ISSN 1000-1239/CN 11-1777/TP
49(7): 1593-1600, 2012

正文排版44行，双栏，每栏22字

基于改进优先级规则的工作流费用优化方法 题目三号

刘灿灿　张卫民　骆志刚　任开军 作者四号
(国防科学技术大学计算机学院　长沙　410073) 单位小五号
(cancanliu@nudt.edu.cn)

英文题目四号

Workflow Cost Optimization Heuristics Based on Advanced Priority Rule

Liu Cancan, Zhang Weimin, Luo Zhigang, and Ren Kaijun 姓名五号
(*College of Computer, National University of Defense Technology, Changsha* 410073) 单位小五号

Abstract The time and cost trade-off problem in workflow scheduling with deadline constraint in utility grids is an NP-hard, but attractive problem. In the past studies, the priority rule, best fit based on time-dependent coupling strength (referred as BFTCS), was adopted to solve this question because BFTCS considers both the features of resources and workflow structures and uses them as two important factors in the iterative heuristic for cost optimization. However, BFTCS did not consider another important factor: the temporal feature of the workflow. In this paper, we define temporal mobility (TM) to describe the temporal feature of the workflow, and incorporate TM into the iterative stage as an important factor to improve the workflow execution cost further. This novel priority rule (referred as best fit based on time-dependent coupling strength and temporal mobility, BFTCSTM) gives priority to the tasks with larger TM and smaller TCS (time-dependent coupling strength) during the iterative stage. Therefore, compared with BFTCS, BFTCSTM slows down the increasing rate of the workflow execution time greatly, and provides more opportunities to other tasks to optimize cost, which decreases the total cost further. The experimental results demonstrate the advantages of BFTCSTM based iterative heuristic. 摘要小五号

Key words workflow scheduling; utility grid; cost optimization; time-dependent coupling strength (TCS); temporal mobility (TM) 关键词小五号

摘　要 效用网格下的工作流时间约束-费用优化调度是一个 NP 难问题，基于时间耦合强度(time-dependent coupling strength, TCS)的最适规则(best fit time-dependent coupling strength, BFTCS)将作业的资源特征与工作流的结构特征作为优先级规则的两个重要方面应用于迭代算法的改进阶段，取得了良好效果，然而，BFTCS 忽略了工作流的时序特征. 在已有工作的基础上，定义任务的时间灵活度(temporal mobility, TM)并设计基于时间耦合强度和时间灵活度的最适规则(best fit based on time-dependent coupling strength and temporal mobility, BFTCSTM). 该规则在 BFTCS 规则的基础上选择 TM 较大的任务优先迭代，有效减缓了迭代过程中工作流长度的增长过程，使其他任务能进一步优化费用的机会增大，改善了工作流的费用优化效果. 实验结果证明了 BFTCSTM 的优越性 摘要小五号

关键词 工作流调度；效用网格；费用优化；时间耦合强度；时间灵活度 关键词小五号

中图法分类号 TP393 中图法小五号

收稿日期：2010-07-16；**修回日期**：2011-04-02

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)

正文五号宋体

网格技术通过标准开放服务体系结构将分布在不同地域不同组织的大量异构资源组织成一个大型虚拟系统,随着越来越多资源提供者和资源消费者的加入,对资源进行付费使用已经成为一种必然趋势[1].在这类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,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比较常见也比较难解的问题.针对这一问题,基于时间分配的启发式是一类非常简单且有效的解决方法[2-5],但这类方法在时间分配过程中忽略了任务的资源特征,因此算法性能受到一定影响;基于遗传算法、粒子群、蚁群算法等在大量解空间中进行有导向的搜索也是比较常见的方法[6-8],但这类方法通常具有较大的时间开销,且费用优化效果与优化算子的选取相关;此外,基于优先级规则的迭代算法是另一类比较常见且有效的方法[9-12],这类方法通过设置一定优先级规则在工作流中选择任务,在满足时间约束的条件下将该任务的当前服务逐步替换成执行时间较长但费用较低的服务,从而优化工作流的执行费用.

