

基于信任路径搜索的分布式虚拟化环境下 可信服务组件选择方法——HA_OTPS

王慧强, 邹世辰, 林俊宇, 吕宏武, 冯光升
(哈尔滨工程大学计算机科学与技术学院, 黑龙江哈尔滨 150001)

摘 要: 可信服务组件选择是实现分布式虚拟化环境下服务可信性保障的关键. 将服务组件选择抽象为多限制优化路径问题后, 可以通过搜索信任路径来选择服务组合所需组件, 但现有的信任路径搜索算法计算复杂度过高、性能表现差, 同时在建立可信服务组件选择模型时缺少从服务组件信任关系和证据传播的角度出发, 对可信服务构建与组合过程中服务组件社会关系方面考虑. 针对以上不足, 本文提出可信质量 QoD 概念, 引入组件亲和度等属性来对分布式虚拟化环境可信服务组件选择模型加以描述与限制, 并设计一种基于 QoD 的启发式优化信任服务路径搜索算法 HA_OTPS, 选择优化信任服务路径上的服务组件来满足用户可信性需求. 实验结果表明, 与现有的信任路径搜索算法相比, HA_OTPS 算法具有运行时间较短、搜索结果效用值较高等优势.

关键词: 分布式虚拟化环境; 可信性保障; 服务组合; 信任路径

中图分类号: TP311

文献标识码: A

文章编号: 0372-2112 (2017)01-0192-08

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2017.01.026

HA_OTPS: A Dependable Service Components Selection Method in Distributed Virtualized Environment Based on Trust Path Search

WANG Hui-qiang, ZOU Shi-chen, LIN Jun-yu, LÜ Hong-wu, FENG Guang-sheng
(College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang 150001, China)

Abstract: Dependable service components selection is the key to implement dependability assurance in distributed virtualized environment. The service components selection can be modeled as Multi-Constrained Optimal Path (MCOP) problem, and service components will be selected by searching trust path. However, existing trust path search algorithms have disadvantages of high complexity and low performance. And the researches about modeling dependable service components selection lack the consideration of trust relationships and evidence spread among service components during service construction and composition. Therefore, we proposed the concept of QoD, the Quality of Dependability, introducing the attributes (e.g. component intimacy) to describe and restrict the model of service components selection in distributed virtualized environment. We also designed a heuristic optimal trust service path search algorithm HA_OTPS, and chose service components on the optimal trust service path to satisfy users' demand for service dependability. The experimental results showed that HA_OTPS has shorter execution time and higher utility of search results than the existing trust path search algorithms.

Key words: distributed virtualized environment; dependability assurance; service composition; trust path

1 引言

分布式虚拟化环境是一类以分布式虚拟化技术为核心的通用计算环境, 具有显著的开放、共享、动态、透明等特性^[1], 通过将分布式计算的资源整合能力与虚

拟化技术的跨平台能力有机结合, 对复杂多样的计算资源实现统一的虚拟化映射和管理, 极大地提高了系统资源利用率, 让“一切皆服务”的思想成为现实.

分布式虚拟化开放环境下服务的种类与数量非常庞大, 传统的个体定制式部署方式已不再适用. 将服务

收稿日期: 2015-05-11; 修回日期: 2015-09-22; 责任编辑: 马兰英

基金项目: 国家自然科学基金 (No. 61370212, No. 61402127, No. 61502118); 黑龙江省自然科学基金 (No. F2015029); 博士点基金优先发展领域项目 (No. 20122304130002); 中央高校基本科研业务费专项资金 (No. HEUCF100601)

组件根据用户需求动态组合,已成为分布式虚拟化环境下最主要的服务提供方式^[2,3]. 如何从海量服务组件中选择恰当的服务组件来构建可信软件系统是实现可信性保障所面临的核心挑战. 部分研究者尝试上下文感知方式^[4,5]来进行可信服务选择,通过对客观运行环境上下文内容的变化与历史信息进行获取分析,并据此做出相应的决策,但这种方式缺少对主观的用户可信性需求进行考虑. 而 QoS 感知方式^[6,7]通过提取服务直接体验信息,可以推荐最优的服务组合路径,但在描述和度量可信属性时存在缺陷. 对此王怀民等人^[8]将可信软件抽象为由程序和软件证据组成,海量、多样化的软件证据蕴含了服务组件间复杂社会关系,但目前缺少从服务组件关系和软件证据传播的角度出发,对服务组合交互过程中所体现的社会性的考虑.

在分布式虚拟化环境下,先由用户给出可信性指标限制,然后选择既满足功能需求、又符合可信性指标限制的服务组件建立服务组合路径已成为服务提供的主流方式^[9]. 因此分布式虚拟化环境中可信服务组件选择可以抽象为经典的多约束优化路径(Multi-Constrained Optimal Path, MCOP)问题. MCOP 问题是一个 NP-Complete 问题^[10],如果将不同属性限制转换为关于某一属性限制的函数,MCOP 问题就简化为多项式复杂度^[11],但这种方式要求问题模型中的各个限制属性有关联性. 更为通用的方式是将各限制属性联合建立一个非线性目标函数模型 $\lambda(p)$,但该模型无法在多项式时间内找到最优解,因此众多研究人员尝试使用启发式搜索、多目标遗传算法等^[9,10,12,13]来求解近似优化解,但仍面临着计算复杂度过高、性能表现差等缺点.

