

基于服务提供者获益的服务响应选择方法

杜立新^{1,2,3}, 孟祥旭¹, 刘士军¹, 武蕾¹

(1. 山东大学计算机科学与技术学院, 250100, 济南; 2. 济南大学信息科学与工程学院, 250022, 济南;
3. 山东省网络环境智能计算技术重点实验室, 250022, 济南)

摘要: 针对当服务请求超出服务提供者能力的时候, 服务提供者如何响应才能满足自身获益最大化问题进行了研究, 指出服务提供者获益本质是其自身价值的体现, 价格收益只是一种常见的获益形式. 提出用响应有效度来表达服务响应的有效程度, 并给出了基于历史相似服务请求的有效度估算方法; 同时采用响应获益度来表达一次响应对于服务提供者获益的影响, 并详细说明了通过将获益目标分解和关系映射, 最终使用服务环境基础数据项进行表达的方法. 实验结果表明, 通过历史相似服务请求对服务响应结果进行预估并据此进行响应选择, 能够较好地提高服务响应的成功率, 降低失败率和等待超时率, 进而提高整体服务提供者获益水平. 所提方法不仅适用于服务响应问题的解决, 同时有助于服务设计者在设计时更好地将服务目标与服务环境相结合, 以提高服务请求者与提供者的协作水平.

关键词: 服务提供者获益; 服务响应; 相似请求; 有效服务

中图分类号: TP311 **文献标志码:** A **文章编号:** 0253-987X(2012)11-0112-08

A Service Response Selection Algorithm Based on Service Provider Benefits

DU Lixin^{1,2,3}, MENG Xiangxu¹, LIU Shijun¹, WU Lei¹

(1. School of Computer Science and Technology, Shandong University, Jinan 250100, China;
2. School of Information Science and Engineering, University of Jinan, Jinan 250022, China;
3. Shandong Provincial Key Laboratory of Network based Intelligent Computing, Jinan 250022, China)

Abstract: A service response selection algorithm is proposed to maximize service provider's benefits (SPB) when service requests arrived exceed provider's capabilities. The nature of SPB is the embodiment of provider's value and the income is only a common form of SPB. The response effective degree (RED) is used to express the effective levels of service response and estimated based on historical similar requests. The response benefits degree (RBD) is used for expressing the effects of one response to SPB, and is obtained by using benefit targets decomposition and relationship mapping to foundation data of service environment. The experimental results show that the proposed approach improves the success rate of service response, reduces the failure rate and waiting timeout rate, and hence improves the overall SPB. The approach can not only be used to solve the response selection problem, but also provide much help to service designers to get better combination between service targets and execution environment, and to improve the cooperation level between requester and provider.

Keywords: service provider benefit; service response; similar service request; effective service

服务计算是将服务作为软件构建基础元素的一种计算范型^[1]. 服务提供者和服务请求者是服务计算的核心研究对象, 服务提供者通过发布服务描述、提供服务相关技术和业务支持来对外提供服务, 服务请求者通过服务描述来找到适合的服务并请求使用^[2]. 提供者接收到请求者使用请求后, 按照双方约定的 QoS 提供服务. 正常情况下, 提供者和请求者可以良好协作以完成服务的请求与执行, 但是在一些应用场景中, 提供者是否响应请求、响应哪些请求将变成一个难以抉择的问题. 例如火车订票系统, 其特点是: 订票服务并发请求量大; 请求者种类众多(个人、团体、订票代理等); 对订票服务响应时间要求高(一般应在数秒内完成响应). 当订票服务接收到来自不同请求者的订票请求时, 由于票源有限, 不可能无限地响应所有的请求, 因此, 按照何种规则来进行服务响应最恰当是一个需要解决的实际问题. 再如网络购物系统, 假设某种商品热销但库存有限, 在较短时间内系统接收的订单请求远大于库存数目, 此时订单服务面临采取个体订单优先、团体订单优先或是订单总价值优先更能使订单服务提供者获益的问题. 类似问题在分布式实时控制系统^[3]中也广泛存在, 总的来看, 此类问题有几个共同特点:

- (1) 请求者和提供者处于同一个应用背景, 双方是协同关系而非买卖关系;
- (2) 请求者和提供者一般是多对一关系;
- (3) 服务请求一般带有较强的 QoS 指标, 如响应时间、截止期等;
- (4) 服务请求者瞬时请求量大, 易于超出服务提供者的服务能力;
- (5) 按照先到先服务原则响应请求, 不能很好地反映应用背景的价值取向.

此类问题对于服务提供者来讲存在着一个瞬时选择问题, 很难通过传统的服务组合、服务编排等完全避免. 本文从服务提供者的角度对问题及解决方法进行研究, 主要做了以下工作: ①探讨了服务提供者获益的概念及通过响应有效度和获益度对其进行表达; ②给出基于相似服务请求实现对服务有效度预测的方法; ③探讨通过服务响应环境参数实现响应获益度的计算; ④进行对比实验并给出结论.

