

· 计算机工程与应用 ·

## 基于资源分组的多约束云 workflow 调度算法

陈爱国, 王 玲, 任金胜, 罗光春

(电子科技大学计算机科学与工程学院 成都 611731)

**【摘要】**已有的云 workflow 调度算法采用全局搜索方式进行资源选取, 存在计算成本高、对大规模云系统适应性差的问题。该文提出了基于资源分组的多约束云 workflow 调度算法, 采用有向无环图的方法, 对云 workflow 中的多任务之间的执行顺序和数据交换等属性进行量化建模; 使用模糊聚类方法实现基于资源多维特征的分组处理, 降低 workflow 任务到资源匹配过程中的搜索空间; 并引入执行时间和成本预算约束, 将 workflow 的任务调度问题转化为有约束条件的极小极大问题进行快速求解。仿真测试表明, 该算法显著降低了任务执行完成时间和成本。

**关键词** 云计算; 云 workflow; 模糊聚类; 资源分配; 调度算法

中图分类号 TP311 文献标志码 A doi:10.3969/j.issn.1001-0548.2017.03.014

## Multi-Constrained Scheduling Algorithm of Cloud Workflow Based on Resource Grouping

CHEN Ai-guo, WANG Ling, REN Jin-sheng, and LUO Guang-chun

(School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 611731)

**Abstract** The existing cloud workflow scheduling algorithms, using the global search for resource selection, exist a high computational cost and poor adaptability for large-scale cloud systems. Aimed at solving these problem, a multi-constrained cloud workflow scheduling algorithm based on resource grouping is proposed in this paper. It uses the direct acyclic graph to model the multi-task in cloud workflow and characterize the execution sequences and data transfer requirement between tasks with the DAG's node and edge's attributes. Then, fuzzy clustering method is employed to classify resources based on multidimensional features and reduce the computational load from workflow tasks to resource selection. By introducing execution time and cost budget constraints, the proposed algorithm transforms the scheduling problem into a minimax problem. Simulation results show that our algorithm significantly reduces the task execution time and cost.

**Key words** cloud computing; cloud workflow; scheduling algorithms; resource allocation; fuzzy clustering

云计算模式中, 云数据中心具有大规模硬件资源, 通过网络服务的方式向多用户提供强大的计算能力。因此, 云计算平台中面临大量动态的计算任务请求。这些特点使得云计算环境中的任务调度比传统分布式环境中的任务调度要面临更多更复杂的问题<sup>[1]</sup>。特别是随着网络功能虚拟化(network function virtualization, NFV)等技术的发展, 云环境中的 workflow 任务越来越多, 其调度成为云计算领域的一个研究热点<sup>[2-3]</sup>。并且, 云环境中用户按资源使用进行付费, 如何权衡性能和成本成为一个普遍问题。越来越多的混合云环境下存在更复杂的资源使

用成本和性能规格。这些都给云 workflow 调度提出了更多的挑战<sup>[3]</sup>。典型的需求场景如公安系统需要对跨地域分布存储的多源数据进行多层关联分析。为了提高计算效率, 完整业务被设计成多个计算步骤, 构成了 workflow 任务。各个任务节点之间有先后关系和数据交换, 通常单个任务节点还需要加载存量历史数据。在可以实现就近资源征用的基础上, 如何从大规模资源中为完整 workflow 计算任务选择合适的资源、降低整体计算成本并提高响应时间具有重要的应用价值。

本文主要针对云 workflow 调度中如何实现时间和

收稿日期: 2016-12-26; 修回日期: 2017-03-21

基金项目: 四川省科技支撑计划(2016GZ0075, 2016GZ0077); 四川省科技厅国际合作项目(2017HH0075)

作者简介: 陈爱国(1981-), 男, 博士, 副教授, 主要从事云计算、信息物理融合系统方面的研究。

成本的约束问题开展算法研究。满足时间约束是指确保 workflow 在用户指定截止时间前完成, 成本约束则要求 workflow 的执行要在一定的成本预算范围内。造成任务整体调度时间和成本提高的原因多种多样, 包括 workflow 中多任务的调度顺序、资源规模以及算法本身的执行效率等。针对已有的 workflow 调度算法存在的搜索成本高等问题, 本文提出了基于资源分组的多约束云 workflow 调度算法, 利用模糊聚类的方法对云资源进行预处理, 并将列表调度和任务调度机制相结合, 在为任务搜索合适的资源时, 降低搜索成本。同时为了实现任务调度过程中最小化完成时间和最小化执行成本这两个目标, 采用有向无环图方法求解资源分配策略。该算法在资源聚类的基础上对 workflow 中的任务进行调度, 考虑任务执行的时间与成本预算, 有效地降低任务执行时间和执行成本, 实现了最小化完成时间和最小化执行成本两个目标。

