

文章编号:1001-9081(2013)01-0239-04

doi:10.3724/SP.J.1087.2013.00239

基于协同过滤的个性化 Web 服务选择方法

贺春林¹, 谢琪^{2,3*}

(1. 西华师范大学 计算机学院, 四川 南充 637002; 2. 西南民族大学 计算机科学与技术学院, 成都 610041;

3. 重庆大学 计算机学院, 重庆 400043)

(* 通信作者电子邮箱 xieqi@cqu.edu.cn)

摘要:对传统的 Web 服务选择算法进行分析,并指出其在动态环境下存在的问题。为了解决该问题,提出了基于协同过滤算法的个性化 Web 服务选择方法,并设计了一个动态的 Web 服务选择架构,架构使用协同过滤算法对 Web 服务的服务质量(QoS)值进行预测并选择最优的 Web 服务以满足用户的需求。使用 150 万条真实 Web 服务的 QoS 数据与其他 4 类算法进行了对比实验,结果表明所提方法的可行性和准确性。

关键词:Web 服务; 服务选择; 协同过滤; 服务质量; 相似性

中图分类号: TP391.1 **文献标志码:**A

Personalized Web services selection method based on collaborative filtering

HE Chunlin¹, XIE Qi^{2,3*}

(1. College of Computer Science, China West Normal University, Nanchong Sichuan 637002, China;

2. School of Computer Science and Technology, Southwest University for Nationalities, Chengdu Sichuan 610041, China;

3. College of Computer Science, Chongqing University, Chongqing 400043, China)

Abstract: The traditional Web services selection algorithms were analyzed and the problems existing in dynamic environment were pointed out. A personalized Web services selection method based on collaborative filtering was proposed to address these problems. And a personalized Web service selection framework was designed, which used the collaborative filtering to predict the Quality of Service (QoS) and selected the best service that met users' requirements. About 1.5 million real world QoS data were employed to evaluate the proposal with other four methods and the experimental results demonstrate that the proposed method is a feasible manner and it provides better prediction results.

Key words: Web service; service selection; collaborative filtering; Quality of Service (QoS); similarity

0 引言

面向服务的体系架构 (Service-Oriented Architecture, SOA)已经成为商业和企业系统中多种分布式应用的重要软件架构^[1]。SOA 具有松散耦合、按需集成的重要特征,这些特征使 Web 服务获得了快速发展。随着复杂多样的分布式系统对 Web 服务需求的不断增长,Web 服务选择在 Web 服务领域内具有越来越重要的位置。在服务的选择过程中不仅需要考虑服务质量 (Quality of Service, QoS) 的功能属性也需要考虑其非功能属性。其中,服务质量的非功能属性主要包括失效率、吞吐率、响应时间、可用性和可靠性等^[2]。具有相似功能属性的 Web 服务可能具有不同的非功能属性。此外,Web 服务的 QoS 属性值可能会因为网络延时、调用失效等因素而发生变化。Web 服务环境的动态性给 Web 服务选择带来了诸多问题。由于服务提供商不能保证 Web 服务拥有它所声称的服务质量值,当调用一些 QoS 值高度依赖于服务所处的地理位置和网络条件的 Web 服务时,这些服务的 QoS 值可能会发生变化,一些服务可能会失效或变为不可用^[3]。极端情况下会导致整个业务流程的失效从而造成巨大的经济损失。

在动态环境下,如何从诸多满足功能需求的候选 Web 服务中为用户选择最优的 Web 服务已经成为服务选择领域内迫切需要解决的难题。本文提出了一种基于协同过滤的个性

化 Web 服务选择算法,并使用个性化 Web 服务选择架构为分布式 Web 服务提供 QoS 等级管理。该架构能自动地为当前用户预测满足需求的候选 Web 服务的 QoS 属性,然后把具有最优预测值的候选服务提供给当前用户的组件服务。该算法将协同过滤理论应用于 Web 服务选择领域以解决动态环境下目前服务选择算法的不足。

1 服务选择算法及其局限性

服务选择是在一组候选服务中挑选出满足功能性需求的服务。在众多具有相同功能属性而具有不同非功能属性的候选 Web 服务中,必须根据优化标准选出一组最优的服务。这些优化标准可以是选择具有最小代价的服务,或是具有最低失效率的服务。从 QoS 最优化角度出发,服务选择可以分为局部最优的服务选择和全局最优的服务选择。局部最优是指为每个任务选择一个最优 QoS 的 Web 服务^[4-6]。与局部最优相比,全局最优是指为每个任务选择服务以获得全局最优的质量^[7-10]。目前基于 QoS 的服务选择算法主要采用整型规划、图论和遗传算法等。例如:Yu 等提出的端到端约束的服务选择算法^[1],Zeng 等提出的质量感知的 Web 服务组合算法^[11]等。这些算法在提高服务选择效率上取得了很好的效果,但这些算法在服务选择过程中没有考虑到 Web 服务的动态环境,在动态环境下 Web 服务的 QoS 值可能会发生变化。