针对以上问题,本文提出一种分布式虚拟化下环境可信服务组件选择方法. 针对服务组件通信交互过程中所体现出的社会网络特性,提出可信质量(QoD, Quality of Dependability)概念来对可信性指标做出描述,共同建立基于 QoD 的效用函数;进而通过启发式搜索算法,在多重端到端 QoD 限制的条件下搜索信任服务路径近似优化解,为后续软件服务的组合与封装提供决策基础.

2 基于端到端 QoD 限制的服务组件选择模型

为便于对可信服务组件选择问题进行建模分析,首先需要建立分布式虚拟化环境服务组件选择模型. 由于分布式虚拟化环境通过虚拟化资源映射等方法^[14]将底层物理实现细节加以屏蔽,因此模型可以适当忽略服务组件与通信链路间的差异,更加关注服务组件间的逻辑关系,服务组件选择模型可简化为如图 1 所示.

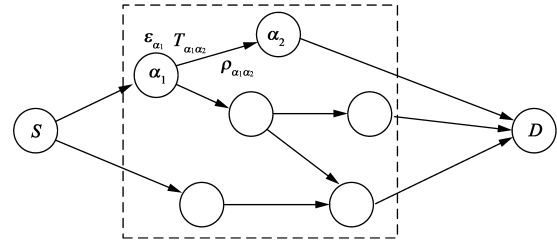


图1 分布式虚拟化环境服务组件选择模型

分布式虚拟化环境下服务组件选择可以被抽象为有向图 $G(N, E, W)$. 其中 N 为服务组件节点集,源点 S 和终点 D 表示服务选择过程的开始与结束,服务组件实体与其可信证据信息映射到 N 中除 S 与 D 以外的节点上; E 为信任边集,每一条边为虚拟链路,表示连接在同一条信任边上的两个服务组件可以通过组合来协同工作; W 为 QoD 权重,为信任服务路径搜索提供依据.

定义 1 信任服务路径是指在分布式虚拟化环境服务组件选择模型 $G(N, E, W)$ 中,从源点 S 出发,经过满足用户功能需求的服务组件节点,到达终点 D 的路径. 沿着服务组件节点 $\alpha_1, \alpha_2, \dots, \alpha_n$ 从源点 S 到终点 D 的信任服务路径记作 $p(S, \alpha_1, \alpha_2, \dots, \alpha_n, D)$.

每一条完整的信任服务路径代表一个可能满足用户需求的组合方案,信任服务路径前进方向表示基于可信证据传播的服务组件组合关系. 基于以上模型定义,提出 QoD 的概念.

定义 2 QoD (Quality of Dependability, 可信质量) 是指在服务选择过程中,信任服务路径上的服务组件可以使用户获得的预期可信性水平,包括证据影响因子、交互信任度、组件亲和度等属性.

下面来详细说明构成 QoD 的各个属性.

(1) 证据影响因子 定义在除源点和终点外的所有服务组件节点上的,在融合各种可信证据后所确定的单个服务组件自身能够提供的可信性水平保障能力值,该因子可对最终的服务组件信任服务路径的构建产生一定影响,证据影响因子记为 ε ,且 $\varepsilon \in [0, 1]$.

在 Trustie 平台可信属性模型^[15]中,软件可信证据被分为可用性、可靠性、安全性等,所有相关可信证据均可参照该模型定义映射为量化性指标. 证据影响因子在将服务组件的各种可信属性统一融合成为一个属性值 ε_{α_i} 后,沿着服务路径的方向进行聚合,方式如下:

$$\varepsilon_{p(S, \alpha_1, \alpha_2, \dots, \alpha_n, D)} = \frac{\sum_{i=1}^n \varepsilon_{\alpha_i}}{n}$$

(2) 交互信任度 定义在信任边集上的,是指在参与信任服务路径构建的服务组件间,一方根据服务组合交互历史而形成的对于另一方实现预期结果的可能性预测,交互信任度记为 T ,且 $T \in [0, 1]$.

交互信任度是一种主观判断,组件间成功组合的交互历史记录是构成组件间相互信任度的基础,服务组件只有建立良好的交互历史,才能参与可信软件的构建并获得利润^[16]. 决定交互信任度的相关信息均可以通过在服务提供、服务注册、服务消费等环节中来获取. 交互信任度在信任服务路径上聚合方式如下:

$$T_{p(S, \alpha_1, \alpha_2, \dots, \alpha_n, D)} = \prod_{(\alpha_i, \alpha_{i+1}) \in p(S, \alpha_1, \alpha_2, \dots, \alpha_n, D)} T_{\alpha_i \alpha_{i+1}}$$

(3) 组件亲和度 定义在信任边集上的,是指在参与信任服务路径构建的服务组件间,根据组件开发信息所确定的兼容程度,组件亲和度记为 ρ ,且 $\rho \in [0, 1]$.

组件亲和度是可信证据传播过程中社会属性的集中体现,服务组合兼容性较好的服务组件间可以赋予较高的组件亲和度. 这些组件关系信息可以从海量的组件证据中数据挖掘得出. 组件亲和度在信任服务路径上的聚合应当遵循非线性且随跳数增多而减弱的原则^[17],聚合方式如下:

$$\rho_{p(S, \alpha_1, \alpha_2, \dots, \alpha_n, D)} = \frac{\prod_{(\alpha_i, \alpha_{i+1}) \in p(S, \alpha_1, \alpha_2, \dots, \alpha_n, D)} \rho_{\alpha_i \alpha_{i+1}}}{n^\tau}$$

其中 $\tau \geq 1$ 为控制减弱程度的参数.