1 相关工作

当前尚未见到直接与本文研究问题相同的研究成果发表, 较接近的研究主要集中在服务响应和服务选择领域. 在服务响应研究领域, 研究主要集中在

与响应时间相关的研究上. 例如: 文献[4]基于缓存机制针对服务响应时间进行了研究; 文献[5]基于队列理论给出了服务响应等待时间的计算模型; 文献[6]则通过服务预约机制实现服务实时性的保证; 文献[7]通过将请求分解执行最后合并结果的方式来提升服务响应时间. 在服务选择领域, 文献[8]从服务非功能性 QoS 角度结合服务上下文针对服务选择问题进行了研究; 文献[9]将端到端 QoS 的服务选择问题模型化为多维多选择背包问题及多约束条件下的优化路径选择问题进行研究; 文献[10]采用带有本地约束的混合整数线性规划方法对组合服务选择问题进行了研究; 文献[11]率先提出从提供者和请求者双方收益角度来考虑组合服务选择问题, 并对服务定价、罚金、收益优化等模型进行了深入研究; 文献[12-15]分别基于偏好、遗传算法、离散微粒群算法、多维服务质量局部优化等针对服务选择问题进行了深入探讨.

从已有的相关研究成果来看, 现有服务响应的研究主要集中在优化提高服务响应的机制与算法上. 在服务选择领域, 研究主要集中在组合中的服务选择问题, 即如何选取适当的候选服务, 研究者一般都隐含假设候选服务能够完成服务, 或者在降低服务 QoS 的情况下可以完成服务. 在实际 SOA 应用中, 一个服务要面向很多的请求者, 每个请求者可能位于不同应用模型中, 因此, 不同请求者之间很难完全按照理想化的组合或编排协调执行. 例如, 订票系统一般提供很多种不同的购票模型, 由于用户购票的随机性, 很容易出现对订票服务资源的竞争. 服务组合或编排本质上是将服务当作一个静态或动态资源由请求者选择或者通过服务代理进行分配, 一般都是从编排者自身利益的角度去考虑, 忽略了服务及服务提供者的主观能动性. 另一方面, 现有研究一般将请求者和提供者定义为买方和卖方, 将服务获益视为价格收益, 双方位于利益的对立面. 但是, 在很多实际服务系统应用中, 双方更多的是通过请求和响应相互协作完成服务系统赋予双方的系统责任和目标. 例如前述的订票场景, 用户可以通过 Internet、售票机、窗口等不同订票客户端服务进行订票, 对于后台的订票服务来讲, 可能在很短的时间内会有成百上千的订票请求从不同类型订票客户端到达后台订票服务. 此时, 对于后台订票服务来讲, 与客户端协作将售票业务吞吐量最大化是订票服务提供者十分重要的目标. 因此, 当服务提供者面临请求超出其服务能力的时候, 需要综合考虑服务所承担

的系统责任和服务目标进行响应.

2 问题定义与求解

2.1 问题定义

设有机器 M , 对外提供服务 s , E 为其执行队列, $|E|$ 为 M 当前并发执行 s 的数目, $\max(E)$ 为其最大服务能力; $R = \{r_1, r_2, \dots, r_n\}$ 是当前等待执行的服务请求队列, r_i 为请求; $f: R \rightarrow E$ 为响应选择调度函数, 当 $|E| < \max(E)$ 时, f 从 R 中选择适当的 r_i 调度到 E 中执行. 研究问题为: 如何设计 f 算法使得其选择能够满足服务提供者获益最大? 图 1 给出了研究问题所对应的服务请求及响应框架.

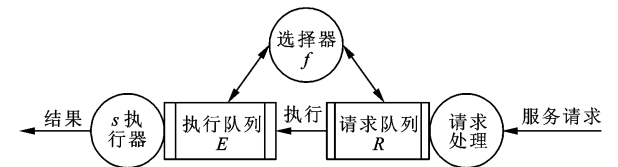


图 1 服务请求及响应交互框架

2.2 问题分析与求解

选择方法需要满足服务提供者获益最大, 首先需要明确什么是服务提供者获益. 考虑前面所述订票服务, 满足尽量多的用户同时订票、预订客户退票率最低、紧急公务订票优先等指标都是订票系统本身对于订票服务所提出的目标要求. 若订票服务能够最大化同时满足上述要求, 则订票服务提供者的自身价值得到了最大体现, 认为其获益最高.

