基于安全保障的边缘计算卸载方案

Security-Based Computation Offloading Scheme in Edge Computing Network

廉晓飞/LIAN Xiaofei，谢人超/XIE Renchao，黄韬/HUANG Tao
（北京邮电大学，北京 100876）
(Beijing University of Posts and Telecommunications, Beijing 100876, China)

摘要：提出了基于安全管理的边缘计算卸载方案，并基于量子进化算法（QEA）设计了卸载决策方案。该方案保证了用户在边缘计算网络中进行计算卸载的安全性。仿真结果表明，与常规计算卸载方案对比，本方案能在保证计算卸载安全的情况下有效降低整个系统的开销。

关键词：移动边缘计算；计算卸载；计算卸载决策；资源分配

Abstract: In this paper, a computation offloading scheme based on security management in edge computing network is proposed, which uses quantum evolution algorithm (QEA) to make reasonable offloading decisions. The scheme can perform secure computing offloading in the edge computing network. The simulation results show that this strategy can effectively reduce the cost of the whole system under the condition of ensuring security.

Key words: mobile edge computing; computation offloading; computation offloading decision; resource allocation

DOI：10.12142/ZTETJ.201902007
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20190328.1700.004.html

网络出版日期：2019-03-28
收稿日期：2019-02-10

为了应对终端设备处理能力不足、资源有限等问题，业界在移动边缘计算（MEC）中引入了计算卸载概念[1]。边缘计算卸载即用户终端（UE）将计算任务卸载到MEC网络中，主要解决设备在资源存储、计算性能以及能效等方面的不足。计算卸载最初是在移动云计算（MCC）中提出[2]，在MCC中，UE可以通过核心网（CN）访问远程的集中式云（CC）的计算和存储资源，将计算过程卸载到云端。然而，MCC通过计算卸载的方式虽然为移动用户提供了更快的数据处理能力，降低了设备损耗，但也引入了高延时以及移动无线网络上的额外负载等问题。MEC技术通过将云端服务进一步“下沉”到网络边缘，解决了MCC中计算卸载延时过长、占用网络资源过多等问题。这使得MEC中的计算卸载可以应用在视频服务、自动驾驶和物联网等多个领域。

此外，随着计算卸载技术的应用，边缘网络在安全方面暴露出一系列问题。例如，边缘计算服务器分布式部署方式，使得单点防护能力降低，而多租户的形式会导致恶意用户潜入网内，利用云平台漏洞攻击网络；因此，设计合理的安全措施显得十分必要。计算任务被卸载到边缘网络中，面临更加复杂的网络环境，原本用于云计算的许多安全解决方案也不再适用于边缘计算的计算卸载；因此，设计基于安全机制的边缘计算网络中的计算卸载方案成为解决以上问题的新途径。

1 传统的MEC中计算卸载方案

MEC提供了离用户更近的计

基金项目：国家科技重大专项（2018ZX03001019-003）

算资源和存储资源；而如何利用更近的资源来提升网络性能并减小延时，则主要由计算卸载技术实现。本节将主要介绍传统MEC中的计算卸载流程以及方案，并对其做简要的对比和分析。

1.1 MEC中计算卸载流程

MEC中的计算卸载技术主要包括卸载决策和资源分配。其中，卸载决策是指UE决定是否卸载、卸载多少以及卸载什么的问题。在卸载系统中，UE一般由代码解析器、系统解析器和决策引擎组成，其执行卸载决策分为3个步骤：首先，代码解析器确定卸载内容，具体内容取决于应用程序类型和代码数据分区；然后，系统解析器负责监测控制各种参数，例如可用带宽、卸载数据大小以及执行本地应用程序所耗费的能量；最后决策引擎根据卸载策略确定是否卸载。

完成卸载决策之后，需要解决资源分配问题，即卸载在哪里的问题。如果UE的计算任务是不可分割的或者可以分割但分割的部分存在联系，这种情况下卸载任务就需要卸载到同一个MEC服务器。对于可以分割但分割的任务不存在联系的计算任务，则可以将其卸载到多个MEC服务器。