本文主要针对基于优先级规则的迭代算法进行研究,该类算法的关键是优先级信息的选取和优先级规则的设置.在工作流的时间约束-费用优化问题中,最适规则(best fit, BF)[11]与基于任务时间耦合强度的最适规则(best fit time-dependent coupling strength, BFTCS)[12]是2种有效的优先级规则,其中BF考虑了任务的资源特征,选择收益比较大的任务优先迭代;而BFTCS规则在BF的基础上考虑工作流的结构特征,将工作流中任务间的依赖关系进行量化并将其作为优先级规则的另一信息引入算法的改进阶段,均取得了较好的调度效果.然而这2种规则均忽略了工作流的时序特征,由于工作流长度由开始任务到结束任务的最长路径决定(路径长度为该路径上所有任务的执行时间之和),因此对不同路径上的任务进行迭代时工作流长度的增长速度各不相同,从而使得其他任务能进一步优化费用的机会也各异.

基于这一重要信息,本文研究工作流的时序特征,将满足时间约束的任务最迟完成时间和满足工作流偏序关系的任务最早完成时间之差定义为任务的时间灵活度(temporal mobility, TM),并将TM作为优先级规则的另一重要信息引入迭代算法的改进阶段,在已有优先级规则BFTCS的基础上设计基于时间耦合强度与时间灵活度的最适规则(best fit based on time-dependent coupling strength and temporal mobility, BFTCSTM).BFTCSTM规则选择TM较大(路径长度较小)的任务优先迭代,由于TM反映了各任务所在路径的长度差异及最大可用的费用优化空间,因此BFTCSTM有效减缓了工作流长度的增长过程,同时也减缓了其他任务费用优化空间的减小过程,在获得费用较大收益的同时也使得其他任务能进一步优化费用的可能性增大,最终达到改善费用优化效果的目标.实验结果表明,基于BFTCSTM规则的迭代算法的平均费用比BFTCS规则改进了3%,比BF规则改进了8%.

1 时间约束-费用优化问题描述

一级标题小四黑

效用网格下工作流的时间约束-费用优化问题已被公认为是一个多项式时间内无法求解的NP难问题[13],该问题需在效用网格中为工作流中的各任务选择一个服务,在满足偏序关系和时间约束的条件下达到工作流执行费用最小化的目标.

工作流W通常通过有向无环图DAG(directed acyclic graph)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,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$pre(i)$,当$pre(i)$全部执行完成时i便达到就绪条件;i的多个后继为后继列表,记为$succ(i)$.对于W中的两任务i和k,如果$(i,k)\notin E$,但存在任务j满足依赖关系:$(i,j)\in E\wedge(j,k)\in E$,则称$i$与$k$存在间接依赖关系,间接依赖关系可传递.此外,设定$W$中只有一个开始任务(不存在任何前驱的任务,记为1)和一个结束任务(不存在任何后继的任务,记为n).

对于W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$\tau_{ik}$与$c_{ik}$分别表示$s_{ik}$的执行时间与执行费用,$S(i)$中的服务均按时间的从小到大进行排列,同时设定执行时间较长的服务执行费用较低,在满足工作流时间约束的条件下,当任务i选择当前服务s_{ik}的右相邻服务$s_{i(k+1)}$执行时,工作流的执行费用得到优化.$\xi=\{\zeta\mid\zeta(i)=s_{ik},\forall i\in V,s_{ik}\in S(i)\}$为工作流的一个调度解,表示为$V$中的每个任务$i$在对应的服务列表

$S(i)$中选择一个服务.假定各服务在执行过程中都能提供与承诺一致的服务质量,当所有任务的服务选定之后,工作流的执行时间与执行费用便确定,其执行时间为从开始任务到结束任务的最长路径的长度(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 W 中所有任务的费用之和.基于以上模型,本文的调度目标可形式化描述为

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

s.t.

$$\beta_n \leqslant \delta, \tag{2}$$

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

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

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

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,δ 为给定的工作流约束时间.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;式(3)描述了工作流的偏序关系;式(4)和式(5)表示每个任务只能选择且必须选择一个服务执行.

2 时间灵活度

在某一调度 ξ 下,如果任务 i 推迟一段时间执行完成后仍能满足工作流的时间约束目标,则称 i 具有松弛时间.在任务的可用松弛时间内,迭代算法为该任务选择当前服务的右相邻服务(执行时间较长但执行费用较低的服务)执行,达到工作流费用优化的目标.首先通过松弛时间来定义任务的时间灵活度:

定义 1. 时间灵活度 TM.将某任务的最大松弛时间定义为该任务的时间灵活度.

