算时间最长的节点。仅仅增加主机数量也无法降低计算卸载的传输时延。计算卸载任务通常都会有不可忽视的数据量，因此计算卸载的过程中存在不可忽视的通信时延，对低时延的 MEC 提出了巨大的挑战。

2.1 横向多主机架构

2.1.1 架构设计

为了降低计算卸载中的计算时延,提出一种横向多主机架构如图 2(b)所示。单主机架构进行横向拓展,为一个基站单元配备多台平行的 MEC 主机,即为横向多主机架构。多台 MEC 主机的部署能够为基站单元带来更充足的资源为用户提供服务。而且,利用某一类卸载任务的可拆分性^[9],这种架构可以带来更灵活的计算卸载方案,即拆分一个完整计算任务,将拆分后的子任务卸载至不同主机进行运算,从而降低用户请求 MEC 系统的计算时延。

2.1.2 信令流设计

对于横向多主机架构,我们设计了完整的端到端

信令流如图 3(a)所示。从 UE(user equipment) app 至 MEC 编排器逐级传递 N 个 MEC app 实例化请求信令, MEC 编排器按需选择合适的 N 台 MEC host, N 台 MEC host 成功实例化终端用户所需的 MEC app 后,逐级回传 N 个 MEC app 进程实例的 IP(internet protocol)地址、端口号等信息信令。UE app 获得完整的回传信息信令后, UE app 会分拆计算卸载任务,直接与 N 个 MEC app 通信,向其发送待计算的分拆子任务数据。 N 个 MEC app 完成子任务计算后回传子结果, UE app 接收回传子结果并根据子结果恢复整个计算卸载任务的总结果,信令流结束。

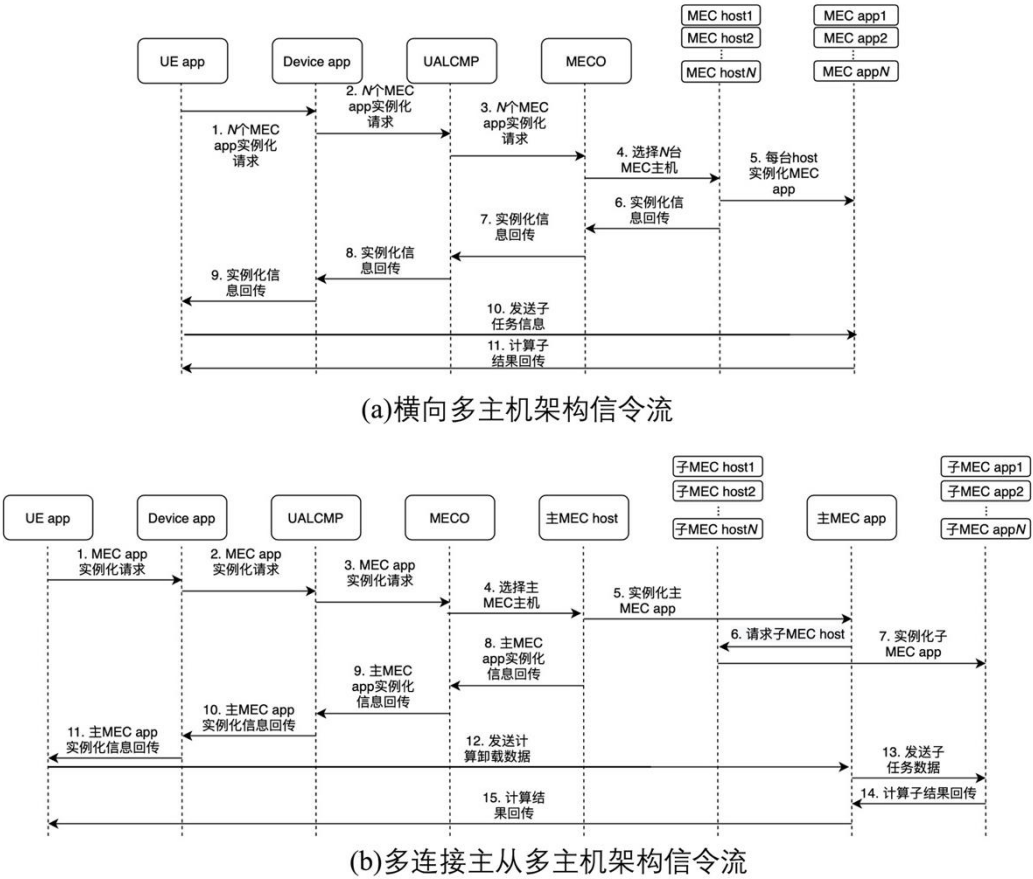


图 3 横向多主机架构与多连接主从多主机架构信令流
Fig.3 Signaling flow of horizontal multi-host architecture and master-slave architecture of multi-connectivity