定义 1 服务提供者获益 (Service Provider Benefits, SPB). 服务提供者获益是指服务提供者通过成功提供有效服务后所体现的自身价值. 具体来讲, 其可通过如价格收益、资源占用率、可靠性、成功率等任何能够体现服务提供者价值的获益目标来表达. 有些获益目标越大越好, 有些则越小越好. SPB 的本质就是提供者通过对服务系统赋予的多个获益目标的追求所体现的自身价值. 例如, 我们将获益目标定为价格收益最大化, 则对于服务提供者来讲其出售服务的价格越高则获益越大. 本文中获益和收益不同, 收益特指价格收益.

成功提供有效服务是指服务提供者不仅及时响应请求、执行服务而且服务结果是有效的. 有效是指提供者提供的服务对请求者来讲一定是有用的, 无用的服务结果不仅对请求者无用而且对提供者来讲是浪费资源. 例如, 请求者提出了超出提供者能力的 QoS 要求, 如果提供者不能及时发现而进行服务响

应和执行, 服务结果必然被请求者丢弃, 这种情况实际上降低了 SPB, 而且导致其他请求由于等待而延迟. 实际应用中还存在一种情况, 由于请求者自身给出的业务逻辑相关服务参数、数据等存在问题, 虽然请求者响应了请求, 服务的结果却达不到请求者预期而被放弃, 此时虽然提供者提供了良好服务, 但是结果其实是无用的. 如果提供者能够预先发现类似服务失败、结果无用等情况, 则可以拒绝请求并将资源提供给其他处于等待的请求者, 这样更有利于提供者获益目标的实现. 因此, 尽量减少无效服务是实现 SPB 最大化的必要条件, 我们通过响应有效度 (定义 2) 来表达响应对请求者的有用程度. 排除结果可能无效的请求后, 实现 SPB 最大化本质上是一个多获益目标的优化过程. 在本文的服务响应背景下, 由于服务请求、响应是不断持续进行的过程, 直接从静态全局的角度实现 SPB 的最大化十分困难, 但如果能够做到每一次响应都尽量达到 SPB 局部最高, 则全局一定趋向 SPB 最大化. 本文通过响应获益度 (定义 3) 来表达每次响应对 SPB 的影响大小.

首先考虑服务响应有效度的计算. 由于有效性是对于服务结果作用的考察, 直接通过请求对有效性做出估计比较困难. 服务请求者出于自身利益的考虑, 往往夸大其请求的重要性、时间紧迫性等 QoS 指标, 服务提供者很难辨别请求指标的真实度. 考虑到研究背景基于请求者和提供者处于具有共同目标的协同环境, 因此要求请求者在服务完成后对于服务有效度给予客观公正的评价具有可操作性, 评价可以作为请求者服务方法的一个必要环节来在服务设计时予以考虑.

定义 2 响应有效度 (Response Effective Degree, RED). 响应有效度是指提供者按照服务请求 r 给定的服务 QoS 指标完成对服务 s 的执行后, 对于请求者的有用程度, 记为 $v(r, s)$, 其取值为 $[0, 1]$, 取 0 表示服务运行失败, 取 1 表示有效度最高. 每次服务请求响应完成后, 由服务请求者根据自身工作给出响应有效度评价. 响应有效度的评价可参考如下原则.

(1) 独立无关联的请求响应评价可取基准值, 如 0.5.

(2) 若请求目标服务为当前流程中的服务, 则有: 越接近流程结束, 流程中关键服务请求响应有效度越高; 流程中非关键服务取其相邻 2 个关键服务的平均值; 流程中最后一个关键服务的有效度为 1, 其后的所有非关键服务有效度为 1; 流程初始的服

务有效度根据流程本身重要性确定。

定义 3 响应获益度 (Response Benefits Degree, RBD). 响应获益度是指服务提供者响应请求后对服务提供者获益的影响程度,具体通过对服务获益目标的影响来表达。

获益目标大部分应属业务领域,而响应行为处于服务运行的环境,只能根据服务环境信息来计算响应对于 SPB 的贡献或影响,因此必须在获益目标与服务响应环境之间建立联系才有可能在响应时进行计算。不同服务由于其承担的业务功能不同而导致其获益目标差别很大。有些获益目标易于表示,如价格,有些则过于抽象难以直接表示,如可靠性、成功率等。考虑到获益目标本质上一一定要通过服务的执行结果来体现,因此服务的每次执行一定会对其获益目标产生影响。另一方面,服务的本质是程序,服务程序内部与外部的联系一定是通过其执行环境信息进行表达的,因此可以通过制定精细的策略将获益目标与服务上下文环境(方法、参数、结果等)建立映射关系。考虑服务 s ,其可用一个四元组表示 $s = \{M_0, M_1, L, D\}$, M_0 为 s 对外服务方法集合, M_1 为 s 作为请求者使用的外部服务方法集合, L 为其内部逻辑功能集合, D 为其自身使用的数据集合。令 P_0 为 M_0 所有方法参数的集合, P_1 为 M_1 所有方法参数的集合, P_d 为 D 的数据项集合,则 $G = P_0 \cup P_1 \cup P_d$ 为服务 s 的基础数据项集合。根据定义 1 和定义 3,服务 s 的响应获益度可以用一系列的获益目标来表达。以订票服务为例,假设服务响应目标是:高级别的优质团体客户优先,其获益目标集合可为 $\{A_1\text{-高级别优先}, A_2\text{-优质客户优先}, A_3\text{-团体客户优先}\}$ 。其中, A_1 目标可以通过考察“订票用户级别”和“订票客户端类别”两个具体的指标来获得,而这两个指标都来自于集合 G 。因此,对于特定的服务 s ,只要确定其获益目标集合 V ,建立其基础数据集合 G 并明确 V 到 G 的映射关系,再通过归一化处理,即可作为服务 s 的响应获益度。

