

基于虚拟归约 workflow 三层决策模型的时间-质量优化算法

罗智勇^{1,2}, 朱梓豪¹, 尤 波², 刘嘉辉¹

(1. 哈尔滨理工大学计算机科学与技术学院, 黑龙江哈尔滨 150080;

2. 哈尔滨理工大学机械动力工程学院, 黑龙江哈尔滨 150080)

摘 要: 工作流的调度问题因为含有多个目标的限制因素, 属于非确定性 (NP) 问题. 本文通过将 workflow 模型分层, 提出一种优化调度算法. 算法分为两大阶段: 检测阶段通过循环加工提高生产质量; 虚拟阶段将各制约节点不断迭代虚拟成一个虚拟节点, 将服务抽象化, 最终简化服务的选择问题. 对比发现, 在相同实验环境下, 本算法相比传统最小关键路径算法具有一定优越性.

关键词: 优化调度; 工作流; 虚拟节点; 生产质量

中图分类号: TP393

文献标识码: A

文章编号: 0372-2112 (2019) 01-0245-07

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2019.01.033

A Time-Quality Optimization Algorithm of Three-Layer Decision Model Based on Virtual Iterative Workflow

LUO Zhi-yong^{1,2}, ZHU Zi-hao¹, YOU Bo², LIU Jia-hui¹

(1. School of Computer Science and Technology Harbin University of Science and Technology Harbin Heilongjiang 150080, China;

2. School of Mechanical and Power Engineering Harbin University of Science and Technology Harbin Heilongjiang 150080, China)

Abstract: The scheduling of workflows is a well-known non-deterministic (NP) problem because it contains multiple targets constraints. This paper proposed a time-quality optimization scheduling algorithm by layering the workflow model into three layers, such as mapping layer, dominant layer, and decision layer, respectively. The algorithm has two stages, stage one is called detection, it can improve the production quality by means of cycle machining in this stage; another stage is virtualization, which uses iterative way to virtualize nodes in mutual constraints into a new virtual node, abstracts the services, and simplifies the selection of services eventually. In comparison, the proposed algorithm was more effective than the traditional minimum critical path algorithm in the same experimental environment.

Key words: scheduling optimization; workflow; virtual node; production quality

1 引言

工作流技术作为现代大规模科学与工程计算的组织和管理工作方式, 对生产和科学领域起着越来越重要的作用. 通常的工作流模型, 包括所需要执行的任务或者活动、任务之间的相互逻辑顺序关系、整个流程的开始和终止条件以及对各个任务的具体描述信息. 由于工作流是由多个任务组成的, 每个任务可以选取不同的服务来执行, 因此, 调度策略的重心就是根据服务质量 (QoS) 来选择最优服务^[1,2].

目前, 执行时间和执行费用被考虑作为 QoS 最重要的两个需求, 并且工作流被模型化, 划分成不同的疏松任

务集合. 然后在截止期内以启发式调度算法去探寻最小代价的路径^[3]. 工作流问题跟查询优化类似, 是困难并且具有挑战的 NP-hard 问题^[4]. 那么, 时间和费用如何抉择的问题, 就成了工作流优化的目标——在相对时间和费用较少的情况下, 达到一定的服务质量. 在近些年的研究中, 吸引研究者注意的是工作流调度的优化目标函数所需要满足的限制因素. 无非两种, 一是在截止期内满足费用最小化; 二是在用户费用预算下, 实现最小调度路径, 通常为时间最短者. 无论哪种方式, 都未能考虑到调度的生产质量因素. 在当今社会注重质量的大环境下, 费用已不再是用户考虑的首要因素. 正因如此, 生产质量与时间的动态平衡关系成为本文研究的主要方向.

收稿日期: 2017-11-02; 修回日期: 2018-04-21; 责任编辑: 孙瑶

基金项目: 国家自然科学基金青年科学基金 (No. 61403109); 黑龙江省留学归国人员科学基金 (No. LC2018030)

本文的主要贡献在于提出了一种时间-生产质量优化算法,该算法可以在相应的截止期内,实现生产质量的最大化.算法通过将任务节点虚拟成一个虚拟任务节点,简化任务对应的服务选择,同时提出一种循环加工的方式解决生产质量不达标的情况,通过层层迭代不断地虚拟节点,最终可以实现在有限的时间内,让生产质量收益最高,达成调度时间与生产质量之间的动态平衡.

