

采用 Petri 网的业务性能分析方法

盖炳帅, 王劲林, 刘学

(中国科学院声学研究所国家网络新媒体工程技术研究中心, 100190, 北京)

摘要: 在业务管理系统中, 针对业务性能分析过程缺乏自动化的问题, 提出一种采用 Petri 网的业务性能分析方法. 该方法通过改进传统 Petri 网模型, 提出业务性能分析 Petri 网模型; 基于该模型设计业务性能计算公式进行自动业务性能计算; 采用特定样本训练业务性能分析 Petri 网模型参数, 提高模型参数的准确性. 实例分析结果表明, 该方法能够建立有效的业务性能分析模型, 实现业务性能自动计算和业务性能瓶颈自动发现, 且能够通过样本训练提高模型参数的准确性.

关键词: 业务管理; Petri 网; 业务性能分析; 样本训练

中图分类号: TP393 **文献标志码:** A **文章编号:** 0253-987X(2012)12-0012-06

An Analysis Method of Service Performance Based on Petri Net

GAI Bingshuai, WANG Jinlin, LIU Xue

(National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: An analysis method of service performance is proposed based on Petri net and is used as an automatic method for the service performance analysis process. A service performance analysis model is presented to improve traditional Petri net model, and formulas to automatically calculate service performance are given in the model. Special sample training is used to improve the accuracy of model parameters. Analysis on examples shows that the proposed method can effectively build service performance analysis models for automatic service performance calculating, automatically find service performance bottleneck, and improve the accuracy of model parameters through sample training.

Keywords: service management; Petri net; service performance analysis; sample training

业务管理系统通过业务性能分析发现业务性能瓶颈, 进而实施相关业务配置来消除业务性能瓶颈, 提高业务性能. 业务性能分析是业务管理系统的基础, 为业务配置提供配置依据, 在业务管理过程中起着至关重要的作用. 国家“十一五”科技支撑计划重点项目中的“支持跨区域、多运营商的新一代综合业务网络管理系统”, 管理不同运营商的数千个业务. 在业务量日益增多的情况下, 为实现高效的业务管理, 降低业务管理成本, 业务管理系统需提高自动化管理的水平, 这就要求有效的业务模型来实现业务性能自动分析过程及业务性能瓶颈自动发现过程.

目前, 主要从 Web 用户体验角度或企业业务重组角度为业务建模, Web 用户体验角度的业务建模为业务发现和业务选择提供参考数据, 而企业业务重组角度的业务建模为业务重组时的业务组件选择提供参考数据. 在业务建模领域中, 缺乏一种有效的业务模型用于自动化的业务性能分析过程. 如何建立有效的业务性能分析模型, 实现自动化的业务性能分析过程, 是当前业务管理领域亟待解决的问题.

当前, 在国内外有多项针对业务性能分析的研究. Santos 等采用 Mashup 设计模式^[1] 糅合 Key-stroke-Level 模型^[2] 和复杂模型^[3], 分析 IT 业务过

程中导致业务性能下降的人力因素^[4]. Li 等^[5]在无线网状网络中,基于用户位置移动队列模型分别提出了静态队列模型的用户移动管理机制和动态队列模型的用户移动管理机制,并采用 Petri 网量化分析两种用户移动管理机制的性能. Zeng 等^[6]采用 AHP(Analytic Hierarchy Process)方法从业务层、应用层、IT 基础设施层建模业务性能指标,并通过业务性能指标运算分析业务性能. AHP 方法能够按照多种业务性能指标分析业务性能,但是没有提出相关方法建模影响业务性能的因素,从而不能自动分析导致业务性能下降的原因,即不能自动发现业务性能瓶颈.

本文作者曾提出一种基于本体的业务建模方法^[7],该业务模型能够很好地从多个方面建模影响业务性能的因素. 基于以上分析,在业务管理系统中,本文提出一种采用 Petri 网^[8]的业务性能分析方法. 该方法基于文献[7]提出的业务模型改进传统 Petri 网模型,提出业务性能分析 Petri 网模型,然后基于该模型设计业务性能计算公式,进行自动业务性能计算,最后,采用特定样本训练业务性能分析 Petri 网模型参数,提高模型参数的准确性. 实例分析结果表明,该方法能够建立有效的业务性能分析模型,实现业务性能自动计算和业务性能瓶颈自动发现,且能够通过样本训练提高模型参数的准确性.

