

doi:10.3969/j.issn.1673-9833.2012.01.014

不完全标记的事务行为踪迹问题研究

欧阳星昱¹, 满君丰¹, 李长云¹, 彭成²

(1. 湖南工业大学 计算机与通信学院, 湖南 株洲 412007; 2. 中南大学 信息科学与工程学院, 湖南 长沙 410083)

摘要: 开放网络环境下分布式软件在交互行为中产生的事件行为踪迹标记丢失, 导致无法对软件行为进行分析和预测。为丢失的踪迹找到事件源, 将不完全标记的事务行为踪迹问题转化为网路最大流问题。采用沿路径推进的增载轨算法, 找到各事务产生的最有可能的行为踪迹序列。仿真实验表明: 本方法可以有效、准确地标记不完全标记的事务行为踪迹序列。

关键词: 不完全标记; 行为踪迹; 最大流; 增载轨算法

中图分类号: TP311

文献标志码: A

文章编号: 1673-9833(2012)01-0061-05

Research on Transaction Behavior Footprint with Incomplete Tokens

Ouyang Xingyu¹, Man Junfeng¹, Li Changyun¹, Peng Cheng²

(1. School of Computer and Communication, Hunan University of Technology, Zhuzhou Hunan 412007, China;

2. School of Information Science and Technology, Central South University, Changsha 410083, China)

Abstract: In open network environment, the loss of behavior footprint tokens producing in the interactions among the distributed software may lead unable to analyze and predict the software behavior. In order to find the event source for the lost token, translates the incomplete token problem of behavior footprint into the problem of network maxflow. Takes the along path forward of augmenting path algorithm to find the most possible behavior sequence for every transaction. The simulation experiment proves that the proposed method has validity and accuracy in marking the transaction of incomplete token.

Keywords: incomplete token; behavior footprint; maxflow; augmenting path algorithm

1 问题的提出

随着计算机技术的发展, 计算机软件已由静态封闭式发展成为开放动态、复杂多变的新分布式。人们对软件的可用性、可靠性、安全性要求越来越高, 新型分布式软件在国民经济中的地位越来越重要^[1]。如何准确地监测软件在交互实体中产生的事

务行为踪迹序列, 并对其进行有效诊断、分析和预测, 提高软件的可信性, 确保软件行为与预期相一致, 已成为软件领域的热点问题。研究不完全标记的事务行为踪迹问题, 在新型分布式软件环境下提高软件的可用性、安全性具有重要的理论意义和使用价值。

新型的分布式软件不同于集中式软件和传统的

收稿日期: 2011-12-01

基金项目: 国家技术创新基金资助项目(11C26214302856), 湖南省自然科学基金资助项目(11JJ4050), 湖南省教育厅科研基金资助项目(11B039, 11W002, 10C0620)

作者简介: 欧阳星昱(1983-), 男, 湖南衡阳人, 湖南工业大学硕士生, 主要研究方向为可信软件,

E-mail: yangxingyu1217@163.com

分布式软件：一是开放动态，二是松散聚合，三是行为复杂^[1]。传统的软件交互行为分析是基于内容的分析，在新型的分布式软件环境下，随着交互实体数量的不断增加，使分析的准确性和有效性难以保证。为解决上述的问题，一般情况下可采用 open-group ARM instrumentation 等工业标准，通过记录日志文件来跟踪事务踪迹，并根据日志文件的时间戳信息和事务实例的唯一标记符，确定一个实体在交互行为过程中产生的一条完整的行为踪迹序列。有些软件实体（特别是复杂多变、分布异构的软件实体）在行为交互过程中，产生的事件可能没有标记或者标记不可用，本文主要讨论开始状态有标记，其它状态部分有标记的情形。事务在软件交互过程中的状态转移实例如图 1 所示，图中用 3 种不同图案表示 3 个不同的事务。如果是单一事务，且事务交互过程中，各事件按时间顺序发生，遵循 FIFO（first-in-first-out）或 LIFO（last-in-first-out）原则，则很容易将各事件连接组成一个完整的行为踪迹^[2]。由于是多个事务并行操作，无法准确鉴别每个行为踪迹的唯一事件源，在分布式网络环境下多个事务并行处理使得每个事务的事件排序不规则，如果这些事件没有标记或者标记不可用，则无法通过日志文件的时间戳信息和事务实例的标记符，来确定每个踪迹的事件源。这对后续的行为分析和预测造成困难，如何对各事务行为踪迹进行准确无误的标记，是提高软件可信度的关键。

