基于 Skyline 的 QoS 感知的动态服务选择

吴 健[1)] 陈 亮[1)] 邓水光[1)] 李 莹[1)] 邝 砾[2)]

[1)](浙江大学计算机学院 杭州 310027)

[2)](杭州师范大学杭州国际服务工程学院 杭州 310012)

摘 要 服务计算相关技术标准的持续完善和不断成熟推动了基于 Web 服务重用的分布式应用系统开发方式的迅速普及.而随着服务数量的爆炸性增长,网络上存在着大量功能相似、非功能特性各异的服务,如何在功能相当的服务集中选择质量较优的服务成为一个亟待解决的问题.传统的基于服务质量的服务选择方法,无论是局部最优或是全局最优策略,均面向服务库中的所有服务进行选择,选择效率受服务数量影响较大,因此不适用于基于大规模服务库的服务选择.文中引入数据库查询中的 skyline 方法,利用 skyline 中的支配关系,在选择过程中仅考虑 skyline 之上的服务,从而大大缩小了服务选择的范围,提高了服务选择的效率.同时针对动态 Web 服务环境,提出一种动态环境下的 skyline 服务维护算法,并通过一系列仿真实验证明了所提算法的高效性及良好的可扩展性.

关键词 动态服务选择;skyline;服务质量

中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2010.02136

QoS-Skyline Based Dynamic Service Selection

WU Jian[1)] CHEN Liang[1)] DENG Shui-Guang[1)] LI Ying[1)] KUANG Li[2)]

[1)](*College of Computer Science & Technology, Zhejiang University, Hangzhou* 310027)

[2)](*Hangzhou Institute of Service Engineering, Hangzhou Normal University, Hangzhou* 310012)

Abstract With the blossom of Web services, there are many function-equivalent services with different QoS (quality of service). It has become a challenge to select services with high-quality from a set of function-equivalent services. Traditional approaches to service selection, with either partial or global optimizing strategy, process selection on all candidate services. These approaches are not suitable for selection oriented to large-scale services, as the efficiency is drastically limited by the number of services. This paper introduces the skyline approaches to improve the efficiency of selection by using the dominance relationship of skyline to prune services. It also proposes a novel skyline maintain algorithm which is suitable for dynamic service environment. An extensive performance study using synthetic data is reported to verify its efficiency.

Keywords dynamic service selection; skyline; QoS

1 引 言

随着网络化应用和“软件作为服务”理念的兴起,互联网环境下软件系统的主要形态、运行方式、生产方式和使用方式正发生着巨大的变化.通过服务重用及动态聚合以构建随需应变的松耦合的分布式应用系统成为未来网络软件开发的重要趋势[1,16].然而,

收稿日期:2010-06-08;最终修改稿收到日期:2010-08-31.本课题得到国家“八六三”高技术研究发展计划项目基金(2008AA01Z141,2009AA01Z121)、国家自然科学基金(60873224,60803004)、浙江省科技项目(2008C03007,2009C31109)资助.**吴 健**,男,1975 年生,博士,副教授,研究方向为 Web 服务、网格计算、数据挖掘.E-mail: wujian2000@zju.edu.cn.**陈 亮**,男,1988 年生,博士研究生,研究方向为服务计算、数据挖掘.**邓水光**,男,1979 年生,博士,副教授,研究方向为 Web 服务、工作流、中间件.**李 莹**,男,1973 年生,博士,副教授,研究方向为软件体系结构、编译技术、中间件.**邝 砾**(通信作者),女,1982 年生,博士,讲师,研究方向为服务计算、流程管理.

随着服务数量的爆炸性增长，网络上分布着大量功能相同、非功能特性各异的服务[21]，如在 Web 服务搜索网站 Seekda 上搜索具有天气预报功能的 Web 服务就有上百项可选服务. 因此，如何在功能相当的服务集合中选择质量较优的服务成为一个亟待解决的问题. 特别是在动态的 Web 服务环境中，经常存在新服务加入、原有服务失效或现有服务质量改变等情况，这无疑都增加了基于 QoS(Quality of Service，即服务质量)的服务选择的难度. 如何在动态的环境下选择满足用户 QoS 需求的服务已经成为学术界和工业界共同关注的问题.

服务选择的需求可分为功能性和非功能性需求. 现有的工作[4,12-13,15]已针对基于功能性需求的服务选择问题进行了大量的研究并提出了相应的算法及解决方案，如整数规划、启发式算法、混合整数模型等，而在服务的功能性需求得以满足的前提下，用户对服务质量日益重视，本文也将着重研究基于 QoS 的动态服务选择.

如图 1 所示，在服务组合过程中，对于每一个子功能模块均存在一些功能相似，QoS 各异的服务可供选择. 基于 QoS 的服务选择的目标即为从每个子功能的可选服务集合中选出满足局部 QoS 要求的服务，并使得全局 QoS 也能满足用户的 QoS 要求并且尽可能地出色.

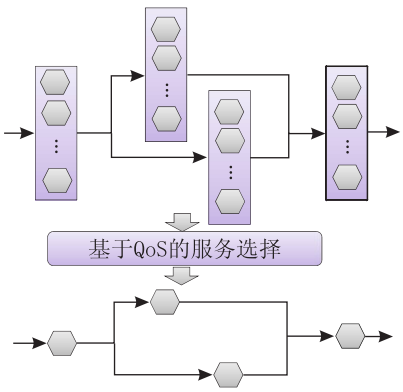

图 1　基于 QoS 的服务选择

当前基于 QoS 的服务选择可大致分为全局最优和局部最优策略两类. 全局最优策略是根据端对端的约束，提供在此约束下的最优单解，虽然在局部选择上未必总是最优，但是全局优化效果较好. 局部最优策略往往是在每个可分解的局部，对候选服务的各属性进行加权打分，选取分数最高的服务，而这些局部最优的服务的组合在全局质量上不一定最优. 这两种策略的服务选择各有优势，但它们在服务选择过程中均将服务库中的所有服务纳入选择范围，在服务库中的服务数量较多的情况下，此两种方法的服务选择效率较低.

本文引入数据库查询中 skyline 的概念，利用 skyline 中的“支配”关系，屏蔽了服务库中所有非 skyline 服务，从而大大地缩小了服务选择的范围，提高了服务选择的效率. Skyline 服务集包含所有不被其它服务所“支配”的服务，保证了服务选择结果的完备性. 另外，针对 Web 环境的动态特性，如新服务的加入、原有服务的失效及现有服务的 QoS 改变等，本文提出了一种能够有效定位发生变化的服务与 skyline 服务关系的纸带模型，并基于此模型提出一种 skyline 服务维护算法. 本文的主要贡献包括：针对传统的全局最优及局部最优策略在服务选择方面的不足，提出一种基于 skyline 的 QoS 感知的服务选择方法；进一步针对动态的 Web 服务环境，提出一种能快速定位变化服务与 skyline 服务关系的纸带模型及基于此模型的 skyline 维护算法 DSCA；最后通过基于 WS-Ben 平台的一系列实验证明所提方法的高效性.