1 业务性能分析 Petri 网模型

业务是由一系列业务活动经串行连接或并行连接组成的,而业务重组通常不改变业务活动内部,只改变业务活动的数量及业务活动的顺序关系等. 业务性能是其业务活动性能的总体表现. Petri 网以研究系统的组织结构和动态行为为目标,着眼于系统中可能发生的各种变化以及变化之间的关系,描述系统中事件之间的各种依赖关系,例如顺序、并发等. 因此, Petri 网适合于建模业务组织结构关系和业务动态变化行为. 本文基于 Petri 网为业务性能分析建模.

1.1 传统 Petri 网模型

传统 Petri 网的结构元素主要包括库所(Place)、变迁(Transition)和有向弧(Arc). 库所用于描述可能的系统局部状态;变迁用于描述系统状态的变化传递关系;有向弧是连接库所和变迁的有向线段,它描述的是系统状态变化的方向. 传统 Petri 网模型用一个六元组定义, $G=(S, T, F, K, W, M_0)$, G 表示传统 Petri 网模型,其元素定义如下:

$N=(S, T, F)$, 表示传统 Petri 网模型的基网, S 为网 N 的库所集, T 为网 N 的变迁集, F 为网 N 的有向弧集,在基网中,采用圆圈表示库所,采用竖线表示变迁,满足 $S \cap T = \emptyset$, 且 $F = (S \times T) \cup (T \times S)$, $\text{Domain}(F) \cup \text{Range}(F) = S \cup T$, 其中 $\text{Domain}(F)$ 表示 F 的定义域, $\text{Range}(F)$ 表示 F 的值域,如图 1 所示; K 表示库所容量函数; W 表示有向弧 F 的权函数; M_0 表示初始状态,满足 $\forall s \in S$, 都有 $M_0(s) \leq K(s)$.

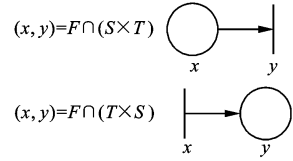


图 1 有向弧 F 图

以 $\cdot t$ 表示变迁 t 前置库所的集合,以 $t \cdot$ 表示变迁 t 后置库所的集合,在传统 Petri 网中,变迁点火后,变迁前置库所中的库所数减少 $W(\cdot t, t)$ 个,变迁后置库所中的库所数增多 $W(t, t \cdot)$ 个. 一个变迁 $t \in T$ 在状态 M 下有发生权的条件是(点火规则): $\forall s \in \cdot t, \forall s' \in t \cdot$ 满足 $M(s) \geq W(s, t) \wedge M(s') + W(t, s') \leq K(s')$.

1.2 业务性能分析 Petri 网模型

文献[7]提出了一种基于本体的业务建模方法. 该方法采用 OWL(Ontology Web Language)^[9] 本体描述业务活动对象、业务所需资源、业务数据等,采用本体活动描述业务活动建模业务过程,从多个方面建模影响业务性能的因素,并通过 SWRL(Semantic Web Rule Language)^[10] 约束这些因素的值域范围. 本文基于文献[7]提出的业务模型,改进传统 Petri 网模型,提出业务性能分析 Petri 网模型,用于自动业务性能分析和自动业务性能瓶颈发现.

业务性能分析 Petri 网模型用一个十元组定义, $H=(S, T, F, K_{\min}, K_{\max}, \lambda, W, M_0, U, T_L)$, H 表示业务性能分析 Petri 网模型,其元素定义如下.

(1) $N=(S, T, F)$, 表示业务性能分析 Petri 网模型的基网, $S=S_c \cup S_a$, S_c 表示影响业务活动性能的因素所对应的库所的集合,是业务性能分析 Petri 网的初始库所; S_a 表示业务活动所对应的库所的集合,是业务性能分析 Petri 网的中间库所.

