

基于优先级规则的网格工作流调度

苑迎春^{1,2}, 李小平^{2,3}, 王茜^{2,3}, 张晓东^{2,3}

(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, 江苏南京 210096;
3. 东南大学计算机网络和信息集成教育部重点实验室, 江苏南京 210096)

摘要: 网格资源需求的不断增长使价格成为资源进行竞争的有效手段, 有向无环图 DAG(Directed Acyclic Graph) 表示的工作流时间费用优化问题是网格环境下一个重要问题. 通常情况下, DAG应用调度属于 NP-Hard 问题. 通过分析活动间的时序特征, 给出时间耦合强度 TCS (Time-dependent Coupling Strength) 的定义, 用于标识一个活动最大的时间耦合活动个数; 将其作为优先级规则的一个重要信息和 BF 规则 (Best Fit) 结合, 设计出时间耦合强度最适规则 BFTCS (Best Fit with Time-dependent Coupling Strength), 用于启发式算法的改进阶段, 逐步提高初始可行解的性能. 模拟实验结果表明, 相对现有的启发式算法, 基于 BFTCS 规则的启发算法能获得最好的性能和较快的运行效率; 最后讨论了问题参数对算法性能和效率的影响.

关键词: 网格计算; 工作流; 有向无环图; 优先级规则; 时间耦合强度

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2009) 07-1457-08

Grid Workflows Schedule Based on Priority Rules

YUAN Ying-chun^{1,2}, LI Xiao-ping^{2,3}, WANG Qian^{2,3}, ZHANG Xiao-dong^{2,3}

(1. Faculty of Information Science and Technology, Agriculture University of Hebei, Baoding, Hebei 071001, China;

2. School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu 210096, China;

3. Ministry of Education Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing, Jiangsu 210096, China)

Abstract: The increasing demand for grid computing resources calls for an incentive-compatible pricing mechanism for differentiated service qualities. The Time-Cost tradeoff problem for grid workflow applications described by Directed Acyclic Graph (DAG) becomes a significant problem. DAG-based optimization problem has been shown to be NP-hard in general cases. In this paper, a new concept called TCS (Time-dependent Coupling Strength) is introduced, which is identified for a given activity the maximum number of time-dependent coupling activities. By incorporating it into priority rule BF (Best Fit) which only takes into account the ratio of the cost improvement to the increase of duration of an activity, a novel priority rule BFTCS (Best Fit with Time-dependent Coupling Strength) is proposed, which is implemented in a heuristic to improve further the feasible initial solutions. Computational experiments indicate that rule BFTCS based heuristic can perform better than other existing heuristics but require a little more computation time. As well, the impact of problem parameters on the heuristics is discussed.

Key words: grid computing; workflow; directed acyclic graph (DAG); priority rules; time-dependent coupling strength

1 引言

科学研究、生产活动和其他大量业务流程实质上都是工作流应用, 每个流程通常由一组活动及活动间的偏序关系组成. 工作流活动执行前需分配资源, 即在活动和资源之间建立映射关系. 能完成同一活动的资源会很多, 且资源通常是分布式的, 如何管理并有效使用资源是一个重要问题. Foster 等^[1]提出网格实现资源共享, 网格下的工作流称为网格工作流^[2-4]. Buyya 等^[2]对网格

工作流分类, 有向无环图 DAG(Directed Acyclic Graph) 是工作流的一种常用描述方式. 李明禄等^[3]提出基于多 Agent 机制的网格工作流管理框架, 用人工智能规划技术实现活动和资源的映射. Deelman^[4]将网格工作流分为抽象和具体两个层次, 活动和资源的映射过程即是抽象工作流到具体工作流的转换过程. 活动和资源的映射实质是一个复杂的优化问题. 目前的网格资源通常以服务形式提供有偿服务^[5], 即活动使用网格资源必须支付费用. 因此, 网格工作流活动到服务资源的映射是一个

收稿日期: 2007-10-30; 修回日期: 2009-02-03

基金项目: 国家自然科学基金 (No. 60672092, No. 60504029, No. 60873236); 国家“863”高技术研究发展计划 (No. 2008AA04Z103); 河北省科学技术研究与发展计划 (No. 072135126); 河北省自然科学基金 (No. F2009000653)

