软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(4):696−714 [doi: 10.3724/SP.J.1001.2013.04272]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

考虑失效恢复的组合服务性能建模与分析*

顾 军[1,2], 罗军舟[1], 曹玖新[1], 李 伟[1]

[1](东南大学 计算机科学与工程学院,江苏 南京 211189)

[2](中国矿业大学 计算机科学与技术学院,江苏 徐州 221116)

通讯作者: 顾军, E-mail: jgu@seu.edu.cn, http://www.seu.edu.cn

摘 要: 互联网环境下运行的组合服务易受到资源故障和组件失效影响而导致失效.已有的失效恢复措施在提高服务可用性的同时也会对服务的性能产生负面影响.为了对失效可恢复情况下的组合服务性能进行量化,通过综合组合服务失效类型和恢复策略,给出一种考虑失效恢复的组合服务性能分析模型.采用排队 Petri 网(queueing Petri net,简称 QPN)描述组合服务的失效发生及其恢复处理过程,重点研究实施重试和替换策略的服务运行情况.详细描述了考虑失效恢复的服务节点和链路QPN模型的内部结构,在此基础上,通过服务交互机制构建组合服务分散执行的性能模型.最后,采用 QPME 工具仿真和比较不同失效发生率、失效类型分布和恢复策略下组合服务模型的性能表现.结果表明,该方法能够定量分析失效恢复对组合服务性能的影响,有助于指导不确定网络环境下的信息服务系统失效恢复策略实施方案的设计.

关键词: 组合服务;分散执行;失效;恢复;交互;排队 Petri 网

中图法分类号: TP311 **文献标识码**: A

中文引用格式: 顾军,罗军舟,曹玖新,李伟.考虑失效恢复的组合服务性能建模与分析.软件学报,2013,24(4):696−714. http://www.jos.org.cn/1000-9825/4272.htm

英文引用格式: Gu J, Luo JZ, Cao JX, Li W. Performance modeling and analysis for composite service considering failure recovery. Ruanjian Xuebao/Journal of Software, 2013,24(4):696−714 (in Chinese). http://www.jos.org.cn/1000-9825/4272.htm

Performance Modeling and Analysis for Composite Service Considering Failure Recovery

GU Jun[1,2], LUO Jun-Zhou[1], CAO Jiu-Xin[1], LI Wei[1]

[1](School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)

[2](School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)

Corresponding author: GU Jun, E-mail: jgu@seu.edu.cn, http://www.seu.edu.cn

Abstract: The Internet provides an open, dynamic, and uncertain environment. A composite service in this environment faces more failure threats caused by resource faults and component failures. The existing failure recovery measures can improve the availability of composite service and at the same time, bring the negative impact on service performance. In this paper, an analytical model considering failure recovery is proposed to quantitatively analyze the performance of composite service with the failure probability and recovery ability through synthesizes the fault taxonomy and recovery strategies of service failure. Queueing Petri nets (QPNs) are used to model the occurrence of failures and the corresponding recovery handling processes, which focuses on targeting the service executing process under retrying and replacement strategies. The inner structure of server node and link QPN models considering failure and recovery impact are expatiated and depicted. On this ground, a performance model of composite service with decentralized execution is given, and

* 基金项目: 国家自然科学基金(61003257, 61070161, 61070158, 61003311, 61272531, 61272054, 61202449); 国家重点基础研究发展计划(973)(2010CB328104); 国家高技术研究发展计划(863)(2013AA013503); 国家科技支撑计划(2010BAI88B03, 2011BAK21B02); 国家核高基科技重大专项(软件类)(2010ZX01044-001-001); 高校博士点专项基金(20110092130002); 江苏省自然科学基金重点项目(BK2008030); 江苏省网络与信息安全重点实验室(BM2003201); 教育部网络与信息集成重点实验室(93K-9)

收稿时间: 2011-05-16; 定稿时间: 2012-05-18

© 中国科学院软件研究所 http://www.jos.org.cn

five interaction relations are taken into account. Finally, QPME tool is used to simulate and compare the performance measure results under different failure rates, failure styles proportions, and recovery strategies. The numerical results show that this method can quantitatively analyze the impact of failure recovery measure on service performance, and can also be helpful for providing rational solutions to implement failure recovery measure for information service system in uncertain network environments.

Key words: composite service; decentralized execution; failure; recovery; interaction; queueing Petri net

面向服务体系架构(service oriented architecture,简称 SOA)[1]通过组合不同提供商的服务来实现复杂业务逻辑,具有可重用、松耦合、易扩展的特点,已经成为构建信息服务系统的首选架构.为了确保服务提供者和消费者按照约定的服务等级协议(service level agreement,简称 SLA)交互,需要对组合服务的服务质量(quality of service,简称 QoS)进行分析和预测.

在 SOA 架构下,服务质量评价指标主要有:服务性能(吞吐量、延迟、响应时间、抖动、错误率)、可靠性、可用性、容量、开销、安全性等[2].针对不同评价指标提出了不少分析方法,如:文献[3]通过随机 Petri 网评价面向服务体系结构的性能;文献[4]提出对组合的 Web 服务的性能和可靠性瓶颈进行闭环分析的随机模型方法;文献[5]使用 UML 表示组合 Web 服务,然后引入框图、故障树和马尔可夫模型对扩展的 UML 模型进行自动映射,实现对组合服务可信赖性的分析;文献[6]认为,组合服务的可靠性和性能的评估不仅要考虑系统所有执行路径的运行情况和服务间的状态转移过程[7],还要考虑系统的运行场景,即对用户如何使用系统的定量刻画.为此,从不同运行场景角度,利用离散时间马尔可夫链相关性质和公式综合估算了服务组合的可靠性和其他性能,并针对具体服务组合的瓶颈进行了分析,提出改进措施;文献[8]采用排队网分析失效影响下的多层 Web 服务的性能,但是基于乘积网络的假设限制了该方法的适用范围;文献[9]研究考虑错误率影响的树形网格服务的性能优化,但是无法涵盖经典组合服务应用模式.

上述研究大都将性能、可靠性、可用性等分开处理,也没有考虑失效可恢复情况下不同恢复措施对服务性能的影响.事实上,服务失效的发生降低了服务的可靠性和可用性,对失效的恢复虽然可以提高可靠性,但也增加了服务节点的处理开销,使得失效恢复期间对服务的有效处理能力下降,导致系统整体可用性和性能的下降.因此,对于可失效和恢复的组合服务,如何评估不同类型失效的发生率、概率分布及其恢复策略的执行效率对性能的影响,是设计合理而有效的失效恢复实施机制的关键.换句话说,失效恢复机制的使用应该是受限的.因此,本文在对组合服务性能的建模与分析过程中,不但考虑到服务执行阶段由于各种故障导致的失效对性能的影响,而且关注失效可恢复情况下服务性能的统一表达,评估失效恢复能力对提高系统服务质量的重要作用.

为了建立组合服务的性能模型,首先要建立服务节点和链路的性能模型,描述无失效发生时节点的结构和性能情况.排队论[10]通过对服务对象到来及服务时间的统计研究,得出这些数量指标(等待时间、排队长度、忙期长短等)的统计规律,然后根据这些规律来改进系统结构或重新组织被服务对象,使得系统既能满足服务对象的需要,又能使系统的费用最经济或某些指标最优.但是排队模型由于存在信息量小、无结构化机制的弊端,只被应用于较为简单的情况[11].其次,要对可能出现失效的节点和链路资源分别建立失效模型和恢复模型,用来描述失效过程及失效后的恢复操作.不同的失效类型所引发的恢复行为是有区别的,建立的模型也有所不同.随机 Petri 网[12]以研究模型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状态变化以及变化之间的关系,非常适合描述松耦合的组合服务系统中不确定的失效恢复行为.最后,要将上述模型结合起来,建立起包含多种交互关系的组合服务性能模型,描述在一个特定的业务流程结构中某些节点或链路失效后,采取的不同失效恢复策略对性能的影响.

排队 Petri 网(queueing Petri net,简称 QPN)[13]继承和发展了排队网模型和随机 Petri 网的优点,既能定量地建模资源的运行性能,也可以刻画多层系统之间的依赖关系,易于对系统状态进行全面而有效的描述,精确刻画系统的不确定行为以及行为组合关系,便于计算各种分析指标,能够很好地建模和定量分析失效恢复对组合服务性能的影响,从而综合评价组合服务的运行状态.排队 Petri 网的基本概念见附录,详细介绍可见文献[13,14]. Kounev 等人的研究工作表明,排队 Petri 网具有很强的定量评价能力和行为描述能力[15,16],能够很好地对分布式系统和网格系统进行建模和模拟,可用于系统的性能预测、能力规划和在线性能管理.本文将在排队 Petri 网的

© 中国科学院软件研究所 http://www.jos.org.cn

系统模型框架上,采用图形化的方式完成组合服务动态行为的描述,将简单的服务组合成复杂的工作流模型,并准确反映和刻画组合服务的性能及可用性的动态性.

本文第 1 节介绍组合服务的分散执行过程.第 2 节分析服务的失效类型与恢复策略对性能的影响.第 3 节给出服务链路和节点的排队 Petri 网表示,在此基础上,建立考虑失效恢复的组合服务性能模型.第 4 节通过仿真实验分析不同失效发生率、失效类型分布和恢复策略下所提组合服务性能模型的有效性.第 5 节总结全文并展望下一步工作.

1 组合服务的分散执行

为了更好地满足用户的需求,基于 SOA 的信息服务系统往往是由现有的一组服务按照一定业务逻辑加以集成而来.多层次、具有高可扩展性的服务组合为用户和提供商提高运行效率提供了极大的便利.组合服务的执行主要有两种形式[17,18]:集中执行(centralized execution)方式需要一个控制中心完成所有服务的调用和数据传输,服务之间不发生直接交互,因此存在性能瓶颈,只适合小规模服务系统;分散执行(decentralized execution)方式允许服务之间按照业务流程需要直接交互和传输数据,有利于减少通信成本、提高组合服务吞吐量,适合构建大规模信息服务系统.本文研究分散执行组合服务的性能问题.

定义 1(顺序(sequential)). 服务 S_i 执行完毕后,S_j 才开始执行,记作 $S_i>S_j$.

定义 2(重复(iterative)). 服务 S_i 重复执行 k 次,记作 $\mu S_i(k)$.

定义 3(并发联合(concurrent-unite)). 服务 S_i 和 S_j 相互独立执行,只有都执行完毕后才会调用后继的服务(集),记作 $S_i|S_j$.

定义 4(并发汇合(concurrent-join)). 服务 S_i 和 S_j 相互独立执行,任何一个执行完毕后都可以调用后继的服务(集),无需等待另一个,记作 $S_i¦S_j$.

定义 5(条件(conditional)). 前驱服务执行完毕后,按照一定的条件概率选择 S_i 和 S_j 中的一个开始执行,记作 $S_i^{(p)}+S_j^{(1-p)}$.

定义 6. 组合服务的分散执行模型是一个七元组,即 $CS=\{N,S,CL,IS,NS,RN,RL\}$:

(1) N,S,CL 分别是有限节点集、可用服务集和服务链路集,其中,服务链路是对跨地域网络基础设施的虚拟化,屏蔽了实际的网络拓扑和通信细节;

(2) $IS:N\times N\to\{>,\mu,|,¦,+\}$是节点间的交互关系函数,其中,$>,\mu,|,¦,+$分别表示顺序、重复、并发联合、并发汇合和条件关系;

(3) $NS:N\to S^*$是节点的可用服务函数,$NS(N_i)=S_i=\{S_{i,1},S_{i,2},\ldots,S_{i,m}\}$表示节点 N_i 的可用服务集,其中,$S_{i,j}$ 是节点 N_i 的第 j 个可用服务;

(4) $RN:N\to(0,1)$是节点成功运行不失效的概率,$RL:CL\to(0,1)$是链路成功通信不失效的概率,根据实际情况,节点和链路的成功概率不为 1.

