

一种基于边缘计算的传感云低耦合方法

梁玉珠¹ 梅雅欣¹ 杨毅¹ 马樱² 贾维嘉³ 王田¹

¹(华侨大学计算机科学与技术学院 福建厦门 361021)
²(数据挖掘与智能推荐福建省高校重点实验室(厦门理工学院) 福建厦门 361024)
³(智慧城市物联网国家重点实验室(澳门大学) 澳门 999078)
(cs_yuzhuliang@163.com)

A Low-Coupling Method in Sensor-Cloud Systems Based on Edge Computing

Liang Yuzhu¹, Mei Yaxin¹, Yang Yi¹, Ma Ying², Jia Weijia³, and Wang Tian¹

¹(College of Computer Science and Technology, Huaqiao University, Xiamen, Fujian 361021)
²(Key Laboratory of Data Mining and Intelligent Recommendation of Fujian Province University (Xiamen University of Technology), Xiamen, Fujian 361024)
³(State Key Laboratory of Internet of Things for Smart City (University of Macau), Macau 999078)

Abstract The rapid development of the IoT and cloud computing has spawned a new network structure-sensor cloud. Sensor cloud is the combination of IoT and cloud computing. Physical sensor nodes in the IoT can be virtualized into multiple nodes through the sensor cloud platform to provide services to users. However, when one sensor node receives multiple service commands at the same time, some service conflicts occur, which named coupling problems. This coupling problem can lead to the failure of services and compromise system security. In order to solve this problem, this paper proposes an extended KM (Kuhn-Munkres) algorithm based on edge computing. Edge computing is an emerging computational paradigm, increasingly utilized in IoT applications, particularly those that cannot be served efficiently using cloud computing due to limitations such as latency. The edge computing platform acts as a middleware platform and provides the scheduling method. Firstly, the edge computing layer merges the similar commands to reduce the downward transmission commands. Secondly, the buffered data in the edge computing layer is scheduled. Finally, the extended KM algorithm is used to achieve the maximum matching of each round. The theoretical analysis and experimental results show that the proposed method can improve the utilization of resources, reduce the calculation cost, and solve the coupling problem in a minimum time.

Key words sensor-cloud; Internet of things; edge computing; low-coupling; Kuhn-Munkres algorithm

收稿日期:2019-08-23;修回日期:2020-01-15
基金项目:数据挖掘与智能推荐福建省高校重点实验室开放基金项目(DM201902);福建省网络计算与智能信息处理重点实验室开放课题(FZDX201908);福建省社会科学规划基金项目(FJ2018B038);福建省自然科学基金项目(2018J01092);华侨大学研究生科研创新基金项目(17013083005)
This work was supported by the Open Fund of Key Laboratory of Data Mining and Intelligent Recommendation of Fujian Province University (DM201902), the Open Foundation of Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing (FZDX201908), the General Projects of Social Sciences in Fujian Province (FJ2018B038), the Natural Science Foundation of Fujian Province of China (2018J01092), and the Subsidized Project for Postgraduates' Innovative Fund in Scientific Research of Huaqiao University (17013083005).
通信作者:王田(cs_tianwang@163.com)

摘要 随着物联网和云计算的快速发展,衍生了一种新的网络结构——传感云.传感云是物联网和云计算结合的产物.物联网中的物理节点可以通过传感云平台虚拟成多个节点,为用户提供服务.然而,当底层物理传感器节点同时接收多个服务命令时,会出现一些服务冲突,即耦合问题.这种耦合问题可能导致服务的失败,并危及系统安全.为了解决这个问题,提出了一种基于边缘计算思想的扩展 KM (Kuhn-Munkres) 算法.边缘计算是一种新兴的计算模式,在物联网中得到越来越广泛的应用,特别是那些由于延迟等限制而无法有效利用云计算的应用.边缘计算作为云和物联网层的中间平台,可以提供调度方法.首先,边缘层对重复请求命令进行合并,减少向下传输的命令数量;其次,优先调用边缘层数据缓存队列里的数据;最后利用改进 KM 算法实现每一轮的最大匹配.理论分析和实验结果表明,提出的方法可以提高资源利用率,减小计算成本,接近最短的时间解决耦合问题.

关键词 传感云;物联网;边缘计算;低耦合;Kuhn-Munkres 算法

中图法分类号 TP393

物联网有着极其广泛的应用场景,包括智慧城市、工业物联网、健康物联网、智能家居以及车联网等^[1].目前随着传感器及带宽成本的大幅下滑以及云计算技术发展和通信标准的落地,物联网已从最初的导入期发展至现在的成长期初期,国外第三方咨询机构普遍预期 2016—2020 年将保持 25%~30% 的复合增速,其中 Growth Enabler and Markets 预期全球物联网市场在 2020 年达到 4 570 亿美元,复合增速 28.5%.2016 年我国物联网整体产业规模达到了 9 300 亿元人民币,同比增长 24%,预计到 2020 年,市场规模将会进一步扩大到 1.8 万亿元.据预测,至 2020 年,物联网的前五大下游应用场景分别为智慧城市、工业物联网、健康物联网、智能家居以及车联网^[2].

近年来,伴随云计算技术的快速发展,研究者提出通过云端控制传感器网络进行信息采集^[3],利用云平台进行信息处理和存储^[4],为不同类型的应用提供开放、灵活、可配置的服务平台.用户不再需要购置或部署传感网,无需关注节点的位置,可通过使用或控制云服务器提供的虚拟节点来满足服务请求^[5].虚拟节点是由物理传感器节点组成的用户可视节点,随着用户需求而创建,服务停止而销毁.当服务复杂度较高时,不同功能的虚拟节点组成虚拟节点组协作完成服务.单个物理节点可以参与多个虚拟节点的组建,单个虚拟节点也可以参与多个虚拟节点组的组建,这种机制极大提高了物理节点的利用率,为传感云服务的高效率和多样化提供了便利.

