doi:10.3969/j.issn.1003-3114.2022.01.007
引用格式:孙钰坤,张兴,雷波,等.边缘算力网络中智能算力感知路由分配策略研究[J].无线电通信技术,2022,48(1):60-67.
[SUN Yukun,ZHANG Xing,LEI Bo.Study on Intelligent Computing Aware Route Allocation Policy in Edge Computing-Aware Networks [J].Radio Communications Technology,2022,48(1):60-67.]

边缘算力网络中智能算力感知路由分配策略研究

孙钰坤[1],张　兴[1],雷　波[2]
(1.北京邮电大学 泛网无线通信教育部重点实验室,北京 100876;
2.中国电信股份有限公司研究院,北京 102209)

摘　要:增强现实、自动驾驶、智慧城市、工业互联网等新型业务应用对网络算力的需求逐渐增强,然而,边缘算力网络系统面临着网络共存的问题——负载不均衡,导致一部分边缘服务器无法满足业务应用的处理需求,另一部分边缘服务器的算力资源处于空闲状态。为了高效协同地感知利用泛在、异构的算力资源,提升6G通信网络的内生感知和算力自适应能力,急需对边缘算力网络中任务路由策略以及算力资源分配进行研究。首先介绍了面向6G网络愿景的算力感知网络的演进和需求,然后构建了任务调度智能决策、存储资源和算力资源按需分配的联合优化路由控制与资源分配的智能任务调度模型,最后提出了一种基于Floyd算法的算力感知路由调度策略解决智能任务调度问题。仿真结果表明,与就近调度固定分配资源策略相比,所提的基于Floyd算法的算力感知路由调度策略可以让更多的用户受益,缩短用户业务的平均处理时延,提高边缘算力网络中存储资源和计算资源的利用率。

关键词:算力感知网络;任务调度;融合资源分配;Floyd算法

中图分类号:TP18　　**文献标志码**:A　　**开放科学(资源服务)标识码(OSID)**:

文章编号:1003-3114(2022)01-0060-08

Study on Intelligent Computing Aware Route Allocation Policy in Edge Computing-Aware Networks

SUN Yukun[1],ZHANG Xing[1],LEI Bo[2]
(1.Key Laboratory of Universal Wireless Communications,Ministry of Education,Beijing University of Posts and Telecommunications,Beijing 100876,China;
2.Research Institute of China Telecom Co.,Ltd.,Beijing 102209,China)

Abstract:Emerging network services and applications such as augmented reality,self-driving,smart cities,industrial Internet,put forward higher requirement for computing power.However,computing-aware network is faced with the problem which traditional network is also faced with,i.g,load unbalance.Some edge servers fail to provide enough resources,but the resources of the others are under-utilized.To aware and leverage ubiquitous and heterogeneous computing resource efficiently and collaboratively,and improve 6G communication network endogenous perception and computing adaptive ability,it is urgent to break through computing-aware networking route policy and resource allocation.First,the evolution and demand of computing-aware networking for the vision of 6G networks are analyzed.Then,an intelligent task scheduling model for joint routing control and resource allocation is constructed,which makes intelligent decisions for task scheduling and resources on demand.Finally,the computing aware-route-allocate policy based on Floyd algorithm is put forward.Numerical simulation results show that the computing aware-route-allocate policy based on Floyd algorithm can satisfy more users,shorten average delay for task process,and increase the utilization ratio of storage and computing resources.

Keywords:computing-aware networking;task scheduling;integrated resource allocation;floyd algorithm

收稿日期:2021-10-29
基金项目:之江实验室开放课题(2020LCOAB01);国家自然科学基金(62071063)
Foundation Item: Open Research Fund of Zhejiang Laboratory (2020LCOAB01);National Natural Science Foundation of China (62071063)

0 引言

随着人工智能与移动互联网技术的不断发展，增强现实、人脸识别、图像渲染、自动驾驶等新型业务应用大量涌现，这些新型业务应用通常需要消耗巨大的计算资源、存储资源以及能耗，目前智能终端设备的计算能力尚且比较有限，电池容量也比较低，无法满足这些新兴业务应用的处理需求。因此，云计算得以提出并持续升温。

