软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(4):849−866 [doi: 10.13328/j.cnki.jos.004750]

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

一种基于 EQ 规则的组合服务运行时自适应方法*

张明卫[1], 朱志良[1], 张 斌[2], 张岳松[2]

[1](东北大学 软件学院,辽宁 沈阳 110004)

[2](东北大学 信息科学与工程学院,辽宁 沈阳 110004)

通讯作者: 张明卫, E-mail: zhangmw@swc.neu.edu.cn

摘 要: 组合服务通常运行在开放、变化和不确定的互联网环境中.动态复杂的运行环境,使得组合服务的执行具有不确定性和不可靠性,从而制约了服务组合技术的实际可用性.针对该问题,面向环境的分析与建模,提出了一种基于 EQ 规则的组合服务运行时自适应方法.该方法在记载组合服务的执行日志和各备选服务的运行环境数据的基础上,挖掘得到反映"一个备选服务在特定的环境状态下所表现出的质量会如何"的 EQ 规则库,从而指导组合服务如何响应实时产生的各类环境变化事件,以驱动组合服务的运行时自适应,保障其可靠运行.仿真实验结果表明:该方法能够提高组合服务在动态环境中的服务质量,增强其运行稳定性,是一种有效的组合服务运行时自适应方法.

关键词: Web 服务;组合服务;运行时自适应;EQ 规则;环境变化事件

中图法分类号: TP311

中文引用格式: 张明卫,朱志良,张斌,张岳松.一种基于 EQ 规则的组合服务运行时自适应方法.软件学报,2015,26(4):849−866. http://www.jos.org.cn/1000-9825/4750.htm

英文引用格式: Zhang MW, Zhu ZL, Zhang B, Zhang YS. Composite service runtime adaptation approach based on EQ rules. Ruan Jian Xue Bao/Journal of Software, 2015,26(4):849−866 (in Chinese). http://www.jos.org.cn/1000-9825/4750.htm

Composite Service Runtime Adaptation Approach Based on EQ Rules

ZHANG Ming-Wei[1], ZHU Zhi-Liang[1], ZHANG Bin[2], ZHANG Yue-Song[2]

[1](School of Software, Northeastern University, Shenyang 110004, China)

[2](School of Information Science and Engineering, Northeastern University, Shenyang 110004, China)

Abstract: Composite services usually run on the open, ever changing and uncertain Internet. Dynamic and complex execution environments make the composite service execution uncertain and unreliable, undermining the practical usability of service composition techniques. To solve this problem, an EQ rule based composite service runtime adaptation approach is proposed from the perspective of environment analysis and modeling. In this approach, the composite service execution log data and the candidate service execution environment data are first collected. Then, the EQ rules which express the knowledge (e.g. "how the performance of one candidate service will be in specific execution environment state") are mined. At last, the discovered EQ rules are applied to response all kinds of real time generated environment change events and to drive composite service runtime adaptation, and as such to ensure reliable execution for composite services. Experimental results show that the proposed approach can improve the quality of composite services and increase their execution stability effectively in dynamic environments.

Key words: Web service; composite service; runtime adaptation; EQ rule; environment change event

Web 服务组合技术可以灵活、高效地实现业务流程构造,快捷地完成新业务系统的构建和原有业务系统的更新和扩展,得到产业界和研究界广泛的关注[1].但是,由于组合服务所处的运行环境是动态变化的,致使它在实

* 基金项目: 国家自然科学基金(61100027, 61374178, 61202085, 61100090, 61100028); 中央高校基本科研业务费专项资金(N130417003, N110604002); 国家教育部博士点基金(20120042120010); 辽宁省博士启动基金(20121002)

收稿时间: 2014-06-29; 修改时间: 2014-10-14; 定稿时间: 2014-11-14

际运行过程中往往不可靠,主要表现在实际执行性能的下降,或者根本不能够执行.因此,如何保证组合服务能够适应其所处的开放、变化和不确定的运行环境,提高它的实际执行性能,减少异常发生的概率,即,保障动态环境中组合服务的自适应可靠运行,已经成为一个亟待解决的问题.

首先分析动态变化的环境如何影响组合服务的可靠运行.将组合服务所处的运行环境分为 3 类:服务器环境、网络环境和执行上下文环境.显然,在各 Web 服务所处服务器的不同负荷状态下,它们的性能表现也会不尽相同.在组合服务执行引擎与备选服务之间传输网络的不同状态下,组合服务端所感知到的各 Web 服务的质量也会不同.另外,一个 Web 服务在不同的时间段、调用时传输不同的数据量、与其一起执行的是不同的备选服务集合等,即:在不同的组合服务执行上下文环境中,其性能表现也可能大不相同.然而在组合服务执行实例的运行过程中,其所处的服务器环境、网络环境、执行上下文环境等都在不断地动态变化,从而使得组合服务所对应的各备选服务的质量也在不断变化,使得该执行实例中可能存在某个或某些未执行的备选服务性能显著下降、或者根本不能够再执行,从而影响了整个组合服务的可靠运行.另外,组合服务所对应的备选服务集中可能存在某些服务的性能显著上升,明显优于当前被选中的服务.考虑该因素,同样可以优化组合服务执行实例的运行性能.

从上述分析可知:由于组合服务所处的运行环境是错综复杂、动态变化的,从而使得组合服务执行实例上的各备选服务的质量也在实时变化,可能导致它们的执行质量与选取时所采用的质量信息差别很大,从而影响了组合服务的运行可靠性.为了解决如何在动态环境中保障组合服务可靠运行这一迫切的问题,研究界分别从不同的侧面提出了一些组合服务运行时自适应的方法[2−13].然而,当前大多数方法均缺乏对影响 Web 服务质量的环境因素的综合分析与应用,从而不能根据执行时刻上的运行环境状态信息准确地预测出各备选服务的实时质量信息,影响了组合服务运行时自适应的前提信息的准确性,降低了自适应运行的优化效果.另外,已有的组合服务运行时自适应方法重点关注自适应的实现机制,缺乏对自适应情境的细化分析.然而,在组合服务需要自适应调节的不同情境下,不同自适应调节动作的优化效果会有较大差异.因此,需要以组合服务的整体运行收益为目标,基于对组合服务运行时自适应情境的细化分析,生成更为恰当、合理的自适应调节策略.

相对于“组合服务”这一基础理论模型,本文从更侧重于实际和可控的基于服务软件系统[4]的角度出发,通过建立 Web 服务运行环境并分析 Web 服务运行环境状态和备选服务 QoS 之间的潜在联系,提出了一种基于 EQ 规则(environment quality rules,环境质量规则)的组合服务运行时自适应方法.EQ 规则是本文提出的核心概念,它反映了一个备选服务在给定的运行环境状态下其质量表现会如何的知识,可以从历史运行数据中挖掘得到并应用于指导组合服务的运行时自适应调节.该方法的大致思路可描述为:1) 记载组合服务的历史执行数据和各备选服务的运行环境数据,并实时产生环境变化事件;2) 基于采集的历史数据挖掘得到各备选服务的 EQ 规则集,以发现备选服务 QoS 与运行环境状态之间的关联知识;3) 将基于 EQ 规则预测出的备选服务 QoS 作为依据,通过响应环境变化事件来对组合服务进行运行时自适应调节.和以往的组合服务运行时自适应方法不同,本文方法完全从环境的角度出发,通过建立 Web 服务运行环境模型和挖掘得到的 EQ 规则来支持组合服务的运行时自适应,以提升组合服务在动态运行环境中的执行性能.另外,本文方法还基于环境变化事件对组合服务运行时自适应情境进行了细化分析,针对不同环境变化事件采用不同的自适应调节策略,提升了组合服务运行时自适应的优化效果,为用户提供较高的 QoS 保障.

本文主要有以下贡献点:

- 首先,本文建立了 Web 服务运行环境模型,可为以后从环境角度分析组合服务提供参考;
- 其次,本文提出了 EQ 规则的概念并给出了 EQ 规则的挖掘与应用方法,可以用来发现运行环境状态与备选服务 QoS 之间的潜在联系,并能够基于运行环境的实时状态较准确地预测出备选服务 QoS,以指导组合服务的选取、推荐和自适应等操作;
- 最后,在设计好环境变化事件概念的基础上,本文提出了一种全新的组合服务运行时自适应方法,以基于 EQ 规则预测出的备选服务 QoS 作为依据,由环境变化事件驱动,进行组合服务的运行时自适应调节,能够增强组合服务在动态环境中的运行稳定性,减少异常发生的概率并提升服务质量.

1 相关研究

为了使组合服务能够适应其所处的动态、不确定的运行环境,为用户提供可靠的服务保障,组合服务运行时自适应方法已成为组合服务领域的一个研究热点.

现有方法主要从两个不同的角度展开研究:

- 一是从用户角度出发,考虑在当前状态下如何进行自适应决策才能使得用户获得最优的服务质量,适合于用户量较少或用户私有的组合服务系统.文献[2]首先提出了“重规划”的概念与算法,以保证组合服务在运行时能够获得最优的 QoS.文献[3]则通过尽可能早地触发重规划来防止组合服务自适应时引起的服务运行中断,以提高服务自适应的性能.文献[4]提出了一种 ASQ 模型用来分析服务行为、系统资源状态和服务 QoS 之间的相互影响关系,并通过建立 ASQ 模型来支持组合服务的自适应.文献[5]提出一种可信的自适应服务组合机制,将组合服务的可信性保证问题转换为基于马尔可夫决策过程框架的自适应控制问题,并设计了相应的求解算法;
- 另外还有少量研究是从服务提供者的角度出发,考虑如何在尽量满足用户 QoS 约束的前提下,使服务提供者获得最大收益,适合于用户量较大的组合服务系统.文献[6]分析了如何平衡 SLA 违约的代价和防止违约的运行时自适应代价,以使得两者所带来的总代价最小,并给出了解决这一复杂优化问题的相关算法.文献[7]在组合服务运行时自适应过程中考虑了各备选服务的业务规则和长期受益,自适应决策不仅要依赖于当前的执行实例,更要依赖组合服务的整个运行周期,并给出了基于部分可观测马尔可夫决策过程的问题求解方法.

