计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
46(2): 194-201, 2009

成本约束的网格工作流时间优化方法

苑迎春[1,2] 李小平[2,3] 王 茜[2,3] 王克俭[1]
[1](河北农业大学信息科学与技术学院 河北保定 071001)
[2](东南大学计算机科学与工程学院 南京 210096)
[3](计算机网络和信息集成教育部重点实验室(东南大学) 南京 210096)
(nd_hd_yyc@163.com)

Time Optimization Heuristics for Scheduling Budget-Constrained Grid Workflows

Yuan Yingchun[1,2], Li Xiaoping[2,3], Wang Qian[2,3], and Wang Kejian[1]
[1](*Faculty of Information Science and Technology, Agricultural University of Hebei, Baoding, Hebei* 071001)
[2](*School of Computer Science and Engineering, Southeast University, Nanjing* 210096)
[3](*Ministry of Education Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing* 210096)

Abstract Workflow scheduling which guarantees anticipated QoS (quality of service) is a fundamental and complex problem in grids. In this paper, the budget-constrained scheduling of workflows represented by DAG (directed acyclic graph) with the objective of time optimization is considered. In general, the optimization problem is NP-hard. Two new iterative heuristics based on priority rules are proposed. According to the property of parallel activities in DAG, two concepts called TL (top level) and BL (bottom level) are defined. By incorporating them with priority rule MP (maximum profit) respectively, two priority rules, i.e. MPTL (maximum profit with top level) and MPBL (maximum profit with bottom level) are designed. They are implemented in iterative heuristics to generate iteratively feasible solutions from leveling-based initial solutions which need bigger total costs. In each step, MPTL and MPBL manage to take into account the maximum decrease in the total cost but the minimum increase of workflow duration. Computational experiments show that MPTL and MPBL can considerably improve the average performance of MP within a few iterations and a little computation time. Moreover, MPBL outperforms MPTL. As well, the impact of budget constraints and the number of Web services on the two heuristics are analyzed.

Key words service grid; workflow; iterative heuristics; top level; bottom level

摘 要 针对成本约束有向无环图 DAG(directed acyclic graph)表示的网格工作流完工时间最小化问题,提出两个基于优先级规则的迭代启发算法.算法利用并行活动特征定义正向分层和逆向分层两个概念,将其分别引入最大收益规则 MP(maximum profit),得到正分层最大收益规则 MPTL(maximum profit with top level)和逆分层最大收益规则 MPBL(maximum profit with bottom level).两规则每次迭代尽量以完工时间的最小增加换取总费用的最大降低,逐步将分层初始解构造为满足成本约束的可行解.模拟结果表明,两规则在获得较少迭代次数和运行时间的同时,能显著改进 MP 规则的平均性能,且 MPBL 优于 MPTL.

收稿日期:2007-11-07;**修回日期**:2008-08-27
基金项目:国家自然科学基金项目(60672092,60504029,60873236);国家"八六三"高技术研究发展计划基金项目(2008AA04Z103);河北省科学技术研究与发展计划基金项目(072135126)

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

关键词 服务网格; 工作流; 迭代启发算法; 正向分层; 逆向分层

中图法分类号 TP393

工作流作为一种面向过程建模和管理的核心技术,能有效描述活动及活动间的复杂约束关系[1].科学研究、电子商务等领域的许多应用都可看做是一些工作流实例,有向无环图 DAG(directed acyclic graph)是其常用的一种描述方式[2-3].由于涉及大量的数据和复杂运算,这些工作流需借助网格环境才能有效运行.作为下一代分布计算模型,网格能有效解决分布、异构环境下的大规模资源共享[4].2001年,Foster 等人[5]将 Web 服务技术引入网格环境,服务网格成为网格架构的主流方向之一.随着 Web 服务数量的不断增长,服务质量 QoS(quality of service)成为描述服务非功能特性的重要参数(如执行时间、费用、可靠性等)[6],用户期望工作流执行能够满足预期的 QoS 需求.因此工作流调度就是要基于 QoS 需求,从众多候选服务中为各活动选择最适的服务执行,协作完成整个应用.

