计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
49(6): 1323－1331, 2012

基于时序一致的工作流费用优化方法

刘灿灿　张卫民　骆志刚　任开军
(国防科学技术大学计算机学院　长沙　410073)
(cancanliu@nudt.edu.cn)

Temporal Consistency Based Heuristics for Cost Optimization in Workflow Scheduling

Liu Cancan, Zhang Weimin, Luo Zhigang, and Ren Kaijun
(*College of Computer, National University of Defense Technology, Changsha* 410073)

Abstract　Leveling heuristics are used to solve the time-cost trade-off problems in the grid workflow scheduling with temporal constraint by distributing the tasks into groups based on levels and scheduling them level by level. However, due to the absence of an effective method to ensure the temporal constraint, the applicability and performance of these leveling heuristics are damaged. Based on the previous heuristic deadline bottom level (DBL), an advanced heuristic referred to as temporal consistency based deadline bottom level (TCDBL) is proposed by studying the temporal properties of workflows and by optimizing their execution cost based on the temporal consistency. TCDBL satisfies the workflow temporal constraint by setting a consistent temporal point for each task, distributes the redundancy time based on the parallel degree of each level, and optimizes the workflow cost with a soft temporal constraint strategy. As a result, the workflow execution cost decreases. The experimental results in this study demonstrate that the average execution cost of TCDBL is 14% less than the cost of DBL.

Key words　temporal consistency; temporal constraint; cost optimization; workflow scheduling; bottom level

摘　要　针对效用网格下的工作流时间约束-费用优化问题,分层算法将工作流进行分层并逐层进行优化调度,取得了良好效果.然而,这类分层算法由于缺乏更有效的截止时间确定策略来保证时间约束而使得算法的适用性受限.在已有算法截止期约束的逆向分层算法(deadline bottom level, DBL)的基础上,研究工作流的时序特征,并基于任务的一致性状态对费用进行优化,提出了基于时序一致的截止期约束逆向分层算法(temporal consistency based deadline bottom level, TCDBL). TCDBL通过一致性时间点来保证时间约束,解决了DBL的适用性受限问题;同时基于各层并行度分配冗余时间,基于宽松时间约束策略进行费用优化,达到了进一步减少工作流执行费用的目标.实验结果表明TCDBL的费用优化效果比DBL改进了约14%.

关键词　时序一致性;截止期约束;费用优化;工作流调度;逆向分层

中图法分类号　TP393

收稿日期:2010-06-30;**修回日期**:2011-05-16
基金项目:国家"八六三"高技术研究发展计划基金项目(2006AA01A123);国家自然科学基金项目(60903042)

(C)1994-2024 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

工作流作为一种组织和管理大量任务协同运行的关键技术,在生产自动化和科研领域中发挥着越来越重要的推动作用,工作流在各种环境下的合理调度也随之成为该领域的一个研究热点.在目前的效用网格中,服务提供者在网格资源上部署了多个服务并提供服务的统一访问入口,同时根据服务消费的资源与服务的 QoS(quality of service)性能制定收费标准,用户按需对服务进行查询并付费使用[1].在效用网格中进行工作流调度时不仅需要考虑工作流的执行时间,同时也需要考虑其执行费用,不同目标间相互联系且相互制约,如何在给定时间内完成工作流并达到执行费用的最小化是一个 NP 难问题[2].

针对这一问题,基于遗传算法、迭代算法及粒子群等进行求解是比较有效的方法[3-5],然而这类算法在工作流规模和服务规模较大时通常具有较大的时间开销.与这类算法相比,基于时间分配的启发式由于具有合理的时间开销而被广泛采用,这类算法按一定策略将截止时间(即约束时间)分配到各任务,任务基于分配的时间进行费用优化:如 TD(time distribution)算法基于任务长度在路径中所占的比例来分配时间[6];DBL(deadline bottom level)与 DTL(deadline top level)分别基于任务的逆向深度和正向深度对工作流进行分层,并按层进行时间分配[7],任务在分配的时间内进行费用优化;BSRD(backward series reduction with deadline)和 FSRD(forward series reduction with deadline)则在 DBL 和 DTL 的基础上,对工作流中的可归约串进行串归约以进一步优化费用[8].这些启发式算法均取得了较好的调度效果,然而也都存在一些缺陷:首先,算法在进行时间分配时需考虑工作流的时间约束目标,必须保证任务分配的时间大于其最小完成时间,因此当工作流的约束时间较小时(小于工作流的最小分层完成时间),算法不适用;其次,在时间分配过程中没有考虑各层并行度(即任务数)之间的差异,将冗余时间在各层进行平均分配,显然,为并行度较高的层分配较大的费用优化空间将有效改善整个工作流的费用优化效果;此外,在费用优化过程中采用严格时间约束策略将任务完成时间严格限制在约束时间之内,当某个候选服务其完成时间稍有超出但费用能得到较大优化时,不能进行更有效的费用优化.