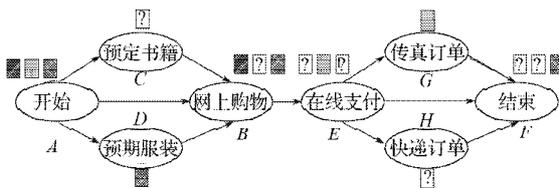


图1 不完全标记状态转移实例

Fig.1 The state translated instance for incomplete token

在事务行为踪迹标记研究方面，主要考虑对单个事务的行为踪迹进行标记。如 K. Chandy 等人^[3]对全局系统中单个事务确定无标记的状态进行监控与分析；M. Mansouri-Samani 等人^[4]在简单事务行为标记方面进行了研究，却没有针对在分布式网络环境下实现对不完全标记的事务行为踪迹标记研究。本文针对在分布式网络环境下多事务并发产生不完全标记的事务行为踪迹问题，提出不完全标记的事务行为踪迹模型，并通过网络最大流距离标号增载轨、容量约束算法^[5-6]，对事务行为踪迹进行标记，并以网上电子交易平台为例，通过 Matlab 进行仿真验证。

2 不完全标记的事务行为踪迹模型

2.1 相关概念

1) 令 $f(x)$ 是连续型随机变量 x 的概率密度函数，则： $f(x) \geq 0, \int_{-\infty}^{+\infty} f(x)dx = 1, P(a \leq x \leq b) = \int_a^b f(x)dx$ 。 $\bar{F}(x) = P\{X > x\}$ 是 $f(x)$ 的互补累积分布函数。

设 A 是一个 $n \times n$ 矩阵， a_{ij} 是 A 的第 i 行，第 j 列的元素。设 π 代表排列矢量 $(1, 2, \dots, n)$ ， $\log x$ 表示 x 的对数。

2) 对于一个有向无环图 $G(V, E)$ ，其中 V 是一个非空集合，称为顶点集； E 是 $V \times V$ 的子集，称为弧集。一个有向无环图，至少有一个没有前驱的源点（起点）和一个没有后继的汇点（终点）。如果 $\langle v_i, v_j \rangle$ 表示 $G(V, E)$ 的一条有向边， v_j 是 v_i 的直接后继， v_i 是 v_j 的直接前驱。 v_i 的直接后继的集合用 $N(v_i)$ 表示， v_j 的直接前驱的集合用 $P(v_j)$ 表示。

3) 单从跳转序列上（转移矩阵）看，马尔可夫链的每个状态的逗留时间是一个随机变量，而且会与跳转前和跳转后的状态有关。马尔可夫过程（Markov process, MP）与半马尔可夫过程（semi-Markov process, SMP）的区别是马尔可夫未来状态只依赖于上一个状态，半马尔可夫未来状态却依赖于后一个状态。定义 $S_i, i=0, 1, \dots, N$ 代表行为序列的第 i 个状态， T_{ij} 代表从状态 S_i 到状态 S_j 的转换时间。转移过的状态序列服从 MP，事件行为过程服从 SMP。转移概率矩阵用 $P=(p_{ij})$ 表示，每一个转换时间 T_{ij} 是一个仅依赖于状态 S_i 和 S_j 随机变量。每个事务行为过程转移符合 SMP。