基于 QoS 的服务选择通常情况下是一个 NP-hard 问题[6],已有许多文献提出模拟退火、遗传、多目标演化算法等求解多 QoS 优化问题[6-8].对时间、费用两个受用户普遍关注的 QoS 优化问题,Buyya 等人[9]提出 3 种启发算法求解不同成本/截止期(budget/deadline)约束的时间费用优化问题,但算法针对独立任务集优化.对 DAG 描述的网格应用时间费用优化问题,Lin 等人[10]指出在特定条件满足下(网格服务足够多,能保证并行活动同时执行),该问题可形式化为项目管理领域中的离散时间费用权衡问题 DTCTP(discrete time/cost tradeoff problem).Akkan 等人[11]对 Deadline 约束的 DTCTP 问题,提出最大收益规则 MP(maximum profit)对列生成算法得到的初始解改进,但该规则仅考虑单个结点在费用上的最大收益率,未考虑 DAG 图的结构信息.对 Deadline 约束的网格工作流费用优化问题,文献[12-14]从性能和效率两方面出发,提出基于截止期分解的启发策略.对 Budget 约束的网格工作流完工时间最小化问题,目前的研究成果较少,Yu 等人[15]提出遗传算法求解该问题.

本文对 Budget 约束的网格工作流完工时间最小化问题展开研究,基于 MP 规则提出两个迭代启发算法.根据 DAG 图中活动的并行特征给出两种分组方法:正向分层 TL(top level)和逆向分层 BL(bottom level);将两分组信息作为优先级信息分别引入 MP 规则,得到正分层最大收益规则 MPTL(maximum profit with top level)和逆分层最大收益规则 MPBL(maximum profit with bottom level);两规则综合考虑 DAG 图中并行活动的特征信息和服务资源的 QoS 信息,试图在一次迭代中实现多个活动在费用上的最大收益率,改变 MP 规则仅在单个结点实现费用最大收益率的策略.通过将分层初始解逐步构造为满足成本约束的可行解,得到目标函数的近似最优解.

1 问题描述及相关定义

DAG 表示的工作流是网格环境下的一种重要应用形式.令 DAG 图记做 $G=(V,E)$,其中 $V=\{1,2,\cdots,n\}$ 表示所有活动(任务)集;活动 1 和活动 n 是添加的虚活动,分别表示唯一入口和出口活动(虚活动是执行时间和成本都为 0 的任务).E 是有向边集合,表示活动的偏序约束,即对 $\forall(i,j)\in E$,i 执行完 j 才能执行(i,j 是活动编号且拓扑有序,即 $i<j$).图 1 是一个 DAG 表示的工作流实例,其中结点 1 和结点 16 是添加的入口和出口任务(用虚线圆表示).

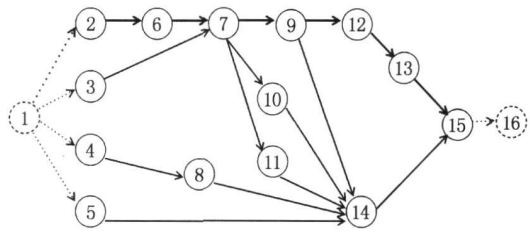

Fig. 1 A DAG-based workflow example.

图 1 一个 DAG 表示的工作流实例

网格服务通过标准接口描述其功能和非功能属性(QoS),并由网格信息服务器统一管理.信息服务器提供属性查询接口.这样,对 $\forall i\in V$,通过查询接口可搜索到完成该活动的一个候选服务集,称为 i 的服务池,记做 $SP(i)$,服务池长度记为 $l(i)$.本文考虑时间和费用两个参数,用三元组 (i_k,t_{ik},c_{ik}) $(1\leqslant k\leqslant l(i))$ 表示完成活动 i 的第 k 个服务(i_k 是服务编号,t_{ik},c_{ik} 表示服务 i_k 执行 i 所需时间和费用),则 $SP(i)=\{(i_k,t_{ik},c_{ik})\mid 1\leqslant k\leqslant l(i)\}$.对 $\forall i\in V$,令 $SP(i)$ 中所有候选服务按时间非递减排列.不失一般性,对任意 i_k,i_{k+1},规定满足 $t_{ik}<t_{i,k+1}$ 且 $c_{i,k+1}<c_{ik}$.

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

成本约束是工作流所有任务完成所需费用的上限,记做 B.完工时间是任务在当前所选服务分配下确定的关键路径运行时间,也是出口任务的完成时间.工作流调度目标就是在满足成本约束下,为各活动从其服务池中选择最合适的服务,使得工作流的完工时间最小,形式化描述为

$$\min\ f_n \tag{1}$$

$$\text{s.t.}\ \sum_{k=1}^{l(i)} x_{ik} = 1,\ \forall\, \text{i} \in \text{V}, \tag{2}$$

$$f_i \leqslant \text{f}_j - \sum_{k=1}^{l(i)} t_{ik} x_{ik},\ \forall\, (\text{i}, \text{j}) \in \text{E}, \tag{3}$$

$$\sum_{i \in \text{V}} \sum_{k=1}^{l(i)} x_{ik} c_{ik} \leqslant \text{B}, \tag{4}$$

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

其中 x_{ik}是布尔变量,当任务 i 选择服务池中第 k 个服务执行时其值为 1,否则为 0; f_i 表示任务 i 的完成时间,f_n 是工作流完工时间; B 是给定成本约束.目标函数式(1)是最小化工作流完工时间; 约束式(2)表示每个任务只能选择一个服务; 式(3)满足偏序约束; 式(4)满足成本约束; 式(5)说明 x_{ik}是布尔变量.

