



运用备份服务位置和概率 QoS 模型的 Web 服务组合算法

文华

(新疆交通职业技术学院, 新疆 乌鲁木齐 831401)

摘要: 针对 workflow 服务数的增加过程中最初规划的潜在成本较高以及很多服务组合算法可靠性不足等问题, 提出了一种利用备份服务位置和概率服务质量 (QoS) 模型的服务组合算法。该算法计算服务集群的 QoS 优化选择, 为每个服务包含足够数量的备份服务, 且考虑了备份服务位置, 以便在单个故障点上进行评估。由于这些备份服务分布均匀, 防止了任务失败的发生。对于服务选择问题, 采用一种改进的多目标优化 (MOO) 算法, 利用聚类 and QoS 模型来计算可行解集合。仿真实验采用 JMETAL 3.1 框架, 评估备份服务位置的收益以及算法的可靠性。结果表明, 相比于其他 MOO 算法, 提出的算法可靠性更高, 从备份服务位置所获得的收益更高。

关键词: 工作流; 服务组合; 概率服务质量; 多目标优化; 聚类

中图分类号: TP391

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2016249

Web services composition algorithm based on the location of backup service and probabilistic QoS model

WEN Hua

Xinjiang Vocational and Technical College of Communications, Urumqi 831401, China

Abstract: In the process of increasing the number of work-flow services, as the potential cost of the initial planning is high, and the reliability of many service composition algorithms is not enough. A service composition algorithm based on the location of backup service and probabilistic quality-of-service (QoS) model was proposed. A service cluster based QoS optimization selection for each service was computed which contained a sufficient number of backup services. The location of backup service was considered to be evaluated at failure of a single point. Because the backup service was distributed evenly, the occurrence of the failure of the task was presented. For the service selection problem, an improved multiple objective optimization (MOO) algorithm was adopted to calculate the feasible solution set using clustering and QoS model. JMETAL 3.1 framework was used to assess the returns of the backup service location and the reliability of the algorithms. The results show that compared to other MOO algorithms, the proposed algorithm is more reliable and the gain from the backup service position is higher.

Key words: workflow, service composition, probabilistic quality-of-service, multiple objective optimization, clustering

1 引言

面向服务的架构在工作流中, 通过安排服务以创建新

的功能。许多情况下, 不同开发商提供的服务质量^[1] (quality-of-service, QoS) 不同, 但可能会执行相同的任务。运行时, 可以使用服务质量感知算法来优化给定的工作流

QoS。此外,用户还可以自己指定 workflow QoS 的约束^[2]。由于搜索空间呈指数增长,因此,如何解决服务组合问题是该项研究的重点。

如不进行精确接口描述^[3]或出现硬件故障^[4],服务调用可能由于各种原因而失败。当调用故障发生时,可以通过寻找一个功能类似的服务^[5]或调用重新规划^[6]来替换错误的服务。如果在响应时间上的约束非常紧急或者非常具体的服务功能崩溃,则替代服务是不可行的。随着 workflow 中服务数的不断增加,服务故障会变得更加多。大多数的服务组合算法忽视了服务故障,仅通过一个简单的加权方案来整合服务质量。

简单介绍几种服务组合和规划算法,如参考文献[6]确定了重新规划范围并扩展了该范围以有效地补偿服务失败。如果没有合适的服务或备份服务的 QoS 属性不充足,则 workflow 的执行将失败。相比于这种自组织算法,本文提出的算法预先考虑了服务失败以及每个服务的独立可靠性。

为了防止 QoS 偏差,参考文献[7]中描述的算法在执行 workflow 期间引入了重新规划。构建有效的变量取值概率模型和更新机制,提高初始种群的质量,即初始种群更加接近最优种群。通过这种方式,该算法可以防止违反某些约束。然而,执行 workflow 的可能结果对用户是隐藏的。

由于最优服务选择^[8]的问题是 NP 难问题,因此采用启发式算法^[9]计算近似最优解。为了提高服务选择的性能,对搜索空间进行修剪,将全局 QoS—约束分为每个单一服务任务的全局约束。因此,选择在全局上具有最好服务的组态。

参考文献[10]提出了使用遗传算法,该算法利用功能聚类以找到功能有效的服务组合,并研究了服务组合概念层和结构层的差距。然而,该算法忽略了服务的物理位置 and 可靠性,且没有提供故障恢复。