时间-费用优化问题:在截止期约束下,最小化 workflow 费用。

工作流时间-费用优化问题在通常情况下是 NP-hard 问题^[6,7]。启发算法由于能在一个合理的运行时间内获得可接受解而被广泛采用。如 Buyya^[5]提出求解独立任务集时间费用优化的三种启发算法;文献[6]提出任务分组法求解 DAG 描述的时间费用优化问题;上述算法将网格看作一个异构、共享的计算资源池。对服务网格下的应用调度问题, Lin 等^[7]指出在特定条件下(服务资源足够多,能保证并行任务同时执行),网格 DAG 应用的时间费用优化问题可看作项目调度领域的 DTCTP (Discrete Time-Cost Tradeoff Problem) 问题^[8]。该问题在通常情况下已被证明是强 NP-hard 问题^[9],因此 Akkan^[10]基于网络图分解,利用列生成技术给出解的上下界计算方法,并提出最适规则 BF (Best Fit) 规则对该解改进。对 DAG 表示的网格 workflow 时间-费用优化问题,文献[11]提出基于正向分层 TL (Top Level) 的截止期分解算法 DTL (Deadline Top Level),但未给出截止期分解的具体过程。作者就同一问题提出基于逆向分层 BL (Bottom Level) 的截止期分解方法 DBL (Deadline Bottom Level)^[12],给出了两种截止期分解的具体实现过程,实验结果表明 DBL 相对 DTL 能获得较好的性能和运行效率。

分析现有的时间费用优化算法发现:(1)BF 最适规则可用于求解网格 workflow 的时间费用优化问题,但由于其仅关注活动所需的资源特征,不能获得较好的性能;(2)分层算法依据活动的并行和同步特征将截止期分解为活动的时间窗口来优化费用,未考虑活动的服务资源特征;且活动在限定时间窗口内选择服务执行,会产生许多不能利用的离散“时间碎片”。综合上述算法,本文提出一个基于优先级规则的启发式算法:通过定义时间耦合强度 TCS (Time-dependent Coupling Strength) 描述一个活动的最大时间耦合活动个数;将该信息引入 BF 规则,设计了时间耦合强度最适规则 BFICS (Best Fit with Time-dependent Coupling Strength),用于启发式算法的改进阶段,逐步提高初始可行解的性能。

2 问题描述

网格中许多大型应用通常由多个活动相互协作完成,其中有些活动由于存在数据或控制依赖关系串行执行,而另外一些活动不存在依赖关系而并行执行。活动之间的这种串并特征可模型化为一个 workflow,有向无环图 DAG 是一种常用描述方式。令 DAG 记作 $G = (V, E)$,其中 $V = \{1, 2, \dots, n\}$ 是结点集(结点、任务和活动在本文中含义相同),表示 workflow 包含的所有活动;

结点 1 和 n 是添加的虚活动(虚活动的执行时间和费用都为 0),分别表示唯一入口和出口任务。 E 是有向边集合,表示活动间的偏序约束,即对 $\forall (i, j) \in E, i$ 执行完后 j 才可执行(i, j 是活动的数字编号且拓扑有序,即 $i < j$)。图 1 是一包含 6 个任务的工作流实例,其中入口和出口任务 1 和 6 用虚线圆表示。

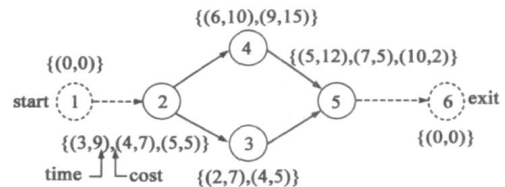


图 1 一个 DAG 表示的简单 workflow

服务网格下的计算、通信、存储、仪器、软件等所有资源都以服务形式存在,给用户提供了统一的资源使用模式。网格资源需求的不断增长使网格中出现了大量功能相同、服务质量 QoS (Quality of Service) 各异的服务,服务的执行时间和费用是 QoS 的两个重要属性。因此,对 G 中任一活动 i ,存在多个能够完成该活动的候选服务,它们有不同的响应时间和执行费用,令 (t_i^k, c_i^k) 表示能完成活动 i 的第 k 个服务,其中 t_i^k, c_i^k 分别表示服务 k 的执行时间和费用。活动 i 的候选服务集称为 i 的服务池, $l(i) = |SP(i)|$ 定义为服务池长度,即 $SP(i) = \{(t_i^k, c_i^k) | 1 \leq k \leq l(i)\}$ (虚活动的服务池为 $SP(i) = \{(0,0)\}$)。不同活动的服务池长度可能不同,如图 1 中活动 2 有 3 个候选服务 $\{(3,9), (4,7), (5,5)\}$,活动 3 有 2 个候选服务 $\{(2,7), (4,5)\}$ 。则对 $\forall i \in V$,令其服务池中所有候选服务按时间非递减排列,即服务编号越小,执行时间越快,服务级别也越高。因此 $k=1$ 表示最高级服务, $k=l(i)$ 表示最低级服务。

假定 workflow 内部处理时间忽略不计,且所有可选服务都能提供自己所承诺的服务质量,则 workflow 完工时间取决于活动所选服务的执行时间。即当各活动选择一个服务执行,就有一条关键路径(运行时间最长的路径)与之对应。workflow 完工时间就是关键路径运行时间,总费用就是所有活动的费用之和。

