



边缘计算多约束可信协同任务迁移策略

乐光学^{1,2}, 戴亚盛², 杨晓慧², 朱友康², 游真旭², 刘建生²

(1. 嘉兴学院数理与信息工程学院, 浙江 嘉兴 314001; 2. 江西理工大学理学院, 江西 赣州 341100)

摘要: 为保障边缘计算的服务质量, 提出一种在多约束条件下边缘计算可信协同任务迁移策略。该策略基于任务需求, 由边缘计算协同服务盟主节点组织调度协同服务盟员, 基于用户任务迁移的 K 维权重指标, 确定协同盟员调度优先级, 以盟员负载均衡性为适应函数, 通过贪心算法执行盟员任务分配与调度, 基于路由捎带选择备用节点, 通过迁移优先级评估, 实现协同服务异常时的调度和迁移, 由此提高边缘计算任务迁移的服务质量, 保障任务迁移的可靠性。仿真实验表明, 该机制能有效完成协同任务分发与迁移调度, 提高边缘计算协同效率, 保障网络服务质量。

关键词: 边缘计算; 任务调度; 多维优化; 负载均衡

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2019275

Multi-constrained trusted cooperative task migration strategy for edge computing

YUE Guangxue^{1,2}, DAI Yasheng², YANG Xiaohui², ZHU Youkang², YOU Zhenxu², LIU Jiansheng²

1. School of Mathematics and Information Engineering, Jiaxing University, Jiaxing 314000, China

2. School of Science, Jiangxi University of Science and Technology, Ganzhou 341100, China

Abstract: In order to guarantee the quality of service of edge computing, a trusted cooperative task distribution strategy for edge computing under multi-dimensional constraints was proposed. The strategy was based on the task requirements, and the edge computing collaborative service leader node organizes the coordinated service alliance members. The priority of cooperative service alliance was determined based on the K -dimension weight index of user task migration. Using the load balancing of the members as the adaptive function, the task allocation and dispatch of alliance members were performed by greedy algorithm. The backup node was selected based on routing piggyback, and the priority of migration was evaluated. The scheduling and migration of collaborative services in case of abnormal collaborative services were realized, which improved the quality of service of edge computing task migration and ensured the reliability of task migration. The simulation results show that the mechanism can effectively complete the task distribution and migration scheduling, improve the collaborative efficiency of edge computing, and guarantee the quality of network service.

Key words: edge computing, task scheduling, multi-constrained optimize, load balancing

收稿日期: 2019-03-15; 修回日期: 2019-06-05

基金项目: 国家自然科学基金资助项目 (No.61462036); 浙江省自然科学基金资助项目 (No.LY16F020028)

Foundation Items: The National Natural Science Foundation of China (No.61462036), The National Natural Science Foundation of Zhejiang Province of China (No.LY16F020028)

1 引言

2018 年 12 月 10 日,工业和信息化部向三大运营商颁发了 5G 系统中低频段试验频率使用许可^[1],标志着 5G 时代正式到来,预计于 2020 年实现全国普及。5G 网络时代高清视频、VR/AR、物联网等数据密集型应用成为通信网络流量的主要业务,对网络带宽、能耗、时延和安全等提出了更高的要求。以 3G、4G 网络为代表的集中式云计算模式,将不能满足高效处理边缘设备产生的数据需求^[2-4]。研究表明,通过在网络边缘局部汇聚计算、存储和通信资源,构建以云计算为核心、分布式扩展边缘计算的模式是解决该问题的有效方法之一^[2-4]。边缘计算将成为 5G 网络的典型应用之一,具有高速、可靠和低时延的特点,受到了广泛关注^[3]。

边缘计算是指在网络边缘提供 IT 服务环境和云计算能力,以减少网络操作和服务交付的时延^[2-3]。当边缘计算面临浪涌式请求时,由于基站节点的带宽和计算资源有限,大量任务聚集造成访问拥塞,数据链路过载形成路由热区,一旦基站节点负载超过临界值,则其服务分发能力将呈指数级下降,甚至出现雪崩效应导致网络系统瘫痪。为解决上述问题,需通过有效的计算迁移和任务卸载机制,基于基站和边缘设备间的协作,有效完成任务分发与迁移调度,从而缓解任务聚集和路由热度,提高网络服务质量。

2 相关研究

任务调度机制是计算系统的关键技术,其中,以缩短任务调度时间或队列长度为优化目标的算法如最早完成时间(heterogeneous earliest finish time, HEFT)算法^[5]、关键路径(critical path on a processor, CPOP)算法^[5]等;以时延为优化目标的算法如 STARS 算法^[6]、MOWS-DTM 算法^[7];以节约资源、提高调度效率为优化目标的算法如

人工鱼群(artificial fish swarm algorithm, AFSA)算法^[8]、感知缓存任务调度算法^[9]等;综合时间和能耗约束的调度算法如 SA-DVSA 算法^[10]等。上述算法可应用于本地计算任务处理调度中,但由于安全隐私、实时性等原因,不能较好地应用于边缘计算节点间协同服务任务调度中。

参考文献[11]提出了一种实时、上下文感知的边缘计算协同框架,综述了移动边缘协调、协作缓存处理和多层干扰消除 3 个方面的边缘计算服务器和移动设备间协作的研究,证明了所提出的方法在促进 5G 网络发展方面的巨大优势,并提出了边缘计算融合 5G 网络需要解决的关键技术挑战和开放性研究问题。参考文献[12]提出了一种移动边缘计算系统构建协同服务的方法,通过用户节点、协同节点和与边缘服务器相连的访问点(AP)节点构建协同通信链路,针对有限块上用户的时延约束计算问题,提出了一种实现联合计算和通信协作的四时隙协议,优化用户和协同节点的计算和通信资源分配,在给定计算时延约束下,最大限度地减少了总能耗。参考文献[13]提出了一种新的边缘计算协同方法,通过 AP 与移动设备构建中继协同通信链路进行卸载提出两阶段法,优化卸载决策并节约能量,实现 AP 的总发射能量最小化。参考文献[14]研究了随机无线信道下移动设备与云克隆协同任务执行问题,提出了一种在满足时间期限的同时,通过战略性地将任务卸载到云端,最大限度地减少移动设备上的能耗,并基于单爬升策略,提出一种多项式时间内协同任务执行的枚举算法,可获得近似最优解,实验结果表明,该方法可以显著节省移动设备的能耗,延长使用周期。参考文献[15]基于大规模 MIMO 天线和毫米波通信技术,将 5G 超密集蜂窝网络部署到蜂窝场景中,并对 5G 超密集网络的最大容量、能耗及效率问题进行研究。参考文献[16]分析比较了不同网络结构和频段下 5G 无线回程网络的能量效率,指出基于分布式的 5G 回程网络



相比集中式具有更优的能量效率,对未来 5G 回程网络发展提出挑战。

参考文献[17]提出了一种边缘计算卸载策略,通过建立从用户为中心的能量感知的移动性管理(EMM)方案,从而优化用户在长期能量消耗约束下的无线接入和计算时延。基于李雅普诺夫优化和多臂抽奖理论,EMM 在没有未来系统状态信息的情况下进行在线工作,有效地处理了不完善的系统状态信息。仿真结果表明,该算法在满足用户能耗约束的前提下,可以获得接近最优的时延性能。参考文献[18]提出了一种节能的动态卸载和资源调度策略,以降低能耗并缩短应用程序完成时间。该方法在满足任务依赖性要求和完成时间期限约束的前提下,基于分布式 EDORS 算法,通过计算卸载选择、时钟频率控制和传输功率分配 3 个子算法,证明了优化的 EDORS 算法可以有效地降低能耗并提高卸载效率。参考文献[19]提出了一种数据关联性任务迁移优化方法 D3S2 (data-dependency-driven scheduling scheme),旨在通过挖掘数据间的关联性,将强关联的数据放置相同位置以减少任务迁移,基于模块化思想对数据进行整体调度,以最小化数据迁移代价。参考文献[20]在考虑完成时间和能量的基础上,将计算卸载问题转化为系统能效成本最小化问题,通过求解优化问题,提出了一种由时钟频率配置、传输功率分配、信道速率调度和卸载策略选择组成的分布式算法。该算法将本地执行的时钟频率与边缘云执行的传输功率进行联合优化,减少了任务卸载能耗,并提高了迁移效率。

