Software Engineering and Applications 软件工程与应用, 2021, 10(6), 746-756
Published Online December 2021 in Hans. http://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2021.106079

Hans汉斯

基于启发式算法的混合云工作流调度算法

朱宇宁，何利力

浙江理工大学，信息学院，浙江 杭州

收稿日期：2021年11月30日；录用日期：2021年12月16日；发布日期：2021年12月27日

摘　要

伴随着“互联网+”技术的飞速发展，世界各地的网络环境越来越好，通过云计算技术操作异地计算资源的情况也越来越多。其中当本地资源不足时，私有云、公有云混合的混合云方案被各大企业广泛应用，本文针对混合云环境中的工作流任务调度问题，研究在满足任务截止期约束的同时使私有云利润最大化，节约企业成本。在私有云环境中，本文提出了一种优化的蚁群算法(Ant Colony Optimization Workflow Scheduling, ACOWS)用于工作流在用户指定的期限内完成任务的执行。在此基础上提出混合云下的动态多工作流调度算法(Hybrid Cloud Deadline-Constrained Cost Workflows Scheduling, HCDCW)，该算法将优先在私有云中调度执行，当任务执行时间超出任务截止期约束时，使用公有云调度部分工作流。在实验阶段，利用WorkflowSim仿真平台对算法进行了验证，实验结果表明在不同截止期，该调度算法相比于传统混合云工作流调度算法能有效的帮助企业在使用混合云过程中降低租用公有云的费用成本，并获得更快的执行时间。

关键词

云计算，混合云，科学工作流，蚁群算法

Hybrid Cloud Workflow Scheduling Algorithm Based on Heuristic Algorithm

Yuning Zhu, Lili He

School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou Zhejiang

Received: Nov. 30th, 2021; accepted: Dec. 16th, 2021; published: Dec. 27th, 2021

Abstract

With the rapid development of the “Internet+” technology, the network environment around the world is getting better and better, and there are more and more cases of operating remote computing resources through cloud computing technology. Among them, when the local resources are

文章引用: 朱宇宁, 何利力. 基于启发式算法的混合云工作流调度算法[J]. 软件工程与应用, 2021, 10(6): 746-756.
DOI: 10.12677/sea.2021.106079

insufficient, the hybrid cloud solution of private cloud and public cloud is widely used by major enterprises. This article focuses on the workflow task scheduling problem in the hybrid cloud environment, and researches to maximize private cloud profits while meeting task deadline constraints, to save business costs. In a private cloud environment, an optimized Ant Colony Optimization Workflow Scheduling (ACOWS) is proposed in the article for the workflow to complete the execution of tasks within the time limit specified by the user. On this basis, a dynamic multi-workflow scheduling algorithm (Hybrid Cloud Deadline-Constrained Cost Workflows Scheduling, HCDCW) under the hybrid cloud is proposed. The algorithm will be scheduled and executed in the private cloud first. When the task execution time exceeds the task deadline constraint, use the public cloud to schedule part of the workflow. In the experimental phase, the algorithm was verified using the WorkflowSim simulation platform. The experimental results show that compared with traditional hybrid cloud workflow scheduling algorithms, this scheduling algorithm can effectively help enterprises reduce the cost of renting public clouds in the process of using hybrid clouds at different deadlines, and get faster execution time.

Keywords

Cloud Computing, Hybrid Cloud, Scientific Workflow, Ant Colony Algorithm

1. 引言

云计算平台使用分布式计算、虚拟化等技术整合了大量的计算设备及基础设施，并通过网络为用户提供计算、数据存储等资源和相关服务，因其高效、廉价和易维护等特点被广泛应用。此外，它还可以帮助公司和企业在不增加成本的前提下利用高效的处理能力创建大型数据中心。其中混合云架构方案被各大企业广泛应用，混合云是公有云和私有云的混合体，企业需要结合使用私有云和公有云来利用它们所提供的独特优势：即应用工作负载在私有云中到达阈值水平后，突然进入公有云，获取额外的计算资源，利用公有云平台为用户节省了时间和金钱[1]。对于云计算来说，如何管理云计算资源至关重要，特别是当多个服务同时在云上使用它们的应用程序时。因此，对于资源管理，云计算需要一个合适的调度策略。良好的任务调度算法可以优化服务质量参数(QoS)，如最大完工时间、响应时间、吞吐量、资源利用率、任务拒绝率、可靠性、可伸缩性、能耗、执行成本等，并可以在不违反服务级别协议(SLA)的前提下，考虑各类约束，例如截止日期、优先级、经济成本等，实现用户的硬指标约束，同时可避免负载不均衡发生。云工作流调度中的云资源的选择，是一个典型的 NP 困难问题[2]。