ε, T, ρ 三个属性共同构成了分布式虚拟化环境服务组件选择模型中的 QoD 权重 W . QoD 需要建立一个效用函数来衡量信任服务路径的可信水平. 设 QoD 效用记为 F , 信任服务路径 $p(S, \alpha_1, \alpha_2, \dots, \alpha_n, D)$ 上的 QoD 效用函数为:

$$F_{p(S, \alpha_1, \alpha_2, \dots, \alpha_n, D)} = \omega_\varepsilon \varepsilon_{p(S, \alpha_1, \alpha_2, \dots, \alpha_n, D)} + \omega_T T_{p(S, \alpha_1, \alpha_2, \dots, \alpha_n, D)} + \omega_\rho \rho_{p(S, \alpha_1, \alpha_2, \dots, \alpha_n, D)}$$

其中 $\omega_\varepsilon, \omega_T, \omega_\rho$ 分别为证据影响因子、交互信任度、组件亲和度的权重值, $\omega_\varepsilon, \omega_T, \omega_\rho \in (0, 1)$ 且 $\omega_\varepsilon + \omega_T + \omega_\rho = 1$.

可信服务组件优化选择的目标就是建立起一个既符合端到端 QoD 限制,又能够使 QoD 效用最大的信任服务路径,因此本文研究重点在于快速有效的建立信任服务路径,对于 QoD 属性值的获取与确定暂不做深入谈论.

3 启发式优化信任服务路径搜索算法

本文基于端到端 QoD 限制的服务组件选择模型,提出一种基于 A* 算法的启发式优化信任服务路径搜索算法——HA_OTPS 算法,首先进行反向搜索,判断从终点 D 向源点 S 来搜索是否存在满足 QoD 各个属性限制的可行解,并将搜索过程信息记录下来作为启发信息. 如果存在可行解,则根据启发信息从 S 到 D 来进行正向搜索,找到一条在满足 QoD 限制条件下使得 QoD 效用最大化的信任服务路径.

在信任服务路径搜索的过程中,如果一条路径满足多重的 QoD 约束,那么就意味着在这条路径上,每个

QoD 属性沿着信任服务路径的聚合值都应当大于用户预先给出的 QoD 约束值. 为了衡量候选信任服务路径 p 满足 QoD 限制的程度,给出如下的目标函数:

$$\Delta(p) = \max\left(\frac{C_p^\varepsilon}{\varepsilon_p}, \frac{C_p^T}{T_p}, \frac{C_p^\rho}{\rho_p}\right)$$

其中 $C_p^\varepsilon, C_p^T, C_p^\rho$ 分别代表证据影响因子、交互信任度和组件亲和度的 QoD 限制. 由目标函数定义显然可以得到引理 1.

引理 1 信任服务路径上所有 QoD 属性值均满足限制,当且仅当 $\Delta(p) \leq 1$.

为了便于算法的描述,表 1 中给出一些形式化符号定义来辅助说明算法过程:

表 1 HA_OTPS 算法形式化符号

符号	定义
$G = (N, E, W)$	参与服务选择的组件所构成的分布式虚拟化环境服务组件选择模型
$C_p^\varepsilon, C_p^T, C_p^\rho$	分别代表用户对于服务选择过程中不同 QoD 属性的限制值
$S, D \in N$	分别表示服务选择的起始和终止节点
p_m, p_l	分别表示通过反向搜索和正向搜索找到的信任服务路径
$\alpha_b^\varepsilon, \alpha_b^T, \alpha_b^\rho$	表示反向搜索过程中, D 到 α 候选路径上各个 QoD 属性的聚合值
$\alpha_b^\Delta, \alpha_b^F$	表示反向搜索过程中, D 到 α 候选路径上 QoD 限制目标函数值及效用值
S_b	表示在反向搜索过程中,已扩展节点的集合
$PREVIOUS(\alpha)$	表示与组件节点 α 相连的先头节点集合
$post(a)$	表示反向搜索过程中候选信任服务路径上 a 的后续节点
Heu_Info	从反向搜索得到的启发信息(从 D 到 G 中各点 QoD 属性值及效用值)
$\alpha_f^\varepsilon, \alpha_f^T, \alpha_f^\rho$	表示正向搜索过程中, S 到 α 候选路径上各个 QoD 属性的聚合值
α_f^F	表示正向搜索过程中, S 经由 α 到达 D 候选路径上 QoD 效用值的倒数
S_f	表示在正向搜索过程中,待扩展节点的集合
$NEXT(\alpha)$	表示与组件节点 α 相连的后续节点集合
$pre(a)$	表示正向搜索过程中候选信任服务路径上 a 的先头节点

3.1 HA_OTPS_BS 反向搜索算法

确定了搜索目标函数之后,就可以开始从 D 向 S 进行反向搜索. 对于每个除终点 D 以外的节点 α_i ,找出从 D 到 α_i 具有最小 Δ 值的路径 p_m 并且记录该条路径上

QoD 属性值与效用值(作为 A * 搜索的启发信息),直到搜索至起点 S. 关于反向搜索算法的正确性可由定理 1 得出.

定理 1 分布式虚拟化环境服务组件选择模型中如果存在能满足 QoD 限制的可行解,那么在反向搜索过程中找到的具有最小 Δ 值的路径必然是其中一个可行解.

