一种工作流环境下能耗感知的多路径服务组合方法

朱 勇[1),2)] 罗军舟[1)] 李 伟[1)]
[1)](东南大学计算机科学与工程学院 南京 211189)
[2)](南京陆军指挥学院教育技术中心 南京 210045)

摘 要 当前,服务组合方法只考虑组合服务 QoS 的优化而不考虑组合服务的能耗优化.针对这一问题,文中首先根据不同情况提出了两种服务能耗模型;其次在基于工作流的服务组合环境下,提出了一种能耗感知的多路径服务组合方法 EAMSC.该方法对服务组合的能耗优化问题进行了数学建模,并提出了一种基于启发式的多路径服务组合算法,该算法包括两个部分:一是组合服务的可行路径查找,即在满足端到端 QoS 约束的前提下找出若干条可行的服务组合路径;二是请求速率的分配,即在可行的服务组合路径上依据服务能耗模型分配请求流量以降低组合服务的总体能耗.最后,仿真实验结果表明:能耗感知的多路径服务组合方法与传统的服务组合方法相比,能够在保证端到端 QoS 约束的基础上有效地减少组合服务的总能耗.

关键词 服务能耗模型;能耗感知的服务组合;多路径服务组合;基于工作流的服务组合;服务负载;绿色计算
中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2012.00627

An Approach for Energy Aware Multipath Service Composition Based on Workflow

ZHU Yong[1),2)] LUO Jun-Zhou[1)] LI Wei[1)]
[1)](*School of Computer Science and Engineering, Southeast University, Nanjing* 211189)
[2)](*Education Technology Center, Nanjing Army Command College, Nanjing* 210045)

Abstract Currently, the approaches to service composition focus on QoS optimization and don't consider the energy consumption on the composite service. First, this paper proposed the two models of service energy consumption for the two different cases; second, the approach for EAMSC (energy aware multipath service composition) was proposed for workflow-based service composition. In this approach, the optimization of energy consumption in service composition was transferred into the mathematical model. The heuristic multipath service composition algorithm was presented, which is including the two parts: one is to find some feasible paths in terms of the end to end QoS constraints; the other is to allocate the request traffic over the feasible paths in terms of the models of service energy consumption in order to reduce energy consumption. At last, the experimental result shows the approach to energy aware multipath service composition can effectively reduce energy consumption in the composite service while respecting the end-to-end QoS constraints.

Keywords model of service energy consumption; energy aware service composition; multipath service composition; workflow-based service composition; service workload; green computing

1 引 言

面向服务的计算已经成为构建网络应用的重要途径.不同服务提供商发布了大量的服务,通过组合这些服务可以构造更加复杂的网络应用,满足网络用户多样化的需求.例如,当用户计划出门旅行时,需要获取相关的服务,如查询天气、预订机票、预订

收稿日期:2011-08-26;最终修改稿收到日期:2012-01-20.本课题得到国家"九七三"重点基础研究发展规划项目基金(2010CB328104)、国家自然科学基金(60903161,61003257,61003311,61070158)、高等学校博士点学科专项科研基金(200802860031,20110092130002)、江苏省自然科学基金(BK2008030)、江苏省网络与信息安全重点实验室资助项目(BM2003201)、计算机网络和信息集成教育部重点实验室(东南大学)(93K-9)资助.**朱 勇**,男,1977年生,博士研究生,讲师,主要研究方向为服务计算和绿色计算.E-mail: zhuyong@seu.edu.cn.
罗军舟,男,1960年生,博士,教授,博士生导师,主要研究领域为下一代网络体系结构、协议工程、网络安全、网格计算和服务计算.
李 伟,男,1978年生,博士,副教授,主要研究方向为下一代网络体系结构、服务计算和网络管理.

酒店以及行程规划等.在实际情况下,这些服务可能隶属于不同的服务提供者,并且存在着大量提供相同或相似功能的服务.服务组合通过选择和聚合合适的服务,并使之协同工作以满足用户的需求.在服务计算领域内服务组合已成为重要的研究方向和研究热点[1].根据文献[2]的观点,服务组合可以分为3类:第1类是基于AI(Artificial Planning)规划的服务组合方法,该类方法主要特点在于仅根据用户的输入和期望的输出自动地生成组合服务,如文献[3];第2类是基于语义的服务组合方法,该类组合方法主要从语义的角度研究服务与需求的匹配以及服务和服务之间的可组合性,如文献[4];第3类是基于工作流的服务组合方法,相比于前两种方法,这种方法应用范围较广.它是在已有工作流模型的基础上根据用户需求,通过服务选择生成组合服务.本文所讨论的服务组合方法属于第3类.

另一方面,在计算机系统与网络性能不断提高以及各种应用日益丰富的情况下,耗电问题已成为网络和信息系统持续发展的障碍[5].如何合理利用各种资源,并在满足用户多样化需求的前提下实现计算机系统与网络的低能耗已经受到广泛关注.以低碳节能为特征的绿色IT在过去的十年受到来自政府、学术界和工业界的重视.为解决这一问题,绿色计算逐步兴起.绿色计算涉及到了计算机系统体系结构、软/硬件设计、制造、部署、运行管理等各个阶段[6].然而,随着面向服务的计算技术的出现和迅速发展,绿色计算的研究内容需要进一步地拓展.在绿色计算的大背景下,本研究以节省服务组合运行过程中的能耗为基本目标,在满足全局QoS需求的前提下,实现能耗感知的服务组合.在当前的应用环境中,节省组合服务运行过程中的能耗至少具有两个方面的重要意义:一方面能够降低服务器自身的能量消耗;另一方面由于服务器的能量消耗最终转化为热能,减少服务的能耗,意味着减少了服务器的散热量,从而进一步减少其它周边设备的投入和运行能耗,如冷却设备.

计算机系统的能耗通常主要涉及3个方面:(1)计算机硬件设备的能耗,这主要取决于硬件的体系结构和电路设计;(2)算法设计与软件运行的效率;(3)系统资源的配置与管理.本文的研究涉及第3个方面.在计算机系统中,能耗与系统资源密切相关,对系统能耗的管理其实质上是对系统资源自身的管理[5].然而,现有的资源管理方法并未从全局层面把能耗作为一种资源进行系统抽象.在能耗节省方面,已有的相关研究大多关注于单个设备或单一系统的能耗问题[7-11],或者仅关注网络的通信能耗问题[12-13].然而在分布式环境下,尤其是面向服务的应用环境下,需要综合考虑网络应用的总体能耗.例如,组合服务的能耗问题.而在服务组合研究方面,尚缺乏关于组合服务能耗的研究,大多相关研究着眼于组合服务的QoS优化[14-19].传统的基于工作流的服务组合方法难以有效地降低组合服务的能耗,其原因主要在两个方面:

(1)缺乏针对服务能耗模型的研究,特别是缺乏服务QoS、服务负载与服务能耗之间关系的研究.而已有的服务器能耗模型[5,10-11]并不能直接用于表示服务能耗;

(2)在服务组合的建模中,大多研究仅关注于针对单个请求的组合服务QoS优化而未考虑处理多个请求的组合服务能耗优化.这可能导致组合服务的QoS较优而能耗较大.

针对当前的不足,本文主要做了以下两个方面的工作:(1)提出了服务的能耗模型,该模型基于M/M/1/PS排队模型描述了服务能耗、负载(即请求的到达速率)和部分QoS属性之间的关系;(2)针对基于工作流的服务组合,提出了一种工作流环境下能耗感知的多路径服务组合方法EAMSC(Energy Aware Multipath Service Composition).不同于传统的服务组合方法仅生成一个QoS优化的服务组合路径,本文方法针对连续到达的多个请求,根据全局QoS约束生成多条服务组合路径,并依据服务能耗模型,有效地在各服务组合路径上分配用户请求的流量,限制服务负载,从而在提供QoS保证的基础上,实现了低能耗地服务组合.

本文第2节介绍相关工作;第3节描述问题的基本模型;第4节提出服务的能耗模型;第5节对能耗感知的多路径服务组合方法进行建模,并提出求解算法;第6节给出了仿真实验,并分析比较实验结果;第7节对全文做出总结并提出下一步的研究方向.

2 相关研究

与本研究相关的工作主要集中在两个方面:一是节省系统能耗;二是服务的选择与组合.