在云计算领域，国内外学者关于工作流调度问题做了许多有价值的研究，提出一些静态调度算法[3]。静态调度算法可以分为基于启发式的算法和随机搜索算法。启发式算法根据使用方的主要思路，又分为：1) 将任务首先按照一定规律进行拓扑排序，然后依次调度的列表类启发式算法；2) 将父节点的任务复制到子节点，从而减少数据传递消耗的任务复制算法；3) 将任务划分为多个集群进行调度的集群类调度算法。随机搜索算法则主要指遗传算法、鸟群算法等群体算法，将可能的调度结果作为一个单独的搜索节点组成搜索空间的方式，对调度策略进行选择，从而得到结果。根据调度目标的不同，工作流调度算法可以分为多种类型，最常见的工作流调度目标是如何在最短时间内完成工作流的策略，这也是早期网格计算环境中的工作流调度通常考虑的问题。例如，文献[4]中的最早时间优先 (Earliest Time First, ETF)算

法在每次挑选下一个被执行的任务时，将从所有已经就绪的任务中选出开始时间最早的一个来调度，以保证所有任务的调度时间最短，而每个任务的开始时间，是该任务在所有处理器上运行的最早开始时间。当两个任务的开始时间相同时，算法将选择静态排序更高的任务，从而确保每次都能选出下一执行节点。文献[5]则是通过比较任务在不同处理器上的最早就绪时间、最早开始时间、最早结束时间等，将任务分配到最快完成的可用机器上去。

针对混合云环境下的任务调度，文献[6]针对混合云提出了两种兼顾完工时间和利润的工作流调度算法，第一种是基于截止时间约束的成本优化单目标混合云中的工作流调度算法 DCOH (Deadline-Constrained Cost Optimization for Hybrid Clouds)，在 DCOH 的基础上，提出了一种以最大完工时间和利润为目标的混合云多目标工作流调度优化算法 MOH (Multi-Objective Optimization for Hybrid Clouds)，仿真结果表明，与现有算法相比，DCOH 可以很好地为用户减少使用公有云的花费，MOH 可以提供很好的成本与完工时间协调方案。但是该方案没有考虑到任务执行时间变化带来的影响，因此本文提出一种考虑到任务截止期限的算法，通过蚁群算法[7] (Ant Colony Optimization, ACO)来优化在私有云环境下的工作流调度问题，该算法目标是寻找到最优解进行高效的工作流调度。这种群体智能优化算法的灵感来源于蚂蚁在寻找食物过程中发现路径的行为。当执行任务的时间超过设定的任务截止期限时，使用本文提出的 HCDCW (Hybrid Cloud Deaalindline-Constrained Cost Workflows Scheduling)算法来将部分工作流调度到公有云环境中执行。