2.2 事务行为踪迹系统模型

事务在状态转移过程中，只考虑有向无回路的简单情况，这样确保了事务转换都是单向转移，即事务在从上一个状态到下一个状态的转换过程中产生的踪迹都是唯一的。状态之间的转移时间服从独立同分布（independent and identically distributed, IID），IID 主要包括正态分布、均匀分布、指数分布，本文主要考虑转移时间服从正态分布，同时，状态之间的转移遵循马尔可夫性。在监测获取事务状态转移实例集合的基础上，分析事务在转换过程中产生的最大相似序列，并估计事务在转移过程中所消耗的时间，采用最大相似规则（maximum likelihood rule, MLR）正确地标记事务在转换过程中所产生的踪迹。由于每一个事务产生的踪迹是独一无二的，所以不同的事务不能共享相同的事务行为踪迹。对事务行为踪迹系统模型中的每个状态 S_k ，用排列矢量 π_k 表示状态 S_k 中各事件有效的匹配，用 T_k 表示状态

S_k 中事件行为踪迹记录的时间戳信息矢量。当事件交互行为踪迹按正确时间次序到达, \mathbf{T}_k 按升序排列, 越早发生的事件排列越后。根据排列矢量 $\boldsymbol{\pi}_k, \mathbf{T}_k^{\pi_k}$ 表示 \mathbf{T}_k 根据中各事件的时间戳信息排列矢量。记开始状态为 \mathbf{T}_0 , \mathbf{T}_0 表示开始状态的时间戳矢量且各事务的标记完全, 开始状态的排列矢量被设置为 $\boldsymbol{\pi}_0$, 即从状态 S_0 开始发现与 S_0 的相关联的所有事务行为踪迹。

在事务转换时间的概率密度函数已知时, 将定量跟踪问题转化为所有事务踪迹正确标记的概率问题, MLR 最大, 事务行为的有效标记最大。对于 $N+1$ 个状态, 事务的最大相似踪迹可以转化为 N 个排列矢量 $\boldsymbol{\pi}_k$ 的集合

$$\{\boldsymbol{\pi}_1^{ML}, \boldsymbol{\pi}_2^{ML}, \dots, \boldsymbol{\pi}_N^{ML}\} = \arg \max_{\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_N} P(\mathbf{T}_1^{\boldsymbol{\pi}_1}, \mathbf{T}_2^{\boldsymbol{\pi}_2}, \dots, \mathbf{T}_N^{\boldsymbol{\pi}_N} | \mathbf{T}_0). \quad (1)$$

3 两状态模型

式 (1) 是一个 NP 难问题, 为简化式 (1) 的难度, 需将多状态系统简化为两状态系统。给定 \mathbf{T}_0 和 \mathbf{T}_1 分别表示各事务在开始状态 S_0 向结束状态 S_1 转移的过程中, 产生事务行为踪迹的时间戳矢量, 式 (1) 可简化为

$$\hat{\boldsymbol{\pi}}^{ML}(\mathbf{T}_0, \mathbf{T}_1; f) = \arg \max_{\boldsymbol{\pi}} P(\mathbf{T}_1^{\boldsymbol{\pi}} | \mathbf{T}_0). \quad (2)$$

图 2 表示两状态系统转移模型, 状态 S_0 和 S_1 产生的事务行为踪迹 \mathbf{T}_0 和 \mathbf{T}_1 通过一个未知的排列矢量 $\boldsymbol{\pi}$ 相联系, 并根据给定的 \mathbf{T}_0 和 \mathbf{T}_1 时间戳信息发现排列矢量 $\boldsymbol{\pi}$, 从而得

$$\mathbf{T}_0(j) = \mathbf{T}_1(\boldsymbol{\pi}(j)) - t(j), \quad 1 \leq j \leq |\mathbf{T}_1|, \quad (3)$$

式中: $\mathbf{T}_0(j)$ 表示第 j 个事务在状态 S_0 产生的行为踪迹的时间戳信息;