然而,无论是从哪个角度出发研究组合服务运行时自适应,由于自适应动作本身需要一定的性能开销,例如组合服务重选取往往被看作是一个最优化问题,具有 NP 难的特征和较高的计算代价,因此,由自适应动作本身所带来的性能开销往往成为组合服务运行时自适应的一个瓶颈.为了解决该问题,研究者主要从两个方面降低组合服务运行时自适应的代价.

- 一是通过合理减少重规划次数来优化组合服务自适应性能.文献[8]提出了变化信息值的概念,并将其作为评估重规划代价收益的依据,仅在查询代价低于系统收益时触发自适应.文献[9]则为了减少运行时环境自适应造成的系统性能下降,提出了基于 QoS 预测机制的服务选取算法;
- 另一方面则是从改进重规划性能角度展开研究.文献[10]提出了一种基于遗传算法的组合服务重规划方法.文献[11]则使用文化算法对组合服务进行重规划,并在性能上与基于遗传算法的方法进行了比较.文献[12]提出了一种初始可行解的启发式查找算法,并通过应用诸如模拟退火等元启发式算法来优化可行解.

与上述方法不同,文献[13]则在合理缩减解空间的基础上提出了一种服务重规划的启发式方法.简言之,以上该类方法是以获得次优解换取对计算时间的压缩.

由于自适应动作本身需要一定的性能开销,因此,组合服务自适应运行时,需要更准确的备选服务实时 QoS 信息和更有效的自适应调节策略.本文方法基于现有的自适应调节动作进行组合服务的运行时自适应.上述组合服务运行时自适应方法大多基于特定条件下备选服务的质量特征生成自适应策略,如服务提供者的声明、对服务质量的动态监测或对服务历史执行信息的统计等,缺乏对产生该服务质量特征的环境因素的建模与应用,缺乏对组合服务运行时自适应情境的细化分析,从而在一定程度上影响了组合服务自适应优化效果.本文方法则完全从环境角度出发,给出了一种全新的组合服务运行时自适应方法.基于 EQ 规则来反映备选服务 QoS 与运行环境状态之间的潜在联系,并对备选服务实时 QoS 进行预测;通过响应环境变化事件来驱动组合服务的运行时自适应,并在事件定制过程中细化分析了组合服务运行时自适应的各种情境,充分考虑了自适应的代价收益,为可靠组合服务的开发提供了一种可行的系统架构.

2 基于 EQ 规则的组合服务运行时自适应系统框架

本文所提出的基于 EQ 规则的组合服务运行时自适应方法的大致过程可描述为:首先,记载组合服务的执

行日志并采集各备选服务的运行环境数据;接下来,挖掘产生用于指导组合服务运行时自适应的 EQ 规则库;最后,由 EQ 规则预测出的备选服务 QoS 作为数据依据,通过响应环境变化事件来对组合服务进行运行时自适应调节,以保证服务能够适应其所处的动态环境.基于 EQ 规则的组合服务运行时自适应的系统框架如图 1 所示,主要包含信息采集、知识挖掘和自适应运行这 3 个主要模块.

- 信息采集模块:该模块为整个系统的基础模块,由日志记载器和环境监测器两类基本组件组成.前者负责生成组合服务的执行记录,并将数据写入到组合服务执行日志库中.后者则主要对各备选服务的运行环境进行监控,并把监测所得数值写入到备选服务运行环境信息库中.同时,若当前在某一环境因子上的监测值变化量已构成一个环境变化事件,则负责产生一个环境变化消息并投递到组合服务端的消息队列中,以驱动组合服务的运行时自适应.该模块的内容将在第 3 节详细介绍.
- 知识挖掘模块:该模块为系统的核心模块,其主要功能包括:1) 从组合服务执行日志库和备选服务运行环境信息库中提取各备选服务初始执行数据集;2) 针对各备选服务的任一 QoS 属性,对该 QoS 属性进行概念分段,确定与该 QoS 属性相关的环境因子并对这些环境因子进行概念分段,以生成各备选服务的待挖掘执行数据集;3) 挖掘得到各备选服务在各 QoS 属性上的 EQ 规则集,用来支持组合服务的运行时自适应.该模块的具体内容将在第 4 节介绍.
- 自适应运行模块:该模块为系统的目标模块,它主要负责从环境变化消息队列中获取消息,基于挖掘得到的 EQ 规则集预测相关备选服务的 QoS 变化情况,并对组合服务执行实例进行恰当的运行时自适应调节,以保证其能够达到当前环境约束下的最优 QoS.该部分内容将在第 5 节中详细介绍.

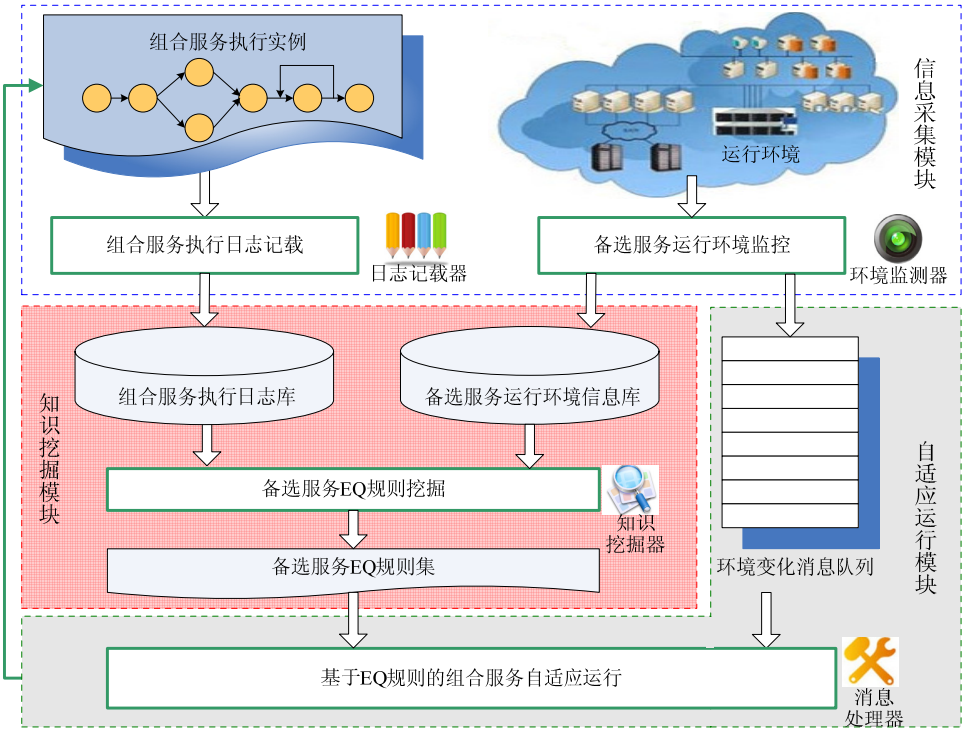


Fig.1 Architecture of EQ rule based composite service runtime adaptation

图 1 基于 EQ 规则的组合服务运行时自适应的体系结构

3 面向组合服务运行时自适应的信息采集

本文支持组合服务运行时自适应的基础信息分为两部分:一部分是由日志记载器记载的组合服务执行日

志,另一部分则是由环境监测器生成的环境变化消息以及其所采集的备选服务运行环境信息.下面将分别介绍这两类数据的采集方法.

3.1 组合服务执行日志的记载

由日志记载器所产生的组合服务执行日志库主要用于提取各备选服务的实际执行质量数据,另外还可以辅助提取一些备选服务的运行环境数据.因此,日志记载器该采用何种实现方式以及执行日志该记载哪些具体内容,都和要提取的备选服务 QoS 指标相关.

现已有一些组合服务日志记载架构的研究[14,15],在我们的前期工作中,也实现了一种能够记录丰富信息的日志记载器 CSLRA[16],本文依旧使用该日志记载框架对组合服务的执行日志进行记载.另外,根据 QoS 指标使用的普遍性,本文特选"响应时间"和"可靠性"这两个具有代表性的 QoS 指标进行研究.为了能够基于执行日志计算以上两个 QoS 指标,需要日志记载器 CSLRA 为每个 SOAP 消息记录定义 1 给出的各项信息.

定义 1(SOAP 日志项(SOAP log item,简称 SLI)). 这是日志记载器为每个 SOAP 消息记录的一条数据,可用七元组⟨*ProcessID*,*InstanceID*,*ServiceID*,*RequestID*,*Type*,*Time*,*Info*⟩表示,各项含义如下:

- *ProcessID*:业务流程号,用于指定产生该日志项的组合服务;
- *InstanceID*:类似于应用程序实例句柄,用于唯一指定一次组合服务的执行;
- *ServiceID*:用于指定发送请求或响应的具体服务,每个具体服务可由三元组⟨*Url*,*portType*,*operation*⟩唯一标识,这些信息可从 WSDL 文件中获取;
- *RequestID*:备选服务请求号,在一次组合服务执行实例中用于唯一指定一次对给定备选服务的请求;
- *Type*:指定 SOAP 消息的类型——请求消息或应答消息;
- *Time*:表示 SOAP 请求/响应消息发送的时间;
- *Info*:这是指 SOAP 消息中所包含的输入或输出信息.

利用定义 1 中 SLI 记录的各项内容,可以为 EQ 规则的挖掘提取包括以上两个 QoS 属性在内的许多 QoS 属性上的质量数据和一些辅助的备选服务运行环境数据.需要指出的是:日志记载框架CSLRA所记载的内容是可定制的,可以根据系统需求对其进行扩充或缩减,以挖掘新类型的 EQ 规则或提升日志记载效率.