2. 混合云工作流系统架构及模型

本节中描述了混合云环境中，工作流的系统架构，工作流的任务模型，混合云中的工作流问题定义。

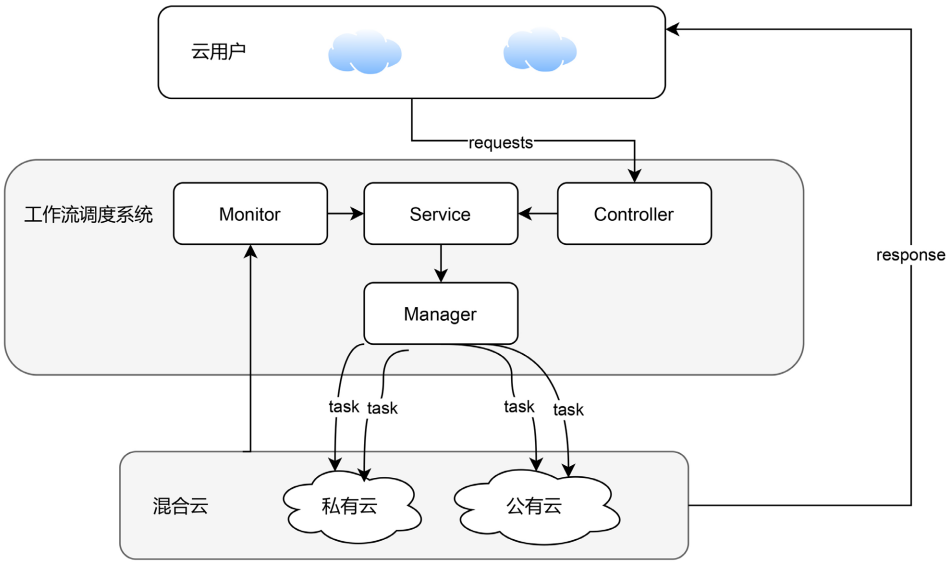

Figure 1. Hybrid cloud workflow system architecture
图 1. 混合云工作流系统架构

2.1. 混合云工作流系统架构

图 1 所示为混合云环境中工作流系统架构[6]。整个系统架构由工作流调度系统、混合云资源组成。在该系统中，由工作流调度系统来接收并判断来自云用户发起的任务请求，当云用户发起的任务连续不断到达时，如果请求被工作流调度系统所接受，则会由工作流调度系统决定云用户的请求是使用私有云

环境中的虚拟机调度还是使用公有云中已经租用的虚拟机进行调度。在工作流调度系统中主要分为四层模型，其中监视器层(Monitor)负责收集并整理云资源中私有云和公有云的 QoS 信息，其中包括云资源的运行状态、响应时间、吞吐量、任务拒绝率等信息。控制层(Controller)负责接收来自云用户的请求，当一个请求到达时会对到达的请求进行过滤，过滤垃圾请求，让系统发起一次合规的工作流调度。接着交由服务层(Service)来处理工作流调度，并根据 Monitor 传来的云资源信息来进行工作流任务的分配，本文所提出的算法既在 Service 层实现调度分配。最后通过 Service 层中所选择的工作流调度算法对工作流任务完成一次调度，与云资源的交互是交由 Manager 层来处理，Service 层不直接与云环境中的虚拟机进行通信，是通过调度 Manager 层所提供的接口来与云虚拟机进行通信。

2.2. 混合云工作流任务模型

独立任务之间彼此没有依赖关系，只需由任务调度器指定到虚拟机即可，而工作流任务存在执行顺序，运行较为复杂。每一个工作流任务，是由一系列的子任务和任务与任务间的数据依赖关系构成，任务与任务之间存在数据上的交互，某些任务需要上一个任务的完成才能继续执行，由于这种依赖关系的复杂性，因此使用有向无环图(Directed Acyclic Graph, DAG)来描述一个工作流任务[8]。图 2 中使用 DAG 图表示一个简单的工作流示例，其中包括了 9 个任务节点，和 12 个依赖关系，节点 1~9 代表了任务节点，边 a-l 代表了依赖关系。

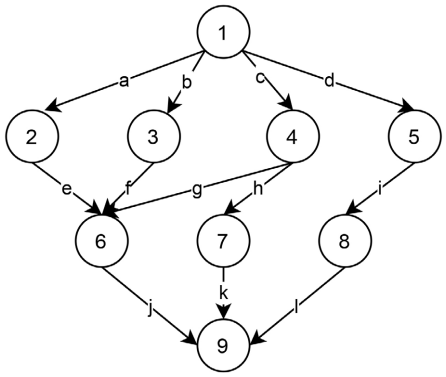

Figure 2. Workflow task model
图 2. 工作流任务模型

通常的工作流模型可以依据 DAG 图来进行建模，$W=(T,E,WD)$ 表示一个工作流任务，$T=\{t_1,t_2,\cdots,t_n\}$ 表示该工作流的子任务集合，$E=\{(t_i,t_j)|t_i,t_j\in T,i\neq j\}$ 表示任务之间的数据依赖关系，工作流 W 使用截止期限来约束，这个截止期限可以描述为 WD (Workflow Deadline)，要求所有的工作流在这个 WD 的截止期限内完成。任务间的依赖关系使用(t_i, t_j)表示，t_j 任务的执行受到 t_i 的限制，需要获取 t_i 执行完成的数据才能使工作流继续执行，任务模型可以描述为$t_i:\{TL\}$，TL (Task Length)表示的在一个虚拟机实例中执行一个工作流任务所需要的执行长度。

2.3. 混合云工作流问题描述

为云用户提供 IaaS 服务的云供应商需要事先告知用户云虚拟机的参数，比如 CPU、RAM、磁盘等信息。其中云服务商提供的虚拟机又可称之为实例，一个虚拟机称之为一个云实例。云实例中的 CPU 计算能力的好坏会决定一个任务在云实例中执行所需要的时间，(t_i, t_j)在这个模型中，两个任务 t_i, t_j 之间需要进行数据传输，耗时主要取决于数据传输之间的带宽。在本文中我们仅考虑 CPU 性能和网络通信带宽