$t(j)$ 表示状态 S_0 中第 j 个事务踪迹转向状态 S_1 所需的转换时间;

$\mathbf{T}_1(\boldsymbol{\pi}(j))$ 是状态 S_1 中的第 j 个元素, 即 $\mathbf{T}_0(j)$ 和 $\mathbf{T}_1(\boldsymbol{\pi}(j))$ 来自同一个事务。

在实际情况中, 总有 $|\mathbf{T}_0| \geq |\mathbf{T}_1|$, 因为有些事务可能依然驻留在状态 S_0 。

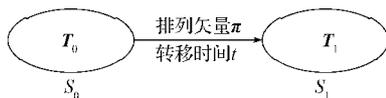


图2 两状态系统转移模型

Fig. 2 The translated model of two states system

因此, 状态 S_0 中所有的事务在行为交互过程中产生的事件行为踪迹通过最大相似规则, 全部踪迹准确标记的最大概率为

$$P^{ML}(\mathbf{T}_0, \mathbf{T}_1; f) = \frac{f_T(\hat{\boldsymbol{\pi}}^{ML} - \mathbf{T}_0)}{\sum_{\boldsymbol{\pi}} f_T(\mathbf{T}_1^{\boldsymbol{\pi}} - \mathbf{T}_0)}. \quad (4)$$

最大相似排列 $\hat{\boldsymbol{\pi}}^{ML}$ 的相似度用所有有效排列的相似度之和来规范化。至少有一个排列有正的相似度, 所以式 (4) 的值严格为正。对两状态系统, 所有事务在行为交互过程中转移时间服从 IID, 通过最大相似规则 MLR, 式 (2) 可简化为

$$\hat{\boldsymbol{\pi}}^{ML}(\mathbf{T}_0, \mathbf{T}_1; f) = \arg \max_{\boldsymbol{\pi}} \left\{ \prod_{1 \leq \pi(i) \leq k} f_T[\mathbf{T}_1(\boldsymbol{\pi}(i)) - \mathbf{T}_0(i)] \cdot \prod_{k < \pi(i) \leq n} \bar{F}_T[\mathbf{T}_1(\boldsymbol{\pi}(k)) - \mathbf{T}_0(i)] \right\}. \quad (5)$$

由于事务在状态转移过程中有些事务可能依然驻留在之前的状态中, 这样将 $\boldsymbol{\pi}(i)$ 划分为 2 个区间: $1 \leq \boldsymbol{\pi}(i) \leq k$, 表示状态 S_0 中第 i 个事务正常转移到 S_1 状态; $k < \boldsymbol{\pi}(i) \leq n$, 表示第 i 个事务仍然驻留在 S_0 状态。对于没有转移的事务, 假设是按照时间顺序到达, 这样两状态系统模型可以转化为一个有向二部图问题模型, 各事务产生的事件行为踪迹构成有向二部图 $G(V_0 \cup V_1, E)$, 并在 V_1 中补充 $n-k$ 个等同节点, 用集合 $V_1(\delta)$ 表示, δ 表示等同节点。对于边 $(i, j) \in E, i \in V_0, j \in V_1$, 其权值

$$W(i, j) = \begin{cases} -\log f_T[\mathbf{T}_1(j) - \mathbf{T}_0(i)], & j \leq k; \\ -\log \bar{F}_T[\mathbf{T}_1(k) - \mathbf{T}_0(i)], & j > k. \end{cases} \quad (6)$$