本文第 2 节综述目前服务选择的相关工作；第 3 节介绍 skyline 的基础理论及服务的 skyline 建模；第 4 节介绍纸带模型在二维的应用及多维的扩展；在此基础上，第 5 节提出适用于动态服务环境的 skyline 维护算法 DSCA；第 6 节通过基于 WS-Ben 平台的一系列实验证明 DSCA 算法的高效性；最后是总结与展望.

2　相关工作

如何发现并选择符合应用需求的基础服务是实现服务重用并通过服务重用构建随需应变的分布式应用系统的关键. Zeng[5]等人针对基于非功能属性(QoS)的服务选择问题，基于全局最优化的策略，提出用整数规划的方法来解决服务选择问题. Ardagna[3]等人在线性规划的基础上进行了扩展，进一步优化了服务选择问题. 在文献[7-8]中也提出用 Heuristic 算法来解决服务选择问题. Li 等人提出用粗糙集的方法来解决服务选择中的匹配问题[6]，重点解决了匹配用户要求过程中的不确定性问题. 蚁群算法等启发式算法也在服务选择中有所应用[19].

然而无论是基于全局或是局部最优策略的服务选择，在选择过程中均将服务库中的所有服务纳入选择范围，而随着服务库中服务的数量增加，此种方

法会导致服务选择效率低下.为此,本文引入 skyline 的概念,利用 skyline 中的支配关系,缩小服务选择的范围.近一两年,在一些顶级会议如万维网上,也出现了少量关于 skyline 在服务领域中的应用的文章,如 Alrifai[9]等人将 skyline 应用在服务组合中,根据组合需求,推荐合适的 skyline 服务进行组合,但是该文只考虑静态情况下的 skyline 服务计算问题,且提出的算法并不适合动态服务环境下 skyline 服务的计算.本文针对动态服务环境下的 skyline 服务计算,提出了具有高效定位功能的纸带模型及动态场景下的 skyline 服务计算维护算法.

Skyline 方法最早由 Borzsonyi[18]等人首次提出,并被广泛地应用于数据库查询,多标准决策等领域.Skyline 计算算法最初只有 BNL 和 D&C 两种,随着 skyline 的应用价值被越来越多的人所接受,skyline 方法引起了更多学者的注意,Bitmap、Nearest Neighborhood、BBS 等算法被相继提出.Tan[14]等人提出用 Bitmap 的方法来实现 skyline 的计算.其主要思想是将数据映射成比特图,通过与操作来比较各点之间的支配关系.但是该方法依赖于比特之间的操作速度,而且计算整条 skyline 的代价较高.Kossmann[10]等人提出 Nearest Neighbor 算法,其思路是首先从所有数据点中找出曼哈顿距离最短的点 p,然后根据此点将数据空间划为几个区域,其中被点 p 支配的区域就不需要进行计算,将余下的区域插入 to-do 队列中,只要该队列非空,就迭代地重复上述操作.Papadias[11]等人分析了 Nearest Neighbor 算法在 I/O 和存储方面的不足并对其进行了改进,提出了目前被公认为在计算 skyline 方面最有效率的 BBS 算法.BBS 算法通过为每个项增加两次支配判断,从而降低了运算和存储的规模.

然而到目前为止,关于 skyline 在动态数据环境下的研究不多,Morse[17]等人提出一种 LookOut 算法,引入堆栈和树形结构,并通过 isSkyline 和 Mini 两个子算法来实验动态环境下的 skyline 计算,具有较高的效率.本文提出的 DSCA 算法与 LookOut 算法相比在评估变化服务对原 skyline 的影响的效率上具有较大优势,因为 DSCA 算法利用纸带模型可以有效地通过比较变化服务与 skyline 服务在纸带上的位置从而判断变化服务对 skyline 的影响,而 LookOut 方法则是通过枚举来实现.在最后的实验部分,通过一系列实验证明了 DSCA 在计算动态 skyline 的效率上优于 LookOut 算法.

3 服务的 Skyline 建模

3.1 Skyline 简介

Skyline 中提出了两个具有多维属性的数据点之间的支配关系,定义如下.

定义 1. 支配关系.对于 n 维数据点集合 $S=\{s_1,s_2,\cdots,s_m\}$,其中每个数据点可表示为$\langle q_1,q_2,\cdots,q_n\rangle$,如果 s_i 支配 s_j,记为 $s_i\prec s_j$,则对于 n 维中任何一维 s_i 的值必优于或者等于 s_j,且存在至少一维 s_i 的值优于 s_j.若每个维度属性值偏好于较小的值,则 $\forall k\in n,\ s_i.q_k\leq s_j.q_k,\ \exists a\in n,\ s_i.q_a<s_j.q_a$.

上述定义表明,对于 n 维的两组数据 s_1 和 s_2,如果数据 s_1 在任一维的值都优于或等于 s_2,且至少在某一维上优于 s_2,则称 s_1 支配 s_2.将这些数据映射到 n 维空间中,所有不被空间中其它点支配的点就构成了此数据集的 skyline.

定义 2. Skyline 集合.对于 n 维数据点集合 $S=\{s_1,s_2,\cdots,s_m\}$,所有不被其它数据点支配的点构成了 skyline,记为 s_k.$s_k=\{s_i\in S\mid \nexists s_j\in S:s_j\prec s_i\}$.

3.2 服务的 skyline 建模

在计算 skyline 服务之前,首先需要定义服务的非功能性属性并定义服务之间的支配关系.在本文中考虑的服务属性如表 1 所示,包括响应时间、可靠性、费用等有代表性的服务质量属性.现有的一些研究,如 Al-Masri[20]等人通过建立 Web 服务信息收集引擎,可以有效地获取现有 Web 服务的 13 种 QoS 属性值,包括表 1 中考虑的 6 个属性.在 QoS 属性值比较上,有些属性如响应时间和费用是数值越小越好,而可靠性等属性则是数值越高越好,为了统一数据的大小偏好关系,针对可靠性等属性进行倒数处理,使得对于本文考虑的非功能属性而言,数值均是越小越好.基于上述工作,服务被表示成由表 1 中 6 个非功能性属性值构成的六元组,即 $S=\{Q_C,Q_{RT},Q_T,Q_A,Q_S,Q_{RE}\}$.