显然,时间灵活度反映了任务可用的费用优化空间,时间灵活度越大,任务能进一步优化费用的可能性也越大.通过任务的最早开始/完成时间(earliest start/finish time, EST/EFT)和最迟开始/完成时间(latest start/finish time, LST/LFT)来衡量调度 ξ 下任务的时间灵活度.

定义 2. 最早开始/完成时间 EST/EFT.任务达到就绪条件的最早时间定义为该任务的最早开始时间;而与最早开始时间对应的任务完成时间则定义为该任务的最早完成时间.

定义 3. 最迟开始/完成时间 LST/LFT.将能使得某任务的所有后继(包括间接后继)能在约束时间内执行完成的最迟时间点定义为该任务的最迟完成时间;而与最迟完成时间对应的任务开始时间定义为最迟开始时间.

通过正向和逆向宽度优先方法对 EST/EFT, LST/LFT 进行求解:

$$\begin{cases} EST(i)=0, & i=1,\\ EST(i)=\max\limits_{j\in pre(i)}EFT(j), & \text{else};\\ EFT(i)=EST(i)+\tau(i). \end{cases} \tag{6}$$

$$\begin{cases} LFT(i)=\delta, & i=n,\\ LFT(i)=\min\limits_{j\in succ(i)}LST(j), & \text{else};\\ LST(i)=LFT(i)-\tau(i). \end{cases} \tag{7}$$

式(6)(7)中 $\tau(i)$ 为调度 ξ 下 i 的执行时间.通过任务 i 的最迟完成时间与最早完成时间之差来衡量 i 的时间灵活度:

$$TM(i)=LFT(i)-EFT(i). \tag{8}$$

由式(6)~(8)可知,$TM(i)$不仅反映了工作流长度与约束时间之间的差异,同时也反映了 i 所在路径与其他路径的长度差异.对于具有相同开始任务(记为 i)和结束任务(记为 j)的多条路径,由于路径长度由 i 与 j 之间的最长路径决定,因此,短路径中任务(不含 i 与 j)的时间灵活度显然小于长路径中任务的时间灵活度.当选择时间灵活度较大的任务(即短路径中的任务)进行迭代时,i 与 j 间的路径长度将保持不变或以较慢速度增长,其他任务仍有机会进行费用优化的可能性比较大;而选择时间灵活度较小的任务(即长路径中的任务)优先迭代时,路径长度将以较快速度增长,其他任务进行费用优化的机会将减少;由此可见,时间灵活度是影响迭代算法费用优化性能的一个重要因素.

3 时间耦合强度

除时间灵活度之外,工作流的结构特征(即各任务间的依赖关系)也是影响迭代算法性能的关键因素.文献[12]对这一特征进行了分析,为任务定义时间耦合强度,并将其作为优先级的一个重要信息引入迭代算法的改进阶段.在已有工作的基础上,我们通过任务间的依赖关系来定义时间耦合强度(time-dependent coupling strength, TCS),并通过宽度优先方法对 TCS 进行求解.

定义 4. 时间耦合任务.工作流中具有依赖关系

(包括直接和间接依赖)的任意两任务互称时间耦合任务.

定义 5. 时间耦合强度 TCS. 定义工作流中任务 i 的时间耦合任务总数为的时间耦合强度,记为 $TCS(i)$.

$TCS(i)$越大,与 i 存在依赖关系的任务个数便越多. 由于 i 执行时间的增长将使其耦合任务的费用优化空间减少,因此选择 TCS 较大的任务进行迭代时将减少较多任务的费用优化空间. 由此可见,TCS 也是优先级规则的一个重要方面. 文献[12]通过可达性矩阵对任务的时间耦合强度进行求解,当工作流规模较大时,该方法将耗费大量的资源空间. 我们通过宽度优先方法来计算 $TCS(i)$. 首先定义 i 的前向依赖列表 $FL(i)$和后向依赖列表 $BL(i)$:

定义 6. 前向依赖列表. 对于 $\forall j \in V$,如果 W 中存在从 j 到 i 的可达路径,则称 j 为 i 的前向依赖任务,i 的所有前向依赖任务组成的列表为 i 的前向依赖列表,记为 $FL(i)$.