在节省能耗方面,早期的研究集中在单个系统的能耗优化方面.文献[7]通过调整工作电压,在满足QoS需求的前提下,实现设备能耗的节省;文献[8]通过任务调度的方法,根据不同处理器速度下的能耗特征,提出了降低处理器能耗的调度算法.近来的研究逐

渐开始关注分布式环境下的系统能耗问题. 文献[9]和文献[10]分别针对 P2P 环境中服务器的数据传输能耗问题和处理器能耗问题，提出了相应的服务器选择算法. 文献[11]研究了服务器能耗与性能之间的关系，建立了在性能约束下的能耗优化模型，在此基础上，提出了一种优化的负载分配算法；文献[12]针对网络通信所产生的能耗问题，提出了能耗优化的路由算法；文献[13]通过利用速率自适应和设备休眠两种节能技术在不影响性能的前提下有效地降低了网络能耗；文献[20]和[21]研究了数据中心的系统能耗优化问题；此外，文献[22]还分别研究了移动设备管理和组合服务配置对能耗的影响，提出了相应的移动设备应用程序管理算法和组合服务配置算法.

在服务的选择与组合方面，文献[14]将服务组合中的服务选择转化成线性规划模型，通过对该模型的求解生成满足全局 QoS 约束的组合服务. 文献[15]分析了多种工作流模式，提出了一种优化的服务选择算法，该算法主要分为两个阶段，第 1 阶段在局部根据设定的阈值排除一些候选服务，第 2 阶段根据分支定界法，选取全局优化的结果；文献[16]将服务选择问题分别建模表示为 MMKP(Multidimensional Multiple-choice Knapsack Problem)模型和 MCOP(Multi-Constrained Optimal Path)问题，由于这些问题是 NP-难问题，该研究进一步提出了相应的启发式算法实现了基于全局约束的优化服务选择. 文献[17]将用户对组合服务的全局 QoS 约束分解为对服务的局部 QoS 约束，通过选择满足局部约束的服务生成优化的组合服务. 文献[18]在文献[17]的基础上，对候选服务进行筛选，选出支配服务(skyline services)参与下一步的服务选择与组合，从而减少了服务选择的计算开销. 以上的服务选择和组合方法仅仅针对单个用户请求进行服务组合；而在实际应用中，用户请求通常连续到达从而形成了一个或多个流，服务组合需要有效地应对这些请求. 文献[19]研究了基于流的优化服务组合问题，在已知用户请求到达速率的基础上，对来自各个流的用户请求进行统一规划，生成满足用户需求的、QoS 优化的服务组合方案.

3　能耗感知的服务组合基本模型

假设存在一类用户请求由 n 个任务组成，每个任务 $Task_i(1\leq i\leq n)$都对应存在一组能够完成该任务并具有不同 QoS 的服务，这类服务组成的集合称为服务类 S_i. 该集合中的服务 $s_{ij}(1\leq j\leq|S_i|)$能够实现一类特定的任务或业务功能，如查询天气、预订机票等. 其中，i 表示服务类的标识，j 为该服务在服务类 S_i 中的标识. 工作流环境下能耗感知的服务组合问题的基本构成要素可以描述为$\langle WM,\Phi(X_1,\cdots,X_n),QoS_ST\rangle$.

其中，WM 为工作流模型，是由一组服务类 S_i(或任务)构成并且各服务类之间具有明确拓扑关系的组合流程模板. WM 可以用 DAG(Directed Acyclic Graph)图来表示，通过组合不同的服务类可以实现更加复杂的业务功能. 工作流模型通常由专家或业务人员设计制定，图 1 显示了一个为用户提供旅游规划的工作流模型. 其中 S_2 和 S_3 属于并发关系，其它的为顺序关系. 工作流模型为服务组合提供了任务组合模板，因而基于工作流的服务组合过程实际上就是根据用户的全局 QoS 约束，在各相关的服务类 S_i 中选择并绑定一个具体的服务，最终生成一个可执行的组合服务的过程.

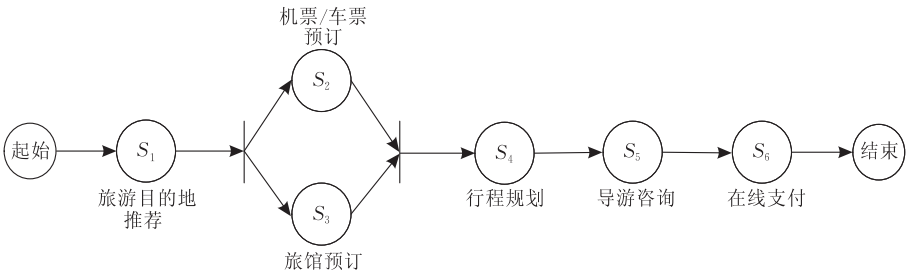


图 1　一个旅游规划的工作流模型

$\Phi(X_1,\cdots,X_n)$为组合服务的总能耗函数. 该函数与各组成服务的能耗密切相关. 其中 $X_1,\cdots,X_n$ 为一组服务组合配置参数，$\boldsymbol{X}_i(1\leq i\leq n)$为一个 m(为简化表示，设 $m=|S_1|=\cdots=|S_n|$)维向量，有

$$\boldsymbol{X}_i=(x_{i1},\cdots,x_{im}) \tag{1}$$

其中，$x_{ij}(1\leq j\leq m)$为指示变量，其值为 0 或 1. 当值为 1 时，表示服务 s_{ij} 参与了服务组合，即 s_{ij} 是一个组合服务中的子服务；当值为 0 时，表示服务 s_{ij} 未参与任何服务组合. 对于一个组合服务而言，有 $|\boldsymbol{X}_i|=1$，即同类服务中有且只有一个服务被选择参

与组合.

QoS_ST 是服务 s_{ij} 的 QoS 模型,可表示为一个 r 维向量 $\boldsymbol{Q}_{ij}=(q_{ij}^{(1)},\cdots,q_{ij}^{(r)})$,其中各分量分别代表一种 QoS 属性,如响应时间、可靠性等. 本文讨论 3 种 QoS 属性,它们是

响应时间 $q_{ij}^{(T)}$:包括了用户请求被处理的时间和用户请求在队列中排队等待处理的时间. 该属性与服务负载等因素密切相关.

成功率 $q_{ij}^{(SR)}$:指服务 s_{ij} 被成功执行的概率. 即在一段时间内,成功完成的请求数量占总请求数量的比重. 本文假设服务的成功率不随服务负载发生变化.

成本 $q_{ij}^{(C)}$:指执行服务 s_{ij} 需要支付的费用.

值得一提的是,引入其它的 QoS 属性并不会改变本文方法的有效性.

用户对组合服务的全局 QoS 需求分别表示为 $Q^{(T)},Q^{(C)},Q^{(SR)}$. 即组合服务的响应时间和成本不超过 $Q^{(T)}$ 和 $Q^{(C)}$;组合服务的成功率不低于 $Q^{(SR)}$.

为实现能耗优化的目标,在服务组合过程中,需要最小化组合服务的总能耗. 因此,能耗感知的服务组合问题的基本模型可以简略地由式(2)表示. 其中,求解目标为最小化组合服务的总能耗函数,约束条件为满足用户的全局 QoS 约束. $G_T(\cdot)$,$G_{SR}(\cdot)$ 和 $G_C(\cdot)$ 分别代表在特定工作流模型下响应时间、成功率和成本的聚合 QoS 函数,该函数实现了组合服务聚合 QoS 的计算. 上述 3 种聚合 QoS 函数的实现方式在文献[23]中有详细的讨论,本文不再赘述.

$$\begin{cases}\min\ \Phi(X_1,\cdots,X_n)\\ \text{subject to}\ G_T(X_1,\cdots,X_n)\leq Q^{(T)}\\ \quad G_{SR}(X_1,\cdots,X_n)\geq Q^{(SR)}\\ \quad G_C(X_1,\cdots,X_n)\leq Q^{(C)}\end{cases}\tag{2}$$

4 服务的能耗模型

服务部署在特定的硬件设备(服务器)上,而服务的运行需要硬件设备和软件平台的支撑,并且产生一定的能耗. 已有的服务器能耗模型不能直接用于描述服务的能耗. 这是因为服务(或服务提供者)需要根据其承诺的 QoS 提供相应等级的 QoS 保证. 例如,服务管理者根据 SLA(Service Level Agreement)设置资源分配. 这意味着服务提供者会根据实际状态(如服务负载)来调整和限制相关计算资源,从而影响了服务的能耗. 本文提出了两种服务的能耗模型用来近似地评估服务运行时的能耗. 为了简化模型,本文做出如下假设:

假设 1. 每个服务都是计算密集型的,主要能耗来自于 CPU 等计算组件.

