

面向服务 QoS 模型中一种需求映射方法

梁 泉¹ 王元卓²

(福建工程学院计算机系 福州 350108)¹ (清华大学计算机系 北京 100084)²

摘 要 对如何在面向服务网格环境下进行 QoS 需求映射,做了较深入研究,提出了相应的 QoS 模型,并对模型进行了详细描述。该模型紧密结合网格系统管理架构和策略,突出了网格系统的特点。对应用程序 QoS 需求,提出使用 QoS 订单方式提交,在此基础上建立了一个 QoS 约束映射模型,并分析了映射参数之间的关系和确定方法。通过举例进行分析,给出了具体 QoS 映射过程,表明该模型在理论和应用两个方面都具有较高价值。

关键词 面向服务, 网格计算, 服务质量, 需求映射

中图分类号 TP393 **文献标识码** A

One Method of Requirement Mapping Based on Service-oriented QoS Model

LIANG Quan¹ WANG Yuan-zhuo²

(Department of Computer and Information Science, Fujian University of Technology, Fuzhou 350108, China)¹

(Department of Computer, Tsinghua University, Beijing 100084, China)²

Abstract For how to control Quality of Service(QoS) in service-oriented grid environment, the paper conducted a deep research, a QoS model with detailed description was presented. The model was in close connection with the management structure and strategies of grid system, and highlighted its characteristics. In this model, QoS requirement of applications was submitted as the form of an order. Afterwards, a QoS constraint mapping model was established. The paper also analyzed the relationship between parameters and the definite method of parameters in the mapping model. Actual examples gave the concrete process of QoS mapping and its theoretical and applied value for QoS control in grid system.

Keywords Service-oriented, Grid computing, Quality of service, Requirement mapping

服务质量(QoS)控制一直是国内外关于网格计算研究的重点领域,而其中的 QoS 模型及其需求映射又是其中的热点之一^[1]。在面向服务(Service-Oriented Architecture, SOA)架构下, QoS 控制呈现出更高的复杂性。相对于传统的分布式系统,面向服务网格采用抽象资源的概念^[2],用户请求的首先是虚拟资源,再通过虚拟组织映射至实际物理资源。虚拟组织的概念,有利于整合资源的处理过程,也使任务执行、资源调度等都有别于之前的形式,所以相应 QoS 模型及需求映射也将不同。在面向服务网格计算环境下,如何建立有效的 QoS 模型,对应用程序的 QoS 需求进行准确映射或转换,本文针对此展开了研究。

1 网格 QoS 模型及需求映射

关于 QoS 模型及相应的需求映射,专门针对面向服务网格环境的研究很少,过去的一些工作和研究成果多基于不同的领域环境或具体系统,这些领域包括计算机网络、分布式计算、多媒体应用,甚至集成制造等,当然它们都对本文研究有参考价值,但系统不同,差异很大。QoS 模型应融入系统整体框架,需要与任务控制和资源调度等系统策略统一,而不是孤

立的功能模块^[3,4,5]。G-QoSM^[6]遵从面向服务的网格架构,提供面向服务的 QoS 管理模型。在性能变化不确定的系统内,不同的应用处于竞争和共享有限的系统资源并受其变化影响的环境中,考虑到系统资源的全局管理特性,仅从应用中增加适应机制是不够的,还需在系统层中增加 QoS 管理机制,为此给出了一种面向构件系统的 QoS 管理模型,使之能适应于变化的系统环境^[7]。服务组合中 QoS 映射将更复杂,不能直接应用普通的映射手段,而要建立基于服务组合的 QoS 模型及其优化结果,才能确保服务质量。为此,提出支持组合服务选取的 QoS 模型层次结构,包括保证基本服务质量的基本服务 QoS 模型、反映组合服务特性的基本服务间关联 QoS 模型以及全面评价组合服务质量的组合服务 QoS 模型^[8,9]。还有些研究考虑到计算网格的多维度 QoS 问题,建立了基于多维度 QoS 的映射和管理模型^[10]。文献[11]提出了 QoS 模板的概念,模板中包含一些常用 QoS 约束参数,且支持用户按需设置各参数的权值,具有良好的个性化和可扩展性,为 QoS 约束映射提供了良好的基础。在具体 QoS 参数映射方法上,多数是考虑所在系统的管理层次,建立管理层次间的映射描述,或是基于参数转换和计算^[12]。上述已有成果