(2) $K_{\min}, K_{\max}: K_{\min}, K_{\max}$ 是 S 到非负数 R_0 的映射, $K_{\min}: S \rightarrow R_0$, 表示库所容量的下限; $K_{\max}: S \rightarrow R_0$, 表示库所容量的上限. 满足 $K_{\min}(s) \geq 0, K_{\max}(s) \leq \infty, s \in S_c; K_{\min}(s) = 0, K_{\max}(s) = 1, s \in S_a$.

(3) λ : 是 S 到区间 $(0, 1)$ 的映射, $\lambda: S \rightarrow (0, 1)$, 表示库所的点火阈值系数, 满足 $0 < \lambda(s) < 1, s \in S_c$; $\lambda(s) = 1, s \in S_a$.

(4) W : 是 F 到区间 $(0, 1)$ 的映射, $W: F \rightarrow (0, 1)$, 表示有向弧的权函数. 满足 $\sum_{\text{Range}(f)=t_0} W(f) = 1$, 变迁 t_0 前置库所对应的有向弧的权值之和等于 1.

(5) M_0 : 是 S 到非负数 R_0 的映射, $M_0: S \rightarrow R_0$, 表示业务性能分析 Petri 网的初始状态, 满足 $K_{\min}(s) < \lambda(s) K_{\max}(s) < M_0(s) \leq K_{\max}(s), s \in S_c$; $M_0(s) = 1, s \in S_a$.

(6) U : 是 T 到区间 $(0, 1)$ 的映射, $U: T \rightarrow (0, 1)$, 表示变迁的权函数, 满足 $\sum_{t'=s_0} U(t) > 0$.

(7) T_L : 是变迁时间标签集合, 初始时 $T_L = \emptyset$, 以 $T_j(n)$ 表示变迁 t_j 在 n 时刻点火, 则变迁 t_j 点火后, $T_L = T_L \cup \{T_j(n)\}$. 变迁时间标签集合的引入有着特殊的意义, 它能够标识业务性能下降的传播路径, 同时可以防止变迁的反复点火.

在业务性能分析 Petri 网中, 将文献[7]提出的业务模型中采用 OWL 描述的影响业务活动性能的因素放在初始库所中, 将采用 OWL 描述的业务活动放在中间库所中, 并将业务因素对业务活动性能的影响通过变迁表示. 经过简单的线性变换, 可以将 SWRL 约束的业务因素的值域范围变换到区间 $(K_{\min}, K_{\max})$ 中, 库所状态值 $M(s)$ 越接近 $K_{\max}(s)$ 表示业务因素状态越好; 库所状态值 $M(s)$ 越接近 $K_{\min}(s)$ 表示业务因素状态越差. 在业务性能分析 Petri 网的变迁点火过程中, 任何一个状态 M 都满足条件 $K_{\min}(s) \leq M(s) \leq K_{\max}(s)$.

定义点火规则: 在业务性能分析 Petri 网中, 当变迁前置库所的业务因素或业务活动的状态下降, 其值小于等于库所点火阈值系数与库所容量上限的乘积时, 变迁发生点火, 即一个变迁 $t \in T$ 在状态 M 下有发生权的条件是 $\exists s \in {}^*t$, 满足 $M(s) \leq \lambda(s) \cdot K_{\max}(s)$.

可以看出, 在业务性能分析 Petri 网中, 当 ${}^*t \subseteq S_a$ 时, 变迁 t 肯定会发生点火, 一旦初始库所 S_c 对应的变迁发生点火后, 中间库所 S_a 对应的变迁将按照顺序点火下去, 直到变迁到达输出库所. 在业务性能分析 Petri 网中, 变迁点火后, 变迁前置库所的状态保持不变, 变迁后置库所的状态下降, 变迁 t 点火后库所的状态变化公式如下

$$M'(s) =$$

$$\begin{cases} M(s) - \sum_{p \in {}^*t} \frac{\lambda(p) K_{\max}(p) - M(p)}{\lambda(p) K_{\max}(p) - K_{\min}(p)} \cdot \\ \quad W(p, t) U(t), & s \in {}^*t \\ M(s), & s \notin {}^*t \end{cases} \quad (1)$$

在业务性能分析 Petri 网中, 变迁前置库所中的业务因素或业务活动的状态值满足点火规则时, 会导致变迁后置库所中业务活动性能下降. 考虑到不同业务因素对业务活动的影响程度不同, 及不同业务活动对其后续业务活动的影响程度不同, 采用权函数 W 表示变迁前置库所对变迁后置库所影响的权值. 因属于同一变迁的多个前置库所从同一方面制约变迁后置库所中业务活动的性能, 故权函数 W 满足 $\sum_{\text{Range}(f)=t_0} W(f) = 1$, $W(f)$ 越大, 该有向弧所对应的库所对变迁后置库所的影响越大.