证明 用反证法. 设 p_m 为从 D 到 S 具有最小 Δ 值的路径,设 p_n 为满足 QoD 限制的可行解. 由 $\Delta(p)$ 目标函数的定义可知, $\Delta(p_m) \leq \Delta(p_n)$. 若 p_m 不是一个可行解,则 p_m 上至少有一个 QoD 属性不满足限制要求,由引理 1 可知 $\Delta(p_m) > 1$, 此时 $\Delta(p_n) > 1$. 又因为 p_n 是可行解,同样由引理 1 可知即 $\Delta(p_n) \leq 1$, 与已知矛盾,证毕.

通过反向搜索总能找出从 D 到 S 具有最小 Δ 值的路径,如果这条路径 $\Delta > 1$, 则说明不存在可行解,即没有满足用户 QoD 限制要求的服务组合方案,HA_OTPS 算法终止;如果 $\Delta \leq 1$, 则下一步则开始执行正向搜索来搜索是否存在可以使得 QoD 效用 F 更大的优化解. HA_OTPS_BS 反向搜索算法的伪代码如算法 1 所示.

算法 1 HA_OTPS_BS 反向搜索算法

```

输入:  $G(N, E, W), C_p^e, C_p^T, C_p^p, S, D$ 
输出:  $p_m, Heu\_Info$ 
 $S_b = \emptyset$ 
PUT D INTO  $S_b$ 
FOR  $i = 1; \text{size}(N)$  DO
     $\alpha_b^A = \min_{(\alpha \in S_b)} \alpha^* \alpha_b^A$ 
    FOREACH  $\alpha' \in \text{PREVIOUS}(\alpha) \&\& \alpha' \notin S_b$  DO
         $\Delta = \max\{(C_p^e/\alpha_b'^e), (C_p^T/\alpha_b'^T), (C_p^p/\alpha_b'^p)\}$  //计算从 D 经过  $\alpha$  到  $\alpha'$  的  $\Delta$  值
        IF  $\Delta < \alpha_b^A$  THEN
            UPDATE  $\alpha_b'^e, \alpha_b'^T, \alpha_b'^p, \alpha_b'^F$  TO  $HeuInfo$  //更新 QoD 各属性值和效用值
            PUT  $\alpha'$  INTO  $S_b$ 
             $post(\alpha') = \alpha$ 
 $p_m \leftarrow post(S)$  to  $post(D)$ 
RETURN  $p_m, \Delta(p_m), Heu\_Info$ 

```

3.2 HA_OTPS_FS 正向搜索算法

如果反向搜索的结果证明确实存在符合 QoD 限制的可行信任服务路径 p_m , 那么就使用反向搜索时所记录的路径作为启发信息, 从 S 向 D 进行正向搜索来判断是否存在其他的可行解 p_l , 使得 $F(p_l) > F(p_m)$. 不妨设 α_i 为候选的中间节点, 从 S 到 α_i 的路径为 p_s , 从反向搜索记录的启发信息中找出从 D 到 α_i 的可行路径为 p_d , 则此时就确定了一条经过 α_i 的信任服务路径 $p_s + p_d$ 且 $F(p_s + p_d)$ 是目前搜索到的最大效用值. 如果路径满足 QoD 限制, 则继续迭代搜索下一跳具有最大效用值的候选中

间节点; 如果不满足限制, 则重新搜索具有最优效用值的候选路径. 关于正向搜索算法的正确性可由定理 2 得出.

定理 2 分布式虚拟化环境服务组件选择模型中如果反向搜索过程能够找到一个满足 QoD 限制的可行解 p_m , 那么正向搜索也必然能够找到一个可行解 p_l , 且 $F(p_l) \geq F(p_m)$.

证明 不妨设反向搜索找到的可行解为 p_m . 已知正向搜索过程是从 S 出发, 基于反向搜索的启发信息来迭代搜索既能满足 QoD 限制又能使效用最大化的路径. 由于在反向搜索中已经确定存在可行解, 因此在正向搜索每次迭代过程中至少能找到一个节点 α_i , 使得从 S 出发, 经过 α_i 到达 D 的路径满足 QoD 限制. 如果每次迭代只能找到一个节点, 那么经过这些节点的路径 p_l 就是反向搜索得到的路径 p_m ; 如果每次迭代能够找到多个满足 QoD 限制节点, 根据效用最大化的原则, 经过这些节点的路径 p_l 的效用值一定优于反向搜索得到的路径 p_m 的效用值. 证毕.