参考文献[21]考虑执行者间的协作相容性,提出了基于协同相容、负载均衡的任务分配模型,并基于多目标联合优化的任务分配方法,提高整个流程实例的执行效率,并保持执行者的负载均衡。参考文献[22]提出了一种新的集群架构的边缘流媒体服务器的设计,并针对集群服务器中的负载均衡问题,提出 MCLBS 缓存替换算法自适应

优化边缘服务器负载,相比已有的缓存替换算法,缓存命中率明显提高,降低了服务占用带宽。参考文献[23]提出了最小化终端的传输时延和最大化其满意度的分布式算法,优化缓存文件分配,提高分发性能。参考文献[24]针对分布式并行处理架构中由于数据本身的倾斜分布以及数据流本身的实时、动态和不可预知导致负载不均衡现象,提出了通过 Key 粒度迁移和元组粒度拆分进行轻量级均衡调整以保证系统负载均衡。

针对边缘计算协同任务卸载和迁移调度问题,已有的研究均针对能耗、时延等特定指标进行优化^[12-30],在其实验场景下效果显著,但基于用户个性化的任务需求,在多维约束条件下进行任务迁移优化,保证边缘计算系统负载均衡的问题,尚待研究。

本文提出一种边缘计算多约束下可信协同任务分配策略。其核心思想是基于任务需求约束,选择可信、可靠的盟员,以用户个性化的多维需求为目标,优化求解盟员均匀分配执行任务的方案,基于路由捎带技术选择备用节点,实现协同服务异常时调度和迁移。由此提高边缘计算任务迁移的服务质量,保障协同盟员的可信性和任务迁移的可靠性。

3 边缘计算可信协同服务与任务调度模型

边缘计算网络由基站(base station, BS)、边缘云服务器(edge cloud server, ECS)和边缘节点(edge node, EN)等组成。基站之间以高速互联网络相连构成边缘计算骨干网络,边缘节点之间可通过自组网方式实施通信。边缘节点通过有线或无线多跳网络接入基站,上传或下载处于边缘云服务器中的资源。基于参考文献[31]通过基站和物联网设备协作的 COMED 框架思想,本文提出每个基站与其通信范围内提供共享计算资源的边缘节点可扩展构建独立的边缘计算协同服务池(edge computing cooperation service pool),并

在边缘计算协同服务池中设立盟主，完成协同服务组织调度及任务分发等管理功能，边缘计算协同服务如图 1 所示。边缘节点和边缘云服务器的计算任务可通过调度本地执行或迁移至边缘计算协同服务池执行。当边缘节点发起任务迁移时，盟主节点启动网络资源聚合机制，快速完成盟员节点间的聚合，构建边缘计算协同服务，分发和映射盟员的任务信息，协同求解目标任务。

盟主节点通常由基站担任，基站节点状态和通信链路质量将直接影响边缘计算协同服务的质量。在协同服务规模较小、协同任务密度不大的情况下，其服务能力是满足需求的。当协同规模超过阈值、边缘计算协同服务池面临浪涌式协同任务请求时，由于基站节点服务能力有限，大量任务聚集造成访问拥塞，数据链路过载，任务处理时延过高，协同服务将出现瓶颈。一旦基站节点负载超过临界值，其服务分发能力将呈指数下降，甚至出现雪崩效应导致网络系统瘫痪。为保障服务质量，在协同规模较大或任务较密集的边缘计算协同服务池中，盟主节点的功能可由基站按一定策略分发至具有一定服务能力且可信、可

靠、安全的盟员节点，这样的节点称为超级节点（super node, SN）。SN 享有代理盟主组织调度、分发边缘计算协同任务等权利，同时承担保障边缘计算满足 QoS 需求和可信、可靠交付的义务。

3.1 边缘计算任务建模

在边缘计算协同服务应用中，实时任务可划分为具有依赖关系的子任务，其依赖关系包括串行和并行。串行关系是指当前任务必须等待前置任务执行完成后方可执行，并行关系是指任务可以同时执行直到后续汇聚任务执行。当实时任务在本地执行时，若出现超出负载等异常事件，需要进行任务迁移时，其并行子任务序列将作为独立的迁移计算任务单元，从本地迁移至边缘计算协同服务池，由盟主组织边缘计算协同节点，构建协同服务调度执行。

由上述可知，盟主在构建协同服务执行任务分配时，本质上是对迁移计算任务单元的分配，并满足如下约束。

(1) 当盟主节点分配迁移计算任务单元时，任务迁移耗时和盟员处理任务的执行时间不得超

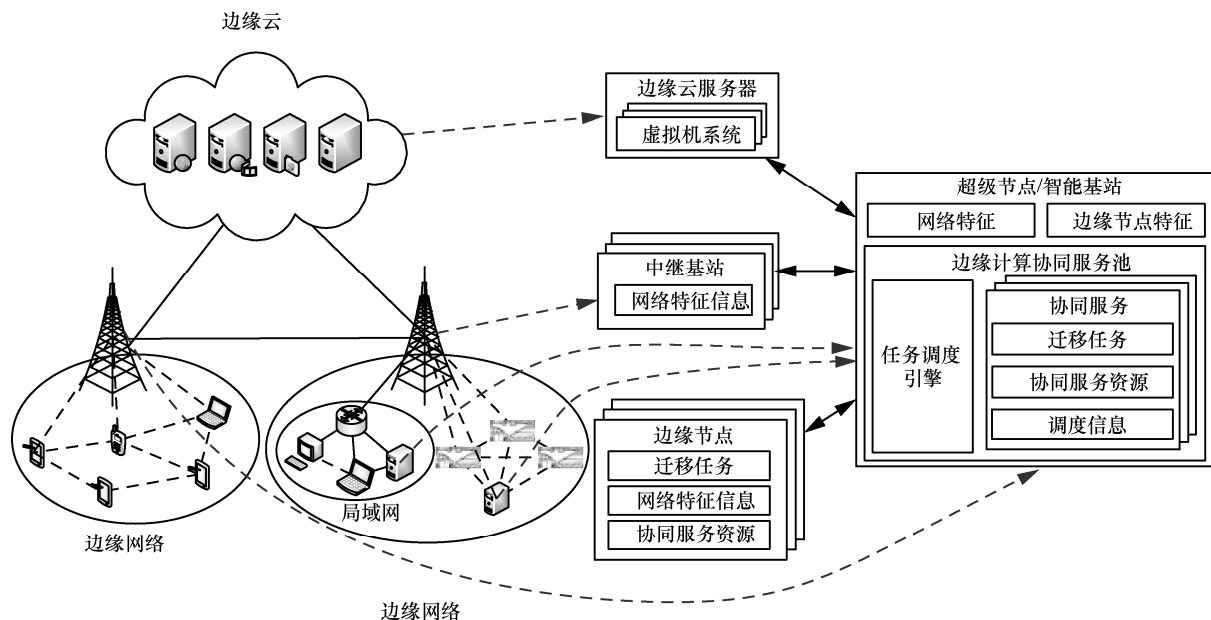


图 1 边缘计算协同服务



过最终截止时间,即满足式(1):

$$\text{Max}(d_{e_1} + t_{e_1}, d_{e_2} + t_{e_2}, \dots, d_{e_n} + t_{e_n}) \leq d_s \quad (1)$$

其中, e_i 表示协同服务盟员节点, d_{e_i} 表示盟员处理任务的执行时间, t_{e_i} 表示节点 e_i 任务迁移占用时间, d_s 为节点发起迁移计算任务时定义的任务截止交付时间。

(2) 迁移任务所占用的带宽 W_{e_i} 不得超过盟员节点提供的共享传输带宽 w_{e_i} , 迁移任务单位时间的计算量 P_{e_i} 不能超过盟员节点的计算能力 ρ_{e_i} , 如式(2):

$$\begin{cases} B_{e_i} \leq b_{e_i} \\ P_{e_i} \leq \rho_{e_i} \end{cases} \quad (2)$$