定义 7. 后向依赖列表. 对于 $\forall j \in V$,如果 W 中存在从 i 到 j 的可达路径,则称 j 为 i 的后向依赖任务,i 的所有后向依赖任务组成的列表为 i 的后向依赖列表,记为 $BL(i)$.

$FL(i)$与 $BL(i)$计算如下:

$$FL(i)=\begin{cases}\text{null}, & i=1;\\ pre(i)+\sum\limits_{j\in pre(i)} FL(j), & \text{else.}\end{cases} \tag{9}$$

$$BL(i)=\begin{cases}\text{null}, & i=n;\\ succ(i)+\sum\limits_{j\in succ(i)} BL(j), & \text{else.}\end{cases} \tag{10}$$

式(9)中的求和操作 $\sum\limits_{j\in pre(i)} FL(j)$为任务列表的非重复求和,当 $FL(i)$中已包含 $FL(j)$中的任务 k 时,k 不重复加入到 $FL(i)$中,即保证 $FL(i)$中任务的唯一性. 同理,式(10)中的求和操作 $\sum\limits_{j\in succ(i)} BL(j)$也为任务列表非重复求和. $TCS(i)$为 $FL(i)$和 $BL(i)$中的任务总数:

$$TCS(i)=|FL(i)|+|BL(i)|, \tag{11}$$

其中$|FL(i)|$与$|BL(i)|$分别为 $FL(i)$和 $BL(i)$中的任务个数.

4 迭代算法描述与复杂性分析

以上对迭代算法中优先级规则的 2 个重要信息进行了分析. 最适规则 BF[11] 考虑了调度空间中任务的资源特征,选择收益比较大的任务优先迭代. 本节将以上 2 个优先级信息与 BF 规则相结合,设计基于时间耦合强度和时间灵活度的最适规则 BFTCSTM,并给出基于优先级规则的迭代算法 IHPR(iterative heuristic based on priority rule)的费用优化过程.

4.1 初始解构造

由于调度空间中各任务的服务列表均按时间的升序(费用的降序)排列,为了保证工作流的时间约束目标,在初始解中为各任务选择候选服务列表中的第一个服务进行调度,即:

$$\zeta(i)=s_{i0}, \forall i \in V. \tag{12}$$

由于工作流中各任务均选择执行时间最小的服务,因此该调度下得到的工作流执行时间为最小执行时间(记为 d),当用户给定的截止时间 $\delta<d$ 时,不存在满足时间约束的调度,因此将 d 设为 δ 的下界.

4.2 优先级规则设置及任务选取过程

初始解给出了工作流执行时间最小但执行费用最大的工作流调度,当 $\delta \geqslant d$ 时,在满足时间约束的条件下,将任务的当前服务替换成右相邻服务便能对费用进行优化. 在此过程中,任务的选取是算法性能的关键,我们将任务的时间灵活度和时间耦合强度作为优先级规则的 2 个重要因素,将其与 BF 规则相结合,设计基于时间耦合度和时间灵活度的最适规则 BFTCSTM,并通过式(13)进行衡量:

$$\Delta(i)=\frac{(c_{ik}-c_{i(k+1)})\times TM(i)}{(\tau_{i(k+1)}-\tau_{ik})\times TCS(i)}, \tag{13}$$

其中 k 为当前调度 ξ 下任务 i 所选服务的服务序号,$k+1$ 则为右相邻服务的服务序号. Δ 反映了调度空间中任务的资源特征、工作流的结构特征及当前调度下的时序特征. 在当前调度 ξ 下,基于以上优先级规则选取任务的过程 $selectTask(W,\xi)$ 描述如下:

算法 1. $selectTask(W,\xi)$.

① 设定 $t=$null;
② for each $i \in V$
③ if(s_{ik}存在右相邻服务 $s_{i(k+1)}$且 $\tau_{i(k+1)} \leqslant LFT(i)-EST(i)$)
④ 按式(13)计算 $\Delta(i)$;
⑤ else
⑥ $\Delta(i)=0$;
⑦ endif
⑧ endfor
⑨ $t=\{i \mid \Delta(i)=\max\limits_{j\in V}\Delta(j) \wedge \Delta(i)\neq 0\}$;

⑩ 返回 t.