令 $V = \{v_1, v_2, \dots, v_n\}$ 为服务 s 获益目标的集合, $G_d = \{e_1, \dots, e_m\}$ 为所有获益目标考察的指标集合, $G_d \subseteq G$ 。对于服务 s 的一个请求 r ,服务提供者的响应获益度 $p(r, s)$ 可以表示为

$$p(r, s) = \sum_{i=1}^n b_i f_i(v_i) = \sum_{i=1}^n b_i f_i\left(\sum_{j=1}^m c_j g_j(e_j)\right) \quad (1)$$

式中: f_i 是目标 v_i 到 $[0, 1]$ 的映射; g_j 是指标 e_j 到

$[0, 1]$ 的映射; b_i 和 c_j 为权重参数,且 $\sum_{i=1}^n b_i = \sum_{j=1}^m c_j = 1$ 。

1. 具体 f_i 和 g_j 由目标本身的逻辑所决定。

计算响应获益度时最重要的是确定获益目标及选择适当的指标。获益目标主要由服务本身所承担的业务功能及利益期望所决定。获益指标是表达获益目标的基本单位,可以分为基础指标和分析指标。基础指标是指根据当前上下文环境可以直接得到的指标;分析指标是指无法通过上下文直接得到而必须通过对历史数据的统计分析才能获得的指标,例如退票率等。指标所对应的基础数据项需标准化处理,方法如下。

(1) 布尔类数据,其值为真、假,直接取 1 和 0。

(2) 离散数据,则将离散空间映射到 $[0, 1]$ 空间。令 D 为所有离散值的获益参数可能的取值空间集合, $\max(D)$ 为最大值, $\min(D)$ 为最小值,映射函数 $f(y) = (y - \min(D)) / (\max(D) - \min(D))$ 。

(3) 连续数据按照方法(2)进行标准化处理。

(4) 对于非数值数据,通过数据变换和数值映射方法,将数据映射到前面 3 种情况再进行处理。

对图 1 中等待请求计算出响应有效度和获益度后,就可以综合考虑二者因素完成响应选择调度。

定义 4 服务请求响应因子。服务请求响应因子用于表达服务提供者响应服务请求的重要程度,是服务提供者选择提供服务对象的依据。设 s 为目标服务,请求响应因子计算公式为

$$F(r, s) = p(r, s)k_1 + v(r, s)k_2 \quad (2)$$

式中: $k_1 + k_2 = 1$, 且 $k_1 \geq 0, k_2 \geq 0$; $p(r, s)$ 由式(1)给出;由于请求处于等待状态,因此只能根据历史请求结果对有效度 $v(r, s)$ 进行预测。考虑定义 3 中服务四元组表示,影响服务执行的主要因素包括环境因素(宿主的硬件、软件、资源等)、请求服务方法的逻辑复杂度、请求参数、处理数据复杂度及 QoS 指标等几个方面。若不考虑环境因素且以每个服务方法作为考察基本单位,如果请求参数、处理数据规模及 QoS 等差异不大,同一个服务方法的实际执行情况(执行时间、结果效果等)应不会差异较大。因此,完全可以利用已执行的类似请求来估算当前请求的服务有效度。为了精确描述两个请求的相似度,引入相似服务请求定义。

定义 5 相似服务请求。令空间 $S^* < a_1(x), a_2(x), \dots, a_n(x) >$ 为服务响应获益空间, $a_i(x)$ 为请求 x 序号为 i 的获益参数指标;给定两个服务请求 $x \in S^*, y \in S^*$,任意给定一个较小的数 $\epsilon > 0$,若对

任意 $i(1 \leq i \leq n)$, 都有 $|a_i(x) - a_i(y)| \leq e$, 则认为 x 和 y 两个请求是相似的, 记为 $x \sim y$. 特别地, 当 e 趋近于 0 时, a 和 b 趋近于相同. 相似关系是自反、对称的, 但不是传递的. 记 S^* 中 x 的所有相似请求集合为 $\hat{S}(x)$.