假设 2. 每个服务可以建模成 M/M/1/PS 排队模型,请求的到达速率为 λ,服务速率为 μ.

单一服务的能耗分为两部分:一是计算环境能耗(或空闲能耗),即在处于空闲状态时支撑服务运转的计算环境所消耗的能量,如维持服务器硬件设备和操作系统正常运转的基本能耗;二是任务能耗,即处理用户请求所产生的能耗,该能耗与服务的动态负载相关. 服务一旦部署并开始运行后,其计算环境能耗不可避免并且相对固定,服务无论是否参与服务组合都不会影响这部分能耗,故计算环境能耗不是本文研究的内容. 而任务能耗与服务负载密切相关进而影响到了服务组合. 因此,服务组合中感知的能耗主要是指任务能耗. 本文后面如无特殊说明,服务能耗均是指任务能耗.

4.1 服务能耗模型 I

在服务能耗模型 I 中,本文假设每个服务均采用自适应服务速率机制来提供服务质量保证. 也就是,服务提供者为了节省能耗和保证服务质量(响应时间)而根据服务的动态负载调整服务处理速率,如采用 DVS(Dynamic Voltage Scaling)来改变服务的处理速度[24].

在 M/M/1/PS 模型中,当服务处于稳定状态时,有

$$RT=\frac{1}{\mu-\lambda},\ \mu>\lambda\tag{3}$$

其中,RT 表示平均响应时间. 当服务负载增加时(如 λ 增大),为保证服务质量,使得 RT 不超过该服务的 $q^{(T)}$(为表述简洁,省略了下标),需要提高服务速率 μ;相反当 λ 减小时,为达到节能效果,可以适当降低服务速率 μ. 在 M/M/1/PS 排队模型下,令 $RT=q^{(T)}$,则 μ 和 λ 之间有如下关系:

$$\mu=\lambda+\frac{1}{q^{(T)}}\tag{4}$$

由于对一个服务而言,其 μ 不可能无限增大,故设服务 s_{ij} 的最大服务速率为 μ_{ij}^{Max},那么在保证服务质量的条件下该服务可接受的最大请求到达速率为 $\lambda_{ij}^{\mathrm{Max}}$. 显然它们之间具有如下关系:

$$\lambda_{ij}^{\mathrm{Max}}=\mu_{ij}^{\mathrm{Max}}-\frac{1}{q_{ij}^{(T)}}\tag{5}$$

根据文献[5],在 t 时刻 s_{ij} 的单位时间任务能耗 $P_{ij}(t)$ 可以简化计算如下:

$$P_{ij}(t)=\beta_{ij}\times\mu_{ij}(t)^{\alpha_{ij}}\tag{6}$$

其中，$\mu_{ij}(t)$表示在 t 时刻 s_{ij} 的服务速率；α_{ij} 和 β_{ij} 为常数，本文称之为能耗参数. 它们的值与具体的服务部署环境有关. 在实际测量中，不同的服务系统表现出不同的单位时间能耗特征. 相关内容将在下一节探讨.

当服务系统处于稳定的工作状态时，服务 s_{ij} 的单位时间能耗 P_{ij} 可以近似表示为

$$P_{ij}=\beta_{ij}\times(\mu_{ij})^{\alpha_{ij}} \tag{7}$$

根据 λ 与 μ 存在对应的约束关系，当服务系统处于稳定的状态时，服务 s_{ij} 的单位时间能耗 P_{ij} 可以表示为 λ_{ij} 的函数：

$$P_{ij}=F(\lambda_{ij})=\begin{cases}\beta_{ij}\times\left(\lambda_{ij}+\dfrac{1}{q_{ij}^{(T)}}\right)^{\alpha_{ij}}, & 0<\lambda_{ij}\leq\lambda_{ij}^{\mathrm{Max}}\\ 0, & \lambda_{ij}=0\end{cases} \tag{8}$$

在时间区间$[t_1,t_2]$内，服务s_{ij} 的能耗$E_{ij}(t_1,t_2)$为

$$E_{ij}(t_1,t_2)=\int_{t_1}^{t_2}P_{ij}(t)\mathrm{d}t \tag{9}$$

当在时间区间$[t_1,t_2]$内，服务处于稳定状态时，服务 s_{ij} 的能耗 $E_{ij}(t_1,t_2)$可以简化表示为

$$E_{ij}(t_1,t_2)=P_{ij}(t_2-t_1)=F(\lambda_{ij})(t_2-t_1) \tag{10}$$

4.2　服务能耗模型 II

在服务能耗模型 II 中，本文假设每个服务采用固定的服务速率 μ，并且通过限制服务的最大请求速率来保证服务质量.

令 $\rho=\lambda/\mu$，表示服务负载率. 在 M/M/1/PS 模型中，为保证服务系统不过载，需保证 $\rho<1$. 而为了保证服务质量，须保证服务请求 λ_{ij} 不超过可接受的最大上限 $\lambda_{ij}^{\mathrm{Max}}$. 根据式(5)，$\lambda_{ij}^{\mathrm{Max}}$计算如下：

$$\lambda_{ij}^{\mathrm{Max}}=\mu_{ij}-\frac{1}{q_{ij}^{(T)}} \tag{11}$$

同时，ρ 为服务处于工作状态的时间占总时间(空闲时间与工作时间之和)的比例. 假设服务处于空闲状态时，其单位时间的任务能耗为 0；那么，服务的期望单位时间任务能耗为

$$\begin{aligned}P_{ij}&=F(\lambda_{ij})=\rho_{ij}\times\beta_{ij}\times(\mu_{ij})^{\alpha_{ij}}\\&=\lambda_{ij}\times\beta_{ij}\times(\mu_{ij})^{\alpha_{ij}-1},\ 0\leq\lambda_{ij}\leq\lambda_{ij}^{\mathrm{Max}}\end{aligned} \tag{12}$$

其中，α_{ij} 和 β_{ij} 为常数(见式(6))．上式也可以简化表示为

$$P_{ij}=\lambda_{ij}\omega_{ij},\ \omega_{ij}=\beta_{ij}\times(\mu_{ij})^{\alpha_{ij}-1} \tag{13}$$

显然，ω_{ij} 为常数. 因而，服务s_{ij} 的能耗 $E_{ij}(t_1,t_2)$可以表示为

$$E_{ij}(t_1,t_2)=P_{ij}(t_2-t_1)=F(\lambda_{ij})(t_2-t_1) \tag{14}$$

4.3　服务能耗的感知方法

为有效地感知服务的能耗，需要根据当前任务负载合理地推断预测服务的单位时间能耗. 在前述的能耗模型中，能耗(即任务能耗)是服务负载的函数. 故在已知服务负载的前提下，服务 s_{ij} 能耗的感知需要获取能耗参数 α_{ij}、β_{ij} 或 ω_{ij}.

在能耗模型 I 中，根据在不同任务到达速率下测量得到的单位时间能耗数据，能耗参数可以用回归分析的方法来测算. 具体方法是根据式(7)，在等式两边取自然对数得

$$\ln P_{ij}=\ln\beta_{ij}+\alpha_{ij}\ln\mu_{ij}=\ln\beta_{ij}+\alpha_{ij}\ln\left(\lambda_{ij}+\frac{1}{q_{ij}^{(T)}}\right) \tag{15}$$

令 $Y_{ij}=\ln P_{ij}$；$B_{ij}=\ln\beta_{ij}$；$A_{ij}=\ln\mu_{ij}$ 则可得

$$Y_{ij}=B_{ij}+\alpha_{ij}A_{ij} \tag{16}$$

将 A_{ij} 视为自变量，Y_{ij} 视为因变量，B_{ij} 和 α_{ij} 视为回归系数，采用一元线性回归方法[25]，可以估算参数 α_{ij} 和 β_{ij}.