迁移计算任务单元的任务量和计算耗时可参考文献[31], 通过已执行过同种任务节点的执行日志进行估算, 这里不再详述。

(3) 盟员节点完成迁移任务后, 剩余能量不得低于设置的能量阈值, 如式(3):

$$E_{e_i} - Er_{e_i} - Ec_{e_i} \geq e_{ei} \quad (3)$$

其中, Er_{e_i} 为任务迁移所消耗的传输能量, Ec_{e_i} 为计算消耗能量, E_{e_i} 为盟员任务迁移前所剩余的能量。

上述约束是盟主选择边缘协同服务盟员的依据。

3.2 边缘计算协同盟员建模

协同服务盟主以第2.1节的任务约束条件, 选择协同服务盟员, 构建协同服务。通常参与边缘计算协同服务的盟员会经历5个状态 $Q = \{T_0, T_1, T_2, T_3, T_4, T_5\}$, 如图2所示。

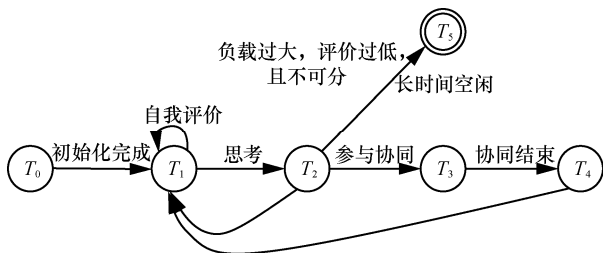


图2 边缘计算协同服务盟员状态

(1) T_0 状态表示盟员尚处于协商阶段, 该状态盟员尚未初始化, 不提供任何服务。

(2) T_1 状态表示盟员初始化完成, 进入等待状态, 该状态下协同节点积极参与协同任务, 并在每个时间周期中完成自我评价。

(3) T_2 状态表示盟员处于思考状态, 由 T_1 完成自我评价后进入重要的决策期, 在决策期内思考是否接受协同任务或退出服务。

(4) T_3 状态表示盟员处于执行状态, 该状态下盟员参与协同服务, 执行分发任务。

(5) T_4 状态表示盟员处于结束状态, 该状态下盟员执行任务完成, 回传执行结果, 并进入 T_1 状态, 执行自我评价。

(6) T_5 状态表示盟员处于终止状态, 该状态下盟员将逐渐中止当前协同服务, 释放协同服务占用的资源。

由于不同盟员节点所处的服务状态不尽相同, 且受到应用服务时延、抖动、分组丢失、地理位置等限制, 其提供计算、存储、网络等边缘计算服务时具有差异。分析可知, 参与协同服务的盟员节点具有如下特点。

(1) 差异性

由于盟员节点的计算配置、通信设备、网络环境具有差异, 其提供的可共享边缘计算协同服务资源差异较大。

(2) 独立性

盟员节点对本身提供的共享资源具有完全控制决定权, 加入、退出、拒绝协同服务等均由盟员节点决定。由于协同盟员节点的独立性, 无法预知盟员节点的行为, 这有可能造成协同服务质量和效率不高的问题。

由于协同服务盟员具有差异且独立自主, 盟主节点在选择协同服务盟员时, 必须建立公平、可靠的盟员节点协同服务能力评价机制。假设当节点 e_i 向节点 e_j 提供协同服务时, 节点 e_i 的服务能力 $\eta_{e_i e_j}$ 如式(4):

$$\eta_{e_i e_j} = G_{e_i} \sqrt{\text{Rtt}_{e_i e_j}} W_{e_i} \rho_{e_i} T_{e_i} Q_{e_i} St_{e_i} \left(1 - \frac{Er_{e_i} + Ec_{e_i}}{E_{e_i} - e_{ei}} \right) \alpha \quad (4)$$

其中, G_{e_i} 为节点 e_i 可用性 ($G_{e_i} \in \{0,1\}$), 当节点 e_i 处于可接收协同任务的状态时, $G_{e_i}=1$, 否则为 0。 $\text{Rtt}_{e_i e_j}$ 为节点 e_i 与节点 e_j 的时延, W_{e_i} 为节点 e_i 的带宽, ρ_{e_i} 为节点 e_i 的计算能力, T_{e_i} 为节点 e_i 的信誉度, 信誉度计算将在第 4.1 节讨论, Q_{e_i} 为节点 e_i 的实时负载评价, St_{e_i} 为节点 e_i 的预期服务时间, α 为放大系数。 $\eta_{e_i e_j}$ 越大, 则节点 e_i 向节点 e_j 提供服务的能力越强。

负载评价 Q_{e_i} 的计算如式 (5):

$$Q_{e_i} = \begin{cases} 1 - \frac{v_{e_i}}{CW_{e_i}}, & v_{e_i} < CW_{e_i} \\ 0, & v_{e_i} \geq CW_{e_i} \end{cases} \quad (5)$$

其中, v_{e_i} 为当前速率, C 为负载控制阈值。 Q_{e_i} 越大, 则节点的负载越低, 服务能力越强。

3.3 边缘计算协同服务任务迁移机制

3.3.1 协同服务任务迁移请求

边缘节点发起任务迁移时, 通常无法预知协同任务的服务质量, 如具体的时延、能耗、完成时间等, 但对其具有一定期望。根据不同的应用服务, 其服务质量需求是不同的, 例如, 实时通信类业务着重于时延、抖动, 在线流媒体服务着重于时延、传输速率。

基于用户多维期望权重 EWATX 思想^[31], 本文定义节点对服务质量期望权重为 $\omega = (a_1, a_2, \dots, a_K)$, 满足:

$$\sum_{i=1}^K |a_i| = 1 \quad (6)$$

其中, $K \geq 1$, $-1 < a_i < 1$, a_i 的正负代表节点对某一指标是越大越好或越小越好。例如, 设 $K=2$, 服务质量权值 a_1 代表节点对时延的重视程度, a_2 代表节点对传输速率的重视程度, 通常 $a_1 < 0$, 代表节点希望时延越小越好; $a_2 > 0$, 代表节点希望传输速率越大越好。

节点发起任务迁移时, 将上述期望捆绑至迁移请求分组消息头中, 发送至边缘计算协同服务盟主。通常边缘计算系统具有多个独立的协同服务池, 节点收到协同服务盟主反馈的协同服务池状态后, 节点需自主决策任务迁移请求协同服务对象。基于多径路由和就近原则的思想, 假设节点 e_i 存在多个可达的协同服务盟主节点集合 $\Gamma = \{q_1, q_2, \dots, q_n\}$, 则节点 e_i 选择协同服务盟主节点 q_j 为自身服务的概率 $p_{e_i q_j}$ 如式 (7):

$$p_{e_i q_j} = \frac{\eta_{q_j} \sqrt{\text{Rtt}_{e_i q_j}}}{\sum_{k=1}^m \eta_{q_k} \sqrt{\text{Rtt}_{e_i q_k}}} \quad (7)$$

迁移请求具体步骤如下。

(1) 任务迁移服务协议初始化, 设定轮询阈值 τ , 当前轮寻次数 $t=0$, 通过邻居监听、泛洪查询等方式记录所有可达的协同服务盟主节点。

(2) 计算所有可达的协同服务盟主迁移优先级概率 $p_{e_i q}$, 按优先级概率随机选择盟主节点 q_j 作为迁移服务节点, 轮寻次数 $t++$ 。

(3) 与协同服务盟主 q_j 建立传输链路后, 倘若 $t < \tau$ 或迁移服务效率过低无法满足第 3.1 节任务需求约束时, 则放弃该节点, 跳转步骤 (2) 重新选择; 若 $t > \tau$ 则表示迁移服务请求失败, 放弃迁移服务请求。