在 e 充分小的情况下, 找出 r 的相似请求集合 $\hat{S}(r) = \{t_1, t_2, \dots, t_k\}$, 可以用 $(\sum_{i=1}^k v(t_i, s))/k$ 作为 r 的有效度预测值. 至此, 结合定义 5 及式(1)、(2), 给出 $F(r, s)$ 的完整计算公式

$$F(r, s) = \sum_{i=1}^n b_i f_i \left(\sum_{j=1}^m c_j g_j(e_j) \right) x_1 + \left(\sum_{i=1}^k v(t_i, s) \right) / k x_2 \quad (3)$$

式中: x_1, x_2 分别为响应获益度、响应有效度的权重参数, $x_1 + x_2 = 1$ 且 $x_1 \in [0, 1], x_2 \in [0, 1]$. 考虑一个特殊的情况, 若有效度只取 1 或 0, 表示成功或失败, 计算出来的平均有效值 $v(r, s) =$ 相似请求中响应成功数/相似请求总数, 即为通常意义上的成功事件概率. 此时, 有效度均值较高意味着历史相似请求成功的概率较高.

3 服务响应选择算法

相似服务请求的查找过程本质是多维数据的检索问题, 本文采用 K 最近邻实现服务有效度估算.

3.1 基于 K 最近邻的响应有效度计算

假设 S^* 为已完成的服务请求样本空间, $r(a_1, a_2, \dots, a_n)$ 为给定请求, e 表示给定相似服务上限值; $a_i(r)$ 为请求 r 在 a_i 维的值, 算法如下.

算法 1 服务有效度预测算法.

(1) $L = S^*$; $m = 2$; // m 表示步长 e 每次缩小的度.

(2) 对 L 中每个请求 t 判断, 若 t 任意一维 a_i 的值 $a_i(t)$, 存在有 $|a_i(t) - a_i(r)| > e$, 则从 L 中排除 t . 经过上述处理, L 为 r 的相似服务集合.

(3) 若 $|L|$ 过大, 超过给定阈值 $I_{\max}$, 则 $e = e/m$, 转(2)继续执行, 直到 $|L|$ 小于 $I_{\max}$.

(4) 对 L 中每个请求 p , 计算与 r 的距离 $d(p, r)$, 并按照 $d(p, r)$ 从小到大排序.

(5) 对 L 中前 K 个 p 计算其响应有效度的和 I_{SUM} .

(6) 返回响应有效度均值为 I_{SUM}/K .

为了降低 KNN 在大样本量情况下的计算量, 第 3 步加入自动缩小步长机制. 但是, 在极端情况下 (如空间中某些点附近密度极大), 算法计算时间还

是会过长. 根据定义 5, 此时两个请求已经非常接近, 可以在不影响结果准确性的基础上采取聚类等方法对样本进行数值规约, 降低样本总量.

3.2 服务响应选择算法

根据算法 1 计算出响应有效度后, 由服务系统预先设计的规则, 可计算出响应获益度, 进而计算出服务响应因子, 并据其完成响应选择, 算法如下.

算法 2 服务响应选择算法.

(1) 若执行队列 E 满, 等待并转(1), 否则转(2).

(2) 若等待队列 R 空, 结束.

(3) 对 R 中的每个请求 r 执行: 计算 r 的响应获益度; 调用算法 1 计算 r 的历史响应有效度均值; 计算 r 的请求响应因子; 对 R 中的请求按照响应因子大小排序.

(4) 调度响应因子最高的请求 r 执行, 转(1).

算法 2 并未区分等待请求数目是否超过空闲执行队列数目, 即请求少于服务能力时也执行有效度预测. 当请求没有超过服务提供者能力时, 显然所有的请求都会被调度执行, 与不进行预测直接调度相比结果并无不同, 只是增加了预测时间, 但如果服务执行失败或有效度较低, 显然会对服务提供者获益水平有负面影响. 在服务系统构建初期, 这些失败或低有效度的响应结果对于历史数据的积累非常重要, 但随着积累数据量增大, 其必要性不断降低. 因此, 在服务能力空闲情况下是否响应失败或低有效度的请求可由系统响应策略加以约束. 无论如何, 保持对服务请求效果的预测, 虽然增加了响应的的时间, 但可以使提供者对请求保持较高的灵活度, 对于后续到来的紧急请求, 可以进行插队等灵活处理.

算法 2 本质属贪心方法, 不断选择响应因子最高的请求调度执行, 因此算法结果是局部最优的. 尽管考察一段时间区间总体未必最优, 但考虑到服务请求和响应是随时间而不断持续的过程且具有时效性, 局部最优更具有实际意义.