2 相关工作

工作流通常都是以减少花费为目的,一般所需要考虑的花费代价为时间和成本,并且需要在工程限制期内动态地选取服务以保持各方目标的平衡.文献[5]中 Maciej 等人将工作流程构建成 AMPL 模型,通过约束时间来降低总成本,不过随着发展,费用问题不再是唯一的主要问题,服务质量也应当成为优化的目标.文献[6]中 Kim 用 Petri 网进行工作流建模,资源类型按照工作流调度方案的工作量比例和策略来保证在约束时限下成本最低,其是采用启发式搜索算法找寻临近最优解;本文在此基础上迭代虚拟化节点寻求局部最大生产质量.文献[7]中 Stavriniades 等人针对异步云平台工作流的实时调度问题,提出了分步策略,任务选择阶段和虚拟机选择阶段;本文也将循环节点和虚拟化节点作为两部分分开处理.文献[8]中 Arabnejad 等人考虑到在工作流调度的结果问题上,可能存在未能在限制期内完成或者选择了错误的任务服务而导致的过高花费问题提出了四阶段的解决算法;本文则针对选取非最优的任务进行调度而未能达到预期生产质量提出了补救措施,具体方法在虚拟迭代归约算法局部实例中进行详细介绍.

在国内,也进行了不少相关研究.文献[9]引入了准备时间这一新的概念,但算法在实际运行时需要消耗大量时间.文献[10]提出了传统的最小关键路径算法,通过最小完工时间确定每个任务的服务选择,在截止期约束下,以此完成整个业务流程;本文则采用这个作为对比算法.文献[11]为解决网格工作流中任务的实时调度问题,提出了截止时间保证度的概念,通过动态的选择实现调度优化.

可见,在对业务工作进行调度时,需要考虑到多重因素,可通过对某属性的约束来实现其他目标的优化,最终达到多目标动态平衡的目的.本文依照这个原则,将完工时间作为约束条件,对生产质量进行优化,以实现完工时间和生产质量的动态平衡.

3 问题描述

工作流是分布式系统中常用的科学或工业应用模

型,一般表示为有向无环图,每个任务由图中节点表示,而任务之间的控制或数据依赖关系则由相应节点之间的有向弧表示. workflow 管理则是根据用户的预算和需求,将不同的资源服务分配给不同的任务,来实现整个工作流的调度过程.

3.1 工作流的相关定义

定义1 工作流模型 $W-M = \langle P, E, S, IS \rangle$. 其中: $P = \{p_1, p_2, \dots, p_i, \dots, p_n\}$ 表示任务池,每个任务都对应一个出度和入度,分别用 λ_i^* 和 λ_i^* 表示; E 表示 $W-M$ 中各任务之间的偏序关系,即任务 p_i 只有当其所有父任务全部完成,才可以执行; $S = \{S_1, S_2, \dots, S_j, \dots, S_m\}$ 表示服务池集合, S_j 为任务 p_i 的服务集合; IS 表示检测节点,指当前工作流的生产质量得达到通过率 σ 才允许通过继续之后的任务服务,若累积的生产质量低于该通过率 σ ,则需要返回至前一阶段重新加工直到生产质量达标.此外,工作流中设置虚拟开始任务 p_B 和结束任务 p_E .

定义2 服务属性 a_{ij} , $a_{ij} = (q_{ij}, h_{ij})$ 指任务 p_i 对应的每个服务 s_j 所具有的属性,其中 q_{ij} 表示可获得的生产质量, h_{ij} 表示所用时间.

定义3 任务完成时间.指进行到任务 p_i 的累积时间,若表示为 H_i ,则

$$H_i = \text{Max} \{ \dots, H_j, \dots \} + h_i \quad (1)$$

其中 $k=1, 2, \dots, \lambda_i^*$; H_j 为 p_i 的父任务 p_j 的执行时间, $j < i$; h_i 表示完成 p_i 花费的时间.

定义4 生产质量 Q_i .指进行到任务 p_i 的累积生产质量.

$$Q_i = \prod_{p_i \in P'} q_j \quad (2)$$

其中 P' 表示到任务 p_i 时的已完成任务集合, q_j 表示 p_j 的生产质量,其中 $p_j \in [0, 1]$, $q_B = q_E = 1$.

定义5 工程限制期 DL .指任务集合 $P' = \{p_1, p_2, \dots, p_i\}$ 组成的工作流的执行限制时间,其中: $P' \in P$, 当 $i = n$ 时,为工作流的执行限制时间.