为虚拟机实例之间传输的主要因素。因此，可以使用 cu 来表示 CPU 的计算能力。使用 bw 来表示带宽。

定义 1. 私有云中的资源模型：一个虚拟机实例可以被描述为公式 1：

$$\mathrm{pr}=\{\mathrm{cu},\mathrm{bw}\} \tag{1}$$

可以使用公式 2 来表示所有私有云环境中的虚拟机实例，其中 $\mathrm{r_i}$ 代表私有云中虚拟机实例，n 代表拥有的虚拟机数量：

$$\mathrm{PR}=\{\mathrm{pr_1},\mathrm{pr_2},\cdots,\mathrm{pr_i},\cdots,\mathrm{pr_m}\} \tag{2}$$

定义 2. 公有云中的资源模型：一个虚拟机实例在公有云中可以描述为公式 3，其中 price 表示是租用一个公有云实例所需的花费：

$$\mathrm{pu}=\{\mathrm{cu},\mathrm{bw},\mathrm{price}\} \tag{3}$$

使用公式 4 来表示共有云环境中的虚拟机实例，$\mathrm{pu_j}$ 代表公有云中的虚拟机实例，k 代表所租用的公有云实例的数量：

$$\mathrm{PU}=\{\mathrm{pu_1},\mathrm{pu_2},\cdots,\mathrm{pu_j},\cdots,\mathrm{pu_k}\} \tag{4}$$

定义 3. 混合云中的资源模型：混合云中的资源模型由公有云和私有云共同组成可以被定义为公式 5：

$$\mathrm{H}=\mathrm{PR}\bigcup\mathrm{PU} \tag{5}$$

定义 4. 混合云中的完工时间模型。

私有云中的一个工作流的完工时间可以由两部分组成，其中一部分为一个工作流中的所有任务 t_i 在虚拟机中的完工时间，使用公式 6 来表示工作流任务在虚拟机中执行所需要的时间，使用 I_p 来表示一个任务在一个虚拟机实例中的完工时间：

$$T_{\mathrm{comp}}\left(t_i\right)=\frac{\mathrm{tl}\left(t_i\right)}{\mathrm{cu}\left(I_p\right)},\ I_p\in PR\bigcup PU \tag{6}$$

另一部分为工作流任务中数据传输所需要的时间可以使用公式 7 来表示，其中 $tt\left(t_i,t_j\right)$ 表示的是两个工作流任务之间数据传输成本，V_p 代表一个虚拟机实例当 $V_p=V_q$ 时，指两个任务在同一台虚拟机中完成：

$$T_{\mathrm{comm}}\left(t_i,t_j\right)=\begin{cases}\dfrac{tt\left(t_i,t_j\right)}{\min\left(bw\left(I_p\right),bw\left(I_q\right)\right)}, & V_p\neq V_q\\ 0, & V_p=V_q\end{cases} \tag{7}$$

因此可以使用公式 8 来表示混合云中的任务完成时间其中 HT 表示的是混合云中执行一个工作流任务的总耗时，PT 指的是在公有云中调度任务所花费的时间：

$$\mathrm{HT}=\sum_{i=1,\,j=i+1}^{n}\left(T_{\mathrm{comp}}\left(t_i\right)+T_{\mathrm{comm}}\left(t_i,t_j\right)\right)+\mathrm{PT} \tag{8}$$

定义 5. 混合云中的花费模型：可以用公式 9 表示混合云中的花费，其中 P_v 表示的是租用的每一台共有云所花费的价格：

$$\mathrm{Cost}=\sum_{i=1}^{m}P_v \tag{9}$$

3. 混合云中的多工作流调度策略

3.1. 私有云环境中的多工作流调度算法

由于私有云中的算法无需考虑价格花费，而公有云中的算法需要云用户去花费大量金额去使用。因

此，要最大化私用云的使用效率，本节中，讲诉了一种基于优化 QoS 为目标的 Ant Colony Optimization Workflow Scheduling (ACOWS)。改算法基于优化的蚁群算法，基于蚁群算法的工作流调度优化技术已被用于资源调度优化的问题，因此采用本算法来进行私有云环境中的工作流调度，而传统的蚁群算法根据优化协商过程寻找最佳匹配会陷入局部最优解，因此本算法在执行工作流调度过程中引入一个随机匹配过程，来防止局部最优解，ACOWS 的具体过程为：1) 确定任务需求，生成随机调度策略；2) 基于资源可用性启动多蚂蚁机制；3) 启动协商过程；4) 生成可行的调度方案；(5) 通过 QoS 对调度方案的效果进行评估；6) 生成合适的调度方案；7) 生成资源分配和调度方案。