在能耗模型 II 中，由于服务速率 μ 固定，故只需要获得服务 s_{ij} 的能耗参数 ω_{ij}. 同理，根据式(13)采用一元线性回归法可以估算 ω_{ij}.

值得注意的是，在上述两个模型的估算中，P_{ij} 是指单位时间内的任务能耗，该值是在实际测量的单位时间能耗基础上减去计算环境能耗(即在服务空闲状态下测出的单位时间能耗)而得出的. 此外，随着时间的推移和计算环境的变化(如服务升级、设备老化等)，能耗参数值(α_{ij}、β_{ij} 或 ω_{ij})可能会发生变化，因此需要定期地更新.

服务通常部署和运行在分布式环境中. 在这种环境中进行服务组合时，存在大量可供选择的服务. 这些服务有着各不相同的能耗特性. 一个服务在不同任务负载下的单位时间能耗也并不相同，而能耗模型和能耗参数能够较好地反映服务在不同负载下的单位时间能耗特性，进而为能耗感知的服务组合提供决策依据.

在本文的服务能耗感知方法中，服务的能耗特性信息由两部分组成，可表示为$\langle EM;EP\rangle$. 其中，EM 为能耗模型；EP 为能耗参数. 例如，服务 s_{12} 采用能耗模型 I，能耗参数为 α_{ij} 和 β_{ij}；服务 s_{13} 采用能耗模型 II 能耗参数为 ω_{ij}. 这些能耗特性信息由服务提供商或第三方在服务描述中说明，并同服务的 QoS 信息一同发布. 这种能耗特性描述方法具有良好的可扩展性. 一方面，该方法可以兼容不同的能耗模型，当出现新的能耗模型时，可以方便地更新现有能耗特性描述信息；另一方面，当已有服务的能耗特性发生变化时，可以通过修改其能耗参数值来实现其能耗特性描述信息的更新.

4.4 进一步讨论

从上面的服务能耗模型可以看出，服务的单位时间能耗(即任务能耗)除了取决于自身的相关参数外，还取决于外部服务请求的负载，即任务到达速率 λ. 所以对于单个服务而言，除了系统自身的改造升级以外，降低能耗的有效办法是减少服务处理负载，也就是降低使用该服务的请求到达速率，特别是当服务能耗随处理负载迅速上升时，减少服务处理负载能够有效地降低服务能耗. 另一方面，每个服务能够处理的最大服务请求速率 $\lambda_{ij}^{\mathrm{Max}}$ 是有限的，为了保证服务质量，也需要限制服务的请求处理负载.

从整个组合服务系统的角度来看，为保证服务质量和减少能耗，需要有效分配服务负载，实现请求流量的合理分流(即采用多路径服务组合的方法)，避免个别服务的过载和高能耗，从而在保证服务质量的前提下，降低整个组合服务的总能耗.

5 能耗感知的多路径服务组合

5.1 基本概念

定义 1. 请求流指在时间维度上依次到达并接受处理的一个用户任务请求序列. 在一个请求流中的每个请求满足以下两个条件：

(1) 共享使用同一个工作流模型，即具有相同的功能需求，如需要旅游规划；

(2) 具有相同的全局 QoS 约束，即具有相同的非功能需求，如所有请求都要求成功率不低于 90%，响应时间不超过 7 s.

定义 2. 服务组合路径(以下简称路径)指为处理用户请求而从起始状态到终止状态构建的一个服务节点处理序列. 显然，这一序列不是唯一的，所以对于同一个请求流而言，存在多个路径. 如图 2 所示，假设工作流模型由 3 个服务类 S_1，S_2，S_3 以顺序结构组成，其中各服务类均包含 3 个服务. 图 2 中显示了两条处理用户请求的路径，即路径 1：start → s_{13} → s_{23} → s_{33} → end；路径 2：start → s_{11} → s_{23} → s_{31} → end. 任何满足用户全局 QoS 约束的路径称之为可行路径. 属于同一个请求流的不同请求可以经过不同的可行路径，例如部分请求可以通过路径 1 完成服务组合；而另一部分请求可以通过路径 2 完成服务组合. 当图 2 的两条路径为可行路径时，这两条路径的端到端 QoS 必须满足用户的全局 QoS 约束，则有

可行路径 1：

$$\begin{aligned} &q_{13}^{(T)}+q_{23}^{(T)}+q_{33}^{(T)}\leq Q^{(T)}\\ &q_{13}^{(SR)}\times q_{23}^{(SR)}\times q_{33}^{(SR)}\geq Q^{(SR)}\\ &q_{13}^{(C)}+q_{23}^{(C)}+q_{33}^{(C)}\leq Q^{(C)} \end{aligned} \tag{17}$$

可行路径 2：

$$\begin{aligned} &q_{11}^{(T)}+q_{23}^{(T)}+q_{31}^{(T)}\leq Q^{(T)}\\ &q_{11}^{(SR)}\times q_{23}^{(SR)}\times q_{31}^{(SR)}\geq Q^{(SR)}\\ &q_{11}^{(C)}+q_{23}^{(C)}+q_{31}^{(C)}\leq Q^{(C)} \end{aligned} \tag{18}$$

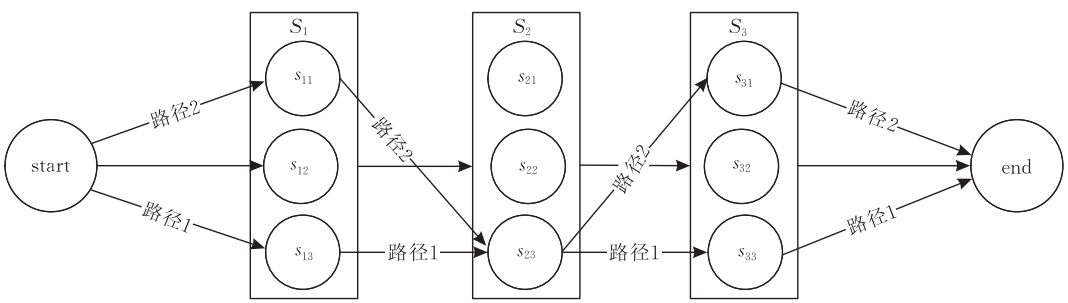


图 2 处理用户请求的服务组合路径

定义 3. 单一路径服务组合指针对一个特定的请求流，在服务组合过程中，构建一个优化的服务组合可行路径，即对于所有到达的用户请求都使用同一个组合服务来处理.

定义 4. 多路径服务组合指针对一个特定的请求流，在服务组合过程中，构建多个服务组合的可行路径，并对于以一定速率连续到达的请求，根据实际环境(如能耗因素或者负载因素)和优化目标在各组合服务路径上分配待处理的请求.

传统的服务组合方法大多属于单一路径的服务组合. 然而，单一路径的服务组合是多路径服务组合的特例. 当服务能耗随负载快速增长时，为了保障服务质量和降低组合服务的总体能耗需要采用多路径的服务组合方法.

5.2 数学模型

工作流模型 WM 一般包含顺序、并发、选择和循环 4 种基本结构. 根据文献[17]，这些结构又可以转化为顺序结构. 本文仅考虑顺序结构的组合服务.

设有一个由 n 个服务类组成的顺序结构的工作流模型(依次为 $S_1\rightarrow S_2\rightarrow\cdots\rightarrow S_n$)，并且每个服务类

有 m 个候选服务. 存在一个请求流 f，f 中的请求以速率为 λ 的泊松流到达，并且用户请求对响应时间、成功率和成本的全局约束分别表示为 $Q^{(T)}$、$Q^{(SR)}$、$Q^{(C)}$. 设 p_k 为一条满足全局 QoS 约束的可行路径，$\Omega(p)$ 表示可行路径 p_k 组成的集合. 本文假设至少存在一条可行路径. 为方便表述，令 $|\Omega(p)|=d$，d 表示最终找到的可行路径数量.