针对这些不足,本文研究工作流的时序特征,为任务设置一致性时间点并基于任务的一致性状态进行费用优化,在 DBL 的基础上提出基于时序一致的截止期约束逆向分层算法(temporal consistency based deadline bottom level, TCDBL).TCDBL 的主要贡献如下:第一,将时间约束目标从时间分配过程中分离出来,通过任务的一致性时间点来保证工作流的时间约束,解决了 DBL 中由于缺乏有效的截止时间确定策略而使得算法适用性受限的问题;第二,在时间分配过程中考虑各层任务的并行度,为任务数较多的层分配较大的费用优化空间,整体上提高了工作流的费用优化效果;第三,在费用优化过程中采用宽松时间约束策略在任务约束时间的两侧选择服务,减少了执行过程中产生的"时间碎片",达到了进一步提高算法性能的目标.

1 时间约束-费用优化问题描述

工作流(W)通常通过有向无环图(directed acrylic graph, DAG)进行描述:$G=\{V,E\}$,其中 $V=\{1,2,\cdots,n\}$ 表示 W 中的任务集;而 $E=\{(i,j)\mid i,j\in V\}$ 表示任务间的依赖关系,$(i,j)\in E$ 表示任务 i 与 j 的依赖关系,同时称 i 为 j 的前驱,j 为 i 的后继;定义 i 的多个前驱为前驱列表,记为 $pre(i)$,而 i 的多个后继为后继列表,记为 $succ(i)$.此外,定义 W 中不存在任何前驱的任务为开始任务,不存在任何后继的任务为结束任务;设定 W 中只有一个开始任务(记为 1)和一个结束任务(记为 n).对于工作流中的任意任务 i,在调度空间中均存在一个对应的候选服务列表,记为 $S(i)$,服务列表的长度(即服务个数)为 $l(i)$.同一列表中不同候选服务的执行时间和执行费用各不相同,设定 $s_{ik}=(\tau_{ik},c_{ik})$ 表示 $S(i)$ 中的第 $k(0\leqslant k<l(i))$ 个服务,其中 τ_{ik} 与 c_{ik} 分别表示 s_{ik} 的执行时间与执行费用,$S(i)$ 中的服务均按执行时间从小到大排列,同时设定执行时间较长的服务执行费用较低.

工作流的调度过程即为各任务在对应的候选服务列表中选择一个服务,假定所有服务都能提供与承诺一致的服务质量.工作流中所有任务的服务均选定之后,工作流的执行时间与执行费用便确定.工作流的执行时间为从 1 到 n 的最长路径的长度(路径长度为该路径上所有任务的执行时间之和);执行费用为所有任务的执行费用之和.本文的调度目标即为 V 中的任意任务 i 在对应的服务列表中 $S(i)$ 中选择一个服务 s_{ik},使得工作流的执行时间在约束时间(记为 δ)之内,执行费用达到最小,该问题形式化描述为

(C)1994-2024 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$\min \sum_{i=1}^{n}\sum_{k=0}^{l(i)-1}(c_{ik}\times x_{ik}), \tag{1}$$

$$\text{s.t.}\quad \beta_n \leqslant \delta, \tag{2}$$

$$\beta_j-\beta_i-\sum_{k=0}^{l(j)-1}\tau_{jk}\times x_{jk}\geqslant 0,\forall (i,j)\in E, \tag{3}$$

$$\sum_{k=0}^{l(i)-1}x_{ik}=1,\forall i\in V, \tag{4}$$

$$x_{ik}\in\{0,1\},\forall i\in V,0\leqslant k<l(i), \tag{5}$$

其中,x_{ik}为布尔函数,当 i 选择 s_{ik} 时,$x_{ik}=1$,否则 $x_{ik}=0$;β_i 为任务 i 的完成时间;式(1)描述了工作流的费用优化目标;式(2)描述了工作流的时间约束目标,其中 β_n 为结束任务 n 的完成时间;式(3)描述了工作流的偏序关系;式(4)(5)表示每个任务只能选择且必须选择一个服务执行.

2 时序一致性

工作流的时间约束目标要求在给定的约束时间内完成工作流,该问题本质上是一个时序一致性问题.目前国内外很多学者针对工作流的时序一致性进行了大量研究[9-11],在已有工作的基础上,本文研究工作流的时序一致性,并基于任务的一致性状态进行费用优化.首先为任务定义一致性状态:

定义 1. 一致性状态. 在调度过程中,对于任务 i,如果其开始时间(记为 α_i)满足:

1) $\alpha_i\leqslant\delta-D(i)$,称 i 为强一致状态(strong consistency, SC);

2) $\delta-D(i)<\alpha_i\leqslant\delta-d(i)$,称 i 为弱一致状态(weak consistency, WC);

3) $\alpha_i>\delta-d(i)$,称 i 为不一致状态(inconsistency, IC).