定义6 任务自由度 t_{DFp_i} .指每个任务可能的开始时间,表示为 $t_{DFp_i} \in [t_{BEp_i}, t_{ENp_i}]$,其中 t_{BEp_i} 和 t_{ENp_i} 分别表示任务 p_i 的最早和最晚开始时间, t_{BEp_i} 和 t_{ENp_i} 可由式(3)求得.

$$\begin{cases} t_{BEp_i} = \text{Max} \{ t_{BEp_{i-1}} + \text{Min} (h_{ij}) \} & t_{BEp_i} = 0, \text{ 招扑} \\ t_{BEp_{i-1}} \in \{ \dots, t_{BEp_j}, \dots \} \\ t_{ENp_i} = \text{Min} \{ t_{ENp_{i-1}} - \text{Min} (h_{ij}) \} & t_{ENp_i} = DL, \text{ 送扑} \\ t_{ENp_{i-1}} \in \{ \dots, t_{ENp_j}, \dots \} \end{cases} \quad (3)$$

式(3)中 $k=1, 2, \dots, \lambda_i^*$, $j=1, 2, \dots, \lambda_i^*$; t_{BEp_i} 和 t_{ENp_i} 分别

为 p_i 的父任务 p_q 的最早开始时间和子任务 p_p 的最晚开始时间.

在 workflow $W-M$ 中, 在工程限制期 DL 约束下, 通过在关键路径选取执行时间最快的服务调度整个 workflow 即为传统的最小关键路径算法. 根据文献 [12] 中的算法即式 (4) 可求得每个任务 p_i 的最早开始时间 t_{α_i} 与最迟完工时间 t_{β_i} .

$$\begin{cases} t_{\alpha_i} = t_{BE_i} \\ t_{\beta_i} = \min \{ t_{EN_{p_j}} \} \mid t_{EN_{p_j}} \in \{ \dots, t_{EN_{p_i}}, \dots \} \end{cases} \quad (4)$$

其中 $k=1, 2, \dots, \lambda_i$; $t_{EN_{p_j}}$ 为任务 p_i 的直接后继任务 p_j 的最迟开始执行时间.

3.2 三层 workflow 模型实例

经分析某公司的生产调度过程存在 9 个任务节点, 将这些节点之间的制约条件抽象为集合 E , 每个任务 p_i 对应一个部门并提供一个服务池 S_i . 将这些信息输入 workflow 模型生成算法 WFG 中, 可得图 1 所对应的 workflow 模型, 且约定工程限制期 $DL = 16$. 该 workflow 模型各任务的服务参数如表 1 所示.