## 1 相关研究

关于云调度算法的研究有很多, 包括基于列表的调度策略的最小完成时间算法、Sufferage 算法、Min-min 算法和 Max-min 算法等算法<sup>[4-6]</sup>。典型的处理器关键路径算法(critical path on processor, CPOP)<sup>[7]</sup>和异构最早完成时间算法(heterogeneous earliest finish time, HEFT)<sup>[8]</sup>都是基于列表的启发式算法, 其中 HEFT 就完成时间而言优于上述所有的算法。HEFT 算法对任务划分优先级, 对优先级较高的优先为其分配处理资源, 在任务调度最小化最早完成时间上取得了良好的效果。但是, HEFT 算法在任务调度过程中并没有考虑成本的因素。

文献[9]详细描述了云计算环境下任务调度中成本和截止时间约束的重要性, 并提出了相应的算法。但是, 所提出的资源模型和算法仅针对同类资源。文献[10]提出了云环境下分段式 workflow 调度算法: 一阶段算法 IC-PCP 和两阶段算法 IC-PCPD2。这两个阶段算法在调度大的 workflow 时都有一个多项式时间复杂度, 并在最后期限约束下最小化 workflow 执行成本。文献[11]也提出了一个算法集来调度截止时间约束的任务以最小化成本执行, 但是这同样只最小化了一个目标。所有这些启发式算法的主要问题是, 他们中大多数解决的是计算环境中的单目标优化的问题, 不能同时针对计算成本和执行时间进行优化。

文献[12]提出了 BHEFT 算法, 充分考虑了用户对成本预算和截止时间的要求。但在进行资源分配

时, 采用全局资源搜索方法造成了过高的时间成本和额外计算成本。特别是当云计算环境资源规模较大时, 全局搜索计算的搜索空间也在攀升。其他研究成果也存在类似问题。

如何在为任务分配合适资源的同时, 降低资源搜索空间, 节约搜索时间成本和执行成本, 是 workflow 调度中一个需要进一步研究的问题。基于上述分析, 为了解决在任务调度之初由于搜索空间比较大造成的搜索时间和执行成本的问题, 本文提出利用模糊聚类的方法对云资源根据资源特征属性进行分类的策略。

## 2 问题模型

云 workflow 是由一系列关联的任务构成的, 将 workflow 调度转换成具有依赖关系的任务拓扑, 而 workflow 调度的核心就是计算任务到资源的映射策略。调度目标主要关注两方面: 1) 从任务提交到任务执行完成所花费的时间最小化, 即最小化完成时间; 2) 从任务提交到任务执行完成所花费的成本最小化, 即最小化执行成本。本文着重讨论基于资源预处理的 workflow 调度模型。假设向云计算系统提交一个 workflow WF, 其包含  $n$  个任务, 系统内存在  $m$  个空闲资源。其中, 对于提交的  $n$  个任务, 用户规定了成本预算  $B$  和任务执行完成时间限制  $D$ 。

workflow 任务调度问题可以描述为: 将  $n$  个任务分配到  $m$  个资源调度执行, 并确保在时间限制  $D$  范围内取得任务的最小完成时间, 在预算  $B$  范围内取得任务的最小执行成本。若预测任务执行完成后, 最小完成时间大于时间限制  $D$  或最小执行成本大于预算  $B$ , 则任务调度失败。如何将任务分配给恰当的资源, 以取得执行时间和执行成本的最小化, 问题定义如下。