2.2 多连接主从多主机架构

2.2.1 架构设计

为了进一步降低计算卸载中由掉队者问题引入的计算时延,同时优化计算卸载过程的传输时延,提出一种多连接主从多主机架构。多连接技术允许用户设备同时接入多个基站,实现向多个基站的并行传输,是一

种有希望实现超高可靠超低时延通信的技术^[16]。通过编码计算方法来缓解掉队者问题是一种非常有效的解决方案^[17]。编码计算的基本思想是通过增加一定量的额外计算来换取计算时延的鲁棒性。在这个过程中存在主从节点的结构,主节点负责调控整体的编码、任务分配与结果恢复工作,子节点负责计算主节点分配的

计算子任务。主从多主机架构实现了编码计算所需要的主从节点结构,为编码计算技术的引入提供了可能。结合多连接与编码计算技术,多连接主从多主机架构如图 2(c)所示。多个基站单元构成一个组,组内基站可供用户设备进行多连接。同一分组内的每一个基站单元都被分配多台 MEC 主机,这些 MEC 主机形成层次化的主从关系结构,构成 MEC 主机集群,集群内分为主节点与工作子节点。主节点负责获取分配到该 MEC 主机集群的计算任务,将计算任务进行编码后,合理地将编码后任务分配给集群内的工作子节点进行计算。通过编码计算技术,不必等到每一个被分配计算任务的工作子节点都完成计算,主节点就能够获取到足够数量的子结果得到这一集群应该产生的计算结果。因此,掉队者问题得到缓解,提高了边缘端计算的鲁棒性,能够有效降低计算时延。另一方面,用户设备多连接至多个基站单元,用户设备端进行一次计算任务拆分后能够并行传输计算卸载的数据,这将有效降低整个计算任务卸载过程中的通信时延。

2.2.2 信令流设计

针对多连接主从多主机架构,我们设计了多连接的任意一条连接链路上完整的端到端信令流,如图 3(b)所示。从 UE app 至 MEC 编排器逐级传递 MEC app 实例化请求信令,MEC 编排器首先选择该条接入链路上的主 MEC host,并向这台主 MEC host 发送实例化终端用户所需主 MEC app 信令,主 MEC app 将负责接收 UE 发送的数据信息并对数据进行拆分、编码与分发。主 MEC app 紧接着会向从属于主 MEC host 的各个子 MEC host 发送实例化子 MEC app 请求信令,子 MEC app 将负责计算主 MEC app 下发的计算任务。最终主 MEC host 向 UE app 逐级回传主 MEC app 实例的 IP 地址、端口号等信息信令。因此,UE app 的交互主体为主 MEC app,并不会与子 MEC host 与子 MEC app 产生直接交互,减轻了用户设备端的复杂度。UE app 获得完整的回传信令后,根据链路数拆分完整的任务卸载数据,再向每条链路上的主 MEC app 发送待计算的数据,每条链路上的 MEC 主机在计算完成后立即回传

子结果,UE app 接收回传子结果,并汇总子结果恢复整个计算任务的结果,信令流结束。

3 仿真与分析

本章我们介绍搭建系统级仿真平台的开源组件库,给出仿真中的重要参数,验证仿真平台的有效性,并分析仿真结果。具体的,用户设备通过 5G 网络向 MEC 卸载 1200 维矩阵-向量乘积计算任务,边缘端完成计算后向用户设备回传结果向量。这个任务包含了上行计算卸载数据传输、边缘端计算与下行结果传输过程,可以对 MEC 系统端到端时延进行精确评估。

3.1 仿真平台搭建

OMNeT++^[18]是一个可扩展的、模块化的、基于组件的 C++ 仿真库和仿真框架,主要用于构建网络仿真。INET^[19]是基于 OMNeT++ 的开源模型库,提供了计算机网络协议、无线和有线网络链路层协议等,以符合 OMNeT++ 的形式将协议封装为各种 module 组件。Simu5G^[20]是一个基于 OMNeT++ 的 5G 仿真组件库,提供了符合 3GPP 标准的链路层和物理层协议模型。Simu5G 拥有定义良好的接口,能够与 INET 库兼容,允许用户实现任意复杂度的端到端、全链路、基于 5G 蜂窝网络接口的 TCP/IP 网络仿真。更重要的是,Simu5G 提供了符合 ETSI 标准的 MEC 相关组件和功能,为仿真 5G-MEC 系统提供了支持。本文基于以上所介绍的开源组件库搭建了仿真平台,并在仿真平台上构建了所提出的多主机架构仿真。