ACOWS 算法的伪代码如下：

算法 1. Ant Colony Optimization Workflow Scheduling

1) 输入：user task and resources
2) 输出：appropriate task resource mapping
3) t_1, t_2, t_3, t_n user tasks
4) r_1, r_2, r_3..r_k cloud resource(VM)
5) ACO Parameters (α, β, ρ, m, max)
6) Fsrt: compute execution cost and time (feasible solutions)
7) Listrt: temporary storage list (contains feasible solutions)
8) Srt: minimum execution cost and time (best solution)
9) For
10) Input: Tasks t_1, t_2, t_3, t_n
11) check task validity
12) check resource availability r_1,r_2,r_3..r_n
13) Step 1
14) get resource availability
15) for (int i =0; i< n; i++)
16) if (rworkload = 0)
17) move into resource ready list
18) r = r[random() % k];
19) else
20) go to step1
21) end if
22) resource ready list
23) while users tasks not nill do
24) multiple ant mechanisms started
25) compute task execution cost and time on available resource
26) ACO processes
27) Defined ACO Parameters (a, b, p, m, max)
28) While max criteria not meet do
29) Fsrt = for r1, rn/t1, tn
30) fs solution generation process start
31) fs contains estimated execution cost and time
32) test virtual matching up to max value(100)
33) Listrt = Timert (r_1)
34) Srt = Sort (Listrt) //evaluation criteria
35) p (pheromone evaporation) //remove unnecessary solutions
36) select best solution (minimum execution cost and time)
37) mapping = Srt
38) resources dispatching //VMs to PMs
39) resource scheduling r_1t_2,r_2t_3,r_3t_1,r_kt_n
40) execution process starts
41) task execution
42) task execution finish, release resources
43) finish
44) go to step 1
45) end

该算法目标是寻找到最优解进行高效的工作流调度。这种方法是其灵感来源于蚂蚁在寻找食物过程中发现路径的行为。由虚拟机帮助我们生成随机初始值，最后通过优化方案在虚拟机上寻找到最优的调度策略。

所提出的 QRAS 代码的工作原理如下：算法过程从用户任务的输入开始。然后，系统分析器对任务进行验证，同时将其列入可使用的列表。开始寻找空闲的资源，并将其放入可用性列表。由于可用性列表中有大量可用资源，多个蚁群系统启动协商过程。它通过查看所有可能的结果来生成初始解。该过程将继续生成解决方案，直到满足终止条件。这些解决方案存储在临时列表中以供评估。此外，基于 QoS 的评估确定了合理的解决方案，其他则发送到评估列表中。

这些标准化解决方案能够以可承受的成本和时间执行每项任务。此外，已经根据虚拟映射形成虚拟机的放置。它通过在所有的解决方案中找到最佳结果来定义最优调度。然后，任务执行过程开始，将最佳工作负载分配给资源。随着任务执行的结束，资源释放到可用状态。此过程将一直持续到用户完成任务为止。

3.2. 混合云环境中的多工作流调度算法

由于目前企业中大多采用混合云形式来部署任务节点，为了节省企业利润，提升私有云资源利用率，因此要最大化利用企业中的私有云资源，又要保证私有云中工作流调度的执行时长等效率问题，因此使用算法 1 的 ACOWS 算法在私有云环境中优先调度工作流任务，当工作流任务的执行时间超过了所期望的工作流调度时间(WD, Workflow Deadline)，则使用公有云环境调度部分工作流。本节中提出一种混合云环境中的调度算法 Hybrid Cloud Deadline-Constrained Cost Workflows Scheduling (HCDCW)，很好的解决了上述问题。

HCDCW 算法伪代码如下：

算法 2. Hybrid Cloud Deadline-Constrained Cost Workflows Scheduling

1) 输入：user task and resources
2) 输出：appropriate task resource mapping
3) t_1, t_2, t_3, t_n user tasks
4) r_1, r_2, r_3..r_k cloud resource(VM)
5) ACO Parameters (α, β, ρ, m, max)
6) fsrt: compute execution cost and time (feasible solutions)
7) Listrt: temporary storage list (contains feasible solutions)
8) Srt: minimum execution cost and time (best solution)
9) For
10) If (jobs. size ! = 0)
11) Excute ACOWS // 执行算法 1 获取私有云中的执行结果
12) If (excute_time> wd)
13) For each subjob in jobi
14) Send subjob to public cloud // 调度公有云资源
15) End
16) End