1.2 MEC中卸载方案分析

目前业界研究MEC中的卸载方案主要基于卸载决策和资源分配2个关键技术点。研究目标主要为降低时延，降低能量消耗。

文献[3]中，作者以降低时延为目标做出合理的卸载决策。在卸载的过程中，首先UE发出卸载请求，然后MEC服务器会给UE返回信道状态信息（CSI），包括应用缓冲队列状态、本地计算和MEC服务器计算消耗的能量以及UE和MEC之间的信道状态信息。UE收到CSI后，根据具体的优化目标做出卸载决策。在文献[3]中由于引入了CSI信息，导致信令开销增大。文献[4]中，作者以优化时延为目标研究了计算卸载过程中的资源分配问题。作者首先对本地计算和MEC服务器计算推导出最优的资源分配算法，然后对于部分卸载模型，采取分段式优化，并证明了最优数据分割策略。基于上述2种结果，找出了最优联合通信和计算资源的分配算法。结果表明，在通信资源充足而计算资源有限的情况下，采用分段式优化的算法能显著较少端到端的延时。文献[5]中，作者提出的以降低时延为目标的最优卸载方案，就是考虑了MEC在计算资源有限的情况下，如何进行卸载决策和资源分配的问题。作者提出了分层的MEC部署架构，并采用Stackelberg博弈论的方法解决了多用户卸载方案。文献[6]中，对于顺序任务，即线性拓扑任务图，作者找到最优的任务卸载到边缘云；而对于并发任务，则采用负载均衡启发式算法将任务卸载到边缘云中，以使UE和MEC服务器之间的并行最大化，达到最小的时延。文献[7]中，作者针对部分卸载模型，提出了任务之间的依赖关系对卸载决策的影响，并且采用多项式时间算法来解决卸载决策的最优方案。

文献[8—10]中，作者以优化能量消耗为目标设计了计算卸载方案，在满足应用时延的同时以优化UE处的能量消耗为目标，提出了2个资源分配方案：第1种策略基于在线学习、网络状态动态的调整，以适应UE的任务要求；第2种策略是预先计算的离线策略，需要每个时隙的数据速率、无线信道状况的信息支持。文献[9]中，作者在离线策略的基础上设计了2种动态离线策略即确定性离线策略和随机离线策略，用于卸载。实验数据也表明，在节能方面有高达78%的提升。文献[10]中，作者提出了在保证时延的情况下对能量进行优化的卸载方案。该方案同时考虑了前传网络和回传网络的链路状况，采用人工鱼群算法进行全局优化。

目前，有关MEC中的计算卸载研究工作大部分没有考虑安全问题。UE在卸载计算任务的过程中会遇到各种各样的安全问题，例如用户恶意卸载、分布式拒绝服务（DoSS）攻击、隐私泄漏等；因此设计一个保证安全的计算卸载方案很有必要。相比于上述研究方案，我们的研究贡献主要有以下几点：

