基于联盟链的可靠边缘计算任务卸载方法

许悦玥[1],刘博文[1],田 臣[1,2],戴海鹏[1],郑嘉琦[1],陈贵海[1],窦万春[1,2,3*]

(1. 南京大学软件新技术国家重点实验室,江苏南京 210023;2. 人工智能与数字经济广东省实验室,广东深圳 518060;
3. 西南林业大学大数据与智能工程学院,云南昆明 650000)

摘 要: 随着移动终端尤其是工业互联网技术的快速发展,终端设备密集分布,无线带宽有限,经常导致业务过程中的集中式云资源调度,难以满足远程终端应用对低时延和低成本计算的需求. 着眼于本地服务器联动云数据中心,边缘计算为这类移动应用提供了一种敏捷的计算服务模式. 虽然边缘计算的敏捷服务模式能够有效缩短移动应用的时延并降低对应的通信成本,但在边缘计算环境下,异构资源之间的任务卸载经常会导致一些潜在的数据安全隐患和服务质量受损. 针对上述应用挑战和技术发展趋势,本文提出了一种基于联盟链的可靠边缘计算任务卸载方法. 该方法利用联盟链进行身份校验和卸载结果反馈,以任务的完成时间、卸载成本与资源可靠度作为评价标准,设计了一种基于遗传算法的卸载策略,支持卸载决策时任务卸载的可靠性评估. 实验结果表明,本文方法能在满足时延约束的前提下提高任务卸载的可靠性,为移动智能应用提供了一种有效的数据安全保障方法.

关键词: 边缘计算;任务卸载;联盟链;遗传算法;资源优化

基金项目: 国家重点研发计划项目(No.2020YFB1707600);人工智能与数字经济广东省实验室(深圳)开放课题(No.GML-KF-22-20);西南林业大学云南省专家工作站课题(No.202105AF150013)

中图分类号: TP393.1 **文献标识码:** A **文章编号:** 0372-2112(2024)01-0232-12

电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20211195

A Consortium Blockchain Based Reliable Task Offloading Approach in Edge Computing

XU Yue-yue[1], LIU Bo-wen[1], TIAN Chen[1,2], DAI Hai-peng[1], ZHENG Jia-qi[1],
CHEN Gui-hai[1], DOU Wan-chun[1,2,3*]

(1. *Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu* 210023, *China*;
2. *Guangdong Provincial Laboratory of Artificial Intelligence and Digital Economy, Shenzhen, Guangdong* 518060, *China*;
3. *College of Big Data and Intelligent Engineering, Southwest Forestry University, Kunming, Yunnan* 650000, *China*)

Abstract: With the rapid development of mobile terminals, especially the industrial Internet technique, the dense distribution of terminal devices and the limitation of wireless mobile bandwidth make it difficult for the centralized cloud resource scheduling of specific business processes to meet the low-latency and low-cost computing needs of remote terminal applications. Focusing on local servers linked to cloud data centers, edge computing provides an agile computing service model for these mobile applications. Although the service pattern of edge computing can effectively reduce the latency of mobile applications and the communication costs, task offloading between heterogeneous resources in the edge computing environment often leads to potential data security hazards and compromised quality of service. In response to the above challenges and technology development trends, we propose a consortium blockchain based reliable task offloading approach in edge computing. In this approach, we design a genetic algorithm-based offloading strategy using the consortium blockchain for identity verification and feedback of offloading results, and using task completion time, offloading cost and resource reliability as evaluation index. The results of simulation experiments show that our approach can improve task offload reliabili-

收稿日期:2021-09-01;修回日期:2022-03-02;责任编辑:王天慧
*通讯作者:窦万春

ty while satisfying the latency constraint, providing an effective data security approach for mobile smart applications.

Key words: edge computing; task offloading; consortium blockchain; genetic algorithm; resource optimization

Foundation Item(s): National Key Research and Development Program of China (No.2020YFB1707600); Open Research Fund from Guangdong Laboratory of Artificial Intelligence Digital Economy (SZ) (No.GML-KF-22-20); Dou Wanchun Expert Workstation of Yunnan Province (No.202105AF150013)

1 引言

随着移动终端产业的快速发展,根据行业数据分析的指数预测,到2022年,全球有超过120亿的终端设备入网[1],传统的集中式云计算已经难以满足时延敏感型移动应用的需求. 边缘计算(Edge Computing, EC)作为云计算的扩展[2],能够充分利用边缘资源支持时延敏感型应用程序,避免了前传与回传链路的网络延迟,提供了一种低延迟的计算服务. 在边缘计算环境中,基础设施提供商部署在基站以及入网节点的边缘服务器将以边缘节点的形式加入边缘网络,为用户提供计算资源. 终端用户可以将计算任务卸载到边缘服务器中进行计算,然而,复杂异构资源环境下,任务卸载面临可靠性保障方面的应用挑战[3,4]. 譬如,终端设备上传的数据可能被未经严格验证的边缘节点修改或滥用,引发隐私数据泄漏或其他安全问题[5,6];而计算资源提供方为获取更多的服务利益,也可能恶意竞争与其实际计算能力不符的计算任务[7],从而导致任务计算过程中的资源实际卸载时间与成本远远高于预期[8,9].

针对上述应用挑战,研究者们提出将区块链技术引入边缘计算[10~12]. 边缘计算环境下,地理分布的异构资源难以集中式管理,未经严格验证的边缘节点可能修改或滥用任务中的隐私数据[13]. 因此,需要一个分布式的可信架构,用以保护隐私数据. 区块链的分布式架构与智能合约技术,正适用于建立可溯源的多方互信平台[14]. 具体分析,区块链根据准入机制可以分为公有链、私有链与联盟链. 公有链的数据公开透明,所有人都可以访问,但是达成共识与出块速度慢[15];私有链虽然交易速度快,但是数据访问权限完全被一个组织掌控;联盟链由多个机构共同参与管理,可以实现细粒度的身份校验与数据访问,同时拥有较快的交易速度. 第二代区块链拓展了智能合约的技术应用[16,17],公共记录账本成为普遍可编程的基础设施,紧密结合边缘计算的应用特点,实现对边缘网络的可靠访问和控制. 譬如,文献[18]将区块链与边缘计算结合,通过区块链实现边缘资源整体的自治与监督. 文献[19]则进一步提出了利用私有链存储隐私数据,只有经过辅助信任和信誉系统授权的节点才能访问隐私数据. 上述方案考虑到了区块链中的安全问题,但大都没有考虑边缘计算任务密集型和时延敏感型应用的卸载需求. 对此,文献[20]设计了一个支持区块链的边缘计算任务卸载框架,以期满足时间和能耗要求的边缘计算任务卸载需求. 但该框架支持计算任务卸载完成后的读取记录与追溯,没有提前进行节点访问的可信验证,也没有考虑计算任务卸载时可能出现的欺骗问题[21],即节点为获取超出其算力的收益而夸大其计算能力,导致实际应用时的卸载时间与成本高于预期.

为解决以上挑战,本文提出了一种基于联盟链的可靠边缘计算任务卸载方法. 该方法首先设计了一个基于联盟链的边缘计算任务卸载架构,通过联盟链实现计算节点的身份校验;在此基础上,提出了一个边缘节点的可靠度评估方法,对任务卸载过程进行可靠性评估. 本文将任务的完成时间、卸载成本与资源可靠度作为卸载决策指标,生成计算任务卸载策略. 不同规模场景和任务配置下进的实验表明,本文所提方法能在任务时限约束下,提高任务卸载的可靠性,优化卸载成本. 具体而言,本文主要的创新贡献如下:

(1)基于联盟链进行计算任务卸载的思想,设计了一个基于联盟链的边缘计算任务卸载架构;

(2)提出一套可靠性度量评估指标,对基于联盟链的可靠边缘计算任务卸载进行了建模分析.