4. 实验与分析

本节中为了验证 HCDCW 算法在混合云环境下多工作流执行的有效性、时间成本、费用成本，使用 WorkflowSim [9]模拟器实现了 HCDCW 算法。WorkflowSim 是基于 CloudSim [10]开发而来的云工作流模拟器，在 CloudSim 开源平台上增加了对工作流调度的实现，增加了对工作流的任务聚簇、资源配置、任

务调度等功能。同时 WorkflowSim 它提供真实云环境的功能，如数据中心、虚拟机、用户、Cloudlet 和带宽等，用于配置和定制。WorkflowSim 工具要求使用 JDK 1.6 以上版本，可以使用 Eclipse、NetBeans 或者 Ant 来使用 WorkflowSim。本文模拟实验使用的环境参数为 MacBook Pro (16-inch, 2019)、处理器为 2.6 GHz 六核 Intel Core i7、内存为 32 GB 2667 MHz DDR4、JDK1.8。

本文将构建一个混合云环境，一个云服务厂商针对虚拟机实例提供多种收费机制。常见的收费机制有按需收费、预留收费和竞价收费，主要考虑按需收费机制，即对虚拟机实例按某个单位时间(例如 Google 以分钟计费，AWS、阿里云等以小时计费)支付计算时长费用，如果不满计费时长则按整计算时长收费。本文采用了表 1 的阿里云的计费方式。

Table 1. Types and prices of virtual machine instances billed by Alibaba Cloud on-demand
表 1. 阿里云按需计费的虚拟机实例类型和价格

实例规格	vCPU	内存(GB)	按量(小时)	标准目录月价(￥)
通用型 ecs.g6.large	2	8	0.5	240
通用型 ecs.g6.xlarge	4	16	1.0	480
内存型 ecs.r6.large	4	32	0.66	318
内存型 ecs.r6.xlarge	8	64	1.33	636
计算型 ecs.c6.large	2	4	0.39	187
计算型 ecs.c6.xlarge	4	8	0.78	374

本实验中使用 Montage 数据集[11]，根据仿真环境的配置，这两种数据集是适合于仿真环境的工作流数据，如图 3 所示为 Montage 工作流的形式。

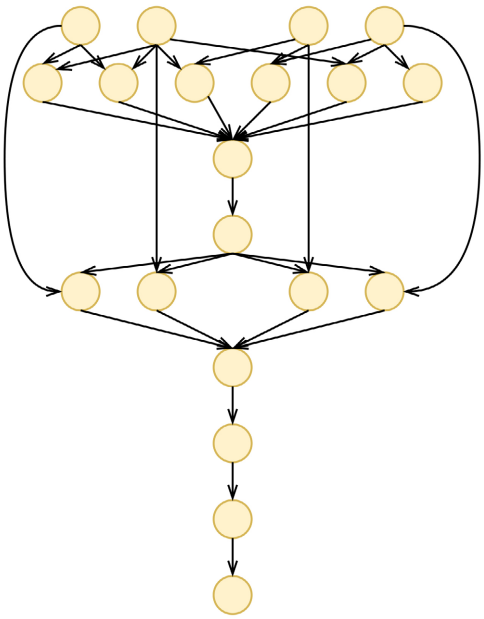

Figure 3. Montage dataset
图 3. Montage 数据集

虚拟机的参数设置为表 2 所示。

Table 2. Virtual machine parameter settings
表 2. 虚拟机参数设置

Datacentersetup
Datacenter 01
Number of hosts 01
VMs configuration
Number of VMs 5
MIPS 1000
Nuner of PEs per VM 1
VM memory. 512
Bandwidth 1000
Workload environment
Number of users 15
Data Set Montage
Numbers of cloudlets 1000
ACO parameters setup
Number of iterations(max) 100
Number of ants(m) 10
α. .3
β. 1
ρ .4

4.1. 执行时间

本节对比了 HCDCW 算法和传统 MINMIN 算法。

图 4 中表示了 100 个工作流任务的时候，私有云执行时间随 wd 最大截止期限的增大而变大。HCDCW 算法的执行时长明显要低于 MINMIN 算法。因为 HCDCW 充分利用私有云资源，减少了私有云中虚拟机等待的空闲时间，提高了资源利用率。