下面给出 $G=(V,E)$的几个相关定义:

定义 1. 给定 $G=(V,E)$,对 $\forall \text{i} \in V$,其正向深度[12] $TD(i)$定义为入口结点到 i 的最大路径长度(路径长度仅指边的个数).

令 $pre(i)$表示结点 i 的直接前驱集合,则 i 的正向深度递归计算公式[12]如下:

$$TD(i) = \begin{cases} 0, & i = 1, \\ \max\limits_{j \in \text{pre(i)}} \{TD(j)\} + 1, & \text{otherwise}. \end{cases} \tag{6}$$

定义 2. 有相同正向深度的结点组成一个正向分组(层)TL,记做 $TL_k(0 \leqslant k \leqslant TD(n))$.

定义 3. 给定 $G=(V,E)$,对 $\forall \text{i} \in V$,其逆向深度[14] $BD(i)$定义为结点 i 到出口结点的最大路径长度(路径长度仅指边的个数).

令 $succ(i)$表示 i 的直接后继集合,则结点 i 的逆向深度递归计算公式[14]为

$$BD(i) = \begin{cases} 0, & i = n, \\ \max\limits_{j \in \text{succ(i)}} \{BD(j)\} + 1, & \text{otherwise}. \end{cases} \tag{7}$$

定义 4. 有相同逆向深度的结点组成一个逆向分组(层)BL,记做 $BL_k(0 \leqslant k \leqslant BD(1))$.

将所有 TL 按正向深度值升序排列可组成 TL 列表,并对任意 $TL_k(0 < k < TD(n))$,规定其前驱层是 TL_{k-1},后继层是 TL_{k+1}.同样,将所有 BL 按逆向深度值降序排列可组成 BL 列表,对任意 $BL_k(0 < k < BD(1))$,规定其前驱层是 BL_{k+1},后继层是 BL_{k-1}.表 1 是图 1 实例两种分组列表的计算结果:

Table 1 Two Level Lists of the Example in Fig.1

表 1 实例 1 中两种分层列表的计算结果

TD	Top Level	BD	Bottom Level
0	{1}	8	{1}
1	{2,3,4,5}	7	{2}
2	{6,8}	6	{3,6}
3	{7}	5	{7}
4	{9,10,11}	4	{4,9}
5	{12,14}	3	{5,8,10,11,12}
6	{13}	2	{13,14}
7	{15}	1	{15}
8	{16}	0	{16}

2 迭代启发算法

2.1 算法基本思想

本文迭代启发算法的基本思想是以一个具有较小完工时间的解(所需费用较高)作初始解,采用迭代规则逐步降低工作流所需费用,直到生成一个满足成本约束的可行解,此时工作流完工时间就是目标函数的近似最优解.迭代算法步骤如下.

算法 1. 基于优先级规则的迭代算法 IAPR (iterative algorithm based priority rules).

1) 构造初始解,并计算其费用 C_{total}和完工时间 f_n;

2) 若 $C_{\text{total}} \leqslant B$,则 f_n 即为最小完工时间,转步骤 4);

3) do while $C_{\text{total}} > B$

① 根据迭代规则选择结点 $v \in V$;

② 为 v 分配时间增量;

③ 调整各活动的时间窗口;

④ 在新时间窗口约束下,各活动选择所需费用最小的服务;

⑤ 计算 C_{total}和 f_n;

4) 输出最小时间解 f_n;

5) 结束.

下面以基于 MPBL 规则的迭代算法为例,介绍算法中各步骤的具体实现过程.

2.2 分层初始解

由文献[14]可知,BL 中同组任务并行执行,且部分活动具有同步完成特征.规定同组任务有相同

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

截止时间,而起始时间分别由其前驱任务确定.当每个活动选择最快服务时,各分组截止时间(δ_{BL_i})以及活动的开始时间(β_i)和截止时间(δ_i)公式[14]为

$$\begin{cases}\beta_j = 0,\ j = 1,\\ \delta_{BL_i} = \max\{\beta_j + t_{j1}\},\ j \in \mathrm{BL}_i,\\ \delta_j = \delta_{BL_i},\ j \in \mathrm{BL}_i,\\ \beta_j = \max\{\delta_i\},\ i \in \mathrm{pred(j)},\ \mathrm{j} = 2,3,\cdots,n.\end{cases} \tag{8}$$

这样,活动 $i \in V$ 可在上式确定的时间窗口[β_i, δ_i]内选择费用最小的服务,得到工作流的一个时间费用解,称为 BL 分层初始解.

