



云计算环境下一一种面向科学 workflows 不确定数据源的视图构造方法^{*}

胡海洋, 刘占晨, 胡 华

(杭州电子科技大学计算机学院 杭州 310018)

摘 要:科学 workflows 的数据源视图根据数据源中任务间的数据流关系,将它们划分为多个复合模块,并在此基础上进行数据抽象与封装,从而可有效降低科研工作者的数据分析工作量并节省数据查询时间。然而在云计算环境中开发与应用科学 workflows 系统时,由于受数据采集的准确度和服务器的可靠性影响,将会导致 workflow 数据源图的不确定性,因此需要提供有效的机制在不确定数据源图中构建合理性视图。针对此方面,首先给出了不确定数据源图及其合理性视图的定义,在此基础上提出了一种检测不合理视图的方法;还进一步分析了数据源图中任务节点与其一阶前序节点之间存在的多种数据流关系及复合任务的局部期望支持度,给出了合理视图的构造方法。设计了相应的多项式时间算法,并分析算法的时间复杂度。最后,对相关方法给出示例,并进行实验分析,验证了其可行性与有效性。

关键词:云计算;科学 workflows;数据源;视图

doi: 10.3969/j.issn.1000-0801.2013.03.017

Constructing View of Uncertain Data Provenance for Scientific Workflow in Cloud Computing

Hu Haiyang, Liu Zhanchen, Hu Hua

(School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: The view of data provenance in scientific workflow provides an approach of data abstraction and encapsulation by partitioning tasks in the data provenance graph (DPG) into a set of composite modules due to the data flow relations among them, so as to efficiently decrease the workload consumed by researchers making analysis on the data provenance and the time needed in doing data querying. Nevertheless, developing and applying the scientific workflow systems in cloud computing environments suffers the problem of uncertainty brought by the inaccuracy of data collection and unreliability of data servers distributed in the internet. Concentrating on this scenario, the definitions of uncertain DPG and its sound view were presented firstly, and then a method for detecting the unsound view of DPG was proposed. Also, a method for constructing sound and high-support view was presented, which is based on the data flow relations among the tasks and their first-order preceding tasks in the graph, and the local expected support of the composite modules. A polynomial-time algorithm was designed, and its maximal time complexity was also analyzed. Additionally, an example and conduct comprehensive experiments were given to show the feasibility and effectiveness of the method.

Key words: cloud computing, scientific workflow, data provenance, view

^{*} 国家自然科学基金资助项目(No.61272188, No.60903053),浙江省自然科学基金资助项目(No.Z1100822, No.LY12F02005)

1 引言

近年来,随着计算机网络技术的飞速发展,作为新型互联网的后端支撑基础——云计算技术已开始逐步走向人类社会生活^[1-6]。云计算环境将大量计算资源、存储资源与软件资源链接在一起,形成巨大规模的共享虚拟IT资源池,从而为海量数据密集型的计算应用提供了广阔的平台^[7,8]。科学工作流(scientific workflow)是一类典型的面向海量密集型数据流、以减少计算成本为目标的应用系统,它的出现为跨多学科和跨不同地域的科研合作提供了方便。在科学工作流系统中,海量数据密集型计算与分布式资源共享彼此间协作、无缝地进行,使得具有不同领域专业知识、地理位置分布的科研工作者可跨组织合作完成相关的科学实验。科学工作流系统现已在各国的科研院所展开应用^[9-12],并在生物信息学、社会关系分析、气象学、物理学等多个领域都发挥了应有的作用^[9-15]。

在云计算环境下进行科学工作流运行时,将产生大量的中间数据和结果,对这些数据进行管理与分析有助于科研工作者对复杂、多步骤的科学实验进行描述和分析。为了保证实验结果的可重复性与可验证性,许多科学工作流系统都提供了对数据源(data provenance)的支持。科学工作流系统中的数据源采集了数据产品的历史缘由,包括原始数据来源、中间数据产品以及应用到数据产品的生产步骤;它一般被定义成工作流中的任务/活动执行及数据传输的关系依赖图,记载科学实验的详细协议。由于复杂、大规模的科学工作流系统在运行过程中将产生大量的数据源。如何对这些海量的数据源进行有效的查询与管理现已成为科研工作者研究的重要问题之一,目前许多研究工作针对此方面展开,它们通过构造数据源视图(view of data provenance)的方式分析数据源,从而达到缩减数据源视图的规模和计算复杂度的目的^[10,16]。在数据源视图中,科学工作流中的任务依据数据流关系,被抽象成高阶(high level)的复合任务,从而达到隐藏不相关的源数据细节、降低数据源图规模的目的,同时在视图级进行数据查询将有效地提高查询效率。

然而在云计算环境中,科学工作流系统运行时往往需要处于异地的多台科研服务器协同工作,运行过程产生的数据源也将存储在这些服务器中。由于受数据采集的准确度和服务器的可靠性影响,当对这些来自不同服务器的数据源进行数据集成(data integration)时,工作流数据的不确

定性/不一致性现象将会产生。一般说来,这样的数据不确定性现象可用概率的形式加以量化^[17,18];在此基础上,不确定科学工作流中的数据源可被定义成任务/活动执行的概率图形式。

目前,面向数据源视图的研究工作主要集中在传统的确定式数据源视图技术方面,而针对不确定数据源视图的研究工作尚未充分展开。基于以上分析,本文首先给出了不确定数据源视图及其合理性视图的定义,在此基础上提出了一种检测不合理视图的方法。本文还进一步分析了数据源视图中任务节点与其一阶前序节点之间存在的多种数据流关系及复合任务的局部期望支持度,给出了多项式时间内的合理视图构造算法,并分析该算法的时间复杂度。最后,本文对所提出的方法进行示例与实验分析,验证了该方法的合理性与有效性。

2 相关工作

近年来面向科学工作流的数据源查询已得到学术界的重视,产生了一系列用来减少数据源查询的复杂性的技术。

参考文献[10]提出了用户视图的概念,并阐述了它在数据源查询中的作用。在此基础上,参考文献[10]设计了一种基于用户需求的构造工作流视图的方法,在该方法构造的复合任务中只有一个用户关心的模块以及若干其他非相关模块,从而能够有效地满足数据源查询的需求。

参考文献[11]在参考文献[10]的基础上,对视图的构造进行了优化,提出了在仅包含串、并联结构的数据源图中时间复杂度更优的算法。

参考文献[16]提出了视图合理性的概念,在此基础上提出了检测不合理视图的方法,并设计了弱局部最优化算法和强局部最优化算法,用于对不合理视图进行重构使之成为合理视图。然而,参考文献[16]中的算法仅针对确定式数据源图,并且依据任务节点是否拥有共同的输入数据流节点集合,将它们划分为多个集合并加以检测。因此,当对某些具有特定结构的数据源图进行视图构造时,将会导致死循环现象的发生,从而限制了算法的进一步应用。

此外,还有一些科学工作流系统允许用户手动构建视图(如 Traverna/MyGrid 系统)^[9],允许用户通过图接口进行视图构造。

另一方面,在确定式图查询技术方面,研究者也提出了许多方法^[19-24],相关的研究成果主要集中在子图查询处



理^[22]、可达查询^[23]、受限可达查询^[24]等方面。

总体说来,目前在科学 workflow 视图构建技术方面的研究尚未深入,现有工作则主要集中在确定式数据源图方面;而面向不确定式数据源视图的合理性检验与构建技术方面的相关研究工作还有待进一步开展。因此本文针对此方面进行研究,提出了在不确定数据源图中检测视图合理性的方法,并进一步基于任务间的数据流和局部期望支持度给出了构建不确定式视图的方法。

3 基本模型

定义 1 科学 workflow 中的不确定数据源图可表示成一个有向无环图 $G = \{V_G, E_G, \pi_G, L_G\}$, 其中 $V_G = \{v_1, \dots, v_n\}$ 表示数据源中的任务集合, $E_G = \{e_{ij}\}$ 代表任务之间的数据流关系集合, π_G 为图中数据流边的存在概率, 即 $\forall e_{ij} \in E_G, 0 \leq \pi_G(e_{ij}) \leq 1$, 且对 $\forall v_i \in V_G, \sum_{v_j \in V_G} \pi_G(e_{ij}) = 1$; L_G 则定义了