**定义 1** 用户提交的包含  $n$  个任务的 workflow WF, 用有向无环图(direct acyclic graph, DAG)描述为:  $WF = (T, E)$ , 其中  $T = \{t_1, t_2, \dots, t_n\}$  代表任务集合, 集合中有  $n$  个任务。  $E = \{e_{ij}\}$  为边集, 代表任务之间的依赖关系, 若  $e_{ij} = 0$  代表任务  $t_i$  和  $t_j$  之间并不存在执行顺序的先后关系; 若  $e_{ij} \neq 0$  代表任务  $t_i$  执行完成后, 任务  $t_j$  才可以执行,  $e_{ij}$  的值代表了任务  $t_i$  和  $t_j$  之间需要传输的数据量。典型 workflow 具有单一的首节点  $t_1$  和尾节点  $t_n$ , 任务  $t_2 \sim t_{n-1}$  可泛化为多个任务路径  $P = \{p_1, p_2, \dots, p_L\}$ 。

**定义 2** 云计算系统中存在的可供调度的资源用  $R = \{r_1, r_2, \dots, r_m\}$  表示,  $m$  为资源的总数。

**定义 3** 矩阵  $W = TR$  是任务执行时间代价矩

阵,其中 $w_{ij}$ 代表任务 $t_i$ 在资源 $r_j$ 上执行花费的时间。

**定义 4** 矩阵 $C = R^2$ 是衡量两个资源之间的通信能力即数据传输能力的矩阵,其中 $c_{ij}$ 代表资源 $r_i$ 和资源 $r_j$ 之间的传输带宽,即数据传输能力。

**定义 5** 任务资源分配策略定义为 $X$ 。对于给定的任务集合 $T$ 和资源集合 $R$ ,分配策略为:

$$X(n, m) = \{x_{11}, x_{12}, \dots, x_{1m}, x_{21}, x_{22}, \dots, x_{ik}, \dots, x_{nm}\} \quad (1)$$

式中, $x_{ij}$ 取值为0或1,表示是否将任务 $t_i$ 分配给资源 $r_j$ 。所有任务的总执行时间定义为:

$$\text{TET} = \sum_{i=1}^n \sum_{j=1}^m x_{ij} w_{ij} \quad (2)$$

式中, $x_{ij} w_{ij}$ 表示任务 $t_i$ 分配给资源 $r_j$ 的执行时间。

所有任务的总执行成本cost为:

$$\text{cost} = \sum_{i=1}^n \sum_{j=1}^m x_{ij} w_{ij} \text{price}_j \quad (3)$$

式中, $\text{price}_j$ 表示分配给特定资源 $r_j$ 的单位时间定价。

根据任务分配策略 $X$ ,将任务 $t_i$ 分配给资源 $r_k$ 执行,任务 $t_j$ 分配给资源 $r_l$ 执行。对于所有的任务,其总数据传输时间为:

$$\text{DTT} = \sum_{i=1}^n \sum_{j=1}^n \frac{e_{ij}}{c_{kl}} \quad (4)$$

根据式(2)和式(4),可以得出所有任务执行完成时间为:

$$\text{makespan} = S + \text{TET} + \text{DTT} \quad (5)$$

式中, $S$ 表示资源搜索时间。

根据以上分析,任务资源分配模型以数学的形式表示为:对于给定的任务集合 $T$ 和资源集合 $R$ ,如何找到一个最优的分配策略,使得任务的完成时间和执行成本最小。根据式(3)和式(5),目标函数定义为:

$$\text{func} = \text{cost} + \text{makespan} \quad (6)$$

定义式(3)~式(6)的目的是为了准确刻画成本,用于最小化求解,其约束条件为:

$$\text{cost} \leq B \text{ and } \text{makespan} \leq D \quad (7)$$

约束条件表明,在成本最小化、完成时间最小化的同时,不能超过预算限制和时间限制。根据式(3)、式(5)和式(6),影响优化目标的主要因素是式(5)中的资源搜索时间 $S$ 、所有任务的执行完成时间 $\text{TET}$ 和所有的任务的总数据传输时间 $\text{DTT}$ 受调度算法本身的影响。因此如何减小 $S$ 、 $\text{TET}$ 、 $\text{DTT}$ 是调度算法的首要考虑因素。