截止期 T (一般用常数表示) 是用户期望的 workflow 最晚完成时间,也是活动 n 的最晚完成时间。则 workflow 调度是在给定截止期约束下,从其服务池中为每个活动选择最合适的服务,使 workflow 所需费用最小,形式化描述如下:

$$\begin{aligned} \min & \sum_{i \in V} \sum_{k=1}^{l(i)} c_i^k x_i^k & (1) \\ \text{s. t.} & \\ & \sum_{k=1}^{l(i)} x_i^k = 1, i \in V & (2) \end{aligned}$$

$$f_i - f_j - \sum_{k=1}^{u(i)} t_j^k x_j^k, \forall (i, j) \in E \quad (3)$$

$$f_n \leq T \quad (4)$$

其中 x_i^k 是布尔变量, 当 i 选择第 k 个服务时值为 1, 否则为 0; $f_i (1 \leq i \leq n)$ 表示 i 的完成时间. 式(1)表示最小化 workflow 费用; 约束(2)是任务只能选择一个服务; 式(3)满足时序约束; 式(4)满足截止期约束.

3 时间耦合强度

在 DAG 图中, 位于同一路径的活动串行执行, 其完成时间是所有活动的执行时间之和, 且受截止期约束. 因此, 若其中一个活动选择级别较低的服务执行, 其他活动会由于截止期约束会选择服务级别较高的服务, 即这些活动的服务选择(实质上是对各活动的分配)彼此相互影响, 由此给出如下定义:

定义 1 给定 $G = (V, E)$, 对 $\forall i, j \in V$, 若 i 到 j 有路径可达, 则活动 i 与 j 互称为时间耦合活动(记作 TCA: Time-dependent Coupling Activity).

因此, 时间耦合活动必位于同一路径之上. 对 j 而言, 前驱活动 i 称为 j 的前向时间耦合活动; 对 i 而言, 后继活动 j 称为 i 的后向时间耦合活动.

定义 2 给定 $G = (V, E)$, 对 $\forall i \in V$, 其最大前向时间耦合活动数称为 i 的前向时间耦合强度(记作 FTCS: Forward Time-dependent Coupling Strength).

定义 3 给定 $G = (V, E)$, 对 $\forall i \in V$, 其最大后向时间耦合活动数称为 i 的后向时间耦合强度(记作 BTCS: Backward Time-dependent Coupling Strength).

令 $FTCS_i$ 和 $BTCS_i$ 分别表示 i 的前向和后向时间耦合强度, P_i^* 和 F_i^* 分别表示 i 的前驱集和后继集, 由定义 1, 定义 2 和定义 3, i 的 $FTCS_i$ 和 $BTCS_i$ 计算公式为

$$\begin{cases} FTCS_i = |P_i^*| \\ BTCS_i = |F_i^*| \end{cases} \quad (5)$$

定义 4 给定 $G = (V, E)$, 对 $\forall i \in V$, 其前向和后向时间耦合强度之和称为 i 的时间耦合强度, 记作 TCS.

令 i 的时间耦合活动集合记为 TCA_i , 其时间耦合强度记作 TCS_i , 则 i 的时间耦合强度公式为:

$$TCS_i = |TCA_i| = |P_i^*| + |F_i^*| \quad (6)$$

可以看出, 活动的时间耦合强度是一种数量度量, 用于标识一个活动的最大时间耦合活动数, 其值可通过 DAG 图的可达性矩阵计算. 如图 2(b) 为图 2(a) 的可达性矩阵. 矩阵中第 i 行求和是 i 的后向时间耦合强度; 第 i 列求和是 i 的前向时间耦合强度; 第 i 行和第 i 列求和则为 i 的时间耦合强度. 图 2(a) 中结点上方的数字表示该活动的时间耦合强度值(虚结点除外).

分析时间耦合强度的定义可知, 一个活动的时间耦合强度越大, 该活动在服务选择时影响的活动的数量越多. 如图 2(a) 中, 活动 7 具有最大时间耦合强度(值为 6), 若该活动优先选择服务级较低(运行时间较长)的服务, 则在截止期约束下, 其时间耦合活动 3, 4, 5, 6(虚结点除外)会因松弛时间减小或不存在而不能选择运行时间较长的服务, 这些活动所需费用就会较高, 算法的局部优化性能就会受到影响; 反之, 若活动 3(时间耦合强度小)优先选择服务级较低的服务, 它仅影响活动 7 的服务选择, 其它活动仍有机会选择服务级较低的服务来进一步降低 workflow 总费用, 即时间耦合强度是优先级规则设计时应考虑的一个重要因素.

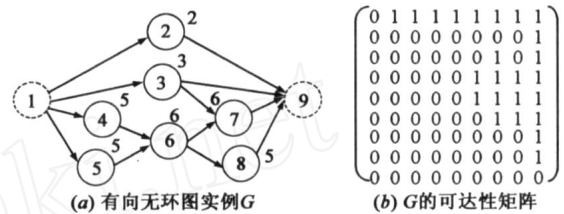


图 2 有向无环图及其可达性矩阵

4 基于优先级规则的迭代启发算法

DAG 表示的 workflow 时间费用优化问题在通常情况下已证明是 NP-hard 问题. 考虑到算法的性能和效率, 本文采用基于优先级规则的启发式求解策略. 算法包括可行解构建阶段和基于优先级规则的改进阶段, 下面介绍两个阶段的具体实现过程.