(4) 记录当前传输路径, 并将节点加入备选迁移集中, 如果发生重新迁移请求时, 优先选择。

3.3.2 协同服务任务迁移响应

盟主节点首先在辖域内的虚拟协同服务池中搜索可参与此次协同服务的节点, 并基于优先级调度分配协同任务。若当前协同服务资源不足时, 则向其他协同服务盟主请求协同调度协同服务盟员。

假设当前盟主节点具有协同服务盟员集 E , 针对发起任务迁移的节点提出的 K 维权重指标, 获取每个协同服务盟员节点 $e_j \in E$ 的相应服务参



数 $X_{e_i} = (b_1^{e_i}, b_2^{e_i}, \dots, b_K^{e_i})$ 。计算每个协同服务盟员的平均离差度 σ ：

$$\sigma(X_{e_i}) = \sum_{k=1}^K a_k \sqrt{(b_k^{e_i} - \bar{b}_k)^2} \quad (8)$$

其中, $\bar{b}_k$ 表示协同服务盟员节点相应服务参数的均值, 离差度 σ 表示该协同服务盟员间的相似性, 通常离差度 σ 过高表示某一指标过大或过小, 无法成为协同服务盟员, 将被摒弃。基于离差度评价后, 将按照优先级评价对协同服务盟员进行排序, 优先级计算 $t(X_{e_i})$ 如式 (9):

$$t(X_{e_i}) = \eta_{e_i q} \sum_{k=1}^K a_k b_k^{e_i} \quad (9)$$

其中, $\eta_{e_i q}$ 为当前协同服务盟主节点 q 对盟员节点 e_i 的服务能力评价。

根据盟员优先级 $t(X_{e_i})$, 以协同服务盟员平均负载为适应值, 当协同服务盟员负载未超出平均水平时, 则分配任务, 若已超出平均水平, 则轮询下一个节点, 直至所有任务被分配完毕。上述贪心求解过程表述为伪代码如下。

算法 1 K 维贪心算法

Input $E = (e_1, e_2, \dots, e_n)$, $\omega = (a_1, a_2, \dots, a_K)$, $S = (s_1, s_2, \dots, s_M)$

Initialize $t \leftarrow (t_1, t_2, \dots, t_n)$, $S' \leftarrow (S'_1, S'_2, \dots, S'_n) // t$ 为迁移优先级, S' 为每个节点的任务量

For each e_i **in** E **do** // 计算盟员的优先级

$t_i \leftarrow e_i \times a_i$

End for

Sort E **by** t // 根据优先级排序

For each s_i **in** S **do**

For each e_i **in** E **do**

$S'_i \leftarrow \frac{s_i}{\rho_i}$

End for

$S'_{\text{average}} \leftarrow \text{Average}(S'_i) //$ 计算平均适应值

For each e_i **in** E **do**

If $S'_i > S'_{\text{average}}$ **then**

$S' \leftarrow s_i //$ 分配任务

End for

End for

3.3.3 协同服务任务迁移异常与自恢复

当协作任务发生链路中断、节点失效、节点可用服务能力或安全可信性无法满足任务需求时, 为减小传输链路重建带来的额外开销, 由异常点相关链路节点执行服务迁移, 迁移节点以中继增强或替代原节点的方式加入协作传输链路。

对于中断型数据链路, 其协同服务迁移的过程是强制性的, 故障区域的节点将放弃原传输链路, 并向邻居节点发送迁移请求, 节点收到迁移请求响应后, 根据任务需求、服务能力、迁移质量等参量选择迁移节点。节点 e_i 向节点 e_j 请求迁移时, 协同服务迁移优先级评价 $Y_{e_i e_j}$ 为:

$$Y_{e_i e_j} = \frac{\eta_{e_i e_j} E v_{e_i e_j}}{\sqrt{\delta}} \quad (10)$$

其中, δ 表示服务迁移将会带来的最大时延抖动, E 表示待迁移任务量, $v_{e_i e_j}$ 表示节点 e_i 向节点 e_j 的迁移速率。假设节点 e_i 的备选迁移协同节点集为 $E = \{e_1, e_2, \dots, e_n\}$, 期望任务迁移速率为 v , 则迁移选择策略如下。

(1) 任务迁移服务协议初始化, 设定轮询阈值 τ , 当前轮寻次数 $t = 0$ 。

(2) 将可迁移协作节点按迁移优先级排序, 计算每个节点作为迁移节点的概率 $P = \frac{Y_{ij}}{\sum_{k=1}^n Y_{ik}}$, 并选择节点 e_j 作为预选迁移节点, 轮寻次数 $t++$ 。

(3) 与节点 e_j 建立传输链路后, 若 $t < \tau$ 且 $\sum v_{ij} < v$, 则跳转 (2) 继续选择; 若 $t > \tau$ 则表示迁移服务请求失败, 放弃异常迁移服务请求。

(4) 将此次参与协作节点加入备选迁移集中, 如果发生重新迁移请求时, 优先选择。

4 边缘计算协同服务质量评价

4.1 边缘计算协同服务信誉评价模型

在边缘计算协同服务中,协作节点的选择直接关系到系统的服务质量,则需建立统一的信誉评价方案,本着自主、公平的原则,评价协同节点的信誉。节点信誉评价主观上是衡量节点的可信度,其意义包含了交付任务的可靠性、节点行为的良好程度,为保障节点信誉评价的公平性和有效性,节点的信誉由节点近期保持良好的任务交付行为所积累,历史交互信息对当前节点的信誉评价影响较小,以时间窗口采样所得的交互数据集对节点信誉进行评价。节点的信誉基准由节点近期完成的最大任务量决定,即节点完成了10个10 MB的任务量与节点完成了1个100 MB的任务量对比,为后者更优,因为能够完成大任务交付的节点具有更连续且稳定的网络时延和服务质量。同时,为维持良好的网络环境,节点信誉增益放缓,节点信誉惩罚力度较大,当节点连续出现交付失败或大任务交付失败时,其信誉度将快速崩塌,当信誉度为0时,节点网络将被限制交互,节点必须通过协助他人积累信誉,当惩罚轮次和时间过后,节点方可解除限制。

由上述规则可知,节点必须在近期能够稳定、可靠地完成大任务交付,才能快速提升信誉。根据规则,建立信誉评价模型:

$$T_i = T_{si} + T_{gi} - T_{ri} \quad (11)$$

其中, T_i 表示节点 i 的信誉度, T_{si} 表示节点 i 的信誉基础分, T_{gi} 表示节点 i 保持良好行为的信誉增益分, T_{ri} 表示节点 i 失败任务交付时的信誉减益分。 T_{si} 、 T_{gi} 和 T_{ri} 计算如式 (12):

$$\begin{cases} T_{si} = S_{\max} \\ T_{gi} = \mu \ln \left(\sum_{j=1}^n \frac{S_j}{S_{\max}} \right) + n \\ T_{ri} = \left(\sum_{l=1}^x \frac{S_l}{S_{\max}} \right) \times \varepsilon^x \\ \text{s.t. } \mu > 0, \varepsilon > 0 \end{cases} \quad (12)$$

其中, $S_{\max}$ 表示节点在近期交付的最大任务价值, S_j 表示节点近期第 j 次成功交付的任务价值, μ 为增益控制因子, n 为时间窗口内交付数量增益因子, S_l 表示节点近期第 l 次失败交付的任务价值, ε 为信誉减益控制因子, x 为近期失败次数。根据式 (12), 信誉度增益函数变化曲线和信誉度减益函数变化曲线如图 3、图 4 所示。