### 3 基于资源分组 workflow 调度算法

针对已有算法的不足,本算法首先进行资源分组预处理,通过选择最优分组解决全局资源搜索成本的问题。并定义了时间和成本约束量化模型,最小化任务执行时间和成本。

#### 3.1 资源分组处理

分组处理将大量各种类型的云计算资源根据其特征和对任务的支持能力划分为若干小类,以解决资源搜索空间大而带来的效率低下、资源浪费、时间复杂度高等问题。

模糊聚类主要是对资源特征矩阵进行处理。完整的模糊聚类过程首先需要采集云计算资源的特征数据,根据资源特征属性,建立每个资源的多重属性特征初始矩阵。由于每个资源的特征属性的数量级与量纲存在差异,需要对初始数据矩阵进行适当的变换,即数据归一化操作,将标准矩阵中的值限定在0~1之间,方便后续的聚类划分。

本文使用 $S = \{s_1, s_2, \dots, s_M\}$ 表示需要被聚类的资源集合,其中 $M$ 表示资源个数;使用向量 $S'(s_i) = \{s_{i1}, s_{i2}, \dots, s_{iV}\} (i=1, 2, \dots, M)$ 表示第 $i$ 个资源的特征属性,其中 $s_{ij}$ 表示第 $i$ 个资源在第 $j$ 个特征属性上的度量值, $V$ 表示特征属性个数。

特征属性是指面向任务处理能力的云计算资源特征或性质,主要包括计算能力、传输能力、内存大小、存储能力、网络位置、连接数等特征。确定特征属性后,再依据特征属性对云计算资源进行聚类,将具有相似特性的资源归到一起。针对不同的任务需求,特征属性的选择会有所不同。结合混合云场景中的典型业务场景,本算法考虑下面4个特征属性:

1)  $c_1$ 代表计算能力属性,表示资源的平均计算能力,即资源节点的计算性能。性能的高低直接关系到任务的执行时间,对任务调度过程中的时间成本有着至关重要的影响。

2)  $c_2$ 代表计算节点间的数据传输能力属性,表示云资源节点间的联通边权值的平均值,即连接链路的通信能力的平均值,描述了资源与资源间的数据传输性能,数据传输性能的高低直接影响数据传输的时间成本。

3)  $c_3$ 表示计算节点与历史数据资源节点间的数据传输能力属性。在跨地域的混合云中,数据资源节点到不同混合云计算资源节点间的数据传输能力差异交大。该能力主要体现为对历史数据的传输加

载时间需求。

4)  $c_4$  代表连接数属性, 表示连接到该资源节点的其他资源的数目。分派到该资源的任务可以与其他多个资源的任务进行数据传输。

模糊聚类资源分组处理分为3个步骤:

#### 1) 数据标准化处理

此步骤主要是消除数据量纲差异。对于原始数据表  $S'$ , 使用平均值和标准差来处理目标系统中的数据从而获得标准化数据, 每个数据的标准化值为:

$$p'_{ik} = (p_{ik} - \bar{c}_k) / S_{ck} \quad (8)$$

式中,  $c_k$  代表原始数据中第  $k$  个特征向量,  $\bar{c}_k$  是  $c_k$  的平均值;  $S_{ck}$  是  $c_k$  的标准化方差。由于标准化值  $p'_{ik}$  并非全属于  $[0,1]$ , 采用极差标准化方法将  $p'_{ik}$  转化得到  $p''_{ik}$ 。极差标准化方法定义如下:

$$p''_{ik} = (p'_{ik} - p'_{k\min}) / (p'_{k\max} - p'_{k\min}) \quad (9)$$

式中,  $p'_{k\min}$  是最小值;  $p'_{k\max}$  是  $p'_{1k}, p'_{2k}, \dots, p'_{Nk}$  的最大值。

#### 2) 模糊矩阵实现

此步骤的主要目的是便于后续根据云计算资源特征属性的相似性对资源进行聚类。利用指数相似系数法可以计算处理单元  $S = \{s_1, s_2, \dots, s_N\}$  的模糊相似关系  $R_s$ 。经过测试, 指数相似系数法具有更好的效果, 计算方法如下:

$$r_{ij} = \frac{1}{n} \sum e^{-\frac{3(p_{ik} - p_{jk})^2}{4 S_{ck}^2}} \quad (10)$$

式中,  $S_{ck}^2$  是第  $k$  个特征向量的方差;  $r_{ij}$  为处理单元  $p_i$  和  $p_j$  的相似度。通过相似系数矩阵的合成运算得到具有传递闭包的模糊等价关系  $R_e$ 。