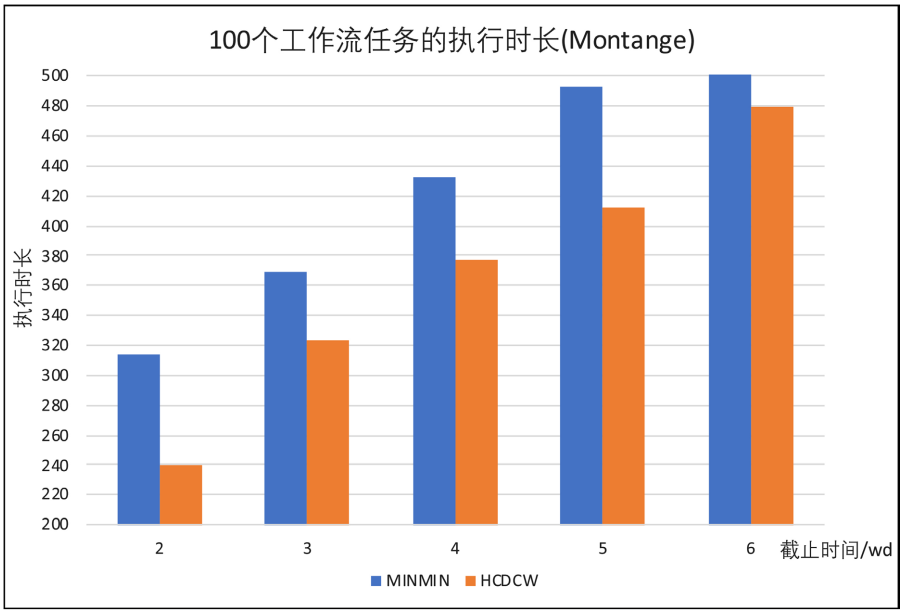

Figure 4. Execution time of 100 workflow tasks
图 4. 100 个工作流任务的执行时间

4.2. 执行费用

本节对比了 HCDCW 算法和经典的 HEFT (Heterogeneous earliest finish time) [3]算法在费用成本上的比较。如图 5 所示，HCDCW 算法相比于经典的 HEFT 算法，在执行花费上节省很多。