根据本文的服务能耗模型和假设条件，组合服务的单位时间总能耗最小也就意味着总能耗最小. 因此，根据式(2)，模型优化的目标是最小化组合服务的单位时间总能耗. 通过对式(2)的进一步细化，能耗感知的多路径服务组合方法可以由下列数学模型表示：

$$\begin{cases}\min \ \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m}P_{ij}\\ \text{subject to} \ \sum\limits_{p_k\in\Omega(p)}l_k=\lambda,\ l_k\geq 0\\ \quad \sum\limits_{p_k\in\Omega(p)}x_{ijk}l_{ik}=\lambda_{ij}\leq\lambda_{ij}^{\max},\ 1\leq i\leq n;\ 1\leq j\leq m\\ \quad \sum\limits_{j=1}^{m}x_{ijk}=1,\ x_{ijk}=0,1;\ 1\leq i\leq n;\ p_k\in\Omega(p)\end{cases} \tag{19}$$

其中，x_{ijk} 代表一个指示变量，取值为 0 和 1，其值为 1 表示第 $k(1\leq k\leq d)$ 个可行路径 p_k 经过了服务节点 s_{ij}；其值为 0，表示第 $k(1\leq k\leq d)$ 个可行路径 p_k 未经过服务节点 s_{ij}；l_k 表示在可行路径 p_k 上分配的流量(也就是请求速率)；l_{ik} 表示在可行路径 p_k 上到达服务类 S_i 的请求速率(也就是在可行路径 p_k 上到达第 i 个服务节点的请求速率). 由于服务 s_{ij} 在处理请求的过程中，存在一定的失败概率 $(1-q_{ij}^{(SR)})$，本文假设请求处理失败后立即被丢弃，不再被该路径上的后继服务节点所处理. 那么 l_{ik} 由下式计算

$$l_{ik}=\begin{cases}l_k, & i=1\\ l_k\times\prod\limits_{u=1}^{i-1}\sum\limits_{j=1}^{m}x_{ujk}q_{uj}^{(SR)}, & i>1\end{cases} \tag{20}$$

模型的第 1 条约束条件保证了在各条可行路径上分配的请求速率 l_k 之和等于请求流 f 的请求到达速率 λ；第 2 条约束保证了服务 s_{ij} 的请求到达速率不超过 $\lambda_{ij}^{\max}$；第 3 条约束保证了在任意一条可行路径 p_k 上，对于任意一个服务类而言 p_k 只能经过也必须经过属于该类的一个服务节点.

根据前面提到的可行路径定义，任意一条可行路径 $p_k\in\Omega(p)$ 必然满足如下约束：

$$\begin{cases}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m}x_{ijk}q_{ij}^{(T)}\leq Q^{(T)}\\ \prod\limits_{i=1}^{n}\sum\limits_{j=1}^{m}x_{ijk}q_{ij}^{(SR)}\geq Q^{(SR)}\\ \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m}x_{ijk}q_{ij}^{(C)}\leq Q^{(C)}\end{cases} \tag{21}$$

可以看出，该问题为受限的多路径问题(Restricted Multipath Problem)，是 NP-难问题[26].

5.3　启发式的多路径服务组合算法

在前面的问题模型中，x_{ijk} 和 l_k 是待求解的变量. 为了求解该问题，本文提出了一种启发式的多路径服务组合算法，该算法步骤分为两大部分：一是找出若干条可行路径，即求解 x_{ijk}；二是在找出的可行路径上分配请求速率，即求解 l_k.

5.3.1　可行路径查找

根据前面的问题，设工作流执行的顺序为服务类 $S_1,S_2,\cdots,S_n$. 可行路径查找算法 FP 的思想可以简单描述为：按照工作流执行顺序，依次地对各服务节点构建若干条到达该服务的路径，最终构建一组从起始状态到终止状态的可行路径. 由于搜索所有可行路径的开销极大，故在各服务节点中，当可达路径数量超过 z 时按照一定筛选规则只记录 z 条路径. 本文用 $\Omega(s_{ij})$ 表示到达服务 s_{ij} 节点的路径集合. 显然，随着 i 的增加，各服务 s_{ij} 节点需要维护计算的路径数量会指数级增加，集合 $\Omega(s_{ij})$ 也随之急剧膨胀，从而导致了计算量的大幅增加. 本算法通过使用代价函数对各服务节点 s_{ij} 的路径进行评估和筛选，使得各服务节点维护的最大路径数量为 z. 显然，最终找到的可行路径数量 $d\leq zm$.

代价函数由两部分组成：QoS 代价函数 $\xi_{\text{QoS}}(v_{ij})$ 和流量代价函数 $\xi_{\lambda}(v_{ij})$. QoS 代价函数 $\xi_{\text{QoS}}(v_{ij})$ 计算如下：

$$\xi_{\text{QoS}}(v_{ij})=\max\left\{\frac{G_T(v_{ij})}{Q^{(T)}},\frac{G_C(v_{ij})}{Q^{(C)}},\frac{Q^{(SR)}}{G_{SR}(v_{ij})}\right\} \tag{22}$$

其中，v_{ij} 为到达服务 s_{ij} 节点的一条路径，它表示为该路径经过的服务节点组成的有序集合；$G_T(v_{ij})$，$G_C(v_{ij})$，$G_{SR}(v_{ij})$ 分别表示在路径 v_{ij} 上的响应时间、成本和成功率的聚合函数，计算方式如下：

$$\begin{cases}G_T(v_{ij})=\sum\limits_{s_{uy}\in v_{ij}}q_{uy}^{(T)}\\ G_C(v_{ij})=\sum\limits_{s_{uy}\in v_{ij}}q_{uy}^{(C)}\\ G_{SR}(v_{ij})=\prod\limits_{s_{uy}\in v_{ij}}q_{uy}^{(SR)}\end{cases} \tag{23}$$

流量代价函数计算如下：

$$\xi_{\lambda}(v_{ij})=\begin{cases}\dfrac{\lambda-\lambda^{*}}{\lambda}, & \lambda^{*}<\lambda\\ 0, & \lambda^{*}\geq\lambda\end{cases} \tag{24}$$

其中 λ^{*} 为路径 v_{ij} 上能够承载的最大流量，由下式计算：

$$\lambda^{*}=\min_{s_{uy}\in v_{ij}}\{\lambda_{uy}^{\mathrm{Max}}\} \tag{25}$$

因此，总代价函数 $\xi(v_{ij})$ 可表示为

$$\xi(v_{ij})=\theta\xi_{\mathrm{QoS}}(v_{ij})+(1-\theta)\xi_{\lambda}(v_{ij}) \tag{26}$$

其中，$\theta(0\leq\theta\leq1)$ 为权重因子，它表示 QoS 代价在总代价中所占的比例.

算法 1. 可行路径查找算法 FP.

输入：$S_1,S_2,\cdots,S_n$, z, λ, θ, $Q^{(T)}$, $Q^{(SR)}$, $Q^{(C)}$
输出：$\Omega(p)$
算法描述：
/ * 初始化 * /
For $i=1$ to n
  For $j=1$ to m
    $\Omega(s_{ij})=$null;
For $j=1$ to m
  {
    $v_{1j}\leftarrow s_{1j}$;
    $\Omega(s_{1j})\leftarrow v_{1j}$;
  }
/ * 依次构建 $\Omega(s_{ij})$ * /
For $i=1$ to n
{
 For $j=1$ to m
  {
  For each v_{ij} in $\Omega(s_{ij})$
    {
   Compute($\xi_{\mathrm{QoS}}(v_{ij})$); *Compute*($\xi(v_{ij})$);
   If ($\xi_{\mathrm{QoS}}(v_{ij})\leq1$) add v_{ij} to Ψ_{ij};
    }
    $\Omega(s_{ij})=$null;
  If ($|\Psi_{ij}|\leq z$) $\Omega(s_{ij})=\Psi_{ij}$;
    Else
    / * 将代价函数值最小的 z 个 v_{ij} 加入 $\Omega(s_{ij})$ * /
    $\Omega(s_{ij})\leftarrow$the z v_{ij}s in Ψ_{ij} with the minimum $\xi(v_{ij})$;
  }
If ($i<n$)
   / * 预构建 $\Omega(s_{i+1,y})$ * /
   For each $s_{i+1,y}$ in S_{i+1}
     For each s_{ij} in S_i
       For each v_{ij} in $\Omega(s_{ij})$
       {
        $v_{i+1,y}=v_{ij}+s_{i+1,y}$;
        $\Omega(s_{i+1,y})\leftarrow v_{i+1,y}$;
       }
}
/ * 将所有可行路径合并 * /
For $j=1$ to m
  $\Omega(p)\leftarrow\Omega(s_{nj})$;
return $\Omega(p)$

FP 算法主要包括两大部分：初始化部分和 $\Omega(s_{ij})$ 构建部分. 在初始化部分的算法复杂度为 $O(nm)$；$\Omega(s_{ij})$ 构建部分总共需要进行 n 次循环. 在一次循环中，算法主要分为两个部分：一部分是构建当前服务类节点的可达路径集合；另一部分是预构建下一个服务类节点的候选可达路径集合. 在第一部分中生成一个服务的可达路径集合需要最多进行 mz 次操作，而当可达路径数量超过 z 时，需要对这些路径的代价函数值进行排序和筛选. 这一个过程可以采用经典的排序算法，如快速排序. 在这一部分共需要对 m 个服务进行可达路径的构建，故第一部分的算法复杂度为 $O(m^2z\log(mz))$；第二部分由一个三重循环组成，其总循环次数为 m^2z. 因此，FP 算法的总时间复杂度为 $O(nm^2z\log(mz))$.