按 BL 调整活动起止时间,实质是进一步增强了同组活动的并行和同步特征.这样当某个活动分配一个时间增量时,同组中其他活动在不增加完工时间的前提下,仍会进一步降低总费用.分组中包含的活动数越多费用降低的可能性就越大.因此分组长度是迭代规则设计时需考虑的一个重要信息.

2.3 迭代规则

文献[11]提出 MP 规则以两个相邻模式的费用减少量和时间增加量的比值作为优先级规则(模式按时间递增排列),公式为

$$\Delta_j = (c_{jk} - c_{j,k+1})/(t_{j,k+1} - t_{jk}),\ j \in \mathrm{V}. \tag{9}$$

该规则规定具有最大值的活动优先级最高,因此每次迭代就能以活动执行时间的最小增加量换取所需费用的最大降低.该规则可用于求解 Budget 约束的工作流完工时间最小化问题,其使用的必要条件是活动当前的所选服务必须存在右相邻服务,这里将满足上述条件的活动称为可更新活动.令 j 当前的所选服务为 j_k,所有可更新活动集记为 X,则

$$X = \{j \in \mathrm{V} \mid \exists\, \mathrm{j}_{k+1}\ \&\&\ k+1 \leqslant \mathrm{l(j)}\}. \tag{10}$$

可以看出,MP 规则仅利用了服务资源的 QoS 信息.在前面分析中,已知同一 BL 中的任务有并行特征;初始解构造时,又对同组任务设置相同截止时间,以进一步增强同组任务的同步特征.当分组中某个任务分配一个时间增量时,该组中所有任务的时间窗口都被相应扩大,在不增加工作流完工时间的前提下,同组中其他可更新活动在新的时间窗口内可重新选择费用较小的服务,以进一步降低工作流费用.分组中的可更新活动数越多,工作流所需费用降低得就会越大.因此,在计算活动的优先级时,将该信息作为权重引入式(9),得到一个改进的 MP 规则:逆分层最大收益规则 MPBL(maximum profit with bottom level).该规则仍选取有最大值的活动优先分配时间增量,表示为

$$P_j = w_j \times \Delta_j,\ j \in \mathrm{V}, \tag{11}$$

其中 $\omega_j = |BL_k \cap X|$, $j \in BL_k$ 表示活动 $j \in V$ 的可更新活动数,它是当前迭代中能进一步降低工作流费用的活动数.由于迭代过程中活动的服务选择在不断更新,集合 X 的长度在迭代过程中会逐渐减小,即 ω_j 是一个动态变化量,因此 MPBL 是一个动态规则.

2.4 时间窗口调整

运用 MPBL 规则,每次选择有最大值的活动,记做 j.为该活动分配时间增量 $\lambda_j = t_{j,k+1} - t_{jk}$,则活动 j 新的时间窗口[β'_j, δ'_j]为

$$\begin{cases}\beta'_j = \beta_j,\\ \delta'_j = \delta_j + \lambda_j.\end{cases} \tag{12}$$

根据初始解时间窗口确定规则,需调整同分组及后继分组中所有活动的时间窗口以保持偏序约束,前驱分组活动的时间窗口不变.

1) 同分组活动的时间窗口调整

已知同分组任务有相同截止时间,因此分组内其他任务的截止时间要增加相同的时间增量 λ_j,开始时间不变.令活动 j 所在分组为 BL_p,则

$$\begin{cases}\beta'_v = \beta_v,\\ \delta'_v = \delta_v + \lambda_j,\end{cases}\quad \forall\, \mathrm{v} \in \mathrm{BL}_p. \tag{13}$$

2) 后继分组活动的时间窗口调整

BL_p 所有后继分组 $allsuc(BL_p)$(包含非直接后继)内的活动,其截止时间都要相应增加 λ_j,而开始时间分别取前驱活动的截止时间最大值,即

$$\begin{cases}\delta'_v = \delta_v + \lambda_v,\ v \in \mathrm{allsuc(BL}_p\mathrm{)},\\ \beta'_v = \max\{\delta'_u\},\ u \in \mathrm{pre(v)}.\end{cases} \tag{14}$$

2.5 服务选择与更新

调整时间窗口后,活动在新限定的时间窗口[β'_i, δ'_i]内重新选择费用最小的服务,总费用是所有活动的所需费用之和.