4 实验及结果分析

本文重点讨论了响应有效度和响应获益度的计算方法, 其中后者是通过服务业务目标的分解并映射到服务环境数据进行计算, 在实际应用中可直接通过制定规则来实现, 故未对其进行专门实验. 实验主要针对基于相似服务请求来预估响应有效度并根据算法 1 和算法 2 进行服务响应选择. 通过采用预测算法前后响应成功率的变化来考察算法的有效

性. 为了使实验具有通用性及便于模拟,选择几个与服务本身执行密切相关但和业务功能无关的基础数据指标进行实验,包括执行截止期、输入复杂度、执行逻辑复杂度、请求级别等. 请求级别用于表示请求的紧急程度,紧急程度高的优先执行. 引入输入复杂度,主要是考虑通常情况下服务需要处理的输入数据越多,其所耗费的资源和时间就越多. 另外,所请求服务方法的执行逻辑越复杂,耗费的资源和时间也会越多,故引入逻辑复杂度. 一旦服务执行超出截止期限限制则执行失败. 模拟服务 s 每次响应一个请求(执行器为 1),每次接收最多 10 个请求(等待队列为 10); s 的理论执行时间设定为和输入数据、逻辑复杂度成线性关系,考虑到等待队列是执行队列的 10 倍,实验将截止期定义为理论执行时间长度的 10 倍左右,以免超时请求过多. 模拟执行服务 s 时,其实际执行时间在理论时间的基础上的随机扰动约为 10%;实际执行时间小于等于截止期表示执行成功,由请求者在 $(0,1]$ 之间随机给出响应有效度,超出截止期则执行失败,响应有效度为 0. 实验只模拟独立请求,未模拟流程中服务请求的情况. 最后,针对多执行器并发执行进行对比实验.

实验样本数据选择 1 000 和 10 000 两种,请求样本预先随机产生,在实验中不再变化. 实验针对表 1 所示 8 种情况各执行 20 次后进行结果对比分析. 由于只有 1 个执行器,且请求顺序固定不变,情况 A 理论上 20 次的执行结果都相同. 其余 7 种情况 B~H 由于请求顺序随机或执行预测算法,每次执行结果理论上都不会完全相同.

表 1 实验分类

情况	输入请求顺序是否随机	结果是否随机扰动 10%	是否执行预测算法
A	否	否	否
B	是	否	否
C	是	是	否
D	否	否	是
E	是	否	是
F	是	是	是
G	否	是	否
H	否	是	是

4.1 单执行器实验结果及分析

图 2 是 1 000 个请求运行 20 次执行成功的结

果. 情况 A 成功数保持 352 不变,符合理论预测;情况 B、C 和 G 由于未执行预测算法,成功数在 350~370 之间;执行预测算法的情况 D、E、F 和 H 开始时由于没有历史样本积累,成功数有一定差别,但是随着历史样本的积累,成功数趋近于 450 左右,成功率提高明显.

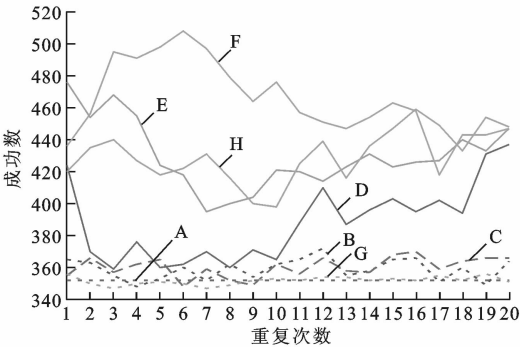


图 2 1 000 次响应重复 20 次的成功结果

图 3 为 1 000 次响应运行 20 次的失败结果. 未执行预测算法时(情况 A、B、C、G),无论执行顺序和结果是否随机变化,失败数基本保持在 25~50 之间. 执行预测算法后(情况 D、E、F、H),无论执行顺序、结果是否随机变化,刚开始由于没有历史样本数据,失败次数较高(基本保持在 15 以下),随着样本数据的积累,失败次数很快降低到 5 以下,说明算法可以明显地降低执行失败数.

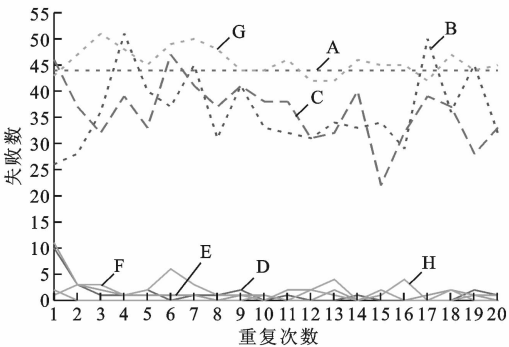


图 3 1 000 次响应重复 20 次的失败结果

图 4 是 1 000 个请求运行 20 次执行等待超时的结果. 未执行预测算法时(情况 A、B、C、G)等待超时数保持在 600 左右,执行预测算法后随着样本数据的积累,等待超时数趋向稳定在 550 左右,明显降低. 但是,总的超时数较高,原因在于实验模拟只有 1 个执行器,而等待队列为 10,导致请求等待过长.