一个分散执行的组合服务运行场景如图 1 所示.起始服务 S_1 执行后调用并发联合的两个服务 S_2 和 S_3,记作 $S_1>(S_2|S_3)$.S_1 称为 $S_2|S_3$ 的前驱服务,$S_2|S_3$ 称为 S_1 的后继服务(集).S_2 和 S_3 必须都完成后才能调用后继服务 S_4.服务 S_4 处于重复方式,连续执行 κ 次后进入后继的并发汇合服务集 $S_5¦S_6$.S_5 和 S_6 同时启动后分别独立执行,只要有一个完成就可以调用后继的条件选择服务集 $S_7^{(p)}+S_8^{(1-p)}$.S_7 被前驱服务(集)选中执行的概率为 p,S_8 被选中的概率为$(1-p)$.该运行场景可以形式化描述为 $S_1>(S_2|S_3)>\mu S_4(k)>(S_5¦S_6)>(S_7^{(p)}+S_8^{(1-p)})$.

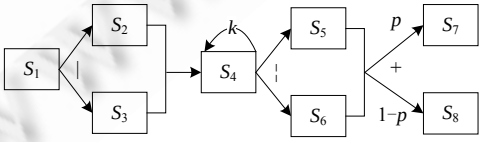


Fig.1 Decentralized execution of composite service

图 1 组合服务的分散执行

© 中国科学院软件研究所 http://www.jos.org.cn

令 S_i 的平均逗留时间(sojourn time)为 St_i,可靠度(reliability)为 Re_i,则不同交互关系下的组合服务平均逗留时间 St 和可靠度 Re 见表 1.

Table 1 Computing *St* and *Re* of interactive services

表 1 计算服务交互的 *St* 和 *Re*

	St	Re
$S_2\|S_3$	$\max(St_2,St_3)$	$Re_2\times Re_3$
$\mu S_4(k)$	$k\times St_4$	$(Re_4)^k$
$S_5\¦S_6$	$\min(St_5,St_6)$	$St_5<St_6$? Re_5: Re_6
$S_7^{(p)}+S_8^{(1-p)}$	$p\times St_7+(1-p)\times St_8$	$p\times Re_7+(1-p)\times Re_8$

并发联合关系下,$S_2|S_3$ 的平均逗留时间由最后完成的服务决定,可靠度由并发服务共同决定;重复关系下,$\mu S_4(k)$的平均逗留时间是单个服务的 k 倍,可靠度随着重复次数的增加反而降低;并发汇合关系下,$S_5\¦S_6$ 的平均逗留时间由最先完成的服务决定,可靠度由最先完成的服务决定;条件关系下,$S_7^{(p)}+S_8^{(1-p)}$ 的平均逗留时间和可靠度取决于选取不同服务的概率,选中率越高,其在最终结果中的比重就越高.然而,可靠度影响下的平均逗留时间的计算却很难直接计算,加上组合服务流程中包含的多种交互关系,使得具有并发特征的组合服务性能分析更加复杂.

2 服务失效与恢复

服务失效是服务软、硬件运行行为对消费者需求的偏离,是服务系统的一个动态特征.服务发生失效后将会停止工作直到失效被恢复.

SOA 体系结构的主要步骤包括发布(publishing)、发现(discovery)、组合(composition)、绑定(binding)和执行(execution),每一步都可能因故障或错误导致服务调用失效[19].本文主要研究失效与恢复对服务执行性能的影响,因此假定执行阶段以外的故障已经得到有效处理和控制.

组合服务的执行由驻留在地理分散的服务节点上的服务软件通过服务链路的数据交换协作完成.负责数据传输的服务链路失效的直接后果是数据丢失,从而导致数据重传.如果运行服务的节点之间只有一条通信物理链路,那么在服务交换信息期间通信链路发生失效后,服务的执行将失败,任务也随之被终止.由于服务链路具有虚拟性,因此链路的失效恢复行为很难直接去建模.本文的处理方法是:如果失效只是引起传输延迟的增加,那么就作为链路模型的一部分来处理;如果失效引起连接失败从而导致服务终止或替换,那么就归入节点来处理.

服务节点一般有输入、等待、计算和输出这 4 步.在有效执行阶段,任务持续进入节点,完成后输出到下一个节点.在失效恢复阶段,不同失效类型和恢复策略对节点行为的影响是不同的.表 2 描述了执行阶段服务失效的基本分类和故障所属的层次,给出了产生失效的原因、导致的结果及相应的恢复策略[20,21].

Table 2 Failure and recovery in services execution

表 2 服务执行时的失效与恢复

失效名称	失效原因	失效结果	恢复策略
服务崩溃	节点故障,软件开发错误	无法执行	替换、局部重构
不正确结果	输入不正确,服务出错	返回错误结果,	替换
超时	节点崩溃,连接失败	不正确结果,没有返回结果	重试、替换
不满意	服务质量偏离预期效果	违反 SLA	重试、替换

重试策略既可以是单个任务的重新执行,也可以是系统的重新启动和复位.替换策略是指发生失效后,前驱服务(集)的任务被调度到新的替代服务上执行,完成后输出到业务流程规定的后继服务(集)中.局部重构是指将受到单个失效影响的流程进行部分的重新构造.因为局部重构可以看作是更大范围和更多次数的替换,其对服务性能影响的评估方式是一致的,所以本文只考虑重试和替换两种策略,以降低模型讨论的复杂性.

© 中国科学院软件研究所 http://www.jos.org.cn

事实上,服务的失效原因和恢复策略常常是相互交叉的,使得失效的分类变得很困难,影响了对失效发生概率的分析.比如,在替换恢复策略下,被替换节点无法接受原来的前驱服务(集)的任务调用,其行为变化被忽略,而新的替换节点完成配置和调用前不能从前驱服务(集)接受请求,因此这段期间的任务请求都将被丢弃.在重试恢复策略下,服务节点没有发生变化,其输入和输出环节可以同步考虑,等待和处理环节也是紧耦合的.因此,本文从失效恢复措施对节点行为影响的角度将失效类型重新分为以下 4 类:

- I 类失效:引发替换恢复策略的失效,忽略被替换节点的遗留任务,从零负载开始工作;
- II 类失效:停止任务请求的输入和服务结果的输出,新到的请求将丢弃,已经进入节点等待的任务将会继续处理.恢复后,开始接受新的服务请求,并将已经完成的服务结果传输给后继服务(集);
- III 类失效:继续接受新的服务任务,处理等待的任务,输出结果.失效的任务将会被要求执行(可能是对结果不满意),因此需要有一个反馈和重新调用的过程,使得系统负载和延迟增加;
- IV 类失效:停止输入、输出和处理,丢弃新到的任务,清空等待的任务.恢复后从零负载开始运行.

引入服务失效恢复机制后,服务执行过程分为有效执行和失效恢复两个阶段.当然,失效恢复具有一定的概率,即失效可以恢复时,恢复后的服务继续执行,直到服务 S_i 执行完毕;反之,失效不可恢复,S_i 被迫终止,在该节点的任务执行宣告失败[22].本文只研究失效可恢复的情况,即当服务节点可修复性故障时,通过恢复策略解决组合服务遇到失效而被迫终止的问题.

为了更好地描述失效修复对服务性能的影响,本文对节点软、硬件和链路的分析满足以下假设:

1) 所有失效都是可恢复的;
2) 服务请求到达率与节点状态是独立的且满足泊松分布[8,23];
3) 节点失效发生率满足泊松分布,I 类~IV 类失效的概率 $p_N(i)$为常数,且满足 $\sum_{i=1}^{4} p_N(i)=1$;
4) 所有服务节点被调用后立即执行,其无失效工作时间服从指数分布;
5) 服务链路的失效概率 φ_L 为常数,无失效工作时间服从指数分布;
6) 由于失效恢复主要通过恢复程序的执行完成,所以失效恢复时间服从指数分布.

3 基于排队 Petri 网的性能建模

3.1 服务链路模型

服务链路的 QPN 模型如图 2 所示.黑色方框图代表瞬时变迁(immediate transition),圆圈表示普通库所(ordinary place),带有竖线的圆圈表示队列库所(queueing place).代表前驱服务(集)的普通库所 Input 向链路模型发送任务令牌 s.由于链路存在失效概率 φ_L,服务链路的传输环节 $LP(G/M/1/FCFS$ 队列)处理后由变迁 lt_2 模拟链路失效的发生.没有发生失效的任务令牌 s 移入库所 SL 后用符号 a 表示,经过变迁 lt_3 触发后输出到库所 Output;而发生失效的任务令牌 s 移入库所 SL 后用符号 f 表示,经过变迁 lt_3 触发后将送回队列库所 LP,以模拟传输延迟的增加.

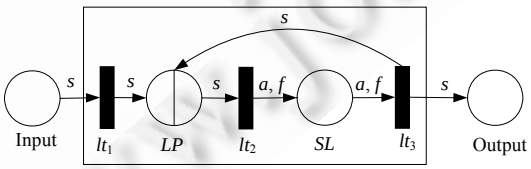


Fig.2 QPN model of the service link

图 2 服务链路的排队 Petri 网表示

图 2 所示的变迁描述见表 3.符号 $A\{x\}\rightarrow B\{y\}$表示模型中变迁触发模式(firing mode),令牌 x 在一次变迁中由库所 A 迁出后以 y 形式放在库所 B 中.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 3 Firing modes of transitions in link model

表 3 链路模型中的变迁触发模式

变迁	触发模式	变迁	触发模式
lt_1	1: $Input\{s\}\rightarrow LP\{s\}$	lt_3	1: $SL\{a\}\rightarrow Output\{s\}$ 2: $SL\{f\}\rightarrow LP\{s\}$
lt_2	1: $LP\{s\}\rightarrow SL\{a\}$ 2: $LP\{s\}\rightarrow SL\{f\}$		

对于 lt_2 这样的变迁属于自由选择冲突模型,选择哪一个变迁模式实施并不依赖于库所中的标识,而取决于变迁模式的触发权值(firing weight),即发生的概率.令变迁 lt_2 的两个变迁模式的触发权值分别为 $w_l(1)$和 $w_l(2)$,其与失效概率φ_L 的关系如下:

$$\varphi_L=\frac{w_l(2)}{w_l(1)+w_l(2)} \tag{1}$$

3.2 服务节点模型

服务节点的 QPN 模型如图 3 所示,白色方框表示时间变迁(timed transition).队列库所 NP 建模服务节点的处理环节,包含 $G/M/n/PS$ 队列,模拟具有 n 个 CPU 的服务节点.队列库所 NP'建模在失效恢复过程中用来替换或重启的服务节点,包含 $G/M/n/PS$ 队列,模拟失效恢复后服务任务从零负载开始的情况.服务节点上可能运行着多类服务,每类服务可以被多个组合服务使用.因此,在分析某个组合服务使用该服务节点的性能时,需要把属于该组合服务的任务区别开来.为了简化模型,假设服务节点上 N_i 上只运行两类服务 $S_{i,1}$ 和 $S_{i,2}$,我们分析的组合服务调用服务 $S_{i,1}$,用符号 s 表示,其他调用服务 $S_{i,1}$ 的组合服务用符号 o 表示,访问服务 $S_{i,2}$ 的令牌用符号 w 表示,3 种令牌的到达率都服从泊松分布.图 3 中的变迁描述详见附录中的表 10 和表 11.