一个典型的简单 Web 服务选择场景如图 1 所示,其中包

收稿日期:2012-07-31;修回日期:2012-09-10。 基金项目:国家自然科学基金重点项目(90818028);西华师范大学校级项目(08B009)。

作者简介:贺春林(1971-),男,四川广安人,教授,主要研究方向:计算机网络安全; 谢琪(1983-),女,四川简阳人,博士,主要研究方向:服务计算、个性化服务推荐。

含 5 个组件服务 $S = \{S_1, \dots, S_5\}$ 。组件服务是满足逻辑和业务需求的虚拟服务模块,每个组件服务 S_i 可以被任意的真实服务(S_{ij})执行,这些真实服务是具有相同功能属性和不同 QoS 属性的一组服务。Web 服务执行计划可通过组合被选择的服务($s_{11}, s_{23}, s_{32}, s_{43}, s_{57}$)以实现组合服务的需求。

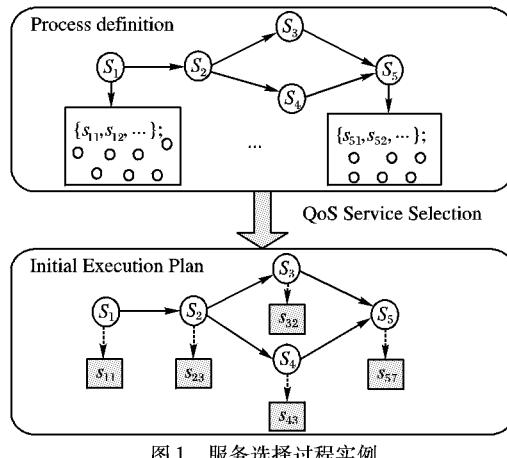


图 1 服务选择过程实例

在图 1 中,每个组件服务可以从一组满足用户需求的候选服务中选择一个真实的服务。假设组件服务 S_2 可调用的候选服务 s_{21}, s_{22}, s_{23} 的失效率分别是 10%, 1%, 0。在常用的 Web 服务选择算法中,不论候选服务的 QoS 值是否发生变化,具有最低失效率值的服务 s_{23} 将被选为组件服务 S_2 的执行服务。但是由于在动态环境下 Web 服务的 QoS 值可能会与服务提供商提供的 QoS 值不相同,真实服务 s_{23} 可能会变为不可用。以往静态的服务选择方法已经不能提供最优的服务选择方案,仅仅考虑当前 Web 服务的 QoS 属性值可能会导致组件服务的失效,更严重的后果可能会使得整个业务流程失效。

2 个性化 Web 服务选择算法

2.1 服务选择架构

如图 2 所示,个性化 Web 服务架构主要由以下 5 个重要步骤构成。

- 1) 查询。用户提交 Web 服务请求,用户代理查询是否存在满足其需求的 Web 服务。
- 2) 功能匹配。服务发现机制检索满足用户功能需求的候选 Web 服务。
- 3) 相似服务发现。使用相似性算法为候选服务提供相似服务。
- 4) 候选服务 QoS 值预测。使用协同过滤算法计算当前用户候选服务的 QoS 值。
- 5) 排序和选择。候选服务根据预测的 QoS 值和用户的需求进行排序,然后把最优的候选服务提供给用户,最后形成最优的选择结果。



图 2 个性化 Web 服务选择架构

该架构的目的是在服务调用之前帮助用户为其组件服务选择最优的服务。在该架构中服务选择是在最后可能的时刻执行。此外,服务选择采用局部最优的方式。该架构的核心问题是如何准确有效地对候选服务进行 QoS 值预测。

在使用协同过滤方法对候选服务进行预测之前,需要获得服务历史的 QoS 调用记录。该 QoS 值可以通过监控系统

获得。因此,本文提出的方法适用于已知历史 QoS 属性值的 Web 服务,而不适用于不能获得具体 QoS 属性值的 Web 服务,如 QoS 属性值是由时间表示的函数。