用户通过传感云系统,足不出户即可获取自己所需的数据.目前已有的传感云系统,如商业化的 Xively、开源的 Nimbits 以及国内的 YeeLink 和阿

里云 IoT^[6-7],他们提供的服务不尽相同,用户可以选择把自己的传感器收集的信息上传到系统,也可以选择把自己的传感器共享给其他用户使用.但是传感云系统同样面临着物联网和云计算自身存在的安全问题,甚至衍生出新的安全问题,如传感云的耦合问题.当多用户同时对传感云系统中的同一个物理节点进行请求时,因为每个物理传感器只有一个通信信道,同一时间不能接收多个消息,那样会出现冲突.例如对于物联网中的移动节点,一个用户通过云平台接口发出的请求是让它向左收集收据,而另一个用户对其发出的请求是向右收集数据,会出现耦合冲突.其次,多用户获取服务时共用同一个虚拟传感节点,从而共用其所代表的所有物理传感器节点,导致用户对服务的反馈命令出现耦合.

耦合问题会带来极大的安全隐患,耦合问题会造成用户等待服务时间长,系统服务质量低下;物理节点调度严重不均,降低系统的使用寿命;甚至会导致系统死锁,导致系统无法正常运行.因此,传感云系统中耦合问题不可忽视且亟待解决.一些研究者提出利用云层来管理这些传感器节点,这种方法存在的主要问题是云层离底层较远,这种方式往往是不及时的.还有一些研究者提出利用传统的调度算法并结合启发式算法提高资源利用率,这种方式简单地把问题转化为调度问题,实际上不能取得很好的效果.此外,还有一些研究者提出结合边缘计算这一新型计算模式来研究耦合问题,但没有给耦合问题下一个形式化的定义,且大都仅仅提出一个框架并没有给出具体的方法来解决这个问题.基于此,本研究首先对边缘计算这一新型计算模式进行了探索.并在边缘层设计了 2 个缓存队列,利用边缘存储

对问题做预处理,简化问题,并改进了 KM(Kuhn-Munkres)算法,实现多次匹配并达到最佳匹配,以最短的时间解决耦合问题。

本文的主要贡献包括 2 个方面:

1) 在传感云中,用户可以通过虚拟传感器来使用传感器,当多个用户或攻击者发送大量请求同时控制一个物理传感器时,可能导致耦合问题。据已有文献调研所知,本文形式化地描述了传感云中的耦合问题;

2) 创新地将边缘计算引入到系统中来解决耦合问题。在边缘层的支持下,系统的响应延迟大大缩短。此外,设计了一个有效的算法,以达到用户请求和资源节点之间的最大匹配,从而实现资源的最佳利用。理论和实验结果验证了该方法的有效性。

1 相关工作

1.1 基于云计算的耦合资源管理方法

Sharma 等人^[8]指出,许多调度方法是为了获得最大的吞吐量和就绪队列中最小等待的时间。他们提出一种基于遗传算法的最优调度方案,基于自适应函数,达到最优调度的目的。刘炳涛等人^[9]提出了一种基于概率控制的新型循环队列调度方法,相比于传统的 FIFO(first in first out)调度,有更好的效果。Li 等人^[10]首先构建了任务模型和资源模型,并分析任务的偏好,然后用模糊聚类算法对资源进行分类。最后计算资源期望并将任务分配给不同的资源集群,从而减少资源选择的复杂性,进而减少任务的等待时间并提高资源利用率。Tang 等人^[11]基于云环境提出了一种新方案。首先计算所有任务的初始调度顺序,并根据异构最早完成时间算法获取整个完成时间和限制时间,该算法可以在低电压和低频率下将任务分配到空闲时隙中。Kayvanfar 等人^[12]提出了一个基于基础设施即服务(infrastructure as a service, IaaS)的科学工作流的资源配置和调度策略,并提出基于元启发式优化技术算法,利用粒子群优化,在满足条件的基础上最小化执行时间。

1.2 基于二分图的耦合资源调度方法

Chopra 等人^[13]通过在匈牙利算法中使用广泛的聚合算子来开发新的赋值算法。基于匈牙利算法,使用有序加权平均距离(ordered weighted average distance, OWAD)算子和诱导 OWAD 算子的新过程。这种方法的主要优点是可以在最小和最大值之间提供一个参数化的聚合算子族。陈晓旭等人^[14]设计

了一种迭代技术,迭代匈牙利法(iterative Hungarian method, IHM)。数值结果表明:所提出的方法可以提供具有多项式复杂度的近似最优性能。Zhang 等人^[15]提出了改进的 Max-min 任务调度算法,均衡调度最长的资源和最短的资源。Zhu 等人^[16]提出一个 Kuhn-Munkres 并行遗传算法来解决集覆盖问题,使用分而治之的降维策略,因此采用 Kuhn-Munkres 并行遗传算法来拼接在每个分区中获得可行的解决方案以改善搜索效率。

1.3 基于边缘计算的耦合资源管理方法

基于边缘计算模式,一些研究者提出了相关的解决方案。在物联网系统中,所有原始事件数据都是从传感器上收集的,并且仅在云中处理,导致不必要的冗余数据和网络开销。事件和动作之间存在更紧密的耦合^[17]。通过基于边缘计算的局部数据处理技术,可以降低成本,提高系统的整体响应性能。De 等人^[18]认为物联网系统的发展制约于物理过程和数字世界的耦合,对下一次工业革命的影响至关重要。考虑了边缘计算范式,并扩展了一个标准兼容的通信体系结构,以支持基于容器的编排机制。Magurawalage 等人^[19]介绍了一种为连接设备提供计算服务的新设想。它基于未来计算资源将与通信资源耦合的关键概念,用于增强用户体验,以及优化基础设施提供者的资源。此外,这种耦合是基于联合/协作的资源分配算法,通过集成计算和通信服务,在网络中集成硬件来实现的。Wang 等人^[20]提出虚拟节点的耦合是阻碍传感云发展的原因之一^[21]。虚拟节点是由物理传感器节点组成的用户可视节点,随着用户需求而创建,服务停止而销毁^[22-23]。文献^[24-25]提出了一种基于数据缓存队列的扩展 KM 算法,使资源的利用率最大化。Gao 等人^[26]提出了一个贪婪算法,利用 KM 算法求解二部图上的最大匹配问题,从而得到原问题的拟最优解的有效算法^[27]。这些算法包括 2 种离线算法和一种在线算法,其中 2 种离线算法返回原问题的上界和下界^[28-29]。Li 等人^[30]提出将数据块优化放置和任务优化调度相结合,以减少提交任务的计算延迟和响应时间,提高用户的体验。其中在数据块的最佳放置中,数据块的放置不仅考虑了数据块的流行程度,还考虑了存储这些数据块的边缘服务器的数据存储容量和替换率。

