

混合云环境中数据敏感 workflow 调度

范菁 沈杰 熊丽荣

(浙江工业大学计算机科学与技术学院 杭州 310023)

摘要 混合云环境下调度包含敏感数据的工作流主要考虑在满足数据安全性以及工作流截止时间的前提下,对工作流任务在混合云上进行分配,实现计算资源与任务的映射,并优化调度费用。采用了整数规划来建模求解包含数据敏感性、截止时间和调度费用3种约束条件的混合云工作流调度问题,同时为优化模型求解速度,基于“帕雷托最优”原理对工作流任务在混合云上的分配方案进行筛选以减小模型求解规模。实验表明,优先排除不合理的任务分配方案可有效减小整数规划模型的求解规模,缩短模型计算时间,在产生较小误差的情况下获得较优的调度结果。

关键词 工作流调度,混合云,敏感性,整数规划

中图分类号 TP393 **文献标识码** A

Scheduling Data Sensitive Workflow in Hybrid Cloud

FAN Jing SHEN Jie XIONG Li-rong

(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract Using public resources to extend the capacity of private cloud is an effective way for the enterprises to achieve high efficiency and elasticity in data storage and computing. Scheduling workflow with sensitive data in hybrid cloud needs to satisfy the requirements of data security and execution deadline. In order to minimize the monetary cost, the scheduler must decide which tasks should be run on the public cloud and on which computing resource each workflow task should be allocated. Integer linear program(ILP) was used to formulate workflow scheduling problem with three objectives, such as data sensibility, deadline and cost. For the purpose of reducing the solve time of ILP model, the task assignment filter strategy based on Pareto optimality theory was designed. The filter strategy can decrease the scale of task assignments, and reduce the mappings between tasks and resources of ILP model. Experiments show that removing the unreasonable task assignment before resource allocation can decrease the ILP model scale and reduce scheduling running time, while the method can obtain a good solution.

Keywords Workflow scheduling, Hybrid cloud, Sensitive, ILP

1 引言

混合云是由公有云与私有云共同组成的新型云环境。企业或机构采用混合云的主要出发点是当自身搭建的私有云计算能力不足时,可按需使用公有云资源来增强计算能力,同时通过将重要数据或任务部署在私有云,将其他任务部署到公有云中执行,来防止企业或机构的重要数据泄露。

云环境中的工作流调度是一个 NP 难问题,通常将工作流抽象成一个向无环图(Directed Acyclic Graph, DAG)。DAG 图中的节点表示工作流中的任务,边表示任务间的依赖关系。混合云环境下的工作流调度需要考虑工作流截止时间、任务数据的敏感性、调度费用等多个因素,其中需要解决的关键问题如下。

第一,任务分配。当私有云计算能力不足时,需要将部分任务转移到公有云中执行。在选择任务的过程中需要考虑任务间的依赖关系、任务调度产生的额外数据传输费用与时间

以及任务数据的安全性。

第二,资源分配。在为每个任务分配计算资源时,应该尽量提高资源利用率,充分发挥任务间的并行性。同时,为了优化调度成本,应为资源选择最合理的计算资源。

本文主要研究在保证满足用户对工作流截止时间与数据敏感性要求的前提下,在混合云中调度包含敏感数据的工作流任务,优化工作流调度费用。为减少工作流任务在混合云上的分配方案数量,本文基于“帕雷托最优”原理对任务分配方案进行筛选,找出可能成为最优解的方案构成最小候选集。然后利用最小候选集,根据整数规划方法对工作流调度问题建模,寻找每种任务分配方案的最佳资源分配策略,找出最优的调度策略。

本文第2节是工作流调度的相关工作;第3节描述系统调度模型;第4节介绍任务分配方案筛选方法;第5节阐述如何利用整数线性规划思想对问题建模;第6节是模拟实验,并通过结果分析算法性能。

本文受国家自然科学基金(61173097),浙江省重大科技专项重大工业项目(2013C01112),杭州市重大科技创新专项(20132011A16)资助。

范菁(1969—),女,博士,教授,主要研究方向为服务计算、虚拟现实, E-mail: fanjing@zjut.edu.cn; 沈杰(1989—),男,硕士生,主要研究方向为云计算; 熊丽荣(1973—),女,硕士,副教授,主要研究方向为服务计算(通信作者)。

2 相关工作

近年来,云环境工作流调度的研究受到了广泛关注^[1]。在多个云环境下调度任务需要考虑以下两个重要因素:1)不同云服务商之间的信息获取;2)应用的安全性约束,安全性约束能够减小任务可选资源的规模^[2]。本文主要考虑第二点内容。

文献[3,4]是有关基于最大努力交付(best-effort)的工作流调度策略的,主要研究在单云环境下调度单个工作流。文献[3]通过在合适的资源空闲时隙中插入任务来完成资源调度。文献[4]提出了一种 PCH(Path Cluster Heuristic)算法,采用任务聚类的方式将多个任务组成一个聚簇并分配到同一个计算资源上执行,使得聚簇内部的任务间不再产生由数据传输导致的传输时延。