表 1 QoS 属性

非功能属性	属性的意义
费用	Q_C 表示使用该服务需支付的费用
响应时间	Q_{RT} 表示从用户对服务发出指令到收到反馈所需的时间
网络流量	Q_T 表示服务运行时的网络流量
有效性	Q_A 表示服务有效的概率
安全性	Q_S 表示服务安全,不被攻击的概率
可靠性	Q_{RE} 表示服务在期望时间内响应用户的概率

定义 3. 服务空间.由服务构成的数据空间,每一维度代表服务的一个属性,某服务在服务空间

中的坐标由其各个属性的值决定.

本文考虑了服务的 6 个主要的非功能属性，相应地，服务空间为 6 维空间，通过将每个服务映射到服务空间，就可以通过计算该空间的 skyline 服务从而大大缩小服务选择的范围. 基于 skyline 中的支配关系，结合上文中提出的服务模型，可以得到如下的基于 QoS 的服务支配关系定义.

定义 4.　基于 QoS 的服务支配. 对于服务 s_1 和 s_2，如果 s_1 支配 s_2，即 $s_1 \prec s_2$，则对于任意一个非功能属性 i，均有 $s_1.q_i \leq s_2.q_i$，且至少存在一个非功能属性 j，使得 $s_1.q_j < s_2.q_j$.

如图 2 所示，以二维属性为例，将多个服务的响应时间和费用映射至一个二维的服务空间(在服务选择方面，响应时间和费用值均是越小越好). 服务 s_7 服务的响应时间和 s_2 相同，但是费用比 s_2 高，根据定义 1，s_7 被 s_2 支配. 同理，s_8 服务和 s_2 费用相同，但是 s_8 的响应时间比 s_2 长，因此 s_8 也被 s_2 服务所支配. 根据定义 2，所有不被其它服务支配的服务构成了 skyline，对于图 2 中的服务来说，s_1，s_2，s_3，s_4，s_5，s_6 这 6 个服务构成了 skyline 服务，而其它灰点所指示的服务都被这 6 个服务中的某个或某几个所支配. 通过对服务库中的服务进行 skyline 集合的计算，提取出相对较少的 skyline 服务，再对 skyline 集合中的服务进行服务选择，可大大提高选择效率. 同时，skyline 服务包含了服务库中所有的优质服务，而不在 skyline 服务集合中的服务在服务质量上均劣于某个或某几个 skyline 服务，因此保证了服务选择结果的完备性.

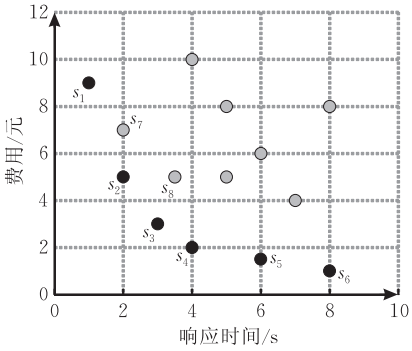

图 2　Skyline 服务示例

4　纸带模型

4.1　二维纸带模型

在服务库中服务不发生变化的情况下，即在静态环境下计算 skyline 服务较为简单，现有的很多算法如 NN 和 BBS 算法都可以很有效地获得 skyline. 然而在真实情况下，服务库中会有新的服务加入，也会有服务失效或 QoS 改变的情况发生，这些都会导致服务在服务空间中的位置发生变化，从而对 skyline 集合产生影响. 由于 QoS 的改变可以视为原服务的失效与新服务的生成两个过程的集合，因此为了阐述方便，在下文中只考虑服务新增与服务失效两种情况. 为了计算动态环境下的 skyline，系统需要了解变化的服务对 skyline 产生的影响，即定位变化服务与 skyline 的位置关系. 首先，根据服务在服务空间中与 skyline 的相对位置关系将空间中的区域划分为 3 类，以图 3 为例，为了显示方便，仍以费用和响应时间这两个属性为例进行说明.

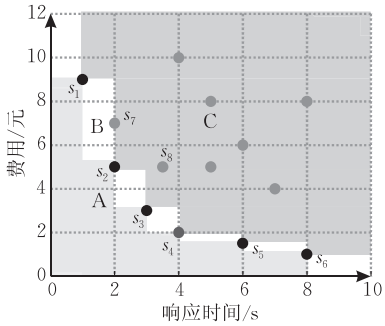

图 3　服务空间中的 3 种区域

(1) A 类区域. 若变化服务 s 落在 A 区域中，则必支配原 skyline 上的一个或多个服务，而且此种情况下的变化服务必为新增服务.

(2) B 类区域. 若变化服务 s 落在 B 区域中，则服务 s 和原 skyline 上的服务互不支配. 需要注意的是 B 区域是以 skyline 上相邻两点为左上角和右下角构成的矩形，但不包括该矩形的四条边，因为矩形的右上两条边上的服务被 skyline 服务支配，而左下两条边上的服务支配原 skyline 服务. 同时对于新增服务 s，若其响应时间或者费用值 小于 skyline 上任意服务，则其也属于 B 类区域.

(3) C 类区域. 若变化服务 s 落在 C 区域内，则此变化不对原 skyline 产生影响，因为服务 s 必被原 skyline 上的服务所支配.

动态环境下 skyline 的计算关键是判断变化的服务与原 skyline 服务的关系，如果将每次变化的服务均与 skyline 上的服务逐个进行比较来判断服务变化地 skyline 的影响，则大大地降低了此过程的效率，特别是在数据维度较大的情况下，效率呈指数级降低. 为了能够迅速地定位变化服务与原 skyline 服务的关系，本文提出了纸带模型.

纸带模型具体建立过程如下:(1)根据服务的非功能属性的数量设置相应数据的纸带,每一个属性对应一条纸带;(2)在每条纸带中逐个记录服务相应属性的值,需要注意的是,纸带中只保存 skyline 服务的值;(3)同时为纸带上的每个服务的属性值添加一个标签,标签值为服务在此纸带上的序列号.

以图 2 为例,若要针对此图建立纸带模型,则需包含 R 纸带和 C 纸带两条纸带,其中 R 纸带负责记录 skyline 上服务的响应时间,C 纸带负责记录 skyline 上服务的费用值,将 R 纸带上各点按照其相应属性值从小到大的顺序排列服务,而 C 纸带则按照其相应属性值从大到小的顺序排列服务. 同时在每条纸带上,根据此服务在纸带上的序列号添加等值的标签,即若服务 s 的响应时间在 skyline 服务中最小,则其在 R 纸带上的标签值为 1. 根据 skyline 的定义,skyline 上的服务互不支配,因此若服务 s 的响应时间最短,则其费用必为 skyline 服务中的最高,因此服务 s 在 C 纸带上的标签也为 1,同理可以发现任意 skyline 服务在 R 纸带和 C 纸带上的标签值相同.