2 边缘计算环境下基于联盟链的任务卸载架构

在边缘环境下,提供计算服务的节点称为边缘计算节点(Edge Computing Node, ECN). x 个 ECN 的集合记为 $\mathrm{ECN}=\{\mathrm{ecn}_1, \mathrm{ecn}_2, \cdots, \mathrm{ecn}_x\}$.

通过虚拟化技术,ECN将计算资源虚拟成虚拟机(Virtual Machine, VM)实例,每个 ECN 可以提供若干 VM 实例,计算卸载到此 ECN 上的任务. 若 $\mathrm{ecn}_i\left(i=\{1, 2, \cdots, x\}\right)$ 上有 y_i 个 VM 实例,这些 VM 的集合可记为 $\mathrm{VM}_i=\{\mathrm{vm}_1, \mathrm{vm}_2, \cdots, \mathrm{vm}_{y_i}\}$,其中每个 VM 可以表示为一个五元组,如式(1)所示.

$$\mathrm{vm}_k=a_k, q_k, e_k^{\mathrm{compute}}, e_k^{\mathrm{idle}}, \beta_k, \quad k=\{1, 2, \cdots, y_i\} \tag{1}$$

其中,a_k 是一个二元变量,表示 vm_k 在该时刻是否空闲,

如式(2)所示；q_k为vm_k的计算能力；e_k^{compute}为vm_k执行计算时单位时间的能耗；e_k^{idle}为vm_k空闲时单位时间的能耗；β_k为vm_k的计算资源单价.

$$a_k=\begin{cases}1, & \mathrm{vm}_k\text{空闲}\\ 0, & \text{其他}\end{cases},k=\{1,2,\cdots,y_i\} \tag{2}$$

若终端设备数量为m，等待卸载任务的终端设备集合记为$\mathrm{TE}=\{\mathrm{te}_1,\mathrm{te}_2,\cdots,\mathrm{te}_m\}$. 设$\mathrm{te}_i(i=\{1,2,\cdots,m\})$，有$n_i$个任务需要卸载. 每台终端设备的待卸载任务集可用式(3)表示.

$$\mathrm{taskset}_i=\{\mathrm{task}_1,\mathrm{task}_2,\cdots,\mathrm{task}_{n_i}\},\quad i=\{1,2,\cdots,m\} \tag{3}$$

每个待卸载任务都可表示为一个三元组，$\mathrm{te}_i(i=\{1,2,\cdots,m\})$的待卸载任务如式(4)所示. 其中，$c_j$为该任务需要计算的数据量，$d_j$为该任务需要传输的数据量，$t_j$为任务的完成时间期限.

$$\mathrm{task}_j=\{c_j,d_j,t_j\},\ j=\{1,2,\cdots,n_i\} \tag{4}$$

任务的数据中包括用户的隐私信息，为保障数据安全，我们将边缘计算技术与联盟链技术结合. 联盟链是一种只针对特定组织成员和有限第三方的区块链，专门为企业级应用程序设计[22]，在传统的公有链网络之上追加了组织和准入规则的核心概念，可以避免工作量证明(Proof of Work，PoW)共识机制引发的大量能源消耗[23]. 同时，联盟链由若干个组织共同参与管理，其中每个组织都运行着一个或多个节点，链上的数据只允许相关组织所属节点进行读写，提供了有效的数据安全保障[24].

结合上述分析，一个基于联盟链技术的任务卸载架构设计如图1所示. 该架构主要由三部分组成：终端设备、联盟链和ECN. ECN上的资源由边缘服务器或个人计算设备以VM实例的形式提供. 联盟链中提供计算服务的ECN都必须先经过身份校验，身份校验成功的ECN提供可信计算服务，不会恶意篡改或泄露用户隐私数据.

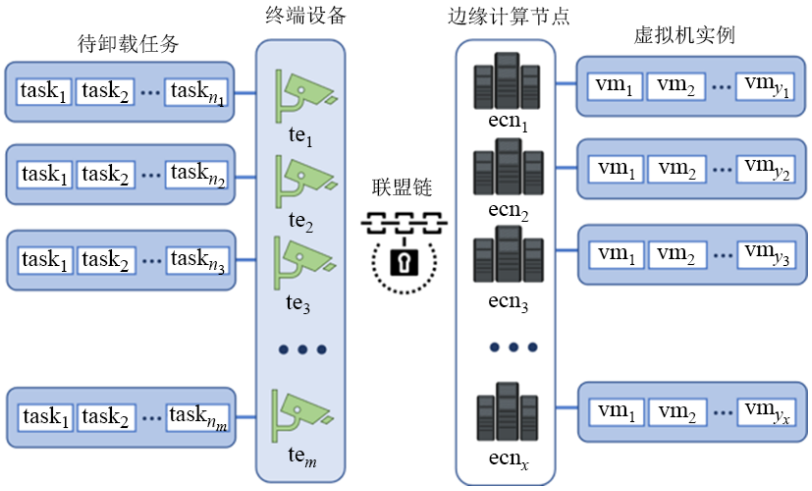


图1 基于联盟链的边缘计算任务卸载架构

3 基于联盟链的可靠边缘计算任务卸载建模

3.1 任务卸载架构中的可靠性评估分析

由第2节中的定义，设联盟链中VM总数量为z，z可根据ECN数量x和每个ECN上的VM数量$y_i(i=\{1,2,\cdots,x\})$，由式(5)计算得到.

$$z=\sum_{i=1}^{x}y_i \tag{5}$$

由于不同终端设备与VM实例间的带宽与传输能耗不同，本文设终端设备te_i与VM实例vm_k间的数据传输速率为$r_{i,k}(i=\{1,2,\cdots,m\},k=\{1,2,\cdots,z\})$，单位时间的数据传输能耗为$e_{i,k}(i=\{1,2,\cdots,m\},k=\{1,2,\cdots,z\})$.

定义1 任务完成时间：每个任务的完成时间，为终端设备与VM之间的数据传输时间与VM上的计算执行时间之和.

$\mathrm{te}_i(i=\{1,2,\cdots,m\})$上的任务$\mathrm{task}_j(j=\{1,2,\cdots,n_i\})$卸载到$\mathrm{vm}_k$上的完成时间可由式(6)计算得出.

$$T(\mathrm{task}_j)=\frac{c_j}{q_k}+\frac{d_j}{r_{i,k}} \tag{6}$$

由于VM在卸载策略决定后就被占用，没有执行计算的闲置状态下依然有基准能耗，所以，在数据传输时，VM的闲置能耗也算入任务的卸载能耗[25]. 卸载能耗包括VM执行该计算任务的能耗、传输数据时VM的闲置能耗和传输数据的能耗. 任务卸载成本与能耗成正比.