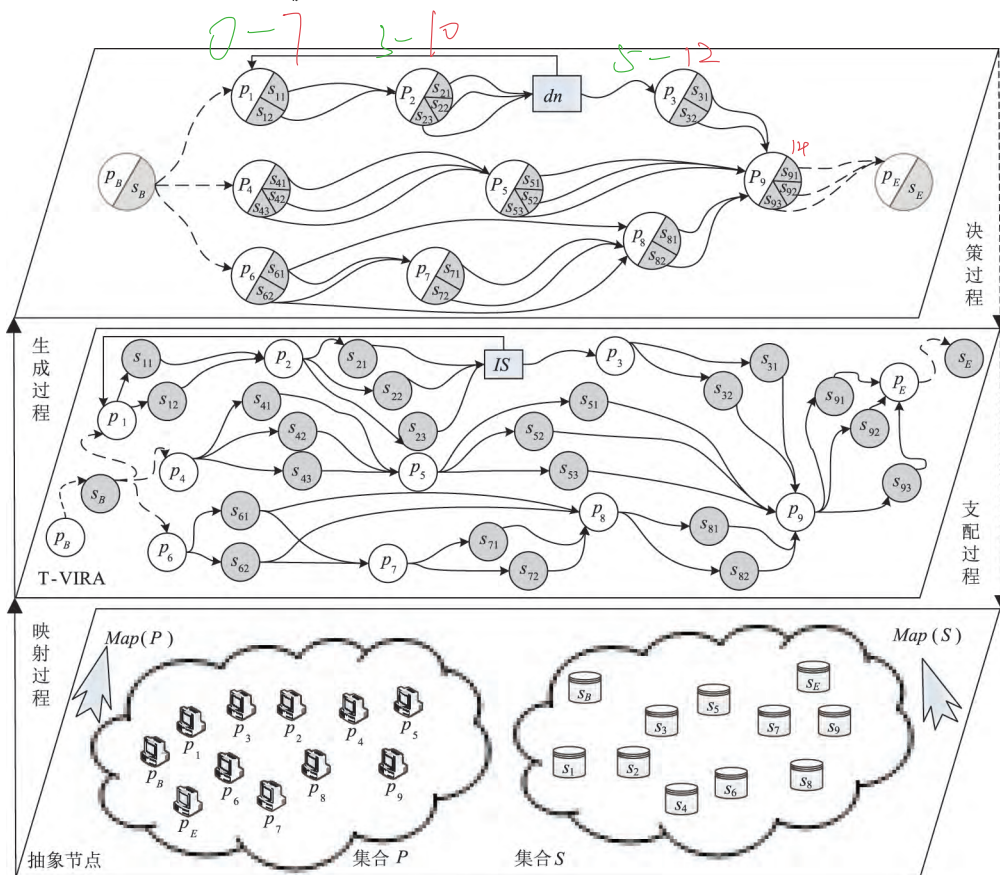


图1 三层 workflow 模型实例

表 1 各任务的服务池

任务 p_i	任务对应的服务池 S_i
p_1	(0.88 3) (0.90 4)
p_2	(0.75 2) (0.82 3) (0.88 4)
p_3	(0.78 2) (0.85 3)
p_4	(0.79 2) (0.88 3) (0.92 4)
p_5	(0.72 1) (0.80 2) (0.87 3)
p_6	(0.86 3) (0.89 4)
p_7	(0.90 4) (0.97 5)
p_8	(0.73 2) (0.90 4)
p_9	(0.74 2) (0.89 4) (0.97 5)

3.3 三层 workflow 模型的约束分析

图 1 所示 workflow 还需满足部分约束条件, 才能在工程限制期内完成对生产质量的优化调度. 具体约束条件如式 (5) 和式 (6) 所示.

$$\begin{cases} Q_q = \prod_{p_i \in P} l_{ij} q_{ij} \\ \text{s. t. } \sum_{j=1}^m l_{ij} = 1 \quad S_i = \{s_1, s_2, \dots, s_m\} \\ l_{ij} \in \{0, 1\} \end{cases} \quad (5)$$

一个任务只选1个服务

其中 P 表示到任务 p_q 时的调度的任务集合; l_{ij} 表示是否选用任务 p_i 时对应的服务集合 S_i 中的服务 s_j 来执行; 若 $n_q = n_k$ 时, 表示 workflow 调度完成并用 Q_k 表示其获得

的生产质量.

$$\begin{cases} H_q = \text{Max} \{ \dots H_{p_i} \dots \} + l_{qj} h_{qj} < DL, \\ k = 1, 2, \dots, n_{p_i} \end{cases} \quad (6)$$

其中 H_p 为任务 p_i 的父任务累积执行的时间.

4 三层虚拟归约算法

4.1 算法描述

任务 p_i 在其执行域内获得的最大生产质量如式 (7) 所示.

$$\begin{cases} f(p_i, h) = \max \{ q_{ij} \} \\ h \in [BE_{p_i}, EN_{p_i}] \quad h + h_{ij} \leq DL \end{cases} \quad (7)$$

任务 p_i 得到的最大生产质量与其父任务 p_{i-1} 得到的最大生产质量之间的关系如式 (8) 所示.

$$\begin{cases} f(p_{i-1}, h_{i-1}) = \max \{ f(p_i, h_{i-1} + h_{i-1j}) \times q_{i-1j} \} \\ h_{p_{i-1}} \in [t_{BE_{p_{i-1}}}, t_{EN_{p_{i-1}}}] \end{cases} \quad (8)$$

三层虚拟归约算法 T-VIRA 步骤如下.

INPUT: workflow 模型 $W-M$

OUTPUT: 生产质量 Q 以及服务选择集合 Set .

1. 初始化 $Queue$ Set Q 为 NULL;
2. 通过式 (5) 计算任务的自由度存于 $t_{DF_{p_i}}$ 中;
3. 根据 $W-M$ 找到出入度大于 1 的任务并加入 $Queue$ 中 (不包括循环结构中的任务);
4. While (!IsEmpty($Queue$) && $Queue$ 中存在出度大于 1 的任务) {
5. 取出第一个出度 > 1 的任务 p_i ;
6. 查找队列中与出度任务距离最小的入度大于 1 的任务记为 p_j ;
7. 根据式 (7) 求两任务间的执行域;
8. 将任务 p_i 和任务 p_j 虚拟成任务 $p_{[i,j]}$;
9. 用式 (2) 求不同时间获得的最大生产质量;
10. 将服务选择保存在 Set 中;
11. While ($W-M$ 中存在串行的任务节点) {
12. 由式 (7) 求串行任务之间的执行域;
13. 将串行任务虚拟化, 用式 (2) 求得不同时间获得的最大生产质量;
14. 将服务选择保存在 Set 中;
15. If (遇到检测节点 IS) {
16. 由式 (1) 计算任务的累计花费时间 h ;
17. While ($Q \leq \sigma_2$) {
18. If ($Q \leq \sigma_1$ && $h + h_1 \leq \psi_{\max}$) {
19. $\{ h += h_1; Q = -Q^2 + 2Q; \}$
20. Else If ($h + h_2 \leq \psi_{\max}$) {
21. $\{ h += h_2; Q = -Q^2 + 2Q; \}$
22. Else break; }
23. } If ($W-M$ 中存在并行的任务节点) {
24. 由式 (7) 求并行任务之间的执行域;
25. 将并行任务进行虚拟化, 用式 (2) 求得不同时间获得的最大生产质量;
26. 将服务选择保存在 Set 中;
27. } If ($W-M$ 未虚拟成一个节点) Goto 9;