若新增一个服务 s_t,见图 4,其中 s_t 的响应时间界与 s_2 相同,s_t 的费用于 s_4 服务的费用相同,则其纸带模型中的表示见图 5.

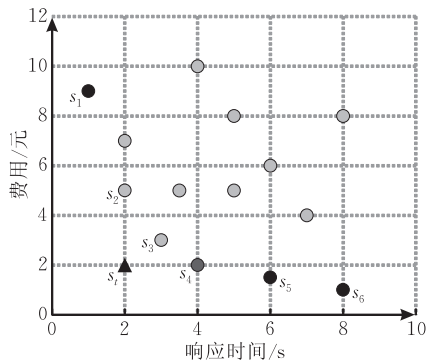

图 4 新增服务示例

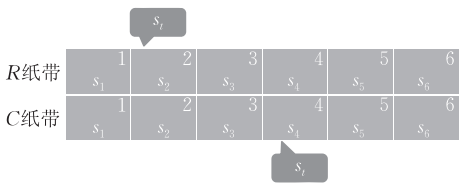

图 5 二维纸带模型

在纸带模型中,我们需要根据变化服务的属性值查找其在各纸带中的对应位置,如图 4 中,在 R 纸带中查找得到 s_t 的位置与 s_2 相同,继续在 C 纸带中查找其位置等同于 s_t.

定义 5. 将服务 s_t 在 R 轴纸带中的标签值表示为 R_t,将其在 C 轴纸带中的标签值表示为 C_t. 如果服务 s_t 在 R 纸带上位于第 k 个服务和第 $k+1$ 个服务之间,则 $R_t=k+0.5$,若服务 s_t 在 R 纸带上的位置和第 k 个服务相同,则 $R_t=k$. 若服务 s_t 在纸带上的位置在第 1 个服务之前,则 $R_t=0.5$,若在最后一个服务 s_n 之后,则 $R_t=R_n+0.5$.

根据定义 5 可知图 5 中 $R_t=2$, $C_t=4$,由此可得 $C_t>R_t$,同时观察图示发现服务 s_t 位于 A 区域. 根据数学归纳法可以得到如下定理 1.

定理 1. 若服务 s_t 处于 A 区域,则 $C_t>R_t$.

证明. 通过查看 A 区域的定义可以发现位于 A 区域中的服务能够支配至少一个原 skyline 上的服务,若服务 s_t 支配 skyline 上的某服务 s_k,则在 R、C 两维度上 s_t 的值都小于或者等于 s_k 点的值并且至少在某一维度上 s_t 点值小于 s_k 点的值. 因为在 R 纸带上各服务的数值是从小到大排列,在 C 轴纸带上是从大到小,若 s_t 点在 R 轴上小于等于 s_k 点的值,即 $R_t\leq R_k$,则在 C 轴上 s_t 点的值小于 s_k 点的值,而根据 C 纸带上的数值排列顺序,则 $C_t>C_k$,而 $C_k=R_k$,所以 $C_t>R_t$. 其它两种情况下同样得出 $C_t>R_t$. 证毕.

同样的,可以得出其它两条定理.

定理 2. 若服务 s_t 处于 B 区域,则 $C_t=R_t$.

定理 3. 若服务 s_t 处于 C 区域,则 $C_t<R_t$.

基于上述 3 条定理,区域概念被成功地迁移至纸带模型中,3 个不同的区域对应 R_t 和 C_t 的 3 种大小关系,从而可以快速地定位变化服务与原 skyline 服务的关系及其影响.

4.2 纸带模型多维扩展

多维空间的 skyline 和二维空间下的 skyline 有所不同,例如,假设三维空间的维度分别为 X、Y、Z,将三维空间下的 skyline 上的服务按照 X 值大小顺序排列,但是其相应的 Y、Z 值并不是有序序列,而是无序的,即 X、Y、Z 三者的数值大小不存在约束关系.

定义 6. $S(X_n>X_t)$ 表示 X 纸带上所处标签值大于 s_t 服务在 X 纸带上的标签值的服务的集合,同理可得 $S(X_n<X_t)$、$S(Y_n>Y_t)$、$S(Y_n<Y_t)$、$S(Z_n>Z_t)$ 和 $S(Z_n<Z_t)$.

若把 X 纸带、Y 纸带、Z 纸带上的服务按其在对应维度的数值从小到大排列,则 $S(X_n>X_t)$ 可以理解成所有 X 值比 t 服务的 X 值大的服务的集合.

定理 4. 若 $S(X_n>X_t)\cap S(Y_n>Y_t)\cap S(Z_n>Z_t)\neq\varnothing$,则变化服务 s_t 属于 A 区域.

证明. 若该条件成立,即存在一个服务 s_t,使

得 $S(X_n>X_t)\cap S(Y_n>Y_t)\cap S(Z_n>Z_t)\neq\varnothing$ 则可知在原 skyline 服务集合中存在某个服务 s_n，它在 3 个维度上的值均大于 s_t，即存在 skyline 服务被 s_t 支配. 所以 s_t 属于 A 区域.　　证毕.

定理 5.　若 $S(X_n>X_t)\cap S(Y_n>Y_t)\cap S(Z_n>Z_t)=\varnothing$，且 $S(X_n<X_t)\cap S(Y_n<Y_t)\cap S(Z_n<Z_t)=\varnothing$ 变化服务 s_t 属于 B 区域.

证明.　若该条件成立，由 $S(X_n>X_t)\cap S(Y_n>Y_t)\cap S(Z_n>Z_t)=\varnothing$ 可知服务 s_t 不支配 skyline 上的服务，由 $S(X_n<X_t)\cap S(Y_n<Y_t)\cap S(Z_n<Z_t)=\varnothing$ 可知服务 s_t 也不被 skyline 上的服务支配. 因此 s_t 属于 B 区域.　　证毕.

定理 6.　若 $S(X_n<X_t)\cap S(Y_n<Y_t)\cap S(Z_n<Z_t)\neq\varnothing$，则变化服务 s_t 属于 C 区域.

证明.　若该条件成立，则可知 skyline 服务集合中存在某服务支配 s_t. 所以服务 s_t 属于 C 区域.　　证毕.

基于上述定理 4～6，在服务拥有多维(3 维以上可继续用本方法扩展)属性的情况下，系统也可通过纸带模型迅速定位变化服务与 skyline 服务的关系，并根据第 5 节中介绍的算法基于纸带模型的定位结果针对 skyline 进行局部的调整.