文献[5,6]研究基于效用调度的工作流调度算法,即在满足用户截止时间的要求下,尽量降低调度成本。其中文献[5]将工作流整体的截止时间分摊到每个子任务中,每个子任务都拥有一个子截止时间。然后根据任务的子截止时间分配合适的计算资源,保证每个子任务都能在子截止时间内完成,从而在满足工作流截止时间约束的前提下,最小化调度费用。文献[6]研究在混合云环境下的调度,先通过 PCH 算法判断工作流在私有云中能否在截止时间之前完成,如果不能,则迭代选取任务到公有云中执行,直到满足截止时间为止。文献[7]则提出了一个在混合云环境下满足安全性约束的调度框架系统,系统由调度器生成一种符合安全性约束与时间约束的调度方案,然后由资源分配器给每个任务分配对应的计算资源。

上述工作流调度算法大多基于 DAG 图属性实现,并采用贪心、剪枝等启发式搜索策略。这些算法通常具有较快的求解速度,一般情况下都能获得理想的调度结果。但是,由于这类算法基于 DAG 图属性实现,在计算这些图属性时,都利用虚拟的同构环境来模拟实际的异构云环境,且这些启发式策略无法兼顾全局优化目标,因而这些调度算法的调度结果受工作流结构、任务等本身属性的影响较大。

由于工作流调度在某种意义上属于组合优化问题,整数线性规划(Integer Linear Programming, ILP)能够较好地解决组合优化问题,因此 Gene 与 Bittencourt 等人^[8]通过建立 ILP 模型来解决在多个服务提供商中调度工作流的问题,并将连续的时间离散化。但是该调度模型规模会随着任务数量以及截止时间的增加而增大,为加快求解速度,文献[9]中通过增大离散时间的粒度来减小时间这一维度的规模,从而减少计算时间。文献[9]的方法虽然有了一定的性能提高,但仍存在求解规模大的问题;此外,在考虑 QoS 约束时没有考虑工作流中数据的敏感性问题,而且计费模型也只考虑了任务计算费用,未考虑数据在网络中传输产生的通信费用。

我们发现在混合云调度中,不合理的任务分配方案将导致频繁的数据传输,这将大大增加调度费用。因此设计计费模型需要将数据传输费用因素考虑在内,而且为了提高工作流调度问题的求解速度,有必要提前排除不合理的任务分配方案。

Zhang 等人^[10]在研究多工作流任务在混合云上的分配情况时,利用帕雷托最优原理筛选任务在混合云上的分配方

案,得到任务分配方案最小候选集。但他们考虑的是一个同构的混合云环境下,并没有考虑到构成私有云与公有云的计算资源类型不同。

3 系统模型

本文针对公有云和私有云异构场景中由不同类型的计算资源带来的任务间资源竞争问题,综合考虑混合云调度中的用户截止时间、调度费用和数据敏感性等 QoS 约束,采用整数规划 ILP 模型寻找在每一种任务分配方案下任务与资源的最佳组合。同时为了优化 ILP 模型求解速度、缩小模型的求解规模,本文借鉴了帕雷托最优原理,除了用于筛选工作流任务在混合云上的分配方案,进一步将其用于减少 ILP 模型中任务与计算资源间的映射关系,从而能够以较快的求解速度得到满足用户截止时间和数据敏感性要求的、费用最低的调度方案。

本文将工作流调度问题抽象为以时间、敏感性为约束,以减少调度费用为目标的优化问题。工作流可以用 DAG 图描述为 $G(P, E)$ 。其中节点集合 P 表示工作流中任务的集合,边的集合 E 表示任务间的关系。若节点 p_i 与 p_j 分别表示工作流中的子任务 u_i 与 u_j ,如果节点 p_i 与 p_j 间存在一条由 p_i 指向 p_j 且权值为 $d_{i,j}$ 的边,则表示任务 u_j 必须在任务 u_i 完成之后才能执行,且两任务间的数据传输量为 $d_{i,j}$ 。更详细的工作流描述如下:

- n :表示工作流中子任务数目($n \in \mathbb{N}^+$);
- $U = \{u_1, \dots, u_n\}$:表示构成工作流的子任务的集合;
- $P = \{p_1, \dots, p_n\}$:表示 DAG 图中任务所对应点的集合;
- $W = \{w_1, \dots, w_n\}$: w_i 表示任务 $u_i \in U$ 所需执行指令的数目($w_i \in \mathbb{N}^+$);
- $d_{i,j}$:表示任务 u_i 到 u_j 的数据传输量($d_{i,j} \in \mathbb{N}^+$);
- $H(j)$:表示任务 u_j 的直接先序节点(父节点)的集合;
- $E = \{e_{i,j} \mid \forall u_i \in H(j); i, j = 1, 2, \dots, n\}$:表示 DAG 图中由父任务指向子任务的边的集合;
- D_G :表示工作流的所要求的截止时间;
- $I = \{I_1 = 0, I_2 = 1\}$: I_1 与 I_2 分别表示私有云和公有云。