其中,$D(i)$与 $d(i)$分别为从任务 i 到结束任务 n 的最大可能路径长度和最小可能路径长度,通过逆向宽度优先方法求解:

$$D(i)=\begin{cases}\max\limits_{0\leqslant k<l(i)}\tau_{ik}, & i=n,\\ \max\limits_{j\in succ(i)}D(j)+\max\limits_{0\leqslant k<l(i)}\tau_{ik}, & \text{else.}\end{cases} \tag{6}$$

$$d(i)=\begin{cases}\min\limits_{0\leqslant k<l(i)}\tau_{ik}, & i=n,\\ \max\limits_{j\in succ(i)}d(j)+\min\limits_{0\leqslant k<l(i)}\tau_{ik}, & \text{else.}\end{cases} \tag{7}$$

易知 $D(1)$为工作流的最大可能长度(记为 $D(W)$),$D(W)=D(1)$;而 $d(1)$为工作流的最小可能长度(记为 $d(W)$),$d(W)=d(1)$.调度过程中不考虑算法运行开销及任务间的数据传输开销,因此定义 1 中任务 i 的开始时间 α_i 即为 i 的就绪时间:

$$\alpha_i=\begin{cases}0, & i=1,\\ \max\limits_{j\in pre(i)}\beta_j, & \text{else.}\end{cases} \tag{8}$$

在定义 1 的基础上定义任务的一致性时间点:

定义 2. 一致性时间点.

1) 当 i 在某一时间点之前执行完成便能保证其后继(包括间接后继)均为 SC,则称该时间点为 i 的强一致时间点,记为 $\delta_{SC}(i)$;

2) 当 i 在某一时间点之前执行完成便能保证其后继(包括间接后继)可不为 IC,则称该时间点为的弱一致时间点,记为 $\delta_{WC}(i)$.

由定义 2 可知,当 $\beta_i<\delta_{SC}(i)$时,i 的所有后继在对应的候选服务列表中选择任意服务执行时均能满足工作流的时间约束,因此这种情况下为各后继选择执行费用最小的候选服务即可;而当 $\beta_i>\delta_{WC}(i)$时,i 的所有后继全选择执行时间最小的服务执行时也无法满足工作流的时间约束目标,因此这种情况下不存在满足时间约束的有效解.由此可见,为满足工作流的时间约束目标,在调度过程中对任务 i 进行调度时需满足以下条件:

$$\beta_i\leqslant\delta_{WC}(i),\forall i\in V. \tag{9}$$

3 基于时序一致的费用优化算法

基于以上定义的一致性状态和一致性时间点,易知当 δ 满足:

1) $\delta<d(W)$,不存在满足时间约束的调度;

2) $d(W)\leqslant\delta<D(W)$,执行费用位于最小费用($\sum_{i=1}^{n}c_{i(l(i)-1)}$)与最大费用($\sum_{i=1}^{n}c_{i0}$)之间;

3) $\delta\geqslant D(W)$,存在费用最小的调度,最小执行费用为 $\sum_{i=1}^{n}c_{i(l(i)-1)}$.

满足条件 1),2)的工作流调度都比较简单:当满足 1)时,向用户提示无法找到满足时间约束的调度;当满足条件 3)时,为每个任务 $i(i\in V)$选择费用最小的候选服务 $s_{i(l(i)-1)}$即可.本节主要对满足条件 2)的工作流调度进行研究,在 DBL 的基础上提出基于时序一致的截止期约束逆向分层算法 TCDBL,该算法主要包括时间分配与费用优化两个阶段.

3.1 基于逆向分层的时间分配

合理的时间分配是工作流费用优化的关键.DBL 基于任务的逆向深度(任务 i 与结束任务 n 之间的最大边数)对工作流进行分层,为各层分配约束时间并逐层进行费用优化.与 DBL 类似,本节基于逆向分层策略进行时间分配,具体步骤如下:

(C)1994-2024 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

步骤 1. 通过逆向宽度优先方法计算任务的逆向深度 BD:

$$BD(i)=\begin{cases}0, & i=n,\\ \max\limits_{j\in succ(i)} BD(j)+1, & \text{else}.\end{cases} \tag{10}$$

易知开始任务的逆向深度最大,记为 $BD(1)$,结束任务的逆向深度最小、为 0.

步骤 2. 基于逆向深度对工作流进行分层,其中第 k 层任务 $BL(k)$包括所有逆向深度为 k 的任务:

$$BL(k)=\{i \mid BD(i)=k, i\in V\}. \tag{11}$$

步骤 3. 计算各层的最小分层完成时间 $\theta_{\mathrm{BL}}(k)$及工作流的最小分层完成时间 $\theta_{\mathrm{BL}}(W)$:

$$\theta_{\mathrm{BL}}(k)=\sum_{m=k}^{BD(1)}\max_{j\in BL(m)}\{\min_{0\leqslant j<l(i)}\tau_{ij}\}. \tag{12}$$

$$\theta_{\mathrm{BL}}(W)=\theta_{\mathrm{BL}}(0). \tag{13}$$

步骤 4. 将冗余时间在各层中按权进行分配(权值为各层任务的并行度),各层的层约束时间 $\delta_{\mathrm{BL}}(k)$为

$$\delta_{\mathrm{BL}}(k)=\theta_{\mathrm{BL}}(k)+(\delta-\theta_{\mathrm{BL}}(W))\times\frac{\sum_{i=k}^{BD(1)}|BL(i)|}{\sum_{i=0}^{BD(1)}|BL(i)|}, \tag{14}$$

其中,$|BL(i)|$为 $BL(i)$中的任务个数,即第 i 层的并行度;$\sum_{i=0}^{BD(1)}|BL(i)|$为工作流的任务总数. 基于各层任务的并行度对冗余时间进行分配将为任务数较多的层分配较大的费用优化空间,从而从整体上改善工作流的费用优化效果.