2.2 相似性计算

在个性化协同过滤推荐系统中,度量相似性的方法有很多。其中最常用的相似性度量方法包括皮尔森(Pearson)相关性、余弦相似性和调整余弦相似性。这些相似性度量方法可分为基于用户的相似性计算和基于项目的相似性计算。其中基于用户和项目的皮尔森相关系数计算公式^[12]分别如式(1)~(2)所示:

$$\text{Sim}(u, v) = \frac{\sum_{i \in I(u) \cap I(v)} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I(u) \cap I(v)} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I(u) \cap I(v)} (r_{v,i} - \bar{r}_v)^2}} \quad (1)$$

$$\text{Sim}(i, j) = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,j} - \bar{r}_j)^2}} \quad (2)$$

其中: u 和 v 表示用户, $\text{Sim}(u, v)$ 表示用户 u 和用户 v 的相似性, $I(u)$ 表示 u 的已知评分项, $I(u) \cap I(v)$ 表示 u 和 v 共同评分的项目组。同理, 在式(2)中: \bar{r}_i 和 \bar{r}_j 分别表示项目 i 和项目 j 的平均评分; $U(i)$ 包括所有对项目 i 评过分的用户, 形式化定义为 $U(i) = \{u \mid r_{u,i} \neq 0\}$, 用户 u 是属于同时对项目 i 和项目 j 评分的用户集。

在使用相似性对目标用户进行预测时,基于用户和基于项目的预测方法也相似。如式(3)所示, $P_{u,i}$ 表示目标用户 u 对项目 i 的评分预测, \bar{r}_u 表示用户 u 对已知的所有项目的评分均值, \bar{r}_v 则表示用户 v 的评分均值。基于项目的预测方法如式(4)所示:

$$P_{u,i} = \bar{r}_u + \frac{\sum_{v \in \text{Sim}(u)} \text{sim}(u, v) \times (r_{v,i} - \bar{r}_v)}{\sum_{v \in \text{Sim}(u)} \text{sim}(u, v)} \quad (3)$$

$$P_{u,i} = \bar{r}_i + \frac{\sum_{j \in \text{Sim}(i)} \text{sim}(i, j) \times (r_{u,j} - \bar{r}_j)}{\sum_{j \in \text{Sim}(i)} \text{sim}(i, j)} \quad (4)$$

其中 $\text{Sim}(i)$ 包括项目 i 的所有相似项目 j 的集合。

基于用户相似性的计算和基于项目相似性计算的区别在于:基于用户相似性是基于评分矩阵中代表用户的行向量相似度求解;而基于项目相似性计算则是使用评分矩阵中代表项目的列向量相似度求解。

2.3 候选服务 QoS 值预测

在本文中使用的协同过滤算法是基于项目的皮尔森相关系数 (Item-based collaborative filtering method using Pearson Correlation Coefficient, IPCC) 算法。传统的协同过滤算法是对用户的评分进行计算,而在本文中使用协同过滤则是对 Web 服务的 QoS 属性值进行计算,这些属性值可以是失效率、响应时间和吞吐率等。

由于式(2)中的相似性计算可能会过高地评估实际并不相似的服务,为了解决这一问题,本文使用相似性权重算法计算其相似性^[3]。不同 Web 服务之间的相似性权重计算公式如式(5) :

$$\text{Sim}'(i, j) = \frac{2 \times |U_i \cap U_j|}{|U_i| + |U_j|} \text{Sim}(i, j) \quad (5)$$

其中 $|U_i \cap U_j|$ 是同时调用 Web 服务 i 和 Web 服务 j 的用户数目。相似性 $Sim'(i, j)$ 的取值范围为 $[-1, 1]$ 。

此外,在为用户预测指定服务的 QoS 值时,预测 QoS 值可能会出现负数。为了缓解负数值对预测性能的影响,本文对传统的基于项目的协同过滤(IPCC)算法进行改进。当预测的 QoS 值为负时,使用项目平均(Item-mean, IMEAN)的方法计算。改进后的 IPCC(Enhanced IPCC, EIPCC)算法预测公式如式(6)所示:

$$p_{u,i} = \begin{cases} \bar{r}_i + \frac{\sum_{j \in Sim(i)} sim'(i, j) \times (r_{u,j} - \bar{r}_j)}{\sum_{j \in Sim(i)} sim'(i, j)}, & p_{u,i} \geq 0 \\ \bar{r}_i, & p_{u,i} < 0 \end{cases}$$
(6)