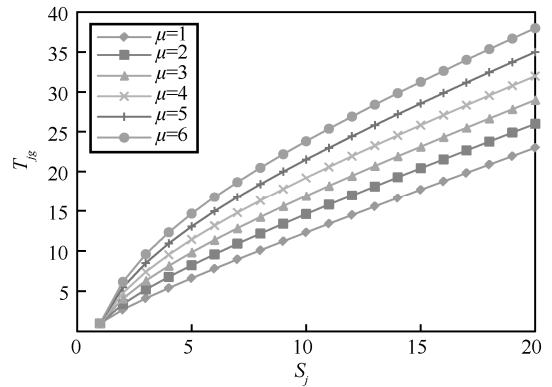


图3 信誉度增益

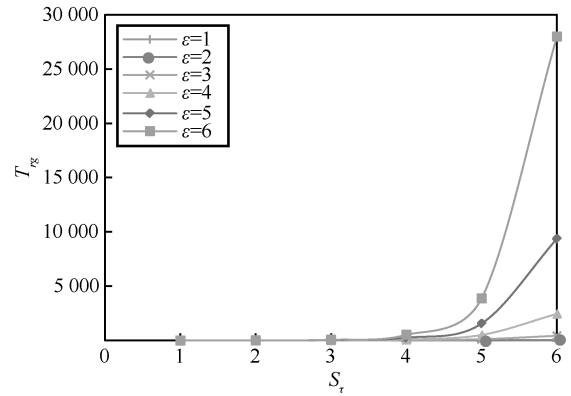


图4 信誉度减益

由图 3、图 4 可知,节点在维持自身良好的网络行为时,其信誉度将以基准分为限,实现按成功次数稳步增长。节点在短时间内多次出现不良行为时,节点的信誉度将快速降低。即使信誉度高于 1 000,任务价值为 0.1,如果有连续 6 次失败,其信誉度将崩塌至 0。由此,高信誉节点更倾向于接收高任务价值的需求,将低价值任务迁移给低信誉节点,这样的行为有助于抑制网络中的搭便车行为。



4.2 边缘计算协同服务满意度评价模型

一般在具体实施过程中, 盟主节点因其承载着协同服务系统中繁复的路由协作业务和策略制定, 故通常为协同服务系统中协作服务能力和链路质量最优的节点。超级节点需要较高的公信力来支撑其完成协同服务系统中信誉评判的任务, 通常为协同服务系统中信誉度最高的节点。

边缘计算协同服务刚初始化, 加入的节点数量较少时, 盟主节点的业务和超级节点的业务可合并在一个节点中完成(通常由基站承载)。当盟主节点的服务质量开始降低或负载过高时, 盟主节点将选择除自身外边缘计算协同服务中信誉度最高的节点成为超级节点, 并将其信誉评判和管理服务迁移至超级节点, 同时以泛洪形式通知系统内节点。

假设边缘计算协同服务存在 m 个节点, 则在第 k 个时间周期 t_k 时, 盟主节点对系统的满意度表征函数 Sat 为:

$$\text{Sat}(t_k) = \frac{1}{m} \sum_{i=1}^m \frac{S_i}{S_i + F_i} \frac{T_i}{T_{\max}} \quad (13)$$

其中, S_i 为边缘计算协同服务中协同节点 i 任务成功交付次数, F_i 为协同节点 i 任务失败交付次数。 T_i 表示协同节点 i 的信誉度, $T_{\max}$ 是盟主节点所在的协同服务系统中协同节点拥有的最大信誉度。

当 Sat 低于阈值 o 时, 盟主节点可选择引导当前的协同服务系统进行分裂或解散当前边缘计算协同服务, 其具体步骤如下。

(1) 盟主节点向超级节点请求系统协同节点近期交付成功及失败记录, 并计算满意度 Sat。

(2) 如果盟主节点负载评价 $Q < \delta$ ($0 < \delta < 1$), 则进行当前边缘计算协同服务分裂示警。

(3) 如果盟主节点负载评价 $Q < \delta'$ ($\delta < \delta' < 1$), 且满意度 $\text{Sat} < o$ 时, 则盟主将引导当前边缘计算协同服务进行分裂。查看当前边缘计算协同服务中常用备选协作节点数量 z , 如果 $z \leq 0$, 则放弃引导, 并进入 (4) 等待重询。如果 $z \geq 1$, 则进入 (5)。

(4) 等待下一个采样周期 t_{k+1} , 如果在此期间 $Q \leq 0$, 且无有效协作节点进行任务迁移, 则放弃引导解散当前边缘计算协同服务, 否则跳转 (1) 计算满意度 Sat。

(5) 盟主节点将所有备选协作节点视为新的边缘计算协同服务备选盟主节点, 通过计算备选盟主节点到系统内所有节点的服务能力的平均值, 进行排序, 选择最优节点正式成为新的边缘计算协同服务盟主节点, 并在系统内公布。

(6) 系统中所有节点计算其到原盟主节点和新盟主节点的服务能力, 按自身意愿, 选择要加入的边缘计算协同服务或观望。

5 仿真实验分析

5.1 仿真环境

(1) 实验中假设如下条件成立

- 在边缘计算协同网络结构组建时, 边缘网络可能已存在相互信任的小规模协同服务节点集合, 实验中假设请求与提供过协同服务的节点相互之间的信任度为 1, 这种原始的信任关系将随着节点在网络中的身份、行为、交互特征改变而动态变化。
- 当服务节点需要选取服务盟员构建边缘计算协同服务时, 首先从其可信的边缘计算协同服务中选取, 以减少计算开销, 节点可以动态无约束地加入和离开网络。

(2) 网络仿真环境

- 基于 Router View 公开数据集, 构建边缘计算系统服务簇; Router View 公开数据集节点数为 6 474 个, 连接边数为 13 895 条, 平均度为 4.292 6, 基尼系数为 0.608, 相对边缘分布熵为 0.854, 聚类系数为 0.959%。
- 从 Router View 公开数据集中随机选取 1 000 个网络节点。为模拟真实边缘环境, 设其骨干路由网络由 1 个中心岸基基站和 5 个中继基站 (router node, RN) 组成, 1 个

源服务发起节点, 节点间自由交互。

- 每个基站路由构建的边缘计算协同服务簇节点规模在 30~120 个, 由数据源边缘服务器 (协同服务发起者)、热心边缘服务器 (本文称为超级节点) 组建的服务簇节点规模数在 20~40 个。当边缘计算协同服务簇节点规模小于设定的下界时进行合并操作, 当簇节点规模大于上界时进行分裂操作。
- 边缘计算可信协同服务系统由数据源边缘服务器 (edge server, ES)、中心基站 (router center, RC)、中继基站、超级节点和边缘节点组成。
- 节点通信制式符合 IEEE 802.11 协议簇标准。

(3) 系统仿真环境

服务器: Intel Xeon E5, 4 核 16 线程, 2.8 GHz, RAM 4 GB, Windows Server 2010, 虚拟机带宽 1 Mbit/s。

客户机: Intel Core i5, 2 核 4 线程, 2.7 GHz, RAM 8 GB, Windows7 系统。

服务端: JDK1.8+ApacheTomcat 8.0, MySQL 5.7。

客户端: Chrome 浏览器。

编程语言: Java、JavaScript。

开发工具: Eclipse Oxygen3A。

5.2 边缘计算协同服务系统探究实验

为分析边缘计算协同服务系统的性能特征, 将本文提出的 K 维贪心 (K -Greedy) 优化算法与随机游走 (random walk) 算法^[34], 以访问量、流量、负载、信誉度、任务接收率和协同成功率等为评价指标, 对边缘计算协同服务进行对比仿真分析。

设数据源边缘服务器集群 (ES) 存在 100 个待迁移计算任务, 每个迁移计算任务量 $500 \text{ MB} \leq M \leq 1024 \text{ MB}$, 对目标计算任务所在边缘服务器 ES 进行浪涌式服务请求测试。用户请求频率 $f \leq 4 \text{ 次/s}$, 每次请求 5 个连续的计算任务, 实验测试时间为 1 h, 访问量、流量采样周期为 10 s, 信誉度、负载、任务接收率和协同成功率采样周期为 5 s。

从 Router View 公开数据集中随机选取 1 000 个网络节点。为模拟真实边缘环境, 设其骨干路由网络由 1 个中心岸基基站和 5 个中继基站组成, 节点间自由交互, 基站簇内形成 11 个超级节点, 构成 11 个边缘计算服务池。所构建的边缘计算服务池平均规模在 32.308~39.545, 网络平均时延为 21.81 ms。系统拓扑结构如图 5 所示。