3.2 备选服务运行环境的监控

环境监控器负责监测备选服务的实时运行环境,记录相关数据以形成服务运行环境信息库,并根据环境实时状态变化情况产生环境变化消息,以驱动组合服务的运行时自适应.备选服务运行环境的监控主要包含两方面的工作:一是分析潜在影响备选服务执行质量的环境因素有哪些,为备选服务运行环境进行建模;二是针对不同的环境因素,确定其数据监测方法.

潜在影响备选服务执行质量的环境因素有很多,大致可分为 3 类:一类是备选服务所在主机的相关环境因素,诸如该主机的内存利用率和 CPU 利用率等[4];第 2 类是组合服务端到备选服务端所在网络的相关环境因素,诸如站点的可达性、吞吐量、带宽利用率、丢包率等[17];第 3 类是在组合服务执行上下文中的相关环境因素,我们的一些前期工作关注了服务关联信息对备选服务质量的影响[18−20],现在也出现了一些该方面新的研究成果[21].另外,诸如服务调用时所处的时间段以及请求响应时所需传输的数据量等组合环境因素对备选服务质量的影响,也受到了我们和其他一些最新研究的关注[20,22].

在组合服务运行监控方面已有许多研究成果.文献[23]提出了一种基于事件的将组合流程设计与服务监控融入一体的框架 DISC.文献[24]则针对使用企业总线技术的 SOA 系统给出了一套完备的监控体系.文献[25]从语言-行为的角度建立了一套多层次的组合服务监控框架.文献[26]则针对服务组合系统提出了一种基于传输数据内容分析的高效的监控方法.然而,大多数组合服务监控方法仅基于备选服务之间相互传递的消息进行分析,类似于本文中日志记载器的功能,而对备选服务所在主机以及所处网络环境的监控关注较少.定义 2 给出了本文所关注的备选服务运行环境因素及其监控方法.

定义 2(备选服务运行环境(execution environment,简称 EE)). 这是指潜在影响备选服务执行质量的运行

环境因素,可用三元组⟨*Host*,*Network*,*Composition*⟩表示,各项的含义如下:

- *host*:是指潜在影响备选服务性能的其所在主机相关的环境因素.本文实验原型系统中的备选服务配置在 Windows XP 操作系统的 IIS 上,所以在众多 Windows 性能计数器中选择了较为常见且较可能影响服务 QoS 的 12 个性能指标,构成了备选服务的主机运行环境.它们分布在 6 个方面:系统方面包括 Process Queue Length(线程等待分配 CPU 资源所排队列长度);处理器方面包括%Processor Time (CPU 利用率)、%User Time(CPU 在用户模式下所花时间百分比)、%DPC Time(CPU 在网络处理上所花时间百分比);内存方面包括 Available Mbytes(剩余的可用物理内存)、Page Faults/sec(每秒钟处理的错误页数)、Pages/sec(为解决硬错误而从硬盘上读取或写入硬盘的页数);备选服务进程方面包括%Processor Time(该进程的 CPU 利用率)、Page Faults/sec(该进程每秒处理的错误页数)、Private Bytes(该进程无法与其他进程共享的字节数量);网络方面包括 Byte Total/sec(网络中传输字节的速度);磁盘方面包括%Disk Time(磁盘读写所用时间百分比).这些环境向量的监测由备选服务所在主机的操作系统提供.
- *Network*:是指潜在影响备选服务执行质量的从组合服务端到备选服务端传输网络相关的环境因素.从监控效率和 EQ 规则挖掘有效性上考虑,本文建立包含以下具有代表性因素的备选服务网络运行环境,分别为 Reachability(备选服务节点可达性),Packet Loss Ratio(网络丢包率)和 RTT(网络传输往返时延).这些网络环境参数通过基于 SNMP 协议的简单监控工具采集.
- *Composition*:是指影响备选服务性能的组合服务执行上下文相关的环境因素,可用三元组⟨*Correlation*,*TimeSeg*,*DadaAmount*⟩来表示,分别表示与该服务潜在相关的其他备选服务的集合、服务调用时所处的时间段和请求响应该服务所需传输的数据量.由以上 3 个因素构成的组合服务执行上下文运行环境通过日志记载器 CSLRA 采集.

定义 2 描述了本文所建立的备选服务运行环境模型和各环境因素的监控方法,需要指出的是,该环境模型可以根据各自服务的领域特征进行扩充或缩减.环境监控器需要根据该定义采集相关数据以生成备选服务运行环境信息库.同时,需要根据第 5 节中所描述的事件确定方法生成环境变化消息,以驱动组合服务的运行时自适应.

4 EQ 规则的挖掘

EQ 规则的挖掘是本文的核心内容,本节将详细描述 EQ 规则的相关概念和 EQ 规则挖掘的各个步骤:1) 生成初始挖掘数据集;2) 对初始挖掘数据进行预处理;3) 挖掘得到各备选服务的 EQ 规则集.其中,对服务关联运行环境因子的 EQ 规则挖掘,还需要进行额外处理,可以参见我们的前期工作[18],这里不再进行特殊说明.

4.1 基本概念

EQ 规则描述了形式上诸如“在特定运行环境状态下,备选服务 QoS 表现会如何”的知识,其用来根据运行环境实时状态的变化判断备选服务质量变化情况,以指导组合服务的运行时自适应.下面给出 EQ 规则及其相关概念.

定义 3(概化 QoS 属性(generalized QoS attribute,简称 GQOS)). 给定备选服务 s 的一个连续型 QoS 属性 A,$\{q_1,q_2,\ldots,q_n\}$是 s 在 A 上的 n 个观测值,且满足 $q_1\leqslant q_2\leqslant\ldots\leqslant q_n$,则 s 在属性 A 的概化 QoS 为 $GQoS=\{q_1^G,q_2^G,\ldots,q_m^G\}$,且有:

$$\begin{cases} q_i^G=\sum_{j=(i-1)\times(n/m)+1}^{i\times(n/m)} q_j \Big/ (n/m), & 1\leqslant i\leqslant m-1 \\ q_i^G=\sum_{j=n-n/m-n\%(n/m)+1}^{n} q_j \Big/ (n/m+n\%(n/m)), & i=m \end{cases} \tag{1}$$

由上式可知:概化 QoS 是将备选服务 s 在属性 A 的 n 个观测值按照等出现次数概化成 m 段,且每一段的概

化 QoS 值为该段内所有观测值的平均数.通常,$m<<n$,m 的取值可在知识发现过程中由用户指定.比如:对于文中所采用的“响应时间”QoS 属性,m 一般可取值为 3 或 5;而对于文中所采用的“可靠性”这类观测值只取“0”或“1”的离散型 QoS 属性,其概化 QoS 取值与原取值相等.

定义 4(概化运行环境因子(generalized execution environment factor,简称 GF)). 给定一个潜在影响备选服务执行质量的连续型运行环境因子 F,$\{s_1,s_2,\ldots,s_n\}$是环境因子 F 的 n 个观测状态值,且满足 $s_1\leqslant s_2\leqslant\ldots\leqslant s_n$.存在 m+1 个区间边界 $b_1,b_2,\ldots,b_{m+1}$,且有$(b_1=s_1)\wedge(b_{m+1}=s_n)\wedge(b_1<b_2<\ldots<b_{m+1})$,将这 n 个状态值分成了 m 个区间,则 F 所对应的概化运行环境因子为

$$GF=\{s_i^F=(b_i+b_{i+1})/2\}(1\leqslant i\leqslant m) \tag{2}$$

概化运行环境因子是针对给定备选服务的特定 QoS 属性,将连续型环境因子进行概念分段的结果;而对于离散型的运行环境因子,其概化环境因子等于其本身.

定义 5(运行环境状态向量(execution environment state vector,简称 SV)). 由各维运行环境因子的状态取值组成的向量称为备选服务的运行环境状态向量.如果向量中包含了定义 2 中所有环境因子的状态值,则称其为运行环境状态完备向量;否则,称为子向量.如果向量中所包含的均是概化运行环境因子的状态值,则称其为运行环境状态概化向量;否则,称为初始向量.

定义 6(备选服务初始执行数据(candidate service initial execution data,简称 IED)). 给定备选服务 s 的一次执行,其在 QoS 属性 A 上的取值 q 加上调用时刻的运行环境状态完备向量 SV 所形成的数据 $q\infty SV$,称为服务 s 在 QoS 属性 A 上的一条初始执行数据.

定义 7(备选服务待挖掘执行数据(candidate service prepared execution data,简称 PED)). 给定备选服务 s 的一次执行,其在概化 QoS 属性 A 上的取值 q^G 加上与 A 相关的在调用时刻运行环境状态概化向量 SV^G 形成的数据 $q^G\infty SV^G$,称为服务 s 在概化 QoS 属性 A 上的一条待挖掘执行数据.

定义 8(备选服务环境质量规则(candidate service environment quality rules,简称 EQ Rules)). 给定备选服务 s 在 QoS 属性 A 上的一个待挖掘执行数据集 $PED=\{ED_1,ED_2,\ldots,ED_n\}$, SV_{sub}^G 是一个运行环境状态概化子向量,q^G 是 s 在 A 所对应的概化 QoS 属性上的一个取值,则存在形如 $SV_{sub}^G\Rightarrow q^G$ 的蕴含式.如果 $SV_{sub}^G\cup\{q^G\}\subseteq ED_i$,则称执行数据 ED_i 包含蕴含式 $SV_{sub}^G\Rightarrow q^G$. 蕴含式 $SV_{sub}^G\Rightarrow q^G$ 在集合 PED 中的支持度为集合 PED 中包含该蕴含式的执行数据占总执行数据的百分比,即:

$$support(SV_{sub}^G\Rightarrow q^G)=P(SV_{sub}^G\cup\{q^G\})=|ED(SV_{sub}^G\cup\{q^G\})|/|PED| \tag{3}$$

其中,$|ED(SV_{sub}^G\cup\{q^G\})|$ 和$|PED|$分别表示集合 $ED(SV_{sub}^G\cup\{q^G\})$ 和 PED 中数据元素个数,而 $ED(SV_{sub}^G\cup\{q^G\})$ 表示集合 PED 中包含 $SV_{sub}^G\cup\{q^G\}$ 的待挖掘执行数据形成的集合.