图中节点 V_G, E_G 的标签。

定义 2 若图 $G' = \{V_{G'}, E_{G'}, \pi_{G'}, L_{G'}\}$ 是不确定数据源图 $G = \{V_G, E_G, \pi_G, L_G\}$ 中所蕴含的一个确定性子图, 则有:

- $V_{G'} \subseteq V_G, E_{G'} \subseteq E_G$;
- $\forall v_i \in V_{G'}, L_{G'}(v_i) = L_G(v_i)$;
- $\forall e_{ij} \in E_{G'}, L_{G'}(e_{ij}) = L_G(e_{ij}), \pi_{G'}(e_{ij}) = 1$ 。

一个不确定数据源图可蕴含着多个确定性子图。若不确定数据源图 $G = \{V_G, E_G, \pi_G, L_G\}$ 蕴含某确定子图 $c_g = \{V', E', \pi', L'\}$ (记作 $G \Rightarrow c_g$), 则蕴含的概率可计算为:

$$\Pr(G \Rightarrow c_g) = \prod_{e_{ij} \in E_G} \pi_G(e_{ij}) \cdot \prod_{e_{ij} \in E_G - E'} (1 - \pi_G(e_{ij})) \quad (1)$$

数据源视图^[10-12, 16]通过将数据源图的所有任务划分为多个复合任务的方法, 提供了对数据源图进行高层抽象的途径, 其可被形式化定义如下。

定义 3 数据源图 G 的视图 $W = \{V_W, E_W, \pi_W, L_W\}$ 是对数据源图的一种高层抽象, 其将 G 中的节点划分为多个不相交的节点集 $V_W = \{P_1, P_2, \dots, P_n\}$, 即 $\forall P_i, P_j \in V_W, P_i \cap P_j = \emptyset, \bigcup_{i=1}^n P_i = V_G$, 且 G 的边集 E_G 和 W 的边集 E_W 构成了从 E_G 到 E_W 的满射 $\psi: E_G \rightarrow E_W$, 即对 $\forall v_p, v_q \in V_G$, 若 $v_p \in P_i, v_q \in P_j (P_i \neq P_j)$, 且存在有边 $e_{pq} \in E_G$, 则有边 $(P_i, P_j) = \psi(e_{pq}) \in E_W$; 若 $v_p \in P_i, v_q \in P_j (P_i = P_j)$, 则 $\psi(e_{pq}) = \varepsilon$ 。

在本文中, 任一 $v_i \in V_G$ 被称为原子任务, 对于任一节点 $P_i \in V_W$, 若 P_i 内部包含多个原子任务, 则它被称为复合

任务。显然, 原子任务是一类特殊的复合任务, 即其内部仅包含其自身; 此外, 复合任务 P_i 也是数据源图 G 的一个子图。

对于图 $G_i = \{V_{G_i}, E_{G_i}\}, G_j = \{V_{G_j}, E_{G_j}\}$ 而言, 若有 $V_{G_j} \subseteq V_{G_i}$ 且 $E_{G_j} \subseteq E_{G_i}$, 本文则也称 G_j 蕴含在 G_i 中。下面基于不确定图数据库中子图期望支持度的概念^[17], 给出计算复合任务期望支持度的方法。

定义 4 对于复合任务 P_i , 其蕴含的确定子图集合为 $Cset_i = \{cg_1, cg_2, \dots, cg_n\}$ 。对于 P_i 中所蕴含的某子图 G_j 而言, 若 G_j 能够同时蕴含在 $Cset_i$ 中的某确定子图 cg_k 内, 则其在 P_i 中的期望支持度可表示为如下形式:

$$eaup(G_j | P_i) = \sum_{cg_k \in Cset_i \wedge cg_k \Rightarrow G_j} \sup_{P_i}(cg_k) \cdot \Pr(P_i \Rightarrow cg_k) \quad (2)$$

其中, 相应的支持度可表示为 $\sup_{P_i}(cg_k) = \frac{1}{|Cset_i|}$; 在

式(2)中, 若 $G_j = P_i$, 则也用 $eaup(P_i)$ 来简化表示 $eaup(P_i | P_i)$ 。

根据定义 3, 视图必须能够保护复合任务之间的数据流; 直观地讲就是数据源图中的两个节点之间存在一条数据流路径, 那么在视图对应的两个节点间也必须存在一条路径, 这样的视图能够保护相邻任务之间的数据流^[16]。反之, 若构造出的视图中某两个节点之间存在一条路径, 而在数据源图中不存在相应的路径, 则这样的视图将不能涵盖数据源图中的数据流关系, 并且传达着错误的信息, 从而导致不正确的数据源查询结果。这样构造出的视图被称为非合理性视图(unsound view)。此外, 当前复合任务的期望支持度须具有局部最大期望支持度, 即任意复合任务与其前后相邻的其他复合任务无法实施进一步合并。基于以上分析, 现给出面向不确定数据源图 G 的合理性视图(sound view)定义如下。

定义 5 不确定数据源图 G 的视图 $W = \{V_W, E_W, \pi_W, L_W\}$ 当且仅当满足如下条件时, 该视图为合理性视图。

(1) $\forall P_i, P_j \in V_W$, 对于从 P_i 到 P_j 的任一路径若包含边 $L_1, L_2, \dots, L_p$, 则在不确定数据源图 $G = \{V_G, E_G, \pi_G, L_G\}$ 中存在对应的一条路径 $\psi^{-1}(L_1) \cdot C_1, \psi^{-1}(L_2) \cdot C_2, \dots, \psi^{-1}(L_p) \cdot C_p$, 其中 $C_m (1 \leq m \leq p-1)$ 是复合任务中的内部路径。

(2) $\forall v_i, v_j \in V_G$, 对于从 v_i 到 v_j 的任一路径若包含 $l_1, l_2, \dots, l_p$, 那么在 V_W 中存在一条对应的路径: $\psi(l_1), \dots, \psi(l_p)$ 。

(3) $\forall P_i \in V_W$, 若存在 $P_j \in V_W$, 且 $(P_i, P_j) \in E_W$ 或 $(P_j, P_i) \in E_W$, 则对 P_i, P_j 合并后生成的新复合任务 $P_k = P_i \oplus P_j$ 及新的视图 $W' = \{V_{W'}, E_{W'}, \pi_{W'}, L_{W'}\}$ 而言, 其将无法同时满足如下条件:

- W' 满足条件(1)和条件(2);
- $eaup(P_k) > eaup(P_i|P_k)$ 且 $eaup(P_k) > eaup(P_j|P_k)$ 。

在定义5中,条件(1)和条件(2)用于保护任务之间的数据流关系;而对于条件(3),则可以理解为当前复合任务的期望支持度是局部最大期望支持度,即任意复合任务 P_i 与其前后相邻的其他任务 P_j 合并之后,无法同时既满足条件(1)和条件(2),又使得合并后的新复合任务的期望支持度大于 P_i 和 P_j 的期望支持度。

4 算法描述

在该部分中首先描述了如何检测不确定数据源视图中的非合理性复合任务,然后对非合理性复合任务进行分解与重构,给出相应的规则与算法。

定义6 对于复合任务 $P=\{V_P, E_P\}$ 内的任一节点 $v_i \in V_P$ 而言,它在 P 中的前序节点集可表示为 $H_P(v_i)=\{v_j|v_j \in V_P, \text{且在 } P \text{ 中存在有路径 } \rho=v_j \cdots v_i\}$;特别的是,若有 $(v_j, v_i) \in E_P$,则称 v_j 是 v_i 在 P 中的一阶前序节点, v_i 在 P 中的一阶前序节点集合,记作 $H_P(v_i, 1)$ 。