$$C_{\mathrm{total}} = \sum_{i=2}^{n-1} \min_{1 \leqslant k \leqslant \mathrm{l(i)}} \{c_{ik} \mid \beta'_i \leqslant \mathrm{t}_{ik} \leqslant \delta'_i,\ (i_k, t_{ik}, c_{ik}) \in \mathrm{SP(i)}\}.$$

2.6 算法复杂度分析

基于 MPBL 规则迭代算法的时间复杂度取决于 算法的迭代次数.由于迭代过程中可更新活动集 X 的集合长度是非递增的,因此算法的最大迭代次数为初始可更新活动集 X 中所含活动的服务池长度之和.令 X_s 表示分层初始解的可更新活动集,则算法的迭代次数最大不超过 $|X_s| \times M$,其中 $M = \max\{l(i) \mid i \in V\}$ 表示活动的服务池最大长度,即算法是收敛的.

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

2.7 基于 MPTL 规则的迭代算法

基于 MPTL 规则的迭代算法和基于 MPBL 规则的迭代算法步骤是一致的，只是其初始解构造和迭代规则都基于 TL 分层信息.

1) 初始解构造：TL 中同组任务可并行执行，且部分活动有同步开始特征.规定同层活动有相同开始时间和截止时间，对应式(8)，各分组截止时间(δ_{TL_i})以及活动的开始时间(β_i)和截止时间(δ_i)公式为

$$\begin{cases}\beta_j = 0, j \in \mathrm{TL}_0,\\ \delta_{TL_i} = \max\{t_{j1}\}, j \in \mathrm{TL}_i,\\ \delta_j = \delta_{TL_i}, j \in \mathrm{TL}_i,\\ \beta_j = \delta_{TL_{i-1}}, j \in \mathrm{TL}_i (i \neq 0).\end{cases}$$

各活动在上式确定的时间窗口$[\beta_i, \delta_i]$内选择费用最小的服务，得到工作流的一个时间费用解，称为 TL 分层初始解.

2) 迭代规则：在 MP 规则基础上引入正向分层信息 $\omega_j = |TL_k \cap X|, j \in TL_k$，得到 MPTL 规则，同样是一个动态规则.

其他步骤同基于 MPBL 规则的迭代算法，不再详细说明.

3 实验结果

为评估两种新算法的性能和运行效率，在模拟环境下对不同工作流实例测试.所有算法采用 Java 语言编程，操作系统为 Windows XP，运行在 Pentium Ⅳ、主频为 2.93 GHz，RAM 为 512 MB 的 PC 机上.

在不同应用实例上对算法测试，测试实例选取实例规模、服务规模和成本约束 3 个参数.实例规模指工作流应用(DAG 表示)中包含的活动数，实验中设置$|V| \in \{30, 60, 90, 120\}$四种规模；服务规模是指活动的服务池长度，实验中设置 $M \in \{10, 20, 30, 40, 50\}$五种服务规模，各活动的时间费用参数在一定区间内随机生成.成本约束用 $B = B_{\min} + \theta(B_{\max} - B_{\min})$确定 9 个不同值，其中 $B_{\min}$是工作流执行所需最小费用(每个任务选择费用最小的服务)，它是成本约束有效值的下界；$B_{\max}$是工作流所需最大费用(用 MCP 算法[14]求得)；$\theta \in \{5\%, 10\%, 15\%, 20\%, 25\%, 30\%, 35\%, 40\%, 45\%\}$.每种实例随机选取 50 个不同实例.综合各参数共生成 $4 \times 5 \times 9 \times 50 = 9000$ 个测试实例.

实验结果与基于 MP 规则的迭代算法比较，该算法以活动选择最快服务为初始解，每次迭代用 MP 规则逐步得到可行解.实验结果取同组实例的平均值.算法选取平均完工时间 ACT(average completion time)、平均迭代次数 AI(average iterations)和平均运行时间 ART(average run time，单位 ms)以及性能提高率作为比较参数.性能提高率定义为：若 A，B 两种算法对同一个(或组)得到的完工时间为 D_A 和 D_B，则 B 相对 A 的性能提高率为$I_{B,A} = \dfrac{D_A - D_B}{D_A} \times 100\%$.

