

基于边缘计算的单任务迁移策略研究

朱霞, 靳昊恺

(金陵科技学院网络与通信工程学院, 江苏 南京 211169)

摘要: 针对单用户场景下的任务迁移问题, 构建单用户的任务迁移模型, 并提出任务迁移策略。通过任务迁移策略判断每个任务是否应该被迁移, 同时得出任务的执行顺序, 并综合评估延时与能耗对应的权重以得到系统总体的最低负载。采用了离散二进制粒子群算法优化了迁移策略来降低时间复杂度。仿真结果表明, 所提出的迁移策略对减少延迟和能耗具有明显的效果。还在现有的联合任务迁移策略上进行了改进, 通过优化二进制粒子群中粒子的初始位置, 进行相应的对比仿真实验, 来降低搜索结果的离散程度。

关键词: 边缘计算; 任务迁移策略; 5G; 云计算; 二进制粒子群算法

中图分类号: TN929.5

文献标识码: A

文章编号: 1672-755X(2020)03-0014-05

The Research on Migrating Strategy Based on Edge Computing

ZHU Xia, JIN Hao-kai

(Jinling Institute of Technology, Nanjing 211169, China)

Abstract: This paper describes the problem of tasks migration in single user scenario, constructs a single user task migration model, and proposes a task migration strategy. The task migration strategy is used to determine whether each task should be migrated, what the execution order of the task is obtained, and the overall minimum load of the system through comprehensive evaluating the weight corresponding to delay and energy consumption. In this paper, discrete binary particle swarm optimization algorithm is used to optimize the migration strategy to reduce the time complexity. Simulation results show that the proposed migration strategy is effective in reducing delay and energy consumption. This paper also proposes an improvement on the existing joint task migration strategy, which reduces the dispersion degree of search results by optimizing the initial position of particles in binary particle swarm, and carries out corresponding comparative simulation experiments.

Key words: edge computing; task migration strategy; 5G; cloud computing; BPS

随着云计算、人工智能、移动互联网、大数据、物联网等技术的快速发展, 人类正在步入万物互联的时代。指数增长的数据流量、海量的物联网设备和越来越多样化的服务场景使得 4G 网络架构和云计算中心无法满足需求。为解决这样的难题, 必须推进 5G 的快速发展。而在 5G 与物联网时代下, 移动通信的服务范围由以人为主体的拓展至人与物、物与物的通信。在一些特殊的应用场景, 如无人驾驶、远程医疗等环境下, 对移动网络提出了高可靠性、低延迟的通信需求。亟待边缘计算(Edge computing, EC)架构为云计算中心预处理网络边缘侧产生的海量数据来分担数据回传至核心网的压力。

收稿日期: 2020-06-20

基金项目: 江苏省大学生创新训练计划项目(201813573029Y)

作者简介: 朱霞(1980—), 女, 江苏南通人, 副教授, 博士, 主要从事智能优化研究。

国内对于边缘计算的定义^[1]。边缘计算是一个开放式平台,可在源或数据源附近的网络边缘集成网络、计算、存储和应用程序等核心功能。并能提供附近的边缘智能服务,以满足敏捷连接、实时业务、数据优化、应用智能、安全和隐私保护等行业数字化的关键需求。

2017年3月邓茂菲^[2]提出了分布式任务迁移决策方案,该方案使得用户可以在一定迭代之后达到纳什均衡,并且在多种场景下都能达到优化任务迁移消耗的目的;2017年5月郝义学^[3]针对边缘缓存中用户移动性问题进行研究,分析了小型基站覆盖区域对用户设备的影响,提出一种移动缓存策略优化问题,并利用贪婪算法求解,较传统缓存策略的命中率有显著提升。2018年6月孔月^[4]通过对任务进行细粒度的划分,并综合考虑了复杂多依赖关系型的边缘计算场景,提出了一种动态的迁移策略。该迁移策略可以在移动边缘计算环境下选择将任务迁移至最优的边缘计算服务器中执行,有效地提高了任务吞吐量和系统资源利用率。并利用改造后的 Cloud analyst 仿真平台对实验进行仿真模拟。2018年6月刘国强^[5]针对多用户场景提出了一个任务卸载策略,构建了多用户博弈模型,并通过两阶段任务迁移求解纳什均衡。2018年3月赵丹^[6]通过将边缘计算环境下多种影响因素纳入模型内,设计出了 D2D 计算迁移模型,并基于该模型提出了迁移策略,通过使用任务迁移策略有效地降低了任务延时和能量消耗。