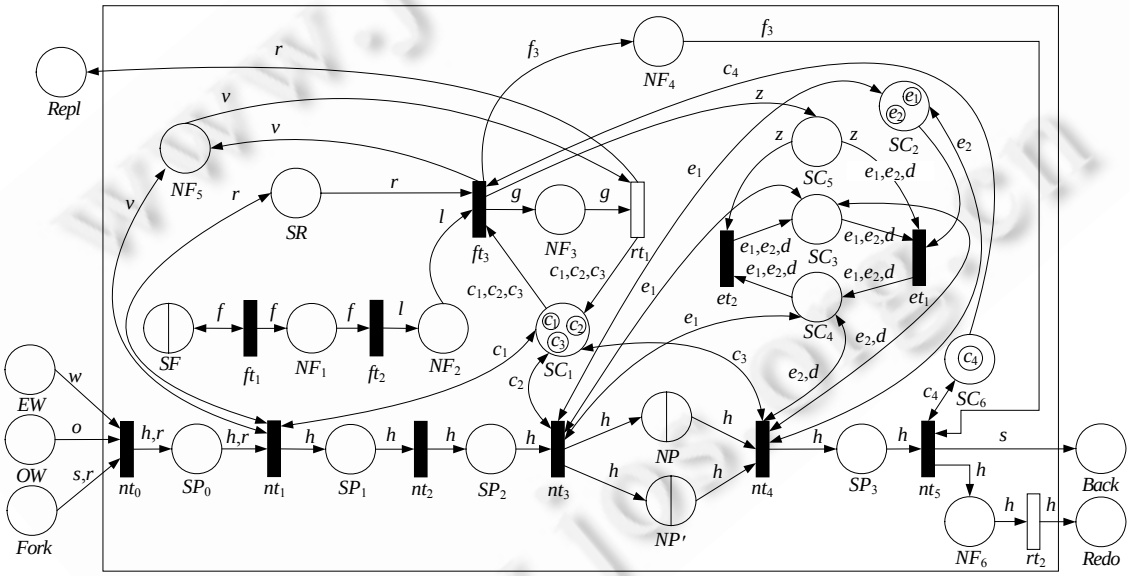


Fig.3 QPN model of the service node

图 3 服务节点的排队 Petri 网表示

模型中的令牌按照功能的不同可以分为如下 3 类:

- 一是任务令牌,包括 s,o 和 w.令牌 s 由组合服务前驱服务集 $Fork$ 输出,令牌 o 由共享同一类服务的 OW 输出,令牌 w 由访问其他类服务的 EW 输出.图 3 中用符号 h 表示令牌 s,o 或 w;
- 二是失效令牌,用于标识失效的发生,分别用符号 f,f_1,f_2,f_3 和 f_4 表示.图 3 中用符号 l 表示 f_1,f_2,f_3 或 f_4,

© 中国科学院软件研究所 http://www.jos.org.cn

符号 g 表示 f_1,f_2 或 f_4.队列库所 SF 建模失效发生器,包含一个 $G/M/\infty/IS$ 队列.当失效发生时,变迁 ft_1 触发后在库所 NF_1 中创建失效令牌 f.变迁 ft_2 的触发将令牌 f 转换成 f_1,f_2,f_3 和 f_4 中的一个.失效令牌 $f_i(i=1,2,3,4)$的出现由失效发生时的出现概率 $p_N(i)$决定.模型中可利用变迁 ft_2 触发模式的权值来实现.令产生令牌 f_i 的变迁模式为 $Mode(i)$,对应的权值为 $w_f(i)$,那么触发权值 $w_f(i)$与概率 $p_N(i)$满足如下关系:

$$p_N(i)=\frac{w_f(i)}{\sum_{j=1}^{4}w_f(i)} \tag{2}$$

- 三是控制令牌,用于控制服务任务在不同的运行状态和恢复策略下的执行过程.针对不同失效类型,服务模型在其恢复期间的行为是不同的.

表 4 给出了不同类型失效对服务模型的影响(√表示允许,×表示禁止,—表示没有发生动作).

Table 4 Impact of failures for service node model

表 4 失效对服务节点模型的影响

失效	接纳(nt_1)	输入(nt_3)	处理(NP,NP')	输出(nt_4)	零负载(NP,NP')	完成(nt_5)	替换
I 类	—	—	—	—	—	—	√
II 类	×	×	√	×	×	×	×
III 类	√	√	√	√	×	×	×
IV 类	×	×	×	×	√	×	×

控制令牌取值为[0,1]:1 表示令牌存在,允许变迁触发;0 表示令牌不存在,禁止变迁触发.控制令牌的使用见表 5.

Table 5 Using control tokens

表 5 控制令牌的使用

令牌	描述	1=允许	0=禁止	令牌	描述	1=允许	0=禁止
r	节点替换,任务接纳	I 类除外	I 类	v	销毁不接纳的任务	I 类,II 类,IV 类	有效,I 类,III 类
c_1	任务接纳	有效,III 类	I 类,II 类,IV 类	e_1	NP 和 NP'互斥输入	任何情况	—
c_2	任务输入	有效,III 类	I 类,II 类,IV 类	e_2	NP 和 NP'互斥输出	任何情况	—
c_3	任务输出	有效,III 类	I 类,II 类,IV 类	d	NP 或 NP'零负载处理	IV 类	IV 类除外
c_4	任务完成	III 类除外	III 类	z	NP 或 NP'零负载使能	IV 类	IV 类除外

基于上述讨论,服务节点模型的执行过程如下:

1) 代表前驱服务(集)的普通库所 $Fork$,OW 和 EW 通过变迁 nt_0 向节点模型传送任务令牌 s,r,o 或 w.进入的令牌在库所 SP_0 中等待进一步处理;
2) 变迁 nt_1 有 3 类触发模式:当节点处于有效状态或者发生 III 类失效时,触发模式{1,2,3}接受新到的任务,将令牌 h 移入库所 SP_1;当节点发生 I 类、II 类和 IV 类失效时,变迁 ft_3 会在库所 NF_5 中创建令牌 v,触发 nt_1 的变迁模式{4,5,6}销毁失效恢复期间到达的任务令牌 h;当 $Fork$ 传送的是令牌 r,触发变迁 nt_1 的模式 7 在库所 SR 中创建令牌 r 时,表示这个服务节点被选中代替其他服务执行任务;
3) SR 中的令牌状态$\{r=1\}$表示节点被选中,没有被替换;SC_1 中的令牌状态$\{c_1=1,c_2=1,c_3=1\}$表示节点可以正常工作;SC_2 中的令牌状态$\{e_1=1,e_2=1\}$表示当前 NP 可用,SC_6 中的令牌状态$\{c_4=1\}$表示没有发生 III 类失效,当前任务令牌 h 不需要被重新执行;
4) 如果$\{c_1=1,r=1\}$,则允许接纳任务 h,变迁 nt_1 触发后将令牌 h 从 SP_0 移入 SP_1;如果$\{c_1=0\}$或$\{r=0\}$,变迁 nt_1 禁止触发,不接受新任务;
5) 有效状态下的任务 h 进入节点库所 NP,完成后,h 输出到 SP_3.如果没有发生失效,那么变迁 nt_5 触发后将 s 送入后继服务 $Back$,而 o 和 w 被销毁;
6) 失效状态令牌 l 进入库所 NF_3,按照不同失效类型进入对应恢复期.时间变迁 rt_1 的触发模式{1,2,3}分

© 中国科学院软件研究所 http://www.jos.org.cn

别模拟 I 类、II 类、IV 类失效恢复时间,时间变迁 rt_2 模拟 III 类失效恢复时间,时间延迟都服从指数分布;

7) 在不同失效的发生和恢复期间,节点中变迁的触发是不同的:

7-1) I 类失效发生后,触发变迁 ft_3 的模式 1 将状态令牌 r 从库所 SR 中移出,以抑制变迁 nt_1 模式{1,2,3}的触发,表明不再接受新到的任务.在库所 NF_5 内创建令牌 v,触发变迁 nt_1 模式{4,5,6}.失效令牌 f_1 经过一个时间变迁 rt_1 后进入代表替代服务的库所 $Repl$,完成替代恢复;

7-2) II 类失效发生后,状态令牌$\{c_1,c_2,c_3\}$从库所 SC_1 中移出,以便抑制变迁 nt_1,nt_3 和 nt_4 的触发,表明在失效期间既不接受新到的任务,也要终止已完成任务的结果输出.但已经进入 NP 或 NP'的任务继续执行,等待失效恢复后输出结果.在库所 NF_5 内创建令牌 v,触发变迁 nt_1 模式{4,5,6};

7-3) III 类失效发生后,变迁 ft_3 从 SC_6 中移出令牌 c_4,禁止库所 SP_3 中当前令牌 h 的输出,同时创建令牌 f_3,通过变迁 nt_5 的模式{4,5,6}从 SP_3 中取出第 1 个令牌 h 到库所 NF_6 中,由时间变迁 rt_2 输出到代表重新执行的库所 $Redo$ 中;

7-4) IV 类失效发生后,状态令牌$\{c_1,c_2,c_3\}$从库所 SC_1 中移出,以便抑制变迁 nt_1,nt_3 和 nt_4 的触发,同时 NP(或 NP')将停止工作,所有正在处理的任务将丢失.在库所 NF_5 内创建令牌 v,触发变迁 nt_1 模式{4,5,6}.通过恢复程序由 NP'(或 NP)替换原来的处理环节,以模拟服务节点从零负载开始执行. nt_4 在库所 SC_5 中创建一个令牌 z 用来启动零负载操作,即触发变迁 et_1,移出 SC_2 中的 e_1 和 e_2,在 SC_4 中创建新的令牌$\{e_1,e_2,d\}$用来控制即将启用的 NP'的行为,同时销毁令牌 z.此时,NP'将以零队列方式等待执行新任务.在 NP 恢复期间和替代环节 NP'执行期间,变迁 et_2 通过令牌 d 消耗失效发生时残留在库所 NP 中的令牌 h.因为失效发生的概率一般较低,所以 NP'有效运行的时间相对比较长,可以保证下一次替换进入的库所 NP 是零队列.当 NP'发生失效时,nt_4 在库所 SC_5 中创建令牌 z 用来启动替换操作,即触发变迁 et_2.此时的 NP 以零队列方式开始工作,其执行过程受到库所 SC_3 中 e_1 和 e_2 的制约.SC_3 中的令牌 d 消耗失效发生时残留在库所 NP'中的令牌 h.上述处理可以保证 NP 和 NP'以零队列方式开始恢复后的工作执行过程.

3.3 建模组合服务

在服务链路模型和服务节点模型的基础上,结合服务间存在的各种交互关系,就可以构造出组合服务的执行模型.通过 Petri 网变迁触发机制的设计可以描述服务组合中存在的多种交互关系,如图 4 所示.加粗的带线圆圈表示一个分层的 QPN 服务模型,N_i 和 N_j 表示节点,$L_{i,j}$ 表示 N_i 通向 N_j 的链路,其内部细节分别如图 2 和图 3 所示.