则 DAG 图 $G(P, E, D_G)$ 就可描述截止时间为 D_G 、由任务集合 P 构成且任务间关系为 E 的工作流。

在完成工作流抽象描述之后,对于工作流中的每一个任务,都可以将其分配到公有云或者私有云,并分配对应的计算资源。一个由 n 个任务组成的工作流,将任务分配到私有云或者公有云上执行,最多存在 2^n 种分配方法。本文将对工作流中所有任务的一种分配方法称为一个分配方案 $f = \langle f^{pri}, f^{pub} \rangle$ 。一个分配方案可分为两部分,分配在私有云上任务的集合记作 f^{pri} ;同理,分配在公有云上的任务的集合记作 f^{pub} 。同时,本文用 f^{pri} 中所有任务的指令的和来表示该分配方案对私有云产生的工作负载,记作 a 。用数据传输与资源使用产生的费用来表示该方案对应的调度费用,记作 b 。式(1)与式(2)分别给出了工作负载与调度费用的计算公式。

$$a = \sum_{i=1}^n w_i \times \theta_i \quad (1)$$

$$b = \alpha \sum_i w_i \times (1 - \theta_i) + \beta \sum_{i,j} d_{i,j} \times |\theta_i - \theta_j|; i, j = 1, \dots, n \quad (2)$$

其中, α 与 β 分别对应计算费用单价与通信费用单价; θ_i 等于

0 或 1, 分别表示任务 u_i 被分配到公有云或者私有云。在混合云中调度工作流的系统模型如图 1 所示。

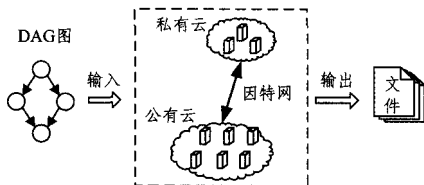


图 1 混合云环境工作流调度系统模型

从图 1 中可以看出, 一个输入的工作流被描述为一个 DAG 图, 工作流任务将被分配到私有云和公有云中执行, 并得到最终的调度结果。在调度工作流任务时, 不仅需要考虑私有云计算能力, 使其满足用户截止时间约束, 还需要考虑数据的敏感性约束。图 2 描述了混合云工作流调度的整体流程。

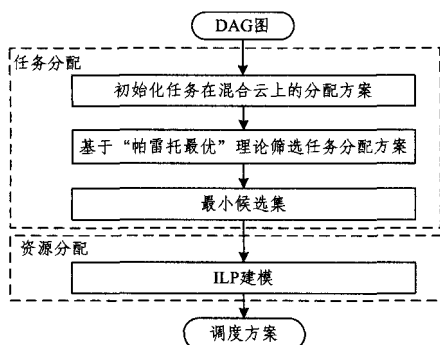


图 2 混合云工作流调度流程

由于一般企业应用包含的子任务数量并不多, 因此任务分配阶段的计算量较小。在初始化任务分配方案阶段, 本文采用穷举的方法初始化了所有的任务在混合云上的分配方案。同时, 为保证数据安全性, 只允许将包含敏感数据的任务分配到私有云中, 从而生成的任务分配方案可以防止敏感数据泄露。

在工作流调度模型中主要的计算时间集中在资源分配阶段, 需要根据任务分配方案通过 ILP 建模实现任务与资源的映射, 为工作流任务分配计算资源。通过在任务分配阶段对初始化得到的任务分配方案进行筛选, 可以缩小资源分配阶段的映射计算规模, 减少计算时间。

4 基于帕雷托最优的任务分配策略

在混合云中调度工作流任务时, 将一个任务指派到公有云中一方面能够缓解私有云工作负载, 减少任务的执行时间; 但另一方面也相应增加了数据传输时间以及调度费用。需要在资源分配阶段之前排除那些不可能成为最优任务分配的方案, 从而大大加快资源分配阶段的求解速度。

帕雷托最优是指资源分配的一种理想状态, 最早由意大利经济学家维弗雷多·帕雷托 (Vilfredo Pareto) 在关于经济效率和收入分配的研究中提出, 是评价一个经济体和政治方针的非常重要的标准, 已经有学者将该理论用于优化云资源分配^[11]。

本文通过判断每种任务分配方案是否为“帕雷托最优”状态, 来排除低效的任务分配策略。由于“帕雷托最优”是成为最优解的最低标准, 且难以通过任务的分配来估计工作流完成时间, 因此在任务筛选阶段并不保证最终的筛选结果都能

够满足截止时间约束。在任务分配阶段, 可保证每种分配策略都能满足用户对数据敏感性的需求, 截止时间约束将在资源分配时得到满足。因而, 本文筛选任务分配方案时选用方案对私有云产生的负载与调度费用作为两个比较标准。