最终选取满足工程限制期的虚拟节点的最大生产质量即为 Q .

4.2 三层虚拟归约算法局部实例

图 1 所示的 IS 为检测节点, IS 的达标率为 0.9, 当生产质量低于 $\sigma_1 = 0.7$ 时, 需要多花费时间 3 进行加工; 高于 σ_1 且低于 $\sigma_2 = 0.8$ 时多花费时间 2 加工, 计算得任务 2 的工程限制期为 12, T-VIRA 处理过程如下 (参考表 2): 将任务 p_1 、任务 p_2 组合成虚拟任务 $p_{[1,2]}$.

表 2 不同工程限制期下累积生产质量 Q_2 的计算过程

开始时间	处理过程
0	$Q_2 = 0.792$ $H_2 = 4 + 4 = 8$; 因 $\sigma_1 < Q_2 < \sigma_2$ 循环加工 $H_2 = 8 + 2 = 10$ 得 $Q_2 = 0.792 + (1 - 0.792) \times 0.792 = 0.957$.
1	$Q_2 = 0.792$ $H_2 = 4 + 4 + 1 = 9$; 因 $\sigma_1 < Q_2 < \sigma_2$ 循环加工 $H_2 = 9 + 2 = 11$ 得 $Q_2 = 0.792 + (1 - 0.792) \times 0.792 = 0.957$.
2	$Q_2 = 0.792$ $H_2 = 4 + 4 + 2 = 10$; 因 $\sigma_1 < Q_2 < \sigma_2$ 循环加工 $H_2 = 10 + 2 = 12$ 得 $Q_2 = 0.792 + (1 - 0.792) \times 0.792 = 0.957$.
3	$Q_2 = 0.792$ $H_2 = 4 + 4 + 3 = 11$; 因 $\sigma_1 < Q_2 < \sigma_2$ 循环加工 $H_2 = 11 + 2 = 13 > \psi$ 不满足
4	$Q_2 = 0.792$ $H_2 = 4 + 4 + 4 = 12$; 不存在空闲时间, 不进行循环加工.
5	$Q_2 = 0.774$ $H_2 = 3 + 4 + 5 = 12$; 不存在空闲时间, 不进行循环加工.
6	$Q_2 = 0.722$ $H_2 = 3 + 3 + 6 = 12$; 不存在空闲时间, 不进行循环加工.
7	$Q_2 = 0.66$ $H_2 = 3 + 2 + 7 = 12$; 不存在空闲时间, 不进行循环加工.

(1) 调用式 (3) 计算任务 p_3 的自由度 $t_{DF_{p_3}} = [5, 12]$;

(2) 利用任务自由度 $t_{DF_{p_3}}$ 并结合式 (5) 可得虚拟节点 $n_{[1,2]}$ 的执行域为 $[0, 7]$ 进而求得虚拟任务的不同完成时间及其所能够达到的最大生产质量, 其计算过程如下.

当 $DL = 12$: $t_{DF_{p_1}} = [0, 7]$; $t_{DF_{p_2}} = [3, 10]$.

任务 p_2 :

$$\begin{aligned} f(2, 3) &= \max\{0.75, 0.82, 0.88\} = 0.88; \\ f(2, 4) &= \max\{0.75, 0.82, 0.88\} = 0.88; \\ f(2, 5) &= \max\{0.75, 0.82, 0.88\} = 0.88; \\ f(2, 6) &= \max\{0.75, 0.82, 0.88\} = 0.88; \\ f(2, 7) &= \max\{0.75, 0.82, 0.88\} = 0.88; \\ f(2, 8) &= \max\{0.75, 0.82, 0.88\} = 0.88; \\ f(2, 9) &= \max\{0.75, 0.82\} = 0.82; \\ f(2, 10) &= \max\{0.75\} = 0.75. \end{aligned}$$