#### 3) 聚类信息评估

采用基于割集的聚类方法, 可以得到一组聚类结果, 用  $CLUSTER = \{CL_1, CL_2, \dots, CL_K\}$  表示,  $CL_j$  表示第  $j$  个资源群组。每个群组的整体性能表示为:

$$PERF(CL_j) = \frac{1}{w} \sum_{p_w \in CL_j} \sum_{i=1}^n \alpha_i P^N[k][i] \quad (11)$$

式中,  $w$  代表群组中处理单元的个数;  $\alpha_i$  代表处理单元第  $i$  个特征向量的权重, 它的值一般可以通过历史数据或经验获得。通过对每个群组的整体性能的计算, 根据性能高低重新排序。任务调度时, 优选排序靠前的资源群组。

### 3.2 时间成本约束定义

在问题模型描述和资源属性建模的基础上, 进一步引入 workflow 的任务时间约束和成本预算约束量化方法, 并通过算法模型实现完成时间和执行成本

这两个目标的权衡优化。

#### 3.2.1 执行时间约束

用户提交 workflow 给云平台执行, 并指定期望的总体执行完成时间  $D$ 。在调度过程中, 调度粒度落到 workflow 中的任务层面, 需要将执行时间约束加入到任务调度选择条件中。为此, 定义以下3个变量:

1) 工作流剩余执行时间要求 SWD: 描述 workflow 中尚未执行任务的执行时间要求。

2) 执行时间适应因子 DAF: 用于调整当前任务的执行时间限制。

3) 当前任务执行时间 CTD: 描述当前任务的执行时间限制。

对于任务  $t_i$ , 其剩余截止时间为:

$$SWD_i = D - \sum_{t_k \in \text{pred}(t_i)} ET(t_k) \quad (12)$$

任务的执行时间  $ET(t_k)$  根据资源分配策略  $X$  计算如下:

$$ET(t_k) = \sum_{j=1}^m x_{kj} w_{kj} \quad (13)$$

根据式(13), 任务  $t_i$  的截止时间适应因子为:

$$DFA_i = \begin{cases} \overline{ET(i)} / \sum_{k=1}^n \overline{ET(k)} & SWD_i \geq 0 \\ 0 & SWD_i < 0 \end{cases} \quad (14)$$

任务  $t_k$  的平均执行时间  $\overline{ET(k)}$  为任务  $t_k$  在所有资源上的执行时间的平均值:

$$\overline{ET(k)} = \sum_{j=1}^m w_{kj} / m \quad (15)$$

根据式(12)~式(15),  $t_i$  的当前任务执行时间为:

$$CTD_i = SWD_i \times DFA_i \quad (16)$$

#### 3.2.2 执行成本约束

前面问题模型描述中约定 workflow 规定的成本预算为  $B$ 。根据对 workflow 执行总成本的要求, 需要将成本约束加入到任务选择条件中。为此, 引入以下3个变量。

1) 剩余成本预算 SWB: 描述未执行的任务的成本限制。

2) 成本预算适应因子 BAF: 用于调整当前任务的预算限制。

3) 当前任务成本预算 CTB: 描述当前任务的执行成本限制。

对于任务  $t_i$ , 其剩余成本预算为:

$$SWB_i = B - \sum_{t_k \in \text{pred}(t_i)} EC(t_k) \quad (17)$$

执行成本  $EC(t_k)$  为:

$$EC(t_k) = \sum_{j=1}^m x_{kj} w_{kj} \text{price}_j \quad (18)$$

任务  $t_i$  的成本预算适应因子为:

$$\text{BAF}_i = \begin{cases} \overline{EC(i)} / \sum_{k=1}^n \overline{EC(k)} & \text{SWB}_i \geq 0 \\ 0 & \text{SWB}_i < 0 \end{cases} \quad (19)$$