$selectTask(W,\xi)$对工作流中的所有任务进行遍历:当 i 存在右相邻服务 $s_{i(k+1)}$ 且满足条件 $\tau_{i(k+1)}\leqslant LFT(i)-EST(i)$时,计算 i 的优先级$\Delta(i)$,其中 $\tau_{i(k+1)}\leqslant LFT(i)-EST(i)$保证了工作流的偏序关系,即 s_{ik} 替换成 $s_{i(k+1)}$ 后,仍能保证 i 的后继任务均能在 δ 内执行完成.否则,如果 $s_{i(k+1)}$ 不存在或 $\tau_{i(k+1)}>LFT(i)-EST(i)$,$i$ 便无法进一步优化费用,此时 $\Delta(i)=0$.当工作流中存在多个任务都能进行费用优化时,$selectTask(W,\xi)$返回 Δ 值最大的任务 t;否则,当所有任务都无法进一步优化费用时,$selectTask(W,\xi)$便返回 null.

4.3 迭代算法描述及复杂性分析

在初始解基础上,迭代算法 IHPR 通过 $selectTask(W,\xi)$在 W 中选择一个任务,将该任务的当前服务替换成右相邻服务以完成一次迭代过程,IHPR 循环此过程直到所有任务均无法进行费用优化为止.IHPR 的过程描述如下:

算法 2. IHPR.

输入:工作流 W 及各任务的候选服务列表;

工作流的约束时间 δ;

输出:工作流的执行费用.

① 按式(9)~(11)计算 $TCS(i)$,$\forall i\in V$;

② 按式(12)构造初始解;

③ 按式(1)~(3)计算当前调度下的 $TM(i)$,$\forall i\in V$;

④ $t=selectTask(W,\xi)$;

⑤ 如果 t 为 null 则转步骤⑦;

⑥ 将 t 的当前服务 s_{ik} 替换成 $s_{i(k+1)}$,转步骤③;

⑦ 计算并返回工作流的执行费用.

IHPR 每次选取一个任务 t,将 t 的当前服务替换成右相邻服务.在此过程中各任务的可选服务范围单调减少.如果约束时间为∞,IHPR 的最大迭代次数为 $n\times m$(其中 n 为工作流中的任务个数,而 m 为服务列表的最大长度);当存在时间约束时,IHPR 的迭代次数一定小于 $n\times m$,由此可见,IHPR 是收敛的.在 IHPR 的每次迭代过程中,语句③计算任务的时间灵活度,该过程需对所有任务的前驱和后继进行遍历,最坏情况下时间复杂度为 $O(n^2)$;语句④对所有任务进行遍历并得到 Δ 值最大的任务,时间复杂度为 $O(n)$;语句⑤~⑥为常数时间;因此完成一次迭代的时间复杂度为 $O(n^2)$.由以上分析可知,算法的最大迭代次数为 $n\times m$,因此 IHPR 的算法复杂度为$O(n^3\times m)$.

5 实验结果与分析

为了衡量基于 BFTCSTM 规则的迭代算法性能,本文在模拟实验环境中实现了基于 BF 和 BFTCS 规则的迭代算法,同时实现了一种著名的基于时间分配的启发式算法 DBL(deadline bottom level)[5],DBL 基于任务的逆向深度对任务进行分层,将具有同步完成特征的任务分配到同一层进行费用优化,有效利用了工作流的费用优化空间,达到了较好的费用优化效果.所有算法均采用 Java 编程,运行于 2.53 GHz 的双核处理器上,内存为 2 GB.