每种分配方案 f 都有对应的由对私有云的工作负载和调度费用组成的二元组。私有云工作负载是指每一种分配方案中, 被分配到私有云任务的指令和; 调度费用是指该分配方案下, 由公有云计算产生的计算费用以及公有云与私有云间数据传输产生的通信费用 (云环境内部通信不计费) 的总和。若分配方案 f_1 与 f_2 可分别对应二元组 (a, b) 与 $(\hat{a}, \hat{b})$, 若 $a \leq \hat{a}$ 且 $b \leq \hat{b}$, 则说明与 f_2 相比, 方案 f_1 对私有云的计算压力更小且产生的费用更低。在帕累托最优理论中, 可认为与方案 f_2 相比, 方案 f_1 在不影响私有云负载的前提下, 优化了调度费用, 方案 f_1 是方案 f_2 的帕累托改善。因此方案 f_2 可被方案 f_1 替代, 不是帕累托最优状态, 可以率先排除。最终剩余的方案都属于帕累托最优状态, 可构成一个最小候选集 $MF = \langle f_1, \dots, f_m \rangle$ 。

在求解最小候选集之前, 必须先求得分配方案所对应的私有云工作负载以及调度费用。私有云工作负载可按式 (1) 求得。在计算调度费用时, 文献[10]的模型通过任务所需的计算能力来描述任务大小, 并通过统一的计算单价来计算调度费用。因此文献[10]考虑的公有云属于同构环境, 不同计算资源收费相同。在本文的计算模型中, 任务通过指令数量描述任务大小, 同时公有云提供的计算资源是异构的, 拥有不同的使用收费标准。因此在计算调度费用前, 首先必须确定式 (2) 中的计算单价 α 以及数据传输单价 β 。

从调度系统模型中可以看出, 私有云和公有云通过同一网络环境连接, 因此根据网络带宽和费用可以换算出传输单位数据所需的费用。由于公有云中存在不同类型的计算资源, 对应的每条指令的计算单价也各不相同, 在该阶段需要估计每条指令的执行费用。为完成计算, 本文利用最昂贵的计算资源使用费用来估计计算单价 α 。

图 3 描述了由 9 个子任务构成的工作流的所有分配方案在二维平面上所对应的点, 最小候选集所对应的点用黑色方块表示, 其连线被称为帕雷托前沿曲线。从图中可以看出, 最小候选集能够很大程度上减少任务在混合云上的分配方案, 从而减小任务在选择计算资源时的计算规模。

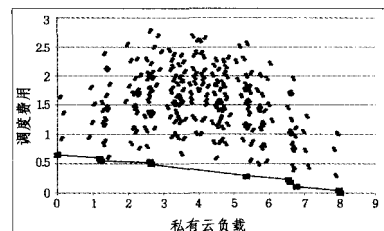


图 3 最小候选集

5 整数线性规划建模

在求得最小候选集之后, 便可知任务与云环境的对应关系。此处将先给出对于云环境以及虚拟机的描述与定义:

- V_i : 云环境 I_i 下的虚拟机的集合。
- $V = \sum_{i \in I} V_i$: 表示所有虚拟机的集合。
- PE_v : 表示虚拟机 $v \in V$ 拥有的计算单元的数目, $PE_v \in \mathbb{N}^+$ 。

• δ_v :表示虚拟机 $v \in V$ 计算一条指令所需的时间单元, $\delta_v \in \mathbb{N}^+$ 。

• $\lambda_{v,\tilde{v}}$:表示在连接虚拟机 $v \in V$ 和 $\tilde{v} \in V$ 的网络上传输单位数据量所需的时间单元, $\lambda_{v,\tilde{v}} \in \mathbb{N}^+$ 。当 $v = \tilde{v}$ 时, $\lambda_{v,\tilde{v}} = \lambda_{v,v} = 0$ 。

• C_v :表示虚拟机 $v \in V$ 执行单位时间所需的费用。

• $B_{i,v} = 0, 1$:表示虚拟机属于云环境 I_i , 0 则不属于。

本文仅考虑私有云与公有云两个云环境,由同一网络环境连接。同时,本文假设在云环境内部局域网带宽足够大,数据传输产生的时延可忽略,网际间的数据传输以发送时延为主,忽略传播时延。因此

$$\lambda_{v,\tilde{v}} = \begin{cases} 0, & v \in I_i, \tilde{v} \in I_j, i=j \\ b, & v \in I_i, \tilde{v} \in I_j, i \neq j \end{cases}$$

$$\Rightarrow \lambda_{v,\tilde{v}} = \lambda_{I_i,I_j} = \begin{cases} 0, & I_i = I_j \\ b, & I_i \neq I_j \end{cases}$$

在前文基础上,可通过 ILP 对调度问题进行建模。首先引入 3 个待求未知变量 $x, y, \tilde{\omega}_i$:

• $x_{u,t,v}$:为由任务 u 、时刻 t 、计算资源 v 组成的三维向量。表示 t 时刻任务 u 是否在计算资源 v 上完成计算。若任务 $u \in U$ 于 t 时刻在计算资源 $v \in V$ 上完成计算,则 $x_{u,t,v} = 1$, 否则 $x_{u,t,v} = 0$ 。

• $y_{i,v}$:为由时刻 t 、计算资源 v 组成的二维向量。表示在时刻 t 时,计算资源 v 是否处于被使用状态。若 t 时刻虚拟机 v 处于被使用状态,则 $y_{i,v} = 1$, 否则 $y_{i,v} = 0$ 。

• $\tilde{\omega}_i$:表示 $f_i \in MF$ 是否为最终被选择的最优分配方案,若 $f_i \in MF$ 为最优分配方案,则 $\tilde{\omega}_i = 1$, 否则 $\tilde{\omega}_i = 0$ 。

在工作流调度中,时间可被当作连续的或者离散的变量。在许多情况下,租用公有云计算资源的时间不到一个单位时间也往往按照一单位时间计费,因此本文将时间看作离散的变量,且工作流调度有时也需要离散的时间来代表离散的决定^[12,13]。从未知变量 $x_{u,t,v}$ 可以看出,计算规模与工作流任务数量、截止时间以及虚拟机数目相关。为减小计算规模,文献[9]通过增大截止时间这一维度的粒度来减小未知变量 $x_{u,t,v}$ 与 $y_{i,v}$ 的规模,从而缩短模型计算时间。

虽然模型不能改变任务数量以及计算资源的数目,但是任务与计算资源间的映射关系可以减小。因此,本文通过求解工作流任务在混合云上分配的最小候选集来减小任务与计算资源两者对应关系的规模。最小候选集中的分配方案规定了每个任务所属的云环境,可大幅减少每个任务的可用资源的数目,减小模型中未知变量的数目。即若任务 u 被分配到公有云,则私有云中的计算资源都为不可用,可以提前将这部分计算资源对应的 x, y 的值设置为 0,减少未知变量的数目,提高求解速度。由于最小候选集包含了多种分配方案,本模型根据候选集中方案被选中的情况,动态调整每个任务在当前被选中方案中所对应的可用资源集,同时设置不可用资源对应的 x, y 变量的值。

在文献[8,9]的基础上,本文利用整数规划建立了如下工作流调度模型:

• 目标函数: $\min \sum_{t \in T} \sum_{v \in V} C_v \times y_{i,v} + \beta \sum_{f_i \in MF} \sum_{u, z \in U} \tilde{\omega}_i \times d_{u,z} \times |\theta_u^i - \theta_z^i|$, β 为传输单位数据所需的费用。目标函数中的第一部分为虚拟机运行产生的费用,第二部分为数据传输产生的费用。

• 约束 1:工作流中的任务存在先后执行顺序,子任务在

其所有父任务完成执行前都不能开始执行。约束 1 规定了任意任务 z 在所有父任务 u 完成之前都不能执行。用数学语言可描述为:对于 $\forall z \in U, \forall u \in H(z), \forall r, v \in V, \forall t \in T, \forall I_i,$

$$I_j \in I \mid B_{I_i,v} = 1, B_{I_j,r} = 1, \text{ 满足 } \sum_{s=\varphi}^{t - \lceil w_z \times \delta_r + d_{u,z} \times \lambda_{I_i,I_j} \rceil} x_{u,s,v} \geq \sum_{s=\varphi}^t x_{z,s,r}.$$

• 约束 2:每个计算资源都只拥有有限个计算核心。该约束限定了在任意时刻 T ,任意计算资源 v 上执行的任务数量不能超过 v 所拥有的计算核心的数目。可描述为:对于 $\forall v \in V, \forall t \in T$, 满足

$$\sum_{u \in U} \sum_{s=t, t \leq D_G - \lceil w_u \times \delta_v \rceil}^{t + \lceil w_u \times \delta_v \rceil - 1} x_{u,s,v} \leq PE_v.$$

• 约束 3:若任务 u 在 t 时刻于计算资源 v 上完成计算,则在任务的计算过程中,计算资源 v 都应处于被使用状态。描述为:对于 $\forall u \in U, \forall v \in V, \forall t \in \lceil w_u \times \delta_v \rceil, \dots, D_G$, 满足

$$\sum_{s=t - \lceil w_u \times \delta_v \rceil + 1}^t y_{s,v} \geq x_{u,t,v} \times \lceil w_u \times \delta_v \rceil.$$

• 约束 4:若两个父子任务 u, z 被分配到不同的计算资源 v, r 上,则会产生数据传输。在数据传输的过程中,计算资源 v, r 也必须处于被使用状态。约束可描述为:对于 $\forall z \in U,$

$$\forall u \in H(z), \forall r, v \in V, \forall s, t \in T, \text{ 满足 } \sum_{p=t-1}^{s - \lceil w_z \times \delta_r \rceil} (y_{p,v} + y_{p,r}) \geq 2 \times (x_{u,t,v} + x_{z,s,r} - 1) \times \pi. \text{ 其中 } \pi = (s - t - \lceil w_z \times \delta_r \rceil), \text{ 且满足 } \pi \geq \lceil d_{u,z} \times \lambda \rceil, \text{ 对于 } \forall I_i, I_j \in I \mid B_{I_i,v} = 1, B_{I_j,r} = 1.$$