上述研究都是资源分配中对经典算法进行的一些改善,采用不同的分配策略,加入不同参数,结合启发式算法或基于边缘计算模式等进行处理。然而在传感云耦合问题上,上述研究提出的方案都无法

有效解决这个问题.其主要原因有:没有足够的信息预测传感云系统的实时需求,结合启发式算法不能很好的解决问题,且增加了时间复杂度.其次,调度最长或者最短的资源可能会造成系统死锁,且这种资源也不一定是最紧急的资源.此外,以上方法最终完成的匹配都是一对一的匹配,而在这个问题上每一次匹配是允许出现一对多的情况,因为同一时间一个用户可以使用多个资源.因此如何对现有的资源管理方法进行改进并结合边缘存储的思想提出复杂度低、效率高的低耦合算法以提升传感云系统的各项指标来扩大其应用范围是一个亟待解决的问题.

2 耦合问题

2.1 问题定义

假设传感云系统中,1个用户可以使用多个资源,而1个资源在同一时刻只能被1个用户使用.如果同一时刻同1个资源被分配给多个用户,这时会产生冲突,称为耦合问题.即需要解决的问题是,假设有用户集 X 和资源集 Y ,给出带权二分图 $G\langle V, E\rangle$, $X\cup Y=V, X\cap Y=\varnothing, M\langle X_i\cup Y_j, E_{ij}\rangle$ 是 G 的一个分配,使 $G^{n+1}=G^n-M$;如此,直到 $G^n=0$;在此过程中,使得 n 最小.假设完成调度的总时间为 T ,目标即最小化 T 值.

2.2 耦合实例

为了让读者更清楚了解耦合问题,本节给出传感云系统中的1个耦合实例.假设传感云系统此时存在5名用户和4类资源,每个用户请求资源的时间片相同,均为 t (单位为s).当5名用户同时发起资源请求时,不同请求对资源的需求不同,如图1所示,用户请求和资源的耦合关系如下:用户1和用户2的耦合资源为 R_1 和 R_2 ,用户2和用户3的耦合资源为 R_3 ,用户3~5的耦合资源为 R_4 .

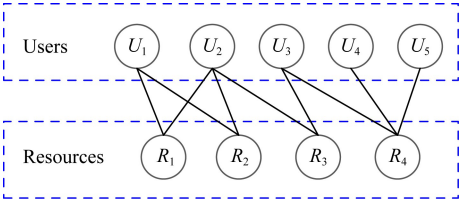


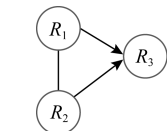
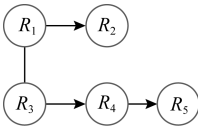
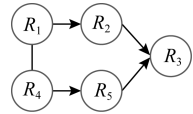

Fig.1 An illustration of the coupling problem
图1 耦合问题实例

表1给出了所有用户请求执行的顺序,及完成所有资源请求消耗的时间.方案 a 是把资源 R_4 的数据实时缓存在边缘层(因为其被请求最多),当用户

发出请求时可以从边缘层直接返回结果.用户2和用户3的耦合资源为 R_3 ,因此用户2和用户3的请求只能串行执行.此时方案 a 先执行用户1和用户3的请求,再执行用户2的请求.方案 b 则先并行执行没有耦合资源的用户1和用户3的请求,然后并行执行用户2和用户4的请求,最后执行用户5的请求.方案 c 同样是先并行执行没有耦合资源的用户1和用户4的请求,但是用户2和用户3的请求有耦合资源 R_3 ,因此用户2和用户3的请求只能串行执行.方案 d 是顺序执行用户的请求.从表1可以看出,方案 a 耗时最短,为 $2t$.

Table 1 Comparisons for Different Methods

表1 不同方法得到结果对比

Schema	Order of Execution	Total Time
a		$2t$
b		$3t$
c		$3t$
d		$5t$

3 算法设计与分析

3.1 算法设计思路

基于边缘计算的耦合管理平台如图2所示.其包括基于边缘计算的耦合管理平台、云层和底层,云层的下方是边缘层.传感器网络处在架构的底层,用于收集数据,将用户需求的数据由各处收集,上传集合到云层中,由云服务进行整合,返回给用户.