蕴含式 $SV_{sub}^G\Rightarrow q^G$ 的置信度为集合 PED 中包含运行环境概化子向量 SV_{sub}^G 的执行数据中同时也包含概化 QoS 值 q^G 的比例,即:

$$confidence(SV_{sub}^G\Rightarrow q^G)=P(\{q^G\}|SV_{sub}^G)=|ED(SV_{sub}^G\cup\{q^G\})|/|ED(SV_{sub}^G)| \tag{4}$$

若蕴含式 $SV_{sub}^G\Rightarrow q^G$ 同时满足以下 3 个条件:

1) $support(SV_{sub}^G\Rightarrow q^G)\geqslant$min_sup,0<min_sup<1 为支持度阈值,则称蕴含式 $SV_{sub}^G\Rightarrow q^G$ 为频繁蕴含式;
2) $confidence(SV_{sub}^G\Rightarrow q^G)\geqslant$min_*conf*,0<min_*conf*<1 为置信度阈值,则称蕴含式 $SV_{sub}^G\Rightarrow q^G$ 为可信蕴含式;
3) $\neg\exists SV'^G_{sub}\subset SV_{sub}^G\,|\,support(SV_{sub}^G\Rightarrow q^G)\geqslant\min\_sup\wedge confidence(SV'^G_{sub}\Rightarrow q^G)\geqslant\min\_conf$,

则称蕴含式 $SV_{sub}^G\Rightarrow q^G$ 为服务 s 的一条 EQ 规则.

蕴含式 $SV_{sub}^G\Rightarrow q^G$ 表达了备选服务 s 在与给定运行环境状态下,其概化 QoS 将会取值 q^G 的知识规则.其中,条件 1)保证了该规则的非偶然性;条件 2)保证了该规则的可信性;条件 3)则用来寻找影响服务 s 质量的最直接的运行环境因素,保证了该规则的简易性.

以上简要给出了备选服务 EQ 规则的概念描述,下面将阐述 EQ 规则的具体挖掘方法.

4.2 备选服务初始执行数据集的提取

根据定义 6,备选服务 s 的初始执行数据包含 s 在各 QoS 属性上的质量数据和在 s 执行时刻上的运行环境完备状态数据.本文采用“响应时间”和“可靠性”这两个具有代表性的 QoS 指标进行分析与阐述,对其他 QoS 指标的研究分析可参考文献[18].下面,首先给出基于组合服务执行日志库的这两个 QoS 指标的计算方法和属性值的提取方法.

- 响应时间:用来评测备选服务完成请求的速度,是组合服务端感知到的备选服务响应一次请求所需要的时间.一个服务 s 的响应时间等于从组合服务端发送一个消息 M 到接受到相应返回消息的时间.
- 可靠性:是指备选服务在被组合服务端调用时能否正确响应服务请求.所谓正确,是指在对收到的服务请求进行处理后,备选服务发出的服务响应符合预先的功能设计.

给定备选服务 s,要基于组合服务执行日志库提取 s 每次执行的“响应时间”和“可靠性”,首先需要基于定义 1 所记载的信息在日志库中找到对 s 每次请求所对应的响应数据;然后,基于 *RequestID* 做连接生成 s 的执行数据.假设$\langle s,RequestID,Time_{req},Time_{resp},Info_{req},Info_{resp}\rangle$是 s 的一条执行数据,P_{in} 和 P_{out} 分别表示 WSDL 文档中描述的备选服务 s 应有的输入和输出,C_{in} 和 C_{out} 则表示 s 在输入和输出上的约束条件.那么,则有如下请求状态变量:

$$s_{req}=\begin{cases}1, & (Info_{req}\supseteq P_{in})\wedge C_{in}(Info_{req})\\ 0, & \neg(Info_{req}\supseteq P_{in})\vee\neg C_{in}(Info_{req})\end{cases} \tag{5}$$

表示实际调用时的输入信息 $Info_{req}$ 包含 s 所要求的输入信息 P_{in},并且当 $Info_{req}$ 满足约束条件 C_{in} 时,s_{req} 取值为 1,为完备请求;否则取值为 0,为非完备请求.

有响应状态变量:

$$s_{resp}=\begin{cases}1, & (Info_{resp}\supseteq P_{out})\wedge C_{out}(Info_{resp})\wedge Time_{resp}\\ 0, & (\neg(Info_{resp}\supseteq P_{out})\vee\neg C_{out}(Info_{resp}))\wedge Time_{resp}\\ -1, & Time_{resp}=NULL\end{cases} \tag{6}$$

表示当没有收到响应信息时,s_{resp}取值为−1;当收到的实际响应信息$Info_{resp}$包含应有响应信息P_{out},且$Info_{resp}$满足约束条件 C_{out} 时,s_{resp} 取值为 1,为正确应答;否则,s_{resp} 取值为 0,为无效应答.

针对备选服务 s 的一次请求,基于以上提取的请求状态变量和响应状态变量,可计算对 s 本次请求的响应时间为

$$V_{ResponseTime}=\begin{cases}Time_{resp}-Time_{req}, & s_{resp}\neq -1\\ -1, & s_{resp}=-1\end{cases} \tag{7}$$

其中,−1 表示未采集到相应的 QoS 属性值.对 s 本次请求的可靠性为

$$V_{Reliability}=\begin{cases}1, & s_{req}=1\wedge s_{resp}=1\\ 0, & s_{req}=1\wedge s_{resp}\neq 1\\ -1, & s_{req}=0\end{cases} \tag{8}$$

以上描述了基于组合服务执行日志库的备选服务质量数据的提取方法.针对备选服务 s 的每次执行,还需要提取服务 s 在调用时刻τ上的运行环境状态完备向量,以形成服务 s 的初始执行数据集.基于定义 2,含有 12 个因子的主机运行环境状态向量和含有 3 个因子的网络运行环境状态向量可以简单地从备选服务运行环境日志库中提取,而含有 3 个因子的组合服务执行上下文运行环境状态向量也可简单地从组合服务执行日志库中提取.将提取出来的备选服务 s 在调用时刻τ上的运行环境状态完备向量和本次执行的质量数据进行连接,即得到了服务 s 的一条初始执行数据.在挖掘服务 s 在某 QoS 属性上的 EQ 规则之前,还需要对 s 的初始执行数据进行预处理,以对连续型的 QoS 属性和运行环境因子进行概念分段,并去掉与该 QoS 属性无关的运行环境因子.

4.3 备选服务待挖掘执行数据集的生成

由定义 7 可知,给定备选服务 s 的一个 QoS 属性 A,其待挖掘执行数据集的生成主要包含以下 3 方面的工作.

1) 如果 A 是连续型 QoS 属性(本文中,如果 A 是“响应时间”),对其进行概念化分段;
2) 选择与 A 相关的备选服务运行环境因子;
3) 对选出的连续型运行环境因子进行概念化分段.

下面将简述以上预处理工作的方法.

- 首先,基于定义 3,对连续型 QoS 属性的概念分段较为简单,只要在初始执行数据集中对该 QoS 属性按照等出现次数概化成 m 段即可(本文中,m 取值为 5),这里不再详述;
- 对于第 2)步工作,通过计算各运行环境因子与 QoS 属性 A 之间的相关系数来确定各环境因子与属性 A 之间的相关程度,从而对众多运行环境因子进行取舍;
- 对于第 3)步工作,本文采用了一种基于基尼系数的概念划分方法,用于递归地划分数值属性的值.

给定一个由概化 QoS 属性 A 和连续型运行环境因子 E 组成的数据元组的集合 S,基于基尼系数对 E 进行概念划分的算法可简要描述如下:

1) E 的每个值可以认为是一个潜在的区间边界或阈值 T.例如,E 的值 v 可以将样本 S 划分成分别满足条件 $E<v$ 和 $E\geqslant v$ 的两个子集,这样就创建了一个二元划分;
2) 给定 S,所选择的阈值是这样的值,它使划分得到的基尼系数最小.基尼系数的度量如下:

$$gini_{split}(S,T)=\frac{|S_1|}{|S|}gini(S_1)+\frac{|S_2|}{|S|}gini(S_2) \tag{9}$$

其中,S_1 和 S_2 分别对应于 S 中满足 $A<T$ 和 $A\geqslant T$ 的样本.对于给定集合,它的基尼系数根据集合中样本的类分布来计算.例如,给定 QoS 属性 A 含有 m 个类,S 的基尼系数为

$$gini(S)=1-\sum_{j=1}^{m}P_j^2 \tag{10}$$

其中,p_j 是类 j 在 S 中的概率,等于 S 中类 j 的样本数除以 S 的样本总数.

3) 确定阈值的过程递归地用于所得到的每个划分,直到满足用户给定的某个终止条件,例如达到了本文要求的分段数.

基于基尼系数的离散化方法使用数据分布反映出的信息量特征,可尽可能地将区间边界定义在准确的位置上,有助于提高概念划分的准确性.

通过以上 3 步的处理,可以在备选服务初始执行数据集上得到各备选服务在各 QoS 属性上的待挖掘执行数据集,以进行 EQ 规则的挖掘.

4.4 备选服务EQ规则的挖掘

给定备选服务 s 在 QoS 属性 A 上的待挖掘执行数据集 $PED=\{ED_1,ED_2,\ldots,ED_n\}$,本文在集合 PED 上挖掘 s 的所有 EQ 规则主要包含两步:第 1 步是查找所有的频繁蕴含式,第 2 步是在这些蕴含式中生成所有的 EQ 规则.由于第 2 步的开销远小于第 1 步,所以算法的总体性能由第 1 步决定.