5.1 实验设计

实验针对多个工作流实例在不同调度空间中的优化性能进行比较.工作流实例由工作流随机生成器生成,各任务的出入度为[1~10]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,生成过程中对服务列表的最大规模(service size, SS)进行控制,设定 $SS=\{10,20,30,40\}$,当 $SS=10$ 时,各任务的服务列表长度取[1~10]之间的随机数.任务 i 的执行时间取[5,60]之间的随机数;执行费用为执行时间的函数,分别对 3 类费用函数(cost function, CF)进行实验,包括凸函数 concave(ccv)、凹函数 convex(cvx)及线性函数 linearity(lin)函数,费用生成过程中保证执行时间较长的服务执行费用较低.此外,实验中对不同约束时间下的算法性能进行比较,由 4.1 节可知,当所有任务均选执行时间最小的服务执行时,δ 存在下界 d;而当所有任务均选执行时间最大的服务执行时,该调度下得到的工作流执行时间为 δ 的上界,记为 $\mathcal{D}$;易知当 $\delta\geqslant\mathcal{D}$ 时,容易得到费用最低的调度(为各任务选择执行费用最小的服务即可);实验过程中通过 δ 的上下界来设置约束时间:$\delta=d+\omega(\mathcal{D}-d)$,设定 $\omega=\{0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9\}$.实验总共进行的实验组数为 $5\times4\times3\times9=540$.

5.2 结果比较与分析

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)、平均执行费用(average cost, AC)及 BFTCSTM 相对于各算法的平均费用改进比(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:

$$BP = \frac{N_B(H)}{N}, \tag{14}$$

$$ADB = \frac{\sum_{i=1}^{N} \frac{C_i(H) - C_i(B)}{C_i(B)}}{N}, \tag{15}$$

$$AC = \frac{\sum_{i=1}^{N} C_i(H)}{N}, \tag{16}$$

$$IPAC = \frac{AC(H) - AC(BFTCSTM)}{AC(H)}, \tag{17}$$

$$ART = \frac{\sum_{i=1}^{N} RT_i(H)}{N}, \tag{18}$$

其中 H 代表算法,$N_B(H)$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.

表 1 为所有问题参数下各算法的比较情况.

Table 1 Comparison of Four Heuristics at All Problem Parameters

表 1 所有问题参数下各算法的性能比较

H	BP/%	$10^2 \times$ADB	AC	IRAC	ART/s
BFTCSTM	**96**	**0.04**	**10 155**	**0**	**11.4**
BFTCS	7	2.83	10 462	0.03	11.3
BF	0	10.29	11 018	0.08	8.7
DBL	1	23.99	12 827	0.21	0.007

由表 1 可见,基于 BFTCSTM 规则的迭代算法在费用优化性能上明显优于其他 3 种算法,其最优解比例为 96%,而平均偏差则仅为 0.000 4,这表明 BFTCSTM 规则不仅具有绝对占优的最优解比例,同时算法稳定性也比较好;其次为基于 BFTCS 规则和基于 BF 规则的迭代算法,而 DBL 算法的费用优化效果最差.从平均执行费用来看,BFTCSTM 规则的执行费用最低,比 BFTCS 规则和 BF 规则分别节省了 3%和 8%,比 DBL 则节省了 21%;这表明 BFTCSTM 规则由于考虑了任务的时序特征,选择时间灵活度较大的任务优先迭代,与另外 2 种规则相比,有效减缓了工作流长度的增长过程,使得其他任务能进行费用优化的机会增大,达到了工作流费用优化的目标;而 DBL 由于忽略了任务的资源特征,与其他 3 种迭代算法相比,费用优化效果比较差.

从算法开销来看,3 种迭代算法的平均时间开销为 10 s 左右,其中 BF 规则的时间开销稍小(为 8.7 s),这是由于 BF 规则仅考虑了任务的资源特征,在迭代过程中工作流长度以较快速度增长,因此算法收敛比较快.而基于时间分配的启发式算法 DBL 的时间开销远小于这 3 种迭代算法,仅为 7 ms,这表明 DBL 算法虽然性能相对比较差,但具有非常高的算法效率.

5.3 不同问题参数下的算法性能比较

由表 1 可知,基于 BFTCSTM 规则的迭代算法性能总体上优于其他 3 种算法.为了对算法进行全面评估,以下分别在不同问题参数(包括工作流规模、服务规模、约束时间及费用函数)下比较算法性能.

图 1 与图 2 分别为不同工作流规模和不同服务规模下各算法平均执行费用的比较情况.