即等同节点 $V_1(\delta)$ 与 $V_0(i)$ 之间的边的权值采用式 (6) 来计算。

4 SMP 优化系统模型

在实际的新型分布式网络环境下, 往往涉及更多的操作状态, 不仅仅是两状态系统模型中考虑的开始状态 S_0 和结束状态 S_1 , 如图 1 就是一个多状态模型实例。由于各事务在各状态之间的转移时间满足 SMP, 每个事务在转换过程中相互独立, 也就是状态之间的转移时间只依赖当前状态和未来状态。据此, 将多状态系统模型转化为两状态模型, 并将其分割成两个部分: 前驱节点集和后继节点集。于是, 式 (1) 可化为

$$P(\mathbf{T}_1^{\boldsymbol{\pi}_1}, \mathbf{T}_2^{\boldsymbol{\pi}_2}, \dots, \mathbf{T}_N^{\boldsymbol{\pi}_N} | \mathbf{T}_0) = \prod_{m>0} P \left(\bigcup_{S_k \in B_m} \mathbf{T}_k^{\boldsymbol{\pi}_k} \mid \bigcup_{S_l \in P(B_m)} \mathbf{T}_l^{\boldsymbol{\pi}_l} \right), \quad (7)$$

式中 B_m 为状态划分子集, $m \geq 1$ 且 $B_1 = S_0$, 每一个状态划分中包含一个或者多个状态, 规定不同分割中的状态不能共享同一个前驱。

由状态划分 B_m 的定义可知, 同一状态不能在不

同的状态划分子集中。对图3所示的事务转移模型，用文献[7]提供的状态划分算法可得

$$B_1=\{A\}, B_2=\{B,C,D\}, B_3=\{E\}, B_4=\{G,F,H\},$$

4个状态子集。再根据式(6)计算出每一条边 (i,j) 的权值。

5 标记算法分析

本文讨论的是开始状态有标记，其它状态部分有标记的情形。图3所示多状态转移模型，转移时间满足SMP，为了对丢失了标记的事务行为踪迹进行标记，采用增载轨算法。制约增载轨算法性能的主要因素^[6]是，增载轨的数量和寻找增载轨所花费的代价。有2种方法可用来减少增载轨的数量，一是每次选择当前容量最大的增载轨；二是每次选择最短路增载轨。

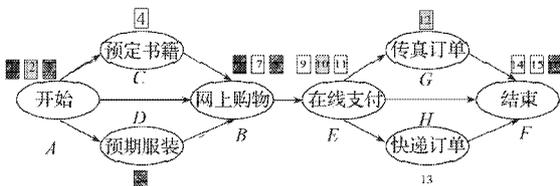


图3 不完全标记的多状态转移模型

Fig. 3 Multi-states translated model with incomplete token

图3中有3个并发事务，且事务在转移过程中开始有标记，中间状态部分有标记。将其转化为一个多源多汇的网络最大流问题，为其添加2个虚拟节点，源点 s 和汇点 t ，再将其转换为单源单汇网络流模型，见图4。

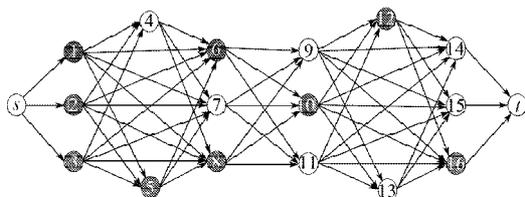


图4 状态转移的单源单汇网络流模型

Fig. 4 Multi-states translated directed net-flowing model

为了准确标记各事务在状态转移时的唯一踪迹，根据模型构建有向无环网络 $G(V,E)$ 。通过事务转移的状态划分算法，将整个多状态模型划分为多个状态，每个状态之间具有相互独立性且转换符合SMP。记开始状态为 B_0 ，结束状态为 B_n ， $n=size(\{B_m\})+1$ 。 $\{B_m\}$ 表示状态划分产生的状态划分集合。 V 表示各状态中不同事务， E 中边的权重由式(6)确定，并对各节点进行反向距离标号，记为 $d(B_k,i,M)$ ，其中 B_k 表示第 k 个状态划分， $k \in \{0,1,\dots,n\}$ ， i 表示 G 中第 i 个节点， M 表示不同事务的标记信息。在最短路增载轨算法中规定下一个节点的距离严格小于上一个节点的距离，这样增加了距离重标号的次数。为解决