定义7 对于复合任务 $P=\{V_P, E_P\}$ 内的任一节点 $v_i \in V_P$ 而言,它在 P 中的后续节点集可表示为 $Q_P(v_i)=\{v_j|v_j \in V_P, \text{且在 } P \text{ 中存在有路径 } P=v_i \cdots v_j\}$;特别的是,若有 $(v_i, v_j) \in E_P$,则称 v_j 是 v_i 在 P 中的一阶后续节点, v_i 在 P 中的一阶后续节点集合,记作 $Q_P(v_i, 1)$ 。

在定义6、定义7的基础上,本文给出复合任务 P 的输入与输出节点集定义如下。

定义8 对于数据源图 G 中的某复合任务 $P=\{V_P, E_P\}$ 而言,其输入节点集为 $IS(P)=\{v_i|v_i \in V_P, \text{且 } H_P(v_i, 1) \neq H_G(v_i, 1)\}$ 。

显然,若 v_i 是 P 的一个输入节点,则存在 $v_j \in H_G(v_i, 1)$ 且 $v_j \notin V_P$ 。

定义9 对于数据源图 G 中的某复合任务 $P=\{V_P, E_P\}$ 而言,其输出节点集为 $OS(P)=\{v_i|v_i \in V_P, \text{且 } Q_P(v_i, 1) \neq Q_G(v_i, 1)\}$ 。

显然,若 v_i 是 P 的一个输出节点,则存在 $v_j \in Q_G(v_i, 1)$ 且 $v_j \notin V_P$ 。

定义10 设数据源图 G 中有复合任务 $P=\{V_P, E_P\}$,则对于 P 内的任一节点 $v_i \in V_P$ 而言,它在 P 中的输入节点集为 $is_P(v_i)=\{v_j|v_j \in H_G(v_i, 1) \cap IS(P)\}$ 。

4.1 不合理复合任务的检测

根据上述概念及定义5可知,若数据源图 G 的视图 $W=\{V_W, E_W, \pi_W, L_W\}$ 为合理性视图,当且仅当 W 中所有的任务均是合理的,即定义5中的条件(1)和(2),可改写为这

样的条件^[16]: $\forall P \in V_W, \forall v_i \in OS(P)$, 则有 $is_P(v_i)=IS(P)$ 。在此基础上,定义5可改写为如下形式。

定义11 数据源图 G 的视图 $W=\{V_W, E_W, \pi_W, L_W\}$ 为合理性视图,当且仅当 W 中所有的复合任务均是合理的,即有:

(1) $\forall P \in V_W, \forall v_i \in OS(P)$, 则有 $is_P(v_i)=IS(P)$;

(2) $\forall P_i \in V_W$, 若有 $P_j \in V_W$, 且 $(P_i, P_j) \in E_W$ 或 $(P_j, P_i) \in E_W$, 则 P_i, P_j 合并后的复合任务 $P_k=P_i \oplus P_j$ 无法同时满足如下条件。

· $\forall v_i \in OS(P_k)$, 则有 $is_{P_k}(v_i)=IS(P_k)$ 。

· $eaup(P_k) > eaup(P_i|P_k)$ 且 $eaup(P_k) > eaup(P_j|P_k)$ 。

根据上述定义可知,对于任意复合任务 P ,如果 $\exists v_i \in OS(P)$, 而 $is_P(v_i) \neq IS(P)$, 则显然 P 是不合理复合任务;此外, P 的期望支持度须是局部最优的,即无法和其前序或后续的其他任务组合,进而生成期望支持度更优的合理性复合任务。根据以上分析,设计相应的视图合理性检测及其重构(Check_ConstructSoundView, CCSV)算法。对于输入的数据源视图,CCSV 算法首先通过函数 $IsSound$ 来检测视图中的每个复合任务是否合理;若发现非合理的视图,它将调用 ConstructSoundView 对其进行重新构造(对算法 ConstructSoundView 的讨论将在4.2节和4.3节中给出)。

检测与构造合理视图的算法如下。

算法1: Check_ConstructSoundView (CCSV)

Input: $W=\{V_W, E_W, \pi_W, L_W\}, G=\{V_G, E_G, \pi_G, L_G\}$

Output: $W'=\{V_W, E_W, \pi_W, L_W'\}$ /* W' is sound and it is reconstructed from W^* /

for each $P_j \in V_W$ do

if(! IsSound (P_j))

ConstructSoundView (P_j, G); //by Algorithm 5

else

for each $P_i \in V_W$ do

if($(P_i, P_j) \in E_W \parallel (P_j, P_i) \in E_W$)

if

$eaup(P_i|P_i \oplus P_j) < eaup(P_i \oplus P_j) \wedge eaup(P_j|$

$P_i \oplus P_j) < eaup(P_i \oplus P_j)$

merge P_i with P_j , and put $P_i \oplus P_j$ into W ;

endfor

return W ;

Function: IsSound (P_i)

Calculate $IS(P_i)$ and $OS(P_i)$;

for each $v_k \in OS(P_i)$ do



```

    calculate  $is_{P_i}(v_k)$ ;
    if  $is_{P_i}(v_k) \neq IS(P_i)$ 
        return false;
    endfor
    return true;

```

4.2 任务节点的合并规则

对于在函数 $IsSound$ 中所检测到的不合理复合任务 $P=(V_P, E_P)$, 对其进行拆分和重新构造, 生成数个合理的复合任务。此外, 由于在确定数据源图中构造优化视图是一个 NP 完全问题^[10], 本文将采用基于贪心方法 (greedy method) 的构造算法, 即从 P 的输出节点集 $OS(P)$ 开始, 逐层判断当前节点和其一阶前序节点是否可以合并。

下面将对不合理复合任务 P 内部的某合理复合任务 P_i 能否和其一阶前序节点合并进行具体讨论。

规则 1 对于合理复合任务 $P_i \in V_P$, 若 $H_C(P_i, 1) = \{v_A\} \subseteq V_P$, 且 $eaup(P_i \oplus v_A) > eaup(P_i | P_i \oplus v_A) \wedge eaup(P_i \oplus v_A) > eaup(v_A | P_i \oplus v_A)$, 则 P_i 可与 v_A 合并成新的复合任务。

规则 2 对于合理复合任务 $P_i \in V_P$, 若存在 $v_A \in V_P$, 如果 $Q_C(v_A, 1) = \{P_i\}$ 且 $eaup(P_i \oplus v_A) > eaup(P_i | P_i \oplus v_A) \wedge eaup(v_A | P_i \oplus v_A)$, 则 P_i 可与 v_A 合并成新的复合任务。

在规则 1、规则 2 中, G 标识整个 workflow 数据源图。下面, 通过定理 1 说明上述规则的正确性。

定理 1 根据规则 1 和规则 2 得到的新复合任务是合理的。

证明 假定 P_i 为一个已存在的合理复合任务 (初始状态的合理复合任务可只包含一个简单任务), 如果 v_A 是 P_i 唯一的一阶前序节点 (如规则 1 所示) 或 P_i 是 v_A 唯一的一阶后序节点 (如规则 2 所示), 则将 v_A 合并到 P_i 之后, 所有指向 v_A 的输入流将能到达 P_i 的其中某个输入节点; 又因为 P_i 是合理复合任务, 从 P_i 的任意输入点出发的数据流肯定能到达 P_i 的所有输出节点 (根据定义 11)。因此从 v_A 出发的数据流能到达 P_i 的所有输出节点。此外, 由于 $eaup(P_i \oplus v_A) > eaup(P_i | P_i \oplus v_A) \wedge eaup(P_i) > eaup(v_A | P_i \oplus v_A)$, 因此, 合并了节点 v_A 的复合任务 P_i 仍然是合理复合任务。