任务  $t_k$  的平均执行成本  $\overline{EC(k)}$  为任务  $t_k$  在所有资源上的执行成本的平均值:

$$\overline{EC(k)} = \sum_{j=1}^m w_{kj} \text{price}_j / m \quad (20)$$

根据式(17)~式(20), 任务  $t_i$  的当前任务预算为:

$$\text{CTB}_i = \text{SWB}_i \times \text{BAF}_i \quad (21)$$

### 3.2.3 约束条件引入

基于任务  $t_i$  对执行时间和成本预算的要求, 构建可为任务  $t_i$  分配的资源集合  $\text{BS}_i$ 。通过预测任务  $t_i$  在资源  $r_p$  所需要的执行成本与执行时间是否满足条件  $EC(i, p) \leq \text{CTB}_i$  和  $ET(i, p) \leq \text{CTD}_i$ , 将满足条件的资源保留下来, 构成可为任务  $t_i$  分配的资源集合  $\text{CL}_{j,i}$ , 表示为:

$$\begin{aligned} \text{CL}_{j,i} &= \{r_p \mid \exists r_p \in \text{CL}_j \text{ and } \text{CL}_j \in \text{CLUSTER} \\ EC(i, p) &\leq \text{CTB}_i \text{ and } ET(i, p) \leq \text{CTD}_i\} \end{aligned} \quad (22)$$

集合  $\text{BS}_i$  表示所有  $\text{CL}_{j,i}$  中的资源所构成的可分配资源集合:

$$\begin{aligned} \text{BS}_i &= \{r_p \mid \exists r_p \in \{\text{CL}_{j,i} \mid \text{CL}_j \in \text{CLUSTER} \text{ and} \\ j &\in [0, |\text{CLUSTER}|]\} \} \end{aligned} \quad (23)$$

在进行任务调度时, 从  $\text{BS}_i$  中选择合适的资源分配给任务。

## 3.3 调度过程

### 3.3.1 调度准备

调度准备阶段的工作主要是为任务级调度做准备。包含根据优先级构造任务列表、资源聚类、依据 workflow 总体时间和成本约束量化任务粒度的时间和成本约束、构造可分配资源集合等操作。这些操作节省了搜索资源空间所花费的时间, 同时也降低了后续任务调度执行的时间成本和预算成本。

### 3.3.2 任务-资源求解与分配

基于3.1和3.2节描述的算法模型, 优化求解任务-资源分配策略。引入时间成本平衡因子, 以平衡用户对时间和成本的要求定义如下分配规则。

1) 如果  $\text{BS}_i \neq \emptyset$ , 使得最小的资源优先被选择, 即优化函数:

$$F_i = \min \{ \alpha ET(i, j) + (1 - \alpha) EC(i, j) \} \quad j \in \text{BS}_i \quad (24)$$

式中,  $\alpha$  是时间成本平衡因子, 取值范围为  $[0, 1]$ , 代表了用户偏好的执行时间和执行成本敏感度。如果对于任务  $t_i$  的多个备选资源节点, 度量值  $F_i$  相等, 则优选任务  $t_i$  前序任务所在的资源节点。

2) 如果  $\text{BS}_i = \emptyset$  且  $\text{SWB} \geq 0$ , 所有的可用资源只要满足式(24), 均可以被选择。因为即使局部节点的可用资源不满足约束条件, 但完整工作流的约束也有可能是满足的。

3) 如果  $\text{BS}_i = \emptyset$  且  $\text{SWB} < 0$ ,  $\text{SWD} < 0$ , 则选择可用资源中价格最小整体性能最高的资源。

完整 workflow 中多个任务路径  $P$  可以并行化执行, 各个路径的执行时间和执行成本分别用  $ET(p_i)$  和  $EC(p_i)$  表示, 用路径上各个任务的执行时间和执行成本求和得到。考虑到路径是可以并行执行的, 因此最优化函数为:

$$\text{PF} = \min_{i=1}^L \text{Max} \{ \alpha EC(p_i) \} + \text{Max}_{i=1}^L \{ (1 - \alpha) ET(p_i) \} \quad (25)$$

则 workflow  $\text{WF}$  的优化求解目标为:

$$\text{WFF} = F_1 + \text{PF} + F_m \quad (26)$$

通过将问题建立成数学模型, 将 workflow 的任务调度问题转化为有约束条件的极小极大问题的求解。该目标函数的最优化结果是由多个任务构成的调度策略, 通过求解该目标函数的最优值, 可得到相应的最优化的资源分配策略, 也就是满足用户提交的工作流任务的最优化调度策略。

## 4 实验与分析

基于云计算仿真软件 CloudSim 开发了实验系统, 核心扩展了 workflow 调度模块。通过实验对比分析了本文提出的算法与 HEFT 算法<sup>[8]</sup>、BHEFT 算法<sup>[12]</sup> 在运行时间和成本方面的性能。设置任务节点数分别为 10、20、50、100、150、200、250、300 的情况下, 分别运行 20 次的模拟实验, 并取得实验结果的平均值进行对比分析。