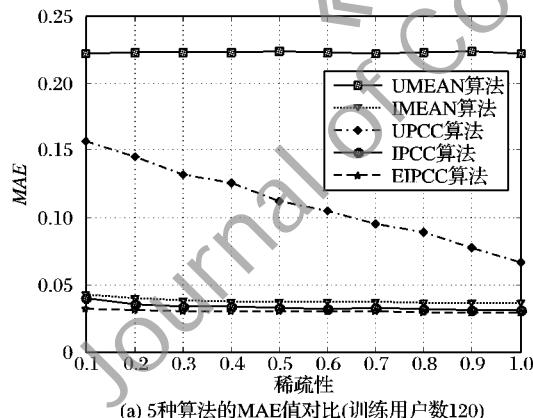
2.4 最优服务选择

通过式(6)可以获得候选服务(s_j)的预测 QoS 值,这些预测的 QoS 值能更准确地反映当前服务实际的 QoS 值。在个性化 Web 服务选择阶段,组件服务(S_i)其候选服务的预测 QoS 值将作为服务选择排序的主要指标。服务用户可以根据自己的需求选择具有最优预测 QoS 值的候选服务。在本文中仅考虑服务选择的局部最优,因此在第 1 章中假设候选服务 s_{21}, s_{22}, s_{23} 预测的失效率值分别为 6%, 1%, 80%。在此情况下,由于候选服务 s_{23} 的失效率值已经变为 80%,服务 s_{23} 将面临随时可能失效的危险。为了保障组件服务功能的正常运行,候选服务 s_{22} 将替代服务 s_{23} 作为组件服务 S_2 的执行服务。

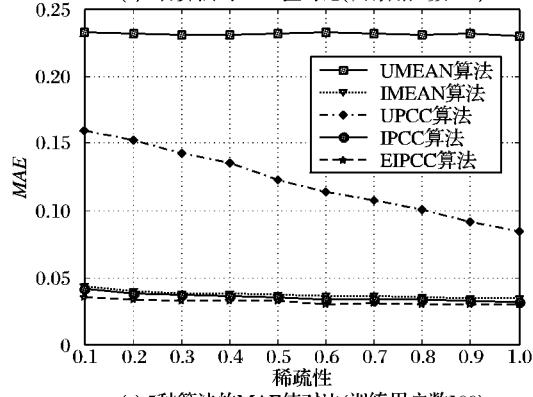
3 实验结果与分析

3.1 实验设置

在该实验中使用大规模真实环境下收集到的 QoS 数据



(a) 5种算法的MAE值对比(训练用户数120)



(c) 5种算法的MAE值对比(训练用户数100)

集(WS-DREAM)^[3,13],该数据集收集了分布在全球 20 多个国家的 150 个服务用户调用 100 个服务的 150 万条调用记录。从该数据集中可以提取 150 × 100 的用户-服务矩阵,矩阵中的每项代表失效率和响应时间。本文讨论失效率在服务预测中的性能,实验过程中使用的评价指标如式(7)所示:

$$MAE = \frac{\sum_{i,j} |r_{i,j} - \hat{r}_{i,j}|}{N}$$
(7)

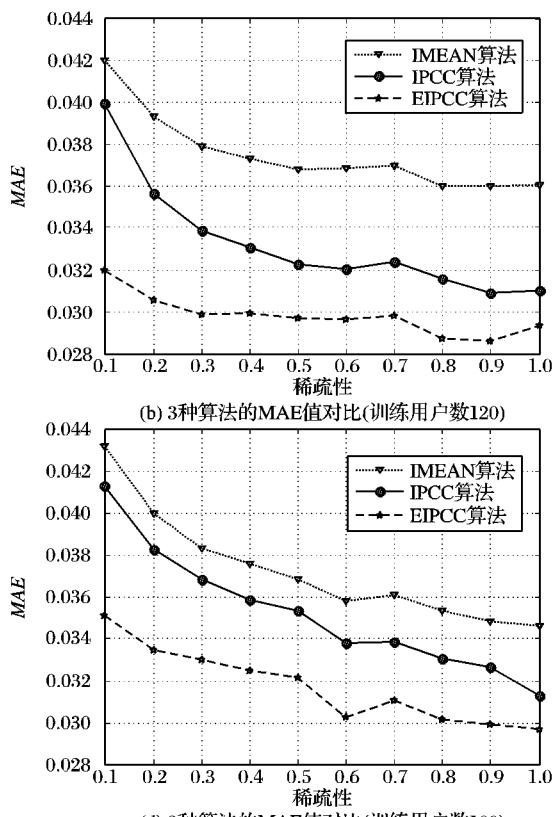
平均绝对误差(Mean Absolute Error, MAE)是通过比较用户预测的评分与用户实际的评分之间的差异来计算。其中: $r_{i,j}$ 表示用户 i 对项目 j 的实际评分,而 $\hat{r}_{i,j}$ 表示用户 i 对项目 j 的预测评分, N 表示所有预测值的数目。 MAE 值越小表明预测性能越好。