任务 p_1 :

$$\begin{aligned} f(1, 0) &= \max\{f(2, 3) \times 0.88, f(2, 4) \times 0.90\} = 0.792; \\ f(1, 1) &= \max\{f(2, 4) \times 0.88, f(2, 5) \times 0.90\} = 0.792; \\ f(1, 2) &= \max\{f(2, 5) \times 0.88, f(2, 6) \times 0.90\} = 0.792; \end{aligned}$$

+ (P, h) 最早结束时间 \rightarrow 最晚结束时间

$f(1, 3) = \max\{f(2, 6) \times 0.88, f(2, 7) \times 0.90\} = 0.792$;
 $f(1, 4) = \max\{f(2, 7) \times 0.88, f(2, 8) \times 0.90\} = 0.792$;
 $f(1, 5) = \max\{f(2, 8) \times 0.88, f(2, 9) \times 0.90\} = 0.774$;
 $f(1, 6) = \max\{f(2, 9) \times 0.88, f(2, 10) \times 0.90\} = 0.722$;
 $f(1, 7) = \max\{f(2, 10) \times 0.88\} = 0.66$.

综上, 对于虚拟任务 $p_{[1-2]}$ 存在 5 个虚拟服务, 即:
 $\{(5, 0.66), (6, 0.722), (7, 0.774), (8, 0.792), (10, 0.957)\}$.

通过以上计算, 在工程限制期 $DL = 14$ 的约束下, 图 1 所示任务 p_1, p_2, p_3 虚拟后的任务 $p_{[1-3]}$ 最大累积生产质量 Q_3 为 0.813. = 0.957 × 0.85

5 案例分析

5.1 实验环境与实验数据

为了评估三层虚拟归约算法 T-VIRA 的性能, 课题组采取了如下的实验环境: 程序编程语言为 Java, Java 的运行环境为 java se runtime environment 8u151, 服务器选用的操作系统为 Windows7, PC 机选用 CPU 为 2.8GHz, 内存的大小为 4G. 为了使实验结论具有说服

性, 三层虚拟归约算法和传统最小关键路径算法均采用如上的软硬件配置进行调度, 具体测试的任务数据如表 1 中所示, 实验设定的检测节点 IS 达标率为 0.9, 整个工作流的工程限制期为 16.

5.2 三层虚拟归约算法分析

将表一的任务数据通过 T-VIRA 算法进行调度, 较为直观地调度步骤如图 2 所示. 具体的归约步骤为将任务 p_1, p_2, p_3 归结为虚拟任务 $p_{[1-3]}$, 将任务 p_6, p_7, p_8 归结为虚拟任务 $p_{[6-8]}$, 任务 p_4, p_5 归结为虚拟任务 $p_{[4-5]}$, 重新遍历任务集合, 形成虚拟任务 $p_{[1-8]}$, 最终以虚拟任务 $p_{[1-8]}$ 、任务 p_9 形成串行路径, 解出该路径的最大生产质量. 最终完工时间 $H_E = 16$, 所达到的最终生产质量 $Q_E = 0.399$. 最小关键路径算法调度的过程这里不再赘述, 经 CPM^[12] 得到最小完工时间 $H = 15$, 所达到的最终生产质量 $Q = 0.310$.

通过对比: 算法 T-VIRA 相对于算法 CPM 的生产质量提高了 28.7%, 显示了算法 T-VIRA 在时间约束下对生产质量产生的优化效果.