3.1 不同实例规模下算法的运行结果比较

表 2 是 3 种算法在不同实例规模下的计算结果比较.从 3 种算法获得的性能数据看，MPTL 和 MPBL 在不同实例规模下得到的工作流完工时间均比 MP 小，相对 MP，MPTL 的性能提高率平均为 14.03%，MPBL 的性能提高率为 22.62%；而且随着实例规模增大，两种算法的性能提高率也逐渐提高.这是由于 1)MP 每次迭代仅对单活动实现了费用的最大收益率，而 MPTL 和 MPBL 利用并行活动共享时间特征，引入可更新活动数作为迭代规则的权重信息，每次迭代实现了多活动在费用上的最大收益率，故能获得较好的性能；2)随着实例规模增大，DAG 图结构变得比较复杂，迭代规则对分层信息的依赖性增加，因此两算法的性能提高率随实例规模增加而增大.

Table 2 Computational Results Depending on Instance Sizes

表 2 不同实例规模下算法的计算结果比较

Tasks	MP			MPTL				MPBL			
	ACT	AI	ART	ACT	AI	ART	$I_{MPTL,MP}$/%	ACT	AI	ART	$I_{MPBL,MP}$/%
30	1004	359	16	911	194	12	9.34	837	183	12	16.64
60	1321	665	68	1149	260	48	13.03	1037	239	49	21.44
90	1583	968	173	1403	324	131	11.32	1239	291	129	21.74
120	1650	1254	338	1316	312	268	20.29	1188	287	267	28.04
Aver	1389	811	149	1195	272	115	14.03	1075	250	114	22.62

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

从算法迭代次数和运行时间看,随实例规模增加,3 种算法的迭代次数和运行时间都相应增加,但 MPTL 和 MPBL 的迭代次数和运行时间在不同实例规模下相对 MP 都有大幅度降低.这是由于 1)从算法的收敛分析看,其迭代次数依赖初始可更新活动集 X_s 的集合长度 $|X_s|$,当实例规模越大时 $|X_s|$ 也越大,算法的迭代次数增加,运行时间也相应延长;2)由于 MPTL 和 MPBL 在每次迭代中能实现多个活动在费用上的最大收益率,收敛速度得到改进.因而迭代次数和运行时间有大幅度降低.

MPBL 和 MPTL 比较,MPBL 在不同实例规模下的性能和迭代次数优于 MPTL.这是由于 BL 分组及时间窗口设置方法相对 TL 分组比较合理(参看文献[14]),因此每次迭代中,基于 BL 的时间窗口调整机制产生的费用收益率高于 TL 分组的概率大,能以最快速度得到可行解.平均性能和迭代次数要优于 MPTL.

3.2 成本约束对算法的影响

成本约束 B 是用户期望工作流运行所需费用的上限.一般来说 B 越大,活动越可在较快的执行时间内完成,工作流的完工时间就越小,即成本约束值对算法的性能和运行效率会产生影响.表 3 是 3 种算法在不同成本约束下的计算结果比较.从表中所列数据可以看出,MPBL 和 MPTL 在不同成本约束下获得的目标函数值都优于 MP,且随成本约束值的提高,两种算法的性能提高率也明显增加.MPTL 的性能从 8.99%提高到 17.87%,而 MPBL 的性能从 11.64%提高到 34.00%.这是由于 MPTL 和 MPBL 分别以 TL 和 BL 分层解作为初始解(具有较快完工时间和较高费用),当成本约束较低时,分层初始解距离可行解较远,算法需较多迭代次数才能收敛到可行解,TL 及 BL 分层初始解的优势不能得到有效利用,而且随着迭代次数增加,优先级规则中的分层特征信息减弱,对 MP 信息的依赖性增强,故性能提高率较低.而当成本约束值较高时,分层初始解距离可行解较近,算法能以较少的迭代次数快速得到可行解,TL 及 BL 分层初始解优势以及优先级规则中的分层特征信息都能有效利用,故两算法的性能提高率都有很大程度提高.

Table 3 Impact of Budgets on the Iterative Heuristics

表 3 成本约束对算法性能和效率的影响

θ/%	MP			MPTL				MPBL			
	ACT	AI	ART	ACT	AI	ART	$I_{MPTL,MP}$/%	ACT	AI	ART	$I_{MPBL,MP}$/%
5	2057	1264	228	1872	412	141	8.99	1818	403	150	11.64
10	1846	1138	206	1606	361	135	12.99	1524	347	143	17.45
15	1646	1017	186	1417	323	130	13.91	1311	305	135	20.33
20	1482	900	165	1263	291	125	14.79	1140	269	126	23.06
25	1331	788	146	1130	262	118	15.08	995	237	117	25.20
30	1203	684	127	1013	235	109	15.80	872	209	105	27.57
35	1083	587	109	909	211	101	16.12	764	183	94	29.47
40	972	500	94	814	188	91	16.28	669	160	84	31.22
45	885	424	80	727	167	82	17.87	584	138	74	34.00
Aver	1389	811	149	1195	272	115	14.03	1075	250	114	22.62

从迭代次数和运行时间看,3 种算法随着成本约束值增加,其迭代次数和运行时间都会逐渐减少;且 MPBL 和 MPTL 在不同成本约束下的迭代次数和运行时间比 MP 有大幅度降低.