步骤 5. 将任务的约束时间设为该层任务的层约束时间:

$$\delta(i)=\delta_{\mathrm{BL}}(k), i\in BL(k), \tag{15}$$

其中,$\delta(i)$为任务 i 的约束时间. 与本文方法不同的是,DBL 在时间分配过程中需要考虑工作流的时间约束目标,必须保证各任务在分配的约束时间内均能找到至少一个满足时间约束的候选服务,因此当 $\delta<\theta_{\mathrm{BL}}(W)$时,算法由于缺乏有效策略来保证时间约束目标,因此算法不适用. 其次,当 $\delta>\theta_{\mathrm{BL}}(W)$时,DBL 将冗余时间在各层中进行平均分配,忽略了各层任务数的差异,显然,为任务数较多的层分配较大的费用优化空间将有效改善工作流的费用优化效果.

3.2 基于宽松时间约束的费用优化

为保证工作流的时间约束目标,DBL 在费用优化过程中采用严格时间约束策略将任务的完成时间严格限制在约束时间之内,在此条件下选择执行费用最小的服务进行调度. 与 DBL 不同的是,本文通过弱一致时间点来保证工作流的时间约束目标(见式(9)),因此在费用优化过程中不一定将任务的完成时间严格限制在约束时间之内. 本文基于式(16)选择服务:

$$s_{ik}=\{s_{ij} \mid \min_{s_{ij}\in S'(i)}\{|\delta(i)-\beta_{ij}|\}\}, \tag{16}$$

其中,β_{ij}为任务 i 选择服务 s_{ij} 的完成时间;$S'(i)$为 $S(i)$中满足式(9)的服务列表,由于 $\delta\geqslant d(W)$,且在调度过程中对 i 的前驱进行调度时保证 i 不为 IC,因此 $S'(i)$中至少存在一个可选服务. 此外,基于式(16)进行费用优化时,能使得同层任务的完成时间与该层约束时间之间的方差达到最小,进一步减少工作流执行过程中产生的“时间碎片”,有效利用费用优化空间并达到减少工作流执行费用的目标.

3.3 TCDBL 算法描述与复杂性分析

基于以上时间分配和费用优化策略,本节提出基于时序一致的截止期约束逆向分层算法 TCDBL,算法描述如下:

算法 1. TCDBL.

输入:工作流 W 及各任务的候选服务列表 $S(i)$、约束时间 δ;

输出:执行费用 C.

1) 按式(6)与式(7)计算 $D(W)$与 $d(W)$;

2) if δ 满足:

① $\delta<d(W)$,无法找到满足时间约束的调度,返回约束时间过小的消息;

② $d(W)\leqslant\delta<D(W)$, $C=sub\_proc(\delta,W)$;

③ $\delta\geqslant D(W)$, $C=\sum_{i=1}^{n}c_{i(l(i)-1)}$;

3) 返回 C.

sub_proc 描述如下.

$sub\_proc(\delta,W)$:

① for $\forall i\in V$,计算 $\delta_{\mathrm{SC}}(i)$, $\delta_{\mathrm{WC}}(i)$及 $\delta(i)$;

② $RL=\{1\}$, $FL=\varnothing$, $C=0$;/* RL 为就绪任务列表, FL 为完成任务列表 */

③ 在 RL 中选择一个任务 i;

④ if(i 为 SC)

⑤ $s_{ik}=s_{i(l(i)-1)}$;

⑥ else

⑦ 在 $S(i)$中查找满足式(9)的服务列表,记为 $S'(i)$;

⑧ 按式(16)选择服务 s_{ik};

⑨ endif

(C)1994-2024 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

⑩ 调度 s_{ik} 执行并更新 RL; /* 从 RL 中移除 i, 并将 $succ(i)$ 中的就绪任务加入 RL 中 */

⑪ $C=C+c_{ik}$;

⑫ $FL=FL+i$;

⑬ if $FL\neq V$,转步骤③;

⑭ 返回 C.

sub_proc 中语句①对工作流中的所有任务进行遍历,在遍历过程中由于需要检查前驱或后继列表,最坏情况下时间复杂度为 $O(n^2)$,n 为工作流规模. if 结构(语句④~⑨)为任务选择服务,该过程需对所有候选服务进行遍历,因此时间复杂度为 $O(m)$,m 为服务规模;语句⑩对后继列表进行遍历,时间复杂度为 $O(n)$;其他语句则为常数时间. 通常情况下,$n\gg m$,因此调度一个任务的时间复杂度为 $O(n)$,调度过程需对所有任务进行遍历,因此 *sub_proc* 的时间复杂度为 $O(n^2)$.