虚拟传感器组对于用户来讲实际上是透明的,每个虚拟传感器由一种或多种物理传感器组成.虚拟传感器使得用户可以在不考虑传感器的物理位置的情况下应用传感器.边缘层离底层的传感器较近(底层一些计算能力较强的传感器也可以作为边缘节点),可实时了解底层传感器的状态,可以更快地和底层交互^[27-28].其中,基于边缘层的数据缓存队列是指利用边缘层缓存的数据可以对问题作预处理,

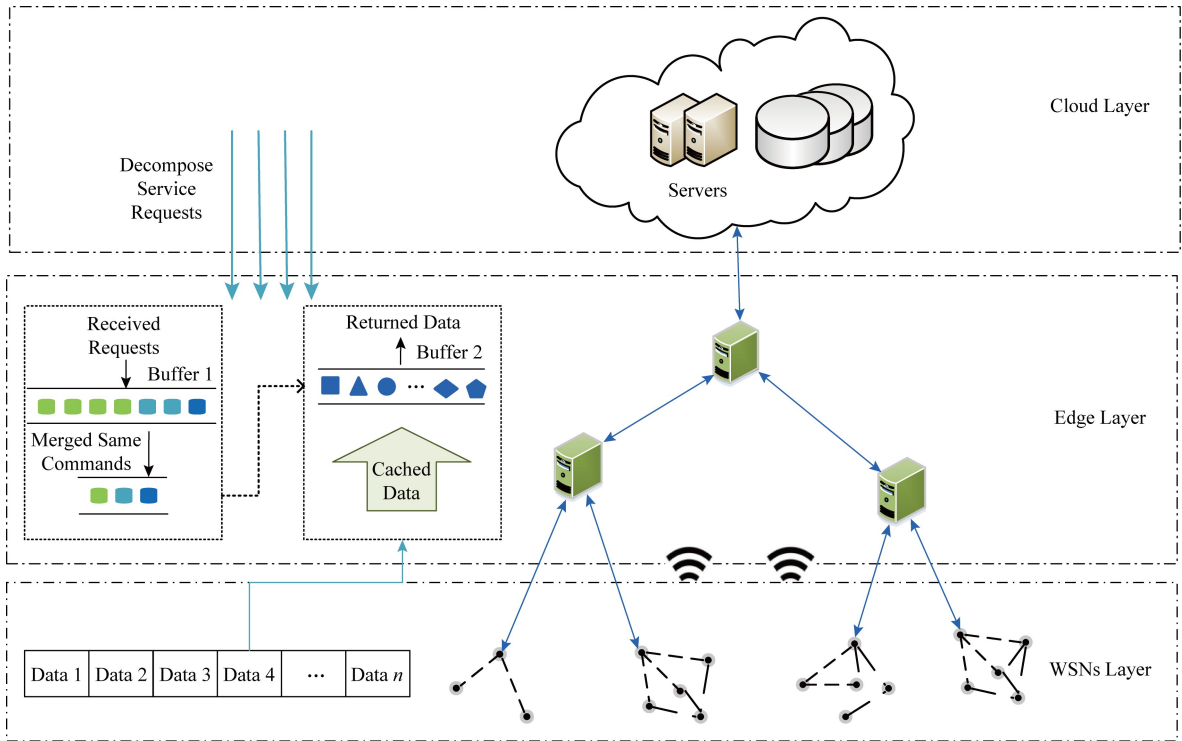


Fig. 2 Edge-based coupling management platform
图 2 基于边缘计算的耦合管理平台

在缓存队列里面的资源可以直接返回结果给用户，降低调用物理节点的次数，减少耦合度，边缘端也可以对云端传送来的多个重复操作命令进行合并，进一步减少耦合度。

3.2 算法设计

首先，通过已有的研究经验，基于时间片思想，为代价矩阵加入权值，形成带权二分图。因此，由解决耦合问题成功转化为解决非均匀服务时间的耦合问题。接着，发现在时间片系统中，绝大多数的处理时间都被浪费到系统的正常运转维护当中，即当用户的占用时间结束时，系统实际上并不需要重新分配资源，时间复杂度很大的资源调度算法此时并不需要运行。所以，在每次时间片中试图将资源分配给原用户，并在不成功时回滚，避免了系统正常运转中对于调度算法的调用，节省了大量时间。

然后，在边缘层设计了 2 个缓存队列，缓存队列 1 缓存来自云端的请求命令，并合并重复的请求命令。合并后的命令请求作为输入传到缓存队列 2，此队列用于缓存底层节点的数据，当请求的数据在缓存队列 2 时，直接从缓存队列 2 中返回数据到云端，进而返回给用户。

最后，在 KM 算法中加入了调度完成状态的检测，通过遍历的手段寻找最优匹配结果。显然，这个

算法的最差时间复杂度极高，但在第 1 层循环中剔除掉了已分配资源，且此算法的最差情况只发生在调度后期矩阵极为稀疏或进程较少的情况下，故相比于算法对于调度轮次的显著减少，其时间耗费影响并不明显。

给出 3 个算法的详细设计步骤。

算法 1 的输入是资源请求矩阵（用户和节点的调度关系），输出是经命令缓存队列处理后的代价矩阵。首先，从资源请求矩阵中获取资源请求。然后，将资源请求放入到一个命令数组中。之后，对数组进行排序，并合并数组中的相同元素。最后，输出合并后的命令矩阵。

算法 1. 命令缓存队列算法。

输入：代价矩阵；
输出：经命令缓存队列处理后的代价矩阵。

- ① 从代价矩阵中获取请求队列；
- ② 请求队列按照先来先服务的原则赋值到命令缓存数组中；
- ③ 排序命令缓存队列；
- ④ for 每个命令 do
- ⑤ 合并重复命令；
- ⑥ end for
- ⑦ 输出经命令缓存队列处理后的代价矩阵。

算法 2 给出了数据缓存队列设计的算法.输入是经命令缓存队列处理后的代价矩阵,输出是经数据缓存队列处理后的代价矩阵.因为缓存队列的大小受限,在仿真实验中,设置雾层的缓存能力是底层传感网某一刻产生数据量 X 的 $1/4$,缓存的节点数据主要依据于节点的状态(优先级,剩余能量,到边缘节点跳数),其中剩余能量和到边缘节点的跳数是并行的影响因素,例如 2 个节点的优先级和剩余能量相同,此时 2 个节点到边缘节点的跳数分别为 2 跳和 3 跳,则优先缓存跳数较小的节点数据.选择缓存节点数据之前,剩余能量和到边缘节点的跳数比重相当.

算法 2. 数据缓存队列算法.

输入:经命令缓存队列处理后的代价矩阵;

输出:经数据缓存队列后的代价矩阵.