• 约束 5:任务在被分配到指定的云环境之后,若不属于该云环境中的计算资源都不能执行该任务,则该任务在任意时刻都不可能在这些计算资源上完成。因此该约束可描述为:对于 $\forall u \in U, \forall v \in V, \forall t \in T$, 满足 $x_{u,t,v} \leq \sum_{f_i \in MF} \tilde{\omega}_i \times B_{\theta_u^i,v}.$

• 约束 6:由于每个任务都必须被执行且只能被执行一次,因此该约束可描述为:对于 $\forall u \in U$, 满足 $\sum_{t \in T} \sum_{v \in V} x_{u,t,v} = 1.$

• 约束 7:该约束描述了在任务所需的执行时间之前,该任务都不可能完成。即:对于 $\forall u \in U$, 满足 $\sum_{v \in V} \sum_{t=\varphi}^{\lceil w_u \times \delta_v \rceil} x_{u,t,v} = 0.$

• 约束 8:与约束 6 类似,最小候选集中的方案中只能选择其中一个作为最终的最优分配方案,因此模型必须满足 $\sum_{f_i \in MF} \tilde{\omega}_i = 1.$

• 约束 9:未知变量 $x_{u,t,v}$ 描述了任务 u 在时刻 t 是否在计算资源 v 上完成,因此变量 $x_{u,t,v}$ 只能为布尔值 0 或 1。同理, $y_{i,v}$ 与 $\tilde{\omega}_i$ 也必须为布尔值。因此 $x_{u,t,v} \in \{0, 1\}, y_{i,v} \in \{0, 1\}, \tilde{\omega}_i \in \{0, 1\}.$

在文献[8]的基础上,本文额外考虑了数据传输费用,约束 5 与约束 8 是本文新添加的约束条件。提前确定任务分配方案可以提前确定方案的数据传输费用,减少 ILP 模型中任务与计算资源的映射关系。约束 5 与约束 8 可减小未知变量的取值范围,达到优化求解速度的目的。

6 实验与分析

本文使用由美国 LINDO 公司开发的 LINGO 软件实现文中提出的利用整数线性规划建立的工作流调度模型。LINGO 是一款被设计用来构建和求解线性、非线性和整数优化模型的功能全面的工具。本文实验在双核 3.1GHz 处理器、16GB 内存的 64 位 Windows7 系统上完成。

6.1 模拟环境

参照文献[8]中的模拟环境设计方法,表1与表2给出了本文模拟环境中私有云与公有云虚拟机参数。其中“计算能力”表示虚拟机中每个核心单位时间内能够执行单位指令的数目。“费用”表示虚拟机运行单位时间产生的费用。本文利用执行工作流所需单位时间的数量来表示工作流执行时间长短。同理,工作流“截止时间”表示允许消耗的单位时间数量的最大值。

表1 私有云虚拟机

云环境	虚拟机类型	核心数目	计算能力	费用
私有云	小型	1	1	3
	小型	1	1	4
	小型	1	1	5

表2 公有云虚拟机

云环境	虚拟机类型	核心数目	计算能力	费用
公有云	小型	1	2	17
	中型	2	2	30
	大型	3	2	40

6.2 实验数据

实验中使用的 workflow 结构由 DAG 生成器随机生成。由于本文的调度模型中需要考虑数据敏感性,因此在实验前需要给 workflow 中的数据随机标注敏感属性。为了让标注结构更贴近实际情况,随机标注必须符合以下两项原则^[10]。

不允许向上转型,即如果当前任务所有的输入数据都是非敏感的,则当前任务产生的数据也必须是非敏感的。这样做是为了防止当前任务被分配到公有云中执行之后产生敏感数据,造成泄漏。

允许向下转型,即任务的输入数据为敏感数据时输出数据可以为非敏感数据。在现实中,敏感数据经过如加密等预处理之后可认为对其他人来说没有意义,因此可以认为被加密后的数据属于非敏感数据。这样做也可以使后续更多的任务能够被分配到公有云中,缓解私有云计算压力。

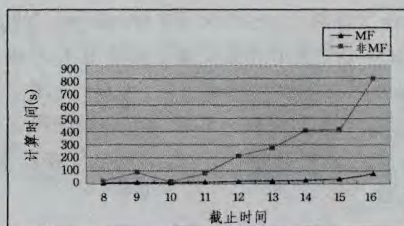
6.3 结果分析

本文提出的调度模型优化思想是通过任务分配方案的筛选来减小计算规模,加快模型在优化资源分配时的求解速度。因此本文通过实验比较使用任务分配方案筛选策略构建最小候选集与在资源分配过程中决定任务分配两种方式下,利用整数规划对工作流调度问题建模求解的性能。由于本文方法的计算过程中求解了最小候选集(minimal configure),因此本文将该方法记作 MF;对应的未做该优化的方法记作非 MF。非 MF 方法基于文献[8]实现,但在调度费用中加入了数据传输费用。