在科学 workflow 系统运行过程中, 由于受数据采集的准确度和服务器的可靠性等因素影响, 将导致数据源信息的不确定性现象产生。这样的不确定性现象通常体现为某些任务之间存在着小概率数据流边, 而这些小概率数据流边的存在, 将直接影响着节点之间能否正确合并。在此情形下, 设有 $v_A \in H_C(P_i, 1)$, $|H_C(P_i, 1)| = 1 \wedge |Q_C(v_A, 1)| > 1$, 则将考察

v_A 的其他后续节点集: $\{v_j | v_j \in Q_C(v_A, 1) \wedge v_j \neq P_i\}$ 。若有 $\sum v_j \pi(V_A, V_j) < \theta$, 则 v_A 到这些节点的数据流可被视为由于系统误差所引起的小概率数据流, 这里 θ 为用户设定的经验阈值。基于以上的分析, 本文现给出针对此情形的合并规则如下。

规则 3 对于节点 $P_i \in V_P$, 设 $|H_P(P_i, 1)| = |H_C(P_i, 1)| = \{v_A\} \wedge |Q_C(v_A, 1)| > 1$; 在此情形下, 对所有 $v_j \in Q_C(v_A, 1) \wedge v_j \neq P_i$, 若有 $\sum v_j \pi(V_A, V_j) < \theta$, 且有 $eaup(P_i \oplus v_A) > eaup(P_i | P_i \oplus v_A) \wedge eaup(P_i \oplus v_A) > eaup(v_A | P_i \oplus v_A)$, 则 P_i 可与 v_A 合并成新的复合任务, 同时对 $\forall v_j \in Q_C(P_i, 1) \wedge v_j \neq P_i$, 可删除数据流边 (v_A, v_j) 。

定理 2 根据规则 3 得到的复合任务是合理的。

证明 根据规则 3, 当 v_A 合并到 P_i 中时, 从 v_A 指向其他任务的小概率数据流将被删除。这样使得 v_A 合并到 P_i 中不会增加额外的输出流, 而 v_A 的输出流能够到达 P_i 的某个输入节点, 又因为 P_i 是合理的, 所以从 v_A 出发的数据流能到达 P_i 所有的输出节点。又由 $eaup(P_i \oplus v_A) > eaup(P_i | P_i \oplus v_A) \wedge eaup(P_i \oplus v_A) > eaup(v_A | P_i \oplus v_A)$, 所以 v_A 合并到 P_i 中后 $P_i \oplus v_A$ 是合理复合任务。

根据以上 3 条规则, 设计了相应的检测及合并算法 Merge_Single_Pre。在该算法中, 将判断当前节点 P_i 和一阶前序节点能否合并。

合并单个一阶前序节点的算法 (Merge_Single_Pre) 如下。

算法 2: Merge_Single_Pre

Input: P_i and v_A ; /* v_A is the first-order preceding vertex of P_i in the unsound composite task P */

Output: true or false; /* if P_i can be merged with v_A , return true; otherwise, return false */

if $(H_C(P_i, 1) = H_P(P_i, 1) = \{v_A\} \wedge eaup(P_i \oplus v_A) > eaup(P_i | P_i \oplus v_A) \wedge eaup(P_i \oplus v_A) > eaup(v_A | P_i \oplus v_A))$

return true;

if $(Q_C(v_A, 1) = Q_P(v_A, 1) = \{P_i\} \wedge eaup(P_i \oplus v_A) > eaup(P_i | P_i \oplus v_A) \wedge eaup(P_i \oplus v_A) > eaup(v_A | P_i \oplus v_A))$

return true;

if $\sum v_j \pi(v_A, v_j) < \theta \quad // \quad v_j \in Q_C(v_A, 1), v_j \neq P_i$

if $eaup(P_i \oplus v_A) > eaup(P_i | P_i \oplus v_A) \wedge eaup(P_i \oplus v_A) > eaup(v_A | P_i \oplus v_A)$

delete all the data flow edges out from v_A except the edge (v_A, P_i) ;

return true;

endif

return false;

另一方面,对于 P_i 和它的前序节点,若它们之间满足如下关系,也可以合并成新的复合任务。

规则 4 对于节点 $P_i \in V_P$, 设存在有任务 $v_A \in H_P(P_i, 1)$, 且对 $\forall v_j \in H_P(P_i, 1) - \{v_A\}$, 有 $Q_C(v_j, 1) = \{v_A, P_i\}$, 在此条件下若有 $e_{aup}(P_i \oplus H_P(P_i, 1)) > e_{aup}(P_i | P_i \oplus H_P(P_i, 1))$, 且对 $\forall v_j \in H_P(P_i, 1)$, $e_{aup}(P_i \oplus H_P(P_i, 1)) > e_{aup}(v_j | P_i \oplus H_P(P_i, 1))$, 则 P_i 可与 $H_P(P_i, 1)$ 中的任务合并成新的合理复合任务。

定理 3 根据规则 4 得到的复合任务是合理的。

证明如下。

(1) 若 $H_P(P_i, 1) = H_C(P_i, 1)$, 则对 $H_P(P_i, 1)$ 中的节点 v_A 而言, 从 $H_P(P_i, 1)$ 中的其他节点出发的数据流均能到达 v_A , 则将 $H_P(P_i, 1)$ 合并到复合任务中之后, 对 $\forall v_j \in H_P(P_i, 1) - \{v_A\}$, 从 v_j 出发的数据流均能通过节点 v_A 到达 P_i 中的某些输入节点; 又根据 $Q_C(v_j, 1) = \{v_A, P_i\}$, 因此从 v_j 出发的另一条数据流将到达复合任务 P_i 。这样使得从集合 $H_P(P_i, 1)$ 中输出的所有数据流都将直接或间接到达 P_i ; 此外, $e_{aup}(P_i \oplus H_P(P_i, 1)) > e_{aup}(P_i | P_i \oplus H_P(P_i, 1))$, 且对 $\forall v_j \in H_P(P_i, 1)$, $e_{aup}(P_i \oplus H_P(P_i, 1)) > e_{aup}(v_j | P_i \oplus H_P(P_i, 1))$ 。因此, 合并了 $H_P(P_i, 1)$ 的复合任务 P_i 同样是合理复合任务。

(2) 若 $H_P(P_i, 1) \neq H_C(P_i, 1)$ (如图 1 所示), 即存在一点 $v_B \in H_C(P_i, 1)$ 且 $v_B \notin H_P(P_i, 1)$, 即 $v_B \notin P_i$ 。因为 v_B 是 P_i 的一阶前序节点, 所以从 v_B 到 P_i 的输入边将指向 P_i 的某个输入节点, 又因为 P_i 是合理复合任务, 所以从 v_B 到 P_i 的输入数据流将能到达 P_i 的所有输出节点, 所以 v_B 的存在不会影响 P_i 与 $H_P(P_i, 1)$ 的合并。

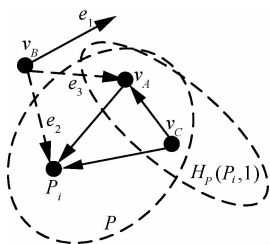


图 1 规则 4 的示例说明

根据以上规则 4, 设计了相应的检测及合并算法 Merge_Multiple_Pre。在该算法中, 将判断当前节点 P_i 和其多个一阶前序节点能否合并。

合并多个一阶前序节点的算法如下。

算法 3: Merge_Multiple_Pre

Input: P_i 和 $H_P(P_i, 1)$;

Output: true or false;

/* if P_i can be merged with vertexs in $H_P(P_i, 1)$,

return true; otherwise return false*/

canMerged=true; len= 0;

for each v_A in $H_P(P_i, 1)$ do

canMerged $\leftarrow$ true;

len ++;

for each v_j in $H_P(P_i, 1) - \{v_A\}$ do

if $Q_C(v_j, 1) = \{v_A, P_i\}$

continue;

else

canMerged $\leftarrow$ false;

break;

endif

endfor;

if (canMerged)

if $e_{aup}(P_i \oplus H_P(P_i, 1)) > e_{aup}(P_i | P_i \oplus H_P(P_i, 1)) \wedge$

$\forall v_j \in H_P(P_i, 1), e_{aup}(P_i \oplus H_P(P_i, 1)) > e_{aup}(v_j | P_i \oplus H_P(P_i, 1))$

return true;

endif;

if (len > | $H_P(P_i, 1)$ |)

return false;