到稿日期:2009-07-27 返修日期:2009-09-29 本文受国家自然科学基金项目(No. 60803123),福建工程学院科研发展预研基金项目(No. GY-Z0878)和科研启动基金项目(No. GY-Z09009)资助。

梁 泉(1972-),男,博士,讲师,主要研究方向为网络服务质量保障和可信计算, E-mail: liangquanlq@126.com; 王元卓(1978-),男,博士,助理研究员,主要研究方向为网络安全和可信计算。

都在各自领域适用,也面向服务的网格环境 QoS 管理有一定参考价值,但显然受到应用环境的局限而不能直接使用。

2 面向服务的网格系统模型

结合日渐融合的网络服务和 Web Service,我们在另一篇论文中,基于物理网格节点联合和以服务为中心的理念,提出了具体的面向服务网格 QoS 保障的系统模型。该模型被称为 SoGSM-QoS 模型^[13],它遵循 OGSA 的规范,融合了 WS-RF 机制、标准,清楚地反映了网格的层次结构和以服务为中心的网格活动机制、规律。SoGSM-QoS 模型分为 3 个层次:用户和应用程序层、虚拟组织层和服务组织层,在 3 个层次之外,有联系 3 个层次,并为 3 个层次提供信息服务和中介、仲裁的公共平台,即 QoS 网格服务信息中心(QoS GSIC)。SoGSM-QoS 模型的这种安排,也充分体现了网格的一些本质特征,例如动态性、协同性、服务质量(QoS)等。SoGSM-QoS 模型可以在面向服务的体系结构中部署,支持基于 QoS 描述的服务发现、选择和映射,在应用、服务层和中间件层提供 QoS 保障,支持服务等级协定 SLA 协商增强 QoS 保障;基于物理网格节点联合的思想,提出了网格服务组织域的概念,以弥补同类节点管理功能重复、网格服务分散的不足,简化网格系统结构,便于网格服务功能组合、应用和进行网格服务预留,对 QoS 控制十分有利;支持虚拟组织,有利于网格服务选择和任务调度,也使服务质量(QoS)管理更加灵活,易于调整和良好的扩展性;支持服务请求的功能需求和 QoS 需求到网格服务的映射,也支持其从网格服务到物理资源的映射,这使得任务的执行可以在更多地关注 QoS 需求的情况下执行,基于 QoS 属性规范服务质量(QoS)等级,也可以基于 QoS 属性进行服务选择和 SLA 协商,从而保障服务质量(QoS)。

由此可知,面向服务的网格系统中,虚拟组织是一个重要的概念,也可以把它当作一个必不可少的构件。由于虚拟组织的存在,对于网格系统的资源管理、作业调度及 QoS 控制等等重要机制的实现提供了极大的便利,整合了系统的管理和应用功能。另一方面,虚拟组织也为系统的扩展提供了操作平台,资源和节点都可以动态加入和退出,功能也可以动态扩充和缩减,但不影响系统的整体结构。实际上,在具体应用中虚拟组织的作用和功能也发挥得越来越成熟。