- ① map = 代价矩阵;
- ② 设缓存队列大小为 $X/4+1$;
- ③ for map 中的每个进程 do
- ④ 在生成代价矩阵的同时获取节点的状态值(优先级 $priority$, 剩余能量 $E_{residual}$ 和到边缘节点的跳数 T_{hop});
- ⑤
$$S = \frac{priority[i]}{\sum_{i=1}^{|Y|} priority[i]} + \frac{E_{residual}[i]}{\sum_{i=1}^{|Y|} E_{residual}[i]} + \frac{T_{hop}[i]}{\sum_{i=1}^{|Y|} T_{hop}[i]}$$
;
- ⑥ 对 S 值的大小进行排序;
- ⑦ 插入排序组建缓存预处理队列;
- ⑧ end for
- ⑨ 输出经数据缓存队列处理后的代价矩阵.

算法 3 给出基于双缓存队列的扩展 KM 算法(extended KM with double buffer queues, EKMDB).算法输入为调度矩阵,输出为总的时间戳以及每一种资源调用完成时算法的状态.其中,加入了对于调度完成状态的检测,在当前用户调度未完成,即当前资源调用进程未运行完毕时算法将跳过调度分配算法,同时剔除已完成的进程.

算法 3. 基于双缓存队列的扩展 KM 算法.

输入:经数据缓存队列后的代价矩阵;

输出:时间戳、每轮调用的矩阵情况及资源调用情况.

- ① while 进程未全部结束 do
- ② 对 map 中进程遍历,寻找完成进程并剔除,设 $running$ 标记;

- ③ if $running=1$
- ④ 遍历当前分配,尝试分配资源,分配失败则回滚;
- ⑤ end if
- ⑥ 使用改进 KM 算法;
- ⑦ for map 中的每一个资源 do
- ⑧ 找到还未使用资源,将还未使用资源和对应的用户加入匹配,退出;
- ⑨ end for
- ⑩ end while

3.3 理论分析

定理 1. 设计传感云中用户和资源的最优调度方案是一个 NP 难问题.

证明. 若存在一个问题 p ,可以被分为 k 个部分($k \geq 1$ 且有限),其中至少有一个部分可以由一个已知的 NP-Hard 问题归约而来,则 p 为 NP-Hard 问题.

第 1 次分治:通过采用时间片算法,将问题转化为在一个时间片范围下对于二分图 G^n 产生最优分配 $M^n \in G^n$,显然,当 $|Y_i| = |Y|$ 时, M 为最优分配.此次分治采用贪婪方法将问题分划,以得出近似最优解,此时时间片长度 n 近似最小.

当 $\max\{Y_i\}$ 有限时,时间片算法使子问题 p_1 可在多项式时间 $O(n)$ 中归约为问题 p .显然,对于问题 p 向子问题 p_1 的分划是在二项式时间内的,即子问题 p_1 可归约为 p .

因为 Y_i 有上界,问题 p_1 可以视为在定上界多约束下的 $1-M$ 二分图组匹配问题,此问题为 NP 完全问题^[22-23],则 p_1 为 NP-Hard 问题,即证 p 为 NP-Hard 问题. 证毕.

定理 2. 基于双缓存队列的扩展 KM 算法的时间复杂度同级于 KM 算法.

证明. 第 2 次分治:对于子问题 p_1 ,采用分治的方法将其拆分成 2 个问题,首先,由二分图 G^n 获得最优匹配 M^n ,然后利用剩余资源检测算法提高剩余资源的利用率,获得了 2 个子问题 p_2 和 p_3 .

对于问题 p_2 ,通过采用了 Munkres 对于 KM 算法的改进算法,其时间复杂度可以估计为

$$O(p_2)=O(|X|^2|Y|). \tag{1}$$

对于问题 p_3 ,通过对剩余资源的类冒泡排序(在单个时间片下实际体现为最小值获取)获取最优分配方案,显然,当 n 有限时,此问题为时间复杂度为 $O(n^3)$ 的 P 类问题,即:

$$O(p_3)=O(n^3)=O(p_2). \tag{2}$$

证毕.

定理 3. 基于双缓存队列的扩展 KM 算法可以最大化资源利用率.

证明. 如图 3 所示,图 3 的粗线(红线)为 KM 算法调用 1 次所得到的结果.即 $U_1 \rightarrow R_5, U_2 \rightarrow R_1, U_3 \rightarrow R_2$.显然,资源 R_3 仍可以被继续调度,出现了资源浪费.此时的资源利用率并非 100%.

通过遍历的手段寻找剩余资源,当检测到资源有 R_3 空闲时,将其分配给用户 U_1 ,提高了资源利用率. 证毕.

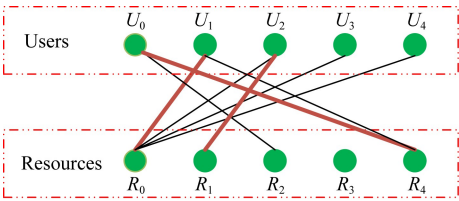


Fig. 3 Maximize resource utilization
图 3 最大化资源利用率示意图

4 实验与结果

4.1 实验环境设置

本节描述实验的基本设置环境.实验采用 Visual studio 和 MATLAB R2017b 作为仿真平台.实验采用了 8 种场景,用户数从 50~400,为等差数列.详细的实验参数如表 2 所示.为了进一步测试提出的算法的扩展性,发现资源的请求符合正态分布.故正态分布下分别进行了实验.为了验证方法的性能,分别与先来先服务方法(FIFO)^[10],扩展 KM 算法(extension of KM, EKM)^[18],仅使用底层数据缓存队列的扩展 KM 算法(EKM with buffer data, EKMB)^[20] 进行比较.