在业务性能分析 Petri 网中, 库所中业务活动性能的下降可能由多个变迁引起, 即多个变迁可能有相同后置库所. 考虑到不同变迁对其后置库所中业务活动性能的影响程度不同, 采用权函数 U 表示变迁对其后置库所中业务活动性能影响的权值. 因具有相同后置库所的多个变迁从不同方面制约变迁后置库所业务活动的性能, 故权函数 U 满足 $\sum_{t'=s_0} U(t) > 0$, $\sum_{t'=s_0} U(t)$ 可能大于 1, 也可能小于 1, $U(t)$ 越大, 变迁 t 对其后置库所 s_0 中业务活动性能的影响越大.

在业务性能分析 Petri 网中, 变迁时间标签集合 T_L 能够记录变迁的点火路径, 变迁的点火路径能够反映业务性能的下降路径, 通过分析变迁时间标签集合 T_L , 能够找到导致业务性能下降最多的业务活动和导致该业务活动性能下降最多的业务因素, 找到业务性能瓶颈, 为业务配置提供配置依据. 在业务性能分析 Petri 网模型进行一次业务性能计算的过程中, 变迁 t_j 点火后, $T_L = T_L \cup \{T_j(n)\}$, 该变迁不再点火.

2 业务性能计算

在业务性能分析 Petri 网模型中, 一次业务性能计算可能的变迁类型有单因单果变迁、多因单果变迁、并行变迁、复合变迁 4 种, 如图 2 所示, 其对应的业务性能计算方法如下.

(1) 单因单果变迁是前置库所数和后置库所数都为 1 的变迁, 其业务性能计算公式为

$$M'(s) = M(s) - \frac{\lambda({}^*t) K_{\max}({}^*t) - M({}^*t)}{\lambda({}^*t) K_{\max}({}^*t) - K_{\min}({}^*t)} U(t) \quad (2)$$

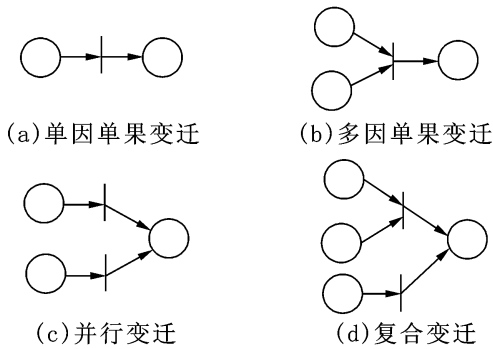


图 2 业务性能分析 Petri 网的 4 种变迁类型

(2)多因单果变迁是前置库所数大于 1、后置库所数等于 1 的变迁,其业务性能计算公式为

$$M'(s) = M(s) - \sum_{p \in \cdot t} \frac{\lambda(p)K_{\max}(p) - M(p)}{\lambda(p)K_{\max}(p) - K_{\min}(p)} \cdot W(p, t)U(t) \quad (3)$$

(3)并行变迁是有相同后置库所的多个单因单果变迁,其业务性能计算公式为

$$M'(s) = M(s) - \sum_{t' \in s} \frac{\lambda(t')K_{\max}(t') - M(t')}{\lambda(t')K_{\max}(t') - K_{\min}(t')} U(t) \quad (4)$$

(4)复合变迁是有相同后置库所的多个单因单果变迁或多因单果变迁,其业务性能计算公式为

$$M'(s) = M(s) - \sum_{t' \in s} \sum_{p \in \cdot t'} \frac{\lambda(p)K_{\max}(p) - M(p)}{\lambda(p)K_{\max}(p) - K_{\min}(p)} W(p, t)U(t) \quad (5)$$