1 单用户的任务迁移策略研究

本文主要研究单基站、单用户的边缘计算场景,用户终端(User terminal, UT)通过无线与边缘节点基站(Edge node base station, ENBS)连接,该基站上搭建了边缘计算服务器(EC server),用来为用户提供计算的服务。在单用户场景,可以认为 EC 服务器的计算资源是充足的,但不是无限的^[7-9]。在本次研究的场景下假设在 EC 服务器上同一时间只可以执行一个任务。在 EC 服务器执行完计算任务后,将计算结果发送给本地终端。

1.1 单用户任务迁移模型

本节中,将会针对单用户场景的问题构建任务迁移模型,先构建一个整体的系统模型,再针对各个组成部分,进行单独构建,包括:计算队列模型、本地终端计算模型、云服务器端计算模型和总体负载模型等。本文设计的整体系统模型如图1所示。从图中可以看出,在本地移动设备中存在一个用于存储所有计算任务的队列。本地移动设备通过任务迁移策略计算任务判断是否需要被迁移,需要迁移的任务会存储到本地的发射队列中,由发射单元(Local transmit unit, LTU)上传给 EC 服务器,并存储到 EC 服务器中的任务队列中。另外本地移动设备和 EC 服务器中都存在 CPU 处理单元,用于执行计算任务。

在服务器端,有一个用于存储被迁移的任务的队列。在本次研究中假定服务器的计算能力远远大于移动设备,但是同时只能执行一个任务,于是便会有已被上传但是未被执行完成的任務,这些任务就被存放在 EC server 上的任务队列中。

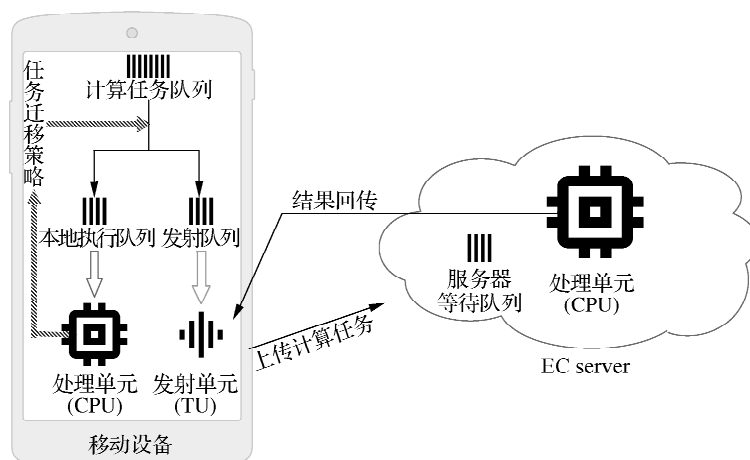


图1 整体系统模型

从图 1 中可以看出,本次设计的整体系统模型存在两个存放任务的队列,其中一个在本地移动设备,来存放所有等待被执行的任务,记为 T_{local} ;另外一个队列在 EC server 端,用来存储被迁移任务,记为 T_{cloud} 。假设在本地移动设备上一共存在 T_{num} 个任务待执行,则任务可以用 $T = \{T_1, T_2, \dots, T_{\text{num}}\}$ 来表示。其中每一个计算任务又存在四个元素: $T_i = (Q_i, d_i, c_i, r_i)$ 。其中 Q_i 是一个 $\{0, 1\}$ 的二进制数,用来表示任务 T_i 是否需要迁移,当 $Q_i = 0$ 时,表示任务在本地终端执行,当 $Q_i = 1$ 时,表示任务需要迁移; d_i 时表示任务 T_i 的数据量, d_i 中包含了这个计算任务的执行程序、参数、依赖数据等,当这个任务需要被上传到 EC 服务器上的时候,这些数据便会通过本地移动设备的 TU 单元发送到 EC 服务器; c_i 表示任务 T_i 所需要的计算量,可以用 CPU 操作数来表示; r_i 则表示任务 T_i 被执行完成后的结果数据,如果任务 T_i 被迁移到 EC 服务器上执行的话, r_i 需要从服务器上下载到本地移动设备。本次研究主要探究的是任务迁移算法,这里对计算结果 r_i 忽略不计。