云计算利用虚拟化技术建立超大容量的算力资源池，使得各种应用可以获得所需的计算资源、存储资源以及软件和平台服务。云计算的出现满足了计算密集型的业务处理需求，但是，自动驾驶这一类智能应用同时具有时延敏感的特性，终端到云端的传输时延在很多情况下无法满足这一类应用对于超低时延的需求。因此，ETSI 于 2014 年 12 月成立移动边缘计算（Mobile Edge Computing，MEC）行业规范组（Industry Specification Group，ISG），启动移动边缘计算标准化，以发展移动边缘计算[1]。ETSI 将 MEC 定义为一种可以在无线接入网络中移动用户附近位置提供 IT 和云计算功能的网络架构，旨在将 IT 和云计算从核心网络迁移到边缘接入网络，以缩短任务处理端到端的时延，并确保数据的安全性与隐私性。2016 年 9 月移动边缘计算的概念被扩展为多接入边缘计算（Multi-access Edge Computing，MEC），将移动边缘计算从电信蜂窝网络进一步延伸至其他无线网络，以扩大边缘计算在包括 WiFi 和固定访问技术在内的异构网络中的适用性。

边缘计算设备和智能终端设备的大量部署，虽然解决了网络中海量数据上传至云计算中心导致的带宽紧缺、网络拥塞、时延过长的问题，但也使得算力资源呈现泛在部署的趋势，不可避免地出现了“算力孤岛”效应。一方面，边缘计算节点没有进行有效的协同处理任务，单一节点的算力资源无法满足如图像渲染等超大型的计算密集型任务的算力资源需求，仍然无法解决同时具有计算密集和时延敏感特性的新型业务的超低时延需求问题；另一方面，虽然一些边缘计算节点出现超负载无法有效处理计算任务的情况，但是由于网络负载的不均衡，势必会有一些计算节点仍然处于空闲的状态，导致边缘网络的算力资源无法得到充分的利用。

因此，为了高效、协同地利用全网异构的算力资源，2019 年由运营商、设备商等主导提出一种基于分布式系统的计算与网络融合的技术方案——算力感知网络[2-5]（Computing-aware Networking，CAN），以实现 ICT 系统的联合优化调度，提供端到端的体验保证。CAN 旨在将云边端多样的算力通过网络的方式连接与协同，实现计算与网络的深度融合及协同感知，以及算力资源的按需调度和高效共享[6-10]。

算力感知路由和算力资源分配是算力感知网络研究中的一个关键问题，在传统的网络架构中，算力与网络通常是单独进行管理。在算力管理方面，计算卸载技术作为边缘计算的一项关键技术，在边缘计算概念被提出之后，便有许多研究者提出基于单用户多节点[11]、多用户单节点[12]以及多用户多节点[13-14]的任务卸载策略，这些策略实质上都是将终端任务与边缘计算节点进行完美匹配。在网络管理方面，研究主要集中在如何优化任务路由策略[15]。

针对目前研究存在的问题，本文在多任务、多路由节点、多边缘服务器的边缘算力网络场景下，分别基于用户业务侧、网络侧、边缘计算资源池侧，构建算力需求、网络状态、算力资源的感知信息模型；基于感知信息模型，在计算资源和存储资源的约束限制下，以用户业务的调度传输时延最小化为优化目标，联合优化路由控制、计算节点的选择和存储、算力资源的分配；最后，本文提出了基于 Floyd 的算力感知路由分配策略，有效地将算力与网络进行协同管理，缓解网络负载不均衡的问题，并通过仿真分析了所提策略在改善用户业务体验以及算网资源利用率方面的性能。

1 任务调度系统建模

1.1 算力感知网络系统概述

如图 1 所示，算力感知网络系统由终端设备、基站以及边缘网关（路由节点）、边缘服务器（边缘计算节点）和云中心组成。其中用户业务由终端设备发起请求，通过无线链路传输至基站；边缘网关主要负责路由控制和数据转发，在实际网络系统中，边缘网关可以部署在基站侧，路由节点之间可以通过实时动态的链路进行数据传输；边缘服务器是以硬件基础设施为虚拟化资源的边缘应用平台，主要负责提供存储和计算资源，进行用户业务的处理，边缘计算节点与路由节点之间通过固定链路进行数据传输；云中心具有充足的存储资源和计算资源，是部署在距离用户较远位置的大型服务器集群，边缘计算节点与云中心节点之间具有固定的链路进行数据传输，虽然在本文的系统模型中，假设用户业务只在由

边缘计算节点组成的边缘算力感知网络中进行处理，但是云中心是算力感知网络架构中不可或缺的一部分，因此在图 1 中予以表示。

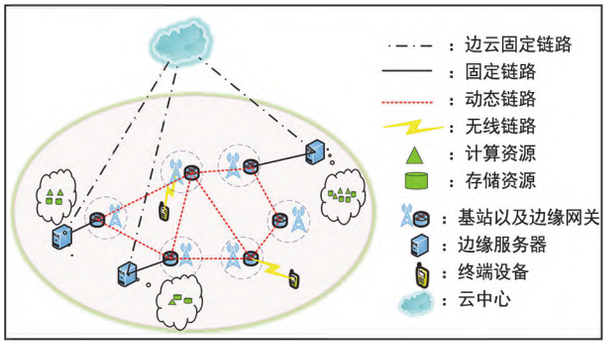

图 1　算力感知网络系统

Fig.1　System of computing-aware networks

1.2　任务模型

假设算力感知网络系统中共有 M 个终端设备，终端设备的集合表示为 $\mathrm{M}=\{1,2,\ldots,M\}$，$m \in \mathrm{M}$ 表示一个终端设备，假设每个终端设备每次最多发起一个计算任务处理请求。