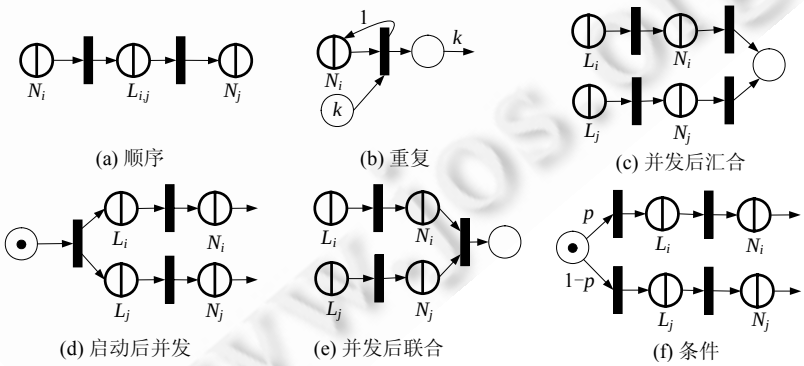


Fig.4 Petri net representations of services node interaction

图 4 服务节点交互关系的 Petri 网表示

图 1 应用场景示例对应的 QPN 模型如图 5 所示.顺序、并发和重复方式比较容易表示,条件方式(S_7+S_8)的

© 中国科学院软件研究所 http://www.jos.org.cn

设计需要考虑前驱服务(S_5或S_6)的松关联导致在后继服务的选择上有了 4 种可能的链路.服务 S_5 完成后待选的链路是 $L_{5,7}$ 和 $L_{5,8}$,服务 S_6 完成后待选的链路是 $L_{6,7}$ 和 $L_{6,8}$.每种情况下的条件选择都通过变迁触发模式权值来实现.变迁 co_1 和 co_2 的触发模式权值和选择概率 p 满足以下关系:

$$p=\frac{w(1)}{w(1)+w(2)} \tag{3}$$

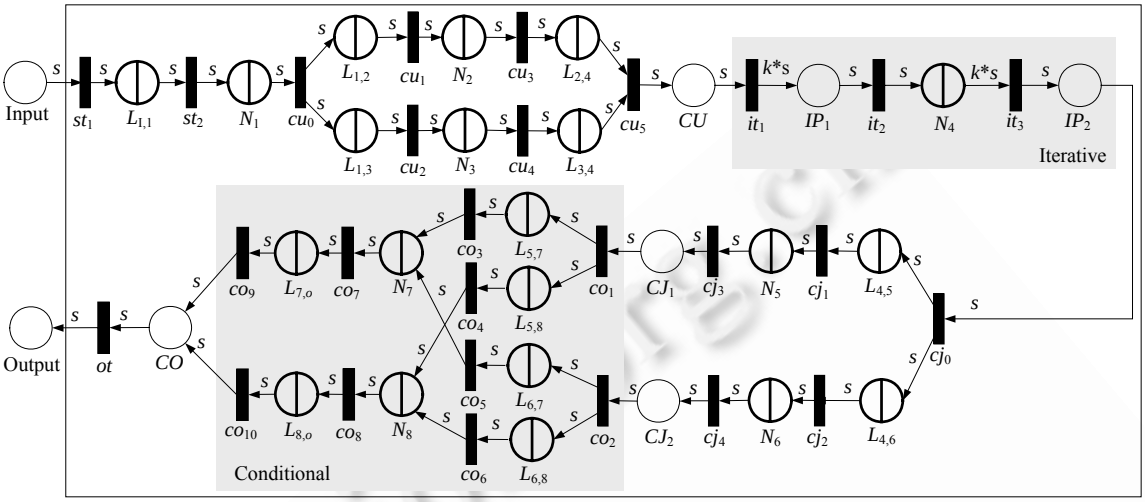


Fig.5 QPN of composite service

图 5 组合服务的排队 Petri 网示例

图 5 中部分变迁描述见表 6.

Table 6 Transition firing modes in composite service

表 6 组合服务的部分变迁触发模式

变迁	触发模式	变迁	触发模式
cu_0	1: $N_1\{s\}\to L_{1,2}\{s\}+L_{1,3}\{s\}$	cj_0	1: $IP_4\{s\}\to L_{4,5}\{s\}+L_{4,6}\{s\}$
cu_5	1: $L_{2,4}\{s\}+L_{3,4}\{s\}\to CU\{s\}$	co_1	1: $CJ_1\{s\}\to L_{5,7}\{s\}$ 2: $CJ_1\{s\}\to L_{5,8}\{s\}$
it_1	1: $CU\{s\}\to IP_1\{k*s\}$		
it_2	1: $IP_1\{s\}\to N_4\{s\}$	co_2	1: $CJ_2\{s\}\to L_{6,7}\{s\}$ 2: $CJ_2\{s\}\to L_{6,8}\{s\}$
it_3	1: $N_5\{k*s\}\to IP_2\{s\}$		

4 实验分析

4.1 实验准备

QPME(queueing Petri net modeling environment)软件包[14,24]是一款基于 QPN 的性能建模工具.Kounev 等人使用 QPME 对 J2EE 软、硬件平台性能测量基准程序 SPECjAppServer2004 进行建模,仿真结果表明,QPME 工具能够很好地反映和预测系统的性能变化[15].

本文使用 QPME 对提出的服务链路模型、服务节点模型和组合服务模型进行仿真,分析在不同的失效概率和恢复时间配置下性能模型的表现.度量属性包括稳态时任务 s 的吞吐率(throughput)X 和平均响应时间(mean response time)T、节点的利用率(utilization)U、任务 s 的丢失率(loss ratio)L 和完成率(accomplish probability)A,仿真数据的置信区间(c.i.)为 95%.丢失率 L 是指任务令牌 s 到达时因为节点失效而未被接受的任务数量占到达任务数目的比率,完成率是指节点接受任务 s 成功完成输出到后继服务(集)的概率.

服务节点仿真场景如图 6 所示,采用开放式队列模型,通过队列库所 $Q(G/M/\infty/IS)$模拟稳定的任务产生源.

© 中国科学院软件研究所 http://www.jos.org.cn

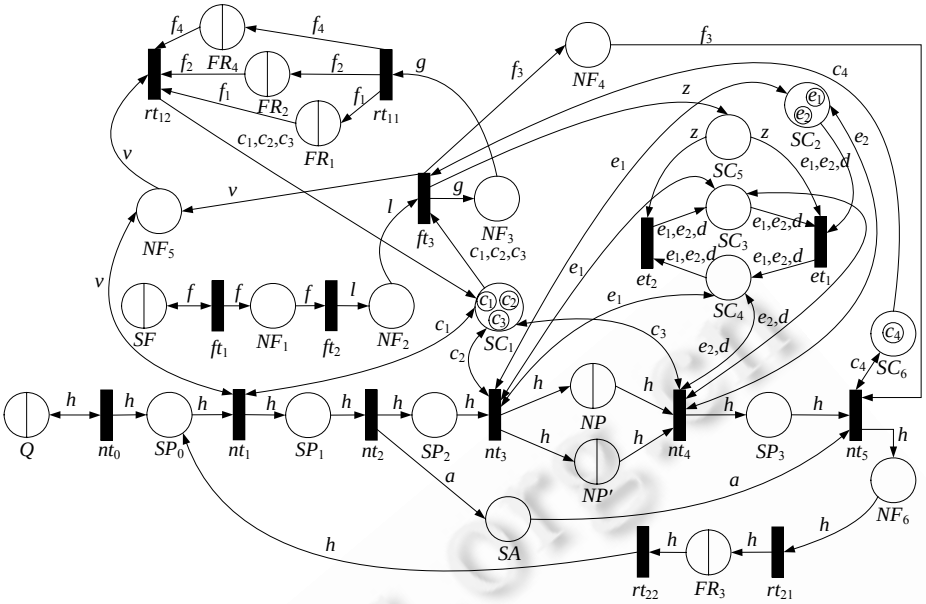


Fig.6 Experimental QPN model of service node

图 6 服务节点的实验性 QPN 模型

图 6 中在图 3 基础上新增的变迁模式见表 7.

Table 7 Firing modes of transitions in experimental node

表 7 实验节点中新增的变迁触发模式

变迁	触发模式	变迁	触发模式
rt_{11}	1: $NF_3\{f_1\}\to FR_1\{f_1\}$ 2: $NF_3\{f_2\}\to FR_2\{f_2\}$ 3: $NF_3\{f_4\}\to FR_4\{f_4\}$	rt_{12}	1: $FR_1\{f_1\}+NF_5\{v\}\to SC_1\{c_1\}+SC_1\{c_2\}+SC_1\{c_3\}$ 2: $FR_2\{f_2\}+NF_5\{v\}\to SC_1\{c_1\}+SC_1\{c_2\}+SC_1\{c_3\}$ 3: $FR_4\{f_4\}+NF_5\{v\}\to SC_1\{c_1\}+SC_1\{c_2\}+SC_1\{c_3\}$
rt_{21}	1: $NF_6\{s\}\to FR_3\{s\}$ 2: $NF_6\{o\}\to FR_3\{o\}$ 3: $NF_6\{w\}\to FR_3\{w\}$	rt_{22}	1: $FR_3\{s\}\to SP_0\{s\}$ 2: $FR_3\{o\}\to SP_0\{o\}$ 3: $FR_3\{w\}\to SP_0\{w\}$
nt_0	1: $Q\{s\}\to SP_0\{s\}+Q\{s\}$ 2: $Q\{o\}\to SP_0\{o\}+Q\{o\}$ 3: $Q\{w\}\to SP_0\{w\}+Q\{w\}$	nt_1	1: $SP_0\{s\}+SC_1\{c_1\}\to SP_1\{s\}+SC_1\{c_1\}$ 2: $SP_0\{o\}+SC_1\{c_1\}\to SP_1\{o\}+SC_1\{c_1\}$ 3: $SP_0\{w\}+SC_1\{c_1\}\to SP_1\{w\}+SC_1\{c_1\}$
nt_2	1: $SP_1\{s\}\to SP_2\{s\}+SA\{a\}$	nt_4	1: $NF_2\{f_1\}+SC_1\{c_1\}+SC_1\{c_2\}+SC_1\{c_3\}\to NF_3\{f_1\}+SC_5\{z\}+NF_5\{v\}$
nt_5	1: $SP_3\{s\}+SC_6\{c_4\}+SA\{a\}\to SC_6\{c_4\}$		

因为 QPME 现有版本还不支持时间变迁的直接使用,所以采用一个队列库所($G/M/1/PS$)和瞬时变迁的组合替换时间变迁的变迁模式.当 I 类失效发生时,采用 IV 类失效恢复的零队列处理方式模拟新服务节点的替换. FR_i 分别模拟 I 类~IV 类失效的恢复时间分布.库所 SA 用来跟踪和描述节点模型的总体执行状况.

服务节点队列库所的初始设置见表 8,其中,任务到达服从泊松分布,服务处理时间和失效恢复时间服从指数分布.失效发生时,失效类型的概率分别为:

- $p_N(1)=5\%$;
- $p_N(2)=45\%$;
- $p_N(3)=36\%$;
- $p_N(4)=14\%$.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 8 Queueing places used in service node QPN model