HA_OTPS_FS 正向搜索算法的伪代码如算法 2 所示

算法 2 HA_OTPS_FS 正向搜索算法

```

输入:  $G, C_p^e, C_p^T, C_p^p, S, D, Heu\_Info$ 
输出:  $p_l, F(p_l)$ 
 $S_f = \emptyset$ 
PUT S INTO  $S_f$ 
FOR  $i = 1; \text{size}(N)$  DO
     $\alpha_f^F = \min_{(\alpha \in S_f)} \alpha_f^* \alpha_f^F$ 
    FOREACH  $\alpha' \in \text{NEXT}(\alpha) \&\& \alpha' \notin S_b$  DO
        IF  $\alpha_f'^e \times \alpha_b'^e \geq C_p^e$  AND  $\alpha_f'^T \times \alpha_b'^T \geq C_p^T$  AND  $(\alpha_f'^p + \alpha_b'^p)/\text{hop} \geq C_p^p$  THEN
            //基于反向搜索的启发信息计算路径 QoD 属性值是否符合限制, 控制搜索范围
             $F = 1/(\omega_e \times (\alpha_f'^e \times \alpha_b'^e) + \omega_T \times (\alpha_f'^T \times \alpha_b'^T) + \omega_p \times ((\alpha_f'^p + \alpha_b'^p)/\text{hop}))$ 
            IF  $F < \alpha_f^F$  THEN
                UPDATE  $\alpha_f'^e, \alpha_f'^T, \alpha_f'^p, \alpha_f'^F$  //更新 QoD 属性值以及效用 F 值
                PUT  $\alpha'$  INTO  $S_f$ 
                 $pre(\alpha') = \alpha$ 
 $p_l \leftarrow pre(S)$  to  $pre(D)$ 
RETURN  $p_l, F(p_l)$ 

```

3.3 HA_OTPS 算法

在确定了反向搜索算法与正向搜索算法的基础上, HA_OTPS 算法可以通过先调用反向搜索算法来判断是否存在满足用户 QoD 限制的信任服务路径可行解并建立启发信息, 进而调用正向搜索算法来实现优化

信任服务路径的搜索. HA_OTPS 算法的伪代码如算法 3 所示.

算法 3 HA_OTPS 算法

输入: $G(N, E, W), C_p^e, C_p^T, C_p^p, S, D$

输出: $p_l, F(p_l)$

$p_l = \emptyset, p_m = \emptyset$

HA_OTPS_BS($G, C_p^e, C_p^T, C_p^p, S, D$)//反向搜索

IF $\Delta(p_m) > 1$ THEN

RETURN NULL//不存在满足 QoD 限制的可行解

ELSE

HA_OTPS_FS($G, C_p^e, C_p^T, C_p^p, S, D, Heu_Info$)//正向搜索

RETURN $p_l, F(p_l)$

4 实验与分析

4.1 场景设置与仿真过程

本节通过仿真来对 HA_OTPS 算法进行分析,并通过与基于 Dijkstra 的 H_MCOP 算法^[10]以及基于 A* 迭代加深的 IDA*_MCOP 算法^[13]对比来分析算法的性能表现. 实验环境为 Intel Xeon(TM) E3-1230v2 3.30GHz CPU, 16GB 内存, 64 位 Windows 7 SP1 操作系统, Matlab R2012b 软件.

本文使用 Enron 公司的 Email 数据集^[18], 从中随机选取 10 封邮件, 通过收发记录扩展成 10 个不同规模的子网. 通过 Salama 拓扑生成算法^[19]将所有子网连接起来, 并对 QoD 各属性在 $[0, 1]$ 区间内随机赋值. 随机裁剪选取两类、共八种不同场景, 以测试在不同系统规模下算法的表现情况, 如表 2 所示.

表 2 不同场景的网络拓扑概况

	场景	节点数	边数
简单场景	1	25	62
	2	37	77
	3	48	97
	4	62	157
复杂场景	5	393	6971
	6	642	17684
	7	1120	51330
	8	1695	49465

为了确保实验过程中能够搜索到信任服务路径的可行优化解, 不妨将用户对于信任服务路径的端到端 QoD 限制值分别设置为 $\varepsilon_p = 0.005$, $T_p = 0.002$, $\rho_p = 0.001$, 组建亲和度聚合时控制减弱程度的参数 $\tau = 1.5$, 信任服务路径的端到端 QoD 效用值的权重分别为 $\omega_e = 0.5$, $\omega_T = 0.3$, $\omega_p = 0.2$.

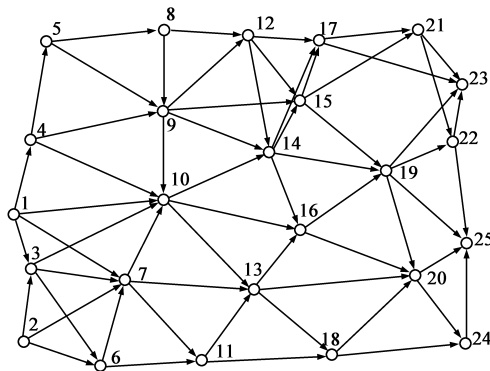


图2 场景1网络拓扑

图2 给出场景1 拓扑情况. 所有场景下的边均为有向边, 并拓扑排序^[20]对所有节点进行编号. 场景1 中源点 S 为节点1, 终点 D 为节点25, 其他场景的拓扑情况与图2 中的场景相似.

下面结合图2 来描述 HA_OTPS 算法在场景1 下的仿真过程. 首先从节点25 出发进行反向搜索, 计算到达相邻节点信任服务路径的 Δ 值, 并更新 QoD 各个属性值和路径效用值等启发信息. 选择相邻节点中具有最小 Δ 值的节点作为待扩展节点, 如果发现从节点25 经过待扩展节点到达某一其他节点 α 的 Δ 值优于目前已记录的信任服务路径的 Δ 值, 则更新到达 α 的信任服务路径及启发信息; 否则不做更新, 继续向其他节点进行扩展. 直到找出一条到节点1 信任服务路径, 如果其 Δ 值大于1, 则说明不存在符合端到端 QoD 限制的信任服务路径, 算法停止运行; 否则从节点1 开始进行正向搜索, 利用反向搜索得到的启发信息来辅助计算信任服务路径效用值 F , 直到找出一条到节点25 的既符合端到端 QoD 限制又使效用值最大化的信任服务路径. 表3 列出了在场景1 下 HA_OTPS 算法得到信任服务路径中间结果.