频繁蕴含式查找采用一种逐层搜索的迭代方法.

- 首先,通过扫描待挖掘执行数据集 PED,查找所有满足支持度阈值的频繁单数据项,记作 L_1.在此基础上,得到仅含有两个数据项的频繁 2 项蕴含式,记作 L_2;
- 然后,用 L_2 寻找所有的频繁 3 项蕴含式 L_3,L_3 用于寻找 L_4.如此下去,直到不能再找到含有更多项的蕴含式为止.

基于频繁 k–1 项蕴含式 L_{k-1} 查找频繁 k 项蕴含式 L_k 包含“连接”和“剪枝”两步.

- 连接步用于产生所有候选 k 项蕴含式 C_k,如果 L_{k-1} 中的两个频繁蕴含式 l_i 和 l_j 只有一个数据项不同,则 l_i 和 l_j 可以连接到一起形成一个候选 k 项集;
- 剪枝步用于生成所有频繁 k 项蕴含式 L_k.剪枝规则有两个,其一称为“Apriori”性质,即,频繁项集的所有非空子集也必须是频繁的.与普通的关联规则挖掘不同,EQ 规则挖掘另外还具有“蕴含式”性质,即:对于 $C_k(k\geqslant 2)$中的任一候选项集 l_i,如果不含有 QoS 属性值项 $q^G\in GQoS$,即不构成 $SV_{sub}^G\Rightarrow q^G$ 形式的

蕴含式,则可以剪去该项集.基于定义 8,可以简单地证明基于“蕴含式”性质剪枝不会丢失任何频繁蕴含式.剪枝后需要扫描数据集 *PED*,确定 C_k 中留下的每个蕴含式的支持度,从而确定 L_k.

备选服务 EQ 规则挖掘算法 EQRules 描述如下.

算法 1. EQRules.

输入:备选服务 *s* 在属性 *A* 上的待挖掘执行数据集 *PED*,最小支持度阈值 min_sup;

输出:服务 *s* 在属性 *A* 上的所有频繁蕴含式.

L_1=*frequent*(*PED*); //查找 *PED* 中的所有频繁 1 项集

FOR (*k*=2; $L_{k-1}\neq\varnothing$; *k*++){

  C_k=*AprioriGen*(L_{k-1}); //生成新的候选项集 C_k

  AprioriDelete(C_k,L_{k-1}); //利用“Apriori”性质剪枝

  FOR (*ED*∈*PED*){

    C_{ED}=*subset*(C_k,*ED*); //查找被任一执行数据 *ED* 包含的 C_k 中的频繁蕴含式集合 C_{ED}

    //更新 C_{ED} 中每个频繁蕴含式的支持度

    FOR(*c*∈C_{ED}) *c*.*UpdateSupport*();

  }

  L_k={*c*∈C_k|*c*.*support*>min¬¬¬¬_sup};

}

RETURN *L*=$\bigcup_k L_k$;

过程 1. AprioriGen.

C_k=∅; //初始化候选项集 C_k

FOR (∀l_i∈L_{k-1})

FOR (∀l_j∈L_{k-1})

  IF ((l_i[1]=l_j[1]) && (l_i[2]=l_j[2]) && ... (l_i[*k*−2]=l_j[*k*−2]) && (l_i[*k*−1]≠l_j[*k*−1])){

    c=l_i∞l_j; //连接满足条件的频繁项集 l_i 和 l_j

    IF (*c*.*Contained*(q^G)) //利用“蕴含式”性质剪枝

      C_k=*c*∪C_k ; //将 *c* 加入到候选蕴含式集 C_k 中

  }

RETURN C_k;

算法基于经典算法 Apriori 的思路,能够挖掘得到备选服务 *s* 在 QoS 属性 *A* 上的所有频繁蕴含式.在此基础上生成服务 *s* 的所有 EQ 规则相对较为简单,只要针对每条频繁蕴含式 $SV_{sub}^{G}\Rightarrow q^G$,生成所有可能的子蕴含式 $SV'^{G}_{sub}\Rightarrow q^G\wedge SV'^{G}_{sub}\subseteq SV^{G}_{sub}$,如果蕴含式 $SV'^{G}_{sub}\Rightarrow q^G$ 的置信度大于阈值 min_*conf*,并且不存在更为简单的蕴含式 $WS''_{sub}\Rightarrow q_g^A$,有 $WS''_{sub}\subseteq WS'_{sub}\wedge confidence(WS''_{sub}\Rightarrow q_g^A)\geqslant$min_*conf*,则蕴含式 $SV'^{G}_{sub}\Rightarrow q^G$ 属于备选服务 *s* 的一条 EQ 规则.

挖掘得到的 EQ 规则用于根据备选服务的运行环境实时状态预测其 QoS,以指导组合服务的运行时自适应调节.

5 基于 EQ 规则的组合服务运行时自适应

基于 EQ 规则的组合服务运行时自适应主要包含两方面工作.

- 其一是根据备选服务的 EQ 规则和其所处的运行环境实时状态预测它的性能表现;
- 其二是对备选服务的环境变化事件进行响应,以使组合服务能够适应动态多变的运行环境,增强服务的 QoS 保障.

5.1 基于EQ规则的备选服务QoS预测

为了能够基于EQ规则准确预测各备选服务的QoS,提出了一种基于反馈度的关联分类算法,除了进行备选服务 QoS 分类预测外,还在许多公用数据集上进行了测试,有着较高的分类准确率.

给定备选服务 s 在 QoS 属性 A 上的 EQ 规则集 EQS,假设满足当前运行环境状态向量 SV^G 的 EQ 规则子集为 $\{SV^G_{sub\cdot 1}\Rightarrow q^G_1, SV^G_{sub\cdot 2}\Rightarrow q^G_2,...,SV^G_{sub\cdot n}\Rightarrow q^G_n\}$,则服务 s 在 QoS 属性 A 上的取值为

$$q(s)=\sum_{j=1}^{n} w_j q_j^G \tag{11}$$

上式中:QoS 概化属性值 q_j^G ($1\leqslant j\leqslant n$)可能相同;而 w_j 是 EQ 规则 $SV^G_{sub\cdot j}\Rightarrow q_j^G$ 的权值,可以通过下式计算:

$$w_j = s_j \mathrm{e}^{k_1 c_j + k_2 f_j} \Big/ \sum_{j=1}^{n} s_j \mathrm{e}^{k_1 c_j + k_2 f_j} \tag{12}$$

其中,s_j 是EQ规则 $SV^G_{sub\cdot j}\Rightarrow q_j^G$ 的支持度;c_j 是该规则的置信度;f_j 是该规则的反馈度,用于衡量该规则的分类准确度;而 k_1 和 k_2 是可由用户设定的权重调节系数,文中 k_1=k_2=10.最终预测出的 s 的概化 QoS 值 $q^G(s)$为:$q(s)$在 A 所对应的概化 QoS 属性上所在区间的质量值.

需要注意的是:并非所有备选服务的质量都能通过 EQ 规则来预测,因为备选服务 s 可能没有满足当前运行环境状态向量的 EQ 规则,甚至根本就没有挖掘出 EQ 规则来.对于不能预测 QoS 的备选服务,仍采用其发布的 QoS 数据作为服务自适应的依据.

5.2 环境变化事件驱动的组合服务运行时自适应调节

备选服务 QoS 受其所处动态运行环境的影响,不断变化的运行环境状态使得组合服务端所感受到的备选服务 QoS 也在不断变化.在第 3.2 节详细描述了本文所建立的备选服务运行环境模型和各环境指标的监控方法,下面描述本文如何合理地确定环境变化事件以及如何针对各类环境变化事件进行运行时自适应调节.

定义 9(环境变化事件(environment change event,简称 ECE)). 给定备选服务 s 基于 QoS 属性 A 划分得到的一个概化运行环境因子 GF,$b_1<b_2<\ldots<b_{m+1}$ 是 GF 的 m 个区间边界,$v_b\in[b_i,b_{i+1}]$($1\leqslant i\leqslant m$)是 GF 的状态基值,v_c 是 GF 的状态实时值,则:

- 如果 $v_b<(b_i+b_{i+1})/2$,当 $i-1>0\wedge v_c\leqslant(b_{i-1}+b_i)/2$ 或 $i\neq m\wedge v_c\geqslant b_{i+1}$ 时,构成一个环境变化事件;
- 如果 $v_b=(b_i+b_{i+1})/2$,当 $i\neq 1\wedge v_c<b_i$ 或 $i\neq m\wedge v_c\geqslant b_{i+1}$ 时,构成一个环境变化事件;
- 如果 $v_b>(b_i+b_{i+1})/2$,当 $i\neq 1\wedge v_c<b_i$ 或 $i<m\wedge v_c\geqslant(b_{i+1}+b_{i+2})/2$ 时,构成一个环境变化事件.

由定义 9 可知:要构成环境变化事件,首先要求运行环境因子 GF 的状态实时值 v_c 和状态基值 v_b 不在同一个概念分段内,这样做是因为,如果 v_b 和 v_c 处于同一个概念分段,则利用 EQ 规则预测出的备选服务质量不变;其次,如果 GF 的状态基值 v_b 偏向于其所处区间的某一侧,并且状态实时值 v_c 向其所偏向的一侧变化,则 v_c 必须超过其所偏向的下一个区间的中值才构成一个环境变化事件,这样做是为了防止运行环境因子的状态实时值 v_c 在区间边界间来回跳跃而产生出过多而又意义不大的环境变化事件.