3.2 计算任务选取

选取高维矩阵乘法作为计算任务进行仿真与分析。图像、视频渲染计算是边缘计算的典型应用场景之一。渲染管线^[21]的逐步推进主要包括以下几个流程:顶点变换和属性构建、图元处理、光栅化、片元计算。顶点变换和属性构建阶段涉及大量模型的平移、旋转、缩放、空间映射操作,这些操作都是通过矩阵乘法描述与实现的,在渲染计算的计算复杂度中占据主要部分^[22]。

分布式机器学习是边缘计算的另一个典型应用场景，其中矩阵乘法是机器学习算法中的关键^[17]。矩阵乘法是众多边缘计算典型应用场景下的基础运算，而且矩阵乘法的计算复杂度极高，因此选取高维度矩阵乘法运算作为仿真的计算任务具有普适性与代表性。

3.3 仿真参数

在 5G NR 的 Sub 6G 场景下，最大可选带宽为 100 MHz。当前 5G 组网最常见的子载波间隔为 30 KHz。更大的带宽和更大的子载波间隔能够带来更高的数据传输速率，因此选定传输带宽为 100 MHz，子载波间隔为 30 KHz。横向多主机架构选用两台 MEC 主机，多连接主从多主机架构选用双连接，每一条链路内拥有一个主节点和三个工作子节点，采用 (3,2)MDS(maximum distance separable)编码。其余主要仿真参数^[20]如表 1 所示。

表 1 主要仿真参数
Table 1 Main simulation parameters

| 序号 | 参数 | 量值 |
|----|----------|--------|
| 1 | gNB 发射功率 | 20 dBm |
| 2 | gNB 高度 | 25 m |
| 3 | gNB 天线增益 | 8 dBi |
| 4 | gNB 噪声系数 | 5 dB |

| 时间(s) | 步骤 |
|-------|-----------------------------|
| 0.10 | UE app启动, 请求MEC服务 |
| 0.12 | MEC编排器选择MEC host, 启动MEC app |
| 0.22 | MEC app进程信息信令回传 |
| 0.23 | 分拆子任务数据下发 |
| 0.27 | 子结果接收 |

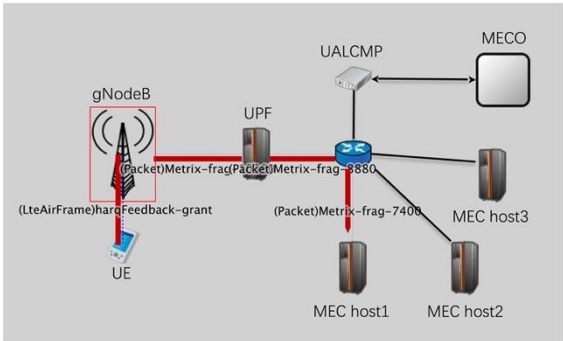
(a) 仿真运行关键步骤

| | | |
|----|-----------|----------------|
| 5 | UE-gNB 距离 | 100 m |
| 6 | UE 高度 | 1.5 m |
| 7 | UE 天线增益 | 0 dBi |
| 8 | UE 噪声系数 | 7 dB |
| 9 | 带宽 | 100 MHz |
| 10 | 多径衰落 | Enabled |
| 11 | 路径损耗模型 | 3GPP-TR 36.873 |
| 12 | 目标误块率 | 0.01 |
| 13 | 误块率 | 0.1 |
| 14 | 阴影衰落 | Enabled |
| 15 | 载波频率 | 2 GHz |
| 16 | 子载波间隔 | 30 KHz |

3.4 仿真结果分析

(1) 仿真平台有效性分析

以横向多主机架构在仿真平台运行结果为例，整理了仿真平台运行生成的日志，总结仿真运行过程中关键步骤发生的时间节点如图 4(a)所示。根据该架构下设计的信令流，仿真平台能够准确输出所需的观测数据，能够有效地完成相应任务来对多主机 MEC 架构进行仿真。由于仿真过程中的关键步骤信息大多以文字日志的形式进行输出，无法在仿真界面框内进行可视化，图 4(b)展示了一个能够可视化展示的步骤，UE 正在向其中一台 MEC host 下发子任务数据。