表 8 服务节点的队列库所属性表

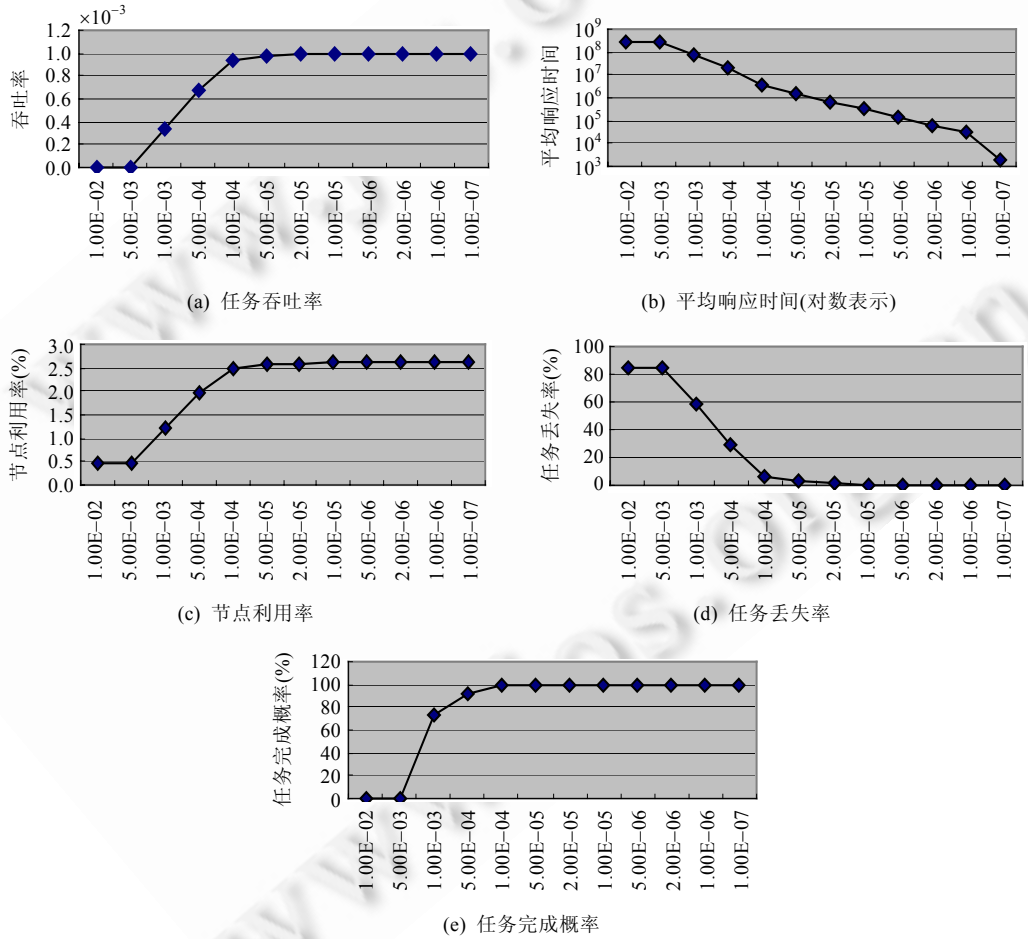
库所	令牌	初始值	队列类型	令牌对应的参数 p_1
Q	$\{s,o,w\}$	$\{1,1,1\}$	$G/M/\infty/IS$	{0.001,0.001,0.001}
NP	$\{s,o,w\}$	$\{0,0,0\}$	$G/M/2/PS$	{0.05,0.05,0.08}
NP'	$\{s,o,w\}$	$\{0,0,0\}$	$G/M/2/PS$	{0.05,0.05,0.08}
SF	$\{f\}$	$\{1\}$	$G/M/\infty/IS$	0.000 02
FR_1	$\{f_1\}$	$\{0\}$	$G/M/1/PS$	0.000 1
FR_2	$\{f_2\}$	$\{0\}$	$G/M/1/PS$	0.007
FR_3	$\{f_3\}$	$\{0\}$	$G/M/1/PS$	0.01
FR_4	$\{f_4\}$	$\{0\}$	$G/M/1/PS$	0.002 5

4.2 服务节点的失效恢复分析

针对失效恢复对服务节点的影响,设计和完成了 3 个实验.

实验 1 分析失效率对服务节点性能的影响.实验方法是保持失效类型的概率分布不变,调整失效发生率,即队列库所 SF 的 p_1,观察服务节点性能的变化趋势.

实验 1 的结果如图 7 所示,其中,每个子图的横轴标示队列库所 SF 的参数 p_1,当 p_1 减小时,失效发生率降低;每个子图的纵轴标示不同的性能参数,响应时间取对数表示.



(a) 任务吞吐率 (b) 平均响应时间(对数表示)

(c) 节点利用率 (d) 任务丢失率

(e) 任务完成概率

Fig.7 Simulation result of Experiment 1

图 7 实验 1 的仿真结果

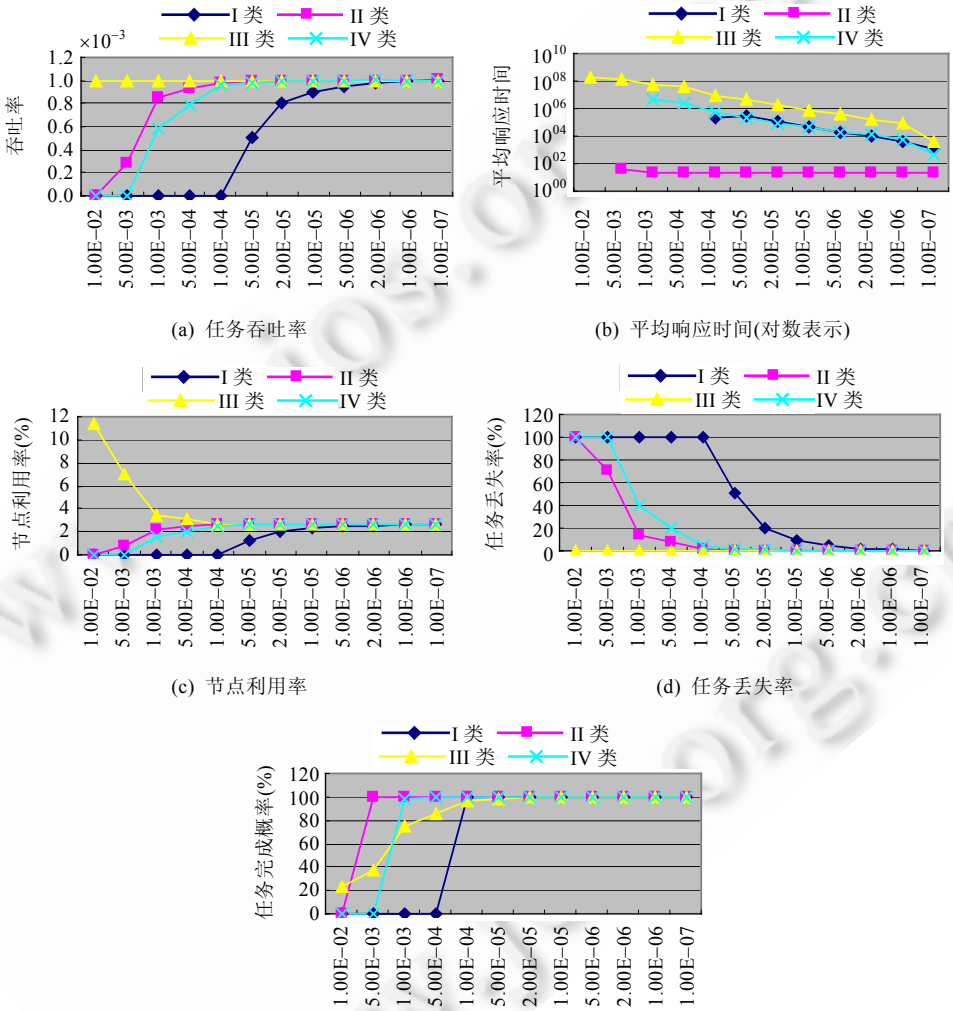
© 中国科学院软件研究所 http://www.jos.org.cn

分析实验结果可知:当 p_1 减小时,失效发生率降低,任务吞吐率增加,响应时间减少,节点利用率提高,任务丢失率减小,任务完成概率提高.

实验结果表明,服务节点模型能够反映各项性能指标随失效发生率发生变化的趋势.

实验 2 分析各类失效对服务节点性能的影响.实验方法是只让某一类失效发生,考察该类失效发生率的变化对性能的影响.具体步骤是:令其中 3 类失效的发生概率为 0,即变迁触发权值为 0,剩下的一类失效概率为 100%,即变迁触发权值为 1,然后调整失效发生率,即队列库所 *SF* 的 p_1,观察服务节点模型性能的变化趋势.

实验 2 的结果如图 8 所示,其中,每个子图中的 4 条曲线分别代表不同类型的失效对服务节点性能的影响;每个子图的横轴标示队列库所 *SF* 的参数 p_1,当 p_1 减小时,失效发生率降低;每个子图的纵轴标示不同的性能参数,响应时间取对数表示.



(a) 任务吞吐率 (b) 平均响应时间(对数表示)

(c) 节点利用率 (d) 任务丢失率

(e) 任务完成概率

Fig.8 Simulation result of Experiment 2

图 8 实验 2 的仿真结果

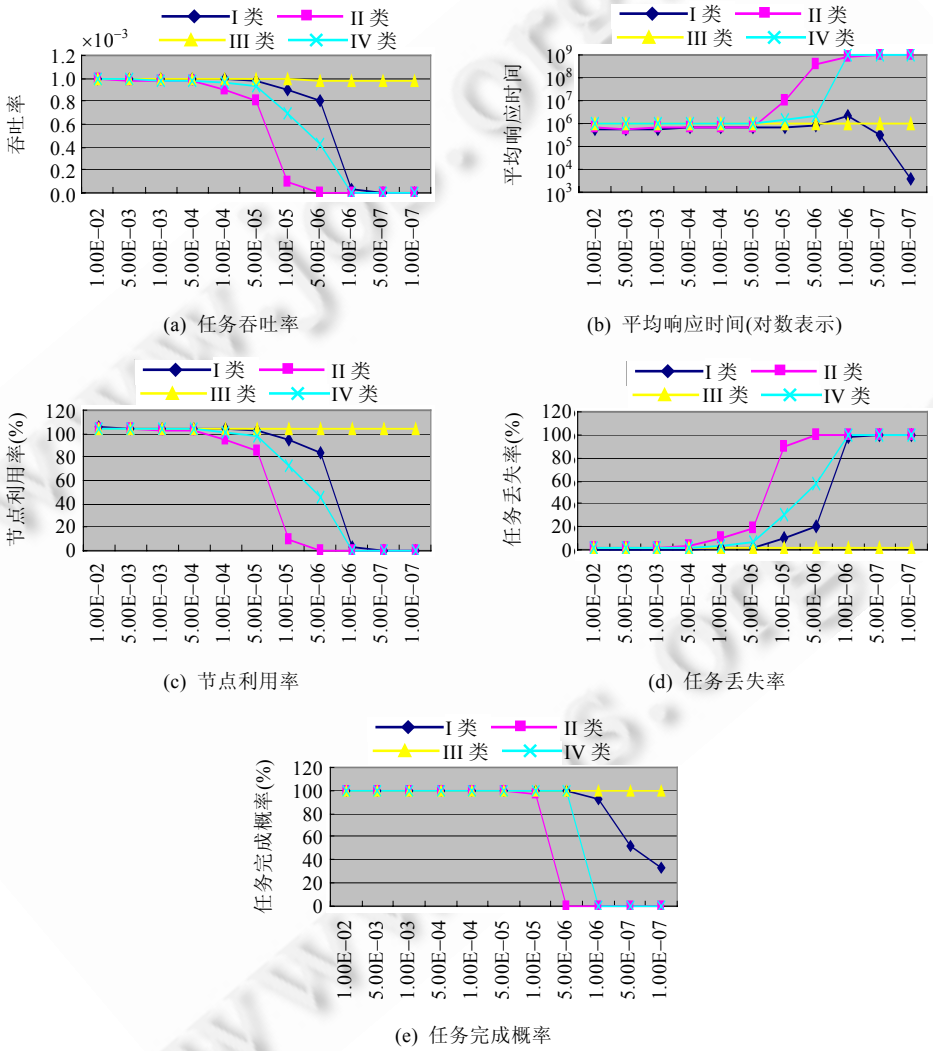
分析实验结果可知:当 p_1 减小时,失效发生率降低.对于 I 类失效,发生率降低后将提升服务节点吞吐率,缩短响应时间,降低任务丢失率,提高任务完成率.对于 II 类失效,在由高发生率下降的初期,吞吐率将快速提升,当发

© 中国科学院软件研究所 http://www.jos.org.cn

生率降低到一定范围内时,吞吐率提升速度变缓并逐渐保持稳定.II 类失效发生率的降低对响应时间的影响不大,但会提高节点利用率,减少任务丢失,提高任务完成的成功概率.对于 III 类失效,重试策略使得进入节点的任务量增大,虽然提高了节点利用率,但是节点队列变长造成等待时间变长,从而延长了任务的响应时间.当失效率降低后,响应时间缩短,节点利用率下降,提高了任务的完成概率,但是不会引起任务丢失,也不会影响任务吞吐率.对于 IV 类失效,当失效发生率降低时,服务节点的吞吐率和利用率提高,响应时间缩短,任务丢失率降低,任务完成概率提高.