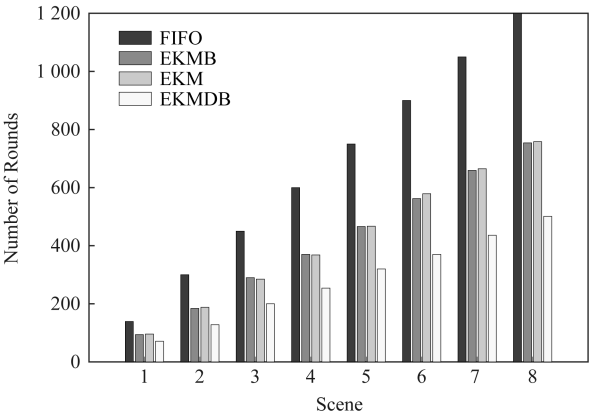
Table 2 Experiment Parameters Descriptions

表 2 实验参数描述

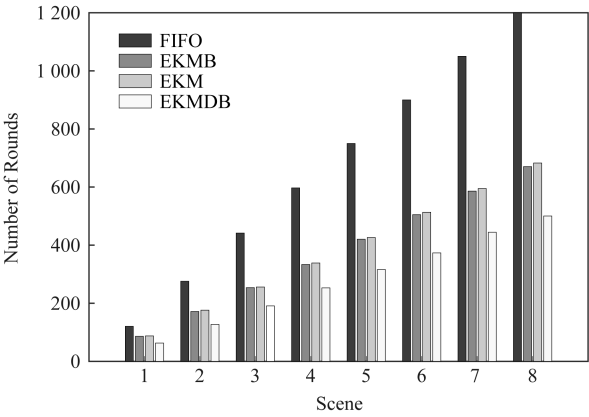
Parameters	Descriptions
Operating System	Windows 10
Memory Size/GB	8
CPU	Intel Core I7
Programing Language	C# /C++
Compiler	MATLAB R2017b/ Visual Studio 2017
Number of Cloud	1
Delay Time from Sensor to Edge/ms	6
Delay Time from Sensor to Cloud/ms	24

4.2 实验和结果

图 4~5 是不同场景下基于正态分布的平均运行轮数和资源利用率的比较.图 4(a)和图 5(a)是用用户值和资源值的比值为 1/5 的情况.图 4(b)和图 5(b)是用用户值和资源值的比值为 2/5 的情况. EKMDB 算法所需的调用轮数是最小的,且较其他算法所耗费的轮数都大幅度的下降,且随着场景越来越复杂,算法有更好的适用性,这是因为在边缘层经过 2 个缓存队列的处理,大大简化了代价矩阵的规模,有更快的处理速度.



(a) The ratio of users to resources is 1/5



(b) The ratio of users to resources is 2/5

Fig. 4 Comparison of the number of rounds
图 4 不同场景运行轮数比较

图 6 是不同场景下基于正态分布不同算法平均运行时间的比较.其中,图 6(a)是用用户值和资源值的比值为 1/5 时的情况.图 6(b)是用用户值和资源值的比值为 2/5 时的情况.由图 6 可知,当场景的规模较小时,算法的运行时间很短,随着场景逐渐复杂,算法的运行时间也越来越大.从结果可以看出,提出的 EKMDB 算法与传统的 KM 算法相比,该算法有较低的时间复杂度,也佐证了定理 2 的正确性.

