软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(10):2373−2384 [doi: 10.13328/j.cnki.jos.004501]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种基于 DAG 动态重构的认知网络服务迁移方法*

林俊宇, 王慧强, 马春光, 卢 旭, 吕宏武

(哈尔滨工程大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

通讯作者: 林俊宇, E-mail: linjunyu@hrbeu.edu.cn

摘 要: 针对认知网络高度动态性带来的服务随机失效问题,提出了一种服务迁移方法以保障认知网络的 QoS.首先,采用先迁移、后优化的思想,重新生成关联服务有向无环图(directed acyclic graph,简称 DAG),并在此基础上提出 DAG 动态重构算法,将关联服务转化为层次化 DAG 服务;其次,计算关键服务迁移路径,并给出可迁移服务死锁避免理论分析,将迁移服务提前迁移到当前网络空闲资源运行,以缩短服务的执行时间.仿真实验测试了 3 种故障注入类型下网络服务迁移方案的服务性能.实验结果显示,该方法在弹性网络负载与未知故障情况下具有较好的 QoS 保障效果.

关键词: 认知网络;QoS;服务迁移;有向无环图;随机失效

中图法分类号: TP393

中文引用格式: 林俊宇,王慧强,马春光,卢旭,吕宏武.一种基于 DAG 动态重构的认知网络服务迁移方法.软件学报,2014,25(10):2373−2384. http://www.jos.org.cn/1000-9825/4501.htm

英文引用格式: Lin JY, Wang HQ, Ma CG, Lu X, Lü HW. Service migration method for cognitive network based on DAG dynamic reconstruction. Ruan Jian Xue Bao/Journal of Software, 2014,25(10):2373−2384 (in Chinese). http://www.jos.org.cn/1000-9825/4501.htm

Service Migration Method for Cognitive Network Based on DAG Dynamic Reconstruction

LIN Jun-Yu, WANG Hui-Qiang, MA Chun-Guang, LU Xu, LÜ Hong-Wu

(College of Computer Science and Technology, Harbin Engineer University, Harbin 150001, China)

Corresponding author: LIN Jun-Yu, E-mail: linjunyu@hrbeu.edu.cn

Abstract: According to randomness of service failure for high dynamicity of cognitive networks, a service migration method is proposed to ensure QoS of cognitive networks. Firstly, with the principle of optimization-after-migration, the directed acyclic graph (DAG) of correlated service is regenerated according to the proposed DAG dynamic reconstruction algorithm to transform the correlated service to layered DAG service. Secondly, the critical service migration route is computed and the analysis of migration service deadlock avoidance is provided. By migrating critical service to current idle resources, service execution time can be reduced markedly. Finally, simulation experiments are conducted to test the service speedup performance of both service migration method and waiting-recovery method with three kinds of faults injected. The experiment results show that service migration method can achieve better QoS assurance quality under the flexible network load and unknown fault injection.

Key words: cognitive network; QoS; service migration; directed acyclic graph; randomness of failure

认知网络(cognitive network,简称 CN)[1]是受认知无线电(cognitive radio,简称 CR)技术启发,在 2000 年后才提出来的新兴领域,其核心思想是:使网络能够感知内外环境变化,实时调整网络的配置,动态、智能地适应外界环境变化,并指导网络的自主决策[2].由于其具有更高的智能性和自主性,在端到端性能和 QoS 保障等领域表现

* 基金项目: 国家自然科学基金(61370212, 60973027, 61402127); 高等学校博士学科点专项科研基金(20122304130002, 20102304120012); 中央高校基本科研业务费专项资金(HEUCF100601, HEUCFZ1213); 黑龙江省自然科学基金(ZD201102); 黑龙江省教育厅科学技术研究资助项目(12513053)

收稿时间: 2012-10-22; 修改时间: 2013-03-22, 2013-06-18; 定稿时间: 2013-09-09

出了巨大的优势,已成为未来网络发展的重要趋势.然而,由于认知网络是一个新兴领域,目前对其 QoS 保障的研究尚处于初始阶段.由于网络节点接入的随机性和负载高度的动态性等特点,对认知网络实时交互式流媒体业务的 QoS 保障还存在巨大挑战[3].

目前,认知网络 QoS 保障的研究主要可以归结为两种思路:

一种思路仍沿用传统互联网络的 QoS 体系,例如,通过对 IntServ[4]和 DiffServ[5]的改进与融合来满足用户需求.IntServ 需要在端到端传输路径上的每个节点为每一信息流建立并维持资源预留,在高度动态性和海量业务流的前提下,导致连接建立和连接释放阶段的额外开销激增;DiffServ 面向 QoS 参数加以区分应用,相比 IntServ 简化了信令,但由于 DiffServ 结构中网络和端网络之间缺乏信令通信,不能提供端到端的 QoS 保障,难以满足未来网络智能化的需求.

第 2 种思路则是建立具有自适应能力的 QoS 保障机制[6].在这些研究中,基于跨层感知的网络体系结构设计是一种重要的方法.Thomas 等人[2]提出采用跨层设计对认知网络进行改进,通过打破网络层间的壁垒,使不相邻的网络层次之间能够直接通信,进而综合多个网络层信息以提升认知网络的端到端 QoS.然而,Thomas 等人的方案仅提出了一个包含行为层、功能层和物理层的认知网络框架,还缺少具体实现的方法.在此基础上,Ali 等人[7]针对不断变化地接入信道的认知网络,采用部分可观察马尔可夫过程对跨层 QoS 进行了分析,为多媒体流传输质量的改进提供了参考.而 Chen 等人[8]提出一种采用了跨层设计的 QoS 自适应算法,通过数据平面、认知平面和知识平面的协作,实现认知网络 QoS 的调节.但是,基于跨层设计的方法一方面需要对现有网络协议栈结构进行改进,满足网络层次跨层感知的需求,这极大地限制了其适用范围;另一方面,跨层设计一般只能基于无线网络,这对于大量有线网络设施并不适合.因此,针对跨层感知的局限性,许多研究者尝试从系统整体上对 QoS 架构进行设计.Chang 和 Hsieh 等人[9]提出了一种面向 Internet 服务质量保障的自适应框架,并提出了一种同步多媒体统一语言 SMIL,能够保证在不干扰媒体流的情况下实现回滚.Attar 等人[10]提出一种 QoS 保障的框架,采用 NASH 讨价还价的方法解决该用户的 QoS 保障.而 Swami 在文献[11]中提出采用图论的方法保证认知网络的 QoS.在国内,辛明军等人[12]以代理技术为实现手段,提出一种复合模型协作求解的自适应 QoS 体系结构,以提高协同计算环境分布式问题协作求解的运行效率和服务水平.