（1）基于MEC网络架构设计基于安全的MEC计算卸载架构，并提出信任管理机制。

（2）在此架构基础上，对MEC计算卸载的过程进行建模分析，以优化整个系统的时延和能耗总的开销为目标，设计了计算卸载策略。

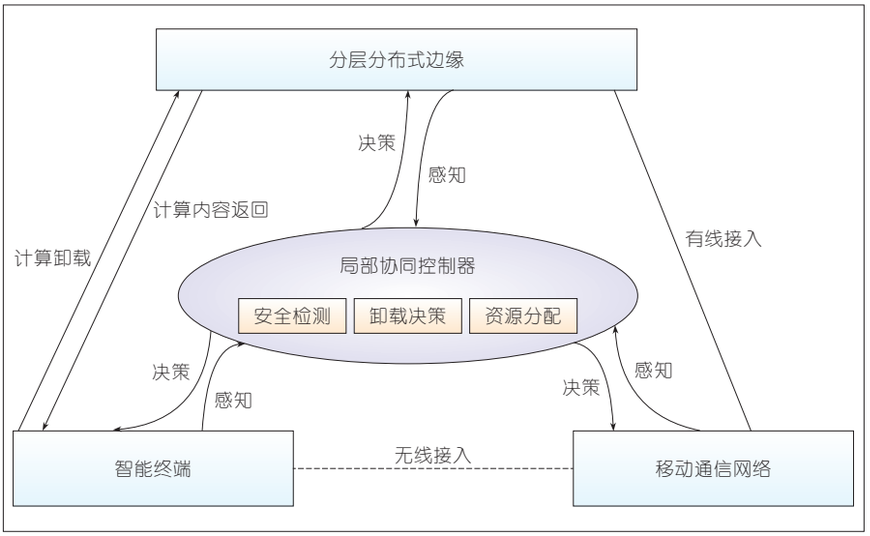
（3）基于量子进化算法（QEA）提出了卸载决策求解算法，并对该

卸载方案进行仿真验证。结果显示，相比于常规卸载策略，本方案不仅使得整个系统开销降低，且提高了安全保障。

2 基于安全的MEC计算卸载方案

2.1 基于安全的MEC架构设计

一般情况下，终端在执行计算卸载任务的过程中，需要不断感知云服务环境的变化，通过与远端服务器通信来制定卸载策略，造成终端能耗浪费和网络资源占用，降低了系统的服务效率。此外，在网络安全方面，没有统一的监测和管控机制，网络安全难以保证。针对这几个问题我们在3层云架构体系中引入局部MEC控制器，可以实时感知网络的资源状况和服务器状态，并与MEC服务器进行交互。局部MEC控制器管理一个接入点下的所有边缘服务器，通过实时获取该区域内网络状态信息、拓扑信息、管理信息以及智能终端信息等构建局部数据库。所有局部控制器由全局控制器集中控制，全局控制器根据局部控制器得到的信息下发合理的资源分配策略或预测策略，并实现资源分配与安全管理控制，以保证用户终端的安全计算卸载。基于安全的MEC网络结构协同机制如图1所示。



▲图1 基于安全的移动边缘计算网络结构协同机制

2.2 基于安全的MEC计算卸载方案设计

本方案的主要目标是设计一种智能解决方案。该方案不仅能解决计算卸载时的安全问题，而且可用于有效地处理MEC卸载时所涉及的动态问题，主要包括对可伸缩性的突然需求、信任状态的预测和估计、计算卸载的智能决策、网络状态的后期分析、更新信任策略等。在本方案中，MEC控制器将有恶意卸载的用户转移到安全监测控制服务器以计算它们每次交互的信任，在不中断网络正常操作的前提下继续监测控制卸载的用户。

终端将卸载的任务信息发送给MEC局部协同控制器，此时会进行安全监测和转移等步骤，然后将决策结果返回给UE。本方案以优化整个系统的能量消耗和时延的均衡为目标，在引入了信任管理机制的同时尽量减少能量消耗。优化能耗主要包括UE能耗和MEC服务器能耗以及信任管理监测控制引入的能耗。时延则是从发起计算任务到执行完并返回结果的时间。

基于上文对卸载方案的描述，基于安全管理的计算卸载方案可转化为卸载决策和资源分配的问题，下面我们将从网络模型、卸载模型、安全模型角度描述建模过程，并得到决策函数。

(1)网络模型。

网络模型由网络不同的边缘节点(EC)组成，每个EC由一组MEC服务器和MEC控制器组成。$C=\{c_1,c_2,...,c_K\}$ 代表不同的边缘节点，每个边缘节点中的MEC集合表示为 $S_{c_k}=\{s_1,s_2,...,c_N\}$，可以为移动终端提供计算卸载服务。UE集合表示为 $N=\{n_1,n_2,...,n_q\}$。

(2)卸载模型。

在该模型中，将每一个时隙内做出卸载决策的判定成为一个策略集，则另 A_t 表示为卸载的决策矩阵，如公式(1)：

$$A_t=\begin{bmatrix} a_{11} & \cdots & a_{1,s} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,s} \end{bmatrix}, \quad (1)$$

其中，$a_{n,s}\in\{0,1\}$ 表示 UE_n 是否卸载计算任务到MEC服务器，$a_{n,s}=1$

代表将 UE_n 的计算任务卸载到MEC服务器，否则代表本地执行。由于每个任务只能由一个MEC服务器执行，因此需要满足公式(2)和公式(3)的约束条件。

$$\sum_{s=1}^{S} a_{n,s} \leq 1 , \tag{2}$$

$$\sum_{n=1}^{N} a_{n,s} \leq 1 。 \tag{3}$$

基于信任管理机制，卸载决策可分为服务器执行计算任务和本地执行任务2种。

● 服务器执行计算任务。

若服务器经过信任监测，决定给某个UE分配计算和通信资源，那么整个计算任务将在服务器执行。这种情况下，整个计算任务在单位时间内所消耗的总能量主要由传送能耗、MEC服务器计算能耗和服务器的监测控制能耗3部分组成，可用公式(4)来表示：

$$E_{s,n} = \frac{P_n B_n}{R_n} + v_{n,s} D_n + E_{monitor} , \tag{4}$$

其中，$n(n=1,2,...)$ 为UE编号，$\frac{P_n B_n}{R_n}$ 代表传送能耗，P_n 代表 UE_n 的传送功率，B_n 代表计算卸载的数据量，R_n 代表传送速率，$v_{n,s} D_n$ 代表MEC服务器的计算能耗，$v_{n,s}$ 表示MEC服务器每个CPU周期消耗的能量，D_n 代表完成该计算任务所需要的CPU，$E_{monitor}$ 表示监测控制能耗。监测控制能耗和信任管理机制中相关参数将会在安全模型中详细介绍。