为了更加详细地分析实验预测结果,实验中,将 EIPCC 算法与其他广泛使用的 4 类协同过滤算法相比较,这 4 类算法分别是用户平均(User-mean, UMEAN)方法、项目平均(IMEAN)方法、基于用户的皮尔森相关系数(User-based collaborative filtering method using Pearson Correlation Coefficient, UPCC)算法、基于项目的皮尔森相关系数(IPCC)算法。

实验中将用户-服务矩阵中的 150 个服务用户任意地划分为两个部分:一部分作为训练用户集;另一部分作为测试用户集。在训练用户集中的用户数目被称为训练用户数。为了模拟 Web 服务的真实应用场景,任意地以矩阵稀疏性 0.1 到 1 的比例从训练用户集中移除属性实体。对于测试集中的服务用户,移除的实体数目从 10,20 变化到 30,称为服务数。然后将该服务数包含在训练用户集中进行预测。

3.2 对比实验

实验结果如图 3 和表 1 所示。



(b) 3种算法的MAE值对比(训练用户数120)

(d) 3种算法的MAE值对比(训练用户数100)

图 3 训练矩阵稀疏性的影响

其中图 3 表示了 5 种不同算法获得的预测 QoS 值的 MAE 值。在该实验中训练数据集的稀疏性从 0.1 以 0.1 为步长递增到 1,以此研究训练数据集的稀疏性对预测性能的影响。图 3(a)和图 3(b)表示当训练用户数为 120 服务数为 30 时,失效率在不同稀疏性情况下的 MAE 值对比结果。同理,图 3(c)和图 3(d)表示当训练用户数为 100 服务数为 30 时,失效率在不同稀疏性情况下的 MAE 值。从图 3(a)和图 3(c)中可以看出在 5 种预测算法中,算法 UMEAN 和算法 UPCC 获得较差的预测性能,而算法 IMEAN、IPCC 和 EIPCC 具有相近的预测性能。为了更具体地分析这 3 类算法,在本实验中使用图 3(b)和图 3(d)单独列出这些算法。从图 3 中可以看出,算法 EIPCC 与其他 4 类算法相比,在任意稀疏性情况下均能取得最优的 MAE 值,表明 EIPCC 具有最优的预测性能。从图 3(b)和图 3(d)中可以发现,当稀疏性由 0.1 递增到 1 的过程中,3 种算法的 MAE 值大体上呈现递减的趋势。这表明当稀疏性增加时,更多的 Web 服务将被发现为指定服务的预测提供更多的 QoS 信息,从而提高预测的性能。此外,从图 3(b)和图 3(d)中还可以发现,当训练用户数由 100 递增到 120 时,3 种算法的预测准确性获得显著提高。例如,当训练用户数为 100,稀疏性为 0.1 的 EIPCC 算法的 MAE 值为 0.03511,而当训练用户数为 120,稀疏性同样为 0.1 时 EIPCC 的 MAE 值为 0.03196。这是因为更大的训练矩阵能为预测提供更多的信息。

Web 服务的 QoS 值预测性能不仅受训练矩阵稀疏性的影响,还受到服务数的影响。表 1 表示当训练用户数为 120 稀疏性为 30% 和 60% 时不同算法失效率的 MAE 值,其中,服务数从 10 递增到 30。从表 1 中可以看出,在各种实验条件下算法 EIPCC 与其他 4 类算法相比能获得较低的 MAE 值,表明 EIPCC 能获得较优的预测性能。在表 1 中还可发现,随着服务数的增加,5 种算法的 MAE 大体上呈现递减的趋势,表明预测性能获得不断地提高。这是因为服务数的增加使得训练集的数据增加,从而为预测提供更多可用的 Web 服务 QoS 信息。此外当稀疏性由 30% 增加到 60% 时,不同服务数情况下,5 种算法的 MAE 值也同样呈现降低的趋势,表明其预测性能获得提高。该观察结果与图 3 中的结果相一致。