算法 1 中步骤①,③的任务调度都比较简单,步骤③的时间复杂度为 $O(n)$,步骤①为常数时间;步骤②的时间复杂度同 *sub_proc*,为 $O(n^2)$. 由于 $n\gg m$,因此 TCDBL 的时间复杂度为 $O(n^2)$.

4 实验结果与分析

为了对 TCDBL 的性能进行全面评估,本文在模拟实验环境中实现了 TCDBL,TCDTL(temporal consistency based deadline top level),DBL,DTL 及 ATD(advanced time distribution)这 5 种算法. TCDTL 算法思想与 TCDBL 基本相同,但基于任务的正向深度进行分层,将具有同步开始特征的任务分配到同一层进行费用优化. ATD 基于 TD 进行改进,费用优化过程中将任务的开始时间设为前驱任务的完成时间而非截止时间,有利于进一步利用费用优化空间进行费用优化. 所有算法均采用 Java 编程,运行于内存为 2 GB、速度为 2.53 GHz 的双核处理器上.

4.1 实例设计

实验针对多个工作流实例在不同调度空间中的调度性能进行比较. 工作流实例由工作流随机生成器生成,各任务的出入度为[1~7]之间的随机数,生成过程中对工作流规模(workflow size, WS)进行控制,设定 $WS=\{200,400,600,800,1\,000\}$;各任务的候选服务列表也随机生成,生成过程中对最大服务列表的规模(service size, SS)进行控制,设定 $SS=\{10,20,30,40\}$;同时设定各候选服务的执行时间为[5~60]之间的随机数,执行费用为执行时间的函数,实验过程中分别对费用函数(cost function, CF)为凸函数 concave(ccv)、凹函数 convex(cvx)及混合函数 hybrid(hyb)这 3 种情况进行比较,并保证执行时间较长的服务执行费用较低;此外,实验过程中对不同约束时间下的算法性能进行比较,约束时间设置如下:$\delta=d(W)+\omega\times(D(W)-d(W))$,$\omega$ 以 0.05 为增量在[0~0.6]之间进行取值. 实验过程中总共进行的实验组数为 $5\times4\times3\times13=780$.

比较结果通过最优解比例(best proportion, BP)、与最优解之间的平均偏差(average difference to best, ADB,以下简称平均偏差)、平均执行费用(average cost, AC)及 TCDBL 相对于各算法平均费用的改进比(improvement ratio of average cost, IRAC)进行衡量;此外,通过算法的平均运行时间(average runtime, ART)比较算法效率. 各性能度量指标为

$$BP=\frac{N_B(H)}{N};\tag{17}$$

$$ADB=\frac{\sum_{i=1}^{N}\frac{C_i(H)-C_i(B)}{C_i(B)}}{N};\tag{18}$$

$$AC=\frac{\sum_{i=1}^{N}C_i(H)}{N};\tag{19}$$

$$IRAC=\frac{AC(H)-AC(TCDBL)}{AC(H)};\tag{20}$$

$$ART=\frac{\sum_{i=1}^{N}RT_i(H)}{N};\tag{21}$$

其中,H 代表不同的算法,$N_B(H)$为该组实验中 H 的目标函数值最小的实例个数,N 为总的实验组数;$C_i(H)$为第 i 个实例中 H 的目标函数值,$C_i(B)$为该组实例中目标函数的最小值,$AC(H)$为所有实例中 H 的平均执行费用;$RT(H)$为 H 的算法开销.

4.2 DBL 与 DTL 的算法适用性分析

由以上分析可知,DBL 在算法适用性方面存在局限:当 $\delta<\theta_{BL}(W)$ 时,DBL 由于无法保证所有任务均能找到满足时间约束的服务,因此算法不适用,此时采用最小关键路径法(minimum critical path, MCP)[7]进行调度;同理,在 DTL 中,当 $\delta<\theta_{TL}(W)$ 时($\theta_{TL}(W)$为基于正向分层的工作流最小分层完成时间),DTL 也采用 MCP 算法进行调度. MCP 为关键路径上的任务选择最快服务进行调度,仅对非关键路径上的任务进行费用优化,因此算法性能比较

(C)1994-2024 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

差;由此可见,DBL 和 DTL 的算法性能与 $\theta_{BL}(W)$ 和 $\theta_{TL}(W)$ 紧密相关;通过 $\theta_{BL}(W)$ 与 $\theta_{TL}(W)$ 在 $[d(W),D(W)]$ 中所处的位置 $\omega_{BL}(\theta)$ 和 $\omega_{TL}(\theta)$ 对算法适用性进行分析:

$$\omega_{BL}(\theta)=\frac{\theta_{BL}(W)-d(W)}{D(W)-d(W)}, \tag{22}$$

$$\omega_{TL}(\theta)=\frac{\theta_{TL}(W)-d(W)}{D(W)-d(W)}. \tag{23}$$

表 1 与表 2 分别为不同工作流规模和服务规模下 DBL 和 DTL 的算法适用性分析情况. 由两表可见,两种算法的平均适用范围为 $\omega>0.23$,当 $\omega<0.23$ 时,DBL 与 DTL 将采用 MCP 算法进行调度,费用优化效果比较差.