3 QoS 描述和映射

3.1 基于虚拟组织的 QoS 模型

从系统的角度看,网格 QoS 管理应包含一些必须操作,譬如资源选择、分配和释放。因此,在一次 QoS 会话中也需要相应的 3 个阶段:第一是建立阶段,二是活动阶段,三是清除阶段^[14]。相对于基本操作和 3 个阶段,管理系统需提供 QoS 描述、映射和协商机制,要实现有效管理,还需要有反馈控制机制。以应用级服务为例,在建立阶段,客户应用程序申明所期望服务和 QoS 规范,系统基于用户 QoS 属性进行服务发现,为之提供可协商的协定。在“活动阶段”,还可能需要其他操作,包括 QoS 监控、反馈控制、记帐和可能的重协商。在“清除阶段”,负责终止会话,终止资源预留、违规处理,等等,从而释放资源,协调 QoS 管理的一致性。

相似的管理模式也应用在其他领域,如计算机网络、多媒体应用和一般的分布式系统,系统有何差别,都需要做到基本

的几个方面,包括:规范的 QoS 需求描述, QoS 需求映射到具体资源的能力, QoS 协商功能,能建立供求双方的服务等级协定(SLA),预留和分配资源,监控 QoS 相关参数,对 QoS 的动态变化做出反应。

SoGSM-QoS 模型的 QoS 管理也同样需要包含上述各阶段及其操作,需要虚拟组织作为事务平台代理或协调 QoS 管理,整合不同实体间的 QoS 映射、QoS 规范及相应的管理策略。从用户期望的 QoS 需求及通过 QoS 管理平台获取系统对其需求的反馈出发,可建立这样有虚拟组织支持的 QoS 模型,如图 1 所示。

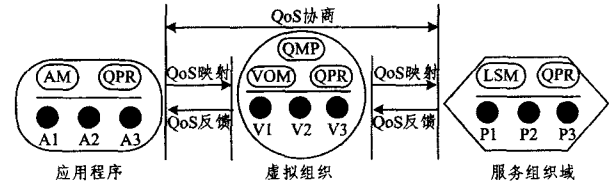


图 1 面向服务的网格 QoS 模型

模型反映了服务需求方与提供方关于 QoS 需求的基本互动关系,应用程序提出期望的 QoS 需求,服务提供方则提供满足该需求的服务,双方为此进行协商,建立合约。应用程序基于相同的规范描述自己的 QoS 需求,A1, A2, A3 分别代表不同的应用程序,它们在自己的域内服从统一的应用程序管理(Applications Manager, AM),专门的 QoS 代理负责应用程序 QoS 的描述、映射、回收及与系统其他实体进行 QoS 协商,至少在同一域内,使用相同的 QoS 策略库(QoS Policies Repository, QPR)。

服务组织域是位置靠近、管理策略相近或结构相似的物理网格节点的联合组织,在同一域内服从统一管理策略,是具体物理资源所在,通过标准的网格中间件(譬如, Globus 等)封装,对外提供各种网格服务。同样,域内进行统一的 QoS 管理,有相同的 QPR,不过与应用程序域所在的 QPR 未必一致,它有自己的管理特点,因此对应用程序 QoS 需求,在映射和反馈过程中需要有解析、转换规则,这些规则包含在自己的 QPR 中,是与应用程序 QPR 的区别之一。服务组织域中 QPR 的存在,有助于整合网格系统中不同域的 QoS 管理,减少系统异构性所带来的影响,通过协调各服务组织域的 QPR 或扩大 QoS 策略规则形成全局统一的 QPR,则可实现全局意义上的 QoS 管理。

虚拟组织充当了应用程序与服务组织域之间的中介平台,本质上它不是实际存在的实体,而是因具体应用的发起而创建的,又因具体应用的完成而消亡的虚拟实体,因此虚拟组织是有生命期的。虽然虚拟组织不是实际实体,但由网格系统全局管理的一个概念,它仍然可起到实体作用,并仍可有实际存在的管理者,同样,在虚拟组织“域”内,亦可建立自己的 QPR,有统一的 QoS 管理规则。虚拟组织充当了服务供求双方的“掮客”,使得服务供求双方获得最大限度的自由,相当于建立了一个自由市场,并同时担当公证角色。不同应用程序 A1, A2, A3 与 V1, V2, V3 以及 P1, P2, P3, 3 个不同实体域可自由映射,极大方便了服务选择和使用。虚拟组织 QPR 接受应用程序的 QoS 映射,并准确传递给服务组织域,因此, QPR 应能理解应用程序 QPR 和服务组织域 QPR 各自的内容,容量至少应超过两者的并集,在虚拟组织层次,最适于建立全局统一的 QoS 策略库。