假设算力感知网络系统中共存在 K 个用户业务，用户业务的集合表示为 $\mathrm{K}=\{1,2,\ldots,K\}$，$k \in \mathrm{K}$ 表示一个用户业务。第 k 个用户业务可以表示为 $S_k(A_k,X_k,C_k,D_{k\max},D_{ka},B_k)$，其中 A_k 表示用户业务 k 所接入的路由节点，X_k 表示用户业务 k 所需要的存储资源的量化值，C_k 表示用户业务 k 所需要的计算资源的量化值（中央处理器的转数/比特数据/秒，即单位时间处理单位比特数据所需要的中央处理器的转数），$D_{k\max}$ 表示用户业务 k 允许的最大业务处理时延，D_{ka} 表示用户业务 k 接入路由节点的时延，B_k 表示用户业务 k 的大小。

1.3　通信模型

终端设备发出请求的用户业务通过无线链路传输到算力感知网络中的边缘无线接入点，根据香农定理，用户业务 k 到所接入的路由节点 A_k 的数据传输速率可以表示为：

$$R_{k,A_k}=B_{k,A_k}\mathrm{lb}(1+\gamma_{k,A_k}), \tag{1}$$

其中，B_{k,A_k} 表示用户业务 k 到所接入的路由节点 A_k 信道的带宽，γ_{k,A_k} 表示用户业务 k 到所接入的路由节点 A_k 传输的信噪比（Signal Noise Ratio，SNR），表示为：

$$\gamma_{k,A_k}=\frac{p_{k,A_k}h_{k,A_k}}{N_0}, \tag{2}$$

式中，p_{k,A_k} 表示用户业务 k 到所接入的路由节点 A_k 的发射功率，N_0 表示加性高斯白噪声的谱密度，h_{k,A_k} 表示用户业务 k 到所接入的路由节点 A_k 的路径损耗，h_{k,A_k} 的大小与用户业务 k 到所接入的路由节点 A_k 之间的距离有关，距离越远，路径损耗越大。本文主要研究终端设备发出的业务请求在算力感知网络中如何基于感知的信息路由调度到最佳的算力节点上进行业务处理，因此没有考虑用户业务的位置分布，而是将用户业务到所接入的路由节点的数据传输速率进行统一的量化表示。

综上所述，用户业务 k 到所接入的路由节点 A_k 的接入时延为：

$$D_{ka}=\frac{B_k}{R_{k,A_k}}。 \tag{3}$$

终端设备发出请求的用户业务在算力感知网络中，通过动态的通信链路在相邻的路由节点之间进行数据传输，假设在算力感知网络中路由节点之间动态链路的数据传输速率可以感知，则表示为：

$$W=\begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,N} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,N} \\ \vdots & \vdots & & \vdots \\ w_{N,1} & w_{N,2} & \cdots & w_{N,N} \end{bmatrix},$$

式中，$w_{i,j}$ 表示路由节点 i 与路由节点 j 之间的数据传输速率，满足

$$w_{i,j}=\begin{cases} \infty, & i==j \\ \text{limited rate}, & i\neq j \text{ 且 } i \text{ 和 } j \text{ 之间有链路连接。} \\ 0, & i\neq j \text{ 但 } i \text{ 和 } j \text{ 之间无链路连接} \end{cases} \tag{4}$$

因此，用户业务 k 在路由节点 i 和路由节点 j 之间的传输时延为：

$$D_{k,i,j}=\frac{B_k}{w_{i,j}}。 \tag{5}$$

算力感知网络中的计算节点与路由节点之间通过固定链路进行数据传输，路由节点 i 到计算节点 n 之间的数据传输速率可以表示为：

$$R_{i,n}=\begin{cases} \infty, & i \text{ 和 } n \text{ 之间存在直接链路} \\ \text{limited rate}, & i \text{ 和 } n \text{ 之间不存在直接链路} \end{cases}。 \tag{6}$$