时延的计算主要由计算时延和传送时延组成，可用公式(5)表示：

$$T_{s,n} = \frac{D_n}{F_{n,s}} + \frac{B_n}{R_n} , \tag{5}$$

其中，$F_{n,s}$ 表示的是MEC的CPU计算能力。

MEC服务器执行的代价函数如公式(6)：

$$Z_{n,s} = \gamma E_{s,n} + (1-\gamma) T_{s,n} , \tag{6}$$

其中，γ 表示时延和能耗的权重，由于不同应用程序的需求不一样，因此权重比例需要根据用户的需求而改变。

● 本地执行计算任务。

若应用程序在本地执行计算任务，能耗主要是指处理计算任务的能耗，可用公式(7)表示：

$$E_{l,n} = v_{n,l} D_n , \tag{7}$$

其中，$v_{n,l}$ 代表本地计算每个CPU周期的能耗，D_n 代表完成该计算任务所需要的CPU。

时延的消耗主要是计算的时延，可用公式(8)表示：

$$T_{l,n} = \frac{D_n}{F_{n,l}} , \tag{8}$$

其中，F 表示UE的CPU计算能力。

本地执行的代价函数如公式(9)所示：

$$Z_{n,l} = \gamma E_{l,n} + (1-\gamma) T_{l,n} 。 \tag{9}$$

(3)安全模型。

通过在MEC控制器中引入安全模块，实现对卸载的计算任务进行安全监测，本文中我们采用X.Qin等人设计的熵检测算法[11]。因为攻击的数据包与正常的数据包很相似，常规检测方法如规则匹配等很难发现异常；而熵检测算法能精准地感知网络参数的变化，然后计算出相对应的信息熵，通过这种方法来具体检测是否是恶意卸载的计算任务。

根据文献[12]可知，卸载的计算任务进行检测时需要消耗能量，我们设计的卸载决策方案直接影响着该能量的变化；因此，找出最小化能耗的卸载决策是基于安全的卸载方案的优化目标。其中涉及到熵检测算法的属性有UE信任度、卸载频率、网络环境和CPU以及内存利用率等。属性z在集合G中的分布属于多项式的分布，概率如公式(10)所示：

$$P(G_z) = \frac{|G|}{\prod_{z=1}^{|G|} z!} \prod_{z=1}^{|G|} G_z^z , \tag{10}$$

$G=\{g_1,g_2,...,g_z\}$，其中 $1 \leq z \leq 5$，G 表示相关属性的集合，G_z 的计算是代表具有属性 z 的用户占整个系统用户的比例，由此可以计算出 $R_{n,i} = \sum_G P$。我们采用最大阈值策略来判断恶意卸载的用户，且这5个参数服从多项分布；因此若大于设定的阈值，则为恶意卸载的用户，卸载决策的判决则是不同意卸载。信息熵的计算公式具体如公式(11)所示：

$$R_n = -\sum_{i=1}^{5} R_{n,i} \log(R_{n,i}) , \tag{11}$$

其中，R_n^H 为检测阈值，因此卸载决策的随着检测结果而变化，具体如公式(12)所示：

$$a_{n,s} = \begin{cases} 1, R_n \geq R_n^H \\ 0, R_n < R_n^H \end{cases} 。 \tag{12}$$

监测控制成本开销如公式(13)：

$$E_{monitor}=\frac{v_{n,s}M_uR_n}{M_s}, \quad (13)$$

$E_{monitor}$ 表示监测控制能耗，它和信任管理机制中的算法涉及参数相关；M_u 表示MEC服务器为一个UE提供的内存资源；M_s 表示整个服务器的内存可用资源。