然而,纵观这些研究都未充分考虑认知网络节点频繁配置、频带动态调整和节点随机接入等特性所带来的网络服务失效问题,这对网络服务调整后的 QoS 产生了巨大的隐患.针对该问题,本文以不终止服务执行条件下的 QoS 保障为目的,参照并改进现有的任务迁移技术[13,14],提出一种基于 DAG 动态重构的认知网络服务迁移机制,并重点从理论上证明具有依赖关系的服务可迁移性判定以及迁移服务的死锁避免.服务迁移机制能够直接应用于现有的网络基础设施,对于提升认知网络的服务保障能力具有重要的现实意义.

本文第 1 节描述认知网络的服务前移问题,即,对服务前移实例进行定义,对关联服务和迁移服务如何避免死锁问题进行描述.第 2 节介绍关键服务的动态迁移,包括对服务执行时间计算、DAG 动态重构和服务可迁移性判定的介绍.第 3 节给出仿真实验与分析.第 4 节介绍相关工作.最后,第 5 节对全文进行总结.

1 认知网络的服务迁移问题描述

认知网络是一种以认知计算为核心的网络形态,其本质特征是网络节点和(或)全局具有反馈控制的认知环反馈控制结构 MDE[2],如图 1 所示.其中,监测器 M 监测环境信息,通过决策器 D 做出决策,并将制定的计划交由执行器 E 执行,而执行的结果将再次反馈给监测器 M.认知网络的智能性和自主性使得网络结构表现出高度的动态性:一方面,节点可以任意地接入和退出,任务目标可以随时更改;另一方面,系统为追求最优服务,将根据环境和任务需求的变化,通过网络,进行包括信道切换、路由更改和服务重配置在内的各种动态自适应.如果仍然沿用现有的 QoS 保障技术,将不可避免地带来网络的服务失效.

针对该问题,本文提出采用服务迁移的思想,将发生随机失效或资源不满足条件的网络节点上的服务迁移到其他节点.同时,认知网络的自主性为失效服务由一个节点迁移到其他节点提供了迁移路径的计算能力和状态暂存能力,为服务迁移的实现奠定了基础.

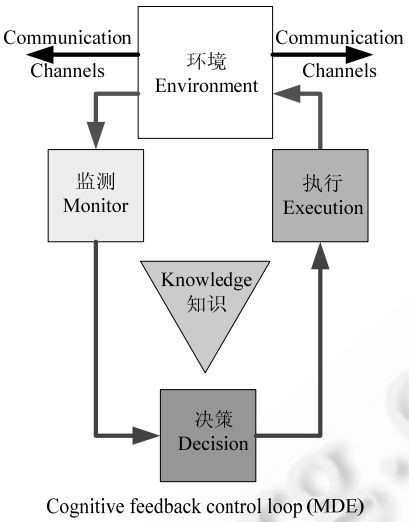

Fig.1 Cognitive feedback control loop structure

图 1 认知反馈控制环结构

1.1 服务迁移实例定义

与传统的服务迁移方法不同,认知网络中的失效服务可迁移是指在节点发生失效情况之后,指定原节点上的服务迁移路径及其时间属性.服务可迁移的目标是:在不破坏节点间偏序关系且不产生死锁的情况下,使得迁移后的服务执行期望最大化,本方法的目标是使服务执行时间最短.当服务被创建即处于初始态,直到网络根据首次服务迁移方案为其分配一个工作节点,此时,服务将在工作节点的服务队列中排队等待执行,即处于就绪态.在服务执行过程中,由于工作节点失效等原因,导致服务无法按照规定执行,则将服务迁移至其他工作节点执行;而当计算过程结束或者取消后,服务执行终止.根据迁移服务生命周期不同阶段的特点,可以给出服务迁移实例定义.

定义 1(服务迁移实例). 一个服务迁移实例由五元组(*Mid*,*r*,*t*,*Node*,*p*)组成,执行一个目标相对独立的计算服务.其中,*Mid* 是可认证的迁移实例标识,*r* 是迁移实例的服务说明,*t* 是迁移实例当前正在执行的服务,*Node* 是分配的允许迁移的工作节点集合,*p* 是迁移实例当前所处的工作节点.

从抽象角度来看,本文迁移实例的服务完成过程经历以下步骤:

步骤 1. 根据初始 DAG 服务分配方案,不同的服务迁移实例进入不同的工作节点服务队列等待并执行.

步骤 2. 检查当前服务的执行状态,当发现节点失效无法完成服务执行时,暂停当前正在执行的服务,并获取各个子服务的执行情况;对于服务迁移实例的创建,均通过当前工作位置生成服务说明,包括迁移实例的服务集合、流程控制逻辑和数据存储,并搜索所有工作位置已经注册的服务[15].

步骤 3. 对 DAG 服务进行分层重构,并计算迁移路径,向迁移位置发送迁移请求,目标位置受到请求后,依据自身状态,做出允许或暂缓迁移应答.接收到允许迁移应答后,进行迁移并告知新的迁移位置,并在新的工作节点进行服务注册,激活挂起的迁移服务.在成功迁移后,原工作节点删除迁移服务的备份,同时释放迁移服务占用的计算资源.

图 2 描述了认知网络关键服务执行与迁移过程,由于服务迁移实例的注册、创建和状态保存等均可利用现有技术予以解决,因此本文的工作主要侧重于服务迁移的关键环节,即,具有依赖关系的服务可迁移性判定以及迁移服务的死锁避免.

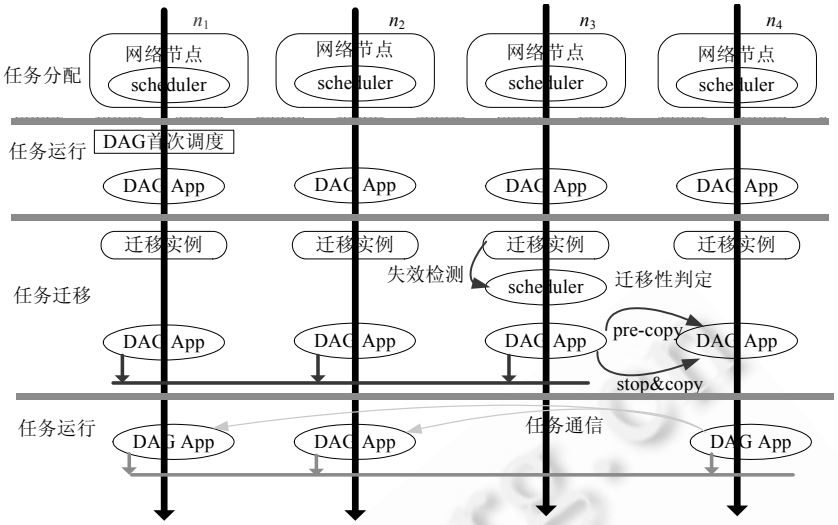

Fig.2 Execution and the migration process of the key services of cognitive network