这是由于 1)本文提出的两个迭代算法都以较高费用解作为初始解,通过逐渐降低总费用构造满足成本约束的可行解,成本约束越高,算法所需的迭代次数越少,运行时间也相应减少;2)由于 MPTL 和 MPBL 在每次迭代中都能实现多活动在费用的最大收益率,因此两种算法都能大幅度降低 MP 的迭代次数和运行时间.

3.3 服务规模对算法的影响

服务规模指活动拥有的最大候选服务个数.表 4 显示了服务规模对各算法性能和效率的影响.

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

Table 4 Impact of the Number of Web Services on the Iterative Heuristics

表 4 服务规模对算法性能和效率的影响

M	MP			MPTL				MPBL			
	ACT	AI	ART	ACT	AI	ART	$I_{MPTL,MP}$/%	ACT	AI	ART	$I_{MPBL,MP}$/%
10	723	270	51	662	87	34	8.44	604	81	33	16.43
20	1018	512	95	903	185	76	11.35	802	166	76	21.23
30	1351	781	144	1174	275	118	13.13	1047	249	113	22.55
40	1730	1079	199	1471	363	152	14.96	1321	334	152	23.64
50	2124	1415	257	1763	451	194	17.04	1602	420	193	24.60
Aver	1389	811	149	1195	272	115	14.03	1075	250	114	22.62

从表 4 数据来看,MPTL 和 MPBL 在不同服务规模下获得的性能都优于 MP,且 MPBL 优于 MPTL.这是由于 1)服务规模变化会对服务的时间费用函数分布产生影响,当服务规模较小时,服务的时间费用分布较稀疏,单个活动的最大收益明显,MP 规则的搜索能力增强,即 MPTL 与 MPBL 规则对 MP 信息的依赖较大;且稀疏的服务分布会使可更新活动的服务选择范围变窄,MPTL 和 MPBL 的性能提高率较小.随着服务规模增加,服务的时间费用函数分布变得密集,MP 规则的搜索能力降低,MPBL(MPTL)对 MP 的依赖性降低,而对时间窗口的调整及服务更新机制的依赖性增加,因此两种算法能获得较好的性能.2)相对 MPTL,MPBL 具有 BL 分组优势,活动能获得较大的时间窗口;而密集的时间费用函数分布又使可更新活动能最大程度地降低费用,故 MPBL 在不同服务规模下都能得到最好的性能.

从算法的迭代次数和运行时间看,3 种算法的迭代次数和运行时间都随服务规模的增加而增加.在前面的算法复杂度分析中,已知迭代次数依赖于活动的最大服务池长度.因此服务规模越大算法的迭代次数越多,运行时间也会相应延长.

4 结 论

本文研究 Budget 约束的工作流完工时间最小化问题,从算法性能和运行效率出发,提出两个基于优先级规则的迭代启发算法.所设计的两个优先级规则 MPBL 和 MPTL 综合利用 DAG 图结构特征和服务的 QoS 特征,期望每次迭代在多个活动上实现费用最大收益率,改变 MP 规则中单个活动获取最大收益率的策略,尽量以完工时间的最小增加换取总费用的最大降低,用最快速度将分层初始解构造为可行解.模拟结果表明,两种算法在减小迭代次数、降低算法运行时间的同时,又能获得近似最优的工作流完工时间.而且由于 MPBL 具有 BL 分组优势,其性能和运行效率均优于 MPTL.

参 考 文 献

[1] Fan Yushun, Luo Haibin, *et al*. Workflow Management Technology Basis [M]. Beijing: Tsinghua University Press, 2001 (in Chinese)
(范玉顺, 罗海滨, 等. 工作流管理技术基础[M]. 北京: 清华大学出版社, 2001)

[2] Deelman E, Blythe J, *et al*. Mapping abstract complex workflows onto grid environments [J]. Journal of Grid Computing, 2003, 1(1): 25-39

[3] Buyya R, Yu J. Taxonomy of scientific workflow systems for grid computing [J]. SIGMOD RECORD, 2005, 34(3): 44-49

[4] Foster I, Kesselman C. The Grid: Blueprint for a Future Computing Infrastructure [M]. San Francisco: Morgan Kaufmann, 1999

[5] Foster I, Kesselman C, Nick J M, *et al*. Grid service for distributed system integration [J]. IEEE Computer, 2002, 35(6): 37-46