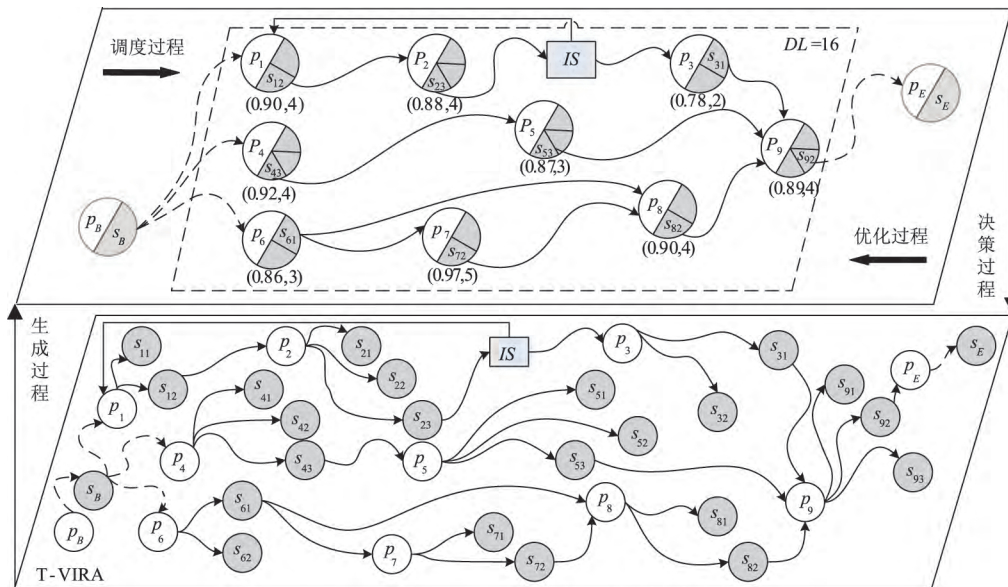


图2 案例三层 workflow 模型的归约过程

6 其他参数对算法性能的影响

由于工作流有多种类型的结构, 并且不同限制期对其调度也会具有一定的影响, 本文这一部分讨论不同的参数对于算法性能的影响.

这里采用的实验环境同实验分析中的一样, 任务对应的服务个数随机为 $[2-5]$ 间的整数, 工程限制期以 16 为基准. 图 3 显示的是任务数对于算法性能的影响, 任务为随机生成的 $[10, 20, 30, 40]$ 个结点, 工程限制期依次递增 10%, 如图 3 所示, 算法性能会随任务数

增多而降低. 图 4 和图 5 分别显示的是工程限制期和循环数目对于算法性能的影响, 任务为随机生成的 $[10, 20, 30]$ 个结点. 图 4 为保持循环数目不变的情况下, 延长工程限制期, 图 5 为保持循环数目不变的情况下, 增加循环数目, 结果分别如图 4 图 5 所示, 增加工程限制期和循环数目都能提升算法的性能.

7 结论

本文基于以限制期为约束的工作流调度, 提出了基于虚拟归约 workflow 三层决策模型的时间-质量优化算

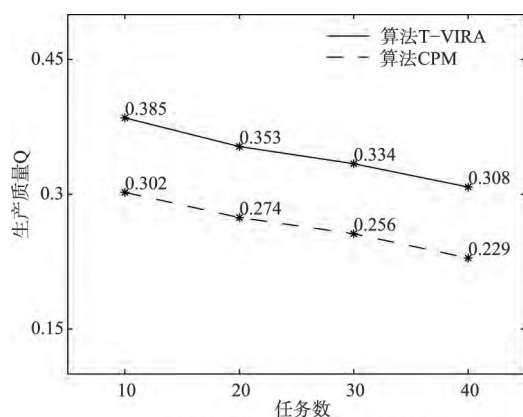


图3 不同任务数对算法T-VIRA性能的影响

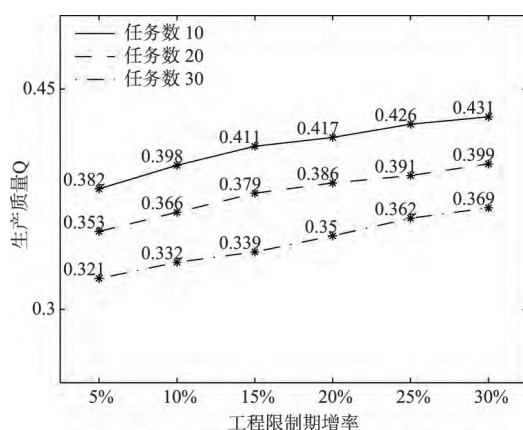


图4 不同工程限制期对算法T-VIRA性能的影响

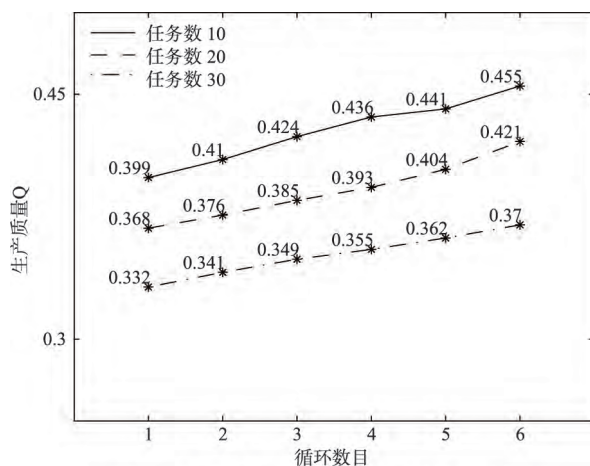


图5 循环结构数对算法T-VIRA性能的影响

法,该算法通过将 workflow 模型分层、抽象化调度过程为两大部分:检测和虚拟阶段,最后通过实验与传统最小关键路径算法进行了对比,得出该算法在提高最大生产质量方面具有一定的可行性。本文存在的不足为未对常见的科学 workflow 模型进行性能测试,笔者将在未来的研究中解决此问题。