3.2 QoS 模型描述

应用程序 QoS 需求应有统一的描述规范,要能方便地在网格系统中映射需求,承载反馈信息,能在有限的操作时间和步骤后,与服务提供方完成合约(Contract)签署。因此,合约与 QoS 需求描述最好能统一为一个 QoS 订单,一次协商不仅能完成服务选择,亦能签署合约。服务提供方只要依据 QoS 订单逐一确认“商品”(资源需求)供给即可,当订单中的“商品”均得到确认后返回给应用程序,合约即可签署,整个过程就是 QoS 协商。另外,这样的 QoS 描述要考虑映射是否方便,是否能正确传递所有信息。

定义 1(QoS 订单, QoS Order, QO) 形式上可用一个三元组描述,包括 3 方面的内容,在 QoS 协商中必不可少, $QO = (OrderNumber, QoSContract, ServiceList, QoSConstraint)$, 组元分别表示订单号、QoS 合约、服务列表、QoS 约束。其中, QoS 约束反映了应用程序的 QoS 需求。OrderNumber 在应用程序的活动期限内系统全局统一,这样便于订单的系统全局识别与管理。

定义 2(服务列表, Service List, SL) 形式上是一个数组, $SL[i] = (ServiceID, ServiceType, Value, Location)$, $i \in Z$, 该数组元素包含四元组,分别表示服务标识号,系统全局统一;服务类型,映射为物理资源类型;服务量值,表现为资源的数量;服务所在的位置,由资源所在服务组织域管理。

应用程序的需求提交之后,虚拟组织管理器会创建相应 QoS 代理,负责根据服务列表及 QoS 约束向有关服务组织域分发订单,进行服务发现和选择,所涉及的服务,最终都会映射转换为具体域内的物理资源,返回给虚拟组织。反映了应用程序对资源的需求及其 QoS 约束,譬如对 CPU 资源的需求、数量、提供的位置,等等。这种描述方式也体现了网格系统的特点,即对资源的需求可由系统全局协同,用户无需考虑地理位置、系统异构性等造成的影响。

定义 3(QoS 约束, QoS Constraint, QC) 用一个数组来描述, $QC[i] = (ResourceType, op, AttributeValue)$, 该数组体现了对各种不同类型资源的需求。其中, $ResourceType$ 为资源类型, op 为约束操作集合,令 $op = \{> | \text{值不小于}\}$, 为操作命令集合,表示不能低于指定的 $AttributeValue$ 值; $AttributeValue$ 则表示应用程序的期望 QoS 值,满足此操作的服务才被纳入 ServiceList。

上述 3 个定义表达了应用程序的 QoS 需求,该需求将通过 QO 形式由虚拟组织 QoS 代理分发给众多的服务提供者,分发的过程将遵循虚拟组织 QPR 规则,由相应映射函数实现,同一种资源需求,可能分发给几个有相同服务能力的提供者,由几个提供者共同为应用程序提供服务。初始状态, ServiceList 为空,此时还没有任何服务被选择。系统在进行服务发现和选择时,将不断调用 QoSConstraint, 执行 QoS 约束操作,比较候选资源与应用程序需求是否吻合,在满足约束条件的情况下,结合全局统一的资源管理策略,相应服务将被纳入到 ServiceList。至于 QoSContract 则是系统标准的合约形式,包含了对服务(资源)的描述及相应的使用规则。而当服务选择完成之后,系统把 QO 返回给应用程序方,此时供需双方可确定是否要签署合约,一旦签署, QoS 协商过程就算完成了。需要指出的是,根据应用程序需要使用服务的情况,同一个 QO 除了在提交方(应用程序)外,可能有多个服务方握有