Table 1 Comparison of Applicability for DBL and DTL under Different Workflow Sizes

表 1 不同工作流规模下 DBL 和 DTL 的算法适用性比较

WS	$\omega_{BL}(\theta)$	$\omega_{TL}(\theta)$
200	0.18	0.17
400	0.2	0.26
600	0.27	0.26
800	0.23	0.22
1000	0.26	0.23
ave	0.23	0.23

Table 2 Comparison of Applicability for DBL and DTL under Different Service Sizes

表 2 不同服务规模下 DBL 和 DTL 的算法适用性比较

SS	$\omega_{BL}(\theta)$	$\omega_{TL}(\theta)$
10	0.31	0.28
20	0.21	0.2
30	0.21	0.21
40	0.18	0.17
ave	0.23	0.22

4.3 结果比较与分析

在对 DBL 和 DTL 的算法适用性进行分析的基础上,本节对算法性能进行比较,表 3 为所有问题参数下各种算法的性能比较情况. 总结来看,TCDBL 的性能最优,其最优解比例为 0.99,平均偏差仅为 0.000 2,而平均执行费用比其他几种算法则改进了 10%~18%不等;这是由于 TCDBL 基于任务的一致性状态进行费用优化,同时基于各层任务数对冗余时间进行分配,为任务数较多的层分配了较大的费用优化空间,整体上提高了工作流的费用优化效果;而基于宽松时间约束策略进行服务选择也使得调度过程中产生的"时间碎片"进一步减少,达到了进一步利用费用优化空间进行费用优化的目标. 同时,从平均偏差和相对于 TCDBL 的平均费用改进比来看,TCDTL 的算法性能稍优于另外 3 种算法,TCDBL 相对于 TCDTL 改进了 10%,而相对于 DBL,DTL 和 ATD 则分别改进了 14%,18%及 16%,这是由于 TCDTL 基于工作流的时序一致性状态进行费用优化,与算法适用性可能受限的 DBL 和 DTL 相比,TCDTL 在约束时间较小时算法性能比较好. 此外,从算法开销来看,各算法的开销都非常小,平均开销仅为 14 ms,这表明各算法均具有非常高的算法效率.

Table 3 Comparison of All Problem Parameters Obtained from Five Heuristics

表 3 所有问题参数下的算法性能比较

H	*BP*/%	*ADB*	*AC*	*IRAC*/%	*ART*/ms
TCDBL	99	0.0002	6461	0	14.04
TCDTL	0	0.1052	7213	10	13.73
DBL	1	0.1377	7494	14	13.97
DTL	0	0.2068	7896	18	13.71
ATD	0	0.1938	7708	16	15.26

4.4 不同问题参数下的算法性能比较

为了对算法性能进行全面评估,以下在不同问题参数下(包括约束时间、工作流规模、服务规模和费用函数)对各算法进行比较.

4.4.1 不同约束时间下的算法性能比较

图 1 为各算法得到的平均执行费用随不同约束时间的变化情况:

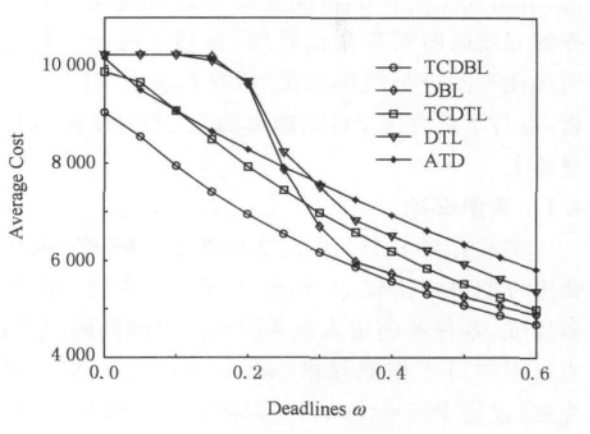

Fig. 1 Comparison of average cost for each heuristic as a function of deadlines.

图 1 各算法随不同约束时间的执行费用曲线图

(C)1994-2024 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

如图1所示,在不同约束时间下,TCDBL的平均执行费用均最小.当约束时间δ较小($\omega<0.25$)时,DBL与DTL适用性受限,采用MCP进行调度时产生的平均执行费用远高于其他3种算法;随着δ的增长,当$\omega>0.25$时,DBL与DTL的适用条件得到满足,其平均执行费用迅速降低,但仍分别稍高于TCDBL与TCDTL平均执行费用.这是由于基于时序一致的费用优化算法(TCDBL与TCDTL)基于各层任务数进行冗余时间分配,为多数任务分配了较大的费用优化空间,同时采用了宽松时间约束策略进行服务选择,进一步减少了执行过程中产生的"时间碎片",因此算法性能整体上有所提高.同时,由图1可知,基于逆向分层的费用优化算法(TCDBL和DBL)性能整体上优于基于正向分层策略的费用优化算法(TCDTL和DTL),这是由于逆向分层策略基于工作流的同步完成特征将具有相同后继的任务分配到同一层进行费用优化,与基于任务同步开始特征的正向分层策略相比,较大限度地利用了费用优化空间,因此算法性能比较好.此外,当$\omega<0.25$时,ATD的性能优于DBL与DTL,但当$\omega>0.25$时,ATD的算法性能在5种算法中最差.