定义 2　任务卸载成本:$\mathrm{vm}_k(k=\{1, 2, \cdots, z\})$的计算资源价格为$\beta_k$. $\mathrm{te}_i(i=\{1, 2, \cdots, m\})$上的任务$\mathrm{task}_j(j=\{1, 2, \cdots, n_i\})$卸载到$\mathrm{vm}_k$上的卸载成本如式(7)所示.

$$E\left(\mathrm{task}_j\right)=\beta_k\cdot\left(\frac{c_j}{q_k}\cdot e_k^{\mathrm{compute}}+\frac{d_j}{r_{i,k}}\cdot\left(e_k^{\mathrm{idle}}+e_{i,k}\right)\right) \tag{7}$$

在边缘环境下,未经过联盟链身份校验的ECN可能恶意篡改或泄露用户的隐私数据. 同时,一旦出现终端设备与ECN之间的通信链路不稳定等情况,任务卸载的实际计算时间与卸载成本就可能远远高于预期. 为提供可靠的计算服务,我们进行以下定义.

定义 3　卸载成功率:设$\mathrm{vm}_k(k=\{1, 2, \cdots, z\})$已卸载了$L$个任务,其中第$j$个任务的完成时间期限为$t_j$,实际完成时间为$T_j$. 若$T_j\in(0, t_j]$,则称第$j$个任务卸载成功;否则卸载失败. 若$L$个任务中有$l$个任务卸载成功,该资源的卸载成功率由式(8)计算.

$$\mathrm{suc}_k=(l/L)\times 100\% \tag{8}$$

设vm_k实际出现卸载失败的概率为p_k,VM之间相互独立,每个VM出现卸载失败的概率也独立. 每次任务卸载完成后,根据卸载结果修改suc_k. p_k是非先验的,但当历史卸载任务数趋近于无穷时,suc_k趋近于$1-p_k$.

定义 4　资源可靠度:若$\mathrm{vm}_k(k=\{1, 2, \cdots, z\})$所属的ECN在联盟链上身份校验成功,则校验值$\mathrm{Ver}_k=1$;否则$\mathrm{Ver}_k=0$. vm_k的资源可靠度$\mathrm{Rel}_k=\mathrm{Ver}_k\cdot\mathrm{suc}_k$.

3.2　基于联盟链的可靠边缘计算任务卸载模型构建

本文针对边缘计算环境下基于联盟链的任务卸载应用,在保护数据安全,满足任务时间期限的前提下,提高任务卸载可靠性,降低卸载成本.

设终端设备te_i上的任务集合$\mathrm{taskset}_i$与联盟链中的全部VM集合$\{\mathrm{vm}_1, \mathrm{vm}_2, \cdots, \mathrm{vm}_z\}$之间的卸载策略由矩阵$\boldsymbol{O}_i=\left[o_{jk}\right]_{n_i\times z}$表示,矩阵中的元素$o_{jk}$表示$\mathrm{te}_i$上的$\mathrm{task}_j$是否卸载到$\mathrm{vm}_k$上,即

$$\begin{cases}\mathrm{sigh}\left(o_{jk}\right)=\begin{cases}1,\ \mathrm{task}_j\text{卸载至}\mathrm{vm}_k\text{上}\\0,\ \text{其他}\end{cases}\\ j=\{1, 2, \cdots, n_i\}, k=\{1, 2, \cdots, z\}, a_j\neq 0\end{cases} \tag{9}$$

由于终端设备上的多个任务可以并行卸载到不同的VM实例上进行计算,根据$\boldsymbol{O}_i$进行任务卸载,te_i完成所有任务的预计完成时间为

$$T\left(\mathrm{taskset}_i\right)=\max_{\substack{j=1,2,\cdots,n_i\\k=1,2,\cdots,z}}\left(\frac{c_j}{q_k}+\frac{d_j}{r_{i,k}}\right)\cdot\mathrm{sigh}(o_{jk}) \tag{10}$$

根据$\boldsymbol{O}_i$进行任务卸载,te_i完成所有任务的预期卸载成本为

$$E\left(\mathrm{taskset}_i\right)=\sum_{k=1}^{z}\sum_{j=1}^{n_i}(E(\mathrm{task}_j)\cdot\mathrm{sigh}(o_{jk})) \tag{11}$$

卸载策略$\boldsymbol{O}_i$对应的卸载资源可靠度由式(12)计算.

$$H\left(\boldsymbol{O}_i\right)=\sum_{j=1}^{z}(\mathrm{Rel}_j\cdot\mathrm{sigh}(o_{ij})), \forall\mathrm{Ver}_j=1 \tag{12}$$

根据上述分析,考虑到任务卸载成本和资源可靠度,te_i根据$\boldsymbol{O}_i$进行任务卸载的代价函数表示为式(13). 其中,λ_e为任务卸载成本的权重,λ_h为资源可靠度的权重. E_i^*为卸载成本归一化后的值,H_i^*为资源可靠度归一化后的值. 如式(14)与(15)所示,其中$\mathrm{Rel}_{\max}$与$\mathrm{Rel}_{\min}$分别表示当前VM集合$\{\mathrm{vm}_1, \mathrm{vm}_2, \cdots, \mathrm{vm}_z\}$中的资源可靠度最高值与最低值.

$$C\left(\boldsymbol{O}_i\right)=\lambda_e\cdot E_i^*+\lambda_h\cdot H_i^{*-1} \tag{13}$$

$$E_i^*=\arctan(E(\mathrm{taskset}_i))\cdot\frac{2}{\pi} \tag{14}$$

$$H_i^*=\frac{H\left(\boldsymbol{O}_i\right)^{-1}-\mathrm{Rel}_{\min}\cdot n_i}{(\mathrm{Rel}_{\max}-\mathrm{Rel}_{\min})\cdot n_i} \tag{15}$$

对te中m台终端设备的卸载策略集合记为$\mathrm{OS}=\{\boldsymbol{O}_1, \boldsymbol{O}_2, \cdots, \boldsymbol{O}_m\}$.

定义 5　优化目标函数:边缘计算环境下基于联盟链的任务卸载优化目标函数可表示为

$$\underset{\mathrm{OS}}{\text{minimize}}\ C=\sum_{i=1}^{m}(\lambda_e\cdot E_i^*+\lambda_h\cdot H_i^{*-1}) \tag{16}$$

$$\text{s.t.}\begin{cases}\mathrm{C1}: \mathrm{Ver}_k=1, \forall k\in\{1, 2, \cdots, z\}\\ \mathrm{C2}: a_k-\mathrm{sigh}\left(o_{jk}\right)\in\{0, 1\}, \forall j\in\{1, 2, \cdots, n_i\}, \forall k\in\{1, 2, \cdots, z\}\\ \mathrm{C3}: \mathrm{Rel}_k>0, \forall k\in\{1, 2, \cdots, z\}\\ \mathrm{C4}: \sum_{k=1}^{z}\mathrm{sigh}\left(o_{jk}\right)=1, \forall j\in\{1, 2, \cdots, n_i\}\\ \mathrm{C5}: \sum_{j=1}^{n_i}\mathrm{sigh}\left(o_{jk}\right)\in\{0, 1\}, \forall k\in\{1, 2, \cdots, z\}\\ \mathrm{C6}: t_j\geqslant T\left(\mathrm{task}_j\right), \forall j\in\{1, 2, \cdots, n_i\}\end{cases} \tag{17}$$

以上约束中,C1约束卸载策略所选择的VM所属ECN身份校验成功,C2约束卸载策略所选择的VM必须处于可用状态;C3约束资源可靠度为正值;C4确保每个任务必须选择且仅选择一个VM执行任务卸载;C5约束每个任务选择不同的VM并行卸载任务;C6约束所有任务卸载时间不得超过其完成时间期限.