此问题，放宽限制条件，允许下一个节点跟上一个节点有相同的距离标号，使其满足式(8)：

$$\begin{cases} d(B_k,t,M)=0; \\ d(B_k,i,M) \leq d(\{B_k,B_{k+1}\},j,M)+1, \\ \forall (i,j) \in V, i \in B_k, j \in \{B_k,B_{k+1}\}. \end{cases} \quad (8)$$

$N(i)$ 表示 i 出发到 $\{B_k, B_{k+1}\}$ 所有的可行流节点集合，有 $j \in N(i)$ 。从节点 i 到节点 j 中寻找可行流时作如下限制：

- 1) $d(B_k,i,M) \leq d(\{B_k,B_{k+1}\},j,M)+1$;
- 2) 在 $d(B_k,i,M)$ 和 $d(\{B_k,B_{k+1}\},j,M)$ 中，存在 $j \in N(i)$ 有相同的标记，则表示节点 j 为 i 匹配的下一个节点；
- 3) 如果2)不满足，则从节点 i 到 $N(i)$ 中寻找容量最大的节点 j ，并记 j 中的 M 为 i 中的标记，表示节点 j 为 i 匹配的下一个节点，使其在后面的步骤中不再被检测。

算法描述如下：

Begin

根据状态划分算法，将模型划分为一个状态划分集合 $\{B_m\}$ ，并设开始状态为 B_0 ，结束状态为 B_n ， $n=size(\{B_m\})+1$ ， B_1 状态下节点全部有标记。

通过距离反向标号为一个状态进行标号。

If 节点处有标记 $\rightarrow M$ 为当前标记；

else $\rightarrow M=null$;

$d(B_k,i,M)=k, k \in \{0,1,\dots,n\}$;

$i \in next(s)$;

begin

while($i \neq t$) {

if 在 $next(i)$ 中存在一个节点 j 与节点 i 标记相同，
 $i=j$;

else 在剩余网络中找寻 $N(i)$ 中容量最大的节点 j ，
对节点 j 进行标号；

$i=j$;

if($i=t$)

得到一条增载轨路径，删除该路径上的所有结点及对应的边；

重新划分状态、距离标号；

$i \in N(s)$;

}

end;

End

6 系统仿真实现

基于笔者自行设计的网上电子交易平台进行仿

真实验,通过对用户的交互行为进行监控,将交互行为信息转化为试验参数,最后通过 Matlab 实现上述标记算法,对不完全标记的事务行为进行标记。根据记录的日志文档可提取与行为踪迹对应的时间戳和事务踪迹的标记信息,如表 1 所示。

表 1 行为踪迹、时间戳和标记对应表

Table 1 The table of footprints, timestamps and marks

踪迹	1	2	3	4	5	6	7	8
时间戳	0.85	0.95	1.15	1.35	1.55	1.25	1.55	1.65
标记	b	g	r	?	r	b	?	r
踪迹	9	10	11	12	13	14	15	16
时间戳	1.95	2.05	2.15	2.25	2.35	2.45	2.15	2.55
标记	?	g	?	g	?	?	?	r

注: b 代表事务 1; g 代表事务 2; r 代表事务 3; ? 表示标记不可用。

假定事务在各状态之间的转移时间 t 服从正态分布且不考虑等同节点的情形,即各状态没有驻留的踪迹。每个事务在转移过程中,事务产生的踪迹具有时间先后顺序,如果行为踪迹序列的时间戳之差为负,则表明此路径不存在。从表 1 可知,踪迹 6 先于踪迹 4 和踪迹 5 产生。因此,状态 B 产生的踪迹 6 不可能由状态 C 或状态 D 下的事务产生;同理,状态 F 产生的踪迹 15 也不可能由状态 G 或状态 H 下的事务产生。踪迹 3,5,8,16 具有相同的标记,表明他们由同一踪迹产生;同理,踪迹 1,6 由同一踪迹产生;踪迹 2,10,12 由同一踪迹产生。利用表 1 提供的时间戳数据,可将其转换为图 4 网络流模型,代入式 (6) 可计算出每一条边的权值,通过本文提供的算法能准确有效地对没有标号的事务行为踪迹进行标记,结果见图 5。