表3 HA_OTPS 算法在场景1 下运行的中间结果

	执行时间	路径	Δ 值	效用值
反向搜索	2.1768 ms	25→19→14→10→3→1	0.0705	0.2985
正向搜索	1.9934 ms	1→3→10→14→15→19→25	0.1041	0.3240

4.2 HA_OTPS 算法性能分析

为了有效的评估算法性能, 在每个场景下分别运行 HA_OTPS 算法5 次, 图3 至图5 详细列出了 HA_OTPS 算法搜索得到的信任服务路径效用值与平均执行时间.

在场景2 下反向搜索到的信任服务路径 Δ 值大于1, 说明场景2 中不存在满足用户 QoD 限制的可行解, 因此算法提前终止. 从图3 中可以看出, 除了场景2 外, 其它7 个场景中通过正向搜索得到的信任服务路径效用值均要优于反向搜索得到满足用户 QoD 限制信任服务路径的效用值, 这一实验结果印证了定理2.

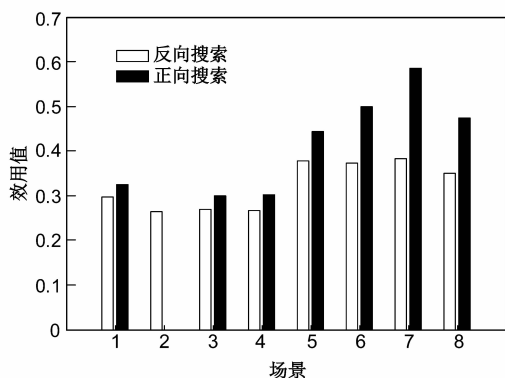


图3 HA_OTPS算法信任服务路径效用值

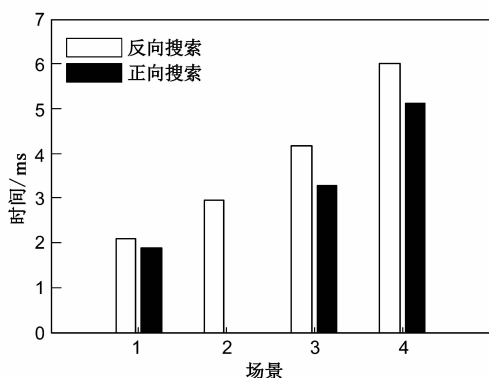


图4 HA_OTPS算法平均执行时间(简单场景)

图4展示了 HA_OTPS 算法在简单场景下平均执行时间,其中在每次执行测试时细分统计了反向搜索和正向搜索的时间.由于场景2中不存在满足用户 QoD 限制的可行解,因此算法没有执行正向搜索.而在其它简单场景下,通过引入启发信息来辅助近似优化的搜索,正向搜索时间花费要优于反向搜索.

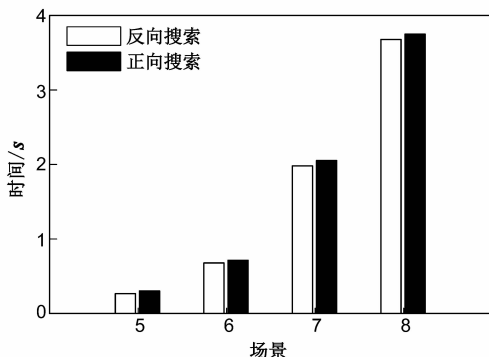


图5 HA_OTPS算法平均执行时间(复杂场景)

图5展示了 HA_OTPS 算法在复杂场景下搜索得到的信任服务路径平均执行时间.从图中可以发现在复杂场景下,正向搜索时间花费反而要大于反向搜索.通过 Matlab-Profiler 代码分析工具分析发现,由于正向搜索过程的计算量以及信息存储规模都要大于反向搜索,大量运行时间是

耗费在计算正向搜索候选信任服务路径是否满足用户 QoD 限制上,如果没有引入启发信息来限制搜索范围,那么正向搜索过程的计算时间将会继续恶化.

4.3 HA_OTPS 算法与其他算法对比

在通过对 HA_OTPS 算法搜索得到的信任服务路径效用值与平均执行时间的分析,从仿真实验的角度证明了 HA_OTPS 算法的正确性与有效性,下面通过与 H_MCOP 算法以及 IDA * _MCOP 算法进行实验对比来说明算法性能差异.

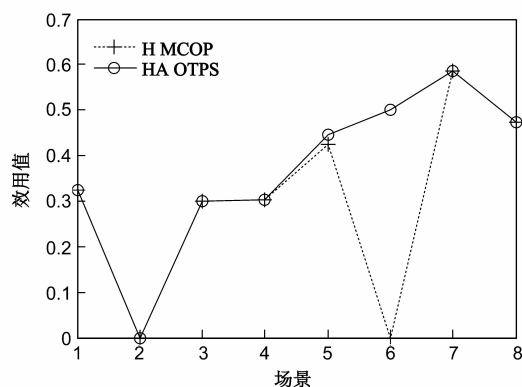


图6 信任服务路径效用值(HA_OTPS算法与H_MCOP算法)