因此，如果用户业务 k 选择计算节点 n，则通过路由节点传输到计算节点 n 的到达时延为：

$$D_{k,n,a}=\frac{B_k}{R_{i,n}}。 \tag{7}$$

1.4　计算、存储资源模型

在算力感知网络系统中，网络控制面需要实时感知计算节点的存储资源以及算力资源状态。假设网络中存在 N 个计算节点，计算节点的集合表示为 $\mathrm{N}=\{1,2,\ldots,N\}$，$n \in \mathrm{N}$ 表示一个计算节点，则第

n 个计算节点可表示为 $Z_n(A_n,X_n,C_n,R_n)$，其中 A_n 表示该计算节点连接的路由节点标签，X_n 表示该计算节点的存储资源大小，C_n 表示该计算节点的计算资源大小（中央处理器的转数/比特数据/秒，即单位时间处理单位比特数据所需要的中央处理器转数），R_n 表示该计算节点到路由节点的数据传输速率。

综上所述，当计算节点的存储资源、计算资源满足用户业务需求时，用户业务的计算时延为：

$$t_{k,C}=\frac{C_k}{B_k}。\tag{8}$$

2 任务调度优化模型

在上述算力感知网络系统架构模型、任务模型、通信模型、计算和存储资源模型的基础上，一个任务处理的总时延包括任务接入时延、任务传输时延、任务到达计算节点时延、任务处理时延和任务等待时延，由于任务处理时延和任务等待时延在确定的算力需求条件下具有统一性，因此可以在优化模型中不予考虑，所以用户业务 k 到调度的计算节点的传输时延为：

$$t_{k,t}=D_{ka}+D_t+D_{k,n,a}。\tag{9}$$

考虑联合优化计算任务路由策略和算力资源的分配，本文将最终的计算任务调度优化问题建模为：

$$\min_{\{N^*,P^*\}}\frac{1}{K}\sum_{k\in\mathrm{K}}t_{k,t}。\tag{10}$$

目标函数的约束条件如下：

$$\sum_{k\in\mathrm{K}_n}X_k\leqslant X_n,\tag{11}$$

$$\sum_{k\in\mathrm{K}_n}C_k\leqslant C_n,\tag{12}$$

式中，K_n 表示调度到计算节点 n 的所有用户业务的集合，约束条件（11）表示调度到计算节点的所有用户业务可利用的存储资源不超过该计算节点可以提供的存储资源，约束条件（12）表示调度到计算节点的所有用户业务可利用的计算资源不超过该计算节点可以提供的计算资源。

由于当算力感知网络处于部分计算节点过载，或者全网计算节点过载的情况时，根据最小化用户业务传输时延的目标得到的调度策略，部分用户业务势必会因为算力资源短缺而无法按需完成任务处理，考虑实际工程问题的可行性以及算法的复杂度，需要重点关注的不是如何精确地求出最优解，而是在如何保证用户公平性的前提下，高效快速地获得一个联合优化路径选择与计算节点选择的次优解，本文提出一种基于 Floyd 的算力感知路由分配（CARA）算法来解决该问题，为计算任务均衡选择优化的路径以及计算节点。

3 基于 Floyd 的算力感知路由分配算法

Floyd 算法是一种利用动态规划的思想寻找给定的加权图中多源点之间最短路径的算法。在算力感知网络架构中，路由控制器可以感知用户业务的数据大小和网络的链路状态（包括链路的数据传输速率、抖动以及丢包率等）。因此，可以根据网络感知的信息计算出用户业务经过网络中任意一条链路的传输时延，所有的终端设备、路由节点、计算节点和动态的链路时延构成一个实时变化的多源点加权图。

算力感知网络中计算任务调度策略的优化目标之一便是最短传输时延的路径选择，基于 Floyd 算法可以计算出将用户业务路由至各个节点的最短时延以及其对应的具体调度路径，进而选择时延最短的计算节点作为最佳计算节点及根据 Floyd 算法确定的路径作为最优路由策略，如算法 1 所示。

算法 1　基于 Floyd 算法的算力感知路由分配策略

输入：算力感知网络中计算节点状态 Z，计算节点数目 N，算力感知网络中链路状态 W，用户业务的算力需求 S，用户业务数目 K

输出：任务调度路径 $\boldsymbol{P}$，任务处理节点 $\boldsymbol{X}$

1：基于 Floyd 算法计算将用户业务路由至各个计算节点的最短时延 $D_{\min}$，获取最优调度路径 P'

2：根据 $D_{\min}$ 选择时延最短的计算节点以及对应路径作为调度策略，计算最短时延 $D^*_{\min}$，获取最优调度路径 P^* 和最佳计算节点 N^*