### 4.1 执行时间对比

根据文献[13], 选择正常调度长度(normalized schedule length, NSL)作为评价算法在调度任务时的时间性能指标, 正常调度长度 NSL 计算如下:

$$\text{NSL} = \frac{\text{total execution time}}{T_{\text{fast}}} \quad (27)$$

式中,  $T_{\text{fast}}$  是 workflow 在最快的资源上执行所有任务的执行时间。NSL 的值越大, 表示任务执行的时间越长。本算法与 BHEFT 算法、HEFT 算法在运行时间方面的性能对比结果如图 1 所示。

可以看出, 随着任务数量的增加, 3个算法的NSL值均呈上升趋势。但是, 相比于BHEFT算法和HEFT算法, 本算法仍然具有相对较小的NSL值; 并且随着任务数量的增加, 在资源够用的情况下本算法的NSL值上升趋势并不明显。这是由于本算法根据 workflow 任务属性, 首先进行了资源分组处理策略, 有效减少了资源搜索空间, 从而降低了资源搜索时间; 其次, 将 workflow 全局资源调度问题转化成了极小极大问题, 可以通过数学公式求解得到优化的资源分配策略。

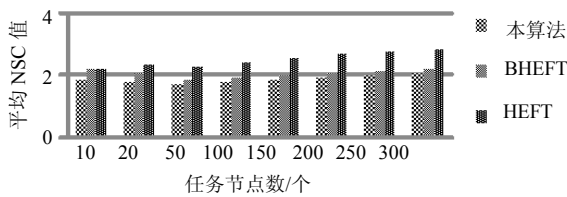


图1 不同算法在不同任务数量下的NSL值

#### 4.2 执行成本对比

根据文献[14], 选择正常调度成本(normalized schedule cost, NSC)作为评价算法在调度任务时的成本性能指标, 计算如下:

$$NSC = \frac{\text{total cost}}{C_{\text{onCheap}}} \quad (28)$$

式中,  $C_{\text{onCheap}}$  是所有任务在最廉价资源上的执行成本。NSC的值越大, 表示耗费的成本越大。本算法与BHEFT算法和HEFT算法在运行成本方面的性能对比结果图2所示。

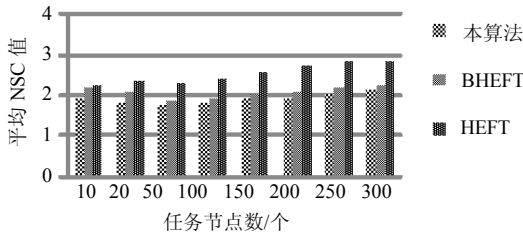


图2 不同算法在不同任务数量下的NSC值

可以看出, 在对 workflow 任务进行调度时, 本算法在成本方面的调度性能明显优于HEFT算法, 略优于BHEFT算法。这是由于本算法引入了成本约束的限制条件, 有效降低了任务调度的成本。

通过上面两个实验, 可以得出在相同的截止时间和预算的约束下, 使用相同的定价模型, 在降低执行成本和完成时间上, 本算法比其他两个算法优越。

#### 4.3 加速比对比

算法执行加速比Speedup<sup>[15]</sup>表示所有任务的最

小顺序执行时间之和与实际完工时间的比值, Speedup计算如下:

$$\text{Speedup} = \frac{\min_{r_j \in R} \left\{ \sum_{t_i \in T} ET(i, j) \right\}}{\text{makespan}} \quad (29)$$

它是衡量算法性能的一个指标, 值越大, 表示算法性能越好。本算法与BHEFT算法和HEFT算法在加速比方面的性能对比结果如图3所示。

由图可知, 在不同的任务数量下, 本算法比其他两个算法具有更好的加速性能, 运行执行效率更快。因为本算法采用全局策略一次性求解方法, 并且本算法构建的策略求解模型计算简单。