因此，总的优化函数表示如公式(14-19)所示：

$$\min\left[\sum_{n\in N}\left(\sum_{s\in S}(a_{n,s}Z_{n,s})+(1-a_n)Z_{n,l}\right)\right] \quad (14)$$

$$a_{n,s}\in\{0,1\}\forall n\in N, \quad (15)$$

$$\sum_{s=1}^{S}a_{n,s}\leqslant 1, \quad (16)$$

$$\sum_{n=1}^{N}a_{n,s}\leqslant 1, \quad (17)$$

$$a_{n,s}=\begin{cases}1, R_n\geqslant R_n^H\\ 0, R_n<R_n^H\end{cases}, \quad (18)$$

$$\sum_{n\in N}a_nF_{n,s}\leqslant F_s。 \quad (19)$$

公式(19)的约束条件中，F代表MEC服务器的总的CPU计算资源，MEC服务器分配的计算资源不能超过总的计算资源。

3 算法设计

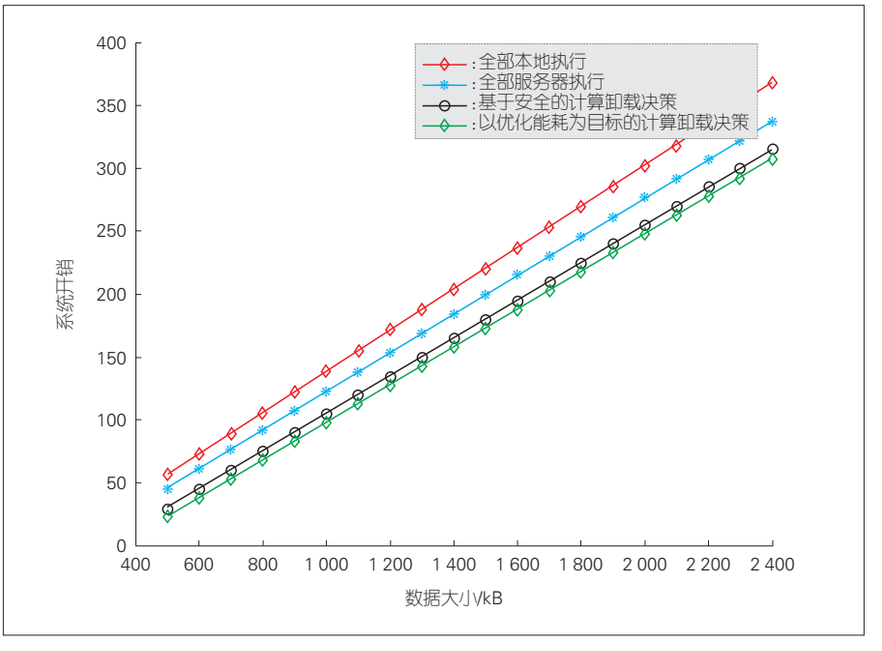
在基于安全的计算卸载方案中，当数据量较大时，该问题是一个NP-hard问题。为了进一步求解该问题，本节采用一个QEA[13]的解决方案来寻找该模型的最优近似解。寻找最优解的过程表示如**算法1**。

算法1 基于量子进化算法的计算卸载算法

```
begin
    当前迭代次数t置为0，并设置最大迭代次数M
    初始化 Q(t)_a 卸载决策矩阵
    通过make子程序观察 Q(t)_a 的状态来确定最优解矩阵 P(t)_a
    通过repair子程序(约束条件)来对 P(t)_a 修正
    评估 P(t)_a 对应的整体开销
    把 P(t)_x 中的最优解存储到 B(t)_a 中
    while(t <M) do
        begin
            当前迭代次数t加1
            通过make子算法观察 Q(t-1)_a 状态来确定 P(t)_a
            评估 P(t)_a 对应的整体开销的最小值
            利用update子程序对 Q(t)_a 进行更新
            把 P(t)_a 和 B(t-1)_a 中的最优解存储到 B(t)_a 中
            把 B(t)_a 中的最优解置为 b_a
            if(当前迭代次数满足迁移条件) then
            把 b_a 或者 b^t_{a,j} 迁移到 B(t)_a 中
            end if
        end
end
```