图 2 认知网络关键服务迁移流程

1.2 关联服务描述

可迁移服务集 M 可由一个有向无环图 DAG 来表示,定义为 $M=(H,U)$,其中,$H=\{H_i|i=1,2,\ldots,n\}$,代表网络 n 个服务的集合;$U_{ij}=\{(H_i,H_j)|H_i,H_j\in H,i<j\}$,$|U|=e$,表示拥有 e 条边的有向边集合.DAG 中的每一个节点代表一个服务,是服务迁移中的最小单位.DAG 服务节点 H_i 的权重为计算成本,记作 $W(H_i)$,U_{ij} 表示服务之间存在的时间上的依赖关系.大部分计算密集型应用均在计算前预装数据,相对于计算开销来说,通信开销可以忽略不计.因此,本文不考虑 DAG 执行中的通信开销问题.在本文的 DAG 模型中,网络节点采用空间共享机制,DAG 中的每一个节点代表一个计算子服务.假设认知网络系统中存在 m 个节点 $Node_i$,$i=0,1,\ldots,m-1$,n 个子服务 H_j,$j=0,1,\ldots,n-1$.为了实现工作流失效可恢复的合理迁移,每一个子服务均分配给一个网络节点,并采用以下 3 个随机变量来描述计算服务 H_j 的执行情况,即,服务执行时间 T_j^C 、服务开始时间 T_j^S 和服务结束时间 T_j^F,且满足 $T_j^F=T_j^S+T_j^C$.

定义 2(死锁 DAG 服务). 若 DAG 服务集 V 中存在非空的服务子集 $D\subseteq V$,D 中每一个服务的先序子服务均包含在该子集中,即:

$$D\subseteq V=\{pre(t_1)\cup pre(t_2)\cup\ldots\cup pre(t_i),t_1,\ldots,t_i\in D\},$$

则该 DAG 服务为死锁 DAG 服务.

定义 3(空闲资源). 设服务 h_i 和 h_j 被分配至工作节点 $Node_i$ 上执行且 h_i 先于 h_j,若 h_i 与 h_j 之间不存在其他服务,则定义服务 h_j 的服务开始时间 T_j^S 与服务 h_i 的服务结束时间 T_i^F 之差 $\Delta=T_j^S-T_i^F$,为空闲时段,该工作节点也称为空闲资源.

当一个工作节点为空闲资源,并不表示节点在任何时刻均能接受迁移服务,而仅仅表示该工作节点存在服务执行的空闲期,具备服务迁移的基本条件.空闲资源存在如下 3 种情况:① 工作节点 $Node_i$ 已被分配服务,但尚未开始执行;② 工作节点 $Node_i$ 已经执行完当前服务,并等待下一个服务开始执行;③ 工作节点 $Node_i$ 所有已分配服务已经执行完毕.

定义 4(关键路径). 若服务 h_i 的前序节点集合为 $PN(h_i)$,$h_j\in PN(h_i)$,则当 h_j 满足 $T_j^F=\max_{h_k\in PN(h_i)}T_k^F$ 时,定义 h_j 为节点服务 h_i 的关键节点.关键路径由一组关键节点构成,且路径的起点为 DAG 服务的入口节点,路径的终点为 DAG 出口节点.

关键节点的含义可以理解为:在服务 h_i 的前序节点中,服务结束时间最迟的节点记为关键节点.而一系列的关键节点组成了关键路径.

1.3 迁移服务死锁避免

在服务迁移过程中,始终要考虑的问题是迁移可能带来的服务死锁问题.由于 DAG 服务存在数据或逻辑上的依赖关系,当简单地将服务迁移至执行其他服务的节点时,存在两种情况:一是新迁移的服务具有较高的优先级,获得了优先执行的权限;二是根据先来先服务的原则,在服务队列中等待执行,分别如图 3 和图 4 所示.这两种迁移服务的执行情况都存在迁移服务与目的节点待执行服务之间相互关联的情况,按照 DAG 服务定义可知,若 DAG 服务中的先序服务被安排在其后续服务之后来执行,则可能出现当前执行服务缺乏逻辑或数据输入无法计算而执行挂起的现象,最终导致了整个 DAG 服务在迁移后的服务死锁.对于执行关键服务的网络节点来说,关键服务死锁是无法容忍的,因此在网络出现失效后的服务迁移中,首要问题是考虑避免服务死锁.

实现迁移服务死锁避免,通行的方法可以有如下两种:

- 一是对每一次迁移服务,都分析目的网络节点上服务之间的关联性,根据前后依赖关系来对待执行服务进行重新排队,确定执行的先后顺序.这种方法由于要求每一次服务迁移均要重新计算多个服务关联性,增加了网络节点上服务迁移的复杂性,难以满足关键服务执行的实时性与可靠性要求;
- 第 2 种方法则是将服务迁移至完全空闲的计算资源,仅执行这一个迁移服务,此类方法尽管也可以避免服务迁移带来的死锁问题,但由于在较大规模服务网络中关键服务的分解度较高,一个关键服务往往分解为数十个子服务执行,相应地要保证有足够多的空闲计算资源,从而会导致网络开发和维护成本较高.

借鉴这两种方法的思想,层次化 DAG 服务的迁移首先对服务进行重构分层,同一层内的 DAG 子服务之间相互独立,通过实现层内服务迁移机制就可以避免迁移服务死锁.