本文提出了一种发现备份服务的算法,以提高 workflow 的可靠性。以前的算法大多假设所有的服务都托管在孤立的服务器上,忽略了服务的物理位置,其结果是没有考虑硬件故障的影响。本文为每个服务器确定了备份服务,考虑了备份服务位置,以便在单个故障点上进行评估;当使用备份服务时,本文的概率 QoS 模型有助于预测预期结果。即使服务崩溃,也可估计可能的 QoS 范围。此外它还包不同位置级别以反映容错偏好;改进了基于多目标优化 (multiple object optimization, MOO) 服务选择的遗传算法^[10],利用聚类和 QoS 模型来计算可行解集合。

2 概率 QoS 模型的运用

本节介绍 QoS 模型,该模型用以评估每个任务的一组服务。

2.1 聚类合并与确定备份服务

图 1 为 workflow 模板和可能服务选择的实例,白色圆圈及其连接表示 workflow 模板,只定义了它们之间的任务和流。为了定义可执行的 workflow,有必要选择每个任务的服务(用深灰色圆圈显示)。

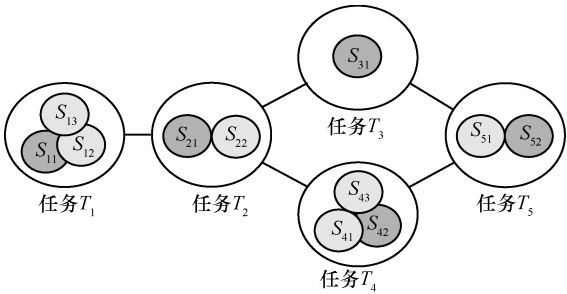


图 1 具有服务任务的工作流模板和相应的选择

功能聚类安排了以服务为引导的修复过程。然而,本文算法的聚类碎片能出现在 workflow 任务中,在图 2 的实例中添加了服务 S_{14} 、 S_{15} 、 S_{24} 和 S_{25} 。任务 T_1 的兼容服务集见表 1。任务 T_2 是来自图 1 中任务 T_1 的服务聚类。在每一个服务组合中,可用 S_{11} 和 S_{13} 替换任务 S_{12} ,因为 S_{11} 和 S_{13} 均可连接到 S_{22} 。

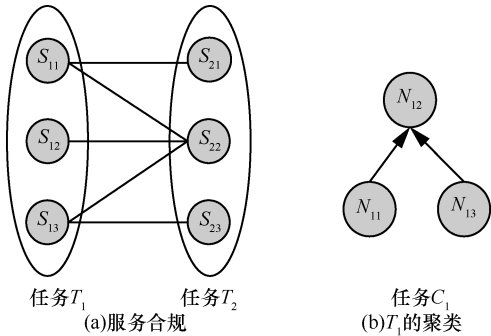


图 2 添加服务后的实例

表 1 图 1 的扩展场景,产生了分散的聚类

服务	组合部分
S_{11}	$\{S_{21}, S_{22}, S_{24}\}$
S_{12}	$\{S_{22}, S_{24}\}$
S_{13}	$\{S_{22}, S_{23}, S_{24}\}$
S_{14}	$\{S_{24}, S_{25}\}$
S_{15}	$\{S_{21}, S_{24}, S_{25}\}$



该场景中,服务 S_{14} 和 S_{15} 变成孤立聚类,尽管它们属于和 S_{11} 、 S_{12} 及 S_{13} 相同的任务。为了合并这样的聚类碎片,在存储库添加了虚拟服务,该存储库包含碎片的功能。通过这种方式,修复算子可以遍历来自相同任务的所有服务。重复这个过程直到只有一个根节点子叶,利用贪婪算法选择根节点。因此,每个虚拟服务可以保证使用更多数量的相关服务,且所有的服务在同一聚类中且可以互相连接。其中,聚类期间自动引入了虚拟服务,只是为了容易遍历连通。它们仅仅表示没有具体实现的、未被真实选择的专用接口。离线执行该聚类。

在计算完最终的聚类之后,为每个聚类和子聚类根检测一组备份服务。如果服务 S 在工作流的执行期间失效,则可采用来自 S 子聚类的任意服务替换 S 。

首先,在 QoS 前沿中安排每个聚类的服务。从第一前沿开始,为每个服务计算到聚类根部的距离。QoS 方面,通过选择到初始聚类根服务最短距离的服务确定备份服务的执行顺序。此外,选择可以最大化到根节点的物理距离服务。实例中, N_{12} 的执行顺序可能为: $S_{12} \rightarrow S_{13} \rightarrow S_{11}$ 。如果服务失败,则用下一个顺序的服务替换它。备份服务的数量取决于服务的可靠性。实验表明,3 个备份服务在大多数情况下就足够了。