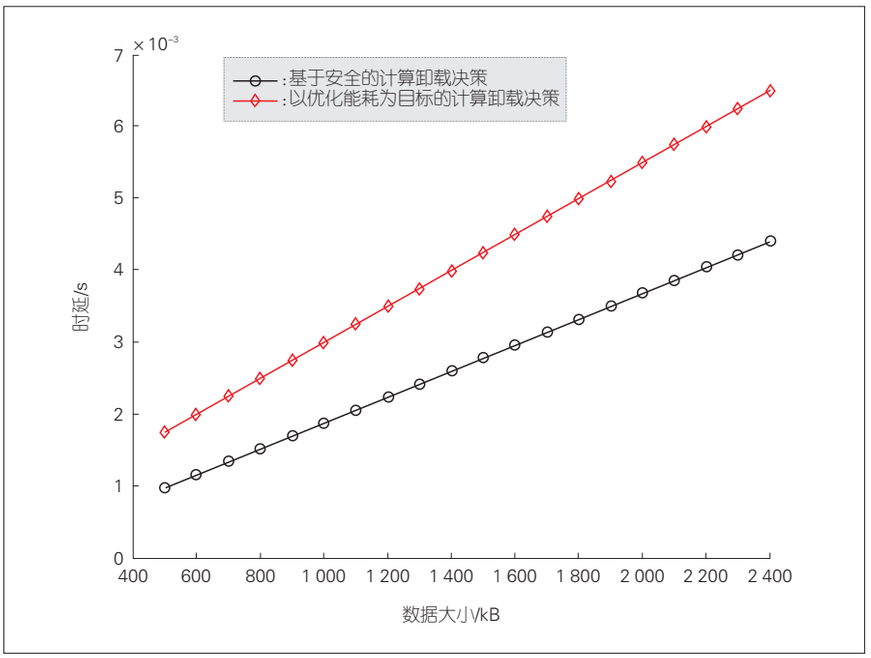
4 仿真分析

图2展示了系统总的开销和卸载任务数据量 B_n 之间的关系。随着数据量的增加，总的开销也在增加。重要参数的数值如下：UE_n 的传送功率是100 mw，数据传送速率 R_n 是10 Mbit/s，CPU周期数是1 000 Megacycles。同时还对比了本方案

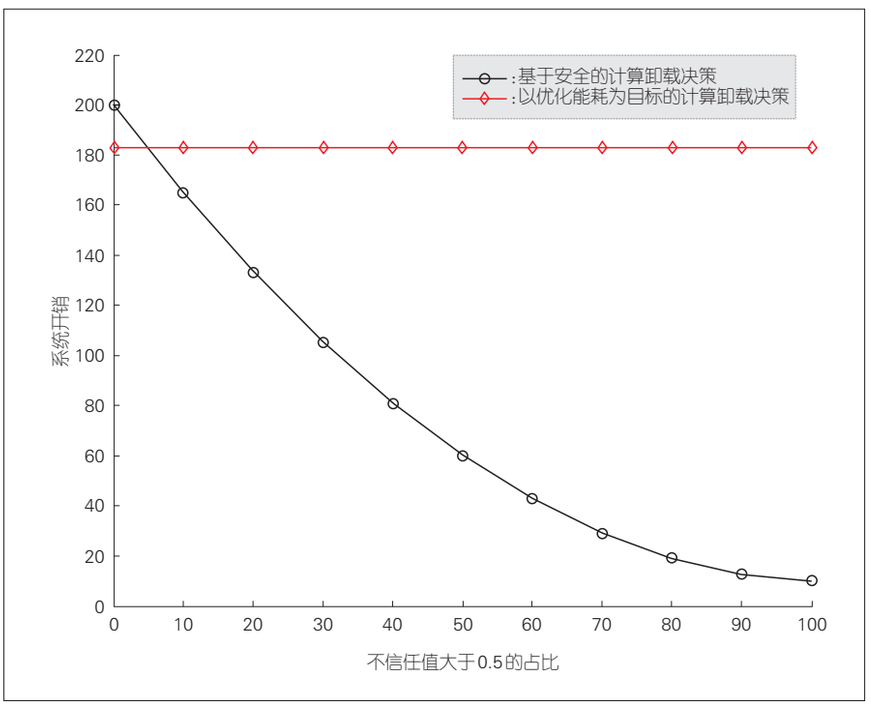


▲图2 系统总的开销和卸载任务数据量 B_n 之间的关系

的卸载决策和其他3种卸载决策的方案。由图3以看出，全部本地执行计算任务的开销最大，全部服务器执行的开销较小，我们提出的基于安全的卸载决策方案是接近最优解的。虽然整体的开销大于文献[14]中提出的仅优化能耗的方案，但仅优化能耗的方案未考虑安全因素和时延的计算。如图3所示，在时延上我们提出的计算卸载方案是优于仅考虑能耗的卸载方案。