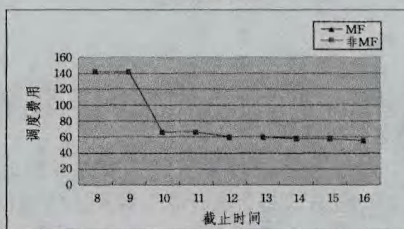
本文分别比较了两种方法在处理由 5、10、15 个节点构成的 workflow 时在不同截止时间要求下的模型计算时间以及调度方案产生的调度费用。截止时间规定了 workflow 任务从开始到结束的最大执行时间。计算时间为模型开始计算到完成计算的时间间隔。调度费用为模型求得的调度策略对应的费用,包含计算费用与数据传输费用。同时在实验中,我们限定模型的最长计算时间为 30 分钟,如果模型在 30 分钟内尚未求得全局最优解,则返回当前的最优解。

图4描述了两种方法在处理节点数为5的 workflow 时的计算时间与调度费用的结果。由于 workflow 节点数较少,从计算时间中可以看出,两种方法都在 30 分钟内完成了计算,找到

了模型的全局最优解。随着截止时间的增加,模型的求解规模逐渐变大,计算时间在整体上呈现增大的趋势。通过比较可以发现,本文的优化方法用时较少且增长较慢。在调度费用方面,由于两种方法都求得了全局最优解,因此调度费用都相同。



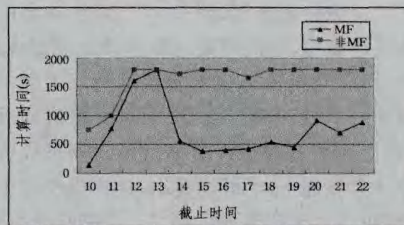
(a)计算时间



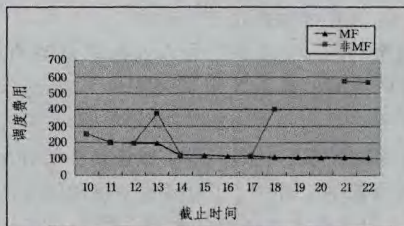
(b)调度费用

图4 5个节点的工作流实验结果

图5给出了两个模型在处理10个节点组成的 workflow 时的计算时间与调度费用的结果。由于数目的增多导致模型规模增大,任务关系也变得更加复杂。



(a)计算时间



(b)调度费用

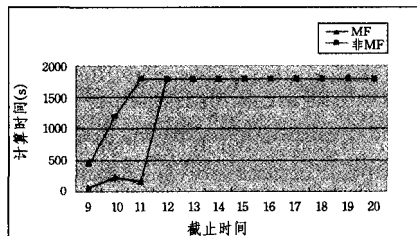
图5 10个节点的工作流实验结果

从图5(a)中可以发现,未经过优化的方法在大部分截止时间要求下的计算都用完了限定的30分钟。由于LINGO求解器本身在模型求解过程中进行了剪枝等优化处理,虽然问题规模随着截止时间增加而扩大了,但是模型的计算时间并没有在严格意义上随着模型变大而增加。本文的方法在截止时间从13变为14的过程中,计算时间骤减,这是由于截止时间的扩大使得对公有云资源的需求减少了。由于本文的优化方法通过最小候选集中的任务分配方案对问题建模,当对公有云资源需求减少时,需要计算的任务分配方案的规模也变小了,从而导致模型整体的计算时间减少。

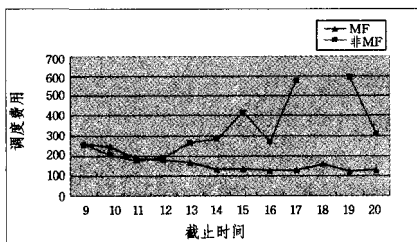
根据图5(b)可以看到,未经过优化的方法在截止时间为

15、16 等情况下,在 30 分钟内没有找到一种可行的、满足用户需求的任务调度方案,在图中表现为出现了断点。在截止时间为 17、18 等情况下,模型求解器恰好能够优化处理,减少了不必要的计算,使其在限定时间内找到了可行解,但可以看出结果并不好。本文的方法仍然能够在给定时间内完成模型计算,得到全局最优解(除了截止时间为 13 的情况)。由于非 MF 方法在某些情况下只能求得在给定时间内的最优解,因此在调度费用方面本文的优化方法的结果较优。