EC 服务器端的任务队列为 T_{cloud} ,其中存放的是已被上传但是未被执行完成的任务。在 EC 服务器端对任务的执行需要遵循先进先出原则,于是在本模型中便可以将 EC 服务器端任务执行顺序视为本地任务迁移的顺序。优化联合任务迁移策略经过上面对单用户任务迁移模型的构建,得到的最终目标是使用任务迁移策略来优化整体负载 $K^{[10-11]}$ 。通过对单用户场景问题描述的分析,得到了本次研究的任务迁移过程主要分为两个步骤:首先,根据任务迁移策略划分出在本地终端执行与在云服务器终端执行的任务集;其次,确定不同任务集的任务执行顺序,得到任务执行顺序为 T_{local} 与 T_{cloud} 。通过分析不难发现,对于这个问题,它的时间复杂度是 $O(2^N N!)$ 。本文使用了优化后的基于二进制粒子群算法和流水车间调度理论组成的优化联合任务迁移策略来降低时间复杂度。

1.2 本地任务最优调度策略

在本小节中,先假设已经获得了本地终端的执行任务列表,由于各个计算任务之间互不依赖,并且根据上一节的本地终端计算模型的分析可以看出,能耗与本地终端的任务执行顺序无关,并且为了最小化执行完成所有本地任务所需的时间,需要对本地任务的时间以升序排列。由此得出本地执行任务为 T_{local} 。

1.3 流水线任务调度策略

在这一小节中,假设已经获得了所需要迁移的计算任务列表 T_{N_2} ,本节就是对这个计算列表进行排列,由于待迁移的任务必须在迁移完成后才能在 EC 服务器上执行,并且任务互不依赖,可以证明,这是一个两机器流水线调度问题,可以使用 Johnson 算法对其进行求解^[5, 12-13]。在 Johnson 算法中,首先将任务队列集合 T_{N_2} 分成两组不相关的子集 A 和 B ,并且 A 和 B 满足如下条件:

$$\begin{aligned} A &= \left\{ T_i \in T_{N_2} \mid \frac{d_i}{R(p_{tu})} < \frac{c_i}{f_{ser}} \right\} \\ B &= \left\{ T_i \in T_{N_2} \mid \frac{d_i}{R(p_{tu})} \geq \frac{c_i}{f_{ser}} \right\} \end{aligned} \quad (1)$$

1.4 离散二进制粒子群算法

粒子群算法(Particle swarm optimization, PSO)^[14]最初是由 J. Kennedy 和 R. C. Eberhart 于 1995 年提出的一个优化算法,这个算法的灵感来源于鸟群、鱼群等群体的捕食行为。在该算法中,每个粒子都有自己的初始位置(Position)和初始速度(Velocity),并通过一个权重来动态地控制 PSO 算法的搜索性能。粒子的位置表示是该粒子的一个可行解,粒子的速度则表示粒子下次的位置与当前的距离。在 J 维空间中,存在 N 个粒子,则粒子位置为 $x_{ij} = (x_{i1}, x_{i2}, \dots, x_{ij})$,将 x_{ij} 代入 $f(x)$ 求出适应值,用 $PBest_j$ 表示第 i 个粒子在第 j 维的历史最优位置,用 $Gbest_j$ 表示全体粒子群在第 j 维的历史最优解。粒子的速度则是用 $v_{ij} = (v_{i1}, v_{i2}, \dots, v_{ij})$ 表示,粒子更新速度与位置使用如下公式:

$$v_{ij}^k = \omega \cdot v_{ij}^{k-1} + c_1 \cdot \text{rand}() \cdot (PBest_j - x_{ij}^{k-1}) + c_2 \cdot \text{rand}() \cdot (GBest_j - x_{ij}^{k-1}) \quad x_{ij}^k = x_{ij}^{k-1} + v_{ij}^k \quad (2)$$

其中 v_{ij}^k 表示粒子第 k 次迭代的速度矢量的 j 维分量, x_{ij}^k 表示粒子第 k 次迭代的位置矢量的 j 维分量, c_1 和 c_2 分别是两个加速度常数, $\text{rand}()$ 表示一个 $(0, 1)$ 的随机数,用来提升搜索的随机性, ω 表示惯性权重

用来调节对空间的搜索范围。经过多次迭代,最终搜索出最优解。但是标准的粒子群优化算法只能在连续的空间进行最优解的搜索。为了对离散空间进行最优解搜索,J. Kennedy 和 R. C. Eberhart 在 1997 年提出粒子群算法的另一个版本,离散二进制粒子群算法(Binary particle swarm optimization, BPSO)^[15],用来适应离散二进制空间进行最优解搜索的问题。

在离散二进制 PSO 算法中,主要是将粒子位置每一维 x_{ij} 和粒子的历史最优解 $PBest_j$ 都设为二进制 0 或 1,而粒子的速度的值则使用 sigmoid 函数映射到区间 $[0,1]$,sigmoid 函数如下:

$$s(v_{ij}) = \frac{1}{1 + \exp(-v_{ij})} \quad (3)$$

这里的 $s(v_{ij})$ 表示粒子的位置 x_{ij} 取值为 1 的概率,粒子使用如下公式改变位置的值:

$$x_{ij} = \begin{cases} 1, & \text{if } rand() \leq s(v_{ij}) \\ 0, & \text{other wise} \end{cases} \quad (4)$$