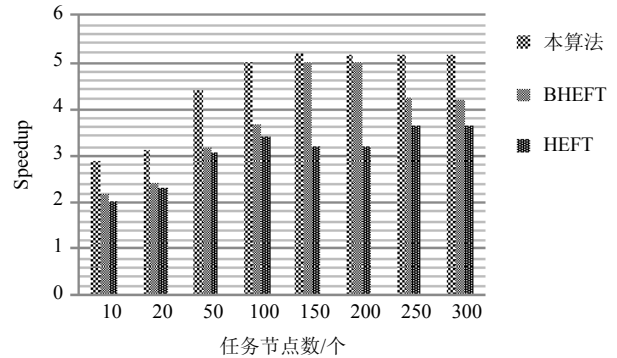


图3 不同的任务数量下算法的Speedup性能

## 5 结束语

本文分析了现有云 workflow 调度模型存在的搜索资源占用时间长, 造成算法执行时间与执行成本增加的问题。提出了基于资源分组的多约束云任务调度算法, 该算法将调度过程分解为资源划分和任务调度策略求解两方面, 首先提出基于模糊聚类的资源分组处理方法; 然后考虑到任务的成本和执行时间约束, 提出了有约束条件的极小极大策略求解算法, 并根据资源分配规则对任务进行调度。未来研究中, 将考虑加入可靠性约束条件, 并以运营的云平台为对象进行验证。

### 参 考 文 献

- [1] FANG Y, WANG F, GE J. A task scheduling algorithm based on load balancing in cloud computing[C]//Web Information Systems and Mining. Sanya, China: Springer, 2010: 271-277.
- [2] VILALTA R, MAYORAL A, MUÑOZ R, et al. The SDN/NFV cloud computing platform and transport network of the ADRENALINE testbed[C]//2015 1st IEEE Conference on Network Softwarization (NetSoft). [S.l.]: IEEE, 2015: 1-5.
- [3] 陈超. 改进CS算法结合决策树的云 workflow 调度[J]. 电子科技大学学报, 2016, 45(6): 974-980.

- CHEN Chao. Workflow task scheduling in cloud computing based on hybrid improved CS algorithm and decision tree[J]. Journal of University of Electronic Science and Technology of China, 2016, 45(6): 974-980.
- [4] MATHEW T, SEKARAN K C, JOSE J. Study and analysis of various task scheduling algorithms in the cloud computing environment[C]//2014 International Conference on Advances in Computing, Communications and Informatics. [S.l.]: IEEE, 2014: 658-664.
- [5] GUO L, ZHAO S, SHEN S, et al. Task scheduling optimization in cloud computing based on heuristic algorithm[J]. Journal of Networks, 2012, 7(3): 547-553.
- [6] VARALAKSHMI P, RAMASWAMY A, BALASUBRAMANIAN A, et al. An optimal workflow based scheduling and resource allocation in cloud[C]//Advances in Computing and Communications. [S.l.]: Springer, 2011: 411-420.
- [7] TOPCUOGLU H, HARIRI S, WU M. Performance-effective and low-complexity task scheduling for heterogeneous computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(3): 260-274.
- [8] ARABNEJAD H, BARBOSA J G. List scheduling algorithm for heterogeneous systems by an optimistic cost table[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(3): 682-694.
- [9] MALAWSKI M, JUVE G, DEELMAN E, et al. Cost-and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds[C]//Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. [S.l.]: IEEE Computer Society, 2012.
- [10] ABRISHAMI S, NAGHIBZADEH M, EPEMA D H J. Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds[J]. Future Generation Computer Systems, 2013, 29(1): 158-169.
- [11] VAN D B R, VANMECHELEN K, BROECKHOVE J. Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds[J]. Future Generation Computer Systems, 2013, 29(4): 973-985.
- [12] ZHENG W, SAKELLARIOU R. Budget-deadline constrained workflow planning for admission control[J]. Journal of Grid Computing, 2013, 11(4): 633-651.
- [13] SHIN K S, CHA M J, JANG M S, et al. Task scheduling algorithm using minimized duplications in homogeneous systems[J]. Journal of Parallel and Distributed Computing, 2008, 68(8): 1146-1156.
- [14] VERMA A, KAUSHAL S. Cost minimized pso based workflow scheduling plan for cloud computing[J]. International Journal of Information Technology and Computer Science (IJITCS), 2015, 7(8): 37.
- [15] CAO H, JIN H, WU X, et al. DAGMap: Efficient and dependable scheduling of DAG workflow job in Grid[J]. The Journal of Supercomputing, 2010, 51(2): 201-223.

编辑 叶芳