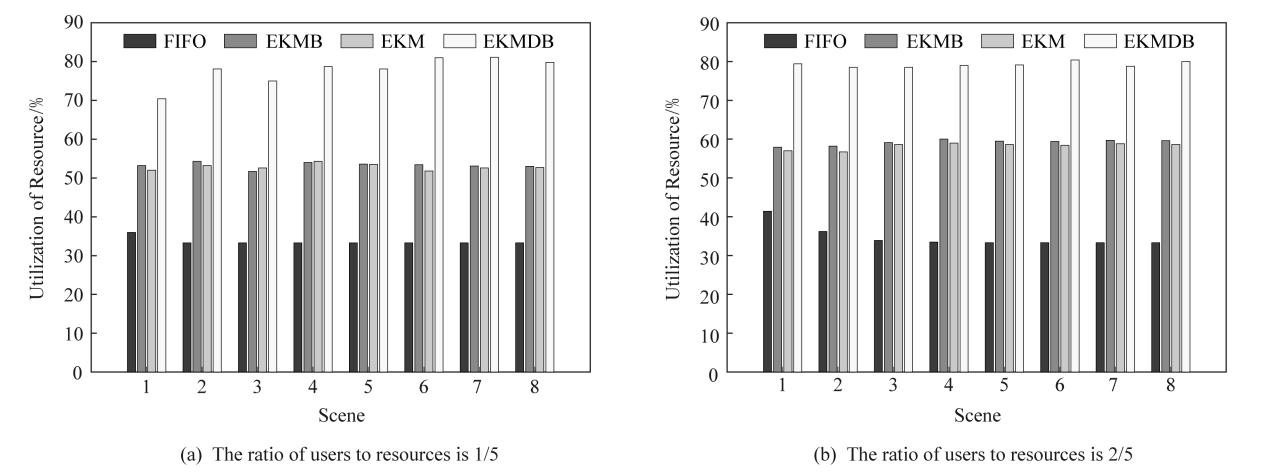


Fig. 5 Comparison of the resource utilization
图5 不同场景资源利用率比较

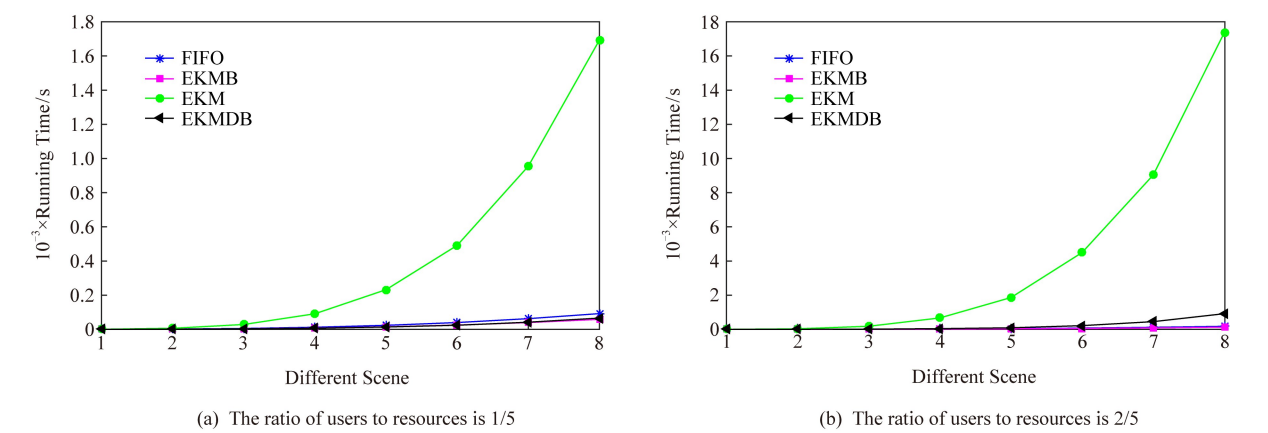


Fig. 6 Comparison of the running time
图6 不同场景运行时间比较

图7和图8显示了任务数从1 000~64 000时运行轮数和运行时间的变化.这些结果表明:提出的算法调用轮数最少,运行时间最短.这也表明在边缘层利用缓存队列进行预处理有很好的性能.

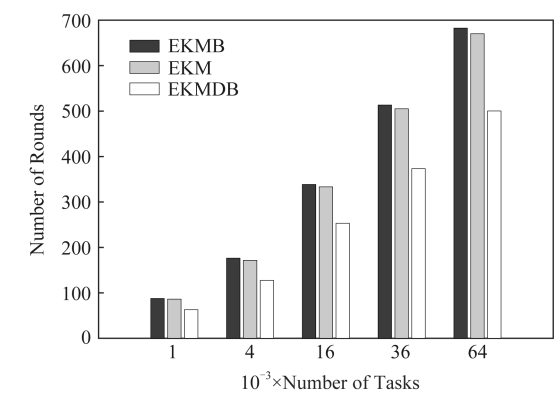


Fig.7 The number of rounds with different tasks
图7 运行轮数和任务数量的关系

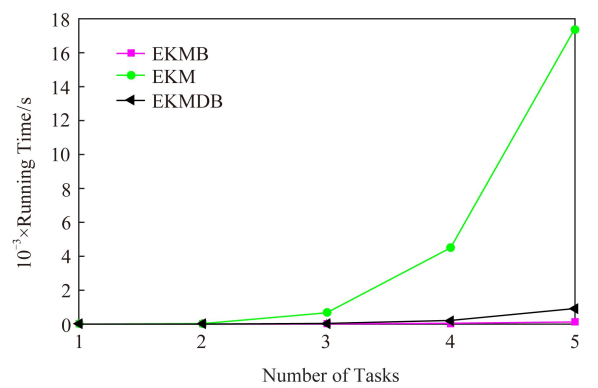


Fig. 8 The running time with different tasks
图8 运行时间和任务数量的关系

5 总 结

本文研究了基于边缘计算的传感云系统低耦合

控制方法.对于传感云中的耦合问题,首先证明了转化的问题是 NP-Hard 问题,基于边缘计算和双缓存队列的思想,提出了基于双缓存队列的扩展 KM 算法解决这一问题.仿真结果验证了相关理论的正确性,同时表明,该方法能有效地降低资源调度时间,提高资源利用率,该方法有效地解决了传感云系统中的耦合问题.所提算法核心思想之一是在边缘层设计了 2 个缓存队列,以“空间换时间”的思想提高算法的性能,越复杂的场景,缓存队列所占空间越大,空间复杂度越高.未来研究工作之一即在有效解决耦合问题的前提下减小算法的空间复杂度.