4.4.2 不同工作流规模下的算法性能比较

表4为不同工作流规模下的算法比较情况.由表可见,TCDBL的性能总体上优于其他算法.在各种工作流规模下,TCDBL的最优解比例保持在97%以上,而平均偏差也保持在0.001以下,这表明,TCDBL算法适用于各种规模的工作流调度.同时,从最优解比例来看,DBL的性能稍优于另外3种算法;而从平均偏差来看,DBL的性能则稍劣于TCDTL,这是由于DBL的适用性受约束时间的限制;此外,DBL和DTL的平均偏差随工作流规模的增长有所增长,这是由于随着工作流规模的增长,DBL和DTL的算法适用性有所降低,因此算法性能也有所降低.从算法开销来看,随着工作流规模的增长,各算法的运行开销均有所增长,当$WS=1\,000$时,各算法的开销为40 ms左右.

Table 4 Comparison of Five Heuristics at Different Workflow Sizes

表4 不同工作流规模下的算法性能比较

WS	*BP*/%					*ADB*					*ART*/ms				
	200	400	600	800	1000	200	400	600	800	1000	200	400	600	800	1000
TCDBL	100	98	97	99	99	0	0.0001	0.0007	0	0	1.33	1.94	8.6	18.17	40.18
TCDTL	0	0	0	0	0	6.6	0.1023	0.1201	0.1313	0.1062	0.93	2.04	8.12	17.99	39.57
DBL	0	2	2	1	1	0.1135	0.1254	0.1564	0.1444	0.1486	1.24	2.04	7.93	19.19	39.28
DTL	0	0	0	0	0	0.1597	0.1988	0.2263	0.2346	0.2146	1.14	1.74	7.92	18.07	39.67
ATD	0	0	1	0	0	0.1582	0.1894	0.1872	0.2154	0.2190	1.44	3.22	9.53	19.91	42.18

4.4.3 不同服务规模下的算法性能比较

表5为不同服务规模下的算法比较情况.在各种服务规模下,TCDBL的最优解比例均保持在96%以上,而平均偏差也保持在0.001以内,同时,各种服务规模下的平均费用比其他算法改进了9%~19%不等,这表明TCDBL在各种服务规模下的调度性能均比较好.而各算法的平均执行费用随服务规模的增长均有一定程度的减少,这是由于服务规模的增长使得各任务的可选服务增加,各候选服务间执行时间的差距减少,在费用优化过程中产生的"时间碎片"减小,因此费用优化效果有所提高.同时,从平均偏差来看,DBL和DTL的平均偏差随服务规模的增长有所降低,这是由于DBL和DTL的算法适用性随服务规模的增长有所提高.

Table 5 Comparison of Five Heuristics at Different Service Sizes

表5 不同服务规模下的算法性能比较

SS	*BP*/%				*ADB*				*AC*				*IRAC*/%			
	10	20	30	40	10	20	30	40	10	20	30	40	10	20	30	40
CDBL	96	100	99	99	0.0002	0	0	0.0005	6911	6406	6251	6275	0	0	0	0
TCDTL	0	0	0	0	0.0947	0.1041	0.1161	0.1057	7623	7129	7058	7043	9	10	11	11
DBL	4	0	1	0	0.1615	0.1268	0.1395	0.1228	8132	7403	7264	7179	15	13	14	13
DTL	0	0	0	0	0.2272	0.1966	0.2014	0.2020	8433	7744	7740	7668	18	17	19	18
ATD	0	0	0	1	0.1498	0.1799	0.2279	0.2177	7962	7538	7689	7642	13	15	19	18

(C)1994-2024 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

4.4.4 不同费用函数下的算法性能比较

表 6 为不同费用函数下的算法比较情况. 从最优解比例与平均偏差来看, TCDBL 在不同费用函数下的费用优化效果都比较稳定, 最优比均为 98% 以上, 而平均偏差仍保持在 0.001 以内. 从平均费用来看, 各算法在凸函数中的平均费用均高于凹函数中的平均费用, 这是由于凸函数中各任务的平均费用总体上高于凹函数中各任务的平均费用.

Table 6 Comparison of Five Heuristics Using Different Cost Functions

表 6 不同费用函数下的算法性能比较

CF	*BP*/%			$10^2 \times ADB$			*AC*			*IRAC*/%		
	ccv	cvx	hyb	con	cov	hyb	ccv	cvx	hyb	ccv	cvx	hyb
TCDBL	98	99	99	0.01	0.03	0	7 140	5 748	6 494	0	0	0
TCDTL	0	0	0	12.47	8.31	10.78	8 114	6 266	7 260	12	8	11
DBL	2	1	1	12.67	14.89	13.74	8 173	6 771	7 540	13	15	14
DTL	0	0	0	21.99	19.17	20.89	8 760	6 984	7 945	18	18	18
ATD	0	0	0	21.93	16.26	19.95	8 658	6 696	7 769	18	14	16