对于复合任务 P 中的部分节点集 $PT = \{v_1, \dots, v_i\} \subseteq V_P$ 而言, 若它们具有共同的前序节点集 $H_P(PT, 1)$, 且满足如下关系, 则也可以合并成新的复合任务。这里 $H_P(PT, 1) = \bigcup H_P(v_i, 1), \forall v_i \in PT$ 。

规则 5 对于 P 中的部分节点集 $PT = \{v_1, \dots, v_i\} \subseteq V_P$ 而言, 若 PT 中节点同时满足如下条件:

- $\forall v_i, v_j \in PT$, 均有 $H_P(PT, 1) = H_C(v_i, 1) = H_C(v_j, 1)$;
- $\forall v_j \in H_C(v_i, 1), Q_C(v_j, 1) = PT$; 则令 P_c 为 PT 和 $H_C(v_i, 1)$ 中所有节点合并而成的新复合任务, 若同时有对 $v_j \in PT \cup H_P(PT, 1), e_{aup}(v_j | P_c) < e_{aup}(P_c)$ 。

则 P_c 为合理的复合任务。

证明 显然在新生成的复合任务 P_c 中, $OS(P_c) = PT$, $IS(P_c) = H_P(PT, 1)$ 。从上述合并方式中可以看出, 对 $\forall v_k \in OS(P_c)$, 均有 $is_{P_c}(v_k) = H_P(PT, 1) = IS(P_c)$; 且对 $\forall v_i \in PT \cup H_P(PT, 1)$, $e_{aup}(v_i | P_c) < e_{aup}(P_c)$ 。故 P_c 为合理的复合任务。

根据以上规则 5, 设计了相应的检测及合并算法 Merge_Com_Pre。在该算法中, 将判断当前节点集能否和其公共的一阶前序节点集合并。



合并公共一阶前序节点的算法如下。

算法 4: Merge_Com_Pre

Input: PT and $H_P(PT,1)$;

Output: true or false;

/* if PT can be merged with its $H_P(PT,1)$,

return true; otherwise return false*/

for each $v_i \in H_P(PT,1)$ do

if $(Q_G(v_i,1) \neq PT)$

return false;

$P_k \leftarrow \Phi$;

for each $v_i \in PT \cup H_P(PT,1)$ do

$P_k \leftarrow P_k \oplus v_i$;

for each $v_i \in PT \cup H_P(PT,1)$ do

if $eaup(v_i|P_k) > eaup(P_k)$

return false;

return true;

4.3 重构算法与时间复杂度

在上述合并规则的基础上,本文现给出不确定数据源图中合理视图的构造 (ConstructSoundView, CSV) 算法如下。

算法 5: ConstructSoundView

Input: unsound P and G ;

Output: a set of sound nodes $\{v_1, v_2, \dots, v_k\}$;

Queue $q \leftarrow \text{NULL}$; Set $s \leftarrow \Phi$;

compute $IS(P)$ and $OS(P)$;

for each v_k in P do

compute $H_C(v_k,1)$ and $Q_C(v_k,1)$;

for each v_k in $OS(P)$ do

push v_k into q ;

while $(q \neq \text{NULL})$

$q' \leftarrow q$;

pop all the elements from q ;

sort the elements $\{v_k\}$ due to the number of their

$\{H_C(v_k,1)\}$;

find those elements which share a common set of

parents:

$\Omega(P) = \{PT_1, PT_2, \dots, PT_n\}$;

for each PT_j in $\Omega(P)$ do

if Merge_Com_Pre ($PT_j, H_P(PT_j,1)$)

$P_c \leftarrow \Phi$;

for each $v_i \in PT \cup H_P(PT,1)$ do

$P_c \leftarrow P_c \oplus v_i$;

push P_c into q ;

else

Set $h \leftarrow \Phi$;

for each v_k in PT_j do

$h \leftarrow h \cup H_P(v_k,1)$;

push all elements in PT_j and h into q ;

endif ;

endfor;

pop all elements from q and put them into s ;

for each element v_k in s do

if Merge_Multiple_Pre ($v_k, H_P(v_k,1)$)

for each v_A in $H_P(v_k,1)$ do

$v_k \leftarrow v_k \oplus v_A$;

put v_k into q ;

else

for each v_A in $H_P(v_k,1)$ do

if Merge_Single_Pre(v_k, v_A)

$v_k \leftarrow v_k \oplus v_A$;

put v_k into q ;

end for;

if $q = q'$

return q ;

end while;

在该算法中,对于检测到的不合理复合任务 P_i ,首先找出其输入节点集和输出节点分别存放在 $IS(P_i)$ 和 $OS(P_i)$ 集合中;接着将 $OS(P_i)$ 中的节点推入队列 q 中;接下来,对队列中的节点及它们的一阶前序节点集按照合并规则 Merge_Single_Pre、Merge_Multiple_Pre 和 Merge_Com_Pre 进行合并,并将合作生成的新复合任务重新推入队列中。当某次扫描队列 q 后,发现没有节点再能与其他节点合并,则算法终止,并返回队列中的元素作为所生成合理视图中的复合任务集。

下面对本文所提出算法的时间复杂度进行分析。当检测复合任务的合理性时,对每个复合任务 $P_i \in V_w$ 需要计算 $IS(P_i)$ 和 $OS(P_i)$ 。设 $P_i = \{V_i, E_i\}$, 则计算 $IS(P_i)$ 和 $OS(P_i)$ 的时间复杂度为 $O(|V_i|)$;此外,对于 $\forall v_j \in OS(P_i)$, 计算 $is_{P_i}(v_j)$ 的时间复杂度为 $O(|V_i|)$ 。因此,检测复合任务 P_i 是否合理,时间复杂度为 $O(|V_i|^2)$ 。

当利用 CSV 算法来重新构造合理复合任务时,在最坏情况下每个点都是一个复合任务,任意点之间不能合并,且每一个点包含的一阶前序节点为最多。则在此情形下,所需的时间复杂度为: $2n+1+\sum_{i=3}^{n-1}(\frac{i}{2}+n+1-i)$, $n=|V_c|$ 。故 CSV 算法的时间复杂度也为 $O(|V_c|^2)$ 。基于以上分析,整个算法的时间复杂度为 $O(|V_c|^2)$, 其中 $|V_c|$ 为整个数据源图 G 中的节点数。

4.4 示例

本节通过一个示例(如图2所示)来描述 CSV 算法的执行流程。在该示例中,假设用户设定的阈值 θ 为 0.15,为了使图的演示更加清晰,在图3中笔者省略了图中边的概率,具体概率值参考图2。

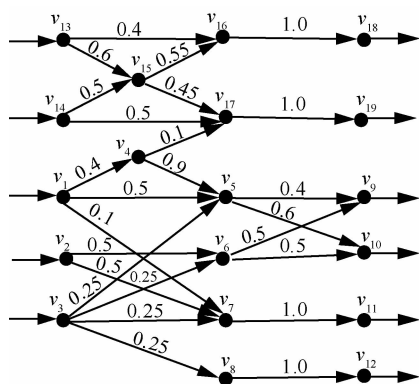


图2 CSV 算法的示例

在初始状态下,整个图2看作视图的一个复合任务,找出其输入节点集 $IS=\{v_1, v_2, v_3, v_{13}, v_{14}\}$ 和输出节点集 $OS=\{v_9, v_{10}, v_{11}, v_{12}, v_{18}, v_{19}\}$ 。显然对于 $v_{19} \in OS$, 有 $is_c(v_{19})=\{v_1, v_{13}, v_{14}\} \neq IS$, 因此需要对图2进行构造。将 OS 中的节点推入队列 q ; 然后对 q 中的节点按照其一阶前序节点集进行排序, 可得: $H_c(v_9, 1)=\{v_5, v_6\}$, $H_c(v_{10}, 1)=\{v_5, v_6\}$, $H_c(v_{11}, 1)=\{v_7\}$, $H_c(v_{12}, 1)=\{v_8\}$, $H_c(v_{18}, 1)=\{v_{16}\}$ 和 $H_c(v_{19}, 1)=\{v_{17}\}$; 可发现 $H_c(v_9, 1)=\{v_5, v_6\}$ 和 $H_c(v_{10}, 1)=\{v_5, v_6\}$ 具有公共一阶前序节点集, 且符合合并规则, 则将 v_5, v_6, v_9, v_{10} 合并到一个复合任务 P_5 中(不失一般性, 选取复合任务中最小节点标签作为新复合任务的名称), 并将 P_5 推入队列 q (如图3(a), 其中虚线方框表示复合节点 P_5)。