这个 QO,具有全局统一的同一个 OrderNumber,它们均是该 QO 所关联合约的签订各方。

3.3 QoS 约束映射

应用程序 QoS 订单提交后,如何在系统中进行处理是个重要步骤,尤其是在向服务提供方动态传递过程中,如何不丢失 QoSConstraint 中的信息至关重要, QoSConstraint 体现了应用程序的 QoS 需求, QoS 映射也主要是在系统中传递 QoSConstraint 的内容。通常情况下,由于网络的异构性, QoS 映射的主要任务是吧应用程序的 QoS 需求传递、解析给服务所在的管理域,使之能正确理解用户需求。如果是用户和服务提供者所在的域相同,映射的过程就很简单,只是把 QoSConstraint 传递给提供者即可;若网络异构,则需要转换。从理论上,映射过程可以是一个分类处理函数,或称之为 QoS 映射代理(QoS Mapping Deputy, QMP), QMP 存在于虚拟组织中,因此也是随具体应用程序建立而建立的,具有生命周期。

定义 4(QoS 映射函数) 由对应虚拟组织 QMP 传递与转换应用程序 QoS 需求的形式模型,把 QoSConstraint 中的应用程序 QoS 需求转换或传递给服务提供者,形式为三元函数:

$$F(x[i], p_v, p_s) = \alpha \cdot x[i] + \beta \cdot p_v + \gamma \cdot p_s \quad x[i] \in QC, \\ i \in Z$$

式中, α, β, γ 是映射函数的转换因子, p_v 为虚拟组织的 QoS 管理策略变量, p_s 为服务策略变量。

p_v 变量影响虚拟组织对 QoS 需求分发的方式,简单地说,决定是否把需求分发给一个还是几个服务提供者;而 p_s 变量受制于具体服务组织域的管理策略,也即由具体服务组织域的 QPR 决定。 α, β, γ 的取值决定了映射操作的性质,若应用程序所在域与服务提供方所在域没有异构性问题,则可以令 $\alpha = 1, \beta = \gamma = 0$, 映射过程实际上只是做了 QoS 需求的传递;否则, α, β, γ 取值决定于服务组织域网络的动态变化,譬如,对于 α 有:

$$\alpha = f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < +\infty$$

式中, $\mu, \sigma (\sigma > 0)$ 是正态分布参数。但对于映射函数 $F(x[i], p_v, p_s)$, 常需要使用的是 α, β, γ 一种线性组合,该类线性组合仍然服从正态分布,即

$$\sum_{i=\alpha, \beta, \gamma} c_i X_i \sim N\left(\sum_{i=\alpha, \beta, \gamma} c_i \mu_i, \sum_{i=\alpha, \beta, \gamma} c_i^2 \sigma_i^2\right)$$

式中, $c_\alpha, c_\beta, c_\gamma$ 是一组常数,它们的值可分别取为各自对应的数学期望值,即 $E(x[i]), E(p_v), E(p_s)$, 而其期望值又可由经验值或大量实验数据的训练获得,由此可以解得一组在稳态平衡状态下的 α, β, γ 值,从而得到实用的 QoS 映射函数 $F(x[i], p_v, p_s)$ 。