从图6中可以看出,与 H_MCOP 算法相比,HA_OTPS 算法得到的信任服务路径的效用值都不会更差.在某些特殊场景中,H_MCOP 算法可能无法建立一个满足用户 QoD 限制的路径(场景6等),但 HA_OTPS 算法却可以得到一个可行的解.究其原因,H_MCOP 算法采取的搜索策略是在当前候选节点停止根据效用最大原则来继续扩展搜索,转而尝试搜索构成最满足用户限制路径的下一跳节点.当拥有最大效用值的信任服务路径恰好是符合用户 QoD 限制的可行解时,HA_OTPS 算法和 H_MCOP 算法都可以搜索到这个优化可行解;一旦不符合上述条件,H_MCOP 算法就难以寻找到近似优化解.而通过定理1和定理2已经证明,一旦满足用户限制的可行信任服务路径存在时,HA_OTPS 算法一定能够搜索到,并且最终搜索到的信任服务路径和效用值的结果是近似最优化解.

从图7中可以看出,通过 HA_OTPS 算法搜索到的信任服务路径的效用值的结果并不低于与 IDA * _MCOP 算法的结果,在部分大规模复杂场景中 HA_OTPS 算法得到的结果更优.究其原因,IDA * _MCOP 算法采用的是深度优先迭代加深的搜索策略,在迭代加深的过程中设定了一个搜索深度范围,当搜索节点的深度达到这个范围或者在此范围内无法搜索到终点时,就不再对该节点进行扩展并执行剪枝,以此来达到对算法时间与空间消耗的控制.但是对于大规模复杂的分布式虚拟化环境下服务组件选择来说,迭代深度

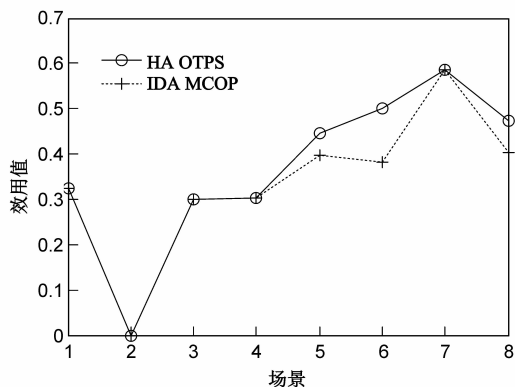


图7 信任服务路径效用值(HA_OTPS算法与IDA*_MCOP算法)

的设置导致了剪枝策略过于激进,IDA*_MCOP算法会陷入局部最优,而深度较大、路径较长的信任服务路径难以通过IDA*_MCOP算法搜索得到,而HA_OTPS算法可以得到更优的信任服务路径效用值,这也正契合用户对于分布式虚拟化环境下服务可信性保障的高需求。

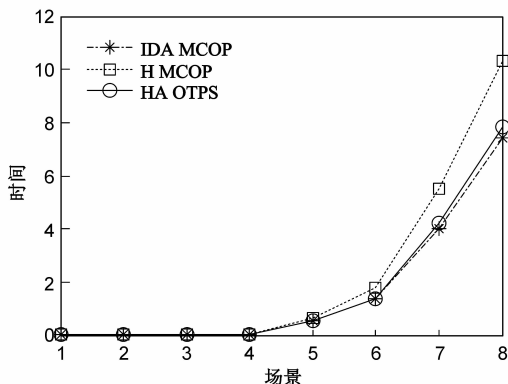


图8 算法执行平均时间对比

图8则是比较了三种算法在执行时间上的表现。可以看出,基于A*的两种算法所耗费的时间明显低于基于Dijkstra的H_MCOP算法,这是由于A*算法中引入了路径启发信息,一方面省去了路径QoD属性聚合值的重复计算,降低算法计算时间;另一方面,信任服务路径也被控制在朝向既满足用户QoD限制、又使路径效用值最大的方向进行拓展搜索,避免了Dijkstra算法的盲目扩展所导致的节点与边的遍历范围过大的问题。此外,HA_OTPS算法与IDA*_MCOP算法的执行时间上差异不大,由于IDA*_MCOP算法限制迭代深度的方法降低了算法执行时间,但是在复杂场景下路径效用值搜索结果并不令人满意;而在无近似优化的场景中(场景2),HA_OTPS算法能够更快地做出判断,这是由于IDA*_MCOP算法在搜索深度不断更新的过程中,起始节点周围的节点会被重复搜索导致的。

5 结束语

分布式虚拟化环境所具有的开放性、共享性、动态性和透明性,使得以服务组件为基本单位进行动态组合成为个性化服务发展的新趋势,其中可信服务组件选择是实现服务可信性保障的关键。本文针对现有研究尚未对服务组件关系和软件可信证据传播社会性充分考虑的不足,提出QoD可信质量的概念,并设计一种启发式优化信任服务路径搜索算法HA_OTPS,在多重端到端QoD限制的条件下搜索信任服务路径的近似优化解,最后在仿真环境下对算法进行了验证与对比分析。实验结果表明,HA_OTPS算法具有运行时间较短、搜索结果效用值较高等优势。

在下一步的研究工作中,我们还要深入研究基于QoD的软件可信证据聚合分析、服务组件伺机替换等方法与技术,结合本文提出服务组件选择方法,尝试设计并实现一套完整有效的面向软件可信性保障的分布式虚拟化环境下服务动态组合系统。