环境变化事件由备选服务运行环境监控器监测,当运行环境因子 GF 的状态实时值 v_c 变化超过边界时,即产生一个环境变化事件并封装成环境变化消息,发送到相应的组合服务端,以驱动服务的运行时自适应,然后将状态基值 v_b 重新设置成 v_c,继续监控.要注意:组合服务执行上下文环境中的环境因子 *Correlation* 和 *TimeSeg* 并不会主动引起备选服务质量的变化,因此其不产生环境变化事件,而仅用于挖掘 EQ 规则以及预测在相应组合背景下的备选服务 QoS.

定义 10(环境变化消息(environment change message,简称 ECM)). 将一个环境变化事件封装而成的数据,可用七元组$\langle Time,ProcessID,ServiceID,A,GF,v_b,v_c\rangle$表示,其中各项的含义分别为:事件产生的时间、事件所属的组合服务、事件所属的备选服务、事件产生所针对的 QoS 属性、产生事件的运行环境因子、运行环境因子的状态基值和状态实时值.

环境变化事件驱动的组合服务运行时自适应的工作流程如图 2 所示.对于组合服务系统,它的主要工作流

程为:

- 针对给定组合服务业务流程 *CS*,维护其执行实例的实时状态信息,即:有哪些 *CS* 的执行实例正在执行,并且各执行实例运行到了哪个抽象服务.
- 同时,它负责从备选服务环境监控器那里接收环境变化消息 *M*,并根据 *M* 中所包含的信息更新备选服务实时状态数据,其中包含各备选服务所处的运行环境实时状态、备选服务 QoS 发布值和基于环境实时状态和 EQ 规则的 QoS 预测值,它是进行组合服务选取、替换和自适应的数据基础.
- 然后,组合服务系统需要判断当前是否有受该环境变化事件影响的执行实例,如果有,则将该消息投递到 *CS* 的环境变化消息队列中.

系统重复进行以上工作,直到运行结束.

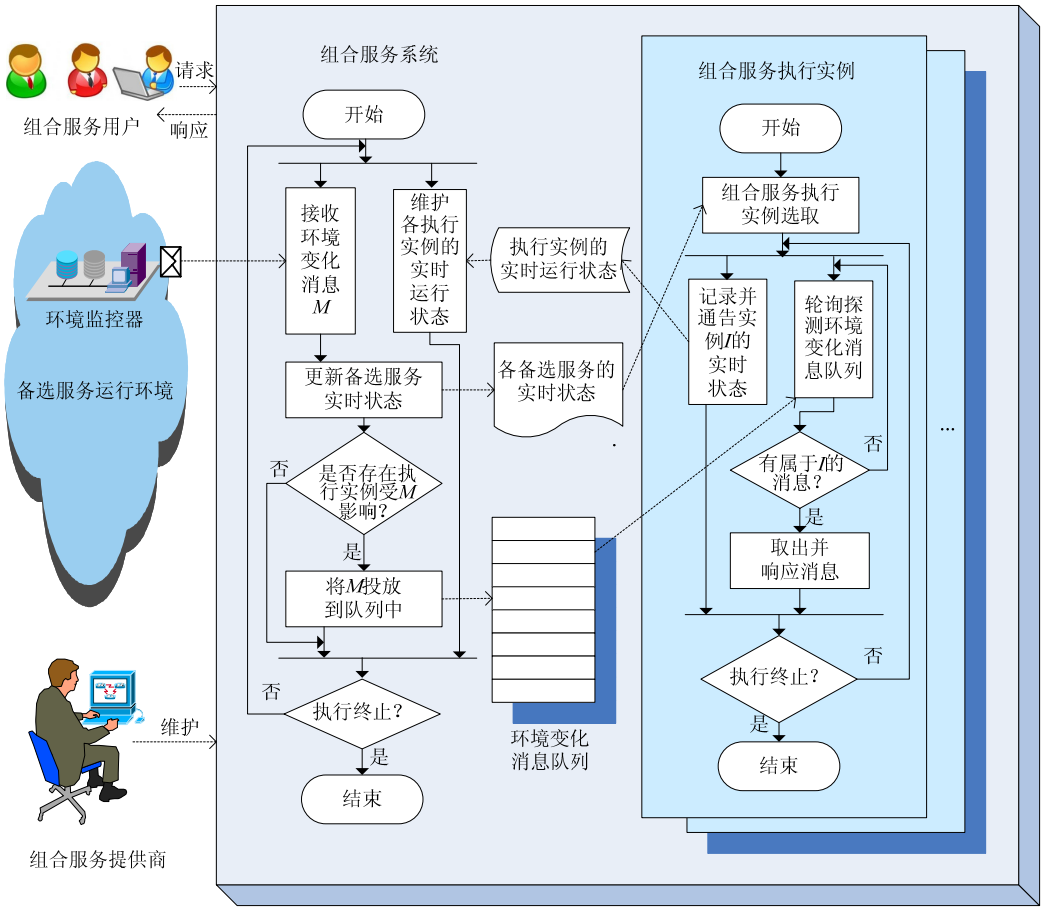


Fig.2 Composite service runtime adaptation workflow driven by environment change events

图 2 环境变化事件驱动的组合服务运行时自适应调节流程

给定组合服务业务流程 *CS* 的一个执行实例 *I*,其工作流程为:

- 首先,基于各备选服务的实时状态数据对实例 *I* 进行选取和初始化.
- 然后,在实例 *I* 的执行过程中记录并通告组合服务系统 *I* 的实时运行状态.
- 同时,轮询探测 *CS* 的环境变化消息队列,如果队列中有属于 *I* 的消息,则一并取出队列中会影响 *I* 运行的所有消息,并删除不再影响 *CS* 其他执行实例的消息,然后基于它们对 *I* 进行运行时自适应调节.

针对执行实例 *I* 的一个环境变化消息 *M*,本文进行组合服务自适应调节的策略是:

- 如果 *M* 所属的备选服务 *s* 在 *I* 中,且 *M* 属于环境因子状态向坏性消息,则触发组合服务重选取,否则不必进行调节;
- 如果 *s* 不在 *I* 中,且 *M* 属于环境因子状态向好性消息,则触发组合服务替换.替换过程中,仅判断用 *s* 替换 *I* 中当前的服务是否性能会更优:如果是,则进行替换,否则,不必进行调节.

本文通过合理确定环境变化事件来对组合服务运行时自适应的代价与收益进行平衡,通过针对不同的环境变化事件触发不同的调节动作来优化自适应的性能,是一种基于知识规则进行预调节的组合服务运行时自适应策略.

6 实 验

为了验证本文组合服务运行时自适应方法的有效性,课题组开发了组合服务运行时自适应原型系统 R-Adaptation,并在此基础上,重点对基于 EQ 规则的备选服务 QoS 预测准确性和环境变化事件驱动的组合服务运行时自适应优化效果这两方面对本文方法进行验证.

6.1 实验环境的搭建

R-Adaptation 系统由在相关课题支持下开发的原型系统基础上发展而来,是为了使用服务组合技术搭建具有环境运行时自适应能力的软件系统提供的具有验证性和参考性的实验平台,在其上可以开发和部署原子服务,建立组合服务的过程模型并选取生成组合服务执行实例,具有监控组合服务运行环境、挖掘原子服务 QoS 和具体环境状态之间的关联知识、对组合服务进行运行时自适应调节等功能.课题组在其上配置了许多实用的或实验性的组合服务.

以 R-Adaptation 系统中包含的前期开发的“旅行计划”组合服务[18]为例,显然,每个抽象服务所对应的任意一个备选服务,比如现实中的“景点搜索”的备选服务 *Expedia*,*CTrip* 或 *Qunar* 等,组合服务端所感知到的它们的 QoS 都受其复杂运行环境的影响,因而需要分析影响备选服务执行质量的复杂环境因素有哪些,以及它们是如何影响备选服务 QoS 的,从而建立本文中 EQ 规则的概念与模型.另外,在能够基于运行环境实时状态准确预测备选服务 QoS 的基础上,需要分析备选服务 QoS 的变化程度以及该采用何种运行时自适应调节动作,因而给出了本文环境变化事件驱动的组合服务运行时自适应方法.R-Adaptation 系统的实现结构如图 3 所示,主要分为原子服务、知识挖掘和服务组合 3 个子系统.

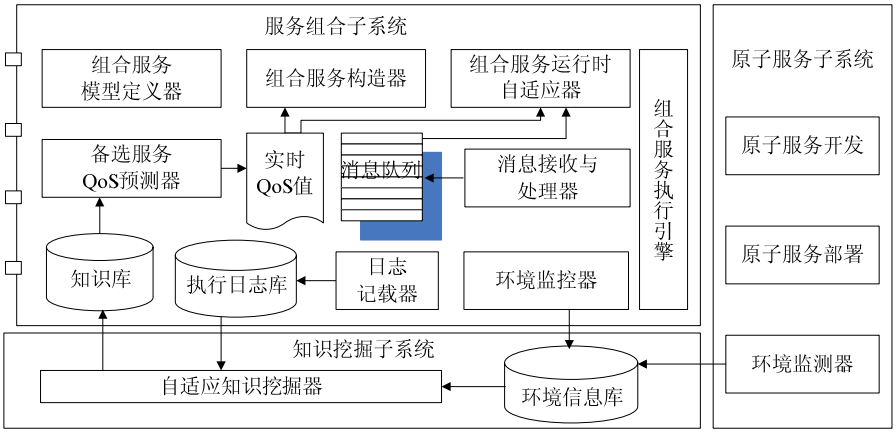


Fig.3 Implementation structure of R-Adaptation system

图 3 R-Adaptation 系统实现结构图

- 原子服务子系统主要是对系统中包含的基本服务进行开发与部署,并配置环境监测器对各服务的主机运行环境进行实时监控.本文生成的原子服务配置在 IIS 中,环境监测器由 MFC 开发,负责根据定义

10 实时产生环境变化消息,并通过 Win Socket 发送给服务组合子系统中的消息接收与处理器.另外,它还负责记载运行环境日志,以备知识挖掘使用.