4 基于联盟链的可靠边缘计算任务卸载方法

4.1 基于联盟链的可靠边缘计算任务卸载方法时序逻辑分析

早期许多区块链相关工作基于以太坊实现[26]. 以太坊是使用PoW共识机制的公共区块链平台,需要消耗大量算力,交易速度较慢,不太适用于有低时延需求的边缘计算环境. 为实现隐私保护与高效交易的目标,联盟链被引入到边缘计算这一研究领域[27,28]. 文献[27]中使用了Hyperledger Sawtooth,文献[28]的架构则基于Hyperledger Fabric实现. 前者主要目标是实现边缘服务器对卸载任务的竞争,没有专门考虑隐私信息安全,故对等体可以访问所有事务数据,采用对权限要求不严格的Sawtooth架构;后者则在边缘网络上利用Fabric实现身份管理和访问控制策略. Fabric有多个隐私级别,可以通过定义每条链的"通道",在多组参与者之间实现完全数据隔离,身份验证速度快,更适合边缘环境下实现可靠的计算任务卸载[29]. Fabric的核心优点有:(1)灵活的智能合约机制,即智能合约的运行与交易背书由不同角色完成;(2)良好的可扩展性,即通用的功能模块和接口设计便于实现应用系统的开发和部署;(3)完整的隐私保护,即带有身份校验的多链设计可以实现数据隔离.

因此,本文采用Hyperledger Fabric的技术方法,设计支持边缘计算环境下可靠任务卸载的应用实施. 具体而言,基于联盟链的可靠边缘计算任务卸载方法的时序逻辑如图2所示,主要由3步骤组成:(1)联盟链身份校验;(2)任务卸载决策;(3)卸载结果反馈.

4.2 联盟链身份校验

ECN需要通过证书签名的身份校验,才能接入联盟链进行任务卸载交易. 这一步骤保证了提供计算服务的节点都已通过预先的可信验证. Fabric框架中的每条链以通道的形式表现,拥有一个共享账本和多个节点. 每个通道只由相关组织的节点所有,其上的数据仅限于这些节点访问. 具体的技术实施上,类似参考文献[30],我们采用基于证书的公钥基础设施(Public Key Infrastructure,PKI)对每个上链节点进行身份校验. 节点在Fabric框架中以组织划分,组织是承担着数据信用责任的可信边缘计算资源提供方. 每个通道都由对应的成员服务提供者(Membership Service Provider,MSP)管理准入可信组织. 在Fabric分层PKI结构中,MSP管理着根证书颁发机构(Root Certification Authority,RCA)和中间证书颁发机构(Intermediate Certification Authority,ICA). 图3展示了MSP的重要组成部分,MSP为通道准入组织和组织内的各个实体创建了一组证书和私钥以验证节点资格.

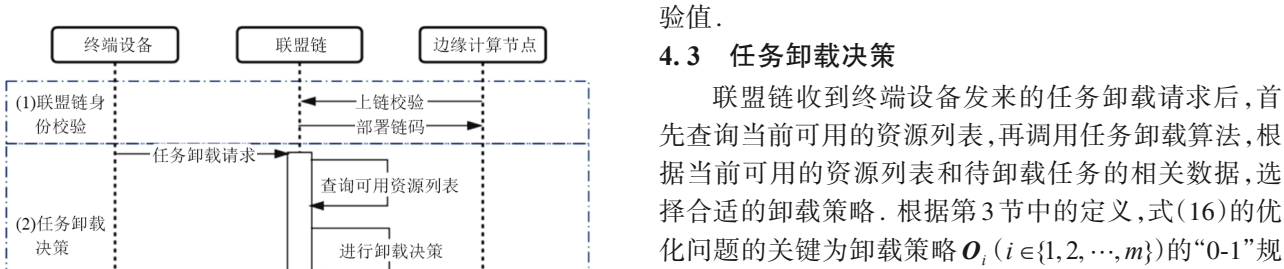


图2 基于联盟链的可靠边缘计算任务卸载方法时序图

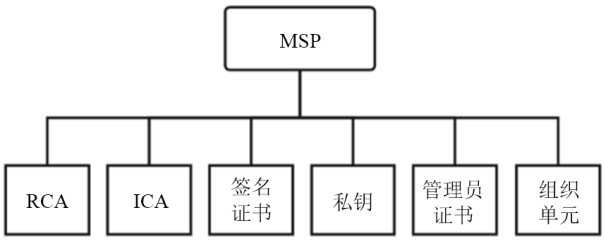


图3 MSP重要组成部分

本文采用的数字签名身份校验方案将摘要算法和非对称加密结合使用. MSP使用的非对称加密算法是椭圆曲线数字签名算法(Elliptic Curve Digital Signature Algorithm,ECDSA),哈希算法是SHA-256. 利用数字签名进行身份校验的过程如算法1所示. 校验结束后,联盟链将为经过校验的ECN上每个VM同步设置身份校验值.

4.3 任务卸载决策

联盟链收到终端设备发来的任务卸载请求后,首先查询当前可用的资源列表,再调用任务卸载算法,根据当前可用的资源列表和待卸载任务的相关数据,选择合适的卸载策略. 根据第3节中的定义,式(16)的优化问题的关键为卸载策略$\boldsymbol{O}_i$($i\in\{1,2,\cdots,m\}$)的"0-1"规划问题,故本文采用元启发式算法中的遗传算法(Genetic Algorithms,GA)对该优化问题进行求解[31]. GA可以并行迭代搜索全局优化解,决策算法的详细步骤如下,对应算法2.

(1)编码与初始化种群:本文采用二进制编码方式进行基因编码,将卸载策略$\boldsymbol{O}_i$($i=1,2,\cdots,m$)表示为一个$W(W=n_i\cdot\lceil\log_2 z\rceil)$位的二进制串. 设定种群规模为$S$,初始化随机生成$S$条染色体个体. 种群记为pop=

算法1　数字签名身份校验

输入: 通道id,ECN的数字签名
输出: 身份校验值Ver
1. 验证标志位 $\text{flag}_v \leftarrow 0$
2. 由通道id获取对应的MSP id
3. 由MSP id获取MSP中序列化的证书和公钥
4. DO
5.　IF 证书无效
6.　　$\text{flag}_v \leftarrow 0$
7.　　BREAK
8.　获取签名的对应哈希算法
9.　用哈希算法生成摘要值
10.　用公钥解密ECN签名
11.　IF 解密后的ECN签名与摘要值不符
12.　　$\text{flag}_v \leftarrow 0$
13.　　BREAK
14.WHILE FALSE
15.IF $\text{flag}_v = 1$
16.　Ver ← 1
17.ELSE
18.　Ver ← 0
19.RETURN Ver

$\{\text{ch}_1, \text{ch}_2, \cdots, \text{ch}_S\}$. 其中每个染色体个体都是一个二进制串,表示为 $\text{ch}_s = \{b_1, b_2, \cdots, b_W\}$ $(s = \{1, 2, \cdots, S\})$,可以映射为一个卸载策略 $\boldsymbol{O} = \left[o_{jk}\right]_{n_i \times z}$. 本步骤对应算法2中的第1步.