(b) 下发子任务数据至MEC host1

图 4 仿真平台功能验证
Fig.4 Simulation platform functional verification

(2) 多连接主从多主机架构对掉队者问题的优化

掉队者问题存在于多主机架构之下^[15]。多连接主从多主机架构为引入编码计算技术优化掉队者问题提供了架构支持。两种多主机架构下 1200 维矩阵-向量乘法任务的主机节点计算时延如图 5 所示。在仿真中，

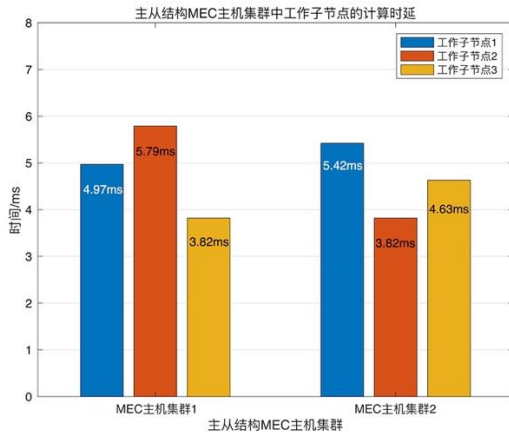
多连接主从多主机架构选用双连接，一个 MEC 主机集群拥有三个工作子节点，因此引入(3, 2)MDS 编码^[23]，将集群的计算任务拆分为两部分，由这两部分编码生成第三部分，分配给三个工作子节点进行并行计算。因此在一个集群内，只要有任意两个工作节点完成计算

任务,这一集群的计算结果就可以恢复。图 5(a)给出了多连接主从多主机架构在仿真中参与计算的两个 MEC 主机集群内各个工作节点的计算时延。MEC 主机集群 1 内计算时延最短的节点耗费 3.82ms,完成计算任务的计算时延为 4.97ms; MEC 主机集群 2 内计算时延最短的节点耗费 3.82ms,完成计算任务的计算时延为 4.63ms。因此多连接主从多主机架构下总计算时延为 4.97ms,落后者效应使计算时延性能下降 30.1%。横向多主机架构无法缓解掉队者问题。在仿真中,横向多主机架构选用两台 MEC 主机。图 5(b)给出了两台平

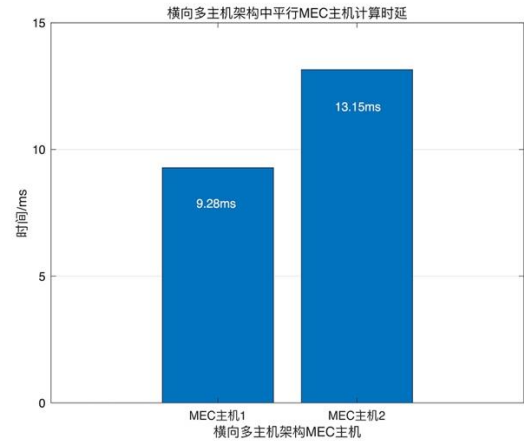
行 MEC 主机的计算时延,分别为 9.28ms 与 13.15ms,因此横向多架构下总计算时延为 13.15ms。落后者效应使计算时延性能下降 41.7%。多连接主从多主机架构优化了并行计算中掉队者问题带来的计算时延性能下降问题,在实际部署中使用冗余度更高的 MDS 码将带来更大的性能增益。