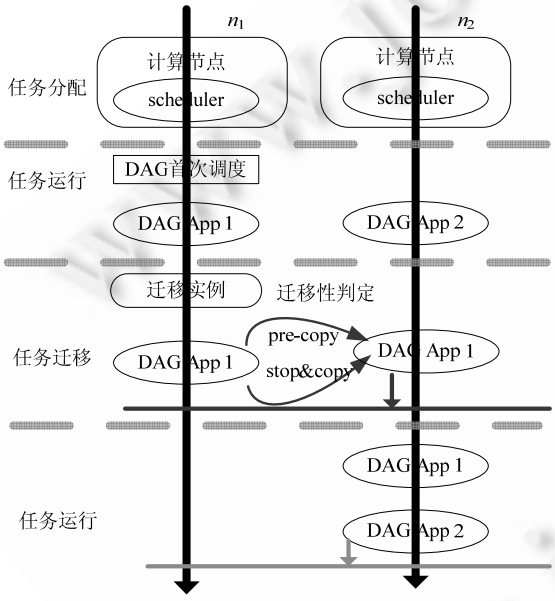

Fig.3 An example of deadlock happened in FIFO situation

图 3 先来先服务型服务死锁

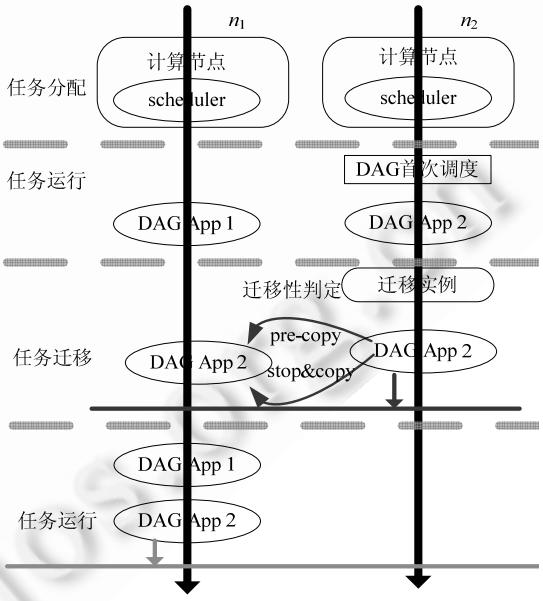

Fig.4 An example of deadlock happened in prioritized situation

图 4 优先级排队型服务死锁

2 关键服务动态迁移

2.1 服务执行时间计算

设计算子服务 S_j 分配给了节点 $Node_i$,根据网络可靠性理论,假设网络失效服从指数分布,即失效的发生属

于随机行为,则失效点的到达过程服从参数为λ_f的泊松分布,均值为μ_f,偏差为σ_f;而网络服务迁移时间则服从一般分布,均值为μ_C,偏差为σ_C;这里的迁移时间指的是失效 DAG 子服务迁移至当前空闲工作节点并开始执行的时间.设ω为子服务负载,服务执行期间发生 S 次失效,则服务计算时间可由下式计算:

$$T_j^C = \omega + Y_1 + Y_2 + ... + Y_S + Z_1 + Z_2 + ... + Z_S \tag{1}$$

其中,$Y_i(1\leqslant i\leqslant S)$为网络宕机时间,$Z_i(1\leqslant i\leqslant S)$表示网络恢复时间,则子服务计算时间的平均期望与方差分别为

$$E(T_j^C) = \left(\frac{1}{1-\lambda_f\mu_f} + \lambda_f\mu_C\right)\omega \tag{2}$$

$$V(T_j^C) = \left(\frac{\mu_f^2+\sigma_f^2}{(1-\lambda\mu)^3} + \mu_C^2 + \sigma_C^2 + 2\frac{\lambda_f\mu_C}{1-\lambda_f\mu_f}\right)\lambda_f\omega \tag{3}$$

引理 1. $E(T_j^C) = \left(\frac{1}{1-\lambda_f\mu_f} + \lambda_f\mu_C\right)\omega.$

证明:

$$E(T_j^C) = E(\omega + Y_1 + Z_1 + Y_2 + ... + Y_S + Z_S \mid S) = E(\omega + SE(Y_1) + SE(Z_1)) = \left(\frac{1}{1-\lambda_f\mu_f} + \lambda_f\mu_C\right)\omega. \quad \square$$

引理 2. $V(T_j^C) = \left(\frac{\mu_f^2+\sigma_f^2}{(1-\lambda\mu)^3} + \mu_C^2 + \sigma_C^2 + 2\frac{\lambda_f\mu_C}{1-\lambda_f\mu_f}\right)\lambda_f\omega.$

证明: 记 $U(S) = \begin{cases} Y_1 + Z_1 + Y_2 + Z_2 + ... + Y_S + Z_S & S > 0 \\ 0 & S = 0 \end{cases}$,

$$\begin{aligned} V(T_j^C) &= E(V(T \mid S)) + V(E(T \mid S)) \\ &= E(V(\omega + U(S) \mid S)) + V(E(w + U(S) \mid S)) \\ &= E(SV(Y_1) + SV(Z_1)) + V(\omega + SE(Y_1) + SE(Z_1)) \\ &= \lambda_f\omega V(Y_1) + \lambda_f\omega V(Z_1) + \lambda_f\omega(E^2(Y_1) + E^2(Z_1) + 2E(Y_1)E(Z_1)) \\ &= \lambda_f\omega E(Y_1^2) + \lambda_f\omega E(Z_1^2) + 2\lambda_f\omega E(Y_1)E(z_1) \\ &= \left(\frac{\mu_f^2+\sigma_f^2}{(1-\lambda\mu)^3} + \mu_C^2 + \sigma_C^2 + 2\frac{\lambda_f\mu_C}{1-\lambda_f\mu_f}\right)\lambda_f\omega. \end{aligned}$$ $\square$

根据引理 1 和引理 2 给出的计算公式,可以计算关键服务执行时间的平均期望与方差,相应地,可以定义服务执行时间的累积分布函数,计算公式如下:

$$P(T_j^C \leqslant t) = P(T_j^C \leqslant t \mid S = 0) + P(T_j^C \leqslant t \mid S > 0)P(S > 0) \tag{4}$$

2.2 DAG动态重构

为了有效迁移失效后 DAG 服务的执行,就必须考察 DAG 中的子服务依赖关系.一般存在两种依赖关系:服务依赖和资源依赖.服务依赖反映了 DAG 图中所规定的依赖关系,而资源依赖则反映了相同节点上子服务间的资源竞争关系.为保证在二次迁移中能够准确刻画服务之间的此类依赖关系,将二次迁移 DAG 图转化为层次化 DAG 图,具体步骤如图 5 所示.

定义 5(分层 DAG). 给定一个 DAG 元组为 $G=(V,E)$,其中,$v\in V$ 且$(u,v)\in E$,则 DAG 为 n 层 DAG 当且仅当:

(1) $V=L_1\cup L_2\cup\ldots\cup L_n, (L_i\cap L_j)=\varnothing, i\neq j$;

(2) 对每一个$(u,v)\in E$,其中 $u\in L_i$ 且 $v\in L_j$,则必有 $i>j$,

其中,L 为 DAG 图中节点的子集,V 为节点,E 为边的集合,u,v 为节点举例.