业务性能分析 Petri 网一次业务性能计算的具体过程如下:

(1)初始时,初始库所的状态满足 $\forall s \in S_c, \lambda(s) \cdot K_{\max}(s) < M(s) \leq K_{\max}(s)$,业务性能分析 Petri 网中没有变迁满足点火条件;

(2)随着业务的运行,初始库所的状态发生变化,部分初始库所中业务因素的状态下降,满足点火条件 $M(s) \leq \lambda(s)K_{\max}(s)$,业务性能分析 Petri 网按顺序执行变迁点火,直到最后一个变迁点火完毕;

(3)在业务性能分析 Petri 网变迁点火过程中,变迁时间标签集合 T_L 记录变迁点火的传播路径,并防止变迁的反复点火;

(4)在业务性能分析 Petri 网变迁点火过程中,按照业务性能计算公式自动完成业务性能计算,业务性能就是输出库所中的状态值;

(5)业务性能计算结果和变迁时间标签集合 T_L 入库保存;

(6)重置中间库所的状态,清空变迁时间标签集合 T_L ,等待下一次初始库所的状态变化满足其对应

变迁的点火条件.

如果业务性能最大值为 1,表示业务各方面都处于最优状态;如果业务性能最小值为 0,表示业务各方面都处于最差状态.结合业务性能计算结果和变迁时间标签集合 T_L 能够发现业务性能瓶颈.

3 模型参数训练

在业务性能分析 Petri 网模型中,变迁权函数 U 通常是根业务开发者经验设定的,缺乏准确性.本节采用特定样本对变迁权函数 U 进行训练,以提高权函数 U 的准确性.

每次采用两个特定样本 (S_{c1}, y_1) 和 (S_{c2}, y_2) 对权函数 U 进行训练,其中, S_{c1} 和 S_{c2} 表示样本中初始库所的集合,满足 $\text{Card}(S_{c1} - S_{c2}) = 1$,且 $\text{Card}(S_{c2} - S_{c1}) = 1$, Card 表示集合中元素的个数, y_1 和 y_2 表示业务性能.权函数 U 的样本训练过程如下:

(1)在业务性能分析 Petri 网中,分别导入样本中的初始库所集 S_{c1} 和 S_{c2} , S_{c1} 、 S_{c2} 中只有一个初始库所 s 的状态值不同,即 $S_{c1} - S_{c2} = S_{c2} - S_{c1} = \{s\}$;

(2)按照前一节所述的业务性能计算方法分别对两组样本进行业务性能计算,则可以得到方程组

$$\left\{ \begin{aligned} 1 - \sum - \frac{\lambda(s)K_{\max}(s) - M_1(s)}{\lambda(s)K_{\max}(s) - K_{\min}(s)} U(t_s) &= y_1 \\ 1 - \sum - \frac{\lambda(s)K_{\max}(s) - M_2(s)}{\lambda(s)K_{\max}(s) - K_{\min}(s)} U(t_s) &= y_2 \end{aligned} \right. \quad (6)$$

$M_1(s)$ 、 $M_2(s)$ 、 y_1 、 y_2 的值在样本 (S_{c1}, y_1) 和 (S_{c2}, y_2) 中可以得到, $\lambda(s)$ 、 $K_{\min}(s)$ 、 $K_{\max}(s)$ 为模型参数,求解此方程组可以得到输入库所 s 对应变迁 t_s 的权函数 $U(t_s)$ 的值;

(3)按照步骤(1)和步骤(2),调节样本中初始库所的值,使 $S_{c1} - S_{c2} = S_{c2} - S_{c1} = \{s'\}$,可以求得库所 s' 对应变迁的权函数的值;

(4)按照步骤(3)继续下去,直到求得所有变迁的权函数的值.

4 误差分析

在业务性能分析 Petri 网模型中,业务性能计算的误差主要来源于输入误差和模型参数误差两部分.输入误差是指初始库所状态值的误差,主要是由初始库所中业务因素的数据采集时间不同步引起的.模型参数误差是指变迁权函数 U 的误差,主要是由变迁权函数 U 训练样本中的业务性能值 y 存在随机误差引起的.