5.3.2 请求速率分配

请求速率分配的目的是在前面获得的可行路径上分配流量（即请求速率）以避免个别服务过载，保证服务质量并降低组合服务的总能耗. 此时，前面所示的问题可以简化表示成如下数学模型：

$$\begin{cases}\min \quad \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{m}P_{ij}\\ \text{subject to } \displaystyle\sum_{k=1}^{d}l_k=\lambda,\ l_k\geq0\\ \lambda_{ij}\leq\lambda_{ij}^{\mathrm{Max}},\ 1\leq i\leq n;\ 1\leq j\leq m\end{cases} \tag{27}$$

其中，$l_1,\cdots,l_d$ 是该模型的解. 当采用能耗模型 I 时，该数学模型为非线性规划模型；当采用能耗模型 II 时，该数学模型为线性规划模型.

非线性规划工具 Lingo 9.0[①] 是一种优秀的求解数学规划问题的工具. 它既可用于求解非线性规划模型，也可以用于求解线性规划模型. 本算法采用 Lingo 9.0 来实现请求速率的分配.

6 仿真实验

6.1 实验环境

本文通过仿真实验来模拟服务组合环境并验证了能耗感知的多路径服务组合方法的有效性. 实验环境为HP Intel P4 CPU、4GB RAM、Windows 2003、C#. 实验采用经典的、具有代表性的全局优化服务组合方法[14]（以下用 Global 表示）作为能耗节省的比

① http://www.lingo.com/

较基准，并采用混合整数规划（Mixed Integrate Programming，MIP）工具 Lpsolver 5.5[①] 实现该方法；本文所提出的方法（EAMSC）采用 C# 程序和非线性规划工具 Lingo 实现.

本实验采用的工作流模型由 n 个服务类以顺序结构构成，每个服务类有 m 个服务. 实验的主要数据（如 QoS 和能耗参数）是随机生成的. 表 1 显示了相关的实验参数配置.

6.2　实验结果与分析

为了验证能耗感知的多路径服务组合方法的节能效率和计算性能，本文定义了如下评价指标.

定义 5.　能耗比 ER 表示采用能耗感知的多路径服务组合方法（EAMSC）所造成的总能耗与基准能耗（即采用 Global 方法所造成的总能耗）之比.

表 1　实验参数配置

参数	参数值/取值范围
请求到达速率（λ，reqs/s）	20～100
每个服务类的服务数（m）	100～500
服务类数量（n）	10～30
用户全局 QoS 约束（$Q^{(T)}$，$Q^{(SR)}$，$Q^{(C)}$）	$Q^{(C)}$ 和 $Q^{(T)}$ 在[0.5n，n]之间随机生成；$Q^{(SR)}$ 在[0.6，1]之间随机生成，并保证至少存在一条可行路径
服务 QoS（$q_{ij}^{(T)}$，$q_{ij}^{(C)}$，$q_{ij}^{(SR)}$）	均在[0.5，1]之间随机生成
能耗参数与最大请求速率 $\lambda_{ij}^{\mathrm{Max}}$	α_{ij} 在[1.5，3.5]之间随机生成；β_{ij} 在[1.1，1.7]之间随机生成；$\lambda_{ij}^{\mathrm{Max}}$ 在[20，180]之间随机生成；ω_{ij} 可根据式(13)计算
算法参数（z，θ）	z 在[5，25]之间；θ 在[0，1]之间

基准能耗的计算方法是：首先将所有 $\lambda_{ij}^{\max}<\lambda$ 的服务从候选服务集合中剔除；其次，在剩余的候选服务中，采用 Global 方法生成组合服务；最后，计算组合服务的总体能耗得到基准能耗.

显然 ER 小于 1 时，表明本文的服务组合方法在节省能量方面具有优势，并且其值越小越说明本文方法的节能效率越高. 该指标主要受请求到达速率 λ、用户全局 QoS 约束和算法参数（z）的影响，它反映了本文方法的节能效率.

定义 6.　组合成功率 CSR 指采用本文的服务组合方法能够成功生成组合服务方案的概率. 该指标主要受算法参数（z，θ）和用户全局 QoS 约束的影响，它反映了本文方法的性能.

定义 7.　计算时间 CT 指算法生成服务组合方案所需的计算时间. 本文服务组合方法的计算时间主要由两部分组成：一是可行路径查找的计算开销；二是请求速率分配的计算开销. 该指标主要受服务规模（n，m）和算法参数（z）的影响，它反映了算法的性能以及适应较大规模服务环境的能力.

6.2.1　能耗比 ER

实验 1.　请求到达速率 λ 对能耗比的影响.

本组实验模拟仿真了请求到达速率 λ 由 20～100 时的能耗比 ER 变化. 图 3 和图 4 显示了两种不同服务能耗模型在不同 λ 下的能耗比. 从这两个图中可以看出当 z 较大时，其能耗比 ER 也相对较低. 在服务能耗模型 I 中，不同 λ 下的能耗比 ER 在大约 0.35 到 0.8 之间变化；在服务能耗模型 II 中，不同 λ 下的能耗比 ER 在大约 0.35～0.7 之间变化. 这说明本文方法具有较好的节能效率，与传统的服务组合方法相比，降低能耗的效果比较明显. 这是由于本文的服务组合方法考虑了服务的能耗和负载因素，在多个可行的组合服务路径上规划请求流量的分配，从而有效地降低了组合服务的总体能耗.

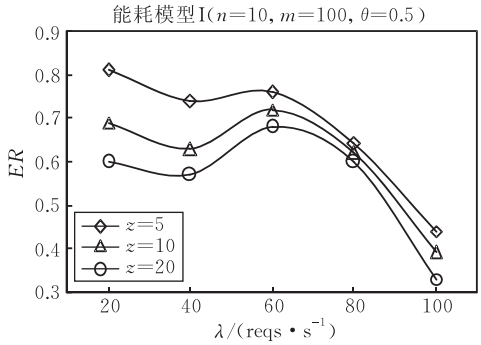


图 3　请求到达速率 λ 对能耗比 ER 的影响（能耗模型 I）

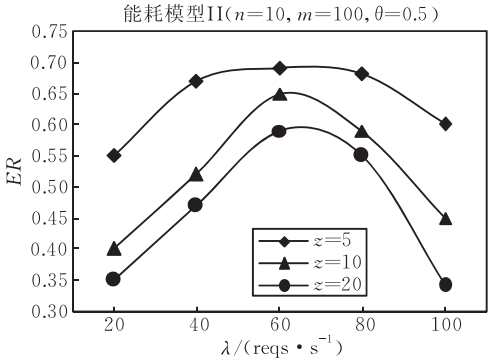


图 4　请求到达速率 λ 对能耗比 ER 的影响（能耗模型 II）

实验 2.　用户全局 QoS 约束对能耗比的影响.