4.1 最早可行解

给定 workflow $G = (V, E)$, 按某种策略为 G 中每个活动从其服务池中选择一服务完成, 就得到 workflow 的一个解. 令 $X = \{k_1, k_2, k_3, \dots, k_n\}$ 表示 workflow 的一个解, 其中 k_i 表示活动 $i \in V$ 选择的服务级. 根据各活动选择的服务, 可计算解的完工时间和所需费用. 若该解的完工时间满足截止期约束, 则称其为可行解.

假定 workflow 各活动分别选择运行最快的服务执行, 即 $X = \{1, 1, 1, \dots, 1\}$, 则 workflow 有最小完工时间, 记作 T_{min} , 这个解 X 称为 G 的最早解. 对 workflow 的任意一个解, 其完工时间都应不小于最早解的完工时间, 也就是说, workflow 截止期的有效范围是 $T - T_{min}$.

因此, 给定 workflow $G = (V, E)$ 及相应的有效截止期 T , 其最早解必是一个可行解, 称为最早可行解.

4.2 迭代改进算法

若可行解中存在这样一些活动, 当他们选择服务级较低的服务更新当前服务后, 新得到的解不仅仍是一个可行解, 而且还可进一步降低 workflow 费用, 则这个可行解的性能还可进一步提高.

4.2.1 改进活动集

设 $X = \{k_1, k_2, k_3, \dots, k_n\}$ 表示 $G = (V, E)$ 的一个可行解, 即 $X = \{(t_j^k, c_j^k) | 1 \leq k_j \leq l(j), j \in V\}$. 采用前向关键路径法^[10]计算各活动的最早开始时间 EF_j 和最早完成时间 EF_j . 令 $ES_1 = EF_1 = 0$, 则

$$\begin{cases} ES_j = \max\{EF_i | i \in P_j^*\} \\ EF_j = ES_j + t_j^k, j = 2, 3, \dots, n \end{cases} \quad (7)$$

同样, 在截止期 T 下, 各活动的最晚完成时间 LF_j 和最晚开始时间 LS_j 可通过后向关键路径法^[10]计算. 初始化 $ES_n = EF_n = T$, 则

$$\begin{cases} LF_j = \min\{LS_h | h \in F_j^*\} \\ LS_j = LF_j - t_j^k, \text{for } j = n - 1, \dots, 1 \end{cases} \quad (8)$$

这样, 对 $\forall j \in V$, 区间 $[ES_j, LF_j]$ 定义了活动 j 在可行解 X 和截止期 T 下必须处理完成的最大时间窗口. 可以看出, 活动的执行时间窗口依赖当前可行解和给定的截止期.

对活动 j 的时间窗口 $[ES_j, LF_j]$, 若其服务池中存在一个级别较低的候选服务 k_j^* , 满足

$$\begin{cases} k_j < k_j^* \quad l(j) \\ t_j^k - LF_j - ES_j \end{cases} \quad (9)$$

则活动 j 选择 k_j^* 更新 k_j 时, 该活动可在其时间窗口完成, 因此新得到的解仍是可行解, 但其所需费用会降低. 也就是说, 满足上述条件的活动具有松弛时间, 该活动利用松弛时间可选择服务级较低的服务进一步降低 workflow 费用, 提高可行解的性能.

定义 5^[10] 给定 $G = (V, E)$ 和有效截止期 T , 令 $X = \{(t_j^k, c_j^k) | 1 \leq k_j \leq l(j), j \in V\}$ 表示 G 的一个可行解, 则所有满足式 (9) 的活动称为 X 的改进活动集, 记作 A .

若当前解的改进活动集中包含多个活动, 则需为活动分配一个优先级, 具有最高优先级的活动优先更新服务选择. 在新可行解下, 重新计算其改进活动集和活动的优先级, 有最高优先级的活动继续更新服务选择. 这个过程重复执行直到 A 为空时, 得到一个费用最优的可行解. 因此, 选择合适的优先级规则是改进阶段要解决的关键问题.

4.2.2 基于时间耦合强度的优先级规则

目前已提出多种用于项目调度的优先级规则^[10,13]. 如最适规则 $BF^{[10]}$, 它将活动当前所选模式 k 和其相邻模式 $k+1$ (模式按时间非递减排列) 的费用与时间的收益比作为优先级计算公式:

$$j = (c_j^k - c_j^{k+1}) / (t_j^{k+1} - t_j^k), j \quad (10)$$

该规则设置最大值的活动优先级最高, 即每次所选活动都以时间的最小增加换取费用的最大降低. 在

前面分析中知道, 时间耦合活动间共享松弛时间. 若 BF 规则中优先级最高的活动在更新服务级时, 其时间耦合活动会因松弛时间减小而退出改进活动集, 它们不能再进一步改进可行解的性能. 该活动的耦合强度越大, 退出改进活动集的活动就会越多, 对改进算法性能的影响就越大. 因此, 若使优先级最高的活动在获得费用最大收益的同时, 又能使其时间耦合活动有机会进一步降低费用, 这样确定的优先级规则会更有效地提高算法的性能. 为此将时间耦合强度和 BF 规则结合, 提出一个新的优先级规则: 时间耦合强度最适规则 $BFICS$, 公式如下:

$$j / TCS_j = ((c_j^k - c_j^{k+1}) / (t_j^{k+1} - t_j^k)) / (|P^*| + |F^*|) \quad (11)$$