2.2 扩展 QoS 模型

通过聚类所使用服务的 QoS 计算工作流属性,应用改进的 QoS 模型计算由服务聚类组成的每个节点的 QoS(即 N^{rel} 、 N^{time} 、 N^{price})。该模型有助于预测执行特定服务选择的预期结果。

对于每个节点,考虑聚类 $cluster(N)$ 中服务和备份服务的可靠性计算,有:

$$N^{rel} = 1 - \prod_{S \in cluster(N)} (1 - S^{rel}) \quad (1)$$

对于价格和响应时间,计算 3 个值:最佳值、最差值和期望值。最佳情况参考在开始时选择的第一个服务可以被成功执行的情况,即其值对应聚类根部的 QoS 值。在最坏情况下,执行顺序中的所有服务除了最后的服务

都会失败。在这种情况下,对所有服务的价格和相应时间分别求和。

期望情况考虑了可靠性,例如,通过式(2)计算聚类节点的价格:

$$E[N^{price}] = \underbrace{S_1^{price}}_{\text{实例 1 的价格}} \cdot \underbrace{S_1^{rel}}_{\text{实例 1 的概率}} + \underbrace{(S_1^{price} + S_2^{price})}_{\text{实例 2 的价格}} \cdot \underbrace{((1 - S_1^{rel}) \cdot S_2^{rel})}_{\text{实例 2 的概率}} + \dots \quad (2)$$

用同样的算法计算聚类节点相应时间的期望值。表 2 列举了 3 个实例服务的 QoS 和聚类节点产生的 QoS。

2.3 最坏情况的概率估计

由于服务事实上趋于高可靠性,至少为 90%,所以最坏场景不可能出现^[11]。想象每个任务中 3 个服务的工作流长度为 3,每个服务的可靠性为 90%。在每个任务中,当可以成功执行前两个服务事故和最后的服务时,就会出现价格和相应时间的最坏情况值。这种情况的概率为:

$$Pr[worst] = \underbrace{((1 - S_1^{rel}) \cdot (1 - S_2^{rel}))}_{\text{前 2 个服务事故}} \cdot \underbrace{S_3^{rel}}_{\text{最后服务保持}} \cdot \underbrace{S_3^{rel}}_{\text{工作流长度}} \cdot 3 \\ = (0.1 \times 0.1 \times 0.9)^3 \approx 7.3 \times 10^{-7} \quad (3)$$

一般是将最坏情况当作多数情况下从不出现的理论值。为了计算最坏情况的可行值范围,根据 Tchebysheffs(切比雪夫)不等式^[12](对于任意 $k > 0$,均有有限预期值为 μ 的变量 X 和有限非零方差 σ^2)对预期的最坏情况应用不同的措施:

$$Pr[|X - \mu| \geq \sigma k] \leq \frac{1}{k^2} \quad (4)$$

由于本文只研究上限,因此:

$$Pr[X \geq \underbrace{\sigma k + \mu}_{\xi \text{ 界限}}] = \frac{1}{k^2} \quad (5)$$

确定界限的概率值 p , 其范围通常为 95%~99.9%: $k = \sqrt{1/(1-p)}$ 。为了计算界限 ξ ,在式(5)中为 k 嵌入这个值,即有:

$$\xi = \sigma \sqrt{\frac{1}{1-p}} + \mu \quad (6)$$

表 2 图 1 场景中的实例 QoS 值

QoS					聚集群 QoS		
服务	价格/元	时间/ms	可靠性	节点	价格: ($[p_{best}, p_{worst}], E[p]$)/元	时间: ($[t_{best}, t_{worst}], E[t]$)/ms	可靠性
S_{11}	2	290	70%	N_{11}	([2,2],2)	[290,290],290)	70%
S_{12}	3	240	30%	N_{12}	([3,10],5.5)	([241,710],481)	83%
S_{13}	5	180	20%	N_{13}	([5,5],5)	([180,180],180)	20%

实验中使用 $p=90\%$, 因为它能产生平均最好的结果。

2.4 位置相关可靠性

为了确认单点故障, 根据如下 4 种冗余级别为每个使用的服务计算可靠性。

- 独立级别: 不论物理位置 (rel_{ind}) 如何, 都将服务当成是独立的。
- 主机级别: 将相同主机上的服务当成单一实体 (rel_{hos})。
- 数据中心级别: 将相同主机或数据中心上的服务当成单一的实体 (rel_{dat})。
- 区域级别: 只将不同区域中的服务当成独立的实体 (rel_{are})。