如果将属于同一业务的采集数据项进行集中采集,则能够保证在业务性能分析误差允许的时间段内完成采集属于同一业务的采集数据项.在业务性能分析 Petri 网模型中,对于初始库所中影响业务性能的因素,采用该方法进行数据采集,能够将输入误差减小到业务性能分析允许的范围内.如图 3 所示的复合变迁中,假设业务活动 a 允许的业务性能分析误差为 ϵ ,则每个初始库所 s_i 允许的数据采集误差均为 $\epsilon/\sum U$,采用前述的数据采集方法能够在该误差范围内实现数据采集.可以看出,初始库所 s_i 允许的输入误差与有向弧的权函数 W_i 和变迁 T_i 的权函数 U_i 都无关,但是对于初始库所 s_i ,若其对应的 W_i 或 U_i 较大,则降低其输入误差能够较大幅度地降低业务性能计算误差,这是因为在业务性能分析 Petri 网模型中,输入误差通过权函数 W_i 和权函数 U_i 传递.因此,对于初始库所 s_i ,若其对应的权函数 W_i 或权函数 U_i 较大,则应当进一步减小其数据采集误差,即其数据采集误差应远小于 $\epsilon/\sum U$.

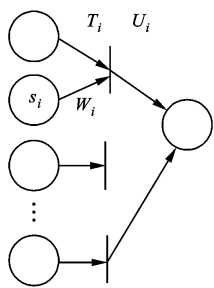


图3 复合变迁误差分析实例

在前一节中,采用特定样本训练业务性能分析 Petri 网模型中的变迁权函数 U 能够提高权函数 U 的准确性,但是,训练样本 (S_c, y) 的 y 值存在随机误差,导致一次样本训练的计算结果存在随机误差.在权函数 U 的样本训练过程中,可以通过对权函数 U 进行多次样本训练取平均值的方式,减少权函数 U 的随机误差.

5 实例分析

本节通过实例分析说明业务性能分析 Petri 网模型的自动业务性能计算过程和自动业务性能瓶颈发现过程,权函数 U 的样本训练过程,以及输入误差和模型参数误差对业务性能计算结果的影响.

5.1 业务性能计算过程实例分析

采用图 4 中的实例,分析业务性能分析 Petri 网进行自动业务性能计算的过程,图中初始库所和中

间库所都处于初始状态,且 $K_{\min}(s)=0, K_{\max}(s)=500, \lambda(s)=0.9, s \in S_c; K_{\min}(s)=0, K_{\max}(s)=1, \lambda(s)=1, s \in S_o$. 由于初始库所集 $\{s_1, s_2, s_3, s_4, s_5\}$ 满足变迁点火条件,该业务性能分析 Petri 网将按照变迁顺序自动点火下去,直到输出库所 d 的状态发生改变,输出业务性能.具体计算过程为:按照复合变迁计算公式得 $M(a)=92.7\%$,按照多因单果变迁计算公式得 $M(b)=94.5\%$,按照单因单果变迁计算公式得 $M(c)=91.2\%$,按照并行变迁计算公式得 $M(d)=82.4\%$,最后得到业务性能计算结果为 82.4% .

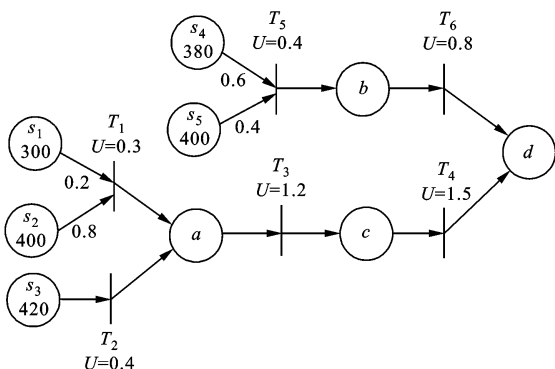


图4 业务性能分析 Petri 网实例

在图 4 所示的业务性能分析 Petri 网实例中,变迁点火结束后,变迁时间标签集合 $T_L = \{T_1(1), T_2(2), T_5(3), T_3(4), T_6(5), T_4(6)\}$. 结合业务性能计算过程,可以看出变迁 T_1 和 T_2, T_5 点火较早,而 $M(a)=92.7\%, M(b)=94.5\%$,可以看出变迁 T_1 和 T_2 点火后,其后置库所中业务活动 a 的性能下降了 7.3% ,幅度较大,变迁 T_1 前置库所中的业务因素和变迁 T_2 前置库所中的业务因素是潜在的业务性能瓶颈.