5 结 论

效用网格中工作流的时间约束-费用优化调度是一个多项式时间内难以求解的 NP 难问题. 本文研究工作流的时序特征并基于任务的一致性状态进行费用优化, 在已有算法 DBL 的基础上提出一种新的算法 TCDBL. TCDBL 为任务设置一致性时间点来保证工作流的时间约束目标, 解决了 DBL 由于缺乏有效的截止时间确定策略带来的适用性受限问题; 同时, 基于各层任务的并行度进行时间分配, 为并行度较高的任务层分配较大的费用优化空间, 整体上提高了费用优化效果; 此外, 在费用优化过程中采用宽松时间约束策略在任务约束时间两侧选择服务, 进一步减少了执行过程中产生的“时间碎片”, 改善了工作流的费用优化效果. 模拟结果证明了 TCDBL 算法的优越性.

参 考 文 献

[1] Buyya R, Abramson D, Jonathan G, et al. Economic models for resource management and scheduling in grid computing [J]. Concurrency and Computation: Practice and Experience, 2004, 14(13-15): 1507-1542

[2] Wieczoreka M, Hoheisel A, Prodan R. Towards a general model of the multi-criteria workflow scheduling on the grid [J]. Future Generation Computer Systems, 2009, 25(3): 237-256

[3] Garg S, Konugurthi P, Buyya R. A linear programming driven genetic algorithm for meta-scheduling on utility grids [C] //Proc of the 16th Int Conf on Advanced Computing and Communication (ADCOM'08). Piscataway, NJ: IEEE, 2008: 19-26

[4] Yuan Yingchun, Li Xiaoping, Wang Qian, et al. Grid workflows schedule based on priority rules [J]. Acta Electronica Sinica, 2009, 37(7): 1457-1464 (in Chinese)

(苑迎春, 李小平, 王茜, 等. 基于优先级规则的网格工作流调度[J]. 电子学报, 2009, 37(7): 1457-1464)

[5] Zhang Xiaodong, Li Xiaoping, Wang Qian, et al. Hybrid particle swarm optimization algorithm for cost minimization in service-workflows with due dates [J]. Journal on Communications, 2008, 29(8): 87-94 (in Chinese)

(张晓东, 李小平, 王茜, 等. 服务工作流的混合粒子群调度算法[J]. 通信学报, 2008, 29(8): 87-94)

[6] Yu J, Buyya R, Ramamohanarao K. Workflow Scheduling Algorithms for Grid Computing [G] //LNCS 146: Metaheuristics for Scheduling in Distributed Computing Environments. Berlin: Springer, 2008: 173-214

[7] Yuan Yingchun, Li Xiaoping, Wang Qian, et al. Bottom level based heuristic for workflow scheduling in grids [J]. Chinese Journal of Computers, 2008, 31(2): 283-290 (in Chinese)

(苑迎春, 李小平, 王茜, 等. 基于逆向分层的网格工作流调度算法[J]. 计算机学报, 2008, 31(2): 282-290)

[8] Yuan Yingchun, Li Xiaoping, Wang Qian. Cost optimization heuristics for grid workflows scheduling based on serial reduction [J]. Journal of Computer Research and Development, 2008, 45(2): 246-253 (in Chinese)

(苑迎春, 李小平, 王茜. 基于串归约的网格工作流费用优化方法[J]. 计算机研究与发展, 2008, 45(2): 246-253)

(C)1994-2024 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

[9] Chen J, Yang Y. Adaptive selection of necessary and sufficient checkpoints for dynamic verification of temporal constraints in grid workflow systems [J]. ACM Trans on Autonomous and Adaptive Systems, 2007, 2(2): 1-25

[10] Wang Yuan, Fan Yushun. A method of time constraint workflow model analysis and verification [J]. Journal of Software, 2007, 18(9): 2153-2161 (in Chinese)
(**王远, 范玉顺. 工作流时序约束模型分析与验证方法**[J]. **软件学报**, 2007, 18(9): 2153-2161)

[11] Du Yanhua, Fan Yushun. A sprouting graph based approach to dynamic check the temporal consistency of workflow multi-process [J]. Acta Electronica Sinica, 2009, 37(10): 2181-2187 (in Chinese)
(**杜彦华, 范玉顺. 基于生成图的工作流多过程动态时序一致性验证方法**[J]. **电子学报**, 2009, 37(10): 2181-2187)

Liu Cancan, born in 1980. PhD. Student member of China Computer Federation. Her main research interests includes high performance computing and scientific workflow.

Zhang Weimin, born in 1966. Professor and PhD supervisor in the National University of Defense Technology. His main research interests include distributed computing and grid computing (wmzhang104@tom.com).

Luo Zhigang, born in 1962. Professor and PhD supervisor in the National University of Defense Technology. His main research interests include parallel and distributed computing (zgluo@nudt.edu.cn).

Ren Kaijun, born in 1975. PhD and associate professor in the National University of Defense Technology. His main research interests include Web service composition and semantic service (renkaijun@nudt.edu.cn).

(C)1994-2024 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net