3：**for** $k=1:K$ **do**

4：　**for** $n=1:N$ **do**

5：　　**if** $N^*(k)==n$

6：　　**if** $S_{k,2}\leqslant N_{n,2}$ 且 $S_{k,3}\leqslant N_{n,3}$

7：　　最佳计算节点 N^* 不变

8：　　最优调度路径 P^* 不变

9：　　更新节点存储资源状态 $N_{n,2}:=N_{n,2}-S_{k,2}$

10：　　更新节点计算资源状态 $N_{n,3}:=N_{n,3}-S_{k,3}$

11：　　**else**

12：　　重复步骤 6 的判断，依次循环遍历 N 个节点直到资源满足或者回到初始节点停止

13：　　更新最佳计算节点 N^*（回到初始不更新）
14：　　更新最优调度路径 P^*（回到初始不更新）
15：　　更新节点存储资源状态
　　$N_{*,2} := N_{*,2} - S_{*,2}$
16：　　更新节点计算资源状态
　　$N_{*,3} := N_{*,3} - S_{*,3}$
17：　　end if
18：　　end if
19：　　end for
20：end for
21：return　最佳计算节点 N^*，最优调度路径 P^*

然而，算力感知网络作为一种面向 B5G/6G 通信的新型网络架构，与传统的无线网络面临同样的问题，即网络负载不均衡，网络中部分计算节点的计算任务超过所能容纳的最大任务数目，导致计算任务的处理时延过长，无法满足时延敏感型的新型网络业务需求。与此同时，网络中另一部分节点可能没有计算任务达到，处于资源空闲状态，没有充分利用全网的算力资源，导致算力资源利用率低下。因此，仅仅基于 Floyd 算法选择最短时延的路由策略，进而确定计算节点的方案在网络负载不均衡的情况下，无法联合优化计算任务路由策略和算力资源的分配。

如算法 1 中第 3~20 步所示，根据 Floyd 算法已经确定的路由分配策略，如果一个计算任务被分配的计算节点已经过载，则将计算任务循环调度到下一个有剩余算力资源的计算节点。为保证用户的公平性，计算任务按照标签顺序优先获取网络的算力资源；为降低算法的计算复杂度，计算任务在重新选择计算节点时，默认循环至下一个有剩余算力资源的计算节点即可，无需再次基于 Floyd 算法优化调度策略。

假设边缘算力网络系统中路由节点的个数为 R，则本文基于 Floyd 算法的 CARA 策略的算法时间复杂度为 $O(R^3KN^2)$，若使用暴力搜索算法来遍历求解用户业务的计算节点选择、路由控制以及存储、算力资源分配，其时间复杂度为 $O(N^KR^R)$，相比于具有指数级时间复杂度的暴力搜索算法，随着路由节点和用户业务数目的增加，本文方法时间复杂度显著下降。

4　仿真实验

本节对基于 Floyd 算法的算力感知路由分配（CARA）策略进行仿真实验。首先对仿真场景和参数进行说明，然后展示仿真结果并对其进行分析。

4.1　仿真场景与仿真参数设置

在仿真实验中，考虑多终端设备、多路由节点、多边缘计算节点的场景，如图 2 所示。在模拟的算力感知网络系统中，包含 3 个边缘计算节点、6 个边缘路由节点，以及若干发出业务请求的终端设备，其数量可以进行具体的设定，所接入的路由节点随机生成。路由节点之间的链路动态连接，相连两个路由节点之间的数据传输速率动态变化，如图 2 的权值所示，计算节点与路由节点之间的链路连接固定，各个节点之间的链路连接情况以及数据数传速率如图 2 所示。

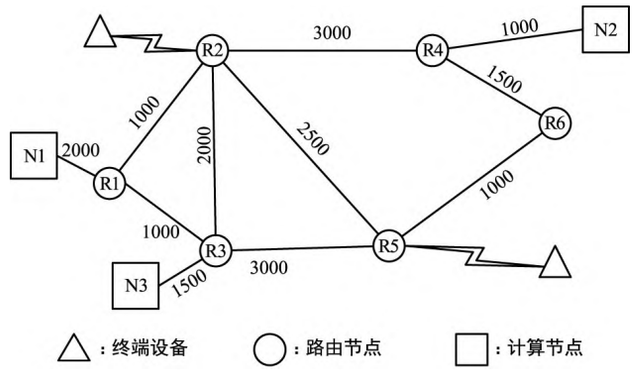

图 2　仿真场景

Fig.2　Simulation topology

如表 1 所示，设置默认情况下的算力感知网络系统参数。

表 1　仿真参数设置

Tab.1　Simulation parameters

参数	数值
任务 k 所需存储资源 X_k/kB	10
任务 k 所需计算资源 C_k/(Cycles/bit/s)	2 000
任务 k 数据量 B_k/bit	1 000
计算节点 N1 存储资源 X_{N1}/kB	40
计算节点 N1 计算资源 C_{N1}/(Cycles/bit/s)	4 000
计算节点 N2 存储资源 X_{N2}/kB	60
计算节点 N2 计算资源 C_{N2}/(Cycles/bit/s)	8 000
计算节点 N3 存储资源 X_{N3}/kB	20
计算节点 N3 计算资源 C_{N3}/(Cycles/bit/s)	4 000
任务 k 接入时延 D_{ka}/s	1
任务 k 允许的最大处理时延 D_{kmax}/s	8