对复合节点的一阶前序节点集 $H_c(P_5, 1)=\{v_1, v_2, v_3\}$ 进行检测, 设首先检测节点 v_4 , v_4 有两条输出数据流, 其中 $\pi(v_4, v_{17})=0.1 < \theta$; 此外, $eaup(P_5 \oplus v_4) > eaup(P_5 | P_5 \oplus v_4)$, 故将 v_4 合并到 P_5 中, 并可删除数据流 (v_4, v_{17}) , 此时 P_5 被

命名为 P_4 (选取复合任务中最小节点标签作为复合任务的名称), 结果如图3(b)所示。然后判断 P_4 和 v_{11} , 虽然 $\pi(v_{11}, v_7)=0.1 < \theta$, 但由于 $eaup(P_4 \oplus v_{11}) < eaup(P_4 | P_4 \oplus v_{11})$, 即 $P_4 \oplus v_{11}$ 期望支持度不是局部最优, 故 v_{11} 不可被合并到 P_4 。同样 v_2 和 v_3 也无法合并到 P_4 中。接下来, 对 v_{11} 进行检查, 由于其前序只有一个点 v_7 , 故两者可直接合并为 P_7 ; P_7 的一阶前序节点集为 $\{v_1, v_2, v_3\}$ 。由于节点 v_1, v_2, v_3 均有指向其他节点的数据流且不可删除, 故它们都不可合并到 P_7 中。将 P_7 推入队列 q 。对 q 中的下一个元素 v_{12} 进行处理。由于 v_{12} 的一阶前序只有 v_8 , 故可合并产生 P_8 ; 而 P_8 的一阶前序节点只有 v_3 , 可以合并产生 P_3 。同理, v_{18} 可与 v_{16} 合并产生 P_{16} , v_{17} 可与 v_{19} 合并产生 P_{17} ; 对于 P_{16} 而言, 其一阶前序节点集为 $\{v_{13}, v_{15}\}$, 而 v_{13} 有两条输出数据流, 其中一条指向 v_{15} , 且合并后的期望支持度较大, 故 P_{16} 可与 v_{13}, v_{15} 合并生成 P_{13} ; 同理 P_{13} 可与 P_{17}, v_{14} 合并生成新的复合任务 P_{13}' 。生成合理视图的整个过程如图3所示, 在图3(f)中, 每个虚线框构成视图中的一个复合任务点。

5 实验

本实验是在 Intel Core 2 CPU 6320 1.86 GHz 主频, 1 GB 内存的 HP 台式机上运行的。在实验评估中, 随机产生不确定数据源图, 并随机连接多个点为一个复合任务, 使得复合任务的数目达到数据源图中节点数目的 70%。在实验中, 对每个复合任务是否合理进行检测与构造, 并且对复合任务之间能否合并也进行检测与处理。

在图4所示的实验中, 数据源图中节点数 $|V_c|$ 从 50 增加到 550, 边数目 $|E_c|$ 设置为节点数的两倍, 即 $|E_c|=2|V_c|$ 。在阈值 θ 分别为 0.1、0.2、0.3 的情况下, 测试了算法所能生成的合理视图中节点个数(即复合任务的数目)及所需删除的边数。从图4中还可以看到, 设定的阈值越低, 删除边的数目越少, 而生成的复合任务数越多, 而且随着 $|V_c|$ 的增多, 这种趋势越明显。此外, 当所删除边的数目增多时, 相应的复合任务数将减少。这是因为某些不合理边的存在导致复合任务无法合并, 删除这些边之后将会使得更多的复合任务合并在一起, 从而减小了其总体数目。另一方面, 由于本算法通过计算复合任务的期望支持度来最终决定任务节点间的边能否删除, 这一特性使得原数据尽可能地完整, 避免了许多边被大量删除, 从而保证了原数据尽可能地完整。

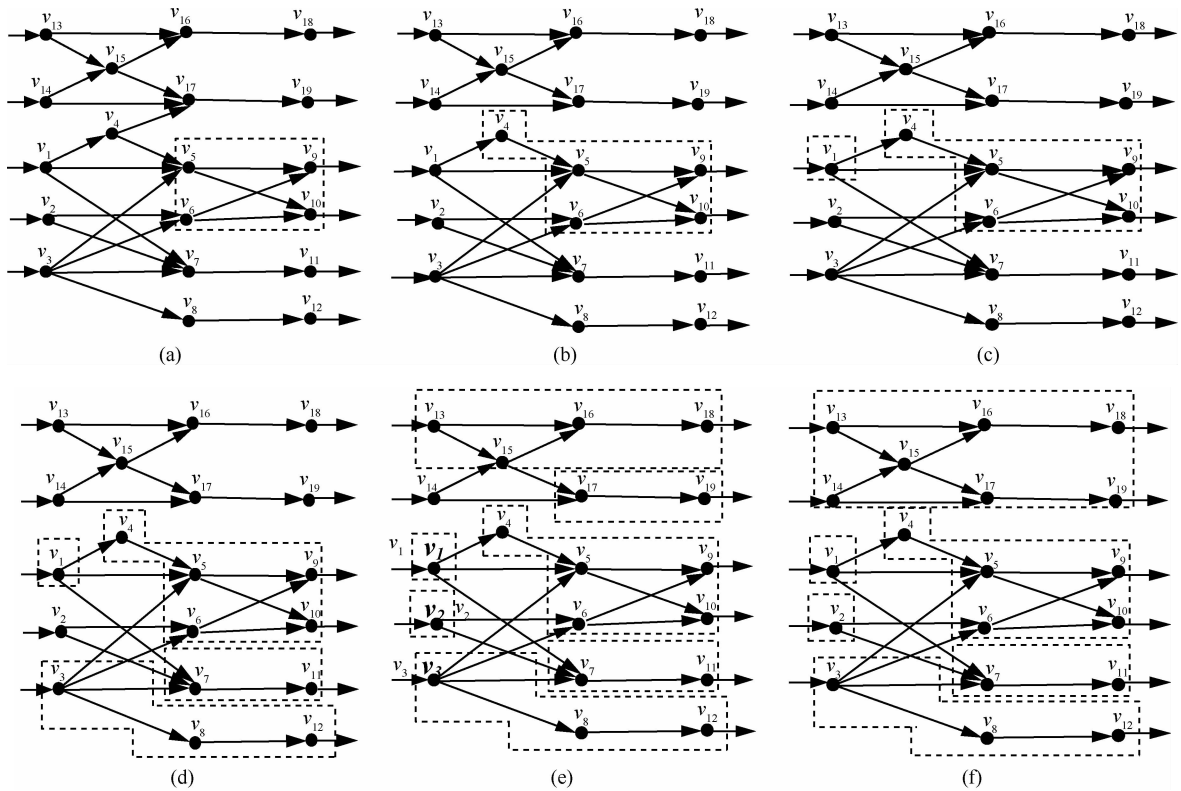
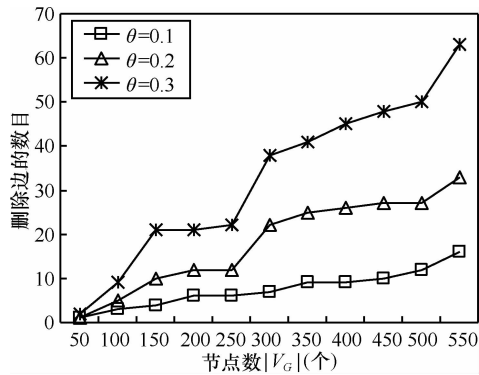
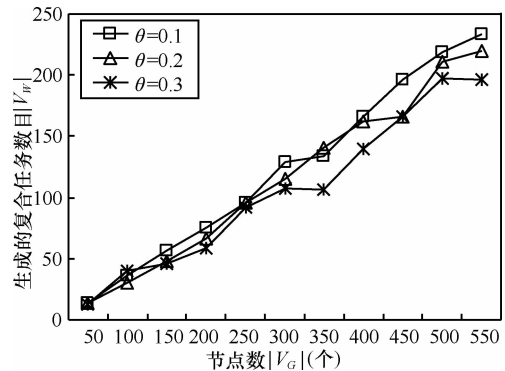