5　Skyline 服务动态维护算法

动态的 Web 服务环境包括服务新增、服务失效和服务 QoS 的改变，而服务 QoS 的改变可以看作是一个原有服务的失效及一个新服务的增加. 在本章节中将给出服务新增时 skyline 的维护算法及服务失效时 skyline 的维护算法.

5.1　服务新增维护算法

以二维纸带模型为例，针对服务的两个属性：响应时间(response time)和费用(cost)，有两条纸带分别为 T_R 和 T_C，服务 s_t 在纸带上的位置分别定义为 R_t 和 C_t. 算法 1 为当有服务添加时，skyline 动态计算维护的算法.

算法 1.　DSCA_AddService.

Input: *R*-Tape, *C*-Tape, $s_t(q_r,q_c)$

Output: *R*-Tape, *C*-Tape

1. R_t←Compare q_r with values in *R*-Tape
2. C_t←Compare q_c with values in *C*-Tape
3. if $R_t<C_t$
4.　　Delete services belong to $[R_t,C_t]$
5.　　Insert $s_t(q_r,q_c)$ and Update *R*-Tape, *C*-Tape
6. end if
7. else if $R_t=C_t$
8.　　Insert $s_t(q_r,q_c)$ and Update *R*-Tape, *C*-Tape
9. end else if
10. else
11. end else

算法首先根据新增服务的响应时间和费用值在相应的纸带中的位置获取该值对应的 R_t 和 C_t，根据定义 5，如果新增服务的响应时间值处于 R 纸带上第 k 个服务和第 $k+1$ 个服务之间，则将其 R_t 赋值为 $k+0.5$(1～2 行)；如果 $R_t<C_t$，根据定理 1，可知该服务支配原 skyline 上的服务，则相应的操作为删除原 skyline 上被新增服务所支配的服务，并将新增服务加入 skyline 中，同时更新两条纸带(3～5 行)；如果 $R_t=C_t$，根据定理 2 可知，新增服务与原 skyline 上的服务互不支配，相应的操作为将新增服务插入其相应位置的 skyline 中，同时更新纸带(7～9 行)；如果 $R_t>C_t$，根据定理 3 可知，新增服务被原 skyline 上服务所支配，因此不对其作任何处理(10～11 行).

5.2　服务失效维护算法

在实现服务失效维护算法之前，首先需引入唯一支配区域的概念. 如图 6 所示，位于图中灰色矩形内的服务只被 skyline 中的 s_2 服务所支配，而不被 skyline 中的其它服务支配，我们称这个区域为服务 s_2 的唯一支配区域. 需要说明的是该矩形区域的右上两条边上的服务不属于 s_2 的唯一支配区域，因为这两条边上的服务分别被 s_1 和 s_3 支配.

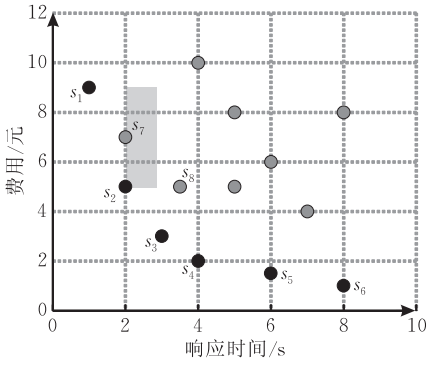

图 6　唯一支配区域示例

定义 7.　唯一支配区域. 若区域 r 为服务 s 的唯一支配区域，则区域 r 内的所有服务被且只被服务 s 所支配.

如果失效服务 s 为被支配服务，则对 skyline 无影响. 如果失效服务 s 为原 skyline 上的服务，则我们需要计算服务 s 的唯一支配区域中的 local-skyline 并将其插入原 skyline 中形成新的 skyline. 因为服务 s 唯一支配区域中的服务被且仅被服务 s 所支配，当服务 s 失效后，原 skyline 上的其它服务

无法支配这些服务，因此唯一支配区域中的 local-skyline，即为新的 skyline 的一部分，将它们添加进原 skyline 中即构成新的 skyline. 以图 6 为例，local-skyline 的计算过程如下，以 s_2 的坐标为左下角，以 s_3 的横坐标和 s_1 的纵坐标合成的坐标为右上角，形成一个矩形的唯一支配区域，可使用 NN 或 BNL 等方法计算位于此区域中的服务的 skyline 即为服务 s_2 的 local-skyline. 在多维的情况下，也可以以同样的方法计算 local-skyline，与图 6 示例的区别是在多维情况下，唯一支配区域也变成了多维空间. 如果服务的动态变化频率不高，可以将计算每个 skyline 服务的 local-skyline 作为一个预处理的过程. 算法 2 为此过程的伪代码实现.

算法 2. DSCA_DeleteService.

Input: R-Tape, C-Tape, $s_t(q_r,q_c)$

Output: R-Tape, C-Tape

1. $R_t \leftarrow$ Compare q_r with values in R-Tape
2. $C_t \leftarrow$ Compare q_c with values in C-Tape
3. if $R_t = C_t$
4. $\sigma \leftarrow$ Compute the local skyline of s_t's only dominate region
5. Delete $s_t(q_r,q_c)$ and insert σ into skyline
6. Update R-Tape, C-Tape
7. end if
8. if $R_t \neq C_t$
9. end if

服务失效维护算法首先在记录响应时间的 R 纸带和记录费用的 C 纸带中检索，通过定义 5，获取失效服务在纸带上的位置即 R_t 和 C_t（第 1 行）；如果 $R_t = C_t$，即失效服务为原 skyline 上的服务，则获取该服务唯一支配区域内的服务，并通过计算该区域中的服务获取该区域的 skyline 即 local-skyline，并将该 local-skyline 插入原 skyline 中即为新的全局 skyline（第 2～5 行）；如果 C_t 不等于 R_t，即失效服务为原来的被支配服务，则此变化对 skyline 无影响，不作处理.

6 实 验

6.1 实验设置

本实验的实验数据为基于 WS-Ben 平台设计的一系列仿真数据. WS-Ben 是一个 Web 服务生成器，可以根据用户定义 WSDL 文件来模拟服务，并可以对服务进行一些简单的操作. 其中 WSDL 文件中可以声明服务的功能、输入输出参数及其它一些非功能性的参数. 本实验通过 WS-Ben 平台模拟一系列功能相同的服务，在一定的取值范围内赋予服务随机的 QoS 值，并将这些服务做为实验数据. 由于每一个测试实验所用的数据均有所差异，因此具体实验数据的设计将在每个实验之前介绍，而不在此处细述. 实验在 Intel Core Duo E7400 2.8GHz 处理器和 2GB 内存的主机上运行，操作系统为 Windows 7. 实验分为 3 部分，首先比较本文所提的服务选择方法与线性规划和启发式算法在服务选择上的效率；然后分析 DSCA 算法在维护 skyline 服务上的效率；最后测试 DSCA 算法在服务属性维度扩展时的表现.