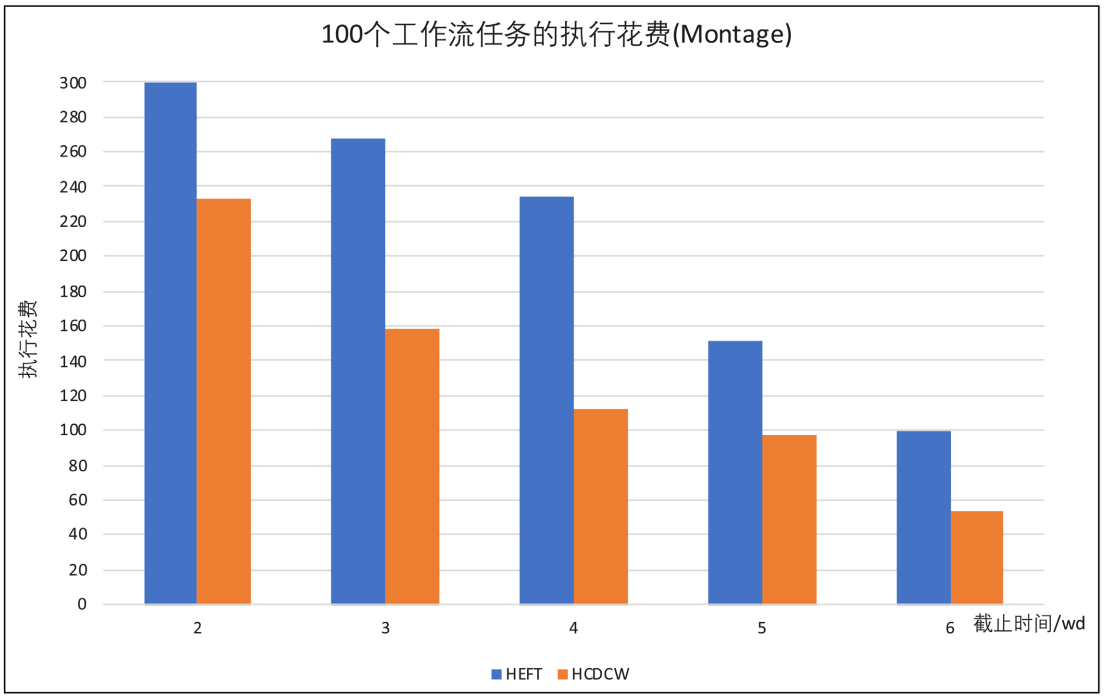

Figure 5. The execution cost of 100 workflow tasks
图 5. 100 个工作流任务的执行花费

5. 结语

本文以优化企业中日益增长的混合云工作流调度为研究目标，提出了混合云环境下动态多工作流优化算法。本文所提出的算法 HCDCW 在使用公有云的情况下为企业降低了费用成本，缩短了任务的执行时间。但由于企业中运行在私有云中的工作流任务，一旦调度到公有云执行，必然存在安全隐私等问题，后续可以考虑如何保证私有云到公有云中的安全传输。

基金项目

课题编号：2018YFB1700702；
课题名称：“互联网+”定制产品概念设计与智能互反馈方法；
计划类别：国家重点研发计划“网络协同制造与智能工厂”重点专项；
承担“互联网+”定制设计要素提取与设计意图理解的研究工作，研发“互联网+”定制产品概念设计与智能互反馈工具 4 个：设计要素提取工具、设计意图理解工具、概念设计案例采集工具、概念设计方案生成工具。

参考文献

[1] Barbosa, F.P. and Charão, A.S. (2012) Impact of *Pay-as-You-Go* Cloud Platforms on Software Pricing and Development: A Review and Case Study. In: Murgante, B., *et al*., Eds., *Computational Science and Its Applications—ICCSA*

2012. *Lecture Notes in Computer Science*, **7336**, 404-417.
https://doi.org/10.1007/978-3-642-31128-4_30

[2] 田倬璟，黄震春，张益农. 云计算环境任务调度方法研究综述[J]. 计算机工程与应用, 2021, 57(2): 1-11.

[3] Topcuoglu, H., Hariri, S. and Wu, M.Y. (2002) Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing. *IEEE Transactions on Parallel and Distributed Systems*, **13**, 260-274.
https://doi.org/10.1109/71.993206

[4] Hwang, J.J., Chow, Y.C., Anger, F.D., *et al.* (1989) Scheduling Precedence Graphs in Systems with Interprocessor Communication Times. *SIAM Journal on Computing*, **18**, 244-257. https://doi.org/10.1137/0218016

[5] Lin, C. and Lu, S.Y. (2011) Scheduling Scientific Workflows Elastically for Cloud Computing. 2011 *IEEE 4th International Conference on Cloud Computing*, **2011**, 746-747.
https://doi.org/10.1109/CLOUD.2011.110

[6] Zhou, J.L., Wang, T. and Cong, P.J. (2019) Cost and Makespan-Aware Workflow Scheduling in Hybrid Clouds. *Journal of Systems Architecture*, **100**, Article No. 101631. https://doi.org/10.1016/j.sysarc.2019.08.004

[7] Ritchie, G. and Levine, J. (2003) A Fast, Effective Local Search for Scheduling Independent Jobs in Heterogeneous Computing Environments. *Proceedings of the 22nd Workshop of the UK Planning and Scheduling Special Interest Group*, Article ID: 15882331.
https://www.semanticscholar.org/paper/A-fast%2C-effective-local-search-for-scheduling-jobs-Ritchie-Levine/cd4153cd3906897827cc2b537211c004ff95e89c

[8] Mo, L., Kritikakou, A. and Sentieys, O. (2019) Approximation-Aware Task Deployment on Asymmetric Multicore Processors. 2019 *Design, Automation & Test in Europe Conference & Exhibition* (*DATE*), **2019**, 1492-1497.
https://doi.org/10.23919/DATE.2019.8715077

[9] Chen, W. and Deelman, E. (2012) WorkflowSim: A Toolkit for Simulating Scientific Workflows in Distributed Environments. 2012 *IEEE 8th International Conference on E-Science*, **2012**, 1-8.
https://doi.org/10.1109/eScience.2012.6404430

[10] Calheiros, R.N., Ranjan, R., Beloglazov, A., *et al.* (2011) CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. *Software*: *Practice and Experience*, **41**, 23-50. https://doi.org/10.1002/spe.995

[11] Bharathi, S., Chervenak, A., Deelman, E., Mehta, G., Su, M. and Vahi, K. (2008) Characterization of Scientific Workflows. 2008 *Third Workshop on Workflows in Support of Large-Scale Science*, **2008**, 1-10.
https://doi.org/10.1109/WORKS.2008.4723958