本组实验模拟仿真了在不同用户全局 QoS 约束下的能耗比变化. 由于本文的 QoS 属性有 3 种，本实验仅以其中的响应时间为例来说明用户全局 QoS 约束对能耗比 ER 的影响. 从图 5 可以看出，在相同规模的服务场景中，随着用户对全局响应时间的要求不断降低（$Q^{(T)}$ 逐渐增大），能耗比 ER 逐渐降低. 这说明当用户对组合服务的端到端 QoS 提出

① http://lpsolve.sourceforge.net/

较高要求时，本文方法的节能效果相对不明显. 其原因主要是当全局 QoS 约束较苛刻时，存在的可行路径较少，因此请求可选择的组合服务路径较少，请求流量的分流效果相对不明显，从而造成其能耗比相对较大. 而这又从另一个角度说明了，应通过协商等方式适当降低用户对组合服务的端到端 QoS 需求，避免用户片面追求高 QoS 而带来的不必要能耗.

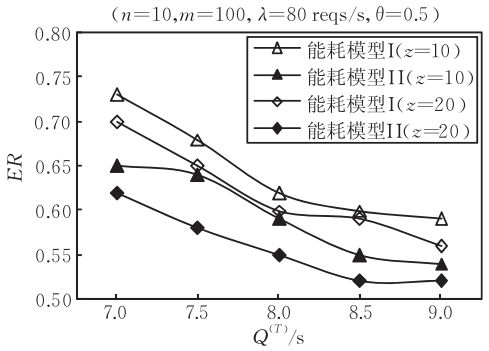


图 5 全局响应时间 $Q^{(T)}$ 对能耗比 ER 的影响

实验 3. 算法参数 z 对能耗比的影响.

本组实验模拟仿真了在不同算法参数 z 下的能耗比变化. 从图 6 可以看出，随着 z 的增加能耗比 ER 逐渐下降. 其原因在于，由于 z 的增长导致算法实际找到的可行路径数量增加，为后面的请求速率分配提供了更多的选择，使请求流量的分流效果更明显，故能耗比会相应的下降. 此外，还可以发现，采用服务能耗模型 II 时的能耗比要低于采用服务能耗模型 I 时的能耗比. 这表明本文所提出的能耗感知的多路径服务组合方法在采用服务能耗模型 II 的组合服务场景中具有更好的节能效果.

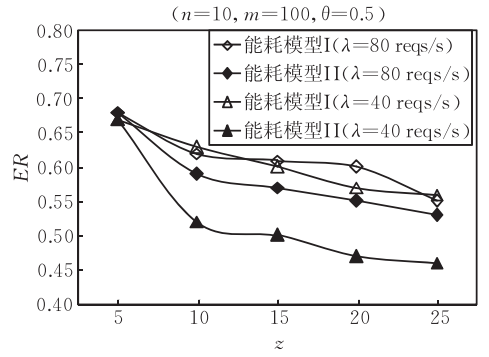


图 6 算法参数 z 对能耗比 ER 的影响

6.2.2 组合成功率 CSR

实验 4. 算法参数(z,θ)对组合成功率的影响.

本组实验测试了在不同算法参数(z,θ)下的组合成功率变化. 图 7 和图 8 所示的组合成功率是在对 100 组随机生成的服务 QoS 数据重复实验的基础上获得的.

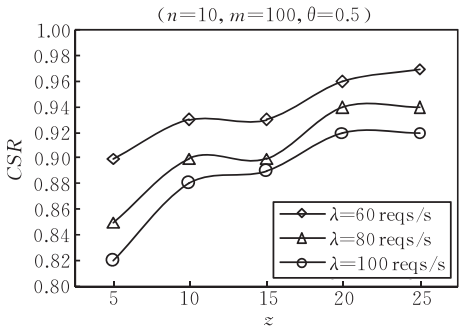


图 7 算法参数 z 对组合成功率 CSR 的影响

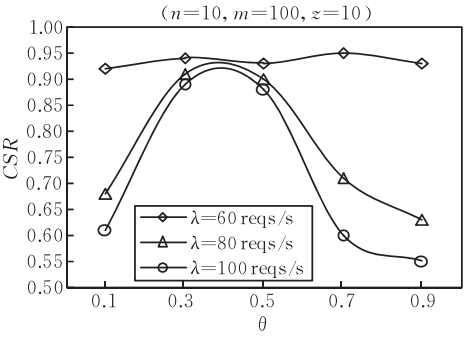


图 8 算法参数 θ 对组合成功率 CSR 的影响

从图 7 可以看出，随着 z 的增长，组合成功率逐渐增加. 这是由于在 FP 算法中，较大的 z 值使得中间的服务类节点可以保存更多的路径信息，这导致了最终找到可行路径的概率相对提高，从而提高了服务组合成功的概率. 此外，当请求到达速率 λ 较大时，组合成功率会相对较低. 这是因为当请求负载较大时，增加了相当一部分服务发生过载的概率从而导致请求流量分配失败.

从图 8 可以看出，不同 θ 会对组合成功率产生一定的影响. 当请求到达速率 λ 较小时($\lambda=60$)，θ 对组合成功率影响较小；而请求到达速率 λ 较大时($\lambda=80,100$)，θ 对组合成功率影响较大. 这是因为当 λ 较小时，大多数服务不会发生过载，因而成本函数中的流量成本为 0 的概率较大，可以忽略. 成本函数主要受 QoS 成本支配，θ 对可行路径查找的影响较小，故 θ 对组合成功率影响较小；而当 λ 较大时，成本函数中的流量成本不能忽略，故 θ 对组合成功率影响较大. 在本组实验环境中，当 λ 较大时 θ 取值在 0.3～0.5 之间可以获得相对较高的组合成功率.

实验 5. 用户全局 QoS 约束对组合成功率的影响.

本组实验测试了在不同用户全局 QoS 约束下的组合成功率变化. 本实验仅以 QoS 中的响应时间为例来说明用户全局 QoS 约束对组合成功率的影响. 图 9 所示的组合成功率是在对 100 个随机生成的服务 QoS 数据进行重复实验的基础上获得的. 从

图 9 可以看出随着对全局响应时间要求的降低，组合的成功率逐渐上升.

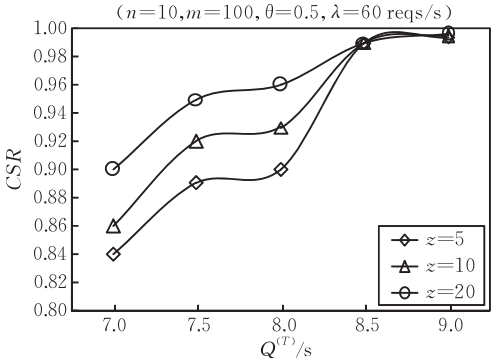


图 9　全局响应时间 $Q^{(T)}$ 对组合成功率 CSR 的影响

6.2.3　计算时间 CT

实验 6.　服务规模对计算时间的影响.

本组实验测试了在不同服务规模 (n,m) 下的计算时间变化. 图 10 显示了在不同 n 时，m 的增长对计算时间的影响. 可以看出随着 m 的增加计算时间开始快速增加. 尤其当 n 较大时，计算时间的增长也较快.

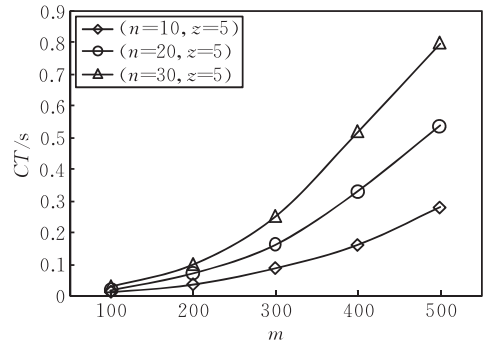


图 10　服务规模 m 对计算时间的影响 CT

实验 7.　算法参数 z 对计算时间的影响.

本组实验测试了本文算法在不同算法参数 z 下的计算时间变化. 图 11 分别显示了在不同服务规模 (n,m) 时，本文算法的计算时间变化. 可以看出，随着 z 的增长，计算时间逐步增加，其增长速度基本呈线性. 当服务规模较大时，其计算时间也增长显著. 可见，较大的 z 虽然有利于提高算法的优化能力，但同时也带来了较大的计算代价，所以需要根据实际情况合理设定 z 值. 例如当服务规模（n 和 m）较大时，可以设定较小的 z 以便计算时间维持在可以接受的范围内.