假设 K 个用户业务均为同时发起，算力感知网络的初始状态为无任何任务执行。

以图 2 所示的网络场景作为测试系统，与本文提出的 CARA 策略进行对比的就近调度算法描述如下：用户业务始终选择部署位置最近的计算节点，即接入 R1、R2 的用户业务选择 N1 计算节点，接入 R3、R5 的用户业务选择 N3 计算节点，接入 R4、R6 的用户业务选择 N2 计算节点。

4.2 评价指标

4.2.1 满意用户数

若任务获得所需要的算力资源，而且任务处理完成总时延不超过允许的最大时延，则认为业务处理成功，即：

$$I(k_1) = \begin{cases} 1, \text{若 } t_{k,t} + t_{k,C} \leqslant D_{k\max} \\ 0, \text{若 } t_{k,t} + t_{k,C} > D_{k\max} \end{cases}; \tag{13}$$

$$I(k_2) = \begin{cases} 1, \text{若获取所需资源} \\ 0, \text{若未获取所需资源} \end{cases}; \tag{14}$$

$$I(k) = I(k_1)I(k_2)。 \tag{15}$$

因此，满意用户数为：

$$N_{\text{satisfy}} = \sum_{k \in \mathrm{K}} I(k)。 \tag{16}$$

4.2.2 业务平均处理时延

为了统计所有 K 个用户业务的平均任务完成时延，本文将网络超负载之后没有得到所需算力资源的用户业务处理时间附加等待时延：

$$t_{k,w} = \sum_{k \in \mathrm{K}_{\text{out}}} t_{k,C}, \tag{17}$$

式中，K_{out} 为过载之后没有获取所需的算力资源的用户业务集合。

综上所述，系统中用户业务的平均处理时延为：

$$t = \frac{1}{K} \sum_{k \in \mathrm{K}} (t_{k,t} + t_{k,C} + t_{k,w})。 \tag{18}$$

4.2.3 算力资源利用率

算力感知网络中计算节点的存储资源利用率为网络中使用的存储资源与网络中节点存储资源总和的比值[16]，表示为：

$$U_X = \frac{\sum_{k \in \mathrm{K}} (X_k I(k_2))}{\sum_{n \in \mathrm{N}} (X_n)}。 \tag{19}$$

算力感知网络中计算节点的计算资源利用率为网络中使用的计算资源与网络中节点的计算资源总和的比值[16]，表示为：

$$U_C = \frac{\sum_{k \in \mathrm{K}} (C_k I(k_2))}{\sum_{n \in \mathrm{N}} (C_n)}。 \tag{20}$$

4.3 仿真结果与分析

4.3.1 满意用户数

系统负载均衡时满意的用户数与系统用户总数的关系如图 3 所示，系统负载不均衡时满意的用户数与系统用户总数的关系如图 4 所示。

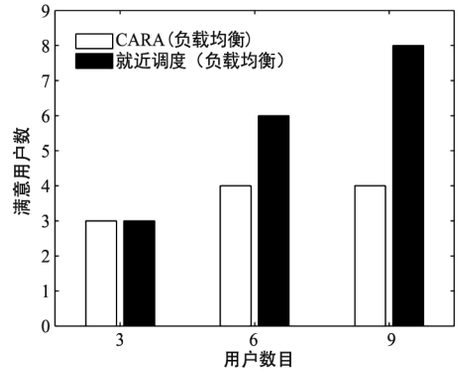

图 3 满意用户数与用户数目关系（负载均衡）

Fig.3 Number of satisfied users vs users（ load balance）

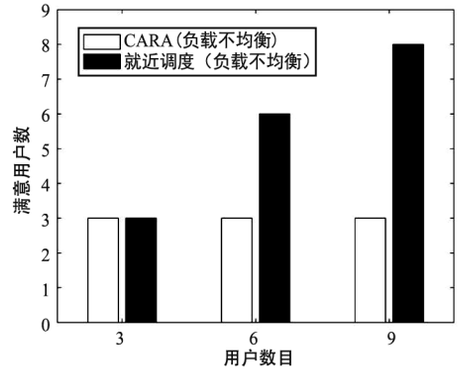

图 4 满意用户数与用户数目关系（负载不均衡）

Fig.4 Number of satisfied users vs users（load un balance）

由图 3 和图 4 可以看出，采取 CARA 策略满意的用户数一直大于或等于就近调度策略，而且随着用户数目的增加或者负载不均衡时，效果更为明显。当负载不均衡时，采取就近调度策略满意用户数目不随着用户总数的增加而增加，很快陷入系统部分饱和状态，而采取 CARA 策略直到系统资源全部耗尽才陷入超负载状态。