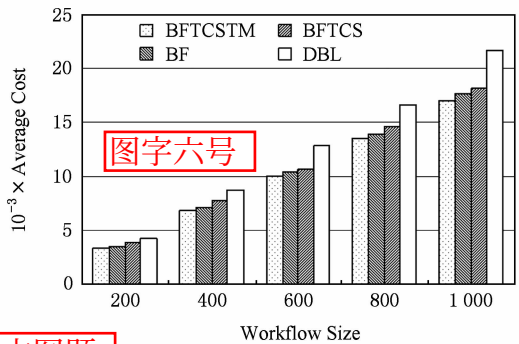

Fig. 1 Comparison of average cost at different workflow sizes.

图 1 不同工作流规模下的平均费用比较

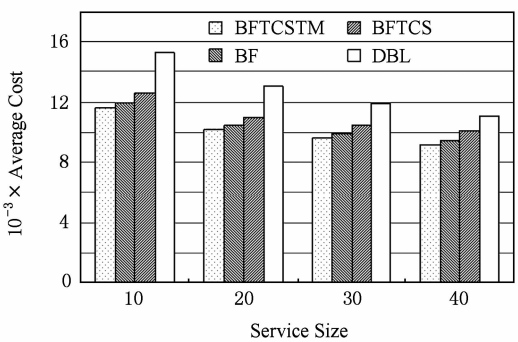

Fig. 2 Comparison of average costs at different service sizes.

图 2 不同服务规模下的平均费用比较

如图所示,在不同的工作流规模和服务规模下,基于 BFTCSTM 规则的算法性能在 4 种算法中总保持最优,而 BFTCS 规则和 BF 规则逐渐次之,DBL 的执行费用相对于这 3 种迭代算法来说则比较高;这表明 BFTCSTM 规则在工作流规模和服务规模

上具有较好的扩展性.随着工作流规模的增长,各算法的平均执行费用都有所增长,这是由于工作流的执行费用由所有任务的执行费用之和决定,当任务数增长时,工作流的执行费用也随之增长.同时,服务规模的增长也使得各算法的平均执行费用有所减少,这是由于服务规模的增长使得各任务可选服务的数量增加,减小了各候选服务间执行时间与执行费用的差距;在3种迭代算法中,服务规模的增长使得各任务能进一步优化费用的可能性增大;而在DBL中,服务规模的增大也使得工作流执行过程中的"时间碎片"减少,因此工作流费用得到减少.

图3为各算法的平均执行费用随约束时间的变化情况.在不同约束时间下,基于BFTCSTM规则的迭代算法的平均执行费用总保持最低,BFTCS与BF规则逐渐次之;而DBL的执行费用总保持最高;尤其当δ较小($\omega<0.3$)时,DBL的执行费用远高于其他3种算法,这是由于约束时间较小时,DBL由于缺乏有效的约束时间确定策略来保证时间约束,采用费用优化效果不理想的最小关键路径法(minimum critical path, MCP)[5]进行调度,因此费用优化效果比较差;但随着δ的增长,DBL的算法适用性逐渐满足,其平均执行费用逐渐接近BF.

表2为不同费用函数下各算法的比较情况.由表可见,各种费用函数下,BFTCSTM规则的费用优化效果总保持最优,但不同费用函数下的性能改进效果有所不同.如在凹函数中,BFTCSTM规则相对于其他2种规则的改进效果不太明显,比BFTCS和BF规则仅改进了1%和2%;而在线性函数中,BFTCSTM的改进效果最明显,比这两种规则分别改进了4%与15%,这是由于线性函数中任务的执行费用为执行时间的线性函数,同一任务的多个服务具有相同的收益比,因此消弱了任务的资源特征给优先级规则带来的影响,从而使得基于BF规则的迭代算法性能有所降低.

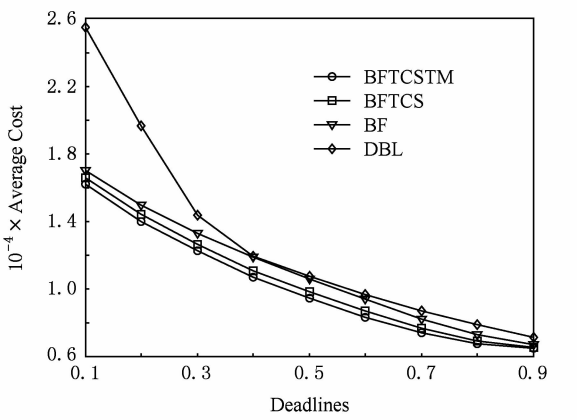

Fig. 3 Variation of average costs as a function of deadlines.