图3 合理视图的构造过程



(a) 需删除的边



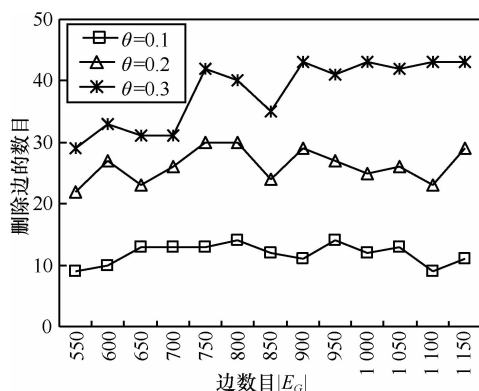
(b) 生成的复合任务数

图4 $|V_G|$ 对实验结果的影响

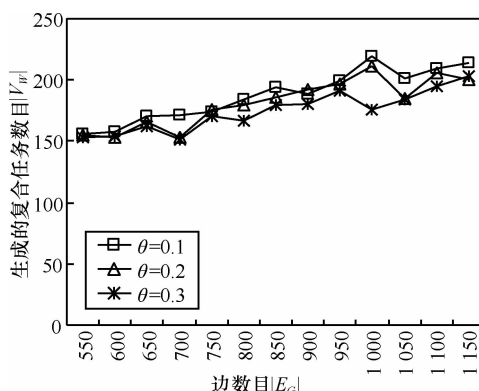
在图5所示的实验中,笔者将数据源图中节点数 $|V_G|$ 固定为500, $|E_G|$ 从550增加至1150。同样,在阈值 θ 分别为0.1、0.2、0.3的情况下,测试了CSV算法所能生成的复合任务的数目及所需删除的边数。由图5(a)可以看出在 $|V_G|$ 确定的情况下,所需删除边的数目随着 $|E_G|$ 增加变化不是很明显,这是因为 $|E_G|$ 的增加虽会导致每条边的概率相应减小,但被删除边的概率总和并不一定会减小,从而限制了大量边的删除。而从图5(b)中可以看出,在 $|V_G|$ 不变时,生成复合任务的数目随 $|E_G|$ 缓慢上升,这是因为当 $|E_G|$ 增多时,既

会造成某一个合理复合任务内部由于边的概率值下降而需要进行重构,也会使得多个复合任务可合并变成新的合理复合任务。

在图6所示的实验中,测试了 $|V_G|$ 和 $|E_G|$ 对构造合理视图所需时间的影响。在图6(a)中,将 $|V_G|$ 从50升值到550, $|E_G|$ 的值设置为 $2|V_G|$,并测试了在阈值 θ 分别为0.1、0.2、0.3的情况下,生成合理视图所需的时间。从图6(a)中可以观察到,当 $|V_G|$ 增加时,生成合理视图所需的时间也随着增加,而在不同的 θ 值情况下,同一个不确定图生成合理



(a) 需删除的边



(b) 生成的复合任务数

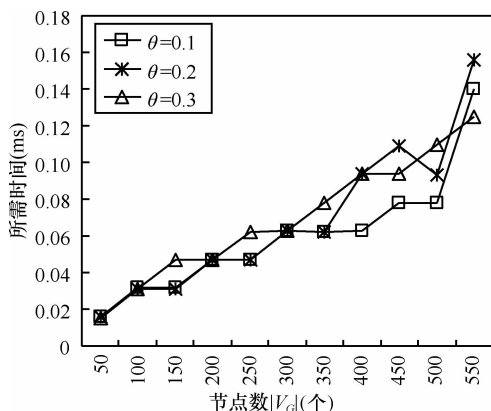
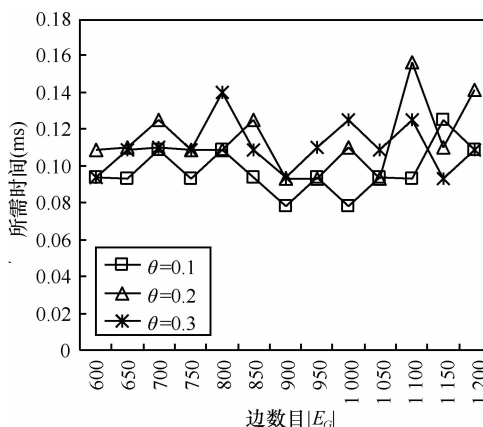
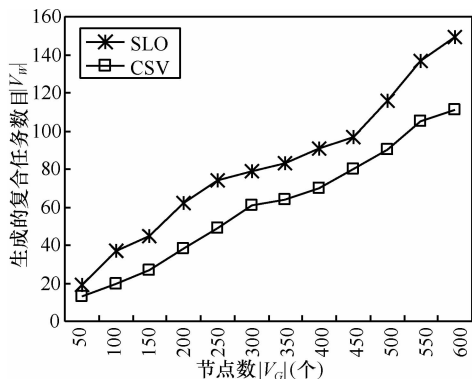
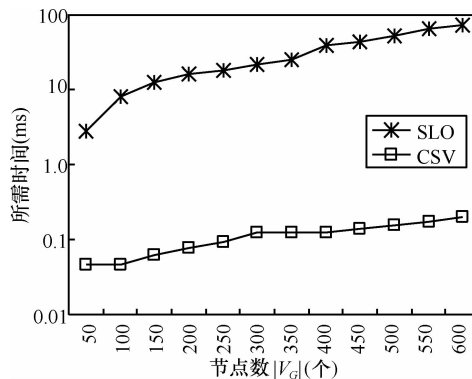
 图 5 $|E_G|$ 对实验的影响

 (a) $|V_G|$ 的影响

 (b) $|E_G|$ 的影响

图 6 算法运行所需的时间

视图所需的时间变化不大。在图 6 (b) 中, 将 $|V_G|$ 设置为 550, 让 $|E_G|$ 的值增加, 并观察在不同的阈值下 CSV 算法生成合理视图所需的时间。从图中可以看出, 当 $|V_G|$ 保持固定, 而 $|E_G|$ 值增加时, CSV 算法生成合理视图所需的时间基本保持不变, 这也符合 4.3 节中对 CSV 算法复杂度的分析结果, 即算法复杂度依赖于 $|V_G|$, 而并不会随着 $|E_G|$ 增加发生明显的变化。

在图 7 中, 将本文的 CSV 算法与参考文献[9]中的强最优(SLO)算法进行了对比实验, 测试了它们生成合理视图所需的执行时间及视图中 $|V_w|$ 的大小。在实验中, 将 $|V_G|$ 的值从 50 增加至 600, $|E_G|$ 的值设置为 $|E_G|=2|V_G|$, 并将所有边上的概率权值均设置为 1.0, 以满足算法 SLO 的需求。从图中可以看出在相同的工作流中, 本文的 CSV 算法生成复合任务数目要略低于 SLO, 尤其当 $|V_G|$ 值较大时更加