区域级别为最严格的级别, 导致了服务的高分布, 还限制了可能服务的数量。最后, 本文在服务选择阶段检索了如下对象。

- 价格: $([P_{best}, P_{worst}], E[p])$ 。
- 响应时间: $([P_{best}, P_{worst}], E[t])$ 。
- 可靠性: $(rel_{ind}, rel_{hos}, rel_{dat}, rel_{are})$ 。

3 服务选择

介绍了本文的算法, 即一种 MOO 算法, 将其作为服务选择问题的解决方案。

3.1 GA 求解最优解

遗传算法 (genetic algorithm, GA) 可以有效地计算近似帕累托最优解。在遗传算法中, 将每个候选解建模为由细胞组成的基因组。在本文问题设置中, 基因组的每个细胞编码使用工作流的每个任务分配的一组服务, 定义适应度函数以评估这些基因组。

GA 中的每次迭代都会在基因组上应用算子以获得具有较高适应度的基因组^[13]。典型的算子有变异 (部分变异)、交叉 (通过挑选现存两个的部分创建后代)。为了在保留各种可能性的同时激励当前最好基因组进化, 以概率的方式应用这些算子。

3.1.1 变异和交叉算子

变异算子使用具有一定概率的功能聚类, 为了取代 S 而挑选随机服务, 选择来自 S 聚类的服务。通过这种算法, 在执行完变异算子之后保存了到 S 的有效链路。

普通交叉算子通过从具有等概率的亲代 P' 中为每个细胞挑选亲代 P 的细胞 C_i 或细胞 C'_i 创建新的后代。在自定义算子中, 概率受每个节点中注解的影响。如果服务 S 的节点 n 具有比 S' 更多的特有服务主机, 则算子挑选概率为 75% 的 S 。通过这种方式, 后代基因组趋于具有更多数

量的服务主机, 因此, 其可靠性更高。图 3 为改进的交叉算子, 特有主机的数量与每个细胞 C_i 相连接, 且其数字显示在哪里挑选具有较大数量的细胞 (C_i 和 C'_i)。

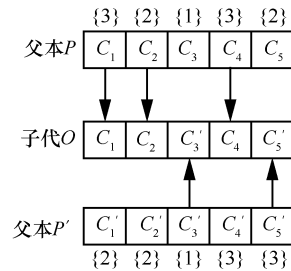


图 3 改进的交叉算子

3.1.2 修复算子

应用了变异和交叉算子之后, 在含有两个服务之间的链路的基因组上应用额外的修复算子。利用概率 P_{rep} 执行该算子, P_{rep} 为 33% 时会有一个好的平衡。为了修复有效链路而不是或多或少地利用随机算法替换服务, 该算法有选择地进行搜寻。

算法 1 修复算子

- (1) 对于每个细胞 $cell \in invalidcell(g)$ 执行
- (2) 设置 $S := c.SC$;
- (3) 对于每个服务 $S, \exists$ 链路 $l = (s, c)$ 执行
- (4) $S := S \cap S.Ol$;
- (5) 结束
- (6) 如果 $S \neq \phi$, 则
- (7) 设置 $cand :=$ 新设置;
- (8) 对于每个服务 $S' \in T'$, 其中 $\exists tl \in tlinks, tl = (T', T(c))$ 执行
- (9) 如果 $S \cap S'.Ol = \phi$, 则 $cand := cand \cup S'$;
- (10) 结束
- (11) 对于每个服务 $S'' \in c.SC$ 执行
- (12) 如果 $S \cap S''.SC = \phi$ 则 $cand := cand \cup S''$;
- (13) 结束
- (14) $g' := g.replace(c, selectsubstitute(cand))$;
- (15) 否则
- (16) 如果 $c \neq root$ 则
- (17) $g' := g.replace(c, c.getrandomparent())$;
- (18) 结束
- (19) 结束
- (20) 结束



首先,利用基因组 g 有效细胞的子聚类初始化 S 集(第2行)。如果细胞选择的服务具有到其他服务的有效链路,则细胞是有效的。接着,该服务的所有新链路计算联合输出列表。如果这个交集非空,则通过降低功能图找到的服务替换相应的服务来修复链路。通过检测是否采用可能的前服务(第8行)和来自子聚类服务计算候选 $cand$ 列表。在最后一步中,14行中从候选列表中挑选具有最多服务主机的服务,通过这种方式,基因组趋于具有更好的可靠性。如果第6行的交集是空集,则 C 的子服务及其相连接服务不可以修复无效的链路。接着,通过在第17行中选择随机父节点来上升功能图,并试图在下次迭代中修复基因组。