在层次化 DAG 中,若一个子服务的起始时间 S_j 是其邻近前序节点最大结束时间,则 DAG 图中子服务起始

时间 T_j^S 的累积分布函数可表示为 $P(T_j^S \leqslant t)=\prod_{S_k \in \psi(S_j)} P(T_k^F \leqslant t)$, 其中,$\psi(S_j)$表示子服务在 DAG 中的前序节点集合.同理, T_j^C 的累积分布函数可利用 $V(T_j^C)$ 进行计算,由于 T_j^S 和 T_j^C 属于相互独立的变量,因此,服务结束时间 T_j^F 的概率密度函数 $f_{T_j^F}(y)$ 可由下式计算:

$$f_{T_j^F}(y)=\int_0^y f_{T_j^S}(y-t) f_{T_j^C}(t)\mathrm{d}t \tag{5}$$

其中, $f_{T_j^S}$ 和 $f_{T_j^C}$ 分别表示 T_j^S 和 T_j^C 的概率密度.在计算分层 DAG 服务执行时间时,首先计算最上层子服务的服务开始、执行和结束时间,作为下一层子服务时间计算的输入,整个计算过程直至 DAG 最底层为止,子服务结束时间即整个 DAG 服务执行时间.

算法 1. 层次化 DAG 服务重构算法.
输入:DAG;
//初始服务集
输出:layered DAG.
//重构服务集
1. 对于所有 $h \in DAG$,计算 h 的先序服务集合 $Pre(h)$;
2. 对于所有 $h \in DAG$,若 $Pre(h)=\varnothing$,则 $Layer(h)=1$;否则,$Layer(h)=0$;
3. 以 *rPool* 表示先序服务已分层的服务集,对于所有 $s \in DAG$,判断是否 $Layer(h)>0$:是,则执行 4;否则,算法停止;
4. 更新 *rPool*,然后计算下式:

$$Layer(h)=\max_{H' \in pre(H)}(Layer(H'))+1$$

Fig.5 Algorithm of reconstructing layered DAG service
图 5 层次化 DAG 服务重构算法

2.3 服务可迁移性判定

DAG 服务的可迁移性判定以及可迁移服务目标状态的确定,是服务迁移的关键环节.由于导致失效发生的故障的不确定性,使得恢复的时间难以确定.而已有研究表明,大型应用程序失效恢复的时间要远大于网络内部进程迁移的时间.对于此类失效事件,应将服务迁移至当前空闲节点,如果没有,则不迁移.DAG 服务的可迁移性要求在当前工作位置下执行的服务类型必须符合迁移模式,即,要求服务实例在迁移前所执行获得的计算结果能够在目标位置下重现.如果 DAG 服务可迁移,还需确定迁移目标的当前状态,以便服务迁移到目标位置后能够继续执行.可迁移性判定的一个重要标准是能够确保服务迁移以后不会发生动态演化错误,如死锁或者重复计算已有结果.

定义 6(迁移目标有效). 若 DAG 服务迁移至目标位置后恢复执行不会引入动态演化错误(例如死锁或重复计算已有结果),则称该迁移目标当前有效.

定义 7(可迁移判定条件). 设 DAG 服务σ的当前位置和迁移目标位置分别为 N_S 和 N_D,则服务σ若由 N_S 迁移至 N_D 需满足两个条件:(1) 迁移目标位置 N_D 处于有效状态;(2) 迁移目标位置拥有空闲资源.

在 t=0 时刻,假设所有节点均正常,而当网络检测到处理节点 N_i 失效后,服务从 N_i 迁移到 N_j 的路径可表示为 $P_{ij}=[KN_j]$,其中,K 等于 0 或 1.节点 N_i 的服务时间、失效时间与恢复时间分别服从参数为 λ_{d_i},λ_{f_i},λ_{r_i} 的指数分布,设 N_i 在 t 时刻失效,则 N_i 在后续 $\lambda_{r_j}^{-1}$ 时间内停止服务.由上文分析可知,共有 $\lambda_{d_j}/\lambda_{r_j}$ 个服务无法继续在失效节点上执行,而 N_j 执行服务的稳态概率为 $\lambda_{r_j}/(\lambda_{f_i}+\lambda_{r_j})$, 因此服务迁移数为

$$L_{ij}^F=\left(\frac{\lambda_{r_j}}{\lambda_{f_j}+\lambda_{r_j}}\right)\left(\frac{\lambda_{d_i}}{\lambda_{r_i}}\right).$$

当存在多个迁移目标位置满足可迁移标准时,利用上文给出的算法对 DAG 服务进行分层重构,并计算每一条迁移路径下的 DAG 服务执行时间期望,则 DAG 执行时间最短的目标位置具有当前最佳的迁移路径.

定理 1. DAG 重构后服务迁移不会引起新的 DAG 死锁.

证明:采用反证法对服务迁移策略进行证明.设原 DAG 服务集合为Δ,为集中讨论问题,假设最初 DAG 服务集Δ不存在死锁;同时,假设迁移服务后 DAG 服务存在死锁.即,存在一个非空服务子集 $D\subseteq V$,且 D 中每一个服务的先序服务均包含在该子集中,则 $D=\{pre(t_1)\cup pre(t_2)\cup\ldots\cup pre(t_i),t_1,\ldots,t_i\in D\}$,从而必存在$(u,v)\in E$ 且 $u,v\in D$ 满足 $pre(u)=v$ 且 $pre(v)=u$;而由于服务迁移采用同一层次 DAG 子服务迁移,根据层次化 DAG 定义可知,同一层次内子服务不存在偏序关系,因此服务迁移前后服务间的偏序关系维持不变.于是,存在$(u,v)\in E$ 且 $u,v\in\Delta$满足 $pre(u)=v$ 且 $pre(v)=u$.即,原 DAG 服务集存在死锁.这与上文假设原 DAG 服务集不存在死锁矛盾,由此可知,采用服务重构策略进行服务迁移不会产生新的死锁,证毕. □

3 仿真实验与分析

针对认知网络 QoS 保障的需求,基于已有服务仿真平台设计开发所提出服务迁移方案的实验环境,核心算法及所有仿真实验均在该平台下实现并调试通过.仿真环境为 2.6GHz,1G 内存,Windows XP 操作系统以及 VS2005.在 VS2005 上实现了 DAG 服务层次化重构与迁移算法,主要数据接口与接口设计采用 C++结构体和类来实现.

为集中考虑服务迁移的核心问题,我们采用仿真手段来模拟网络不同节点之间的服务迁移以及恢复机制,以测试节点失效后服务迁移和服务等待两种方式的优劣.为考察复杂环境下多类型故障发生的服务迁移效率,实验共实现了 3 种网络节点失效,即 CPU 失效、内存失效以及服务通信失效.根据可信计算中经典的故障-错误-失效理论,失效是由于产生故障进而出现错误,错误又传递至服务层面而发生的,因此,实验实现了 3 种故障注入.实验中,DAG 服务图随机产生.对两种服务失效恢复方案的性能进行比较,一种是服务等待恢复方案,简称 WR 方案,另一种是本文所提出的服务迁移方案,简称 TM 方案.WR 方案是传统的服务恢复方案,是指当发生失效后服务中止,当节点修复后,该服务重新执行.