▲图3 系统时延和卸载任务数据量B_n之间的关系



▲图4 系统开销与用户终端信任值大小的关系

在进行熵检测时，我们通过设置5个影响属性的值来模拟不同安全程度的用户。随着不信任用户的转移，整个系统的开销如图4所示。结果表明，基于安全管理的计算卸载方案通过对不信任用户的转移和集中控制能够使得系统的开销整体下降。

5 结束语

文章中，我们提出了基于安全的边缘计算卸载方案。该方案不仅使得计算卸载过程中的整体开销最小化，而且保证了卸载的安全。在该方案中，我们采用基于QEA的算法来找到最优的卸载决策矩阵。仿真结果表明，在保证安全性的前提下，本方案在整体开销的性能上优于其他常规方案。

参考文献
[1] HU Y C, PATEL M, SABELLA D, et al. Mobile Edge Computing—A Key Technology towards 5G[J]. ETSI White Paper, 201 5, 11(11): 1–16
[2] KHAN A U R, OTHMAN M, MADANI S A, et al. A Survey of Mobile Cloud Computing Application Models[J]. IEEE Communications Surveys & Tutorials, 2014, 16(1): 393–413. DOI:10.1109/SURV.2013.062613.00160
[3] LIU J, MAO Y Y, ZHANG J, et al. Delay–Optimal Computation Task Scheduling for Mobile–Edge Computing Systems[C]//2016 IEEE International Symposium on Information Theory (ISIT).Spain: IEEE, 2016: 1451–1455. DOI:10.1109/ISIT.2016.7541539
[4] REN J K, YU G D, CAI Y L, et al. Latency Optimization for Resource Allocation in Mobile–Edge Computation Offloading[J]. IEEE

➡下转第56页

脱节等情况；缺乏用于AI应用研发的数据集。

针对上述问题，建议引导制定各专业网络数据采集和存储的行业标准，统一各厂家设备能力；鼓励运营商、设备商、互联网公司、垂直行业在数据领域进行融合；建立各领域的开源数据集，用于科研和AI应用开发验证。

（3）算力建设。

以深度学习为代表的AI模型训练需要强大算力支持，目前AI芯片硬件及配套AI开发工具目前高度依赖以英伟达GPU、Google Tensor flow为代表的其他国家的产品和技术；GPU在AI训练方面具有明显优势，基本处于垄断地位，价格较高，研发成本较高。

应当鼓励国产AI芯片及AI工具的发展，建立健康产业链，并且鼓励发展基于云计算的AI平台服务，按需租用训练所需的算力资源。

参考文献

[1] 李颖.新一代人工智能发展白皮书(2017)[R].中国电子学会,2017
[2] YOSHUAB, YANNL, GEOFFREYH. Deep Learning [J]. Nature, 521: 436–444. DOI: 10.1038/nature14539
[3] ETSI. New ETSI Group on Improving Operator Experience Using AI [EB/OL]. (2017–02–21) [2019–01–10].https://goo.gl/zLZZso
[4] ETSI ISG. Experiential Networked Intelligence [EB/OL]. [2019–01–10].https://portal.etsi.org/tb.aspx?tbid=857&SubTB=857
[5] ETSI. Experiential GS ENI 001v1.1.1 [R/OL]. [2019–01–10].http://www.etsi.org/deliver/etsi_gr/ENI/001_099/001/01.01.01_60/gr_ENI001v010101p.pdf
[6] ETSI. Experiential GS ENI 002v1.1.1 [R/OL]. [2019–01–10].http://www.etsi.org/deliver/etsi_gs/ENI/001_099/002/01.01.01_60/gs_ENI002v010101p.pdf
[7] ETSI GS ENI 005. Experiential Networked Intelligence (ENI); System Architecture [EB/OL].[2019–01–10].https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=54085
[8] ETSI GR ENI 006. Experiential Networked Intelligence (ENI); PoC framework [EB/OL]. [2019–01–10].https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=54509
[9] Study of Enablers for Network Automation for 5G: 3GPP TR 23.791[S/OL].[2019–01–10]. https://www.3gpp.org/DynaReport/23–series.htm
[10] ITU–T FG–ML5G. Focus Group on Machine Learning for Future Networks Including 5G [EB/OL].[2019–01–10].https://www.itu.int/en/ITU–T/focusgroups/ml5g/Pages/default.aspx
[11] 王海宁.网络人工智能应用白皮书（2018年）[R].SDN/NFV产业联盟,2018