图 5~图 7 是 10 000 次请求运行 20 次的成功、失败和超时结果. 与前面 1 000 次的结果类似,执行预测算法后(情况 D、E、F、H),成功数有较大提高,

执行失败和等待超时数有明显降低. 由于 10 000 次请求随着执行次数增加, 样本数量增长很快, 因此在

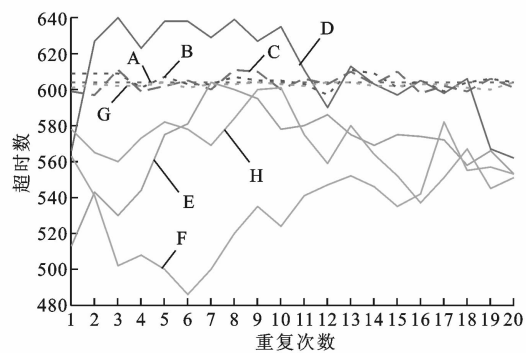


图 4 1 000 次响应重复 20 次的超时结果

对可能执行失败的请求的预测上效果比 1 000 次请求时有明显提高. 这说明服务提供者通过对请求、结果等信息的积累, 能够较为准确地实现对服务执行结果好坏的预测.

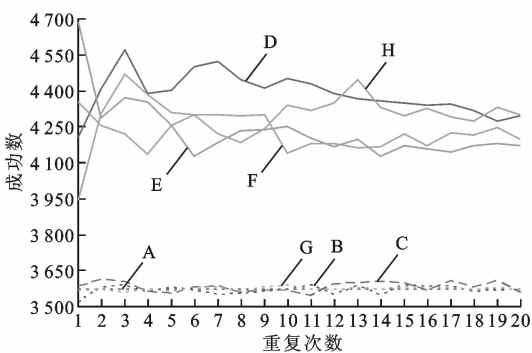


图 5 10 000 次响应重复 20 次的成功结果

综合上述实验结果可以看出, 引入预测算法后对服务响应的成功率有较明显提高. 同时也可以看出, 相同请求由于请求到来顺序不同、实际执行时间的波动等都会对执行的实际结果产生影响, 但总的趋势是比较明显的, 这也更符合实际环境情况.

4.2 多执行器并发响应对成功率的影响

1 个执行器时的等待超时数非常高(图 4、图 7), 因此针对执行器数目为 1、3、5 等 3 种情况进行了对比实验, 实验等待队列长度为 10, 结果如表 2 所示. 只有 1 个执行器时, 等待超时率超过 56%; 执行器为 3 个时, 降低到 37%; 5 个执行器时, 降低到 30%. 实验结果说明, 当服务提供者提供并发请求响应时, 可以有效降低请求的等待超时率, 提高成功率, 但是对于整体响应失败率影响不大.

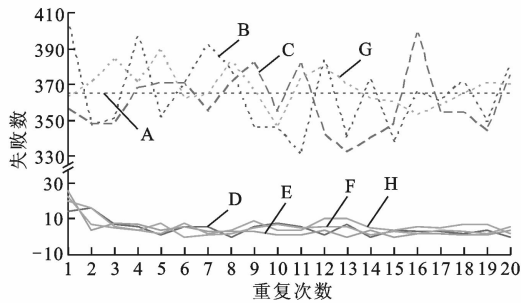


图 6 10 000 次响应重复 20 次的失败结果

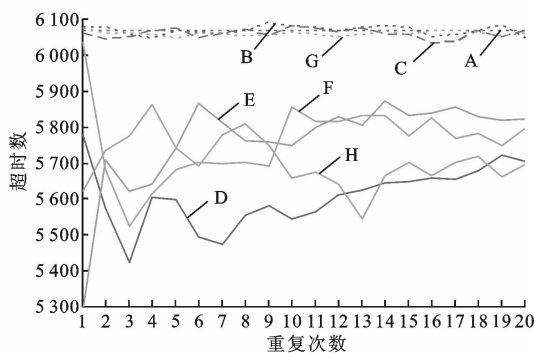


图 7 10 000 次响应重复 20 次的超时结果

5 结束语

与传统研究所考虑的服务提供者和服务请求者

表 2 多执行器并发执行结果统计

重复次数	1 个执行器			3 个执行器			5 个执行器		
	成功数	失败数	超时数	成功数	失败数	超时数	成功数	失败数	超时数
1	4 374	55	5 571	6 254	52	3 694	7 042	48	2 910
2	4 376	28	5 596	6 223	19	3 758	7 066	13	2 921
3	4 368	15	5 617	6 280	18	3 702	6 948	14	3 038
4	4 296	18	5 686	6 275	24	3 701	6 905	15	3 080
5	4 260	12	5 728	6 197	14	3 789	6 840	12	3 148
平均	4 335	26	5 640	6 246	25	3 729	6 960	20	3 019
百分比/%	43.35	0.26	56.4	62.46	0.25	37.29	69.6	0.2	30.19