若无特殊说明,本实验所包含的节点数是改变的,如 DAG-20 指的是包含 50 个节点.这样做的目的是为了尽量减小节点数目变化对 DAG 重构带来的影响.每个节点的服务平均失效比率为每小时 1.5×10^{-3}~2.5×10^{-3} 等间距分布,每次失效停机时间随机选择,范围为 2~4 小时.错误恢复的时间开销固定为 2 小时,每个节点的负载是固定的.为了便于对比,WR 方案与 TR 方案的参数设置一致,即,每个节点的服务平均失效比率为每小时 1.5×10^{-3}~2.5×10^{-3} 等间距分布,每次失效停机时间和错误恢复的时间相同.

DAG 服务重构实验共生成 3 类 DAG 服务,根据服务数不同分为 DAG-20,DAG-40 和 DAG-100 这 3 幅服务图,并分别测试了层数分别为 2,5,10 的 DAG 服务重构时间开销,如图 6 所示.

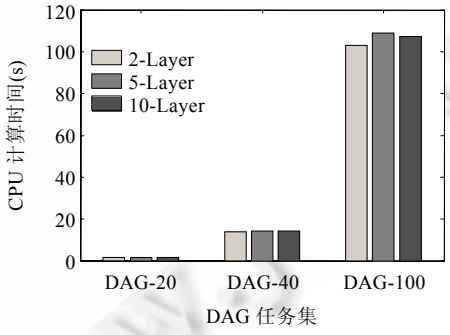

Fig.6 Time cost of DAG service reconstruction

图 6 DAG 服务重构时间开销

由图 6 可知,DAG 服务重构时间开销随着服务数量的增加而递增,其中,DAG-20 的分层重构时间不超过 5s;而 DAG-40 则增至 10s 以上;DAG-100 的分层重构时间最长,但也未超过 110s.考虑到实际应用中执行 DAG 服务网络一般均具有并发执行能力,重构时间将大为缩短,服务分层重构时间开销远小于进程迁移或者检查点备

份恢复时间.

实验还考察了服务迁移效果与失效检测准确性之间的关系,如图 7 所示.失效检测存在两类错误检测情况,即,无失效情况下的失效检测错误响应和发生失效后的失效检测不响应.失效发生后检测准确率越低,则网络失效后恢复时间越长,导致服务执行时间也相应地延长.在极端情况下,即失效检测误检率达到 90%的情况下,DAG-100 服务集的执行时间增加了 22%.考虑到这只是网络单点失效的情况,实际网络中失效发生的不确定性将导致服务执行时间不可避免地延长.

实验同时考察了失效漏检率与服务执行的关系,如图 8 所示.与失效误检率不同,失效漏检率是在发生失效的情况下网络未能检测到失效的概率.由图 8 可知,服务执行时间随着失效漏检率的增加而延长,且延长趋势较之失效误检情况更为明显.例如,在极端情况即失效漏检率达到 90%时,DAG-100 服务集的执行时间增加了 150%.这是因为失效误检后,网络仅会对被误检节点进行有限时间内的重启恢复,而当失效发生后,如果没有及时检测到,则网络不会采取任何恢复措施,因而较为明显地延长了 DAG 服务执行时间.

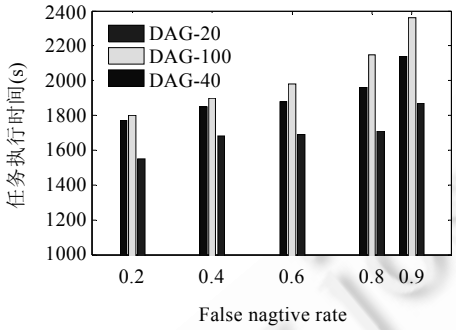

Fig.7 Relationship between service execution time and the false positive ratio of failure detection

图 7 服务执行与失效误检率的关系

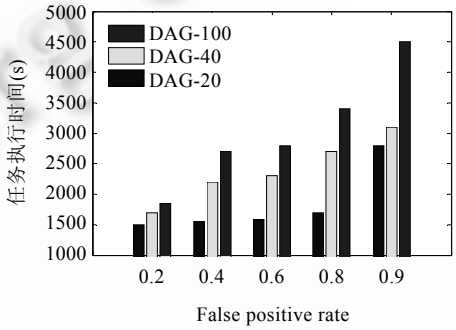

Fig.8 Relationship between service execution time and the false negative ratio of failure detection

图 8 服务执行与失效漏检率的关系

实验随机产生一组网络节点迁移集合,其中,节点个数共 4 个,随机产生 4 个服务集,其服务个数分别为 4,8,12,20 且负载固定.4 组服务集分别命名为 Group1,Group2,Group3,Group4.每组迁移均要执行 10 次,每次迁移均采用 CR 方案和 TM 方案进行迁移,并对 10 次迁移后的平均加速比进行比较.实验首先考察服务数目和网络节点比例对于服务失效迁移算法的影响,如图 9 所示.从实验中可以看出:随着服务数目与资源数目比值的增大,TM 方案的服务执行加速比逐渐提高,当比值为 3:1 时,加速比提升幅度达到最大.同样的并行加速比变化趋势也发生在 WR 方案中:当服务数目超过 4:1 时,两种方案迁移质量接近.这是因为在迁移过程中,所有服务都已经分配至各自的计算资源上,并组成了待执行服务队列,当服务队列长度增加,导致对待执行服务的管理所耗费的时间增加,抵消了服务失效后迁移执行所节约的时间,因此性能提供非常有限.对比 3 类固定负载下不同故障注入的 DAG 迁移质量可以发现:对于 CPU 故障,TM 方案要优于 WR 方案,CPU 故障恢复时间较长,在一段时间内会极大地影响网络节点的性能,此时,将服务迁移至其他空闲资源上执行,可有效缩短服务等待时间;而网络拥塞故障会加大服务迁移的通信开销,导致服务迁移的时间过长,降低了服务迁移质量.