[6] Jin Hai, Chen Hanhua, Lü Zhipeng, *et al*. QoS optimizing model and solving for composite service in CGSP job manager [J]. Chinese Journal of Computers, 2005, 28(4): 578-588 (in Chinese)
(金海, 陈汉华, 吕志鹏, 等. CGSP 作业管理其合成服务的 QoS 优化模型及求解[J]. 计算机学报, 2005, 28(4): 578-588)

[7] Zhang C W, Su S, Chen J J. DiGA: Population diversity handling genetic algorithm for QoS-aware Web services selection [J]. Computer Communications, 2007, 30(3): 1082-1090

[8] Zhang Weizhe, Hu Mingzeng, Zhang Hongli, *et al*. A multiobjective evolutionary algorithm for grid job scheduling of multi-QoS constraints [J]. Journal of Computer Research and Development, 2006, 43(11): 1855-1862 (in Chinese)

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

(张伟哲, 胡铭曾, 张宏莉, 等. 多 QoS 约束网格作业调度问题的多目标演化算法[J]. 计算机研究与发展, 2006, 43(11): 1855-1862)

[9] Buyya R, Abramson D, Giddy J, *et al*. Economic models for resource management and scheduling in grid computing [J]. Journal on Concurrency and Computation: Practice and Experience, Special Issue on Grid Computing Environments, 2002, 14(13-15): 1507-1542

[10] Lin M, Lin Z X. A cost-effective critical path approach for service priority selections in grid computing economy [J]. Decision Support Systems, 2006, 42(3): 1628-1640

[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, 165(2): 339-358

[12] Yu J, Buyya R, Tham C K. Cost-based scheduling of workflow applications on utility grids [C] //Proc of the 1st IEEE Int Conf on e-Science and Grid Computing. Los Alamitos, CA: IEEE Computer Society, 2005: 130-137

[13] Yuan Yingchun, Li Xiaoping, Wang Qian. Cost optimization for scheduling grid workflows 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时间-费用优化算法[J]. 计算机研究与发展, 2008, 45(2): 246-253)

[14] Yuan Yingchun, Li Xiaoping, Wang Qian, *et al*. Bottom level based heuristic for scheduling workflows in grids [J]. Chinese Journal of Computers, 2008, 31(2): 282-290 (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[J]. 计算机学报, 2008, 31(2): 282-290)

[15] Yu J, Buyya R. Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms [J]. Scientific Programming Journal, 2006, 14(3-4): 217-230

Yuan Yingchun, born in 1970. Ph. D. and associate professor. Her main research interests include grid computing and service composition.

苑迎春, 1970 年生, 博士, 副教授, 主要研究方向为网格计算、服务组合.

Li Xiaoping, born in 1970. Ph. D., associate professor and Ph. D. supervisor. His main research interests include machine scheduling, project scheduling, and service computing.

李小平, 1970 年生, 博士, 副教授, 博士生导师, 主要研究方向为机器调度、项目调度和服务计算.

Wang Qian, born in 1946. Professor and Ph. D. supervisor. Her main research interests include information integration, database technology, and CSCW.

王 茜, 1946 年生, 教授, 博士生导师, 主要研究方向为信息集成技术、数据库技术以及计算机协同工作.

Wang Kejian, born in 1971. Master and associate professor. Her main research interests include artificial intelligence and pattern recognition.

王克俭, 1971 年生, 硕士, 副教授, 主要研究方向为人工智能与模式识别.

Research Background

Grid computing has emerged as a promising distributed computing paradigm for solving large-scale computational and data intensive problems in science, engineering and commerce. Its key problems are resource management and application scheduling. However, due to the highly heterogeneity in resources, management policies, users, and application requirements, different and effective performance parameters (time, cost, reliability and so on) are applied to reflect the actual characteristics. Thus, grid applications scheduling based on QoS (quality of service) requirements becomes a complex problem. Focusing on the scheduling problem of DAG-based grid workflow applications widely existing in different scientific domains such as bioinformatics and astronomy, two iterative heuristics based on priority rules are proposed to optimize the duration while meeting the given budget. Unlike rule MP, the proposed priority rules take into account the information of the QoS parameters of Web services and the property of parallel activities. Computational experiments show that MPTL and MPBL can considerably improve the average performance of MP within a few iterations and a little computation time. Moreover, MPBL outperforms MPTL. This work is supported by the National Natural Science Foundation of China under grants (No. 60672092, No. 60504029, No. 60873236), the National High Technology Research and Development Program of China (863 Program) (No. 2008AA04Z103), and the Research Program of Hebei Provincial Science Technology Department (No. 072135126).

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