的利益诉求相悖不同,本文基于二者协作的背景下,针对服务请求超出服务提供者服务能力时,如何从有益于提供者获益角度进行响应选择问题进行了研究;探讨了服务提供者获益的概念,以及通过服务响应有效度和获益度来进行响应选择的方法,并基于历史相似请求采用 KNN 算法实现对响应有效度的预测。实验结果表明,引入预测方法后可有效地提高服务响应的成功率,降低失败率和超时率。所述方法在实践中可帮助提供者更好地决策,实现将有限资源提供给最有助于其获益目标实现的请求者使用。

在今后的工作中,将对本文相关问题进一步深入研究,包括服务获益目标与服务环境参数之间关系的评价与自动化处理、算法在保证实时性方面的优化,以及响应选择策略在服务组合环境下对组合策略的影响等。

参考文献:

- [1] PAPAOGLOU M P, GEORGAKOPOULOS D. Service-oriented computing[J]. Communications of the ACM, 2003, 46(10): 25-28.
- [2] PAPAOGLOU M P. Service-oriented computing: concepts, characteristics and directions[C]//Proceedings of the Fourth International Conference on Web Information Systems Engineering. Washington, DC, USA: IEEE CS, 2003: 3-12.
- [3] DU L, DUAN C, LIU S, et al. Research on service bus for distributed real-time control systems[C]//Proceedings of 2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference. Washington, DC, USA: IEEE CS, 2011: 401-405.
- [4] LI W, TORDSSON J E E. An aspect-oriented approach to consistency-preserving caching and compression of Web service response messages[C]//Proceeding of 2010 IEEE Eighth International Conference on Web Services. Washington, DC, USA: IEEE CS, 2010: 526-533.
- [5] LI H. A queue theory based response time model for Web services chain[C]//Proceeding of 2010 International Conference on Computational Intelligence and Software Engineering. Piscataway, NJ, USA: IEEE, 2010: 1-4.
- [6] NIE W, ZHANG J, LIN K J. Estimating real-time service process response time using server utilizations [C]//Proceeding of 2010 IEEE International Conference on Service-Oriented Computing and Applications. Washington, DC, USA: IEEE CS, 2010: 1-8.
- [7] SCHARF M. On the response time of the large-scale

composite Web services[EB/OL]. [2012-06-05]. http://www.ikr.uni-stuttgart.de/Content/Publications/Archive/Sf_ITC19_36420.pdf.

- [8] REIFF-MARGANIEC S, YU H, TILLY M. Service selection based on non-functional properties [J]. Lecture Notes in Computer Science, 2009, 4907: 128-138.
- [9] 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, 2007, 1(1): 6.
- [10] ARDAGNA D, PERNICI B. Global and local QoS guarantee in web service selection[J]. Lecture Notes in Computer Science, 2006, 3812: 32-46.
- [11] 王显志,徐晓飞,王忠杰. 面向组合服务收益优化的动态服务选择方法[J]. 计算机学报, 2010, 33(11): 2104-2115.
WANG Xianzhi, XU Xiaofei, WANG Zhongjie. A profit optimization oriented service selection method for dynamic service composition[J]. Chinese Journal of Computers, 2010, 33(11): 2104-2115.
- [12] 朱锐,王怀民,冯大为. 基于偏好推荐的可信服务选择 [J]. 软件学报, 2011, 22(5): 852-864.
ZHU Rui, WANG Huaimin, FENG Dawei. Trustworthy services selection based on preference recommendation[J]. Journal of Software, 2011, 22(5): 852-864.
- [13] 张成文,苏森,陈俊亮. 基于遗传算法的 QoS 感知的 Web 服务选择[J]. 计算机学报, 2006, 29(7): 1029-1037.
ZHANG Chengwen, SU Sen, CHEN Junliang. Genetic algorithm on Web services selection supporting QoS [J]. Chinese Journal of Computers, 2006, 29(7): 1029-1037.
- [14] 范小芹,蒋昌俊,方贤文,等. 基于离散微粒群算法的动态 Web 服务选择[J]. 计算机研究与发展, 2010, 47(1): 147-156.
FAN Xiaoqin, JIANG Cangjun, FANG Xianwen, et al. Dynamic Web service selection based on discrete particle swarm optimization[J]. Journal of Computer Research and Development, 2010, 47(1): 147-156.
- [15] 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 Jianqiang, LI Juanzi, LIAO Guiping. A multi-QoS based local optimal model of service selection[J]. Chinese Journal of Computers, 2010, 33(3): 526-534.

(编辑 荆树蓉 赵大良)