(3) 上行传输时延对比

在不同架构下仿真测得的上行传输 1200 维方阵的传输时延如图 6(a)所示。单主机架构和横向多主机架构在用户设备与基站间的无线传输阶段都是单链路传



(a) 多连接主从多主机架构工作子节点计算时延



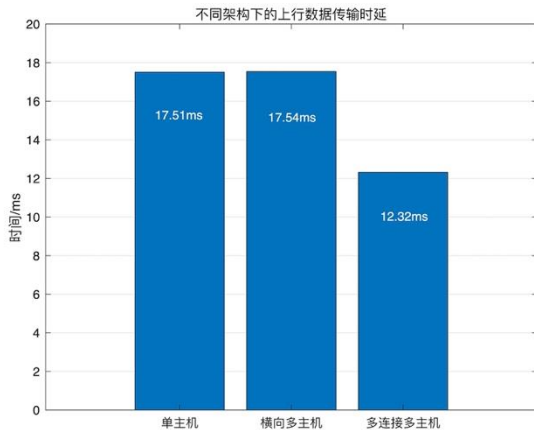
(b) 横向多主机架构平行 MEC 主机计算时延

图 5 两种多主机架构下参与计算的主机的计算时延

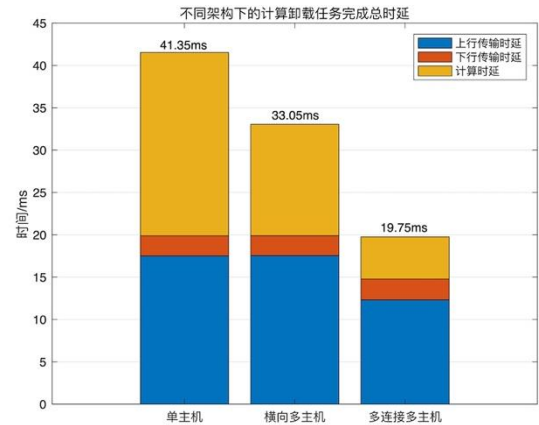
Fig.5 Computational latency of the hosts involved in the two multi-host architectures

输,因此两种方案的上行传输时延是一致的。多连接主从多主机架构在无线传输阶段为双链路并行传输,明显降低了上行数据传输时延。但由于存在主从架构,主

节点收到数据后还要对数据进行拆分、编码等工作,接着分配给所属的工作节点,因此最终的上行传输时延并非前两种方案的一半。



(a) 上行传输时延对比



(b) 计算卸载任务完成总时延对比

图6 上行传输时延与计算卸载任务完成总时延对比
Fig.6 Comparison of uplink transmission delays and completion delays of computation offloading

(4) 计算卸载时延架构间横向对比

不同架构下通过 5G 网络向 MEC 卸载 1200 维矩阵-向量乘法任务的完成总时延如图 6(b)所示。单主机架构花费了最长的时间完成整个任务。横向多主机架构在一定程度上降低了计算时延,但是由于主机间的异质性问题,导致计算时延没有下降至单主机架构的 1/2。由于卸载任务数据量不变,横向多主机架构仍然是单链路上行传输,传输时延没有得到优化。在多连接主从多架构下,传输时延和计算时延同时得到了优化。多连接使得上行数据可以在多条链路上并行传输,有效降低了数据的传输时延。MEC 主机主从关系结构的加入让编码计算策略得以实施,有效缓解了多节点并行计算中的掉队者问题。性能最好的多连接主从多主机架构成功将千维矩阵乘法计算卸载任务的时延控制在 20ms 以内。

4 结论

本文针对低时延 MEC 系统架构设计,提出了两种多主机架构,设计了两种架构下完整的信令流,搭建多主机 MEC 架构仿真平台以评估架构的关键性能指标。针对计算时延的优化问题,提出了横向多主机架构,设计了该架构下完整信令流。针对传输时延和掉队者问题的优化,提出了多连接主从多主机架构,设计了该架构下完整的信令流。实验与仿真表明,横向多主机架构降低了计算卸载的计算时延,多连接主从多主机架构进一步优化了计算卸载的传输时延和并行计算的掉队者问题,搭建的仿真平台能够有效评估多主机 MEC 架构的时延性能。仿真平台的代码已在 GitHub 平台开源[‡]。下一步将继续探索性能更优的 MEC 系统复杂架构设计,扩充多主机 MEC 架构仿真平台的功能。

参考文献

- [1] You X H, Wang C X, Huang J, et al. Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts[J]. Science China Information Sciences, 2021, 64(1): 110301. DOI: 10.1007/s11432-020-2955-6.
- [2] Liu G Y, Li N, Deng J, et al. The SOLIDS 6G mobile network architecture: Driving forces, features, and functional topology[J]. Engineering, 2022, 8: 42-59. DOI: 10.1016/j.eng.2021.07.013.
- [3] Ji B F, Wang Y N, Song K, et al. A survey of computational intelligence for 6G: Key technologies, applications and trends[J]. IEEE Transactions on Industrial Informatics, 2021, 17(10): 7145-7154. DOI: 10.1109/TII.2021.3052531.
- [4] Taleb T, Samdanis K, Mada B, et al. On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1657-1681. DOI: 10.1109/COMST.2017.2705720.
- [5] Wang J R, Liu K Y, Ni M M, et al. Learning based mobility management under uncertainties for mobile edge computing[C]//2018 IEEE Global Communications Conference (GLOBECOM). December 9-13, 2018, Abu Dhabi, United Arab Emirates. IEEE, 2019: 1-6. DOI: 10.1109/GLOCOM.2018.8647718.
- [6] Wang J D, Zhao L, Liu J J, et al. Smart resource allocation for mobile edge computing: A deep reinforcement learning approach[J]. IEEE Transactions on Emerging Topics in Computing, 2019, 9(3): 1529-1541. DOI: 10.1109/TETC.2019.2902661.
- [7] Chen M, Hao Y X. Task offloading for mobile edge computing in software defined ultra-dense network[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(3): 587-597. DOI: 10.1109/JSAC.2018.2815360.
- [8] Ali B, Gregory M A, Li S. Multi-access edge computing architecture, data security and privacy: A