- 知识挖掘子系统采用 MFC 开发,挖掘算法由 DLL 封装,以便维护和替换.其主要负责基于组合服务执行日志和运行环境日志挖掘得到备选服务 EQ 规则集,以支持组合服务的运行时自适应.
- 服务组合子系统是整个原型系统的核心,采用 WebSASE 和 eclipse 开发.主要包含以下模块:
  1) 组合服务模型定义器:建立组合服务的过程模型;
  2) 组合服务构造器:根据实时 QoS 值选择备选服务,生成初始执行实例;
  3) 组合服务运行时自适应器:为每个组合服务执行实例产生一个对应线程,负责从环境变化消息队列中轮询探测自己所属的消息,并基于第 5.2 节所描述的方法对其进行响应,以对组合服务进行自适应调节;
  4) 组合服务执行引擎:负责解析 BPEL 文档,绑定并调用具体的备选服务;
  5) 消息接收与处理器:负责接收来自环境监测器发送过来的消息,将其投入到消息队列中,并调用备选服务 QoS 预测器对对应备选服务的实时 QoS 进行预测更新;
  6) 备选服务 QoS 预测器:负责基于挖掘得到的知识规则集和运行环境实时状态,以第 5.1 节中描述的方法更新预测各备选服务的实时 QoS 值;
  7) 日志记载器:负责记载组合服务的执行信息,以备知识挖掘使用;
  8) 环境监控器:负责监控组合服务至备选服务间的网络环境和组合服务自身环境,并产生环境变化消息、记载环境日志.

6.2 实验步骤

为了评价本文方法的有效性,首先在 R-Adaptation 系统中设计了包含“旅行计划”组合服务在内的 10 个服务组合流程,每个含有 2~15 个抽象服务,并为每个抽象服务生成不超过 5 个备选服务,将这些总共生成的 275 个备选服务分别配置在不同局域网环境下的不同主机上.

为了收集各备选服务的执行质量数据和运行环境数据,为每个组合服务各形成 5 个执行计划,每个备选服务包含在某个执行计划中.然后针对各执行计划,在 10 天的时间内总共调用 2 000 次,以生成 EQ 规则挖掘的初始数据.为了简化实验操作过程,在调用过程中,各备选服务的参数采用预配置好的方式提供.

在以上收集的原始数据的基础上,进行了针对各备选服务在各 QoS 属性上的初始执行数据集的提取、待挖掘执行数据集的生成和 EQ 规则挖掘的工作.实验进行过多次,最终为这 275 个备选服务在“响应时间”和“可靠性”这两个 QoS 属性上总共生成 18 535 条 EQ 规则,平均每个备选服务在每个 QoS 属性上拥有 33.7 条 EQ 规则.挖掘过程总共耗时 2 小时 35 分.虽然挖掘时间较长,但这是针对系统中的所有组合服务进行挖掘所需要的时间,并且一次挖掘得到的知识可用于长时间的组合服务构造与运行时自适应调节,因此在一定程度上可以忽略.这些挖掘得到的 EQ 规则用来进行下面的实验分析.

6.3 实验结果与分析

(1) 基于 EQ 规则的备选服务 QoS 预测准确性分析

实验首先分析本文基于 EQ 规则预测备选服务 QoS 的准确性.在实验过程中,分别调用数据收集阶段为每个组合服务生成的每个执行计划各 100 次.表 1 以组合服务为单位列出了这 100 次调用各备选服务在各概化 QoS 属性上的预测准确率,即,该准确率为一个组合服务所对应的所有备选服务的预测准确率的平均值.

从表 1 可以看出,各组合服务平均预测出的概化 QoS 值准确率均高于 90%,由此可以判断本文基于 EQ 规则和运行环境的实时状态,能够较准确地预测出各备选服务的 QoS,从而可用来有效地指导组合服务运行时自适应.

(2) 环境变化事件驱动的组合服务运行时自适应的优化效果分析

接下来重点分析本文环境变化事件驱动的组合服务运行时自适应调节的优化效果.因为本文方法可使用

现有的大多数组合服务重选取算法进行自适应调节,不失一般性,实验采用 Zeng 等人提出的较为经典的整数规划组合服务选取算法[2]进行测试.为了进行对比分析,总是为该算法选取出的组合服务执行计划同时启动两个执行实例:一个进行运行时自适应调节,而另一个则不进行这样的操作.实验运行 20 天,每天调用每个组合服务各 50 次,并模拟真实环境每天设置 2.3%的服务失效率.图 4 和图 5 分别给出了各组合服务在所有 20 天调用过程中的平均响应时间和执行成功率.

Table 1 Analysis of GQoS predictive accuracy

表 1 概化 QoS 值预测准确率分析

组合服务	响应时间准确率(%)	可靠性准确率(%)
CS_1	95.7	96.2
CS_2	98.5	97.0
CS_3	92.1	94.6
CS_4	91.8	95.3
CS_5	93.5	96.4
CS_6	95.6	93.7
CS_7	94.2	98.5
CS_8	93.7	96.9
CS_9	95.1	94.8
CS_{10}	96.3	97.5

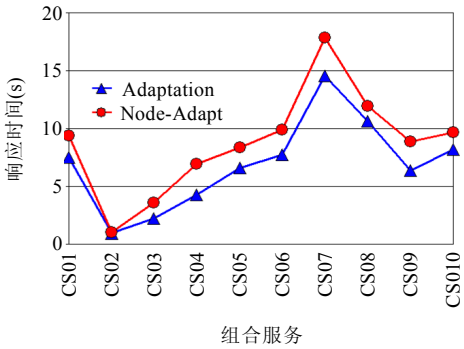


Fig.4 Response time values of composite services

图 4 各组合服务的响应时间取值

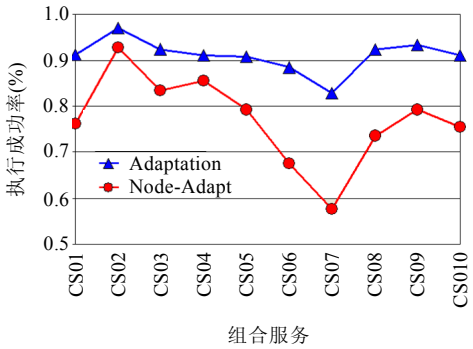


Fig.5 Successful execution ratio values of composite services

图 5 各组合服务的成功执行率取值

在上两图中,Adaptation 表示进行过本文运行时自适应调节后的组合服务执行性能,而 None-Adapt 则表示未进行自适应调节的组合服务执行性能.从图 4 和图 5 可以看出:因为能够基于 EQ 规则较准确地预测出各备选服务的实时 QoS 值并采用合理的自适应调节策略,在进行了环境变化事件驱动的基于 EQ 规则预测的运行时自适应调节后,组合服务的响应时间和执行成功率明显优于未进行运行时自适应调节的组合服务.

图 6 和图 7 分别显示了所有组合服务的平均响应时间和执行成功率在各天上的取值.

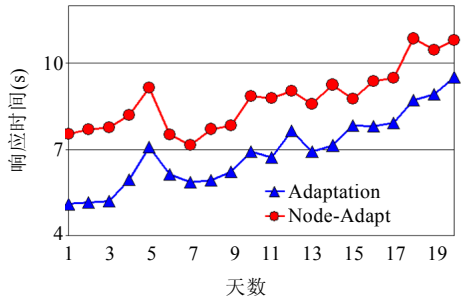


Fig.6 Average response time along with days

图 6 各天的组合服务平均响应时间

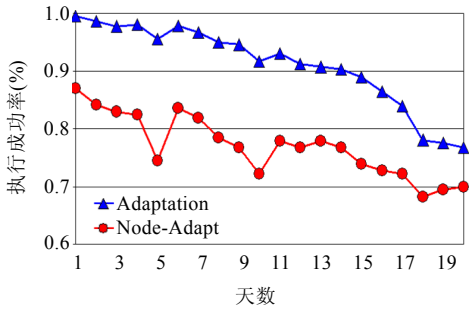


Fig.7 Average successful execution ratio along with days

图 7 各天的组合服务平均执行成功率

由上两图可以看出:在对组合服务进行运行时自适应调节后,无论响应时间还是执行成功率都有明显的提升.但是,随着时间的推移,由于备选服务的动态变化性,有无自适应调节的组合服务性能都有明显下降,特别是基于自适应调节的方法中组合服务的性能下降得更明显,说明本文方法在运行一段时间后,需要重新发掘知识规则,以反映备选服务的最新状态.虽然 EQ 规则的重挖掘需要一定的性能开销,但一次挖掘的结果可应用于较长时间的组合服务运行时自适应调节,所以可在一定程度上忽略其性能开销.

Stephen 等人[4]提出的基于服务的自适应软件系统与本文方法有一定的可比性.他们通过开发一个语音通信服务,围绕着其所提出的 ASQ 方法进行了验证.从实验结果上分析,不同环境状态下语音通信服务的不同 QoS 表现体现了文献[4]和本文研究的必要性.从单个备选服务上看,虽然采用了不同的技术路线,但两者都能较准确地预测出备选服务的实时 QoS 值,但本文方法将环境因素扩展到组合服务层,并就组合服务的运行时自适应效果进行了实验分析.

总之,由以上实验结果可以看出:本文建立的 EQ 规则模型能够较准确地预测出备选服务的实时质量,基于此所提出的环境变化事件驱动的组合服务运行时自适应方法能够提高组合服务在复杂、动态运行环境中的执行性能,减少异常发生的概率,增加组合服务的实际可用性,但本文方法需要初始执行数据的积累和定期的知识挖掘更新.