图4为修复算子如何遍历功能。两个引入服务 S_N 和 $S_{N'}$ 用节点 N' 和 N'' 表示,这两个服务都是无效的。为了判断是否通过降低功能修复该链路,利用子聚类的 SC 集合计算并集 $outlist(N') \cup outlist(N'') = \{S_i\}$ 。因为只有 $N_{2,SC}$ 包含服务 S_i ,所以用 S_{N_2} 替换初始服务 S_N 。

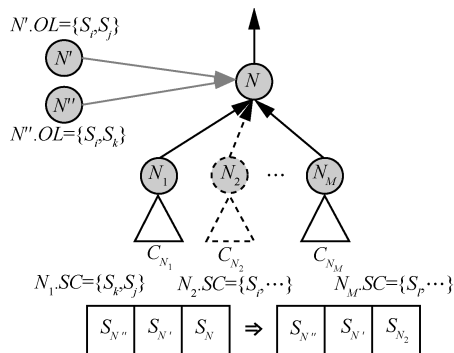


图4 选择子聚类以修复 S_N 和其两个引入的服务 $S_{N'}$ 及 $S_{N''}$ 之间链路

3.2 组合选择接口

在计算完近似帕累托最优解集之后,通过接口选择解。用户定义关于 QoS(价格、响应时间和可靠性)的偏好。此外,用户可能为最好情况和最坏情况提供风险偏好值。基于这些设置,根据式(7)计算解上的初始排序。

$$\mu_r(W) = \sum_i w_i \cdot Q_i^{\text{risk}}(W) \quad (7)$$

$$Q_i^{\text{risk}} = \frac{2}{3}(w_{\text{risk}} Q_i^{\text{best}} + (1 - w_{\text{risk}}) Q_i^{\text{worst}}) + \frac{1}{3} E[Q_i] \quad (8)$$

风险感知 QoS 值 Q_i^{risk} 集合了每个 QoS 的 3 个值,反映了用户的风险偏好。权值 w_{risk} 表示用户如何优化最好情况和最坏情况结果。这个权值可以反映用户的风险偏好,即尽管最坏情况中的质量很低,但当用户想要在最好情况

中拥有更高的质量时, w_{risk} 可以很大。此外还反映了概率的期望值,例如,当用户相信最好情况比更坏情况更可能发生, w_{risk} 可以很大。风险中性参考由考虑概率决定的值(即 $0.5 \times Pr[\text{best case}] / Pr[\text{best case}] + Pr[\text{worst case}]$)。

4 评估与分析

通过两个场景比较本文算法和相关 MOO 算法。在第一个场景 S_{loc} 中,知道服务的位置。所以评估了将位置信息融合到优化算法中的效益;在第二个场景 S_{opt} 中,所有算法采用改进的聚类和服务相关可靠性信息。该实验的目的是确定在这个问题设置中产生最佳结果的算法。

每个实验中,本文改变工作流长度和平均服务合规性。平均服务合规性表示一个服务可被连接到另一个服务的概率。具有低合规性的服务呈现出较高的功能多样性。每个算法都必须满足功能性和非功能性约束。

采用 JMETAL 3.1 框架^[14],该框架提供了 15 个多目标优化算法。利用工作流和地图中可视化的百度地图,运用 C++ 和 Qt 编程语言实现。

4.1 工作流仿真器

为了评估本文的算法,为工作流执行的仿真器提供了一个仿真的全局时间。随机生成工作流且每个任务有 50 个可选的服务。因为服务具有功能异构性,所以某些服务链路是无效的。每个服务都具有价格、响应时间和可靠性三大属性。每个服务的响应时间和可靠性均取自 QWS 数据集^[15],价格的 50% 是随机生成的,50% 与这两个值相关联。低的响应时间和高的可靠性产生高的价格,反之亦然,并应用了变化的价格和响应时间约束。

如果一个解包含一个无效的链路或违背了 QoS 约束,则该解是无效的。每个服务还与服务主机相连。根据执行时间,使用 10 个有效的独立服务提供者。为了获得服务主机工作时间的真实估计,通过在 24 h/1 000 ms 来 ping 它们,监控了 10 个真实服务,产生了 8.64×10^6 次测量。每个虚拟主机都与随机现实时间服务主机有关系。如果该主机不可用或仿真的服务出现事故,则该服务调用失败。因为服务主机可以提供来自不同任务的服务,所以可能会影响多个任务。如果来自执行顺序的所有任务均失败,则整个工作流执行失败。

为每个数据点评估了 100 个测试案例,并限制每个算法的运行时间为 5 000 ms。在仿真器运行 1 000 次中执行每个算法的解,同时记录价格和响应时间的最小值、最大