4.3.2 业务平均处理时延

算力感知网络中用户业务平均处理时延与系统中用户总数的关系如图 5 所示。随着系统用户数目的增加，CARA 调度策略和就近调度策略产生的平均时延几乎都在增加。在网络负载均衡和负载不均衡两种情况下，采取 CARA 调度策略产生的用户业务平均处理时延均小于就近调度策略，而且随着系统用户数目的增加，采取 CARA 策略缩短的时延效果更为显著。在系统负载不均衡时，采取 CARA 调度策略缩短的时延也更为明显，而且随着系统用户数目增加到一定数量之后，在负载不均衡的情况下

采取 CARA 调度策略的时延，也小于在负载均衡情况下采取就近调度策略的时延。

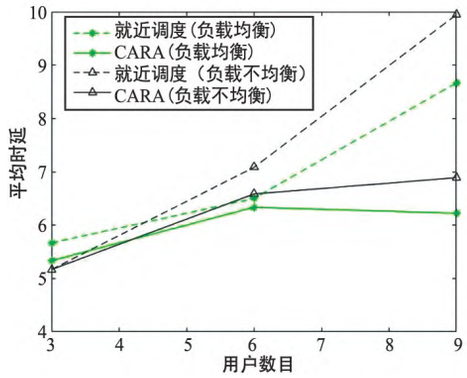

图 5　时延与用户数目关系

Fig.5　Average delay vs the number of users

4.3.3　存储资源利用率

存储资源利用率与系统中用户总数的关系如图 6所示。

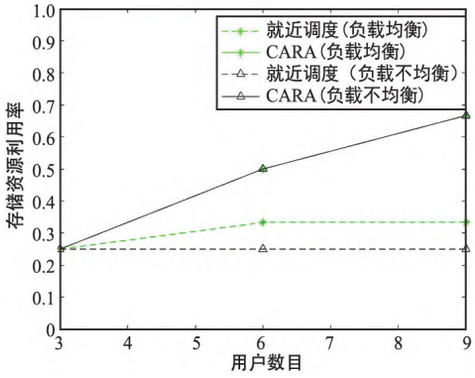

图 6　存储资源利用率

Fig.6　Storage resource utilization ratio

由图 6 可以看出，在系统负载均衡与系统负载不均衡两种情况下，采取 CARA 调度策略时系统的存储资源利用率随着用户数目增加一直在提高，即使在超负载的情况下，存储资源也没有被完全利用，分析原因为此时系统的计算资源已经完全耗尽，存储资源与计算资源是协同处理计算任务的，当存储资源与计算资源没有完全适配计算任务时，会有一种资源无法得到充分利用。采取 CARA 调度策略得到的存储资源利用率随着用户数目的增加越来越高于采取就近调度策略得到的存储资源利用率，而且在系统负载不均衡时，优势更为明显。

4.3.4　计算资源利用率

计算资源利用率与系统中用户总数的关系如图 7所示。

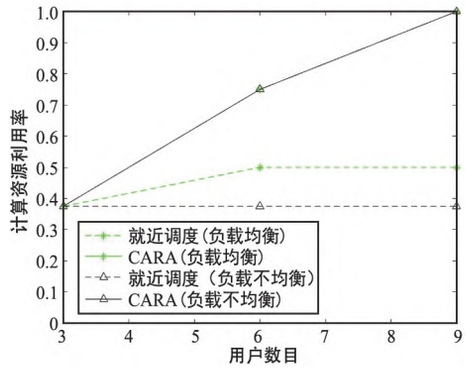

图 7　计算资源利用率

Fig.7　Computing resource utilization ratio

由图 7 可以看出，在系统负载均衡和系统负载不均衡两种情况下，采取 CARA 调度策略时系统的计算资源利用率随着用户数目的增加一直在增加，直到负载达到饱和状态时，计算资源完全利用。与存储资源同样的效果，随着用户数目的增加，采取 CARA 调度策略得到的计算资源利用率越来越高于采取就近调度策略，而且在系统负载不均衡时，优势更为明显。

5　结束语

算力感知网络通过感知算网资源信息以及业务需求信息，将业务对于算力的需求与泛在分布式的算力资源进行映射，并通过无所不至的网络进行连接，动态、弹性地调度算力资源为用户提供服务，提高算网资源的利用率，达到计算密集型以及时延敏感型的业务对于未来边缘网络的需求。本文提出一种算力感知网络中算力感知路由分配系统，定义一个由通信时延组成的目标函数，并建模一个计算任务调度问题。为解决这一问题，提出了一种基于 Floyd 的算力感知路由分配(CARA)策略，联合优化了路由策略与算力资源分配。数值仿真结果表明，CARA 策略能够在满意的用户数目和系统用户业务处理时延方面提供比就近调度策略更优的性能，并且能够提高网络的存储资源和计算资源利用率。

参 考 文 献

[1] MACH P，BECVAR Z.Mobile Edge Computing：A Survey on Architecture and Computation Offloading[J]. IEEE Communications Surveys & Tutorials，2017，19(3)：1628-1656.