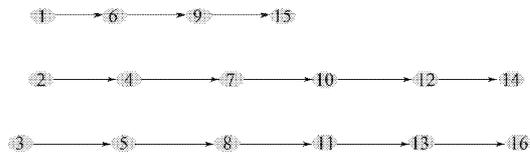


图 5 不完全标记的事务行为踪迹标记结果

Fig. 5 The mark result of footprint with incomplete token

由图 5 可知,3 个不同事务可得到 3 条完整的行为踪迹序列: 1-6-9-15, 2-4-7-10-12-14, 3-5-8-11-13-16。通过反复监控获取实验参数,并仿真实验表明,对不完全标记的事务行为踪迹标记是有效、准确的。

图 6 为事务踪迹标记的异常比例。由图可知,通过增加开始状态的节点数,测试和观察事务行为踪迹的标记结果,该算法在随着开始状态节点数的增加,总体标记异常比例约为 12%。这种算法在保证事务踪迹标记准确的前提下,提高了后续行为踪迹分

析与预测的有效性和准确性,因此,对不完全标记的事务行为踪迹分析具有较高的实用价值。

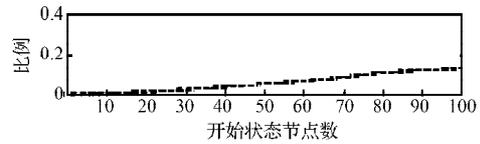


图 6 事务踪迹标记异常比例

Fig. 6 The mark exceptional rate of footprints

7 结语

本文研究了在新型分布式软件环境下不完全标记的事务行为踪迹问题,讨论了开始状态有标记,中间状态部分有标记的情形。将事务转移状态划分为多个相互独立的子状态集合,通过采用最短路增载轨算法,对其进行距离标号、容量约束,为不完全标记的事务行为踪迹进行标记处理,仿真结果表明该方法是准确有效的。

参考文献:

- [1] Li C Y, Man J F, Wang Z B, et al. Research on Interactive Behavior Analyzing in New-Type Distributed Software System[C]// Proceedings of the Automation and Control. San Antonio: TSI Press, 2010: 513-517.
- [2] 满君丰,李长云,文志诚,等.不完全标记的多个并行事务踪迹的“剥离”[J].软件学报,2010,21(增刊1): 224-237.
Man Junfeng, Li Changyun, Wen Zhicheng, et al. Stripping Multiple Parallel Transaction Footprints with Incomplete Tokens[J]. Journal of Software, 2010, 21(S1): 224-237.
- [3] Chandy K M, Lamport L. Distributed Snapshots: Determining Global States of Distributed Systems[J]. ACM Transactions on Computer Systems, 1985, 3(1): 3-7.
- [4] Mansouri-Samani M, Sloman M. Monitoring Distributed Systems[J]. Network, IEEE, 1993, 7(6): 20-30.
- [5] Ahuja R K, Magnanti T L, Borlin J. Network Flows: Theory, Algorithms and Application[M]. New Jersey: Prentice-Hall, 1993: 58-62.
- [6] 张宪超,江贺.一个新的最大流问题增载轨算法[J].小型微型计算机系统,2006,27(9): 1726-1730.
Zhang Xianchao, Jiang He. New Augmenting Path Algorithm for the Maximum Flow Problem[J]. Journal of Chinese Computer Systems, 2006, 27(9): 1726-1730.
- [7] Monnot J, Toulouse S. The Path Partition Problem and Related Problems in Bipartite Graphs[J]. Operations Research Letters, 2007, 35(5): 677-684.

(责任编辑:邓光辉)