值和平均值。通过取得运行成功数与总运行数之间的比例测量平均可靠性。

4.2 提供者相关性 (场景 S_{loc})

首先考虑了服务位置的影响。图 5 给出了固定服务兼容性为 60% 时计算解的可靠性。图 5(a) 为变化 workflow 长度时的结果。比较了前 10 个解的可靠性, 随着 workflow 长度的增加, 除了本文算法, 其他算法的可靠性均降低, 本文算法的解可靠性至少为 85%。很明显其他算法的提供者相关性阻碍其备份策略。

图 5(b) 为固定 workflow 长度为 10 时, 随着服务兼容性变化的可靠性。提出的算法再次高出其他算法。如果只有 20% 的服务是兼容的, 则每个服务的计算解的可靠性会降低, 因为明显没有足够的可用备份服务。然而本文算法实现的可靠性仍然大于 80%。

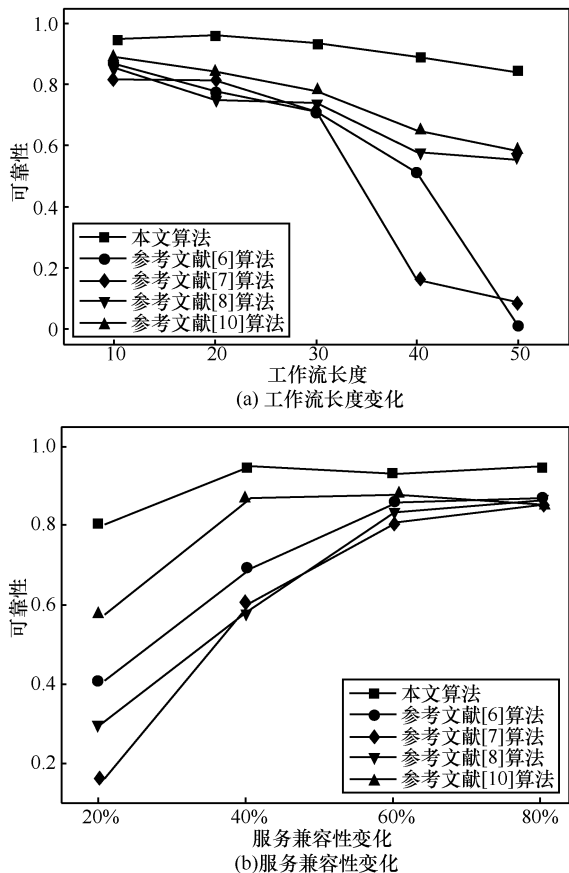


图 5 场景 S_{loc} 中算法的可靠性

下面评估了找到解的质量, 如图 6 所示。本文为每个算法计算了所有测试情况下每个算法前沿的超体积和算法合并后的前沿超体积的比率。图 6(a) 为 workflow 长度变化时的结果。很明显, 提出的算法高于其他算法。参考文献[10]超过

其他 MOO 算法, 参考文献[6-8]在没有合适的服务或备份服务时, 其 QoS 属性不充足, 工作流的执行容易失败。图 6(b) 中, 使用相同的设置, 但改变了服务兼容性。提出的算法高于其他算法, 但是当服务兼容性为 20% 时, 其 HV 比例下降。很明显, 对提出算法有用的备份服务不足。