该规则仍设置有最大值的活动优先级最高. 由于图中添加两个虚结点, 该优先级规则的计算结果均为有效值 (分母不会为 0).

4.2.3 改进算法流程

综合节 4.2.1 和 4.2.2 描述, 改进算法描述如下:

- (1) 利用式 (7) 和 (8), 计算当前可行解 X 下所有活动的窗口;
- (2) 根据式 (9) 确定可行解 X 的改进活动集 A ;
- (3) 若 $A \neq \emptyset$, 转步骤 (7);
- (4) 根据 $BFICS$ 优先级规则从 A 中选择优先级最高的活动, 记作 v ;
- (5) 从 $SP(v)$ 中选择当前候选服务 k_v 的相邻服务 $k_v + 1$, 替代当前服务分配;
- (6) 更新可行解 X , 转步骤 (1);
- (7) 计算总费用;
- (8) 结束.

4.3 迭代算法的收敛性

基于 $BFICS$ 规则的启发算法是一个迭代算法, 时间复杂度主要取决于改进阶段的迭代次数. 已知迭代次数依赖改进活动集中包含的活动数和服务池的最大长度. 而由定义 5 可知, 改进活动集的集合长度又依赖可行解 X . 令 A 表示改进活动集, 更新其中任意一个活动 j 的服务级后, 得到的新可行解记为 X' , 对应的改进活动集记作 A' . 对两个改进集合 A 和 A' , 有如下性质成立:

性质 1 A' 的集合长度不大于 A 的集合长度, 即 $|A'| \leq |A|$.

证明: 对 $\forall j \in A$, 令 $A = TCA_j$ 表示活动 j 包含在集合 A 中的所有时间耦合活动.

对活动 j , 当其选择服务级别低的服务更新当前服务后, 由式 (7)、(8) 知, 对 $\forall p \in TCA_j$, 在新的可行解 X' 下, 其时间窗口或者不变或者减小, 则 A 中活动的时间窗口也会减小或不变.

在新的时间窗口下,对集合 A 分两种情况讨论:

(1) 若对 $\forall p \in A$ 仍满足式(9),则 A 不变,即

$$|A| = |A|;$$

(2) 若至少 $\exists p \in A$ 不满足式(9),集合 A 不包含 p ,

则 A 也不包含 p ,即 $|A| = |A| - 1 < |A|$.

综合(1)和(2)可知, $|A| \leq |A|$ 成立.

由性质 1,改进活动集的集合长度在迭代过程中是非递增的,即初始改进集合长度具有最大长度.令初始可行解的改进集为 A_s .由于每次活动的服务更新都能至少减少其服务池中一个候选服务.因此改进算法的迭代次数最大不超过 $|A_s| * M$,其中 $M = \max\{l(i) | i \in V\}$ 是所有活动的服务池最大长度,即算法是收敛的.

5 实例设置与实验结果比较

将本文提出的 BFICS 规则与 DTL^[11]、DBL^[12]和基于 BF 规则的启发算法比较.基于 BF 规则的启发算法以最早可行解为初始解,以 BF 规则为改进阶段的优先级计算公式.所有算法均采用 Java 编程,运行于操作系统为 Windows XP,主频 2.93GHz、内存 512M、CPU 为 Pentium IV 的 PC 机上.

5.1 实例生成及性能参数设置

为比较 BFICS 规则与 DTL^[11]、DBL^[12]和 BF 规则的性能和效率,对大量不同结构的工作流实例模拟测试.工作流实例采用随机生成器生成,随机生成器设置 4 个控制参数,分别是应用规模,服务规模,费用函数和截止期.应用规模是工作流实例包含的活动数,设置 $|V| \in \{30, 60, 90, 120\}$,选自项目实例库中的 benchmark 实例^[14],每种规模分别选取 50 个.服务规模表示活动的最大服务池长度,设置 $\max\{l(i) | i \in V\} \in \{10, 20, 30\}$;费用函数表示服务池中的服务所需费用随执行时间的变化规律,选取凹函数 concave(ccv),凸函数 convex(cvx)和混合函数 hybrid(hyb,随机函数)三种类型,其生成方法参见文献[10].截止期用 $T = T_{\min} + \alpha * (T_{\max} - T_{\min})$ 确定 4 个值,即 $\alpha \in \{10\%, 20\%, 30\%, 40\%\}$, T_{\min} 是工作流最小完工时间, T_{\max} 是工作流最大完工时间.综合上面各问题参数,共生 $4 * 3 * 3 * 4 * 50 = 7200$ 个测试实例.