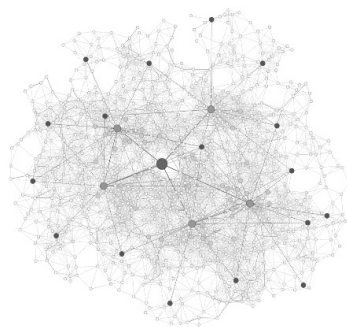


图 5 系统拓扑结构

5.2.1 边缘计算协同服务系统分发效率评价

根据系统流量、接收率和成功率分析在浪涌访问下, 边缘计算协同服务系统分发效率。其中, 边缘计算协同服务系统流量如图 6 所示。

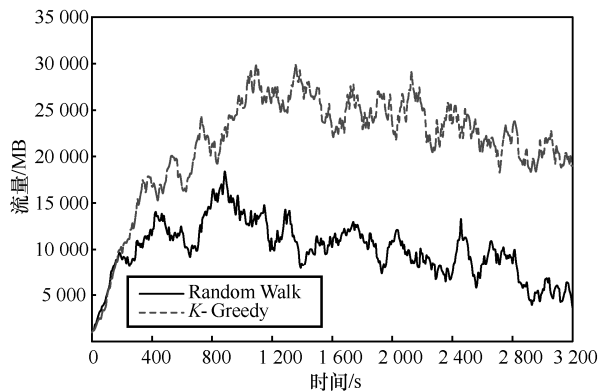


图 6 边缘计算协同服务系统流量

由图 6 可知, 在浪涌访问测试下, K -Greedy 和 Random Walk 构建的边缘计算协同服务系统峰值流量分别为 30 159 MB、17 182 MB, 平均流量分别为 20 891 MB、10 491 MB。本文的 K -Greedy 与 Random Walk 相比, 峰值流量提高了 75.52%, 平均流量提高了 99.13%。



边缘计算协同服务系统任务接收率和交付成功率分别如图 7 和图 8 所示。

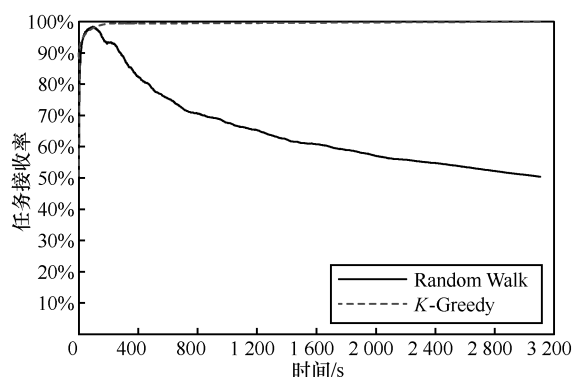


图 7 边缘计算协同服务系统任务接收率

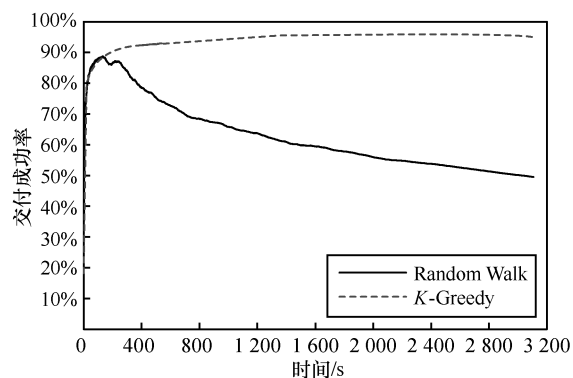


图 8 边缘计算协同服务系统任务交付成功率

由图 7、图 8 可知,在访问测试结束时, K -Greedy 和 Random Walk 构建的边缘计算协同服务任务接收率分别为 99%和 51%,任务交付成功率均分别为 94%、49%。测试期间, K -Greedy 和 Random Walk 构建的边缘计算协同服务平均任务接收率为 99%和 61%,平均任务交付成功率均分别为 94%、58%。本文的 K -Greedy 与 Random Walk 相比,协同任务成功率提高近 48%,任务交付成功率提高近 36%。

综上所述,本文的 K -Greedy 与 Random Walk 相比,具有更优的任务分发效率。

5.2.2 边缘计算协同服务系统负载均衡性评价

根据协同服务盟员、基站、超级节点的带宽占用负载分析在浪涌访问下,边缘计算协同服务系统的均衡性。其中,边缘计算协同服务盟员平均负载率如图 9 所示。

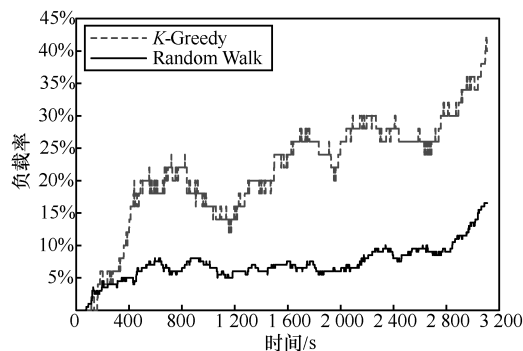


图 9 边缘计算协同服务盟员平均负载率

由图 9 可知,在浪涌访问测试下, K -Greedy 和 Random Walk 构建的边缘计算协同服务盟员峰值负载分别为 42%和 16%,平均负载分别为 25%和 5%。本文的 K -Greedy 与 Random Walk 相比,盟员峰值负载提高了 26%,平均负载提高了 20%。

边缘计算协同服务系统超级节点和基站带宽占用负载率分别如图 10、图 11 所示。

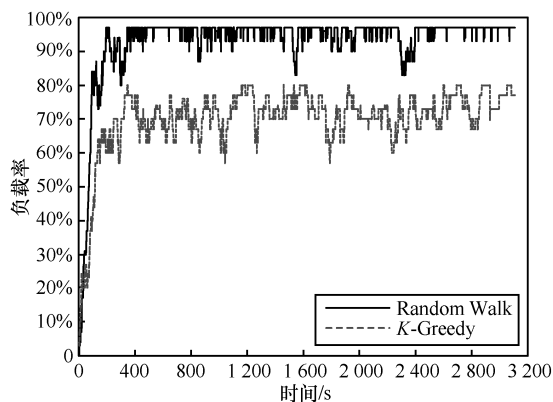


图 10 超级节点带宽占用负载率

由图 10、图 11 可知,在浪涌访问测试下, K -Greedy 和 Random Walk 构建的边缘计算协同服务达到稳定时, K -Greedy 和 Random Walk 的超级节点最大负载率分别为 80%和 96%,最低负载率分别为 57%和 83%,平均负载率分别为 69%和 94%。 K -Greedy 和 Random Walk 的基站节点最大负载率分别为 77%和 96%,最低负载率分别为 69%和 92%,平均负载率分别为 73%和 94%。本文的 K -Greedy 与 Random Walk 相比,超级节点的负载率降低 25%,基站平均负载率降低 21%。

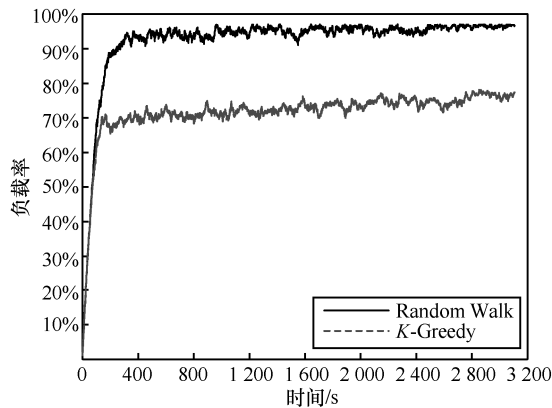


图 11 基站带宽占用负载率

综上所述, 本文的 K -Greedy 与 Random Walk 相比, 基站、超级节点的协同任务被更好地分发给盟员节点, 基站、超级节点的负载更均衡, 协同服务盟员的参与度更高。

5.2.3 边缘计算协同服务协同用户服务质量评价

以用户对协同服务系统的信誉度、链路时延、分组丢失率和迁移传输速率分析在浪涌访问下, 边缘计算协同服务系统的服务质量。其中, 边缘计算协同服务盟主的信誉度如图 12 所示。