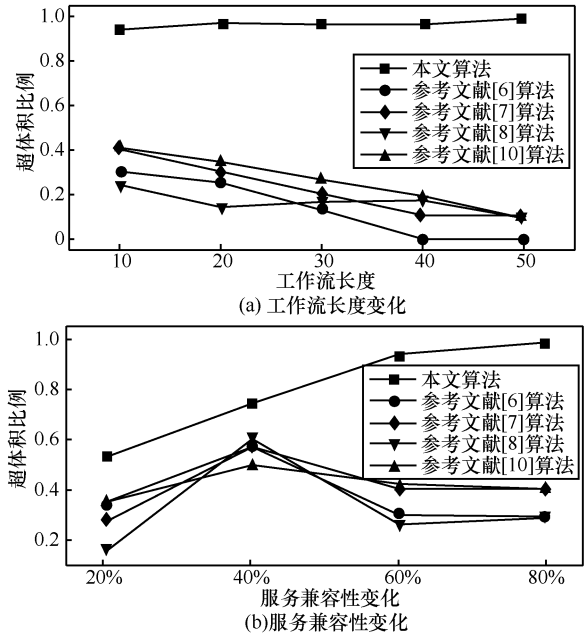


图 6 场景 S_{loc} 算法的超体积比例

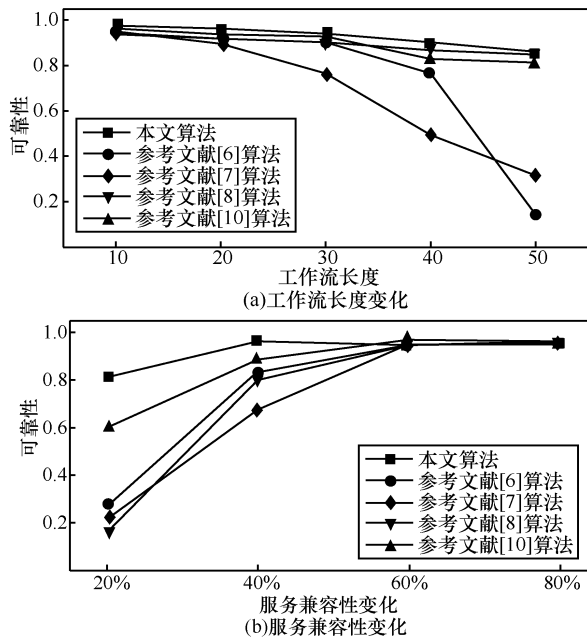
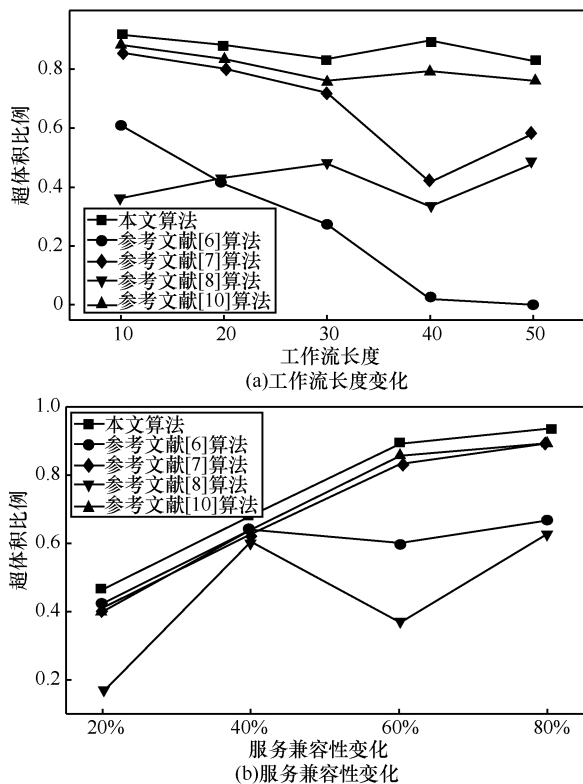
4.3 MOO 算法的性能 (S_{opt})

评估完服务位置的效益之后, 将服务位置上的信息集成到所有的 MOO 算法中。通过这种方式, 评估该模型是否改进了其他算法。

首先, 评估随着 workflow 长度变化时解的可靠性, 如图 7 所示。如图 7(a) 所示, 所有算法的结果都改进了, 然而就可靠性而言, 在所有情况中提出的算法仍然高于其他算法。同样的算法, 本文评估了服务兼容性变化时的可靠性, 如图 7(b) 所示。该结果类似于图 5(b), 表明当考虑服务位置时所有算法得以改进。

最后一个实验结果如图 8 所示, 通过比较 HV 比例评估这些算法。图 8(a) 为 workflow 长度变化时的结果, 相比于之前的场景 (如图 6(a) 所示), 所有算法获得的 HV 比例都显著增加了, 尤其是参考文献[10]和参考文献[7]中的算法。比较图 6(b) 和图 8(b) 中服务兼容性变换时的结果, 可以看出所有算法收益于位置信息。其中, 本文算法收益最高。

基于以上结果可以得出结论, 有关服务物理位置的信息对 S_{loc} 中提供足够的可靠性和 QoS 最优解至关重要, 使用功能

图7 场景中 S_{OPT} 算法的可靠性图8 场景中 S_{qq} 算法的超体积比例

聚类的缓存信息可以有效地促进选择过程。

5 结束语

提出了一种新的 QoS 模型,该模型通过考虑初始选择

阶段的服务故障预测工作流产生的 QoS。对于每个服务,事先计算了一组可能的备份服务及其相关 QoS。为了最小化硬件和网络故障的影响,考虑了物理位置。根据物理位置,导致更多数量的备份服务被分组。服务的功能聚类能使算法很容易地检测到备份服务并确定物理相关性。实验结果表明本文算法优于其他同类 MOO 算法。计划在计算工作流模板的服务计划中应用提出的算法,并考虑扩展内部服务相关的 QoS 属性,通过它们的时延可以估计服务的物理位置。