参考文献

- [1] LIU J, PACITTI E, PATRICK V, et al. A survey of data-intensive scientific workflow management [J]. Journal of Grid Computing, 2014, 13(4): 457-493.
- [2] VERMA A, KAUSHAL S. Bi-criteria priority based particle swarm optimization workflow scheduling algorithm for cloud [A]. Recent Advances in Engineering and Computational Sciences (RAECS) [C]. Chandigarh India: IEEE, 2014. 1-6.
- [3] ARABNEJAD H, BARBOSA J G. A budget constrained scheduling algorithm for workflow applications [J]. Journal of Grid Computing, 2014, 13(4): 665-679.
- [4] KIANPISHEH S, CHARKARI N M, KARGAHI M. Reliability-driven scheduling of time/cost-constrained grid workflows [J]. Future Generation Computer Systems, 2016, 55: 1-16.
- [5] ZENG L, VEERAVALLI B, ZOMAYA A Y. An integrated task computation and data management scheduling strategy for workflow applications in cloud environments [J]. Journal of Network and Computer Applications, 2015, 50: 39-48.
- [6] MACIEJ M, KAMIL F, MARIAN B, et al. Cost optimization of execution of multi-level deadline-constrained scientific workflows on clouds [A]. Parallel Processing and Applied Mathematics [C]. Berlin: Springer, 2014, 8384: 251-260.
- [7] KIM S H, JOO K N, HA Y G, et al. A phased workflow scheduling scheme with task division policy in cloud broker [A]. International Conference on Cloudcomp [C]. Berlin: Springer, 2014, 142: 76-86.
- [8] STAVRINIDES G L, KARATZA H D. A cost-effective and qos-aware approach to scheduling real-time workflow applications in paas and saas clouds [A]. Proceedings of the 3rd International Conference on Future Internet of Things and Cloud, Rome [C]. US: ACM, 2015. 231-239.
- [9] ARABNEJAD V, BUBENDORFER K. Cost effective and deadline constrained scientific workflow scheduling for commercial clouds [A]. IEEE 14th International Symposium on Network Computing and Applications [C]. Cambridge, MA: IEEE, 2015. 106-113.
- [10] 沈虹, 李小平. 带准备时间和截止期约束的云服务 workflow 调度算法 [J]. 通信学报, 2015, 36(6): 187-196.
SHEN Hong, LI Xiao-ping. Algorithm for the cloud service workflow scheduling with setup time and deadline constraints [J]. Journal on Communications, 2015, 36(6): 187-196. (in Chinese)
- [11] 苑迎春, 李小平, 王茜, 等. 基于优先级规则的网格 workflow 调度 [J]. 电子学报, 2009, 37(7): 1457-1464.

YUAN Ying-chun ,LI Xiao-ping ,WANG Qian ,et al.
Gridworkflows schedule based on priority rules [J]. Acta
Electronica Sinica ,2009 ,37 (7) : 1457 – 1464. (in Chi-
nese)

[12] 肖鹏 胡志刚. 截止时间约束下独立网格任务的协同调

度模型 [J]. 电子学报 ,2011 ,39 (8) : 1852 – 1857.

XIAO Peng ,HU Zhi-gang. Co-scheduling model for inde-
pendent tasks with deadline constraint in computational
grid [J]. Acta Electronica Sinica ,2011 ,39 (8) : 1852 –
1857. (in Chinese)

作者简介



罗智勇(通信作者) 男 ,1978 年出生. 哈尔
滨理工大学计算机科学与技术学院副教授、硕士
生导师. 研究方向为 workflow 智能调度、网络安全.
E-mail: luozhiyong_hust@ sina. com



朱梓豪 男 ,1994 年出生. 哈尔滨理工大学
计算机科学与技术学院硕士生. 研究方向为 workflow
智能调度.
E-mail: tzuhaozhu@ gmail. com



尤 波 男 ,1962 年出生. 哈尔滨理工大学
机械动力工程学院教授、博士、博士生导师. 研究
方向为机电一体化技术、机器人控制、自动化技术
应用.
E-mail: youbo@ hrbust. edu. cn



刘嘉辉 男 ,1974 年出生. 哈尔滨理工大学
计算机科学与技术学院副教授、硕士生导师. 研
究方向为混沌理论、信息安全、并行计算.
E-mail: liujiahui@ hrbust. edu. cn