实验结果用平均偏差 (AVDEV)、最优解比率 (OPT%) 及算法平均运行时间 (ART,单位 ms) 作为度量指标,各参数计算公式如下:

$$\text{平均偏差 (AVDEV)} = \frac{1}{N} \sum_{i=1}^N ((U_i(H) - best_i) / best_i) / N$$

$$\text{最好解比率 (OPT \%)} = 100 * C(H) / N$$

$$\text{平均运行时间 (ART)} = \frac{1}{N} \sum_{i=1}^N t_i(H) / N$$

其中 H 表示算法名称, $U_i(H)$ 表示算法 H 在实例 i 上获得的目标函数值; N 是同组实例个数; $C(H)$ 是 H

在 N 个实例上得到的最好解次数; $best_i$ 表示实验中所有算法在实例 i 上获得的最好解; $t_i(H)$ 是算法 H 在第 i 个实例上的运行时间.平均偏差表示算法 H 在测试实例上获得的解与最好解距离的平均值,即稳定性,算法的平均偏差越小,稳定性越高;最优解比率表示算法 N 在个测试实例上获得最好解次数的百分比,体现了算法的最优性.平均偏差和最优性衡量算法的性能,而平均运行时间衡量算法的效率.

5.2 实验结果比较

表 1 给出了 4 个算法在所有实例上的计算结果.从表中数据看出,优先级规则 BFICS 在所有性能参数上均获得了最好值,其平均偏差为 0.004,远远小于其他算法,说明算法在所测试实例上具有很好的稳定性;在所测试的 7200 个实例上,BFICS 的最优解比率为 92.3%,远大于其它几个算法,有最好的最优性.BF 规则与 DBL 相比,BF 规则的平均偏差较好于 DBL 算法,但在最优解比率上较劣于 DBL 算法,说明 DBL 的性能受实例影响较大,稳定性较差.分析上述运行结果,原因在于:(1)BFICS 综合利用两个较好的优先级信息,每次迭代都有很强的局部优化能力,松弛时间能在耦合活动间得到充分共享;(2)BF 规则仅利用服务资源信息迭代优化,DBL 算法仅依赖 DAG 图结构特征分解截止期,两者都仅考虑了单一方面信息.因此两者性能较差.

表 1 四个算法的计算结果

算法名称	AVDEV	OPT(%)	ART
BFICS	0.004	92.3	55.0
BF	0.343	1.8	36.2
DTL	0.734	0.0	0.2
DBL	0.375	5.9	0.2

从运行效率看,截止期分解算法 DTL (DBL) 通过分层一次分解截止期,算法的时间复杂度最小,所需运行时间平均不到 1ms;而优先级算法采用迭代策略搜索局部最优解,其迭代次数依赖于改进活动集的集合长度和服务规模,同时每次迭代都需计算活动的优先级,花费时间较长,BF 规则的平均运行时间约为 36ms,BFICS 规则的平均运行时间约为 55ms,算法也具有较快的运行效率.

6 问题参数对算法的影响

根据第 5 节的实验结果,BFICS 规则虽然所有性能参数上得到了最好结果,但并不能在所有测试实例上产生最好的性能,其性能受所求解问题参数的影响,这一节将测试不同问题参数对算法性能和效率的影响.

表 2 是应用规模对各算法性能和运行效率的影响.

可以看出,BFTCS 规则在不同应用规模上有最好的稳定性和最优性,但随应用规模增大,其性能逐渐下降,BF 规则的性能同样随应用规模增大逐渐下降,而 DBL 和 DTL 的性能随应用规模增大而逐渐提高.这是因为应用规模的扩大使 DAG 图结构变得复杂,算法受 DAG 结构特征的影响愈加明显.由于 BF 规则未考虑 DAG 结构特

征,性能会逐渐下降.而两个截止期算法依据 DAG 结构分解截止期,随应用规模增大,受活动偏序约束的强度明显增加,性能会逐渐提高.BFTCS 优先级规则由于考虑了衡量 DAG 结构特征的时间耦合强度信息,所以在不同规模上获得了最好性能,但由于 DAG 结构的复杂性,性能也受到影响,随应用规模增大性能逐渐下降.