作者简介

王海宁，中国电信股份有限公司北京研究院网络人工智能研究中心主任，中关村高端领军人才，高级工程师，并担任ETSIISG ENI副主席、ITU–T Q6/11报告人、CCSA NFV标准特设起草组组长、SDN/NFV/AI技术标准与产业推进委员会网络人工智能应用工作组组长等多个标准组织的管理职位；近年主要研究方向包括5G网络技术、SDN/NFV、网络人工智能等；2017年获得北京市委组织部青年骨干个人项目资助；主持编制10余项国际标准和行业标准，拥有20余项授权专利，发表文章多篇。

⬅上接第46页

Transactions on Wireless Communications, 2018, 17(8): 5506–5519. DOI:10.1109/TWC.2018.2845360
[5] ZHANG K, MAO Y M, LENG S P, et al. Optimal Delay Constrained Offloading for Vehicular Edge Computing Networks[C]//2017 IEEE International Conference on Communications (ICC), 2017. France: IEEE, 2017: 1–6. DOI:10.1109/ICC.2017.7997360
[6] JIA M K, CAO J N, YANG L. Heuristic Offloading of Concurrent Tasks for Computation Intensive Applications in Mobile Cloud Computing[C]//2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Canada: IEEE, 2014: 352–357. DOI:10.1109/INFCOMW.2014.6849257
[7] KAO Y H, KRISHNAMACHARI B, RA M R, et al. Hermes: Latency Optimal Task Assignment for Resource–Constrained Mobile Computing [C]//2015 IEEE Conference on Computer Communications (INFOCOM), 2015. Hong Kong, China: IEEE, 2015: 1894–1902. DOI: 10.1109/INFOCOM.2015.7218572
[8] KAMOUN M, LABIDI W, SARKISS M. Joint Resource Allocation and Offloading Strategies in Cloud Enabled Cellular Networks[C]//2015 IEEE International Conference on Communications (ICC), 2015. UK: IEEE, 2015: 5529–5534. DOI:10.1109/ICC.2015.7249203
[9] LABIDI W, SARKISS M, KAMOUN M. Energy–Optimal Resource Scheduling and Computation Offloading in Small Cell Networks [C]//2015 22nd International Conference on Telecommunications (ICT). Australia: IEEE, 2015: 313–318. DOI:10.1109/ICT.2015.7124703
[10] ZHANG H L, GUO J, YANG L C, et al. Computation Offloading Considering Fronthaul and Backhaul in Small–Cell Networks Integrated with MEC[C]//2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2017. USA: IEEE, 2017: 115–120. DOI:10.1109/INFCOMW.2017.8116362
[11] QIN X, XU T G, WANG C. DDoS Attack Detection Using Flow Entropy and Clustering Technique[C]//2015 11th International Conference on Computational Intelligence and Security (CIS). China: IEEE, 2015: 412–415. DOI:10.1109/CIS.2015.105
[12] SHARMA V, YOU I, KUMAR R, et al. Computational Offloading for Efficient Trust Management in Pervasive Online Social Networks Using Osmotic Computing[J]. IEEE Access, 2017, 5: 5084–5103. DOI:10.1109/access.2017.2683159
[13] HAN K H, KIM J H. Quantum–Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(6): 580–593. DOI:10.1109/tevc.2002.804320
[14] LABIDI W, SARKISS M, KAMOUN M. Energy–Optimal Resource Scheduling and Computation Offloading in Small Cell Networks[C]//2015 22nd International Conference on Telecommunications (ICT). Australia: IEEE, 2015: 313–318. DOI:10.1109/ICT.2015.7124703

作者简介

廉晓飞，北京邮电大学未来网络理论与应用实验室在读硕士生；主要研究方向为5G网络、移动边缘计算等。

谢人超，北京邮电大学未来网络理论与应用实验室副教授、硕士生导师；主要研究方向为信息中心网络、移动网络内容分发技术和移动边缘计算等；主持国家与省部级项目4项；已发表论文40余篇，申请授权专利10余项。

黄韬，北京邮电大学未来网络理论与应用实验室教授、博士生导师；主要研究方向为新型网络体系架构、内容分发网络、软件定义网络等；主持国家与省部级项目10余项；已发表论文100余篇，申请授权专利40余项。