(2)适应度与选择操作. 先将染色体个体映射为卸载策略,再计算个体的适应度,根据适应度选择遗传到下一代种群中的个体. 适应度较高的个体遗传到下一代种群的概率较大,本文设置适应度函数为式(19). 其中,ω_1 为卸载成本的权重,ω_2 为资源可靠度的权重,δ 表示任务是否满足时间期限,ρ 为超过任务时间期限的惩罚值.

$$\delta = \begin{cases} 1, & T(\text{task}_i) \leqslant t_j \\ 0, & T(\text{task}_i) > t_j \end{cases}, j = \{1, 2, \cdots, n_i\} \tag{18}$$

$$\begin{cases} F\left(\text{ch}_s\right) = \left(\omega_1 \cdot E_i^* + \omega_2 \cdot H_i^{*-1} - \delta \cdot \rho\right)^{-1} \\ s = \{1, 2, \cdots, S\} \end{cases} \tag{19}$$

选择策略采用锦标赛选择算子[32],先从种群中随机抽取 θ 个个体,再选取其中适应度最高的个体. 若第 n 代的种群适应度呈正态分布 $\text{N}(\mu_F, \sigma_F^2)$,适应度为 f 的个体被选取的概率如式(20)所示,该式代表了适应度 f 的个体与具有较低适应度分数的 $\theta - 1$ 个个体一起出现的概率. 该算子保持了种群多样性,不易陷入局部最优解. 本步骤对应算法2中的第4~7步.

$$p(f = \max(f_1, f_2, \cdots, f_s)) = s \cdot P(F < f)^{s-1} p(f) \tag{20}$$

(3)交叉与变异操作. 交叉操作使用 k 点交叉算子,根据交叉概率 P_{cor} 对亲代染色体进行部分染色体交换得到新个体. 变异操作采用位反转突变算子产生新个体,突变概率为 P_{mut}. 交叉与变异操作对二进制串表示的个体处理速度快. 本步骤对应算法2中的第8~17步.

(4)迭代寻找优化解. 新一代种群迭代执行步骤(2)和步骤(3). 若迭代次数已经达到预设的最大迭代次数 $\text{iter}_{\max}$,则结束迭代,输出适应度最高的染色体对应的卸载策略,以及该策略的预期卸载成本;否则将持续进行迭代操作. 本步骤对应算法2中的第18~23步.

算法2　任务卸载决策

输入: 染色体个体位数 W,种群规模 S,算子参数 θ,算子参数 k,交叉概率 P_{cor},突变概率 P_{mut},最大迭代次数 $\text{iter}_{\max}$,待卸载任务集taskset,集合VM
输出: 适应度最高的染色体对应卸载策略,策略的预期卸载成本 e
1.　初始化染色体种群pop
2.　设置迭代次数 iter ← 0
3.　DO
4.　　FOR $i \leftarrow 1$ to S
5.　　将 ch_i 映射为卸载策略
6.　　根据卸载策略计算 ch_i 适应度
7.　　执行锦标赛选择算法,舍弃部分个体
8.　　DO
9.　　随机选取两个亲代个体
10.　　IF random (0, 1) < P_{cor}
11.　　执行 k 点交叉操作,产生新个体
12.　　随机选取一个亲代个体
13.　　FOR $i \leftarrow 1$ to W
14.　　IF random (0, 1) < P_{cor}
15.　　执行亲代个体第 i 位反转突变操作
16.　　产生新个体
17.　　UNTIL 种群中有 S 个个体
18.　　记录种群中适应度最高的个体 $\text{ch}_{\max}$
19.　　iter ← iter + 1
20. UNTIL iter ⩾ $\text{iter}_{\max}$
21. 将 $\text{ch}_{\max}$ 映射为卸载策略 $\boldsymbol{O}_{\max}$
22. 根据 $\boldsymbol{O}_{\max}$ 计算预期卸载成本 e
23. RETURN $\boldsymbol{O}_{\max}$ 和 e

4.4　卸载结果反馈

待卸载任务被卸载到第4.3节得到的卸载策略指定的计算资源,任务卸载完成后,其计算结果将被发回给终端设备,联盟链对比本次卸载的实际完成时间与任务完成时间期限,判断本次卸载是否成功,修改相应的资源可靠度. 联盟链上进行卸载结果反馈的具体步骤如算法3所示.

ECN提供的计算能力由联盟链进行身份校验时记

算法3 任务卸载决策

输入: 任务时间期限t,实际卸载时间T,资源历史卸载成功率Suc,资源历史卸载任务数量K,资源身份身份校验值Ver

输出: VM的资源可靠度Rel

1. 设置卸载结果标志位flag_r
2. IF $t<T$
3. $\quad \text{flag}_r \leftarrow 0$
4. ELSE
5. $\quad \text{flag}_r \leftarrow 1$
6. $\text{suc} \leftarrow \left(K \times \text{suc} + \text{flag}_r\right) / \left(K+1\right)$
7. $\text{Rel} \leftarrow \text{Ver} \times \text{suc}$
8. RETURN Rel

录,难以篡改,若节点出现欺骗行为,提供的计算服务与记录不符,将导致卸载时间与预期不符,资源可靠度降低. 本文方法将资源可靠度纳入卸载决策指标,倾向于选择可靠度高的节点,且随着卸载任务次数增加,有欺骗行为的节点被选择可能性将越来越低.

5 实验与分析

5.1 实验设置

本文方法(a consortium Blockchain based Reliable task Offloading Approach in edge computing,BROA)基于Hyperledger Fabric 2.3.2和Python 3.8实现. 本文基于真实的边缘计算平台数据进行实验验证[1],通过应用不同的计算资源与任务配置,对比评估本文方法的性能. 边缘云平台虚拟机相关参数见表1,其中计算、闲置与传输价格根据其相应的能耗与资源价格得到,资源价格参考阿里云边缘计算服务(Edge Node Service, ENS)的边缘算力与带宽计费[2].

表1 边缘云平台虚拟机参数

VM类型	I型VM	II型VM	III型VM	IV型VM
核心数量	1	2	4	8
内存/(GB)	2	4	16	32
存储空间/(GB)	20	20	40	40
计算价格/(元/s)	5.17×10^{-3}	1.00×10^{-2}	2.75×10^{-2}	5.75×10^{-2}
闲置价格/(元/s)	2.17×10^{-3}	4.00×10^{-3}	1.55×10^{-2}	3.33×10^{-2}
传输价格/(元/s)	4.33×10^{-4}	4.55×10^{-3}	1.36×10^{-1}	4.26×10^{-1}

本文参考文献[27]的研究思路,设置了多种不同规模的边缘计算场景,以验证本文方法在不同规模场景下的有效性. 场景1与场景2的VM数量相同,卸载成功概率范围不同;场景2与场景3的卸载成功率概率范围相同,VM数量不同. 场景的具体参数如表2所示.

以任务的传输数据量与计算数据量为变量,划分了两类任务,任务的数据量服从方差为0.5的正态分布,具体数值如表3所示.

表2 场景参数

场景类型	场景1	场景2	场景3
ECN数量	14	14	4
I型VM数量	12	12	2
II型VM数量	6	6	4
III型VM数量	37	37	6
IV型VM数量	9	9	2
卸载成功概率	30%~100%	60%~100%	60%~100%

表3 计算任务参数

任务类型	A类任务	B类任务
传输数据量均值/MB	6.82×10^{-2}	7.19
计算数据量均值/MB	1.52	2.67×10^{2}

BROA的参数设置为$\omega_1=0.6$,$\omega_2=0.4$,$\rho=1\times10^3$. 本文选用相关工作Coopedge[27]与两种经典的运筹学优化算法作为对比算法,用以本文方法的性能验证分析:

(1)Coopedge:基于区块链的边缘计算任务卸载算法,决策指标为边缘服务提供者的历史卸载信誉和时延的加权和,卸载信誉权重$\omega_{1\,\text{coop}}=0.6$,时延权重$\omega_{2\,\text{coop}}=0.4$,卸载信誉权重随时间减少的速度$\rho_{\text{coop}}=0.2$.