参考文献:

- [1] 曹瑞彤, 曹旭. 一种基于支持向量机的服务质量评估方案[J]. 电信科学, 2014, 30(2): 123-126.
CAO R T, CAO X. A scheme of QoS evaluation based on support vector machine [J]. Telecommunications Science, 2014, 30 (2): 123-126.
- [2] 徐战, 王劲林, 吴刚, 等. 基于路径 QoS 加权分解的工作流调度算法[J]. 系统仿真学报, 2012, 24(5):1035-1040.
XU Z, WANG J L, WU G, et al. Workflow scheduling algorithm based on weighted decomposition of paths' QoS [J]. Journal of System Simulation, 2012, 24(5): 1035-1040.
- [3] ISHIKAWA F, KATAFUCHI S, WAGNER F, et al. Bridging the gap between semantic Web services composition and common implementation architectures [C]//IEEE International Conference on Services Computing, July 4-9, 2011, Washington, DC, USA. New Jersey: IEEE Press, 2011: 152-159.
- [4] GUINARD D, TRIFA V, SPIESS P, et al. Discovery and on-demand provisioning of real-world web services [C]//IEEE International Conference on Web Services, July 6-10, 2009, Los Angeles, CA, USA. New Jersey: IEEE Press, 2009: 583-590.
- [5] 李强. 网构化软件服务的动态替换技术研究[D]. 长沙:中南大学, 2014.
LI Q. Research on dynamic replacement technology of Web based software service[D]. Changsha: Central South University, 2014.
- [6] LIN K J, ZHANG J, ZHAI Y. An efficient approach for service process reconfiguration in SOA with end-to-end QoS constraints[C]// IEEE Conference on Commerce and Enterprise Computing, July 20-23, 2009, Vienna, Austria. New Jersey: IEEE Press, 2009: 597-600.
- [7] 沈虹, 李小平. 带准备时间和截止期约束的云服务工作流调度算法[J]. 通信学报, 2015, 36(6): 183-192.
SHEN H, LI X P. Algorithm for the cloud service workflow scheduling with set up time and deadline constraints [J]. Journal on Communications, 2015, 36(6): 183-192.
- [8] 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[J]. 计算机学报, 2010, 33(3): 526-534.

- HU J Q, LI J Z, LIAO G P. A multi-Qos based local optimal model of service selection[J]. Chinese Journal of Computers, 2010, 33(3): 526-534.
- [9] KLEIN A, ISHIKAWA F, HONIDEN S. Efficient heuristic approach with improved time complexity for QoS-aware service composition [C]//IEEE International Conference on Web Services, July 4-9, 2011, Washington, DC, USA. New Jersey: IEEE Press, 2011: 436-443.
- [10] MONTIELNELSON J A. Multi-objective genetic algorithms applied to low power pressure micro-sensor design[J]. Engineering Computations, 2014, 30(30): 1128-1146.
- [11] 赵莹映, 章韵. 基于推理机制的逻辑语言在 Web 服务组合机制中的应用[J]. 电信科学, 2012, 28(12): 70-75.
- ZHAO Y Y, ZHANG Y. Application of the logic language based on reasoning mechanism in Web composite service[J]. Telecommunications Science, 2012, 28(12): 70-75.
- [12] 黄宣国. 凸函数与琴生不等式[M]. 上海: 上海教育出版社, 1991.
- HUANG X G. Convex function and the inequality of birth[M]. Shanghai: Shanghai Education Publishing House, 1991.
- [13] 周明. 遗传算法原理及应用 [M]. 北京: 国防工业出版社, 1999.
- ZHOU M. Principle and application of genetic algorithm[M]. Beijing: National Defence Industry Press, 1999.
- [14] 陆彦坤. 使用精英策略的多目标遗传算法的研究 [D]. 广州: 中山大学, 2010.
- LU Y K. Research on the multi objective genetic algorithm using the elite strategy[D]. Guangzhou: Sun Yat-sen University, 2010.
- [15] TRAN V X, TSUJI H. QoS based ranking for Web services: fuzzy approaches [C]// International Conference on Next Generation Web Services Practices, IEEE Computer Society, July 6-10, 2008, Washington DC, USA. New Jersey: IEEE Press, 2008: 77-82.

[作者简介]



文华(1980-),女,新疆交通职业技术学院讲师,主要研究方向为 Web 调度、网站安全等。