 (a) 对生成 $|V_G|$ 数目的影响


(b) 两种算法运行所需时间的比较

图 7 SLO 与 CSV 的对比实验



明显,这表明 CSV 算法更能有效地发现任务之间的数据流关系并加以合并,从而使得所生成的视图更加紧凑。另一方面,从图 7(b)中可以看出,CSV 算法生成合理视图所需的时间要明显低于算法 SLO,这时 SLO 的时间复杂度为 $O(n^3)^{[9]}$,而 CSV 算法的时间复杂度为 $O(n^2)$ 。

6 结束语

目前针对科学工作流视图构建技术方面的研究主要集中在确定式数据源图方面,而此方面进行研究,首先给出了在不确定数据源中检测不合理视图的方法,在此基础上综合考虑任务之间的多种数据流关系,给出了相应的检测和构建规则。笔者设计了相应的算法,从理论上分析了算法的总体时间复杂度,并通过相关实验分析了所提出方法的正确性及其有效性。今后的工作将集中在优化算法时间复杂度及提高视图整体的概率期望支持度等方面,而这些将有待进一步努力。

参考文献

- 1 IBM Blue Cloud Solution. <http://www-900.ibm.com/ibm/ideasfromibm/cn/cloud/solutions/index.shtm>
- 2 Sun Cloud Architecture Introduction White Paper. http://developers.sun.com.cn/blog/functionalca/resource/sun_353cloudcomputing_chinese.pdf
- 3 Barroso L A, Dean J, Holze U. Web search for a planet: the Google cluster architecture. *IEEE Micro*, 2003,23(2):22~28
- 4 International Telegraph Union (ITU). <http://www.itu.int/en/pages/default.aspx>
- 5 Organization for the Advancement of Structured Information Standards (OASIS). <http://www.oasis-open.org/>
- 6 冯登国,张敏,张妍等.云计算安全研究.软件学报,2011,22(1):71~83
- 7 郑湃,崔立真,王海洋等.云计算环境下面向数据密集型应用的数据布局策略与方法.计算机学报,2010,33(8):1472~1480
- 8 Oinn T, Addis M, Ferris J, *et al.* Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 2004, 20(17):3045~3054
- 9 Altintas I, Berkley C, Jaeger E, *et al.* Kepler: an extensible system for design and execution of scientific workflows. *Proceedings of the 16th International Conference on Scientific and Statistical Database Management*, Santorini Island, Greece, 2004:423~424
- 10 Biton O, Boulakia S C, Davidson S B, *et al.* Querying and managing provenance through user views in scientific workflows. *Proceedings of the 24th International Conference on Data Engineering*, Cancun, Mexico, 2008:1072~1081
- 11 Biton O, Davidson S B, Khanna S, *et al.* Optimizing user views for workflows. *Proceedings of the 12th International Conference on Database Theory*, St. Petersburg, Russia, 2009:310~323
- 12 Cohen S, Boulakia S C, Davidson S B. Towards a model of provenance and user views in scientific workflows. *Proceedings of Lecture Notes in Computer Science Data Integration in the Life Sciences*, Hinxton, UK, 2006: 264~279
- 13 Shao Q, Chen Y, Tao S, *et al.* EasyTicket: a ticket routing recommendation engine for enterprise problem resolution. *Proceedings of the VLDB Endowment*, Auckland, 2008: 1436~1439
- 14 Shao Q, Chen Y, Tao S, *et al.* Efficient ticket routing by resolution sequence mining. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, 2008:605~613
- 15 Shao Q, Sun P, Chen Y. Efficiently discovering critical workflows in scientific explorations. *Future Generation Computer System*, 2009, 25(5): 577~585
- 16 Sun P, Liu Z Y, Susan D, *et al.* Detecting and resolving unsound workflow views for correct provenance analysis. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Rhode Island, 2009:549~562
- 17 邹兆年,李建中,高宏等.从不确定图中挖掘频繁子图模式.软件学报,2009,20(11):2965~2976
- 18 Chui C K, Kao B, Hung E. Mining frequent itemsets from uncertain data. *Proceedings of the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, Nanjing, 2007: 47~58
- 19 Zhou S G, Yu Z C, Jiang H L. Concepts, issues, and advances of searching in graph structured data. *Communication*, 2007, 3(8): 59~65
- 20 Shasha D, Wang T L, Guigno R. Algorithmic and applications of tree and graph searching. *Proceedings of the 21st ACM SIGMOD-SIGART Symposium on Principles of Database Systems*, Madison, 2002:39~52
- 21 Yan X, Yu P S, Han J. Graph indexing: a frequent structure based approach. *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, Paris, 2001: 335~346
- 22 Hintsanen P, Toivonen H. Finding reliable subgraphs from large probabilistic graphs. *Data Mining and Knowledge Discovery*, 2008, 17(1):3~23
- 23 Cheng J, Yu J, Lin X. Fast computing reachability labelings for large graphs with high compression rate. *Proceedings of 11th*



- 5 Jharkharia S. Selection of logistics service provider: an analytic network process (ANP) approach. *Omega*, 2007, 35(3):274~289
- 6 Solesvik M Z. Partner selection for interfirm collaboration in ship design. *Industrial Management & Data Systems*, 2010, 11 (5): 701~710
- 7 张卫,潘晓弘,刘志等. 基于云模型蚁群优化的制造服务调度策略. *计算机集成制造*, 2012, 18(1):201~207
- 8 Wu C, Barnes D. Formulating partner selection criteria for agile supply chains: a dempster - shafer belief acceptability optimization approach. *International Journal of Production Economics*, 2010, 125(2):284~293
- 9 Hassan M M, Song B, Huh E N. A market-oriented dynamic collaborative cloud services platform. *Annals of Telecommunications*, 2010, 65(11-12):669~688
- 10 Cavusgil S T, Evirgen C. Use of expert systems in international marketing: an application for co-operative venture partner selection. *European Journal of Marketing*, 1997, 31(1):73~86
- 11 Mitsuhashi H, Greve H R. A matching theory of alliance formation and organizational success: complementary and compatibility. *Academy of Management Journal*, 2009, 52 (5): 975~995
- 12 黄彬,高诚辉,陈亮. 模糊完工时间和模糊交货期下的虚拟企业伙伴选择. *系统工程理论与实践*, 2010, 30(6):1085~109
- 13 杨虎,张东戈,刘浩等. 一种基于间接互惠的计算网格合作激

励机制研究. *电信科学*, 2011(9): 42~47

- 14 任声策, 宣国良. 基于学习和能力互补动态的研发联盟稳定性研究. *中国管理科学*, 2005, 13(5):111~115
- 15 张海青, 田军. IT 人员业务能力与技术能力的互补性对企业竞争优势的影响. *管理学报*, 2011, 8(9): 1359~1364
- 16 iiMedia Research: 2012 年 Q3 中国智能手机市场销量达到 4980 万部. <http://www.199it.com/archives/83266.html>

[作者简介]



据春华,男,博士,浙江工商大学计算机信息与工程学院、浙江工商大学现代商贸中心教授,浙江工商大学管理学院博士生导师,主要研究方向为智能信息管理、电子商务与物流的研究、决策支持系统等。



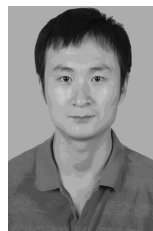
傅小康,女,浙江工商大学管理学院博士研究生、讲师,主要研究方向为决策支持系统、服务协同、供应链管理等。

(收稿日期:2013-02-23)

(上接第 100 页)

International Conference on Extending Database Technology, Nantes, 2008:193~204

- 24 Jin R, Hong H, Wang H X, *et al.* Computing label-constraint reachability in graph databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*. Indianapolis, Indiana, 2010:123~134



刘占晨,男,杭州电子科技大学硕士研究生,主要研究方向为图数据库、工作流技术。

[作者简介]



胡海洋,男,博士,杭州电子科技大学副教授、CCF 高级会员、CCF 系统软件专业委员会委员,主要研究方向为形式化方法、工作流技术。



胡华,男,博士,杭州电子科技大学教授、CCF 高级会员,主要研究方向为数据库技术、软件协同技术。

(收稿日期:2012-12-15)