由实例分析结果可以看出,业务性能分析 Petri 网模型能够很好地建模影响业务活动性能的因素和业务活动之间的逻辑连接关系,且模型一旦建立后,通过变迁点火规则及业务性能计算公式,能够实现自动业务性能计算,同时依据变迁时间标签集合 T_L ,能够实现自动业务性能瓶颈发现.业务性能分析结果用于后续相关业务配置,提高业务性能.

5.2 权函数样本训练过程实例分析

以图 5 中的并行变迁为例,分析权函数 U 的样本训练过程,并行变迁的输入库所数为 3. 设定样本 (S_{c1}, y_1) 中 $M_1(s_1)=300, M_1(s_2)=400, M_1(s_3)=400, y_1=0.8$; 样本 (S_{c2}, y_2) 中 $M_2(s_1)=350, M_2(s_2)=400, M_2(s_3)=400, y_2=0.85$. 则有 $S_{c1}-S_{c2}=S_{c2}-S_{c1}=\{s_1\}, \text{Card}(S_{c1}-S_{c2})=\text{Card}(S_{c2}-S_{c1})=1$.

在业务性能分析 Petri 网模型中, $\lambda(s_i)=0.9$, $K_{\min}(s_i)=0, K_{\max}(s_i)=500, i=1, 2, 3$, 分别导入样本中的初始库所集 S_{c1} 和 S_{c2} 后, 在权函数 U 的样本训练过程中可以得到方程组

$$\begin{aligned} 1 - \frac{450-400}{450}U_2 - \frac{450-400}{450}U_3 - \frac{450-300}{450}U_1 &= 0.8 \\ 1 - \frac{450-400}{450}U_2 - \frac{450-400}{450}U_3 - \frac{450-350}{450}U_1 &= 0.85 \end{aligned}$$

求解此方程组可以得到权函数 $U_1=0.45$.

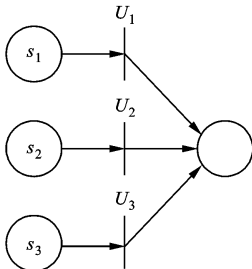


图 5 并行变迁实例

同样, 调节样本中初始库所的状态值, 设定样本 (S_{c1}, y_1) 和样本 (S_{c2}, y_2) , 使其满足条件 $S_{c1}-S_{c2}=\{s_2\}$, 则可以求得权函数 U_2 的值; 调节样本中初始库所的状态值, 设定样本 (S_{c1}, y_1) 和样本 (S_{c2}, y_2) , 使其满足条件 $S_{c1}-S_{c2}=S_{c2}-S_{c1}=\{s_3\}$, 则可以求得权函数 U_3 的值.

5.3 输入误差和模型参数误差实例分析

采用图 4 中的业务性能分析 Petri 网实例进行输入误差分析, 假设初始库所 s_2 和初始库所 s_4 存在输入误差, 输入误差均为 2%. 按照业务性能计算方法进行业务性能计算, 初始库所 s_2 的输入误差和初始库所 s_4 的输入误差引起的业务性能计算结果误差分别为 0.864% 和 0.384%, 表明初始库所 s_2 的输入误差对业务性能计算结果影响较大, 这是因为初始库所 s_2 对应的权函数 W 和权函数 U 的乘积为 $W_2 * U_1 * U_3 * U_4=0.432$; 初始库所 s_4 对应的权函数 W 和权函数 U 的乘积为 $W_4 * U_5 * U_6=0.192$. 初始库所 s_2 对应的权函数 W 和权函数 U 的乘积较大, 其输入误差对业务性能计算结果影响较大, 这与第 4 节中输入误差的分析结果相符.