表 2 不同应用规模下各算法的计算结果

算法名称	AVDEV				OPT (%)				ART(ms)			
	30	60	90	120	30	60	90	120	30	60	90	120
BFTCS	0.003	0.003	0.001	0.009	93.3	93.2	97.4	85.3	6.7	27.1	70.7	115.5
BF	0.254	0.318	0.461	0.337	5.4	1.6	0.1	0.0	4.5	18.9	44.8	76.4
DTL	0.798	0.757	0.839	0.541	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.3
DBL	0.475	0.395	0.411	0.219	1.4	5.2	2.4	14.7	0.1	0.1	0.3	0.4

表 3 给出了各个算法在不同截止期下的计算结果.可以看出,BFTCS 规则在不同截止期下获得了最好的稳定性和最优性,远远优于其它几个算法的性能.BF 规则与 DBL 相比,当截止期较小时,DBL 的性能要好于 BF 规则,而当截止期增大时,DBL 要劣于 BF 规则;但从获得的平均偏差结果看,四个算法随截止期的增加,性能都有下降趋势.这是由于当截止期较小时,松弛时间较小,算法性能受 DAG 结构和服务资源信息影响较大;当

截止期较大时,松弛时间较大,优先级信息和 DAG 结构特征的优势减弱.由于 DBL 和 BF 规则仅考虑了实例的单一特征信息,当截止期增大时,较大的松弛时间不能得到较好的利用,DBL 和 BF 规则的性能逐渐下降;而 BFTCS 规则由于综合考虑两方面信息,其性能受截止期影响较小.虽然性能随截止期增加逐渐下降,但也在不同截止期下都获得了最好性能.

表 3 不同截止期约束下各算法的计算结果

算法名称	AVDEV				OPT (%)				ART(ms)			
	10 %	20 %	30 %	40 %	10 %	20 %	30 %	40 %	10 %	20 %	30 %	40 %
BFTCS	0.004	0.002	0.003	0.008	89.6	94.9	93.8	90.8	28.6	48.0	64.5	79.0
BF	0.160	0.305	0.412	0.494	2.4	0.9	1.6	2.2	15.1	29.0	43.2	57.4
DTL	0.304	0.582	0.880	1.169	0.0	0.0	0.0	0.0	0.3	0.2	0.2	0.2
DBL	0.131	0.282	0.450	0.636	8.1	4.1	4.6	7.0	0.2	0.2	0.2	0.3

服务规模对各种算法的影响见表 4.从表 4 的计算结果可以看出,BFTCS 规则在不同服务规模下仍有最好的稳定性和最优性,且性能随服务规模增大有所提高;而其他几个算法随服务规模增加,性能有所下降;但总的变化情况不大.这是因为服务规模的变化仅改变了服务资源费用函数分布.当服务规模较小时,费用函数分布比较稀疏,单个活动的费用收益明显,BF 规则和

两个截止期算法取得了较好性能;而当服务规模增加时,费用函数分布变得密集,单个活动的费用收益有所下降.因此,BF 规则和两个截止期分解算法的性能随服务规模增加有所下降.而 BFTCS 在服务规模较低时利用 BF 规则优势,在服务规模较高时利用时间耦合强度信息的优势,因此性能随服务规模增大有所提高,但总的来说变化不大.

表 4 不同服务规模下各算法的计算结果

算法名称	AVDEV			OPT (%)			ART(ms)		
	10	20	30	10	20	30	10	20	30
BFTCS	0.005	0.004	0.003	90.1	92.8	93.9	25.5	57.1	82.5
BF	0.308	0.313	0.407	1.7	2.2	1.5	16.7	39.5	52.4
DTL	0.656	0.718	0.828	0.0	0.0	0.0	0.2	0.2	0.2
DBL	0.315	0.357	0.452	8.2	5.0	4.7	0.2	0.2	0.3

不同类型的费用函数对各个算法性能和效率的影响见表 5.可以看出,BFTCS 规则在不同费用函数上仍有最好的稳定性和最优性,且在凹函数上表现出最好的性能;DBL 和 DTL 在凸函数上表现了较好性能.从表

中数据还可看出,在凹函数上各算法的性能参数取值差距很大,而在凸函数上各算法参数取值差距较小.原因在于:凸函数的费用曲线随时间增大由陡峭下滑趋于平缓,费用收益初期较大,有利于 BF 规则和两个截

止期算法,获得的性能较好;而凹函数的费用曲线随时间增大由平缓到陡峭下滑,费用收益初期较小,对 BF 规则和两个截止期算法的性能影响较大,BFTCS 规则获得

了最好的性能,平均偏差仅为 0.0001,最优性达到了 97.8%。混合函数由于没有明显变化规律,除 BF 规则外,各个算法获得的性能介于两者之间。

表 5 不同费用函数下各算法的计算结果

算法名称	AVDEV			OPT(%)			ART(ms)		
	ccv	cvx	hyb	ccv	cvx	hyb	ccv	cvx	hyb
BFTCS	0.001	0.008	0.003	97.8	87.1	91.9	49.1	64.2	51.8
BF	0.514	0.305	0.209	1.8	0.0	3.5	32.0	42.3	34.3
DIL	1.095	0.440	0.667	0.0	0.0	0.0	0.2	0.2	0.2
DBL	0.676	0.143	0.305	0.4	12.8	4.6	0.2	0.2	0.3