实验 3 分析失效恢复时间对服务节点性能的影响.节点模型中队列库所的失效发生率、失效类型概率分布和恢复时间分布函数参数 p_1 取实验 1 中的初始值,然后保持其中 3 类恢复时间不变,调整不同类型的失效恢复时间,即队列库所 FR_i 的 p_1,考察性能参数的变化趋势.

实验 3 的结果如图 9 所示,其中,每个子图中的 4 条曲线分别代表不同类型的失效对服务节点性能的影响;每个子图的横轴标示队列库所 FR_i 的参数 p_1,当 p_1 减小时,失效恢复时间变长;每个子图的纵轴标示不同的性能参数,响应时间取对数表示.



(a) 任务吞吐率

(b) 平均响应时间(对数表示)

(c) 节点利用率

(d) 任务丢失率

(e) 任务完成概率

Fig.9 Simulation result of Experiment 3

图 9 实验 3 的仿真结果

© 中国科学院软件研究所 http://www.jos.org.cn

分析实验结果可知:当参数 p_1 变小时,失效恢复时间变长,节点的有效运行时间变短.通过考察吞吐率变化曲线发现,III 类失效恢复时间的延长对吞吐率影响不大,但是其他类型失效的发生会降低吞吐率.当恢复时间较长时,II 类失效下的吞吐率变化趋势最快,I 类失效下的吞吐率变化趋势最慢.考察响应时间变化曲线,III 类失效下的响应时间始终处于稳定状态.当恢复时间不长时,每类失效下的响应时间波动都不大.当恢复时间较长时,II 类和 IV 类失效下的响应时间变长,但 I 类失效下的响应时间有变短的趋势.分析其原因是,在替换策略下,较长恢复时间使得任务队列长度变短,提高了处理效率.III 类失效恢复时间的延长对节点利用率、丢失率、任务完成概率影响不大,但是另外 3 类失效情况下的利用率逐渐下降,丢失率降低,而 II 类失效情况下的成功率会在一个边界值条件下迅速下降.当失效恢复时间没有达到某个边界值时,I 类和 IV 类失效情况下完成率是一致的.当超过此边界值时,I 类和 IV 类失效情况下的完成率会有所下降,但是由于 I 类失效发生概率较低(5%),所以下降趋势相对平缓.

4.3 组合服务的失效恢复分析

实验 4 对图 1 的组合服务应用场景进行建模和仿真,分析组合服务中不同交互关系对性能的影响.采用开放式队列模型,通过一个 $G/M/\infty/IS$ 队列库所 Q'模拟按照一定速率到达的任务.为了简化模型,假定服务 S_i 驻留在节点 N_i 上,每个节点 N_i 中表示失效发生率、失效类型概率分布和恢复时间分布函数的库所参数 p_1 取实验 1 中的初始值,N_4 的重复次数 k=3,N_7 的条件选择概率 p=30%,且除 N_2 外,所有节点的 I 类失效概率 $p_N(1)=0$,N_r 是 N_2 的替换节点,仿真场景如图 10 所示.

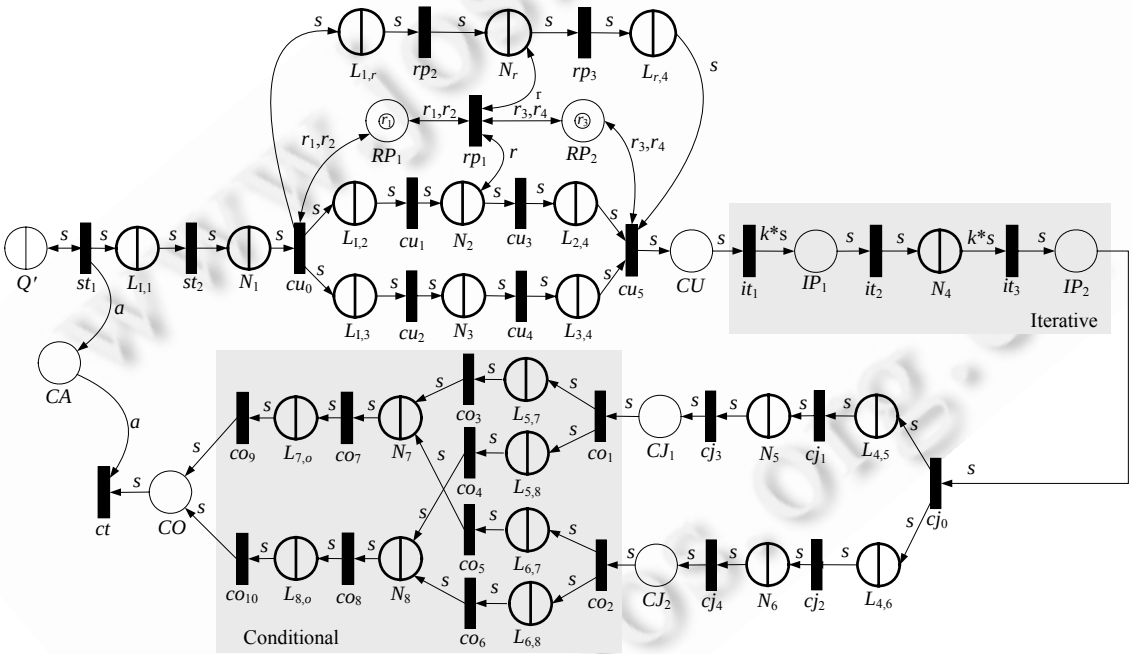


Fig.10 QPN model of Experiment 4

图 10 实验 4 的 QPN 模型

图 10 中增加和修改的变迁模式见表 9.

实验 4 的具体步骤如下:

(1) 调整 N_2 的失效发生率,保持其他节点的参数配置不变,考察并发联合交互关系对组合服务性能的影响;

(2) 调整 N_4 的失效发生率,保持其他节点的参数配置不变,考察重复交互关系对组合服务性能的影响;

© 中国科学院软件研究所 http://www.jos.org.cn

(3) 调整 N_5 的失效发生率,保持其他节点的参数配置不变,考察并发汇合交互关系对性能的影响;

(4) 调整 N_7 的失效发生率,保持其他节点的参数配置不变,考察条件交互关系对性能的影响.

选取组合服务的平均响应时间和任务完成概率作为性能评价指标.

Table 9 Firing modes of transitions in Experiment 4

表 9 实验 4 中增修的变迁触发模式

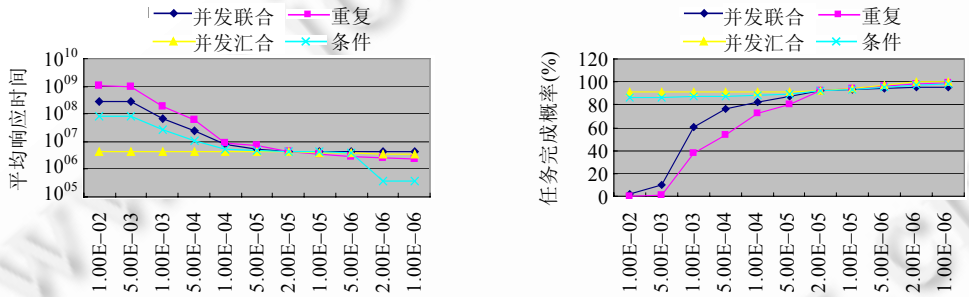
变迁	触发模式
cu_0	1: $N_1\{s\}+RP_1\{r_1\}\rightarrow L_{1,2}\{s\}+L_{1,3}\{s\}+RP_1\{r_1\}$ 2: $N_1\{s\}+RP_1\{r_2\}\rightarrow L_{1,r}\{s\}+L_{1,3}\{s\}+RP_1\{r_2\}$
cu_5	1: $L_{2,4}\{s\}+L_{3,4}\{s\}+RP_2\{r_3\}\rightarrow CU\{s\}+RP_2\{r_3\}$ 2: $L_{r,4}\{s\}+L_{3,4}\{s\}+RP_2\{r_4\}\rightarrow CU\{s\}+RP_2\{r_4\}$
rp_1	1: $N_2\{r\}+RP_1\{r_1\}+RP_2\{r_3\}\rightarrow N_r\{r\}+RP_1\{r_2\}+RP_2\{r_4\}$ 2: $N_r\{r\}+RP_1\{r_2\}+RP_2\{r_4\}\rightarrow N_2\{r\}+RP_1\{r_1\}+RP_2\{r_3\}$
rp_2	1: $L_{1,r}\{s\}\rightarrow N_r\{s\}$
rp_3	1: $N_r\{s\}\rightarrow L_{r,4}\{s\}$
st_1	1: $Q'\{s\}\rightarrow L_{1,1}\{s\}+CA\{a\}+Q'\{s\}$
ct	1: $CO\{s\}+CA\{a\}\rightarrow\{\}$

实验 4 的结果如图 11 所示,其中,

(1) 每个子图中的 4 条曲线分别代表调整不同交互关系的失效发生率时对服务节点性能的影响;

(2) 每个子图的横轴标示队列库所 SF 的参数 p_1,当 p_1 减小时,对应服务节点的失效发生率增加;

(3) 子图的纵轴标示组合服务不同的性能参数,响应时间取对数表示.



(a) 平均响应时间(对数表示) (b) 任务完成概率

Fig.11 Simulation result of Experiment 4

图 11 实验 4 的仿真结果

分析实验结果可知:当服务节点 N_i 中库所 SF 的参数 p_1 减小时,其失效发生率降低,平均响应时间减少,任务完成概率增加.但不同交互关系下,服务节点 N_i 的失效恢复对组合服务整体性能的影响是不同的,具体如下:

1) 在并发联合交互关系下,N_2 的失效发生率变化对组合服务的性能影响分为两个阶段,以 N_2 和 N_3 的库所 SF 的 p_1 都取值 0.000 02 为界.当 N_2 的 p_1 取值大于 0.000 02 时,随着 N_2 的失效率降低,组合服务的平均响应时间减少,任务完成概率提高;当 N_2 的 p_1 取值小于 0.000 02 时,随着 N_2 的失效率的降低,组合服务的平均响应时间变化不大,但任务完成概率有所提高.由此可见,当 N_2 的失效发生率高于 N_3 的失效发生率时,$N_2|N_3$ 的平均响应时间由 N_2 决定,即由最后完成的服务节点决定,而任务完成概率则有 N_2 和 N_3 对共同决定;

2) 在重复关系下,N_4 失效发生率的降低会导致组合服务的平均响应时间大幅减少,任务完成概率快速提高.由此可见,N_4 的重复次数(k=3)放大了其对组合服务性能的影响;

3) 在并发汇合关系下,N_5 的失效发生率变化对组合服务的性能影响也分为两个阶段,以 N_5 和 N_6 的库所

© 中国科学院软件研究所 http://www.jos.org.cn

SF 的 p_1 都取值 0.000 02 为界.当 N_5 的 p_1 取值大于 0.000 02 时,随着 N_5 的失效率的降低,组合服务的平均响应时间和任务完成概率变化都不大;当 N_5 的 p_1 取值小于 0.000 02 时,随着 N_5 的失效率的降低,组合服务的平均响应时间减少,任务完成概率提高.由此可见,当 N_5 的失效发生率低于 N_6 的失效发生率时,$N_5|N_6$ 的平均响应时间和任务完成概率由 N_5 决定,即由最先完成的服务节点决定.由于 N_5 只是组合服务中的一部分,又处于并发汇合交互关系下,所以对组合服务性能影响的变化趋势比较平缓;

4) 在条件关系下,N_7 的失效发生率的降低会导致组合服务的平均响应时间减少,任务完成概率增加,但变化趋势平缓.由此可见,$N_7^{(p)}+S_8^{(1-p)}$ 的条件选择概率 p=30%的取值降低了 N_7 对组合服务性能的影响力.