6.2 实验结果

实验 1. 对基于 DSCA 算法的服务选择方法与基于线性规划或启发式算法的方法在执行基于 QoS 的服务选择时的效率进行分析比较. 为了测试不同服务选择方法的效率，在实验 1 中设计了如下的场景：某组合服务由 10 个子功能组合而成，每个子功能有相同数量的具有不同 QoS 的可供替换的服务集合，通过使用基于 QoS 的服务选择方法选出合适的服务来优化组合服务的 QoS，直到效用函数最优为止. 在实验 1 中我们基于 WS-Ben 平台，对 10 个类型的服务进行了功能描述和非功能性属性声明（非功能性属性数目为 2，分别为响应时间和费用值），并为每个子功能模拟生成 500 个不同 QoS 的服务，并使用 3 种不同的基于 QoS 的服务选择方法来进行仿真实验.

实验 1 中使用的基于 QoS 的服务选择方法如下：

(1) 线性规划方法. 通过使用线性规划的方法对基于 QoS 的服务选择进行全局优化处理，选择出最优组合，即整体 QoS 效用函数值最高.

(2) 启发式算法. 以蚁群算法作为启发式算法的代表，将基于 QoS 的服务选择组合问题转化为基于图的最优路径选择问题，通过对路径上的信息素的浓度分析，找出最优路径，即选择出能满足整体 QoS 最优化的一系列服务.

(3) DSCA 方法. 选择每个子功能的服务集合中的 skyline 服务，然后再基于 skyline 服务使用线性规划的方法进行服务选择.

基于 QoS 的服务选择旨在使得整体的 QoS 能够满足用户的需求且尽可能地高. 因此在使用上述方法进行基于 QoS 的服务选择时，使用整体 QoS 效用值 U 来作为评判指标，当选出一系列子服务的组合能使得组合服务 U 值最大时，结束服务选择的

过程，并记录计算时间作为服务选择方法效率的评价指标. 效用值 U 的具体计算公式如下：

$$U_{\text{ovsi}}(k)=\frac{Q_{\text{ovmax}}(k)-Q_{\text{ovsi}}(k)}{Q_{\text{ovmax}}(k)-Q_{\text{ovmin}}(k)},$$

$$U_{\text{ovsi}}=\sum_{k=1}^{n}U_{\text{ovsi}}(k)\times w_k.$$

其中，$Q_{\text{ovmax}}(k)$ 为所有可能的组合中在 k 属性上的最大值，$Q_{\text{ovmin}}(k)$ 为所有可能的组合中在 k 属性上的最小值，$Q_{\text{ovsi}}(k)$ 为当前的选择结果构成的组合服务在 k 属性上的效用值，w_k 为 k 属性的权重，在本实验中设定每个属性权重相同，而 U_{ovsi} 即为当前选择结果的整体效用值.

如图 7 所示，横坐标为每个子功能的可选服务集合中的服务数目，纵坐标为选择出效用函数值最高的一系列子服务组合所需要的计算时间. 与利用线性规划方法或启发式算法进行基于 QoS 的服务选择相比，基于 DSCA 算法的服务选择方法在选择的效率上有较大优势. 这与前文中的理论分析一致，因为通过 DSCA 算法，系统在服务选择时仅考虑服务库中的 skyline 服务，而 skyline 服务的数量远远小于服务库中的服务数量，这使得基于 DSCA 算法的服务选择具有较高的效率.

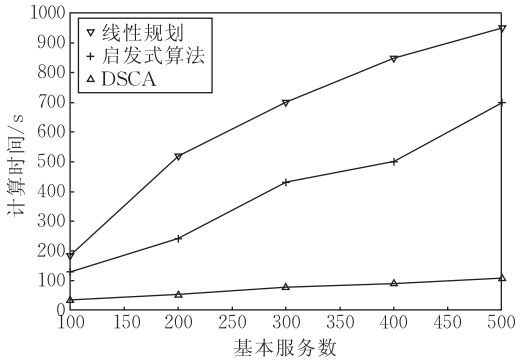

图 7　服务选择整体表现比较

实验 2.　分析 DSCA 算法在计算动态环境下的 skyline 服务的效率. 在此实验中，实验数据仍为基于 WS-Ben 模拟生成的功能相同而 QoS 不同的服务. 由于此实验旨在测试算法计算 skyline 的效率，因此我们利用 WS-Ben 声明一个服务的 WSDL 文件，并定义 2 个非功能属性为响应时间和费用，同时随机生成 10000 份含上述 2 属性值的数值对，从而来模拟 10000 个功能相同而 QoS 各异的服务. 在下述的实验中，将 DSCA 与以下算法进行对比：

(1) Repeat 算法. 在每次发生服务变化的时候，重新计算一次 skyline 服务；

(2) LookOut 算法. 由 Morse[17] 等人提出，引入堆栈和树形结构来对 skyline 的变化进行维护，对每次服务变化进行 IsSkline 和 Mini 函数的判断.

在具体的比较过程中，实验 2 考虑了如下 3 种不同的服务变化场景：① 只有新服务增加；② 只有旧服务失效；③ 服务新增和服务失效均有可能发生.

图 8 和图 9 为场景 1(即只有新服务加入)的情况下，3 种算法计算动态 skyline 服务的效率. 图 8 的实验场景为在服务基数一定的情况下，每次增加 1 个新服务，总共增加的服务数量为 10，比较 3 种不同的动态 skyline 计算方法的效率. 其中图 8 的横坐标为基数服务的数目，纵坐标为计算动态 skyline 服务所需的时间. 由图 8 可以看出，Repeat 算法效率最低，因为每次新增一个服务，该算法都要重新计算一次 skyline 服务，LookOut 算法优于 Repeat 算法，因为其算法的思想是在最初的时候计算一次 skyline 服务，在之后的服务变化过程中，根据变化的服务与 skyline 的关系进行 skyline 调整，但是其效率比 DSCA 算法低，因为在定位变化服务与 skyline 关系的过程中，LookOut 是用枚举的方法，而 DSCA 利用属性纸带，可以高效地进行关系定位，提高了整体效率.