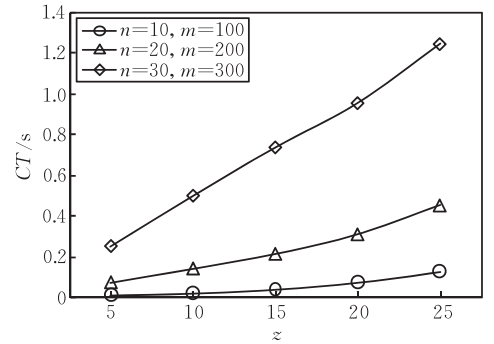


图 11　算法参数 z 对计算时间 CT 的影响

7　结论与下一步工作

本文研究了服务组合的能耗问题，并基于工作流模型和服务能耗模型，提出了一种能耗感知的多路径服务组合方法. 该方法通过对服务组合的能耗优化问题进行数学建模，并提出了基于启发式的多路径服务组合算法. 仿真实验表明本文方法能够在保证端到端 QoS 的前提下有效地节省组合服务的能耗.

本文假设服务为计算密集型，故仅考虑了服务的计算资源所造成的能耗. 下一步将研究组合服务之间数据通信所造成的网络能耗问题.

参　考　文　献

[1] Papazoglou M P, Traverso P, Dustdar S, Leymann F. Service-oriented computing: State of the art and research challenges. Computer, 2007, 40(11): 38-45

[2] Zhang M W, Zhang B, Liu Y et al. Web service composition based on QoS rules. Journal of Computer Science and Technology, 2010, 25(6): 1143-1156

[3] Oh S C, Lee D W, Kumara S R T. Effective Web service composition in diverse and large-scale service networks. IEEE Transactions on Services Computing, 2008, 1(1): 15-32

[4] Medjahed B, Bouguettaya A, Elmagarmid A. Composing Web services on the semantic Web. The International Journal on Very Large Data Bases (The VLDB Journal), 2003, 12(4): 333-351

[5] Lin Chuang, Tian Yuan, Yao Min. Green network and green evaluation: Mechanism, modeling and evaluation. Chinese Journal of Computers, 2011, 34(4): 593-612(in Chinese)
(林闯，田源，姚敏. 绿色网络和绿色评价：节能机制、模型和评价. 计算机学报，2011，34(4)：593-612)

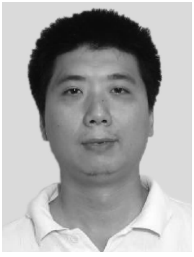
[6] Guo Bing, Shen Yan, Shao Zi-Li. The redefinition and some discussion of green computing. Chinese Journal of Computers, 2009, 32(12): 2311-2319(in Chinese)
(郭兵，沈艳，邵子立. 绿色计算的重定义与若干探讨. 计算机学报，2009，32(12)：2311-2319)

[7] Wong Jennifer L, Qu Gang, Potkonjak Miodrag. Power minimization in QoS sensitive systems. IEEE Transactions on Very Large Scale Integration(VLSI) Systems, 2004, 12(6): 553-561

[8] Frances Yao F, Demers Alan J, Shenker Scott. A scheduling model for reduced CPU energy//Proceedings of the 36th Annual Symposium on Foundations of Computer Science. Milwaukee, USA, 1995: 374-382

[9] Enokido Tomoya, Aikebaier Ailixier, Takizawa Makoto. Energy-efficient server selection algorithms for network applications//Proceedings of International Conference on

Broadband, Wireless Computing, Communication and Applications. Fukuoka, Japan, 2010: 159-166
[10] Enokido Tomoya, Aikebaier Ailixier, Takizawa Makoto. A model for reducing power consumption in Peer-to-Peer systems. IEEE Systems Journal, 2010, 4(2): 221-229
[11] Yang Y, Xiong N, Aikebaier A et al. Minimizing power consumption with performance efficiency constraint in Web server clusters//Proceedings of the 12th International Conference on Network-Based Information Systems. Indianapolis, USA, 2009: 45-51
[12] Andrews M, Anta A F, Zhang L et al. Routing for energy minimization in the speed scaling model//Proceedings of the 29th Annual IEEE International Conference on Computer Communications (INFOCOM 2010). San Diego, USA, 2010: 1-9
[13] Nedevschi Sergiu, Popa Lucian, Iannaccone Gianluca et al. Reducing network energy consumption via sleeping and rate-adaptation//Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation. USENIX Association Berkeley, USA, 2008: 323-336
[14] Zeng Liangzhao, Benatallah Boualem, Ngu Anne H H et al. QoS-aware middleware for Web services composition. IEEE Transactions on Software Engineering, 2004, 30(5): 311-327
[15] Huang A F, Lan C W, Yang S J. An optimal QoS-based Web service selection scheme. Information Sciences, 2009, 179(19): 3309-3322
[16] Yu Tao, Zhang Yue, Lin Kwei-Jay. Efficient algorithms for Web services selection with end-to-end QoS constraints. ACM Transactions on Web (TWEB), 2007, 1(1): 1-26
[17] Alrifai M, Risse T. Combining global optimization with local selection for efficient QoS-aware service composition//Proceedings of the 18th International Conference on World Wide Web (WWW 2009). New York, USA, 2009: 881-890
[18] Alrifai M, Skoutas D, Risse T. Selecting skyline services for QoS-based web service composition//Proceedings of the 19th International Conference on World Wide Web (WWW 2010). Raleigh, USA, 2010: 11-20
[19] Ardagna Danilo, Mirandola Raffaela. Per-flow optimal service selection for Web services based processes. The Journal of Systems and Software, 2010, 83(8): 1512-1523
[20] Rao L, Liu X, Liu W. Minimizing electricity cost: Optimization of distributed internet data centers in a muti-electricity market environment//Proceedings of the 29th Annual IEEE International Conference on Computer Communications (INFOCOM 2010). San Diego, USA, 2010: 1-9
[21] Shang Y, Li D, Xu M. Energy aware routing in data center network//Proceedings of the 1st ACM SIGCOMM Workshop on Green Networking. New York, NY, USA, 2010: 1-8
[22] Eunjeong Park, Heonshik Shin. Reconfigurable service composition and categorization for power-aware mobile computing. IEEE Transactions on Parallel and Distributed Systems, 2008, 19(11): 1553-1564
[23] Jaeger M C, Rojec-Goldmann G, Mhl G. QoS aggregation for Web service composition using workflow patterns//Proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference. Monterey, 2004: 149-159
[24] Bang Sung-Yong, Bang Kwanhu, Yoon Sungroh et al. Runtime adaptive workload estimation for dynamic voltage scaling. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2009, 28(9): 1334-1347
[25] Montgomery D, Peck E. Introduction to Linear Regression Analysis. New York: John Wiley & Sons Inc, 1982
[26] Banner R, Orda A. Multipath routing algorithms for congestion minimization. IEEE/ACM Transactions on Networking (TON), 2007, 15(2): 413-424



ZHU Yong, born in 1977, Ph. D. candidate, lecturer. His current research interests include service computing and green computing.

LUO Jun-Zhou, born in 1960, Ph. D., professor, Ph. D. supervisor. His research interests include next generation network architecture, protocol engineering, network security, grid computing and service computing.

LI Wei, born in 1978, Ph. D., associate professor. His research interests include next generation network architecture, service computing and network management.

Background

This work is supported by the National Basic Research Program (973 Program) of China under Grant No. 2010CB328104, National Natural Science Foundation of China under Grant Nos. 60903161, 61003257, 61003311, 61070158, China Specialized Research Fund for the Doctoral Program of Higher Education under Grant Nos. 200802860031, 20110092130002, Jiangsu Provincial Natural Science Foundation of China under Grant No. BK2008030, Jiangsu Provincial Key Laboratory of Network and Information Security under Grant No. BM2003201, and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grant No. 93K-9.

Green Computing has gained more attention in the past years. Some technologies are proposed to save energy in the field of computer science. However, few researches are involved with the energy consumption of services, especially that of service composition. Currently, the approaches to service composition focus on QoS optimization and don't consider the energy consumption on the composite service. The problem is nontrivial actually. Solving the problem will further facilitate the development of green computing and service computing. Therefore the authors proposed the model of service energy consumption, and then the problem of energy savings in workflow-based service composition was transferred into mathematical model. At last, the approach to energy aware multipath service composition was presented to solve the above problem. The experimental result shows energy aware multipath service composition can effectively reduce energy consumption in the composite service.