QoS 反馈针对 QoS 订单的发起方而言,订单提交后,凡被纳入 ServiceList 的服务,均被视同同意签署合约的服务提供者,相应的 ServiceList 内容会被刷新,系统以内容刷新后的 QO 反向传递给应用程序方即可。QoS 反馈过程不需要反向使用映射函数,因为应用程序并不关心服务提供者是如何进行的,它只需要知道是否能满足自身的 QoS 需求,如果能,则该服务已在 ServiceList 当中,应用程序只需要考虑其服务费用,以决定是否签署合约。传递回来的 QO 亦可能不是最后完成的版本,若还没有完成,应用程序方可再次将 QO 提交,请求系统再次分发,寻找满足需求的服务,直到 QoSCon-

straint 中所涉及的服务均被纳入 ServiceList 为止。当然也有可能无法完成所有服务的选择,即系统不能提供应用程序的需求,此种情况订单将被系统取消,QoS 协商终止。所以,QoS 反馈过程亦可用来进行 QoS 动态控制,是 QoS 动态协商必不可少的一个环节。

4 分析与讨论

定义 4 是 QoS 约束映射的形式表示,在复杂和异构的网格环境下,屏蔽供求双方网络、服务描述、管理策略等的差异,反映了对 QoS 需求的一致理解,这是消除网络差异、实现网格服务共享的必要条件。该映射形式,同时揭示了网格活动的实质,即异构广域网格环境里任务协同和资源共享,均能实现无缝连接。

从理论上来说,转换因子 α, β, γ 通过其正态分布特征是可以计算的,具体的数值则需要大量实验数据训练得到,或来自于经验值,无论哪种情况,均与具体网格系统有关。而 p_v, p_s 策略变量则由系统管理策略库确定。特别要指出的是,一般情况下 $\gamma=0$,理论上的解释是服务组织域内部的管理策略,只与如何管理具体服务有关,对 QoS 需求映射不产生实质上的影响,这与映射模型是不矛盾的,映射模型反映的是网格活动全过程的各个环节可能产生的影响。任何成熟的网格系统,必有其活动特性,其正态分布特征是可解的;同时,任何网格系统也必有确定的管理策略库,拥有有效的策略规则。由此可知 $\alpha, \beta, \gamma, p_v, p_s$ 均可求解或取得,所以定义 4 中的映射模型亦可求解,是有效的。不过网格环境的动态性决定了 $\alpha, \beta, \gamma, p_v, p_s$ 取值也是动态变化的,因此不同 QoS 需求映射数值上也应是变化的,甚至一次网格活动中的同一 QoS 需求在不同时刻也都会变化,但无论如何变化,均能保持映射形式上的唯一性,这是映射模型的理论意义所在。

具体应用时,由 QoS 映射模型中的各因子 $\alpha, \beta, \gamma, p_v, p_s$, 可进行映射计算。譬如,假定 QoS 需求形式为: $Q=(q_{cpu}, q_{net}, q_{tim}, q_{rel})$,即用户需求表现为 CPU 占用、网络带宽、响应时间、服务可靠度,对应单位分别为多少个 CPU 单位、多少兆带宽资源、毫秒和可靠性百分比。设应用程序 A1, A2 的具体 QoS 需求分别为: $Q_{A1}=(10, 2, 500, 90), Q_{A2}=(8, 6, 400, 90)$ 。

对 A1, $x[0]=10, x[1]=2, x[2]=500, x[3]=90$,令映射函数中 $\alpha=1, \beta=\gamma=0$,则映射以后的数值分别为 $F(x[0], p_v, p_s)=10, F(x[1], p_v, p_s)=2, F(x[2], p_v, p_s)=500, F(x[3], p_v, p_s)=90$ 。从结果看,映射前后的 QoS 值并没有发生任何改变,实际上只是传递给了一个服务提供者。

对 A2, $x[0]=8, x[1]=6, x[2]=400, x[3]=90$,若取 $\alpha=0.5, p_v=2, \gamma=0$,则对 CPU 占用,有 $\beta_1=2, F(x[0], p_v, p_s)_1=4, F(x[0], p_v, p_s)_2=4$,QoS 需求被映射到两个服务,即由两个不同的服务(所在域亦有可能不同)来为应用程序提供 CPU 计算服务;对网络带宽也有 $\beta_2=2, F(x[1], p_v, p_s)_1=3, F(x[1], p_v, p_s)_2=3$,同样由两个不同服务各提供 3 兆的网络带宽服务。至于响应时间和可靠度,则由 p_s 规则,是不可以降格的,因此 $F(x[2], p_v, p_s)=400, F(x[3], p_v, p_s)=90$ 。A2 的实例反映了网格任务的协同性和资源可全局共享的特点。