分析表 2 到表 5 中各算法的运行时间,可以得出 DIL 和 DBL 两个算法在各种情况下均具有最好的运行效率,其运行时间不到 1ms。而对两个优先级算法来说,除费用函数外,应用规模、服务规模和截止期对各算法的运行时间都有较大影响,随这些参数取值的增大,算法的运行时间也增大。

7 结束语

针对 DAG 描述的网格 workflow 时间费用优化问题,本文提出基于优先级规则 BFTCS 的启发式求解策略。优先级规则 BFTCS 综合考虑活动的服务资源信息和时间耦合强度信息,在以时间的最小增加换取费用最大降低的同时,能使松弛时间在时间耦合活动间得到充分共享。实验结果表明,基于 BFTCS 规则的启发算法在所测试的实例集上均获得了最好的性能,且具有很好的运行效率。当然,本文仅对优先级规则的构造方法进行了改进,初始解构建方法还需进一步研究。另外,多维 QoS 的服务选择已受到普遍关注^[15-17],多 QoS 的组合服务优化也是下一步研究的工作。

参考文献:

- [1] I Foster, C Kesselman, J M Nick, S Tuecke. Grid service for distributed system integration [J]. IEEE Computer, 2002, 35 (6): 37 - 46.
- [2] J Yu, R Buyya. Taxonomy of scientific workflow systems for Grid computing [J]. Sigmod Record, 2005, 34(3): 44 - 49.
- [3] L Cao, M L Li, J Cao, et al. Introduction to an Agent-based Grid Workflow Management System [M]. LNCS 3759, 2005. 559 - 568.
- [4] E Deelman, J Blythe et al. Mapping abstract complex workflows onto grid environments [J]. Journal of Grid Computing, 2003, 1(1): 25 - 39.
- [5] R Buyya, D Abramson, J Giddy, H Stockinger. Economic models for resource management and scheduling in grid computing. concurrency and computation [J]. Practice and Experience Journal, 2002, 14(13-15): 1507 - 1542.
- [6] 陈宏伟, 王汝传. 费用-时间优化的网格有向无环图调度

算法 [J]. 电子学报, 2005, 33(8): 1375 - 1380.

Chen Hongwei, Wang Ruchuan. A grid scheduling algorithm for cost-time optimization [J]. Acta Electronica Sinica, 2005, 33 (8): 1375 - 1380. (in Chinese)

- [7] M Lin, Z Lin. A cost-effective critical path approach for service priority selections in grid computing economy [J]. Decision Support Systems, 2006, 42(3): 1628 - 1640.
- [8] P De, E J Dunne, J B Ghosh, C E Wells. The discrete time-cost tradeoff problem revisited [J]. European Journal of Operational Research, 1995, 81(2): 225 - 238.
- [9] P De, E J Dunne, J B Ghosh, C E Wells. Complexity of the discrete time-cost tradeoff problem for project networks [J]. Operations Research, 1997, 45(2): 302 - 306.
- [10] C Akkan, A Drexler, A Kimms. Network decomposition-based benchmark results for the discrete time-cost tradeoff problem [J]. European Journal of Operational Research, 2005, 165 (3): 339 - 358.
- [11] J Yu, R Buyya, C K Tham. Cost-based scheduling of workflow applications on utility grids [A]. Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing [C], Los Alamitos, CA, USA: IEEE Computer Society Press, 2005. 130 - 137.
- [12] 苑迎春, 李小平, 王茜, 张毅. 基于逆向分层的网格 workflow 调度 [J]. 计算机学报, 2008, 31(2): 280 - 290.
Y Yuan, X Li, Q Wang, Y Zhang. Bottom level based heuristic for scheduling workflows in grids [J]. Chinese Journal of Computers, 2008, 31(2): 280 - 290. (in Chinese)
- [13] R Klein. Bidirectional planning: improving priority rule-based heuristics for scheduling resource-constrained projects [J]. European Journal of Operational Research, 2000, 127(4): 619 - 638.
- [14] Project Scheduling problem library — PSPLIB [OL]. <http://129.187.106.231/psplib/>, 2005.
- [15] L Z Zeng, B Benatallah, A H H Ngu, et al. QoS-aware middleware for web services composition [J]. IEEE Transactions on Software Engineering, 2004, 30(5): 311 - 327.
- [16] 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.

Jin Hai, Chen Hanhua, Lu Zhipeng, et al. QoS optimizing model and solving for composite service in CGSP job manager [J]. Chinese Journal of Computer, 2005, 28 (4) : 578 - 588. (in Chinese)

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

作者简介:



苑迎春 女, 1970 年出生于河北省清苑县, 博士, 河北农业大学信息科学与技术学院副教授. 主要研究领域为网格计算、服务组合.
Email : nd. hd. yyc @163. com



李小平 男, 1970 年出生于重庆市开县, 博士, 东南大学计算机科学与工程学院副教授, 博士生导师. 主要研究领域为机器调度、项目调度、服务计算.
Email : xpli @seu. edu. cn

www.cnki.net