这里的 $rand()$ 是一个 $[0,1]$ 的随机数,为了防止 $s(v_{ij})$ 过于接近 0 或 1 并且使用参数 $v_{\max}$ 作为最大速度值用于限制 v_{ij} 的范围, $v_{ij} \in [-v_{\max}, v_{\max}]$ 目的是为了限制粒子位置 x_{ij} 取 0 或 1 的概率。

2 基于离散二进制粒子群的任务迁移策略

本文对刘国强提出的联合计算任务迁移算法(Joint task offloading, JTO)^[5]进行了优化改进,并对二进制粒子群的迭代次数进行了优化设置,使其在任务数较多的时候仍然能够获得相对稳定的局部最优负载。在上一节中构建了 $Q_i \in \{0,1\}$ 用于表示任务 T_i 是否迁移。0 表示任务在本地终端执行,1 表示任务迁移至云服务器端执行。于是向量 $Q = (Q_1, Q_2, \dots, Q_{T_{\text{num}}})$ 表示任务迁移的结果。在本小节中,先随机给出一组向量 Q ,用枚举法求出其最优解的时间复杂度为 $O(2^N)$,而本小节尝试通过二进制粒子群算法进行优化求解。

通过向量 Q 可知每个粒子的位置判断任务是否需要迁移,并通过算法得到任务执行序列 T_{local} 和 T_{server} ,便可以由此计算出系统的整体负载 K ,通过比较判断当前粒子所在位置是否为其最佳位置,从而更新任务迁移策略。经过一定的迭代,算法便能求出一个相对的优化解^[14-17]。本文设置的仿真场景为:一个移动设备、一个基站、一个 EC 服务器,移动设备与基站之间通过无线网络连接,基站处设置了 EC 服务器集群。假设移动设备与基站之间的通信状况好,信道空闲无占用,在本次仿真实验中忽略传输数据时发生的波动,将发射模块的发送速率设为固定的数值。本文在联合任务迁移策略的基础上,设想通过优化二进制粒子群算法中粒子的初始位置,来降低粒子群中历史最优解的离散程度。并通过实验验证了这一设想,在一定达到任务数的时候,结果是较为明显的。实验结果如图 2 所示。

图 2 描述的是:在一定任务数的条件下,经过 G_{num} 次的迭代后得到的一组向量 K_{history} ,将 K_{history} 中的最大值记为 $K_{\max}$,最小值记为 $K_{\min}$,那么 ΔK 即为 $K_{\max}$ 与 $K_{\min}$ 的差。从图中可以看出,当任务数量达到 700 左右的时候,优化的 ΔK 逐渐开始大幅低于优化前的 ΔK ,可以看出经过优化粒子的初始位置,能够在任务数,也就是粒子数量较多的条件下,有效的降低搜索结果的离散程度。但是通过比较优化前和优化后的联合任务迁移策略在搜索最优负载的过程中,由于二进制粒子群算法的随机性,优化后的策略没有很明显的优势,这是需要继续深入研究的内容。