图 6 给出了 15 个节点的工作流的实验结果。随着节点数目进一步增加和截止时间的扩大,两种模型都不能在 30 分钟内完成计算,但在截止时间较小时仍然能够找到全局最优解。在图 6(b)调度费用比较中可以发现,在截止时间较小时,两种方法虽然都能够求得模型的全局最优解,非 MF 的方法的调度费优于 MF。这是由于 MF 在计算最小候选集时未能准确估计每条指令在公有云中的调度费用,产生了误差,导致最优的任务分配方案未被选入最小候选集中,但本文认为该误差较小,可以接受。在截止时间大于 12 时,虽然两个模型都未能在给定时间内完成计算,但是本文的方法所得到的调度方案的费用仍优于未作优化的模型。图 6(b)中出现断点的原因与图 5(b)相同。



(a) 计算时间



(b) 调度费用

图 6 15 个节点的工作流实验结果

通过上述实验数据可以得到以下结论：

1. 非 MF 的方法建立的整数规划模型通过求解器在求解过程中进行剪枝来优化计算速度,但由于需要进行大量计算判断某种调度策略是否可行,因此无效计算较多。本文的优化方法提前对任务分配进行了筛选,减小了模型求解过程中的计算规模,因此在求解速度上有了改进。

2. 在处理小规模的工作流调度问题时,本文方法能有效减少求解任务调度策略的计算时间,同时获得良好的实验结果。随着工作流规模增加,本文的优化方法在给定时间内更容易求得可行解,实验结果比较稳定,误差较小。但对于大规模的工作流调度问题,本文的方法也存在局限性。

结束语 本文在分析现有调度算法的基础上,提出了在混合云环境下调度数据敏感的工作流的调度算法。该算法能够显著减少调度时间,且计算结果与最优解误差较小。但该算法在计算最小候选集时,仍需穷举任务所有可能的分配。

因此当任务规模增加时,计算最小候选集所需的时间也将显著增加,这成为算法的一个瓶颈。在后续的工作中,可以采用启发式算法、演化算法等方法计算最小候选集。另一方面,在计算最小候选集时,由于虚拟机种类不同,很难准确估计每一个任务真正的计算费用,而对最终结果会产生一定的误差。在今后的工作中,我们试图加入自适应算法来适时调整每一个任务的费用单价,从而进一步减少误差。

参考文献

- [1] 储雅, 马廷淮, 赵立成. 云计算资源调度: 策略与算法[J]. 计算机科学, 2013, 40(11): 8-13
- [2] Bittencourt L F, Madeira E R M, Fonseca N L S D, et al. Scheduling in Hybrid Clouds[J]. Communications Magazine IEEE, 2012, 50(9): 42-47
- [3] Topcuoglu H, Society I C, Wu M, et al. Performance-effective and low-complexity task scheduling for heterogeneous computing[J]. IEEE Transactions on Parallel & Distributed Systems, 2002, 13(3): 260-274
- [4] Bittencourt L F, Madeira E R M. A performance-oriented adaptive scheduler for dependent tasks on grids[J]. Concurrency & Computation Practice & Experience, 2008, 20(9): 1029-1049
- [5] Abrishami S, Naghibzadeh M, Epema D H J. Cost-driven scheduling of grid workflows using Partial Critical Paths[C] // IEEE/ACM International Conference on Grid Computing. IEEE, 2010: 81-88
- [6] Luiz F B, Madeira E R M. HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds[J]. Journal of Internet Services & Applications, 2011, 2(3): 207-227
- [7] Marcon D S, Bittencourt L F, Dantas R, et al. Workflow specification and scheduling with security constraints in hybrid clouds[C] // IEEE Latin American Conference on Cloud Computing & Communications. 2013: 29-34
- [8] Genes T A L, Bittencourt L F, Madeira E R M. Workflow scheduling for SaaS/PaaS cloud providers considering two SLA levels[C] // IEEE Network Operations & Management Symposium. IEEE, 2012: 906-912
- [9] Bittencourt L F, Madeira E R M. Using Time Discretization to Schedule Scientific Workflows in Multiple Cloud Providers[C] // IEEE Sixth International Conference on Cloud Computing. IEEE Computer Society, 2013: 123-130
- [10] Zhang C, Chang E C. Processing of Mixed-Sensitivity Video Surveillance Streams on Hybrid Clouds[C] // IEEE International Conference on Cloud Computing. IEEE, 2014: 9-16
- [11] Li H, Li H. A Research of Resource Scheduling Strategy for Cloud Computing Based on Pareto Optimality M×N Production Model[C] // International Conference on Management & Service Science. 2011: 1-5
- [12] Floudas C A, Lin X. Mixed Integer Linear Programming in Process Scheduling: Modeling, Algorithms, and Applications[J]. Annals of Operations Research, 2005, 139(1): 131-162
- [13] Stefansson H, Sigmarsdottir S, Jensson P, et al. Discrete and Continuous Time Representations and Mathematical Models for Large Production Scheduling Problems: a Case Study From the Pharmaceutical Industry[J]. European Journal of Operational Research, 2011, 215(2): 383-392