实验 4 的仿真结果和表 1 的理论公式计算结果在变化趋势上是基本一致的,但在具体比例关系上存在着差异.究其原因,主要是组合服务的多种交互关系并发执行后的相互影响比计算分析所假设的情况更加复杂.由此可见,基于纯数学公式的理论计算方法对组合服务运行过程的分析能力还是有限的,而基于排队 Petri 网的仿真分析方法既能描述组合服务的并发行为,又能定量分析其性能指标,具有明显的优势.限于篇幅,本文只对图 1 所示的组合服务应用场景进行了建模和仿真,但已足以表明基于排队 Petri 网的性能分析模型的有效性.

5 结束语

在不确定的、复杂的各种外部因素的影响下,组合服务不可能永远稳定、可靠地运行.随着服务计算系统的不断演化,组合服务的结构更加复杂,动态性增强,现有的性能分析方法逐渐显出能力上的不足.为此,本文针对分散执行的组合服务运行场景引入失效恢复机制,提出了基于排队 Petri 网的组合服务性能建模与分析方法.该方法根据服务组合的不同交互关系,考虑不同的失效类型和对应的恢复策略对性能的影响.利用排队 Petri 网对服务链路和服务节点进行建模,并依据服务的交互关系生成组合服务的性能评价模型.在此基础上,应用 QPME 1.01 软件包验证了性能模型的定量分析能力.实验结果表明,该方法能够定量分析失效率、恢复时间和交互关系在组合服务性能评价中的影响,具有很好的可行性和有效性.与现有组合服务性能分析方法相比,本文所提方法的优势在于:

1) 从多方面考虑服务组合出现故障的情况,对不同故障造成的失效情况和恢复策略进行了分析,从而清晰地表达了失效恢复影响下的组合服务运行机制;
2) 将失效和恢复处理通过排队 Petri 网直观地加以表示,尤其是对替换策略下服务节点执行过程的描述,从而将对失效的恢复处理转化为对所得到模型的分析,有效地解决了恢复行为的定量分析问题;
3) 使用排队 Petri 网对由链路和节点在不同交互关系(顺序、并发联合、重复、并发汇合和条件)下形成的组合服务进行了建模,能够根据实际需要对服务进行增减,提高了组合过程的灵活性;
4) 与其他性能分析方法相比,能够对不同恢复方法的适用条件进行判断,而且量化的准确性有明显改善,有助于指导不确定网络环境下的信息服务系统失效恢复策略实施方案的设计.

本文关于服务处理时间和失效修复时间服从指数分布以及每个服务节点只提供一类服务的假设具有一定的局限性,在下一步的工作中,我们将继续完善组合服务性能分析模型,细化失效类型和恢复策略的定量分析方法.另外,我们将研究如何优化和控制组合服务的性能,基于所提出的性能分析模型开发相应的原型系统.

致谢 衷心感谢 Kounev 博士提供排队 Petri 网仿真软件 QPME 1.01,给予本文研究工作的帮助.衷心感谢匿名评阅人对本文的工作提出的宝贵意见和建议.

References:

[1] Papazoglou MP, Traverso P, Dustdar S, Leymann F. Service-Oriented computing: State of the art and research challenges. Computer, 2007,40(11):38−45. [doi: 10.1109/MC.2007.400]

© 中国科学院软件研究所 http://www.jos.org.cn

[2] Hyun LS, Ryeol SD. Web service QoS in multi-domain. In: Lee S, *et al*., eds. Proc. of the 10th Int'l Conf. on Advanced Communication Technology (ICACT 2008). Gangwon-Do: IEEE Computer Society, 2008. 1759−1762. [doi: 10.1109/ICACT.2008.4494123]

[3] Teixeira M, Lima R, Oliveira C, Maciel P. Performance evaluation of service-oriented architecture through stochastic Petri nets. In: Proc. of the 2009 IEEE Int'l Conf. on Systems, Man, and Cybernetics. San Antonio: IEEE, 2009. 2831−2836. [doi: 10.1109/ICSMC.2009.5346593]

[4] Sato N. Trivedi KS. Stochastic modeling of composite Web services for closed-form analysis of their performance and reliability bottlenecks. In: Krämer B, Lin KJ, Narasimhan P, eds. Proc. of the ICSOC 2007. LNCS 4749, Berlin, Heidelberg: Springer-Verlag, 2007. 107−118. [doi: 10.1007/978-3-540-74974-5_9]

[5] Zarras A, Vassiliadis P, Issarny V. Model-Driven dependability analysis of Web services. In: Meersman R, Tari Z, eds. Proc. of the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE. LNCS 3291, Berlin, Heidelberg: Springer-Verlag, 2004. 1608−1625. [doi: 10.1007/978-3-540-30469-2_48]

[6] Cao KQ, Gu Q, Ren YX, Chen DX. Reliability and performance analysis of Web service composition based on DTMC. Computer Science, 2009,36(1):179−182, 196 (in Chinese with English abstract).

[7] Cai KY, Bai CG, Zhong XJ. Introduction to reliability models of component-based software system. Journal of Xi'an Jiaotong University, 2003,37(6):551−554 (in Chinese with English abstract).

[8] Xie B, Chen Y, Sun XH, Jin H. Performance under failure of multi-tier Web services. In: Werner B, ed. Proc. of the 2009 15th Int'l Conf. on Parallel and Distributed Systems. Shenzhen: IEEE, 2009. 776−781. [doi: 10.1109/ICPADS.2009.84]

[9] Abudhagir US, Shanmugavel S. Performance optimization of tree structured grid services considering influence of error rate. In: Proc. of the 2009 Int'l Conf. on Information Management and Engineering. IEEE Computer Society, 2009. 471−474. [doi: 10.1109/ICIME.2009.60]

[10] Lu CL. Queuing Theory. 2nd ed., Beijing: Beijing University of Posts and Telecommunications Press, 2009 (in Chinese).

[11] Wang YZ, Lin C, Yang Y, Shan ZG. Research on manageability of grid service model method and management strategies. Chinese Journal of Computers, 2008,31(10):1716−1726 (in Chinese with English abstract).

[12] Lin C, Wang YZ, Yang Y, Qu Y. Research on network dependability analysis methods based on stochastic Petri net. Acta Electronica Sinica, 2006,34(2):322−332 (in Chinese with English abstract).

[13] Bause F. Queueing Petri nets—A formalism for the combined qualitative and quantitative analysis of systems. In: Proc. of the 5th Int'l Workshop on Petri Nets and Performance Models. Toulouse: IEEE, 1993. 14−23. [doi: 10.1109/PNPM.1993.393439]

[14] Kounev S, Buchmann A. SimQPN: A tool and methodology for analyzing queueing Petri net models by means of simulation. Performance Evaluation, 2006,63(4-5):364−394. [doi: 10.1016/j.peva.2005.03.004]

[15] Kounev S. Performance modeling and evaluation of distributed component-based systems using queueing Petri nets. IEEE Trans. on Software Engineering, 2006,32(7):486−502. [doi: 10.1109/TSE.2006.69]

[16] Kounev S, Nou R, Torres J. Autonomic QoS-aware resource management in grid computing using online performance models. In: Glynn P, ed. Proc. of the 2nd Int'l Conf. on Performance Evaluation Methodologies and Tools (VALUETOOLS 2007). New York: ACM, 2007. 23−25.

[17] Nanda MG, Chandra S, Sarkar V. Decentralizing execution of composite Web services. In: Vlissides J, ed. Proc. of the OOPSLA 2004. New York: ACM, 2004. 170−187. [doi: 10.1145/1028976.1028991]

[18] Kang XT, Liu XD, Sun HL, Huang YJ, Zhou C. Improving performance for decentralized execution of composite Web services. In: Bilof R, ed. Proc. of 2010 IEEE the 6th World Congress on Services. Miami: IEEE Computer Society, 2010. 582−589. [doi: 10.1109/SERVICES.2010.38]

[19] Brüning S, Weißleder S, Miroslaw M. A fault taxonomy for service-oriented architecture. In: Cukic B, Dong J, eds. Proc. of the 10th IEEE Symp. on High Assurance Systems Engineering. Dallas: IEEE, 2007. 367−368. [doi: 10.1109/HASE.2007.46]

[20] Tang Y, Jin KY, Zhou K, Song CY. Fault taxonomy method for Web services. Journal of Hu'nan University of Technology, 2009, 23(2):73−76 (in Chinese with English abstract).

[21] Avižienis A, Laprie JC, Randell B, Landwehr C. Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. on Dependable and Secure Computing, 2004,1(1):11−32. [doi: 10.1109/TDSC.2004.2]

[22] Guo SC, Yang B, Huang HZ. Modeling and analysis for grid service reliability considering node recovery. Journal of Xi'an Jiao tong University, 2008,42(6):693−697, 790 (in Chinese with English abstract).

[23] Wu M, Sun XH, Jin H. Performance under failure of high-end computing. In: Verastegui B, ed. Proc. of the ACM/IEEE Super Computing Conf. 2007 (SC 2007). New York: ACM, 2007. 1−11. [doi: 10.1145/1362622.1362687]

© 中国科学院软件研究所 http://www.jos.org.cn

[24] Kounev S, Dutz C, Buchmann A. QPME: Queueing Petri net modeling environment. In: Werner B, ed. Proc. of the 3rd Int'l Conf. on Quantitative Evaluation of SysTems (QEST). Riverside: IEEE, 2006. 115−116. [doi: 10.1109/QEST.2006.44]

附中文参考文献:

[6] 曹科强,顾庆,任颖新,陈道蓄.服务组合中基于 DTMC 的可靠性和性能分析.计算机科学,2009,10,36(10):179−182,196.

[7] 蔡开元,白成刚,钟小军.构件软件系统的可靠性评估模型简介.西安交通大学学报,2003,37(6):551−554.

[10] 陆传赉.排队论.第 2 版,北京:北京邮电大学出版社, 2009.

[11] 王元卓,林闯,杨扬,单志广.网格服务可管理性模型及策略研究.计算机学报,2008,31(10):1716−1726.

[12] 林闯,王元卓,杨扬,曲扬.基于随机 Petri 网的网络可信赖性分析方法研究.电子学报,2006,34(2):322−332.

[20] 唐渊,金可音,周昆,宋春阳.Web 服务失败分类法.湖南工业大学学报,2009,23(2):73−76.

[22] 郭夙昌,杨波,黄洪钟.考虑节点失效恢复能力的网格服务可靠性建模与分析.西安交通大学学报,2008,42(6):693−697,790.