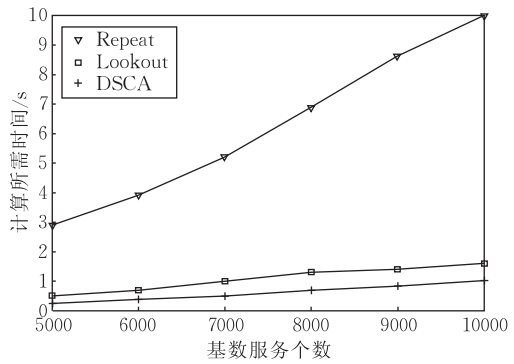

图 8　固定新增服务个数(10 个)情况下的表现

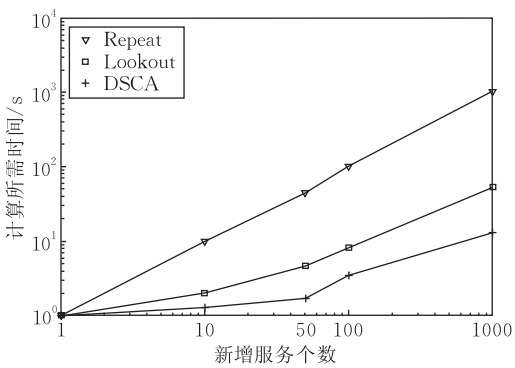

图 9　计算时间与新增服务个数的关系

图 9 的实验为在固定 10000 个基数服务的前提

下,针对有 10、50、100、1000 个新增服务的情况下,测试 3 种算法的不同表现. Repeat 算法随着新增服务数量的增加,计算时间呈线性增长趋势;LookOut 算法增长幅度较小,因为其策略是根据服务的变化进行 skyline 的局部调整;DSCA 算法的计算时间增长幅度最小,因为其策略虽与 LookOut 一样,但纸带模型的快速定位,使得 skyline 的维护成本比 LookOut 低.

图 10 和图 11 为场景 2(即只有旧服务失效)的情况下,3 种不同方法计算动态 skyline 服务的效率分析比较. 与图 8、图 9 中的曲线相似,DSCA 算法无论是在失效服务数量一定而基数服务变化的情况下,还是在基数服务数量一定而失效服务数量变化的情况下,由于其计算策略和纸带模型的迅速定位使得其计算动态 skyline 的效率都远远优于 Repeat 算法,并略优于 LookOut 算法.

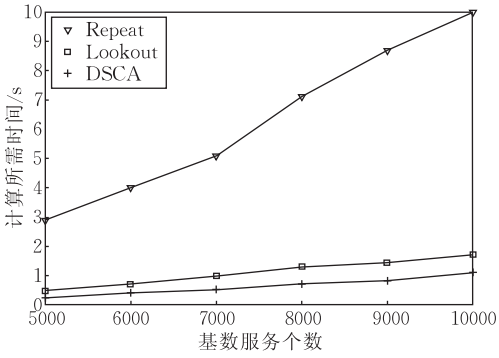

图 10 固定失效服务个数(10 个)情况下的表现

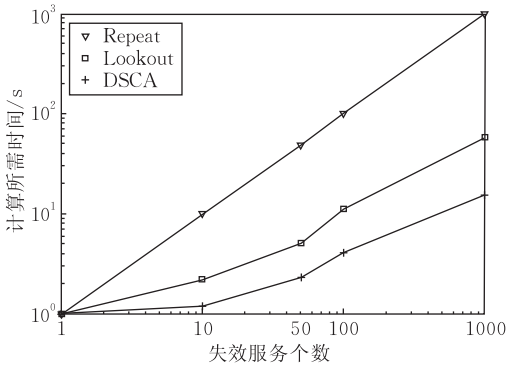

图 11 计算时间与失效服务个数的关系

图 12 和图 13 为服务新增和失效等概率发生的情况下,3 种算法计算动态 skyline 服务的效率比较. 由于可以将场景 3 下的计算视为场景 1 和场景 2 下的计算的组合,而场景 1 与场景 2 下 DSCA 算法的效率均优于 LookOut 算法和 Repeat 算法,因此在场景 3 下 DSCA 算法的效率仍然最优.

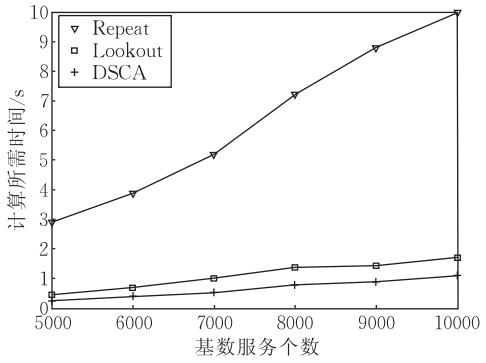

图 12 固定随机服务个数(10 个)情况下的表现

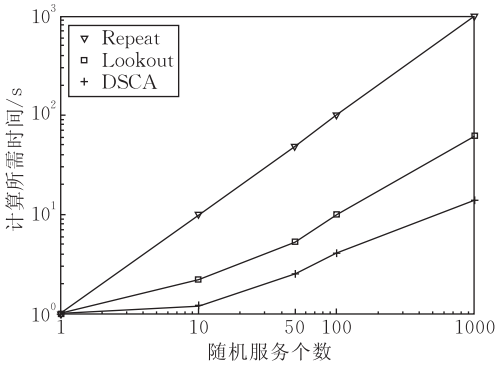

图 13 计算时间与随机服务个数的关系

实验 3. 测试服务的非功能属性维度增加时,DSCA 算法在多维可扩展性方面的表现. 在此实验中定义了 3 类服务,它们分别包含 2 个、4 个及 6 个非功能性属性,同时系统为每类服务分别随机生成 10000 份 QoS 数据,即每类服务各有 10000 个不同 QoS 的实例. 如图 14 所示,横坐标为基数服务,纵坐标为计算动态 skyline 服务所需时间. 实验场景如同实验 2 中的第 3 个场景,既有新增服务也有旧服务的失效,在本实验中,变化服务的数量固定为 10 个. 从图中可以看出,考虑的服务属性数目增加导致的计算时间增加趋势强于线性增长,但非指数级增长. 如当基数服务数量为 4000 时,当属性数目由 2 增加到 4 时,所需时间增加 150%左右,当属性数目

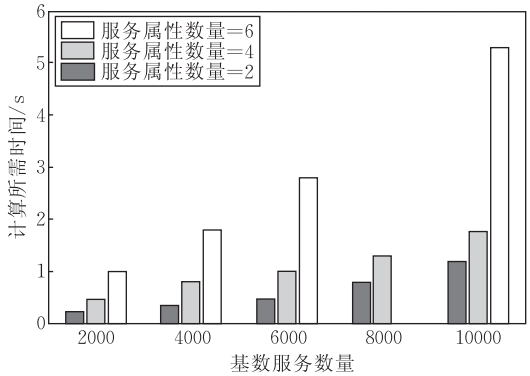

图 14 不同维度下的 DSCA 算法效率

加到 6 时，相对属性数目为 2 的情况，计算时间增加 400%左右，该增长趋势非指数级增长. 在其它不同数量的基数服务情况下，随着属性数量增加，计算所需时间增长趋势大致相同. 这是由于纸带模型在多维上依然可扩展，使得 DSCA 算法计算动态 skyline 服务的计算时间增加幅度较小，因此 DSCA 算法在多维度上具有较好的可扩展性.