(2)原始遗传算法(Original Genetic Algorithm, OGA):采用与BROA相同算子的遗传算法,不考虑可靠性评估,决策指标为任务的完成时间与卸载成本.

(3)基于计算速度的贪心算法(computational Speed based Greedy Algorithm,SGA)优先选择计算速度最快的资源以满足任务时间期限.

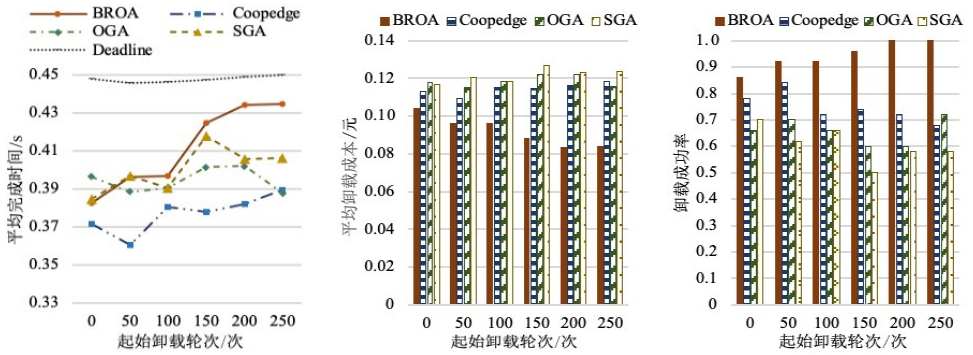
每完成50次任务卸载后,统计这50次卸载结果进行效果评估. 评价指标为平均任务完成时间、卸载成本和卸载成功率.

5.2 实验结果与分析

结合第4节的方法应用,分别在3种场景下,对两类任务的卸载效果进行实验分析.

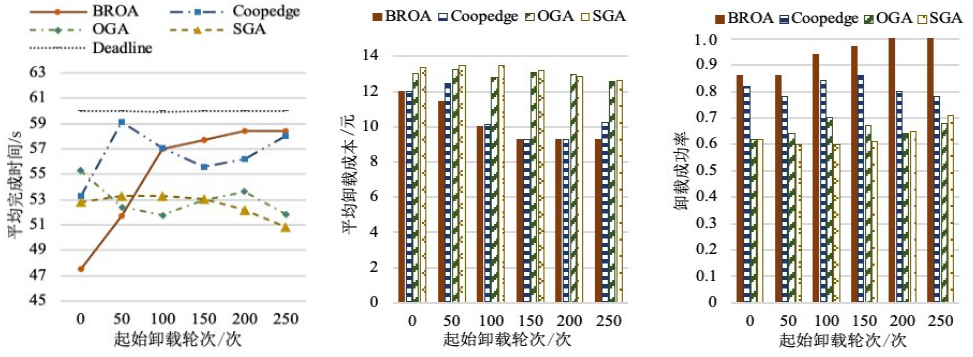
在场景1中卸载两类任务,实验结果如图4、图5所示. 相较于Coopedge,OGA和SGA,BROA能在平均完成时间满足任务时间期限(deadline)约束下,一直保持卸载成功率在85%以上,卸载成本最低. 随着卸载轮次增加,BROA可以保持较低的卸载成本和极高的卸载成功率. OGA和SGA由于未考虑可靠性评估,卸载成功率不稳定,卸载成本相对较高. Coopedge针对B类任务的卸载成功率提高更明显. 统计场景1的全部卸载结果,BROA相较于Coopedge平均优化卸载成本11.5%,平均卸载成功率高16.1%;相较于OGA平均优化卸载成本21.8%,平均卸载成功率高28.3%;相较于SGA平均优化卸载成本23.4%,平均卸载成功率高32.2%.

在场景2中卸载两类任务,实验结果如图6和图



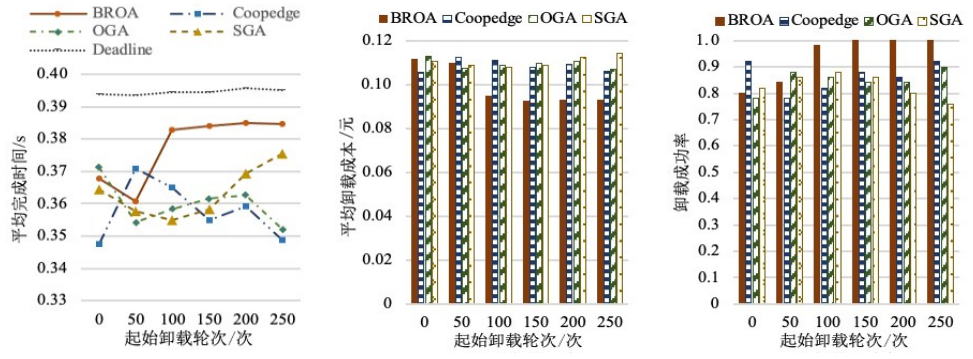
(a) 平均完成时间　(b) 平均卸载成本　(c) 卸载成功率

图4　场景1-A类任务卸载结果



(a) 平均完成时间　(b) 平均卸载成本　(c) 卸载成功率

图5　场景1-B类任务卸载结果

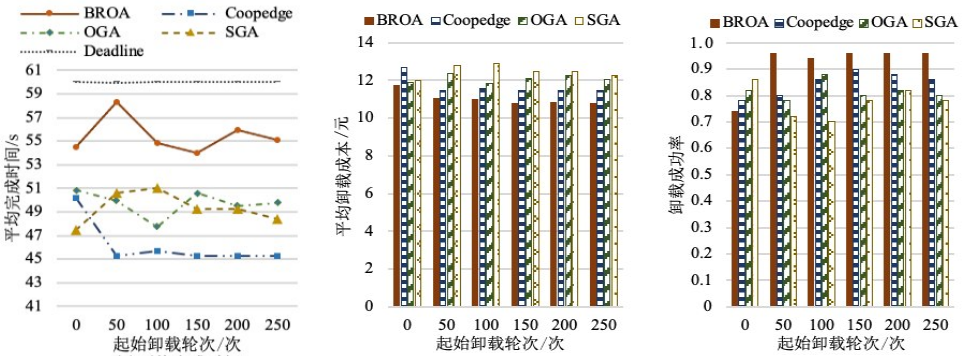


(a) 平均完成时间　(b) 平均卸载成本　(c) 卸载成功率

图6　场景2-A类任务卸载结果

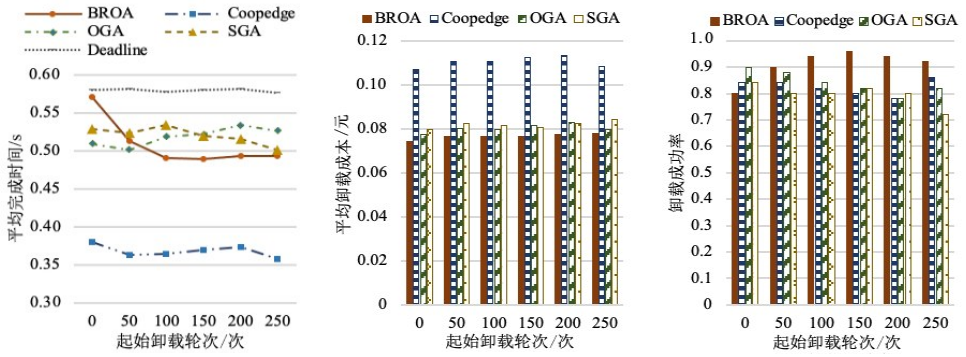
7所示.由于VM成功完成卸载任务的概率较场景1更高,BROA与Coopedge,OGA和SGA的卸载成功率差距缩小.BROA虽然在前50轮次的卸载成功率低于Coopedge,但从100轮次开始一直保持卸载成功率最高,平均卸载成本最低.统计场景2的全部卸载结果,BROA相较于Coopedge平均优化卸载成本7.2%,平均卸载成功率高7.3%;相较于OGA平均优化卸载成本9.1%,平均卸载成功率高9.5%;相较于SGA平均优化卸载成本11.0%,平均卸载成功率高12.5%.