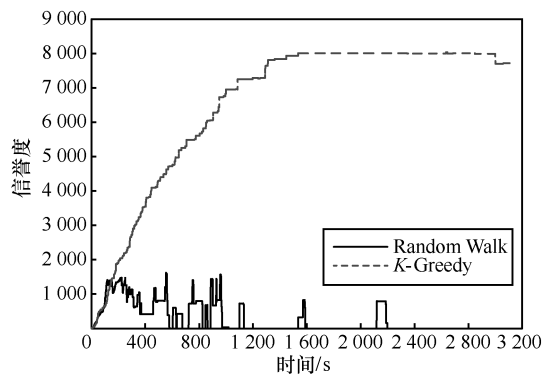


图 12 边缘计算协同服务盟主的信誉度

由图 12 可知, 在浪涌访问测试下, K -Greedy 和构建的边缘计算协同服务盟主节点的信誉度能够达到 8 000, 而 Random Walk 为 0。这是因为本文的 K -Greedy 与 Random Walk 相比, 能够更好地分发任务, 防止了热区拥塞。

由图 13~图 15 可知, 当测试结束时, K -Greedy 盟员节点平均时延由 10 ms 增加至 39 ms, 链路平

均分组丢失率由 0 增加至 26%, 边缘节点迁移链路平均速率为 0.68 Mbit/s, 并呈线性增长态势。而 Random Walk 盟员节点平均时延由 10 ms 增加至 20 ms 后不断下降至 10 ms, 分组丢失率

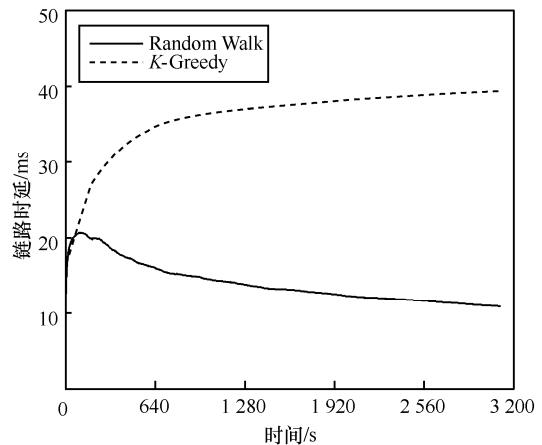


图 13 协同服务盟员平均链路时延

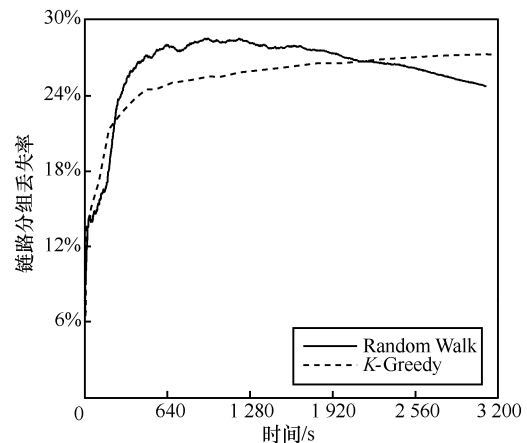


图 14 协同服务盟员链路平均分组丢失率

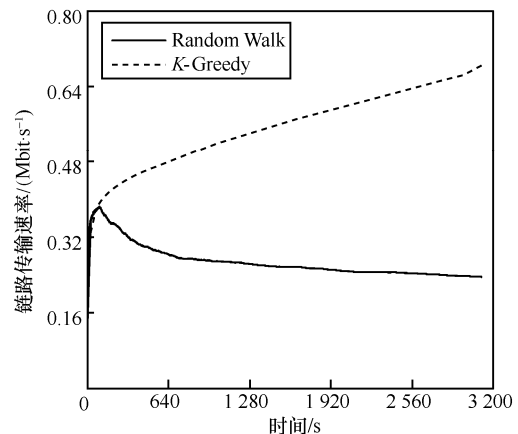


图 15 边缘节点迁移链路平均传输速率



由 0 增加至 28% 后不断下降至 24%，边缘节点迁移链路平均速率为 0.23 Mbit/s，并趋于平稳。由于任务聚集，本文的 *K-Greedy* 与 *Random Walk* 相比，链路时延增加 19 ms，但分组丢失率下降 2%，传输速率增加 0.45 Mbit/s。

综上所述，本文的 *K-Greedy* 与 *Random Walk* 相比，在协同服务信誉度、时延、分组丢失率和传输速率方面，*K-Greedy* 更优。

6 结束语

边缘计算协同服务任务调度是边缘计算系统管理的重要研究内容。针对边缘计算在面临浪涌式访问请求下，如何有效避免访问拥塞、负载超限、分发效率过低的问题，本文提出多约束可信协同任务调度机制。该机制根据任务需求约束选择协同节点并结合用户个性化的服务质量需求，通过贪心算法优化任务分配。仿真实验结果表明，本文所提方法能够有效提高边缘计算协同服务任务分发效率，实现协同共享、负载均衡、服务满意的目的，提高了边缘计算网络的服务质量。

参考文献:

- [1] 中华人民共和国工业和信息化部. 工业和信息化部向基础电信运营企业发放 5G 系统试验频率使用许可[EB]. 2018. Ministry of Industry and Information Technology of People's Republic of China. Ministry of industry and information technology issues 5G system test frequency use license to basic telecom operators[EB]. 2018.
- [2] 施巍松, 张星洲, 王一帆, 等. 边缘计算: 现状与展望[J]. 计算机研究与发展, 2019, 56(1): 69-89. SHI W S, ZHANG X Z, WANG Y F, et al. Edge computing: state-of-the-art and future directions[J]. Journal of Computer Research and Development, 2019, 56(1): 69-89.
- [3] 施巍松, 孙辉, 曹杰, 等. 边缘计算: 万物互联时代新型计算模型[J]. 计算机研究与发展, 2017, 54(5): 907-924. SHI W S, SUN H, CAO J, et al. Edge computing-an emerging computing model for the Internet of everything era[J]. Journal of Computer Research and Development, 2017, 54(5): 907-924.
- [4] 谢人超, 廉晓飞, 贾庆民, 等. 移动边缘计算卸载技术综述[J]. 通信学报, 2018, 39(11): 138-155. XIE R C, LIAN X F, JIA Q M, et al. Survey on computation offloading in mobile edge computing[J]. Journal on Communications, 2018, 39(11): 138-155.
- [5] ZHANG W, WEN Y. Energy-efficient task execution for application as a general topology in mobile cloud computing[J]. IEEE Transaction on Cloud Computing, 2018, 6(3): 708-719.
- [6] 陈黄科, 祝江汉, 朱晓敏, 等. 云计算中资源延迟感知的实时任务调度方法[J]. 计算机研究与发展, 2017, 54(2): 446-456. CHEN H K, ZHU J H, ZHU X M, et al. Resource-delay-aware scheduling for real-time tasks in clouds[J]. Journal of Computer Research and Development, 2017, 54(2): 446-456.
- [7] 周业茂, 李忠金, 葛季栋, 等. 移动云计算中基于延时传输的多目标 workflow 调度[J]. 软件学报, 2018, 29(11): 3306-3325. ZHOU YM, LI Z J, GE J D, et al. Multi-objective workflow scheduling based on delay transmission in mobile cloud computing[J]. Journal of Software, 2018, 29(11): 3306-3325.
- [8] ZHANG H, GUO J, YANG L, 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), May 4, 2017, Atlanta, GA, USA. Piscataway: IEEE Press, 2017.
- [9] LI C L, TANG J, TANG H L, et al. Collaborative cache allocation and task scheduling for data-intensive applications in edge computing environment[J]. Future Generation Computer Systems, 2019(95): 249-264.
- [10] 胡海洋, 刘润华, 胡华. 移动云计算环境下任务调度的多目标优化方法[J]. 计算机研究与发展, 2017, 54(9): 1909-1919. HU H Y, LIU R H, HU H. Multi-objective optimization for task scheduling in mobile cloud computing[J]. Journal of Computer Research and Development, 2017, 54(9): 1909-1919.
- [11] TUYEN X, TRAN, ABOLFAZL H, et al. Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges[J]. IEEE Communications Magazine, 2017, 55(4): 54-61.
- [12] CAO X, WANG F, XU J, et al. Joint computation and communication cooperation for energy-efficient mobile edge computing[J]. IEEE Internet of Things Journal, 2018. doi: 10.1109/JIOT.2018.2875246.
- [13] HU X, WONG K K, KUN Y. Wireless powered cooperation-assisted mobile edge computing[J]. IEEE Transactions on Wireless Communications, 2018, 17(4): 2375-2388.
- [14] ZHANG W W, WEN Y G, WU D O. Collaborative task execution in mobile cloud computing under a stochastic wireless channel[J]. IEEE Transactions on Wireless Communications, 2015, 14(1): 81-93.
- [15] GE X, TU S, MAO G, et al. 5G ultra-dense cellular networks[J].