结束语 本文研究了面向服务网格系统中的 QoS 模型

及其需求映射的形式表示,所提 QoS 模型概括了面向服务架构下网格活动的原理与机制,对于可控服务质量的实现有重要理论和应用意义。而 QoS 需求映射形式模型,更多地体现在维护 QoS 需求在整个网格任务执行活动中的一致性,这对于保障用户 QoS 是至关重要的。给出的形式模型规范了 QoS 需求在活动各个阶段的映射规则,体现了网格系统各个环节对映射的影响,以最大限度地保证应用程序 QoS 需求在映射中的精确性。如何选取更好的 QoS 映射函数并精确求解相关转换因子,在此基础上研究更多的实际应用,是下一步需要做的工作。

参考文献

- [1] 梁泉,杨扬,梁开健,等. 网格环境下的服务质量(QoS)研究[J]. 计算机科学,2006,33(7):42-46
- [2] Amin K, Hategan M, von Laszeswski G, et al. Abstracting the Grid[C]// Proceedings of the 12-th Euromicro Conference on Parallel, Distributed and Network based Processing (PDP 2004). Coruna, Spain, 2004:563-569
- [3] Guennoun K, Drira K, van Wambeke N, et al. A framework of models for QoS-oriented adaptive deployment of multi-layer communication services in group cooperative activities[J]. Computer Communications, 2008, 31(13):3003-3017
- [4] Mantar H A. A scalable QoS routing model for diffserv over MPLS networks[J]. Telecommunication Systems, 2007, 34(3/4):107-115
- [5] Kenneth C, Iman P. QoS-aware model driven architecture through the UML and CIM[J]. Information Systems Frontiers, 2007, 9(2/3):209-224
- [6] Al-Ali R, Amin K, Laszewski G, et al. An OGSA-Based Quality of Service Framework[C]// Proceedings of the Second International Workshop on Grid and Cooperative Computing(GCC2003). Berlin, Germany: Springer-Verlag, 2003:225-233
- [7] 廖渊,淮晓永,李明树. QuCOM:一种面向构件系统的 QoS 管理模型[J]. 计算机研究与发展, 2005, 42(10):1802-1808
- [8] 金海,陈汉华,等. CGSP 作业管理器合成服务的 QoS 优化模型及求解[J]. 计算机学报, 2005, 28(4):578-588
- [9] 代钰,杨雷,张斌,等. 支持组合服务选取的 QoS 模型及优化求解[J]. 计算机学报, 2006, 29(7):1167-1178
- [10] Menascé D A. QoS in Grid Computing [J]. IEEE Internet Computing, IEEE computer Society, 2004 (7/8):85-87
- [11] Aggarwal R, Verma K, Miller J, et al. Constraint driven Web service composition in METEOR-S[C]// Proceedings of the IEEE International Conference on Services Computing. Shanghai, China, 2004:23-30
- [12] Ito Y, Tasaka S. Feasibility of QoS control based on QoS mapping over IP networks[J]. Computer Communications, 2008, 31(10):2589-2595
- [13] 梁泉,杨扬,刘丽. 一个具有服务质量保障的面向服务网格系统模型[J]. 信息与控制, 2007, 36(4):401-409
- [14] Czajkowski K, Foster I, Kesselman C, et al. SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management in Distributed Systems[C]// Proceedings of the 8th Workshop on Job Scheduling Strategies for Parallel Processing. Berlin, Germany: Springer Verlag, 2002:1265-1280