在场景3中卸载两类任务,实验结果如图8和图9所示.场景3中的VM数量较少,在这种情况下,Coopedge卸载A类任务的平均完成时间较低,但卸载成本远高于BROA,OGA和SGA.BROA从50轮次后一直保持卸载成本最低,卸载成功率最高,且满足任务时间



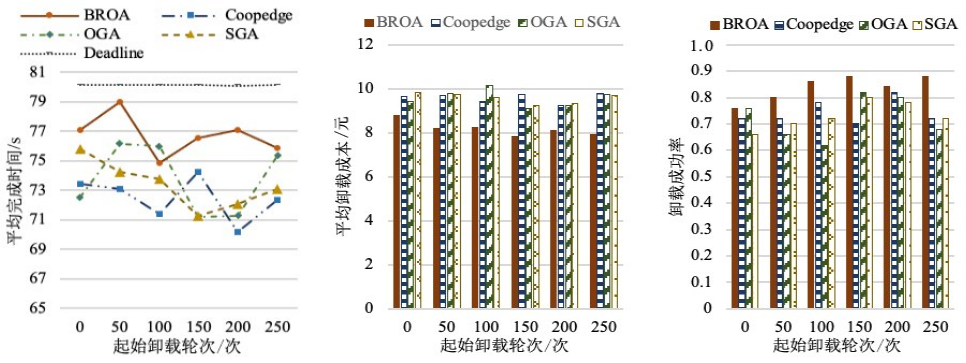
(a) 平均完成时间 (b) 平均卸载成本 (c) 卸载成功率

图7 场景2-B类任务卸载结果



(a) 平均完成时间 (b) 平均卸载成本 (c) 卸载成功率

图8 场景3-A类任务卸载结果



(a) 平均完成时间 (b) 平均卸载成本 (c) 卸载成功率

图9 场景3-B类任务卸载结果

期限约束. 统计场景3的全部卸载结果,BROA相较于Coopedge平均优化卸载成本22.7%,平均卸载成功率高9%;相较于OGA平均优化卸载成本9.7%,平均卸载成功率高9.2%;相较于SGA平均优化卸载成本10.6%,平均卸载成功率高11%.

从总体情况分析,BROA对计算量较小的A类任务卸载成本优化效果更好,相较于相关工作Coopedge平均优化卸载成本19.8%,平均卸载成功率提高11.9%. 在所设立的3种场景下,我们的算法对两类任务的卸载都能满足任务时间期限约束,总体保持较高的卸载成功率和较低的卸载成本.

6 总结与展望

针对边缘计算环境下任务卸载的数据安全和服务质量问题,本文设计了一个基于联盟链的可靠边缘计算任务卸载应用架构;进而基于遗传算法的技术应用,提出了一种基于联盟链的可靠边缘计算任务卸载方法,利用联盟链的身份校验和卸载结果反馈机制,获取任务卸载问题的优化解;最后在不同规模的应用场景和任务配置下进行实验,通过与相关工作和经典运筹学优化算法进行对比评估,验证了本文算法的有效性.实验结果表明,本文提出的方法能够在满足任务时间期限约束的前提下,有效降低卸载成本,提高卸载成功率.

目前已有基于联盟链的区块链网络投入商业应用,如亚马逊的 Amazon Managed Blockchain 和阿里云的 Blockchain as a Service 都使用联盟链 Hyperledger Fabric,但可用地域仅限4~6个关键的大型城市,且价格昂贵. 近年来许多研究工作针对区块链与雾计算、物联网等边缘计算相关技术的结合展开. 扩大联盟链网络覆盖区域,结合边缘计算提升服务能力,将是未来联盟链网络服务的发展方向[11,33]. 联盟链与边缘计算结合技术适用于终端设备多、隐私安全要求高的时延敏感型应用,例如智慧医疗系统[34,35]和无人机管理系统等[36,37]. 以智慧医疗系统为例,大量医疗图像识别任务和患者数据分析任务,在进行卸载时需要保障患者的数据隐私安全,而基于联盟链的边缘计算支持细粒度访问的低时延任务卸载,应用本文方法,可以在满足任务时间期限约束、保护患者隐私数据的情况下,提高任务卸载成功率,降低卸载成本. 我们将在未来工作中,进一步研究本文方法投入应用的针对性优化.

参考文献

[1] 思科中国.VNI 完整预测聚焦[EB/OL]. [2018](2021). https://www.cisco.com/c/m/zh_cn/solutions/service-provider/vni-forecast-highlights.html.

[2] KHAN W Z, AHMED E, HAKAK S, et al. Edge computing: A survey[J]. Future Generation Computer Systems, 2019, 97: 219-235.

[3] ALROWAILY M, LU Z. Secure edge computing in IoT systems: Review and case studies[C]//2018 IEEE/ACM Symposium on Edge Computing (SEC). Piscataway: IEEE, 2018: 440-444.

[4] QI L, WANG X, XU X, et al. Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing[J]. IEEE Transactions on Network Science and Engineering, 2021, 8(2): 1145-1153.

[5] ROMAN R, LOPEZ J, MAMBO M. Mobile edge computing, fog: A survey and analysis of security threats and challenges[J]. Future Generation Computer Systems, 2018, 78: 680-698.

[6] XU X, HUANG Q, ZHU H, et al. Secure service offloading for Internet of vehicles in SDN-enabled mobile edge computing[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(6): 3720-3729.

[7] LE M, SONG Z, KWON Y W, et al. Reliable and efficient mobile edge computing in highly dynamic and volatile environments[C]//2017 Second International Conference on Fog and Mobile Edge Computing (FMEC). Piscataway: IEEE, 2017: 113-120.

[8] YANG Y, CHANG X, HAN Z, et al. Delay-aware secure computation offloading mechanism in a fog-cloud framework[C]//2018 IEEE SPA. Piscataway: IEEE, 2018: 346-353.

[9] 周平, 殷波, 邱雪松, 等. 面向服务可靠性的云资源调度方法[J]. 电子学报, 2019, 47(5): 1036-1043.
ZHOU P, YIN B, QIU X S, et al. Service reliability oriented cloud resource scheduling method[J]. Acta Electronica Sinica, 2019, 47(5): 1036-1043. (in Chinese)

[10] 蔡晓晴, 邓尧, 张亮, 等. 区块链原理及其核心技术[J]. 计算机学报, 2019, 42(1): 84-131.
CAI X Q, DENG Y, ZHANG L, et al. The principle and core technology of blockchain[J]. Chinese Journal of Computers, 2021, 44(1): 84-131. (in Chinese)

[11] YANG R, YU F R, SI P, et al. Integrated blockchain and edge computing systems: A survey, some research issues and challenges[J]. IEEE Communications Surveys & Tutorials, 2019, 21(2): 1508-1532.