- IEEE Wireless Communications, 2016, 23(1): 72-79.
- [16] GE X, CHENG H, GUIZANI M. 5G wireless backhaul networks: challenges and research advances[J]. IEEE Network, 2014, 28(6): 6-11.
- [17] SUN Y X, ZHOU S, XU J. EMM: energy-aware mobility management for mobile edge computing in ultra dense networks[J]. IEEE Journal on Selected Areas in Communications, 2017, 35(11): 2637-2646.
- [18] GUO S, LIU J, YANG Y, et al. Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing[J]. 2019. doi:10.1109/TMC.2018.2831230.
- [19] 王玢, 吴雅婧, 阳小龙, 等. 关联性驱动的大数据处理任务调度方案[J]. 软件学报, 2017, 28(12): 3385-3398.
- WANG B, WU Y J, YANG X L, et al. Dependency-driven task scheduling scheme of big data processing[J]. Journal of Software, 2017, 28(12): 3385-3398.
- [20] WANG Q Y, GUO S T, LIU J D, et al. Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing[J]. Sustainable Computing: Informatics and Systems, 2019, 21: 154-164.
- [21] 胡海洋, 姬朝配, 胡华, 等. 基于协作相容性的工作流任务分配优化方法[J]. 计算机研究与发展, 2017, 54(4): 872-885.
- HU H Y, JI C P, HU H, et al. Method for optimizing task allocation in workflow system based on cooperative compatibility[J]. Journal of Computer Research and Development, 2017, 54(4): 872-885.
- [22] 柴云鹏, 顾雷, 李三立. 移动网格中负载自平衡的集群式边缘流媒体服务器[J]. 计算机研究与发展, 2007, 44(12): 2136-2142.
- CHAI Y P, GU L, LI S L. Cluster-based edge streaming server with adaptive load balance in mobile grid[J]. Journal of Computer Research and Development, 2007, 44(12): 2136-2142.
- [23] LIU T T, LI J, FENG S, et al. Distributed resource allocation in caching-enabled heterogeneous cellular networks based on matching theory[J]. CCF Transactions on Networking, 2018.
- [24] 房俊华, 王晓桐, 张蓉, 等. 分布式数据流上的高性能分发策略[J]. 软件学报, 2017, 28(3): 563-578.
- FANG J H, WANG X T, ZHANG R, et al. High-performance data distribution algorithm on distributed stream systems[J]. Journal of Software, 2017, 28(3): 563-578.
- [25] TONG L, GAO W. Application-aware traffic scheduling for workload offloading in mobile clouds[C]//The 35th Annual IEEE International Conference on Computer Communications, April 16, 2016, San Francisco, CA, USA. Piscataway: IEEE Press, 2016: 1-9.
- [26] ELGAZZAR K, MARTIN P, HASSANEIN H. Cloud-assisted computation offloading to support mobile services[J]. IEEE Transaction on Cloud Computing, 2016, 4(3): 279-292.
- [27] CHANG Z, GONG J, ZHOU Z, et al. Resource allocation and data offloading for energy efficiency in wireless power transfer enabled collaborative mobile clouds[C]//2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), April 26, 2015, Hong Kong, China. Piscataway: IEEE Press, 2015: 336-341.
- [28] CHEN X. Decentralized computation offloading game for mobile cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2015, 26(4): 974-983.
- [29] LIN X, WANG Y Z, XIE Q, et al. Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment[J]. IEEE Transactions on Services Computing, 2015, 8(2): 175-186.
- [30] DENG S, HUANG L, TAHERI J, et al. Computation offloading for service workflow in mobile cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2015, 26(12): 3317-3329.
- [31] 于博文, 蒲凌君, 谢玉婷, 等. 移动边缘计算任务卸载和基站关联协同决策问题研究[J]. 计算机研究与发展, 2018, 55(3): 537-550.
- YU B W, PU L J, XIE Y T, et al. Joint task offloading and base station association in mobile edge computing[J]. Journal of Computer Research and Development, 2018, 55(3): 537-550.
- [32] 邓晓衡, 关培源, 万志文, 等. 基于综合信任的边缘计算资源协同研究[J]. 计算机研究与发展, 2018, 55(3): 449-477.
- DENG X H, GUAN P Y, WAN Z W, et al. Integrated trust based resource cooperation in edge computing[J]. Journal of Computer Research and Development, 2018, 55(3): 449-477.
- [33] FANG X, YANG D J, GUNDECHA P, et al. Multi-constrained anypath routing in wireless mesh networks[C]//2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), June 21, 2010, Boston, MA, USA. Piscataway: IEEE Press, 2010: 475-483.
- [34] 张以文, 汪开斌, 严远亭, 等. 基于覆盖随机游走算法的服务质量预测[J]. 计算机学报, 2018, 41(12): 2756-2768.
- ZHANG Y W, WANG K B, YAN Y T, et al. Service quality prediction based on covering random walk algorithm[J]. Journal of Computer Science, 2018, 41(12): 2756-2768.
- [35] ZHONG Y, QUEK T, GE X. Heterogeneous cellular networks with spatio-temporal traffic: delay analysis and scheduling[J]. IEEE Journal Selected Areas Communications, 2017, 35(6): 1373-1386.
- [36] GE X, YANG B, YE J, MAO G, et al. Spatial spectrum and energy efficiency of random cellular networks[J]. IEEE Transactions on Communications, 2015, 63(3): 1019-1030.



- [37] ZHONG Y, GE X, YANG H H, et al. Traffic matching in 5G ultra-dense networks[J]. IEEE Communications Magazine, 2018, 56(8): 100-105.
- [38] 程日涛, 张海涛, 王乐. 5G 无线网部署策略[J]. 电信科学, 2018, 34(S1): 1-8.
CHENG R T, ZHANG H T, WANG L. Deployment strategy of 5G wireless network[J]. Telecommunications Science, 2018, 34(S1): 1-8.
- [39] 王祖阳, 杨传祥, 张进, 等. 5G 无线网技术特征及部署应对策略分析[J]. 电信科学, 2018, 34(S1): 9-16.
WANG Z Y, YANG C X, ZHANG J, et al. Analysis on technology characteristics and deployment strategies of 5G wireless network[J]. Telecommunications Science, 2018, 34(S1): 9-16.

[作者简介]



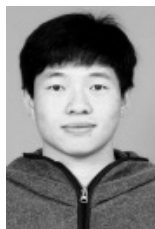
乐光学（1963—），男，博士，嘉兴学院数理与信息工程学院教授，主要研究方向为多云融合与协同服务、无线 Mesh 网络与移动云计算。



戴亚盛（1993—），男，江西理工大学理学院硕士生，主要研究方向为边缘计算协同服务。



杨晓慧（1996—），女，江西理工大学理学院硕士生，主要研究方向为移动云计算。



朱友康（1993—），男，江西理工大学理学院硕士生，主要研究方向为边缘计算和计算迁移。



游真旭（1993—），女，江西理工大学理学院硕士生，主要研究方向为数据挖掘和推荐算法。



刘建生（1959—），男，江西理工大学理学院教授，主要研究方向为深度学习。