表 1 各种算法的 MAE 性能比较

算法	稀疏性	服务数		
		10	20	30
UMEAN 算法		0.226 69	0.221 60	0.222 77
IMEAN 算法		0.036 57	0.037 77	0.037 85
UPCC 算法	30%	0.169 52	0.149 93	0.131 98
IPCC 算法		0.038 43	0.036 02	0.033 84
EIPCC 算法		0.032 19	0.031 87	0.029 86
UMEAN 算法		0.220 19	0.225 64	0.223 13
IMEAN 算法		0.035 46	0.036 28	0.036 81
UPCC 算法	60%	0.137 60	0.116 68	0.105 31
IPCC 算法		0.036 34	0.034 15	0.032 01
EIPCC 算法		0.030 88	0.030 77	0.029 60

4 结语

随着 Web 服务应用的不断扩大,为了保证服务的正常运行在服务的选择阶段应充分考虑服务环境的动态性。本文提出了一种基于协同过滤的个性化 Web 服务选择算法,该算法

使用个性化协同过滤算法对当前候选的 Web 服务进行 QoS 值预测,从而根据其预测的 QoS 值和用户的需求为用户提供最优的候选服务。在服务的选择阶段预测候选服务的 QoS 值能有效减少 Web 服务的失效。本文提出的方法为解决动态环境下服务选择问题提供了一种新的尝试和解决方案。此外,在本文中使用的是局部最优的服务选择算法,在后续的研究中将考虑服务选择的全局最优以满足用户更多的个性化 Web 服务需求。

参考文献:

- [1] YU T, LIN K J. Service selection algorithms for Web services with end-to-end QoS constraints [C]// Proceedings of the 2004 IEEE International Conference on E-Commerce Technology. Washington, DC: IEEE Computer Society, 2004: 129 - 136.
- [2] MENASCE D A. Composing Web services: a QoS view [J]. IEEE Internet Computing, 2004, 8(6): 88 - 90.
- [3] ZHENG Z B, MA H, LYU M R, et al. WSRec: a collaborative filtering based Web service recommender system [C]// ICWS2009: Proceedings of the 7th IEEE International Conference on Web Services. Washington, DC: IEEE Computer Society, 2009: 437 - 444.
- [4] CASATI F, SHAN M C. Dynamic and adaptive composition of E-services [J]. Information Systems, 2001, 26(3): 143 - 163.
- [5] JURCA R, FALTINCS B, BINDER W. Reliable QoS monitoring based on client feedback [C]// Proceedings of the 16th International Conference on World Wide Web. Washington, DC: IEEE Computer Society, 2007: 1003 - 1012.
- [6] HUANG C L, LO C C, CHAO K M, et al. Reaching consensus: a moderated fuzzy Web services discovery method [J]. Information and Software Technology, 2006, 48(6): 410 - 423.
- [7] ARDAGNA D, PERNICI B. Global and local QoS constraints guarantee in Web service selection [C]// Proceedings of the 2005 IEEE International Conference on Web Services. Washington, DC: IEEE Computer Society, 2005: 805 - 806.
- [8] LAMPARTER S, ANKOLEKAR A, STUDER R, et al. Preference-based selection of highly configurable Web services [C]// Proceedings of the 16th International Conference on World Wide Web. Washington, DC: IEEE Computer Society, 2007: 1013 - 1022.
- [9] GAO Y, ZHANG B, NA J, et al. Optimal selection of Web services for composition based on interface-matching and weighted multistage graph [C]// Proceedings of the 6th International Conference on Parallel and Distributed Computing Applications and Technologies. Washington, DC: IEEE Computer Society, 2005: 336 - 338.
- [10] CANFORA G, di PENTA M, ESPOSITO R, et al. An approach for QoS-aware service composition based on genetic algorithms [C]// Proceedings of the 2005 Conference on Genetic and Evolutionary Computation. New York: ACM Press, 2005: 1069 - 1075.
- [11] ZENG L Z, BENATALLAH B, NGU A H, et al. QoS-aware middleware for Web services composition [J]. IEEE Transactions on Software Engineering, 2004, 30(5): 311 - 327.
- [12] BREESE J S, HECKERMAN D, KADIE C. Empirical analysis of predictive algorithms for collaborative filtering [C]// UAI-98: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers, 1998: 43 - 52.
- [13] ZHENG Z B, ZHANG Y L, LYU M R. Distributed QoS evaluation for real-world Web services [C]// Proceedings of the 8th International Conference on Web Services. Washington, DC: IEEE Computer Society, 2010: 83 - 90.