参考文献

- [1] Voith T, Oberle K, Stein M. Quality of service provisioning for distributed data center inter-connectivity enabled by network virtualization [J]. Future Generation Computer Systems, 2012, 28(3): 554-562.
- [2] 杨美清, 吕建, 梅宏. 网构软件技术体系: 一种以体系结构为中心的途径 [J]. 中国科学: 信息科学, 2008, 38(6): 818-828.
- [3] Li H, Zhao H, Cai W, et al. A modular attachment mechanism for software network evolution [J]. Physica A: Statistical Mechanics and its Applications, 2013, 392(9): 2025-2037.
- [4] Fujii K, Suda T. Semantics-based context-aware dynamic service composition [J]. ACM Transactions on Autonomous and Adaptive Systems, 2009, 4(2): article 12:1-31.
- [5] Zhou J, Gilman E, Palola J, et al. Context-aware pervasive service composition and its implementation [J]. Personal and Ubiquitous Computing, 2011, 15(3): 291-303.
- [6] Zeng L Z, Banatallah B, Ngu A H H, et al. QoS-Aware middleware for web services composition [J]. IEEE Transactions on Software Engineering, 2004, 30(5): 311-327.
- [7] Upadhyaya B, Zou Y, Ng J, et al. Towards quality of experience driven service composition [A]. 2014 IEEE World Congress on Services (SERVICES) [C]. Anchorage, USA, 2014. 18-20.
- [8] 王怀民, 尹刚, 谢冰, 等. 基于网络的可信软件大规模协同开发与演化 [J]. 中国科学: 信息科学, 2014, 44(1): 1-19.
- [9] 刘书雷, 刘云翔, 张帆, 等. 一种服务聚合中 QoS 全局最

- 优服务动态选择算法[J]. 软件学报, 2007, 18(3): 646-656.
- Liu Shulei, Liu Yunxiang, Zhang Fan, et al. A dynamic web services selection algorithm with QoS global optimal in web services composition [J]. Journal of Software, 2007, 18(3): 646-656.
- [10] Korkmaz T, Krunz M. Multi-constrained optimal path selection [A]. Proceedings of INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies [C]. Anchorage, USA: IEEE, 2001. 2: 834-843.
- [11] Liu L, Feng G. Simulated annealing based multi-constrained QoS routing in mobile ad hoc networks [J]. Wireless Personal Communications, 2007, 41(3): 393-405.
- [12] Yu T, Zhang Y, Lin K J. Efficient algorithms for Web services selection with end-to-end QoS constraints [J]. ACM Transactions on the Web (TWEB), 2007, 1(1): 6.
- [13] 马跃勇, 王海梅, 廖建军. 多约束最优路径算法比较研究 [J]. 南京理工大学学报: 自然科学版, 2012, 35(6): 749-754.
- Ma Yueyong, Wang Haimei, Liao Jianjun. Comparative study of multi-constrained optimal path algorithms [J]. Journal of Nanjing University of Science and Technology, 2012, 35(6): 749-754. (in Chinese)
- [14] 朱强, 王慧强, 吕宏武, 等. VNE-AFS: 基于人工鱼群的网络虚拟化映射算法 [J]. 通信学报, 2012 (S1): 170-177.
- Zhu Qiang, Wang Huiqiang, Lü Hongwu, et al. VNE-AFS: virtual network embedding based on artificial fish swarm [J]. Journal on Communications, 2012 (S1): 170-177. (in Chinese)
- [15] 王怀民, 刘旭东, 郎波. 软件可信分级规范 [EB/OL]. <http://www.trustie.com/clinks/trustiecriteria.jsp>, 2009.
- Wang Huaimin, Liu Xudong, Lang Bo. Software Trustworthiness Classification Specification [EB/OL]. <http://www.trustie.com/clinks/trustiecriteria.jsp>, 2009. (in Chinese)
- [16] 邹世辰, 王慧强, 冯光升, 等. 基于 Moran 过程的认知中继网络多策略信任演化模型 [J]. 小型微型计算机系统, 2014, 35(10): 2209-2214.
- Zou Shichen, Wang Huiqiang, Feng Guangsheng, et al. A multi-strategy trust evolution model for cognitive relay network based on Moran process [J]. Journal of Chinese Computer Systems, 2014, 35(10): 2209-2214. (in Chinese)
- [17] R Miller, D Perlman. Intimate Relationships [M]. McGraw Hill Higher Education, New York, USA, 2012.
- [18] Enron Email Dataset [EB/OL]. <http://www.cs.cmu.edu/~enron/>, 2015.
- [19] Salama H F. Multicast Routing for Real-Time Communication of High-Speed Networks [M]. North Carolina State University, Raleigh, USA, 1996.
- [20] Cormen T H. Introduction to Algorithms [M]. MIT press, Cambridge, USA, 2009.

作者简介



王慧强 男, 1960 年出生, 博士, 教授, 博士生导师, 研究方向为自律计算、可信计算、认知网络和未来网络。
E-mail: wanghuiqiang@hrbeu.edu.cn



邹世辰 (通信作者) 男, 1988 年出生, 博士研究生, 研究方向为可信性保障、信任管理。
E-mail: zoushichen@hrbeu.edu.cn



林俊宇 男, 1981 年出生, 博士, 助理研究员, 研究方向为 QoS、移动云计算。



吕宏武 男, 1983 年出生, 博士, 讲师, 研究方向为自律计算、博弈论、进程代数。



冯光升 男, 1980 年出生, 博士, 讲师, 研究方向为虚拟化、认知网络。