附录

(1) 排队 Petri 网的基本概念

QPN 可以形式化定义为一个八元组$(P,T,C,\Gamma,I^{+},M_0,Q,W)$,其中,

① P 是非空有限库所(place)集合$(p_1,p_2,\ldots,p_{|P|})$;

② T 是非空有限变迁(transition)集合$(t_1,t_2,\ldots,t_{|T|})$;

③ $P\cap T=\varnothing$;

④ C 为颜色函数, $C:P\cup T\rightarrow\sum C$ 把每个库所 p 都映射到一个颜色集 $C(p)$,把每个变迁 t 都映射到一个颜色集 $C(t)$.也就是说,p 和 t 中的每个令牌(token)都属于颜色类型;

⑤ Γ,I^{+}分别是 $P\times T$ 上的后向和前向关联函数,使得对所有$(p,t)\in P\times T$ 有

$$I^{-},(p,t),I^{+},(p,t):C(t)\rightarrow C(p)_{MS};$$

⑥ M_0 称为 CPN 的初始标识,对于所有 $p\in P$ 都有 $M_0(p)\in C(p)_{MS}$;

⑦ $Q=(\tilde{Q}_1,\tilde{Q}_2,(q_1,q_2,...,q_{|P|}))$, 其中,

- $\tilde{Q}_1\subseteq P$ 表示时间队列库所;
- $\tilde{Q}_2\subseteq P$ 表示瞬时队列库所,且 $\tilde{Q}_1\cap\tilde{Q}_2=\varnothing$;
- 如果 p_i 是队列库所,那么 q_i 表示具有 $C(p_i)$中所有颜色的队列;如果 p_i 是普通库所,那么 q_i 等价于关键字 *null*;

⑧ $W=(\tilde{W}_1,\tilde{W}_2,(w_1,w_2,...,w_{|T|}))$, 其中,

- $\tilde{W}_1\subseteq T$ 表示时间变迁;
- $\tilde{W}_2\subseteq T$ 表示瞬时变迁,且 $T=\tilde{W}_1\cup\tilde{W}_2,\tilde{W}_1\cap\tilde{W}_2=\varnothing$;
- $w_i\in[C(t_i)\mapsto\Re^{+}]$使得对于所有的 $c\in C(t_i)$有:如果 t_i 是时间变迁,那么 $w_i(c)$描述了与颜色 c 有关的描述变迁触发时延的概率分布函数;如果 t_i 是瞬时变迁,那么 $w_i(c)$描述了与颜色 c 有关的描述变迁触发频率的权值.

QPN 的结构元素包括队列库所(queueing place)、普通库所(ordinary place)、时间变迁(timed transition)、瞬时变迁(immediate transition)、弧(arc)和颜色令牌(color token).队列库所描述具有随机服务特征的业务处理行为,包括队列(queue)和存储库(depository).普通库所描述可能的系统局部状态(条件或状况);时间变迁描述具有时延特征的改变系统状态的事件;瞬时变迁描述系统状态改变的事件;弧用于规定局部状态和事件之间的关系;颜色令牌用于描述不同类型的服务实例.

(2) 服务节点的排队 Petri 网表示中的变迁描述

图 3 中的变迁描述见表 10 和表 11.符号 $A\{x\}\rightarrow\{\}$表示令牌 x 在变迁中由库所 A 迁出后将被销毁.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 10 Firing modes of transitions (rt_i, ft_i, et_i) in service node

表 10 服务节点中变迁 rt_i,ft_i 和 et_i 的触发模式

变迁	触发模式	变迁	触发模式
rt_1	1: $NF_3\{f_1\}+NF_5\{v\}\rightarrow Repl\{r\}$ 2: $NF_3\{f_2\}+NF_5\{v\}\rightarrow SC_1\{c_1\}+SC_1\{c_2\}+SC_1\{c_3\}$ 3: $NF_3\{f_4\}+NF_5\{v\}\rightarrow SC_1\{c_1\}+SC_1\{c_2\}+SC_1\{c_3\}$	rt_2	1: $NF_6\{s\}\rightarrow Redo\{s\}$ 2: $NF_6\{o\}\rightarrow Redo\{o\}$ 3: $NF_6\{w\}\rightarrow Redo\{w\}$
ft_2	1: $NF_1\{f\}\rightarrow NF_2\{f_1\}$ 2: $NF_1\{f\}\rightarrow NF_2\{f_2\}$ 3: $NF_1\{f\}\rightarrow NF_2\{f_3\}$ 4: $NF_1\{f\}\rightarrow NF_2\{f_4\}$	ft_3	1: $NF_2\{f_1\}+SR\{r\}\rightarrow NF_3\{f_1\}+NF_5\{v\}$ 2: $NF_2\{f_2\}+SC_1\{c_1\}+SC_1\{c_2\}+SC_1\{c_3\}$ $\rightarrow NF_3\{f_2\}+NF_5\{v\}$ 3: $NF_2\{f_3\}+SC_6\{c_4\}\rightarrow NF_4\{f_3\}$ 4: $NF_2\{f_4\}+SC_1\{c_1\}+SC_1\{c_2\}+SC_1\{c_3\}$ $\rightarrow NF_3\{f_4\}+SC_5\{z\}+NF_5\{v\}$
et_1	1: $SC_5\{z\}+SC_2\{e_1\}+SC_2\{e_2\}$ $\rightarrow SC_4\{e_1\}+SC_4\{e_2\}+SC_4\{d\}$ 2: $SC_5\{z\}+SC_3\{e_1\}+SC_3\{e_2\}+SC_3\{d\}$ $\rightarrow SC_4\{e_1\}+SC_4\{e_2\}+SC_4\{d\}$	et_2	1: $SC_5\{z\}+SC_4\{e_1\}+SC_4\{e_2\}+SC_4\{d\}$ $\rightarrow SC_3\{e_1\}+SC_3\{e_2\}+SC_3\{d\}$
ft_1	1: $SF\{f\}\rightarrow NF_1\{f\}+SF\{f\}$		

Table 11 Firing modes of transitions nt_i in service node

表 11 服务节点中变迁 nt_i 的触发模式

变迁	触发模式	变迁	触发模式
nt_0	1: $Fork\{s\}\rightarrow SP_0\{s\}$ 2: $OW\{o\}\rightarrow SP_0\{o\}$ 3: $EW\{w\}\rightarrow SP_0\{w\}$ 4: $Fork\{r\}\rightarrow SP_0\{r\}$	nt_1	1: $SP_0\{s\}+SC_1\{c_1\}+SR\{r\}\rightarrow SP_1\{s\}+SC_1\{c_1\}+SR\{r\}$ 2: $SP_0\{o\}+SC_1\{c_1\}+SR\{r\}\rightarrow SP_1\{o\}+SC_1\{c_1\}+SR\{r\}$ 3: $SP_0\{w\}+SC_1\{c_1\}+SR\{r\}\rightarrow SP_1\{w\}+SC_1\{c_1\}+SR\{r\}$ 4: $SP_0\{s\}+NF_5\{v\}\rightarrow NF_5\{v\}$ 5: $SP_0\{o\}+NF_5\{v\}\rightarrow NF_5\{v\}$ 6: $SP_0\{w\}+NF_5\{v\}\rightarrow NF_5\{v\}$ 7: $SP_0\{r\}\rightarrow SR\{r\}$
nt_2	1: $SP_1\{s\}\rightarrow SP_2\{s\}$ 2: $SP_1\{o\}\rightarrow SP_2\{o\}$ 3: $SP_1\{w\}\rightarrow SP_2\{w\}$		
nt_5	1: $SP_3\{s\}+SC_6\{c_4\}\rightarrow Back\{s\}+SC_6\{c_4\}$ 2: $SP_3\{o\}+SC_6\{c_4\}\rightarrow SC_6\{c_4\}$ 3: $SP_3\{w\}+SC_6\{c_4\}\rightarrow SC_6\{c_4\}$ 4: $NF_4\{f\}+SP_3\{s\}\rightarrow NF_6\{s\}$ 5: $NF_4\{f\}+SP_3\{o\}\rightarrow NF_6\{o\}$ 6: $NF_4\{f\}+SP_3\{w\}\rightarrow NF_6\{w\}$	nt_4	1: $NP\{s\}+SC_1\{c_3\}+SC_2\{e_2\}\rightarrow SP_3\{s\}+SC_1\{c_3\}+SC_2\{e_2\}$ 2: $NP\{w\}+SC_1\{c_3\}+SC_2\{e_2\}\rightarrow SP_3\{w\}+SC_1\{c_3\}+SC_2\{e_2\}$ 3: $NP\{o\}+SC_1\{c_3\}+SC_2\{e_2\}\rightarrow SP_3\{o\}+SC_1\{c_3\}+SC_2\{e_2\}$ 4: $NP\{s\}+SC_1\{c_3\}+SC_3\{e_2\}\rightarrow SP_3\{s\}+SC_1\{c_3\}+SC_3\{e_2\}$ 5: $NP\{w\}+SC_1\{c_3\}+SC_3\{e_2\}\rightarrow SP_3\{w\}+SC_1\{c_3\}+SC_3\{e_2\}$ 6: $NP\{o\}+SC_1\{c_3\}+SC_3\{e_2\}\rightarrow SP_3\{o\}+SC_1\{c_3\}+SC_3\{e_2\}$ 7: $NP'\{s\}+SC_1\{c_3\}+SC_4\{e_2\}\rightarrow SP_4\{s\}+SC_1\{c_3\}+SC_4\{e_2\}$ 8: $NP'\{w\}+SC_1\{c_3\}+SC_4\{e_2\}\rightarrow SP_3\{w\}+SC_1\{c_3\}+SC_4\{e_2\}$ 9: $NP'\{o\}+SC_1\{c_3\}+SC_4\{e_2\}\rightarrow SP_3\{o\}+SC_1\{c_3\}+SC_4\{e_2\}$ 10: $NP\{s\}+SC_4\{d\}\rightarrow SC_4\{d\}$ 11: $NP\{w\}+SC_4\{d\}\rightarrow SC_4\{d\}$ 12: $NP\{o\}+SC_4\{d\}\rightarrow SC_4\{d\}$ 13: $NP'\{s\}+SC_3\{d\}\rightarrow SC_3\{d\}$ 14: $NP'\{w\}+SC_3\{d\}\rightarrow SC_3\{d\}$ 15: $NP'\{o\}+SC_3\{d\}\rightarrow SC_3\{d\}$
nt_3	1:$SP_2\{s\}+SC_1\{c_2\}+SC_2\{e_1\}\rightarrow NP\{s\}+SC_1\{c_2\}+SC_2\{e_1\}$ 2:$SP_2\{w\}+SC_1\{c_2\}+SC_2\{e_1\}\rightarrow NP\{w\}+SC_1\{c_2\}+SC_2\{e_1\}$ 3: $SP_2\{o\}+SC_1\{c_2\}+SC_2\{e_1\}\rightarrow NP\{o\}+SC_1\{c_2\}+SC_2\{e_1\}$ 4: $SP_2\{s\}+SC_1\{c_2\}+SC_3\{e_1\}\rightarrow NP\{s\}+SC_1\{c_2\}+SC_3\{e_1\}$ 5: $SP_2\{w\}+SC_1\{c_2\}+SC_3\{e_1\}\rightarrow NP\{w\}+SC_1\{c_2\}+SC_3\{e_1\}$ 6: $SP_2\{o\}+SC_1\{c_2\}+SC_3\{e_1\}\rightarrow NP\{o\}+SC_1\{c_2\}+SC_3\{e_1\}$ 7: $SP_2\{s\}+SC_1\{c_2\}+SC_4\{e_1\}\rightarrow NP'\{s\}+SC_1\{c_2\}+SC_4\{e_1\}$ 8: $SP_2\{w\}+SC_1\{c_2\}+SC_4\{e_1\}$ $\rightarrow NP'\{w\}+SC_1\{c_2\}+SC_4\{e_1\}$ 9: $SP_2\{o\}+SC_1\{c_2\}+SC_4\{e_1\}\rightarrow NP'\{o\}+SC_1\{c_2\}+SC_4\{e_1\}$		

顾军(1977－),男,江苏大丰人,博士生,讲师,CCF 会员,主要研究领域为网络与服务计算.

E-mail: jgu@seu.edu.cn

曹玖新(1967－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络安全,服务计算,社会计算,虚拟技术.

E-mail: Jx.cao@seu.edu.cn

罗军舟(1960－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为下一代网络体系结构,协议工程,网络安全,云计算,无线局域网.

E-mail: jluo@seu.edu.cn

李伟(1978－),男,博士,副教授,CCF 高级会员,主要研究领域为下一代网络体系结构,服务计算.

E-mail: xchlw@seu.edu.cn

© 中国科学院软件研究所 http://www.jos.org.cn