采用图 5 中的并行变迁实例进行模型参数误差分析, 在权函数样本训练实例分析过程中, 假设样本 (S_{c1}, y_1) 中 y_1 的值存在随机误差, 误差值为 1%, 则在权函数 U_1 的一次计算结果中, 权函数 U_1 的随机

误差为 9%. 为了减少权函数 U_1 的随机误差, 设计样本训练实验, 对权函数 U_1 进行多次样本训练, 采用计算结果的平均值作为权函数 U_1 的值. 采用图 5 中的并行变迁实例对权函数 U_1 进行多次样本训练, 权函数 U_1 的随机误差随着实验次数的变化关系如图 6 所示. 可以看出, 经 6 次样本训练后, 权函数 U_1 的随机误差减小到 2% 以下.

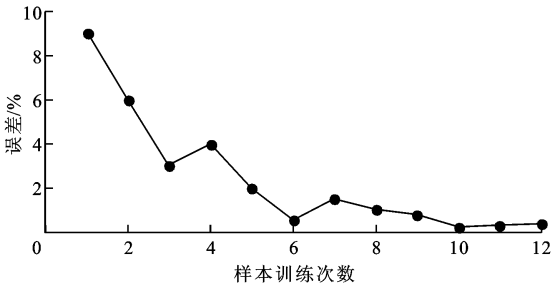


图 6 权函数 U_1 的随机误差随着实验次数的变化关系

6 结 论

在业务管理系统中, 本文提出的采用 Petri 网的业务性能分析方法能够建立有效的业务性能分析模型, 实现业务性能自动计算和业务性能瓶颈自动发现. 基于本文的业务性能分析结果, 研究高效的业务配置策略是后续的研究工作, 主要包括以下两个方面: 将业务特征和业务活动对应, 实施业务活动配置, 消除影响业务活动的瓶颈业务特征, 提高业务活动性能; 将业务活动和业务过程对应, 实施业务配置, 消除影响业务过程的瓶颈业务活动, 进而提高业务性能.

参考文献:

[1] OGRINZ M. Mashup patterns: designs and examples for the modern enterprise [M]. Beijing: China Machine Press, 2010.

[2] KIERAS D. Using the keystroke-level model to estimate execution times [EB/OL]. (2010-11-09)[2012-03-15]. <http://www-personal.umich.edu/~itm/688/KierasKLMTutorial2001.pdf>.

[3] DIAO Yixin, KELLER A. Quantifying the complexity of IT service management processes [M]. Berlin, Germany: Springer-Verlag, 2006.

[4] DOS SANTOS C R P, CHENG W, LOEWENSTERN D. Performance management and quantitative modeling of IT service processes using mashup patterns [C] //Proceedings of the International Conference on Network and Service Management. Piscataway, NJ, USA: IEEE, 2011:24-28.

- [5] LI Yinan, CHEN Ing-ray. Design and performance analysis of mobility management schemes based on pointer forwarding for wireless mesh networks [J]. IEEE Transactions on Mobile Computing, 2011, 10(3):349-361.
- [6] ZENG Sen, HUANG Shuangxi, FAN Yushun. Service oriented enterprise network performance analysis [J]. Tsinghua Science and Technology, 2009, 14(4): 492-503.
- [7] GAI Bingshuai, WANG Jinlin, LIU Xue. Ontology-based service modeling method for service performance management system [C]//Proceeding of the 2nd International Conference on Business Management and Electronic Information. Piscataway, NJ, USA: IEEE, 2012.
- [8] REISIG W. Petri nets: an introduction [M]. Berlin, Germany: Springer-Verlag, 1985.
- [9] MCGUINNESS D L, HARMELEN F V. OWL web ontology language overview [EB/OL]. (2004-02-10) [2012-03-16]. <http://www.w3.org/TR/owl-features/>.
- [10] HORROCKS I, PATEL-SCHNEIDER P F, BOLEY H, et al. SWRL: a semantic web rule language combining OWL and ruleML [EB/OL]. (2004-05-21) [2012-03-16]. <http://www.w3.org/Submission/SWRL/>.
- (编辑 武红江)