实验同时考察了服务负载对于迁移质量的影响,如图 10 所示.实验以 Group3 为测试服务集,Group3 服务集合中服务个数为 12,服务负载变化区间为 10 分钟~60 分钟,服务负载依次递增,每次迁移均分别用 CR 与 TM 方案执行.图 10 给出了不同服务负载下迁移方案对于并行加速比的影响:对于 CPU 故障,TM 方案迁移质量的提升幅度最初随着服务负载的增大而增大,但在负载增加至 30 分钟后,无明显提升;对于网络故障,迁移质量的变化趋势也呈现增长-平稳的趋势.这是因为,对于计算时间较长即负载较大的计算服务来说,将失效服务迁移至当前空闲资源,需要较多时间来将服务执行的计算结果以及状态进行保存和传递,频繁的迁移则导致网络整体性能降低,因此,高负载条件下的 TM 服务迁移平均加速比提升不明显.综合上述实验结果分析可知,失效服务迁移

的迁移质量与故障发生的类型有着比较紧密的关系.失效网络节点上的待执行服务可以通过备份恢复过程来保证服务执行的连续性,该等待恢复方案适用于多种故障注入后的迁移过程.而服务迁移方案则更适用于故障原因未知或者故障恢复时间较长的情况,由于网络恢复时间较长,将待执行服务迁移至其他空闲资源往往能够加快服务的执行时间,从而保证关键服务的实时性.在实际应用中,随着服务并发执行的程度和服务划分粒度的不同,服务迁移方案的迁移质量往往也呈现出较大的区别.要保证服务迁移机制的实际应用效果,要求网络设计者在设计之初就考虑好网络内负载快速迁移的底层支持机制.

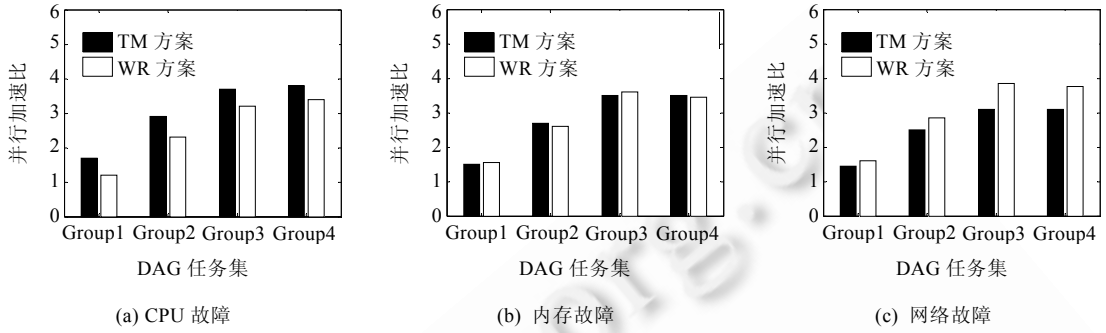

Fig.9 Relationship between the acceleration ratio and the number of DAG services

图 9 DAG 服务数目与执行加速比关系变化趋势

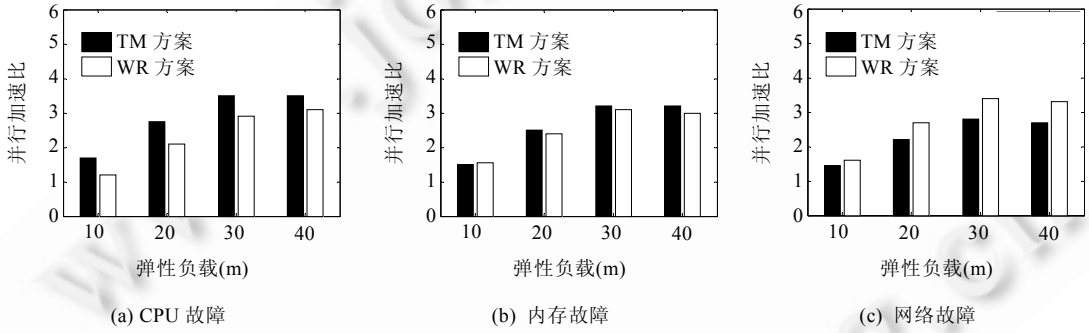

Fig.10 Relationship between the acceleration and service workload

图 10 弹性负载与执行加速比关系变化趋势

4 相关工作

认知网络是在 2000 年以后发展起来的一种新型网络,相对于传统网络,认知网络具有更高的智能性和自主性,能够根据外界环境的变化动态、适时地调整网络的配置.然而,认知网络节点随机接入和负载动态变化等增加了网络的高度动态性和复杂性,甚至会使服务失效概率成倍增长,是网络 QoS 保障的巨大挑战.

面对认知网络 QoS 的保障,传统的 IntServ 和 DiffServ 方法已经难于满足当前的需求.无论是 IntServ 还是 DiffServ 方法均沿用了传统互联网络的 QoS 体系,IntServ 需要在端到端传输路径上为每一信息流建立并维持资源预留,在高度动态性和海量业务流的前提下导致连接建立和连接释放阶段的额外开销激增[10];DiffServ 结构中,网络和端网络之间缺乏信令通信,不能提供端到端的 QoS 保障[11],难以满足未来网络智能化的需求.而且,它们都不能解决服务失效带来的 QoS 保障问题.

目前,认知网络 QoS 保障的思路一般是建立自适应的 QoS 保障体系,其中最为典型的一类方案是采用跨层设计方法,以保证 QoS 决策考虑到多个网络层次的因素.然而,以文献[6]为代表的跨层模式需要对现有网络协议栈结构加以改进,以满足网络层次跨层感知的需求,这极大地限制了其适用范围;而 Ali 等人[7]提出的部分可观

察马尔可夫过程也只能处理节点正常的情况,而且节点也没有考虑失效时的 QoS 保障问题.针对这些缺点,人们尝试着从系统整体上对 QoS 架构进行设计.文献[9]的服务质量自适应框架,采用同步多媒体统一语言 SMIL 保证在不干扰媒体流的情况下实现回滚;Attar 等人[10]提出采用 NASH 讨价还价的方法解决该用户的 QoS 保障;而 Swami 提出采用图论的方法保证认知网络的 QoS.但是上述研究都侧重于 QoS 的最大化,没有充分考虑认知网络节点频繁配置、频带动态调整和节点随机接入等特性带来的网络服务失效问题,这给网络服务调整后的 QoS 带来了巨大隐患[16].

针对该问题,本文参照并改进现有的任务迁移技术,提出一种基于 DAG 动态重构的认知网络服务迁移机制,并重点从理论上证明具有依赖关系的服务可迁移性判定以及迁移服务的死锁避免.

本文所作的改进主要表现在以下几个方面:

(1) 采用服务迁移技术,解决了认知网络节点频繁配置、频带动态调整和节点随机接入等特性带来的网络服务失效问题;

(2) 通过将关联服务转化为层次化 DAG 服务,从理论上证明了本文提出服务迁移算法的死锁避免问题;

(3) 与文献[13,14]等现有任务迁移技术相比,采用先迁移、后优化的思想,更多考虑失效恢复中待执行任务的处理问题,将迁移任务提前调度到当前空闲资源运行,达到了缩短任务执行时间的目的.