[12] XU X, PAUTASSO C, ZHU L, et al. The blockchain as a software connector[C]//2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA). Piscataway: IEEE, 2016: 182-191.

[13] YANG C S, TAO X L, ZHAO F, et al. Secure data transfer and deletion from counting bloom filter in cloud computing[J]. Chinese Journal of Electronics, 2020, 29(2): 273-280.

[14] DAI M L, XU S Y, SHAO S J, et al. Blockchain-based reliable fog-cloud service solution for IoT[J]. Chinese Journal of Electronics, 2021, 30(2): 359-366.

[15] 秦超霞, 郭兵, 沈艳, 等. 区块链的安全风险评估模型[J]. 电子学报, 2021, 49(1): 117-124.

QIN C X, GUO B, SHEN Y, et al. Security risk assess ment model of blockchain[J]. Acta Electronica Sinica, 2021, 49(1): 117-124. (in Chinese)

[16] KOLVART M, POOLA M, RULL A. Smart contracts [M]//The Future of Law and Etechnologies. Cham: Springer, 2016: 133-147.

[17] CHRISTIDIS K, DEVETSIKIOTIS M. Blockchains and smart contracts for the internet of things[J]. IEEE Access, 2016, 4: 2292-2303.

[18] XU J, WANG S, ZHOU A, et al. Edgence: A blockchain-enabled edge-computing platform for intelligent IoT-based dapps[J]. China Communications, 2020, 17(4): 78-87.

[19] PUTRA G D, DEDEOGLU V, KANHERE S S, et al. Trust-based blockchain authorization for IoT[J]. IEEE Transactions on Network and Service Management, 2021, 18(2): 1646-1658.

[20] XU X, ZANG X, GAO H, et al. Become: Blockchain-enabled computation offloading for IoT in mobile edge computing[J]. IEEE Transactions on Industrial Informatics, 2019, 16(6): 4187-4195.

[21] SINGH S, BAJPAI P, SINGH A, et al. A new incentive based algorithm for avoiding free riding in peer to peer network[C]//2018 International Conference on Computational and Characterization Techniques in Engineering & Sciences (CCTES). Piscataway: IEEE, 2018: 156-159.

[22] YAO W, YE J, MURIMI R, et al. A survey on consortium blockchain consensus mechanisms[EB/OL]. (2021) [2021]. https://arxiv.org/abs/2102.12058.

[23] 袁勇, 倪晓春, 曾帅, 等. 区块链共识算法的发展现状与展望[J]. 自动化学报, 2018, 44(11): 2011-2022.
YUAN Y, NI X C, ZENG S, et al. Blockchain consensus algorithms: The state of the art and future trends[J]. Acta Automatica Sinica, 2018, 44(11), 2011-2022. (in Chinese)

[24] 乔蕊, 曹琰, 王清贤. 基于联盟链的物联网动态数据溯源机制[J]. 软件学报, 2019, 30(6): 1614-1631.
QIAO R, CAO Y, WANG Q X. Traceability mechanism of dynamic data in Internet of Things based on consortium blockchain[J]. Journal of Software, 2019, 30(6): 1614-1631. (in Chinese)

[25] QI L, CHEN Y, YUAN Y, et al. A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems[J]. World Wide Web, 2020, 23(2): 1275-1297.

[26] ZHENG Z, XIE S, DAI H, et al. An overview of blockchain technology: Architecture, consensus, and future trends[C]//2017 IEEE International Congress on Big Data. Piscataway: IEEE, 2017: 557-564.

[27] YUAN L, HE Q, TAN S, et al. Coopedge: A decentralized blockchain-based platform for cooperative edge computing[C]//Proceedings of the Web Conference 2021. New York: ACM, 2021: 2245-2257.

[28] GUO H, LI W, NEJAD M, et al. Access control for electronic health records with hybrid blockchain-edge architecture[C]//2019 IEEE International Conference on Blockchain. Piscataway: IEEE, 2019: 44-51.

[29] CECHIN C. Architecture of the hyperledger blockchain fabric[C]//Workshop on Distributed Cryptocurrencies and Consensus Ledgers. New York: ACM, 2016: 310-314.

[30] HOUSLEY R, FORD W, POLK W, et al. RFC2459 Internetx. 509 Public Key Infrastructure Certificate and Crl Profile[R/OL]. (1999) [2021]. https://tools.ietf.org/html/rfc2459.

[31] WHITLEY D. A genetic algorithm tutorial[J]. Statistics and computing, 1994, 4(2): 65-85.

[32] BLICKLE T. Tournament selection[J]. Evolutionary Computation, 2000, 1: 181-186.

[33] WU Y, XU X, QIAN L, et al. Revenue-sharing based computation-resource allocation for mobile blockchain [C]//IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Piscataway: IEEE, 2020: 56-61.

[34] 邱宇, 王持, 齐开悦, 等. 智慧健康研究综述: 从云端到边缘的系统[J]. 计算机研究与发展, 2020, 57(1): 53-73.
QIU Y, WANG C, QI K Y, et al. A survey of smart health: System design from the cloud to the edge[J]. Journal of Computer Research and Development, 2020, 57(1): 53-73. (in Chinese)

[35] AUJLA G S, JINDAL A. A decoupled blockchain approach for edge-envisioned IoT-based healthcare monitoring[J]. IEEE Journal on Selected Areas in Communications, 2020, 39(2): 491-499.

[36] DAWALIBY S, ABERKANE A, BRADAI A. Blockchain-based IoT platform for autonomous drone operations management[C]//Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond. New York: ACM, 2020: 31-36.

[37] POKHREL S R. Federated learning meets blockchain at 6G edge: A drone-assisted networking for disaster re-

sponse[C]//Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond. New York: ACM, 2020: 49-54.

作者简介

许悦玥　女,1997年生,四川乐山人. 现为南京大学计算机科学与技术系硕士. 主要研究方向为边缘计算.
E-mail: xuyuey@smail.nju.edu.cn

刘博文　男,1997年生,辽宁大连人. 现为南京大学计算机科学与技术系博士. 主要研究方向为边缘计算.
E-mail: liubw@smail.nju.edu.cn

田　臣　男,1978年生,湖北武汉人. 现为南京大学计算机科学与技术系教授. 主要研究方向为数据中心网络、分布式系统等. 中国电子学会会员编号:E190092288M.
E-mail: tianchen@nju.edu.cn

戴海鹏　男,1985年生,湖南双峰人. 现为南京大学计算机科学与技术系副教授. 主要研究方向为数据挖掘、无线网络等.
E-mail: haipengdai@nju.edu.cn

郑嘉琦　男,1986年生,河北平顶山人. 现为南京大学计算机科学与技术系副研究员. 主要研究方向为大规模计算机网络.
E-mail: jzheng@nju.edu.cn

陈贵海　男,1968年生,江苏盐城人. 现为南京大学计算机科学与技术系教授. 主要研究方向为互联网络、高性能计算机架构等.
E-mail: gchen@nju.edu.cn

窦万春　男,1971年生,江苏徐州人. 现为南京大学计算机科学与技术系教授. 主要研究方向为大数据与边缘计算等.
E-mail: douwc@nju.edu.cn