图3 平均执行费用随约束时间的变化图

Table 2 Performance of Each Heuristic Using Different Cost Functions 表题小五号

表2 不同费用函数各算法的性能比较

H	BP/%			$10^2\times$ADB			AC			IRAC		
	con	*ccv*	*lin*	*con*	*ccv*	*lin*	*con*	*ccv*	*lin*	*con* 表字六号		
BFTCSTM	93	97	97	0.1	0.01	0.000 1	12 081	7 972	10 412	0	0	0
BFTCS	7	3	3	3.77	0.97	3.76	12 455	8 052	10 878	0.03	0.01	0.04
BF	0	1	1	6.51	2.46	21.91	12 690	8 156	12 208	0.05	0.02	0.15
DBL	1	0	0	25.15	24.38	23.42	14 848	10 525	13 108	0.19	0.24	0.21

6 结 论

迭代算法在解决工作流的离散时间-费用平衡问题中取得了较好的效果,基于BFTCS规则的迭代算法考虑了任务的资源特征及工作流的结构特征,也取得了较好的效果.本文在BFTCS规则的基础上,考虑工作流的时序特征,为任务定义时间灵活度,并将时间灵活度作为优先级规则的另一个重要信息加入迭代算法的改进阶段,在BFTCS规则的基础上设计BFTCSTM规则.与BFTCS相比,BFTCSTM规则选择灵活度较大的任务优先迭代,有效减缓了工作流长度的增长过程,使得其他任务能进一步优化费用的可能性增大,达到了进一步减少工作流执行费用的目标.实验结果证实了基于BFTCSTM的迭代算法的优越性.

参 考 文 献

文献六号

[1] Buyya R, Abramson D, Venugeopal S. The grid economy [J]. Proceeding of the IEEE. 2005, 93(3): 698-714

[2] Yu J, Buyya R, Tham C K. Cost-based scheduling of scientific workflow applications on utility grids [C] //Proc of the 1st Int Conf on e-Science and Grid Computing (e-Science'05). Piscataway, NJ: IEEE, 2005:140-147

[3] 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[J]. 计算机研究与发展, 2008, 45(2): 246-253)

[4] Yuan Yingchun, Li Xiaoping, Wang Qian, et al. Deadline division-based heuristic for cost optimization in workflow scheduling [J]. Information Sciences, 2009, 179(15): 2562-2575

[5] 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)

[6] Garg S, Konugurthi P, Buyya R. A linear programming driven genetic heuristic 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

[7] Ji Yimu, Wang Ruchuan. Study on PSO heuristic in solving grid task scheduling [J]. Journal on Communications, 2007, 28(10): 60-66 (in Chinese)
(季一木, 王汝传. 基于粒子群的网格任务调度算法研究[J]. 通信学报, 2007, 28(10): 60-66)

[8] Neng C W, Jun Z. An Ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements [J]. IEEE Trans on Systems, Man and Cybernetics—Part C: Applications and Reviews, 2009, 39(1): 29-43

[9] Sakellariou R, Zhao H. Scheduling workflows with budget constraints [M]. Berlin: Springer, 2007: 189-202

[10] Yuan Yingchun, Li Xiaoping, Wang Qian, et al. Time optimization heuristics for scheduling budget-constrained grid workflows [J]. Journal of Computer Research and Development, 2009, 46(2): 194-201 (in Chinese)
(苑迎春, 李小平, 王茜, 等. 成本约束的网格工作流时间优化方法[J]. 计算机研究与发展, 2009, 46(2): 194-201)

[11] Akkan C, Drexl A, Kimms A. Network decomposition-based benchmark results for the discrete time - cost tradeoff problem [J]. European Journal of Operational Research, 2005, 163: 339-358

[12] 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)

[13] 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

作者介绍小五号

Liu Cancan, born in 1980. Received her PhD from the National University of Defense Technology. 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.

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.

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.