参 考 文 献

- [1] Wang Tian, Zhang Guangxue, Liu Anfeng, et al. A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing [J]. IEEE Internet of Things Journal, 2019, 6(3): 4831-4843
- [2] Wang Wenhua, Wang Tian, Wu Qun, et al. Survey of delay-constrained data collection with mobile elements in WSNs [J]. Journal of Computer Research and Development, 2017, 54(3): 474-492 (in Chinese)
(王文华, 王田, 吴群, 等. 传感网中时延受限的移动式数据收集方法综述[J]. 计算机研究与发展, 2017, 54(3): 474-492)
- [3] Wang Tian, Shen Xuewei, Luo Hao, et al. Research progress of trusted sensor-cloud based on fog computing [J]. Journal on Communications, 2019, 40(3): 170-181 (in Chinese)
(王田, 沈雪微, 罗皓, 等. 基于雾计算的可信传感云研究进展[J]. 通信学报, 2019, 40(3): 170-181)
- [4] Wang Tian, Zhang Guangxue, Cai Shaobin, et al. Survey on trust evaluation mechanism in sensor-cloud [J]. Journal on Communications, 2018, 39(6): 37-51 (in Chinese)
(王田, 张广学, 蔡绍滨, 等. 传感云中的信任评价机制研究进展[J]. 通信学报, 2018, 39(6): 37-51)
- [5] Zeng Deze, Gu Lin, Guo Song, et al. Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system [J]. IEEE Transactions on Computers, 2016, 65(12): 3702-3712
- [6] LogMeIn Inc. Xively by LogMeIn [EB/OL]. [2019-08-20]. <https://www.xively.com>
- [7] Github. Nimbit-promote yourself [EB/OL]. [2019-08-20]. <http://www.nimbits.com>
- [8] Sharma M, Sindhwani P, Maheshwari V. Genetic algorithm optimal approach for scheduling processes in operating system [J]. International Journal of Computer Science and Network Security, 2014, 14(5): 91-94
- [9] Liu Bingtao, Wang Da, Ye Xiaochun, et al. The data-flow block based spatial instruction scheduling method [J]. Journal of Computer Research and Development, 2017, 54(4): 750-763 (in Chinese)
(刘炳涛, 王达, 叶笑春, 等. 基于数据流块的空间指令调度方法[J]. 计算机研究与发展, 2017, 54(4): 750-763)
- [10] Li Jian, Ma Tinghuai, Tang Meili, et al. Improved FIFO scheduling algorithm based on fuzzy clustering in cloud computing [OL]. [2019-08-20]. <https://doi.org/10.3390/info8010025>
- [11] Tang Zhuo, Qi Ling, Cheng Zhenzhen, et al. An energy-efficient task scheduling algorithm in DVFS-enabled cloud environment [J]. Journal of Grid Computing, 2016, 14(1): 55-74
- [12] Kayvanfar V, Zandieh M, Teymourian E. An intelligent water drop algorithm to identical parallel machine scheduling with controllable processing times: A just-in-time approach [J]. Computational and Applied Mathematics, 2017, 36(1): 159-184
- [13] Chopra S, Notarstefano G, Rice M, et al. A distributed version of the hungarian method for multirobot assignment [J]. IEEE Transactions on Robotics, 2017, 33(4): 932-947
- [14] Chen Xiaoxu, Wu Heng, Wu Yuewen, et al. Large-scale resource scheduling based on minimum cost maximum flow [J]. Journal of Software, 2017, 28(3): 598-610 (in Chinese)
(陈晓旭, 吴恒, 吴悦文, 等. 基于最小费用最大流的大规模资源调度方法[J]. 软件学报, 2017, 28(3): 598-610)
- [15] Zhang Xinyuan, Zhang Jun, Gong Yuejiao, et al. Kuhn - Munkres parallel genetic algorithm for the set cover problem and its application to large-scale wireless sensor networks [J]. IEEE Transactions on Evolutionary Computation, 2016, 20(5): 695-710
- [16] Zhu Haibin, Liu Dongning, Zhang Siqin, et al. Solving the group multirole assignment problem by improving the ILOG approach [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017, 47(12): 3418-3424
- [17] Wen Zhenyu, Yang Renyu, Garraghan P, et al. Fog orchestration for Internet of things services [J]. IEEE Internet Computing, 2017, 21(2): 16-24
- [18] De Brito M S, Hoque S, Steinke R, et al. Application of the fog computing paradigm to smart factories and cyber-physical systems [OL]. [2019-08-20]. <https://doi.org/10.1002/ett.3184>
- [19] Magurawalage C S, Yang Kun, Wang Kezhi, Aqua computing: Coupling computing and communications [J]. arXiv preprint, arXiv: 1510.07250, 2015
- [20] Wang Tian, Liang Yuzhu, Jia Weijia, et al. Coupling resource management based on fog computing in smart city systems [J]. Journal of Network and Computer Applications, 2019, 135: 11-19

[21] Li Chunlin, Tang Jianhang, Tang Hengliang, 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

[22] Gregory J P. Complexity of a disjoint matching problem on bipartite graphs [J]. Information Processing Letters, 2016, 116: 649-652

[23] Itai A, Rodeh M, Tanimoto S L. Some matching problem for bipartite graphs [J]. Journal of the Assocmtton for Computmg Machinery, 1978, 25(4): 517-525

[24] Wang Tian, Li Yang, Jia Weijia, et al. Research progress of sensor-cloud security [J]. Journal on Communications, 2018, 39(3): 35-52 (in Chinese)
(王田, 李洋, 贾维嘉, 等. 传感云安全研究进展[J]. 通信学报, 2018, 39(3): 35-52)

[25] Zeng Jiandian, Wang Tian, Jia Weijia, et al. A survey on sensor-cloud [J]. Journal of Computer Research and Development, 2017, 54(5): 925-939 (in Chinese)
(曾建电, 王田, 贾维嘉, 等. 传感云研究综述[J]. 计算机研究与发展, 2017, 54(5), 925-939)

[26] Wang Tian, Zhou Jiyaun, Liu Anfeng, et al. Fog-based computing and storage offloading for data synchronization in IoT [J]. IEEE Internet of Things Journal, 2019, 6(3): 4272-4282

[27] Wang Tian, Zhou Jiyuan, Huang Minzhe, et al, Fog-based storage technology to fight with cyber threat [J]. Future Generation Computer Systems, 2018, 83: 208-218

[28] Wang Tian, Li Yang, Wang Guojun, et al, Sustainable and efficient data collection from WSNs to cloud [J]. IEEE Transactions on Sustainable Computing, 2019, 4(2): 252-262

[29] Gao Zhenguo, Chen Danjie, Sun Peng, et al. KM-based efficient algorithms for optimal packet scheduling problem in cellular/infostation integrated networks [J]. Ad Hoc Networks, 2018, 77: 84-94

[30] Li Chunlin, Bai Jingpan, Tang Jianhang. Joint optimization of data placement and scheduling for improving user experience in edge computing [J]. Journal of Parallel and Distributed Computing, 2019, 125: 93-105



Liang Yuzhu, born in 1994. Master candidate. His main research interests include wireless sensor networks, mobile computing and fog computing.



Mei Yaxin, born in 1996. Master candidate. Her main research interests include wireless sensor networks, mobile computing and edge computing.



Yang Yi, born in 1997. Master candidate. His main research interests include wireless sensor networks, fog computing and mobile computing.



Ma Ying, born in 1982. PhD, associate professor. His main research interests include big data, edge computing and cloud computing.



Jia Weijia, born in 1957. PhD, professor and PhD supervisor. His main research interests include next generation IoT, knowledge graph constructions, multicast and anycast QoS routing protocols.



Wang Tian, born in 1982. PhD, professor. His main research interests include wireless sensor networks, edge computing and mobile computing.