7　总　结

为了能高效地选择满足用户非功能性需求的服务，本文提出使用 skyline 方法有效地对候选服务进行过滤处理. 针对现实中的动态服务环境，本文提出一个能有效地定位变化服务与原 skyline 服务关系的纸带模型，并基于此模型提出了一种动态 skyline 维护算法 DSCA. 通过一系列仿真实验，证明了 DSCA 算法在基于 QoS 的服务选择上的高效性及良好的多维可扩展性.

随着服务数量的增加，考虑的服务属性增加，skyline 上的服务数量也将相应增加. 本文的后续工作将更多地考虑如何从较多的 skyline 服务中提取出更有代表性的服务.

参　考　文　献

[1] Cardoso J, Sheth A, Miller J et al. Quality of service and semantic composition workflows. Journal of Web Semantics, 2004, 1(3): 281-308

[2] Benatallah B, Casati F, Toumani F. Representing, analyzing and managing Web service protocols. Data & Knowledge Engineering, 2006, 58(3): 327-357

[3] Ardagna D, Pernici B. Adaptive service composition in flexible processes. IEEE Transactions on Software Engineering, 2007, 33(6): 369-384

[4] Bordeaux L, Salaün G, Berardi D, Mecella M. When are two Web services compatible? Lecture Notes in Computer Science 3324. Springer, 2005: 15-28

[5] Zeng L, Benatallah B, Dumas M, Kalagnanam J, Sheng Q Z. Quality driven Web services composition//Proceedings of the 12th International Conference on World Wide Web (WWW). Budapest, Hungary, 2003: 411-421

[6] Li Maozhen, Yu Bin, Rana Omer, Wang Zidong. Grid service discovery with rough sets. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(6): 851-862

[7] Maros I. Computational Techniques of the Simplex Method. London: Springer, 2003

[8] Yu T, Zhang Y, Lin K-J. Efficient algorithms for Web services selection with end-to-end QoS constraints. ACM Transactions on the Web, 2007, 1(1): 1-26

[9] Alrifai Mohammad, Skoutas Dimitrios, Risse Thomas. Selecting skyline services for QoS-based Web service composition//Proceedings of the 12th International Conference on World Wide Web (WWW). Raleigh, USA, 2010: 11-20

[10] Kossmann D, Ramsak F, Rost S. Shooting stars in the sky: An online algorithm for skyline queries//Proceedings of the 28th International Conference on Very Large Data Bases (VLDB), Hong Kong, China, 2002: 275-286

[11] Papadias D, Tao Y, Fu G, Seeger B. An optimal and progressive algorithm for skyline queries//Proceedings of the ACM SIGMOD International Conference on Management of Data. San Diego, California, 2003: 467-478

[12] Liu F, Zhang L, Shi Y, Lin L, Shi B. Formal analysis of compatibility of Web services via CCS//Proceedings of the International Conference on Next Generation Web Services Practices. Seoul, Korea, 2005: 143-149

[13] Bohm C, Kriegel H. Determining the convex hull in large multidimensional database//Proceedings of the International Conference on Data Warehousing and Knowledge Discovery (DaWaK). Munich, Germany, 2001: 294-306

[14] Tan K-L, Eng P-K, Ooi B C. Efficient progressive skyline computation//Proceedings of the 27th International Conference on Very Large Data Bases (VLDB). San Francisco, CA, USA, 2001: 301-310

[15] Pathak J, Basu S, Honavar V. On context-specific substitutability of Web services//Proceedings of the International Conference on Web Services (ICWS). Salt Lake City, Utah, USA, 2007: 192-199

[16] Hu Jian-Qiang, Li Juan-Zi, Liao Gui-Ping. A multi-QoS based local optimal model of service selection. Chinese Journal of Computers, 2010, 33(3): 527-533(in Chinese)
(胡建强，李涓子，廖桂平. 一种基于多维服务质量的局部最优服务选择模型. 计算机学报，2010，33(3)：527-533)

[17] Morse M, Patel J M, Grosky W I. Efficient continuous skyline computation. Information Sciences, 2007, 177: 3411-3437

[18] Borzsonyi S, Kossmann D, Stocker K. The skyline operator//Proceedings of the 17th International Conference on Data Engineering (ICDE). Heidelberg, Germany, 2001: 421-430

[19] Wang Yong, Dai Gui-Ping, Hou Ya-Rong. Dynamic methods of trust-aware composite service selection. Chinese Journal of Computers, 2009, 32(8): 1669-1775(in Chinese)
(王勇，代桂平，侯亚荣. 信任感知的组合服务动态选择方法. 计算机学报，2009，32(8)：1669-1775)

[20] Al-Masri E, Mahmoud Q H. Discovering the best Web service//Proceedings of the 16th International Conference on World Wide Web (WWW). Banff, Canada, 2007: 1257-1258

[21] Alrifai Mohammad, Risse Thomas. Combining global optimization with local selection for efficient QoS-aware service composition//Proceedings of the 18th International Conference on World Wide Web (WWW). Madrid, Spain, 2009: 881-890

WU Jian, born in 1975, Ph. D., associate professor. His research interests include semantic Web, Web service, and data mining.

CHEN Liang, born in 1988, Ph. D. candidate. His main research interests include Web service, data mining and information retrieval.

DENG Shui-Guang, born in 1979, Ph. D., associate professor. His main research interests include Web service, workflow and middleware.

LI Ying, born in 1973, Ph. D., associate professor. His main research interests include software architecture, compilation technology and middleware.

KUANG Li, born in 1982, Ph. D., lecturer. Her current research interests include Web services, middleware and Pi-calculation.

Background

With the development of services, more and more services which have the same function but have different non-functional attribute values appear. Users' attention is focused on the quality of services (QoS) in the process of selection. The dynamic Web service environment, including appearance of new services, disappearance of old services and change of QoS, makes the selection more difficult. How to satisfy users' requirements for QoS in dynamic service selection becomes a hot issue.

This paper introduces the notion of "skyline", use the dominance relationship, effectively filter the dominance services, and improve the efficiency of service selection largely. In order to be suit for dynamic environment, this paper proposes a paper-tape model which could judge the relation between the changed service and skyline services quickly, and proposes an efficient skyline maintain algorithm "DSCA" based on the paper-tape model.