[‡] <https://github.com/eva-ding/simu5G-for-MEC>

- review[J]. IEEE Access, 2021, 9: 18706-18721. DOI: 10.1109/ACCESS.2021.3053233.
- [9] Mao Y Y, You C S, Zhang J, et al. A survey on mobile edge computing: The communication perspective[J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322-2358. DOI: 10.1109/COMST.2017.2745201.
- [10] Sabella D, Vaillant A, Kuure P, et al. Mobile-edge computing architecture: The role of MEC in the Internet of Things[J]. IEEE Consumer Electronics Magazine, 2016, 5(4):84-91. DOI: 10.1109/MCE.2016.2590118.
- [11] Ksentini A, Frangoudis P A. On extending ETSI MEC to support LoRa for efficient IoT application deployment at the edge[J]. IEEE Communications Standards Magazine, 2020, 4(2): 57-63. DOI: 10.1109/MCOMSTD.001.1900051.
- [12] Avino G, Giordanino M, Franzoudis P A, et al. A MEC-based extended virtual sensing for automotive services[C]//2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE). July 2-4, 2019, Turin, Italy. IEEE, 2019: 1-6. DOI: 10.23919/EETA.2019.8804512.
- [13] Huang P H, Hsieh F C, Hsieh W J, et al. Prioritized traffic shaping for low-latency MEC flows in MEC-enabled cellular networks[C]//2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC). January 8-11, 2022, Las Vegas, NV, USA. IEEE, 2022: 120-125. DOI: 10.1109/CCNC49033.2022.9700715.
- [14] Virdis A, Nardini G, Stea G, et al. End-to-end performance evaluation of MEC deployments in 5G scenarios[J]. Journal of Sensor and Actuator Networks, 2020, 9(4): 57. DOI: 10.3390/jsan9040057.
- [15] Dean J, Barroso L A. The tail at scale[J]. Communications of the ACM, 2013, 56(2): 74-80. DOI: 10.1145/2408776.2408794.
- [16] Thiruvassagam P K, Chakraborty A, Murthy C S R. Resilient and latency-aware orchestration of network slices using multi-connectivity in MEC-enabled 5G networks[J]. IEEE Transactions on Network and Service Management, 2021, 18(3): 2502-2514. DOI: 10.1109/TNSM.2021.3091053.
- [17] Lee K, Lam M, Pedarsani R, et al. Speeding up distributed machine learning using codes[J]. IEEE Transactions on Information Theory, 2018, 64(3): 1514-1529. DOI: 10.1109/TIT.2017.2736066.
- [18] 赵永利, 张杰. OMNeT++与网络仿真[M]. 北京: 人民邮电出版社, 2012.
- [19] Virdis A, Kirsche M. Recent advances in network simulation: the OMNeT++ environment and its ecosystem[M]. Cham, Switzerland: Springer, 2019. DOI:10.1007/978-3-030-12842-5.
- [20] Nardini G, Sabella D, Stea G, et al. Simu5G-an OMNeT++ library for end-to-end performance evaluation of 5G networks[J]. IEEE Access, 2020, 8: 181176-181191. DOI: 10.1109/ACCESS.2020.3028550.
- [21] Ganovelli F, Corsini M, Pattanaik S, et al. Introduction to Computer Graphics: A Practical Learning Approach[M]. Florida: Chapman and Hall/CRC, 2014. DOI: 10.1201/b15978.
- [22] 王彬. 高自由度视频编码及渲染关键技术研究[D]. 杭州: 浙江大学, 2020.
- [23] Sathiamoorthy M, Asteris M, Papailiopoulos D, et al. XORing elephants: novel erasure codes for big data[EB/OL].(2013-01-16)[2022-12-06] <https://doi.org/10.48550/arXiv.1301.3791>.