5 结 论

认知网络作为未来网络研究的核心内容,认知网络关键服务的 QoS 保障是认知网络研究与发展的关键问题.本文针对认知网络节点失效发生随机性以及关键服务运行可持续性的特点,提出了面向 QoS 保障的认知网络关键服务迁移方案.通过重新生成关联服务有向无环图,提出 DAG 动态重构算法,将关联服务转化为层次化 DAG 服务,然后计算关键服务迁移路径并给出可迁移服务死锁避免理论分析,将迁移服务提前迁移到当前网络空闲资源运行.仿真实验结果表明,服务迁移方案在弹性网络负载与未知故障情况下具有较好的 QoS 保障质量.未来将进一步研究基于自配置的认知网络 QoS 保障机制.

References:

[1] Fortuna C, Mohorcic M. Trends in the development of communication networks: Cognitive networks. Computer Networks, 2009, 53(9):1354−1376. [doi: 10.1016/j.comnet.2009.01.002]

[2] Thomas RW, DaSilva LA, MacKenzie AB. Cognitive networks. In: Arslan H, ed. Proc. of the IEEE Int'l Symp. on New Frontiers in Dynamic Spectrum Access Networks. New York: IEEE Computer Society Press, 2005. 352−360. [doi: 10.1109/DYSPAN.2005.1542652]

[3] Calinescu R, Grunske L, Kwiatkowska M, Mirandola R, Tamburrelli G. Dynamic QoS management and optimisation in service-based systems. IEEE Trans. on Software Engineering, 2011,37(3):287−409. [doi: 10.1109/TSE.2010.92]

[4] Awad C, Sanso B, Girard A. DiffServ for differentiated reliability in meshed IP/WDM networks. Computer Networks, 2008,52(10): 1988−2012. [doi: 10.1016/j.comnet.2008.02.023]

[5] Mammeri Z. Framework for parameter mapping to provide end-to-end QoS guarantees in IntServ/DiffServ architectures. Computer Communications, 2005,28(9):1074−1092. [doi: 10.1016/j.comcom.2005.01.008]

[6] Thomas RW, Friend DH, DaSilva LA, MacKenzie AB. Cognitive networks: Adaptation and learning to achieve end-to-end performance objectives. IEEE Communications Magazine, 2006,44(12):51−57. [doi: 10.1109/MCOM.2006.273099]

[7] Ali S, Yu FR. Cross-Layer QoS provisioning for multimedia transmissions in cognitive radio networks. In: Proc. of the IEEE Wireless Communications and Networking Conf. (WCNC 2009). Washington: IEEE Computer Society Press, 2009. 1−5. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?reload=true&punumber=4917480 [doi: 10.1109/WCNC.2009.4917655]

[8] Chen JL, Liu SW, Wu SL, Chen MC. Cross-Layer and cognitive QoS management system for next-generation networking. Int'l Journal of Communication Systems, 2011,24(9):1150−1162. [doi: 10.1002/dac.1218]

[9] Chang IC, Hsieh SW. An adaptive QoS guarantee framework for SMIL multimedia presentations with ATM ABR service. In: Proc. of the 2002 IEEE Global Telecommunications Conf. (GLOBECOM 2002). Washington: IEEE Computer Society Press, 2002. 1784−1788. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8454 [doi: 10.1109/GLOCOM.2002.1188505]

[10] Attar A, Nakhai MR, Aghvami AH. Cognitive radio game: A framework for efficiency, fairness and QoS guarantee. In: Proc. of the 2008 IEEE Int'l Conf. on Communications. Washington: IEEE Computer Society Press, 2008. 4170−4174. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4533035 [doi: 10.1109/ICC.2008.783]

[11] Swami S, Ghosh C, Dhekne RP, Agrawal DP, Berman KA. Graph theoretic approach to QoS-guaranteed spectrum allocation in cognitive radio networks. In: Proc. of the 2008 IEEE Int'l Performance, Computing and Communications Conf. Washington: IEEE Computer Society Press, 2008. 354−359. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4712735 [doi: 10.1109/PCCC.2008.4745077]

[12] Xin MJ, Wu C, Li WH. Self-Adaptive QoS mechanism for composite model solving based on multi-agent. Computer Engineering, 2007,33(18):72−74 (in Chinese with English abstract).

[13] Jin H, Sun XH, Zheng ZM, Lan ZL, Xie B. Performance under failures of DAG-based parallel computing. In: Proc. of the 9th IEEE/ACM Int'l Symp. on Cluster Computing and the Grid. Washington: IEEE Computer Society Press, 2009. 236−243. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5071832 [doi: 10.1109/CCGRID.2009.55]

[14] Wu M, Sun XH, Jin H. Performance under failures of high-end computing. In: Proc. of the ACM/IEEE Super-Computing Conf. New York: ACM Press, 2007. 1−11. http://dl.acm.org/citation.cfm?id=1362622&picked=prox&CFID=363999207&CFTOKEN=85969958 [doi: 10.1145/1362622.1362687]

[15] Song W, Ma XX, Lü J. Instance migration in dynamic evolution of Web service compositions. Chinese Journal of Computers, 2009,32(9):1816−1831 (in Chinese with English abstract).

[16] Lin C, Wang YZ, Ren FY. Research on QoS in next generation network. Chinese Journal of Computers, 2008,31(9):1525−1535 (in Chinese with English abstract).

附中文参考文献:

[12] 辛明军,吴超,李伟华.基于多 Agent 的复合模型求解自适应 QoS 机制.计算机工程,2007,33(18):72−74.

[15] 宋巍,马晓星,吕建.Web 服务组合动态演化的实例可迁移性.计算机学报,2009,32(9):1816−1831.

[16] 林闯,王元卓,任丰原.新一代网络 QoS 研究.计算机学报,2008,31(9):1525−1535.

林俊宇(1981－),男,广西博白人,博士,助理研究员,CCF 会员,主要研究领域为自律计算,未来网络和 QoS.
E-mail: linjunyu@hrbeu.edu.cn

卢旭(1983－),男,博士,工程师,主要研究领域为认知网络,网络安全,自律计算,性能评价.
E-mail: luxu@hrbeu.edu.cn

王慧强(1960－),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,认知网络,自律计算.
E-mail: wanghuiqiang@hrbeu.edu.cn

吕宏武(1983－),男,博士,讲师,CCF 会员,主要研究领域为自律计算,脆弱性分析,性能评估.
E-mail: lvhongwu@hrbeu.edu.cn

马春光(1974－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为密码学,信息安全.
E-mail: machunguang@hrbeu.edu.cn