7 结 论

组合服务运行在开放、复杂和持续变化的环境中,一个能为用户提供 QoS 保障的组合服务一定是一个具有运行时自适应能力的服务.为了使组合服务具备适应其所处动态运行环境的能力,本文从更为实际与可控的基于服务软件系统的角度出发,面向运行环境的分析与建模,提出了一种基于 EQ 规则的组合服务运行时自适应方法.该方法首先分析了潜在影响备选服务 QoS 的运行环境因素有哪些,并建立了备选服务运行环境模型;接下来,通过给出 EQ 规则的概念来分析备选服务 QoS 与运行环境状态之间的内在联系,以用来在组合服务初始选取和自适应调节时,能够根据运行环境实时状态准确地预测出各备选服务 QoS 值;最后,通过合理地定义环境变化事件来驱动组合服务运行时自适应调节的高效运行.文中描述了信息采集阶段的备选服务运行环境监控方法和组合服务执行日志记载方法;详细给出了 EQ 规则挖掘阶段的备选服务初始执行数据集提取方法、数据预处理方法和具体的 EQ 规则挖掘算法;描述了服务运行时自适应阶段的基于 EQ 规则的备选服务 QoS 预测方法、环境变化事件产生及事件驱动的组合服务运行时自适应调节方法.最后的实验分析表明:本文方法能够较准确地预测出备选服务的实时质量,并采用预调节策略,能够较大程度地提高组合服务在动态运行环境中的执行性能.

利用本文方法进行组合服务运行时自适应需要初始数据的积累,包括组合服务的历史执行数据和备选服务的运行环境数据.如何在组合服务运行初期,或者是针对那些执行不频繁、原始数据较少的组合服务进行有效的运行时自适应,是本文下一步的研究内容.其次,组合服务执行过程中往往需要和用户不断交互,可能需要大量时间等待用户输入并满足用户新给出的约束,下一步将研究如何改进本文方法,以能够针对交互式组合服务充分利用交互信息对其进行运行时自适应.最后,本文是从用户角度出发,分析组合服务如何能够在当前运行环境约束下为一个用户提供最为满意的 QoS 保障,而下一步将会从服务提供商的角度出发,研究如何使组合服务能够根据其所处的动态运行环境,合理地为众多用户根据其各自特点调节运行策略,以取得最大的运营收益.

References:

[1] Zheng HY, Zhao WL, Yang J, Bouguettaya A. QoS analysis for Web service compositions with complex structures. IEEE Trans. on Services Computing, 2013,6(3):373−386. [doi: 10.1109/TSC.2012.7]

[2] Zeng LZ, Benatallah B, Ngu AHH, Dumas M, Kalagnanam J, Chang H. QoS-Aware middleware for Web services composition. IEEE Trans. on Software Engineering, 2004,30(5):311−327. [doi: 10.1109/TSE.2004.11]

[3] Gerardo C, Massimiliano PD, Raffaele E, Maria LV. A framework for QoS-aware binding and re-binding of composite Web services. Journal of Systems and Software, 2008,81(10):1754−1769. [doi: 10.1016/j.jss.2007.12.792]

[4] Stephen YS, Nong Y, Hessam SS, Huang D, Auttawut R, Mustafa GB, Mohammed AM. Toward development of adaptive service-based software systems. IEEE Trans. on Services Computing, 2009,2(3):247−260. [doi: 10.1109/TSC.2009.17]

[5] Guo HP, Huai JP, Deng T, Li Y. A dependable and adaptive approach to supporting Web service composition. Chinese Journal of Computers, 2008,31(8):1434−1444 (in Chinese with English abstract).

[6] Leitner P, Hummer W, Dustdar S. Cost-Based optimization of service compositions. IEEE Trans. on Services Computing, 2013, 6(2):239−251. [doi: 10.1109/TSC.2011.53]

[7] Na J, Zhang B, Gao Y, Zhang L, Zhu ZL. Long-Term benefit driven adaptation in service-based software systems. In: Proc. of the IEEE Int'l Conf. on Web Services. Piscataway: IEEE, 2011. 572−579. [doi: 10.1109/ICWS.2011.82]

[8] Harney J, Doshi P. Selective querying for adapting Web service compositions using the value of changed information. IEEE Trans. on Services Computing, 2008,1(3):169−185. [doi: 10.1109/TSC.2008.11]

[9] Li M, Huai JP, Guo HP. An adaptive Web services selection method based on the QoS prediction mechanism. In: Proc. of the IEEE/WIC/ACM Int'l Joint Conf. on Web Intelligence and Intelligent Agent. Piscataway: IEEE, 2009. 395−402. [doi: 10.1109/WI-IAT.2009.363]

[10] Liu H, Zhong FR, Bang Q, Wu JJ. An approach for QoS-aware Web service composition based on improved genetic algorithm. In: Proc. of the Int'l Conf. on Web Information Systems and Mining. Piscataway: IEEE, 2010. 123−128. [doi: 10.1109/WISM.2010.128]

[11] Ziad K, Wang ZY. An adaptive approach for QoS-aware Web service composition using cultural algorithms. In: Proc. of the 20th Australian Joint Conf. on Artificial Intelligence. Berlin: Springer-Verlag, 2007. 140−149. [doi: 10.1007/978-3-540-76928-6_16]

[12] Berbner R, Spahn M, Repp N, Heckmann O, Steinmetz R. Heuristics for QoS-aware Web service composition. In: Proc. of the Int'l Conf. on Web Services. Piscataway: IEEE, 2006. 72−82. [doi: 10.1109/ICWS.2006.69]

[13] Menascé DA, Casalicchio E, Dubey V. A heuristic approach to optimal service selection in service oriented architectures. In: Proc. of the 7th Int'l Workshop on Software and Performance. New York: ACM Press, 2008. 13−24. [doi: 10.1145/1383559.1383562]

[14] da Cruz SMS, Campos MLM, Pires PF, Campos LM. Monitoring e-business Web services usage through a log based architecture. In: Proc. of the IEEE Int'l Conf. on Web Services. Piscataway: IEEE, 2004. 61−69. [doi: 10.1109/ICWS.2004.1314724]

[15] Ringelstein C, Staab S. DIALOG: Distributed auditing logs. In: Proc. of the IEEE Int'l Conf. on Web Services, Piscataway: IEEE, 2009. 429−436. [doi: 10.1109/ICWS.2009.50]

[16] Zhang MW, Wei WJ, Zhang B, Zhang XZ, Zhu ZL. Research on service selection approach based on composite service execution information. Chinese Journal of Computers, 2008,31(8):1398−1411 (in Chinese with English abstract).

[17] Zhu JM, Kang Y, Zheng ZB, Michael RL. WSP: A network coordinate based Web service positioning framework for response time prediction. In: Proc. of the IEEE Int'l Conf. on Web Services. Piscataway: IEEE, 2012. 90−97. [doi: 10.1109/ICWS.2012.81]

[18] Zhang MW, Zhang B, Liu Y, Na J, Zhu ZL. Web service composition based on QoS rules. Journal of Computer Science and Technology, 2010,25(6):1143−1156. [doi: 10.1007/s11390-010-9395-0]

[19] Zhang MW, Zhang B, Zhang XZ, Zhu ZL. A division based composite service selection approach. Journal of Computer Research and Development, 2012,49(5):1005−1017 (in Chinese with English abstract).

[20] Zhang MW, Liu CF, Yu J, Zhu ZL, Zhang B. A correlation context aware approach for composite service selection. Concurrency and Computation: Practice and Experience, 2013,25(13):1909−1927. [doi: 10.1002/cpe.2988]

[21] Barakat L, Miles S, Luck M. Efficient correlation-aware service selection. In: Proc. of the IEEE Int'l Conf. on Web Services. Piscataway: IEEE, 2012. 1−8. [doi: 10.1109/ICWS.2012.62]

[22] Wagner F, Klein A, Klopper B, Ishikawa F, Honiden S. Multi-Objective service composition with time- and input-dependent QoS. In: Proc. of the IEEE Int'l Conf. on Web Services. Piscataway: IEEE, 2012. 234−241. [doi: 10.1109/ICWS.2012.40]

[23] Zahoor E, Perrin O, Godart C. An event-based reasoning approach to Web services monitoring. In: Proc. of the IEEE Int'l Conf. on Web Services. Piscataway: IEEE, 2011. 628−635. [doi: 10.1109/ICWS.2011.97]

[24] Psiuk M, Bujok T, Zielinski K. Enterprise service bus monitoring framework for SOA systems. IEEE Trans. on Services Computing, 2012,5(3):450−466. [doi: 10.1109/TSC.2011.32]

[25] Robinson WN, Purao S. Monitoring service systems from a language-action perspective. IEEE Trans. on Services Computing, 2011, 4(1):17−30. [doi: 10.1109/TSC.2010.41]

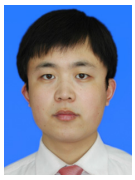
[26] Wu GQ, Wei J, Ye CY, Shao XZ, Zhong H, Huang T. Runtime monitoring of data-centric temporal properties for Web services. In: Proc. of the IEEE Int'l Conf. on Web Services. Piscataway: IEEE, 2011. 161−170. [doi: 10.1109/ICWS.2011.124]

附中文参考文献:

[5] 郭慧鹏,怀进鹏,邓婷,李扬.一种可信的自适应服务组合机制.计算机学报,2008,31(8):1434−1444.

[16] 张明卫,魏伟杰,张斌,张锡哲,朱志良.基于组合服务执行信息的服务选取方法研究.计算机学报,2008,31(8):1398−1411.

[19] 张明卫,张斌,张锡哲,朱志良.一种基于划分的组合服务选取方法.计算机研究与发展,2012,49(5):1005−1017.



张明卫(1979－),男,山东胶州人,博士,讲师,CCF 会员,主要研究领域为服务计算,数据挖掘.

张斌(1964－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为服务计算,信息集成.

朱志良(1962－),男,博士,教授,博士生导师,CCF 会员,主要研究领域为云计算,复杂网络.

张岳松(1986－),男,博士生,主要研究领域为服务计算.