[2] 徐勇.IMT-2030(6G)推进组正式发布《6G 总体愿景与潜在关键技术》白皮书[N].人民邮电.2021-06-10.

[3] 中国联通集团研究院.算力网络架构与技术体系白皮书[R/OL].(2020-10-19)[2021-09-15].https://www.doc88.com/p-67316950633601.html? r=1.
[4] 中国移动通信集团研究院.算力感知网络(CAN)技术白皮书[R/OL].(2021-05-28)[2021-08-16].https://www.doc88.com/p-89516049518768.html.
[5] GENG L,CAI H,FU Y,et al.Draft New Recommendation ITU-T Y.IMT2020-CAN-req:Use Cases and Requirements of Computing-aware Networking for Future Networks Including IMT-2020[R/OL].(2020-05-18)[2021-09-16].https://www.itu.int/md/T17-SG13-200720-TD-WP1-0561.
[6] 姚惠娟,陆璐,段晓东.算力感知网络架构与关键技术[J].中兴通讯技术,2021,27(3):7-11.
[7] GENG L,WILLIS P.Compute First Networking (CFN) Scenarios and Requirements[R/OL].(2020-01-08)[2021-09-15].https://www.doc88.com/p-9025927277028.html.
[8] 雷波,刘增义,王旭亮,等.基于云、网、边融合的边缘计算新方案:算力网络[J].电信科学,2019,35(9):44-51.
[9] 曹畅,唐雄燕.算力网络关键技术及发展挑战分析[J].信息通信技术与政策,2021,47(3):6-11.
[10] 黄光平,罗鉴,周建锋.算力网络架构与场景分析[J].信息通信技术,2020,14(4):16-22.
[11] YANG Y,WANG K,ZHANG G,et al.MEETS:Maximal Energy Efficient Task Scheduling in Homogeneous Fog Networks[J].IEEE Internet of Things Journal,2018,5(5):4076-4087.
[12] NATH S,WU J.Dynamic Computation Offloading and Resource Allocation for Multi-user Mobile Edge Computing[C]//GLOBECOM 2020 - 2020 IEEE Global Communications Conference.Taibei:IEEE,2020:1-6.
[13] ALE L,ZHANG N,FANG X,et al.Delay-aware and Energy-Efficient Computation Offloading in Mobile Edge Computing Using Deep Reinforcement Learning[J].IEEE Transactions on Cognitive Communications and Networking,2021,7(3):881-892.
[14] WANG J,HU J,MIN G,et al.Computation Offloading in Multi-Access Edge Computing Using a Deep Sequential Model Based on Reinforcement Learning[J].IEEE Communications Magazine,2019,57(5):64-69.
[15] 李少鹤,李泰新,周旭.算力网络:以网络为中心的融合资源供给[J].中兴通讯技术,2021,27(3):29-34.
[16] 王婷,黄昊楠,张兴,等.空天地一体化网络基于服务功能链的资源分配[J].无线电通信技术,2021,47(5):611-617.

作者简介：

孙钰坤 北京邮电大学博士研究生。主要研究方向：算力网络、移动边缘计算。

张 兴 北京邮电大学信息与通信工程学院教授，博士生导师，IEEE 高级会员，牛津大学高级访问学者。主要研究方向：5G/6G 移动通信系统、移动边缘计算与数据分析、天地一体化信息网络、认知无线电与协同通信。主持/参与国家重点课题 30 余项。在包括 IEEE JSAC/Trans/Magazine 等期刊与会议上发表 100 余篇论文，申请发明专利 50 余项，GreenTouch/IMT-2020(5G)等标准化提案 20 余项。荣获省部级科研奖励 2 项、国际会议最佳论文奖 6 项等。

雷 波 中国电信股份有限公司研究院高级工程师，边缘计算产业联盟 ECNI 工作组联席主席，CCSA“网络 5.0 技术标准推进委员会”管理与运营组组长。主要研究方向：未来网络架构、新型 IP 网络技术等。发表论文 10 余篇，出版《边缘计算与算力网络》《边缘计算2.0：网络架构与技术体系》等专著。