图 3 展示的是在任务数量达到 3 000 时,负载和任务数的关系。并且这里选取了优化初始粒子位置后的二进制粒子群算法,与上一节提到的其它四种迁移策略作对比。

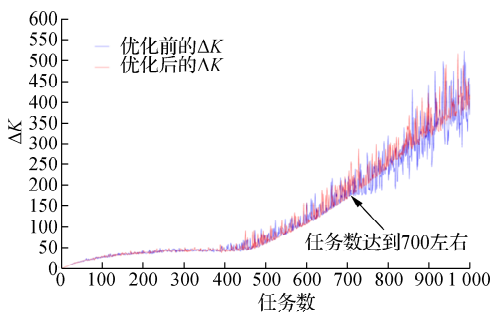


图2 负载离散图

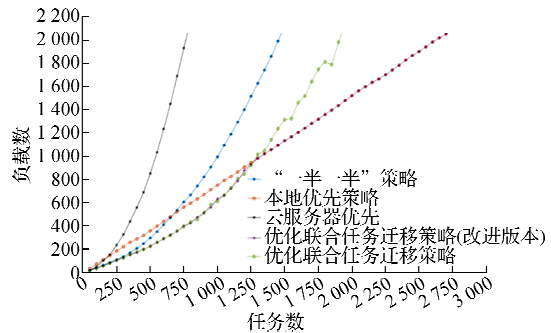


图3 任务数为1~3000时系统负载变化图

从图3可以看到在任务数量小于1 250的情况下,优化前和优化后的迁移策略在对负载 K 的计算中得出的结果比较接近。但是由于二进制粒子群算法的随机性,使得两者的结果相差并不大,并且在一定范围内波动。当任务数量大于1 250的情况下,原始的优化联合任务迁移策略计算出的负载 K 开始逐渐大于本地优先策略计算出的负载 K ,说明在任务数量较多的时候,让任务在本地执行是一个更好的选择,但是通过一定的迭代未改进的联合任务迁移策略在任务数较多的情况下没有搜索到更优的负载 K 。而优化过粒子初始位置的联合任务迁移策略计算出的负载 K 与其他策略计算出的负载 K 相比,在大部分情况下都是更低的。从图中也可以发现,优化后的联合任务迁移策略的曲线大致为其它迁移策略的最低点的连线。由此也可以认为改进后的联合任务迁移策略在不同任务数下搜索最优负载 K 的表现中有着更好的表现。

3 结 语

本文主要研究了单用户场景下的任务迁移策略。首先描述了单用户场景下的问题,通过分析单用户场景下的问题来确定研究方向;其次针对问题构建系统模型,并将其转化为在一定条件限制下的最小化负载问题,并由此提出了任务迁移策略;然后通过设计迁移策略,并利用复杂度较低的二进制粒子群算法优化了任务迁移策略;最后并通过实验仿真对迁移策略进行了检验。通过仿真模型实验的结果可以看出:本文所提出的优化联合任务迁移策略能有效地降低执行任务的能耗和延时。本文对刘国强提出的联合计算任务迁移算法进行了优化改进,并对二进制粒子群的迭代次数进行了优化设置,使其在任务数较多的时候仍然能够获得相对稳定的局部最优负载。

参考文献:

- [1] 边缘计算产业联盟(ECC),工业互联网产业联盟(AII). 边缘计算产业联盟白皮书[EB/OL]. <https://www.huawei.com/minisite/iot/img/edge-computing-cn.pdf>, 2018(11)
- [2] 邓茂菲. 基于移动边缘计算的任务迁移策略研究[D]. 北京:北京邮电大学, 2017
- [3] 郝义学. 5G网络移动边缘缓存与计算研究[D]. 武汉:华中科技大学, 2017
- [4] 孔月. 移动边缘计算环境下的任务迁移策略研究[D]. 西安:西安理工大学, 2018
- [5] 刘国强. 基于移动边缘计算的任务卸载策略研究[D]. 哈尔滨:哈尔滨工业大学, 2018
- [6] 赵丹. 面向5G网络的边缘计算迁移策略研究与仿真[D]. 北京:北京邮电大学, 2018
- [7] Mao Y, Zhang J, Letaief K B. Dynamic computation off loading for mobile-edge computing with energy harvesting devices[J]. IEEE Journal on Selected Areas in Communications, 2016(12): 236 - 243
- [8] 陈建业. 移动边缘计算环境下的服务迁移策略设计与实现[D]. 北京:北京邮电大学, 2018
- [9] 王伟. 面向边缘计算的光网络业务提供技术研究[D]. 北京:北京邮电大学, 2018
- [10] 李博. 移动社会网络内容分发服务及边缘缓存策略研究[D]. 成都:电子科技大学, 2018
- [11] 葛志诚. 一种移动内容分发网络的分层协同缓存机制[J]. 计算机学报, 2018(10): 236 - 245
- [12] 王梓. 移动边缘计算中基于移动模型的任务迁移算法与协议研究[D]. 成都:电子科技大学, 2018
- [13] Emmons H, Vairaktarakis G. Flow shop scheduling: theoretical results, algorithms, and applications[J]. Springer, 2013 (3): 56 - 62
- [14] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]. New York: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995: 245 - 255
- [15] Kennedy J, Eberhart R C. A discrete binary version of the particle swarm algorithm[C]. Nagoya: 1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, 1997: 562 - 270
- [16] Li T, Wu M, Zhao M. Consumption considered optimal scheme for task offloading in mobile edge computing[C]. Africa: International Conference on Telecommunications IEEE, 2016: 123 - 129
- [17] Wang J, Peng J, Wei Y, et al. Adaptive application offloading decision and transmission scheduling for mobile cloud computing[J]. China Communications, 2017(3): 96 - 102

(责任编辑:谭彩霞)