

拟态构造 Web 服务器的服务质量量化方法

张杰鑫 庞建民 张铮 邵铭 刘浩

(数学工程与先进计算国家重点实验室 郑州 450001)

摘要 随着新兴的“互联网+”快速成为驱动社会经济发展的重要动力,Web 服务的地位越来越重要,其面临的安全问题也越来越严重。拟态构造 Web 服务器是一种基于拟态防御原理的新型 Web 防御系统,其利用异构性、动态性、冗余性等特性阻断或扰乱网络攻击。虽然其已经展开应用部署,并取得了较好的防御效果,但至今仍缺乏有效的服务质量量化评估方法。首先在分析拟态构造 Web 服务器系统架构的基础上,讨论其服务质量量化与传统的 Web 服务质量量化的区别和关键问题,分析了影响其服务质量的因素;然后基于“木桶”原理提出了拟态构造 Web 服务器服务质量的量化评估方法,并利用向量相似度方法量化服务质量的损耗值。文中在理论上为拟态构造 Web 服务器服务质量量化评估提供了一种新方法,在工程实践上为优化其服务质量提供了指导。仿真和实验结果表明,与现有的评价方法相比,提出的量化方法能够更加有效地量化评估拟态构造 Web 服务器的服务质量。

关键词 Web 服务,服务质量,拟态防御,QoS 属性,量化方法

中图分类号 TP311 **文献标识码** A **DOI** 10.11896/jsjcx.181001922

QoS Quantification Method for Web Server with Mimic Construction

ZHANG Jie-xin PANG Jian-min ZHANG Zheng TAI Ming LIU Hao

(State key Laboratory of Mathematical Engineering and Advanced Computing,Zhengzhou 450001,China)

Abstract As the emerging “Internet Plus” has quickly become an important driving force of social and economic development,Web service plays an increasing role in society,but its security issues are worsening. The Web server with mimic construction is a new Web defense system based on the principle of mimic defense,and it uses the heterogeneity,dynamics,redundancy and other characteristics to block or disrupt network attacks. Although it has been deployed and some better defense effects have been gotten,there is still a lack of effective methods for quantifying its QoS. On the basis of analyzing the system architecture of the Web server with mimic construction,this paper discussed the difference and issues between the quantification of its QoS and the quantification of traditional Web servers’ QoS,and analyzed the factors affecting its QoS. Based on the “Wood Barrel” principle,this paper proposed a quantitative evaluation method for the service quality of the Web server with mimic construction,and used the vector similarity method to quantify the loss value of the QoS. This effort provides a new method for quantifying the QoS of the Web server with mimic construction in theory,and provides guidance for optimizing its service quality in engineering practice. The simulation and experimental results show the proposed quantification method can effectively quantify and evaluate the QoS of the Web server with mimic construction compared with the existing evaluation methods.

Keywords Web service,Quality of service,Mimic defense,QoS attributes,Quantification method

随着互联网的快速发展,以及智能终端的不断增加,人们在金融服务、工业控制等领域对网络的依赖程度日益加剧,网络成为了人类社会发展的工具。与此同时,来自网络空间的安全威胁越来越大,各种涉及网络安全的事件频频发生,

严重影响了人类社会的各个方面。实践证明,绝大部分的网络安全事件都是借助漏洞和后门进行的^[1]。拟态防御技术为应对网络空间中已知和未知的漏洞和后门的威胁,以多样化技术为基础,采用“动态异构冗余”(Dynamic Heterogeneous

到稿日期:2018-10-15 返修日期:2019-02-06 本文受国家自然科学基金(61472447),国家重点研发计划(2016YFB0800104),上海市科学技术委员会科研计划(16DZ1120502)资助。

张杰鑫(1989-),男,博士,主要研究方向为网络空间安全、高效能计算;庞建民(1964-),男,教授,博士生导师,CCF 会员,主要研究方向为高性能计算、信息安全,E-mail:jianmin_pang@hotmail.com(通信作者);张铮(1976-),男,副教授,硕士生导师,主要研究方向为网络空间安全、先进计算;邵铭(1967-),男,副教授,主要研究方向为网络空间安全、先进计算;刘浩(1997-),男,硕士生,主要研究方向为网络空间安全、先进计算。

Redundancy, DHR)的系统架构,使用动态性、异构性和冗余等特性使系统具有内生的安全机理来提高系统的安全性^[2-3]。拟态构造 Web 服务器是拟态防御技术的一种典型实例,在不改变原有系统功能属性的基础上,使得系统具有一种内生的防御机制,但是拟态防御机制的引入对某些 Web 服务器的性能指标造成了一定的负面影响^[4-5]。这种负面影响可以被视为拟态防御技术对 Web 服务质量的影响,对拟态构造 Web 服务器的服务质量进行量化的研究是十分必要的,但是至今还缺乏对其进行量化的方法。

本文的主要贡献如下:

(1)分析拟态构造 Web 服务器的架构和特点,提出其服务质量量化的难点,以及与现有基于 QoS 的服务组合技术的异同;

(2)选取量化拟态构造 Web 服务器的服务质量的 QoS 属性,结合拟态防御技术的特点提出了一种拟态构造 Web 服务器服务质量的量化方法;

(3)提出 3 种提升拟态构造 Web 服务器服务质量的方法,并通过仿真和实验验证了这 3 种方法的有效性。

1 相关工作

随着 Web 服务技术的不断发展,越来越多的 Web 服务被发布出来以供用户使用。但单个 Web 服务所提供的服务能力有限,难以满足用户的各种需求,因此,出现了 Web 服务组合(Web Service Composition, WSC)技术^[6]。Web 服务组合技术是利用现有的多个原子 Web 服务,通过按照一定顺序或者改变组合顺序,来创建出功能更加强大、质量更高的服务提供给用户^[7-8]。拟态构造 Web 服务器是将多个功能等价的 Web 服务冗余执行,因此可以从现有的 Web 服务组合技术中借鉴量化思路。

虽然 Web 服务组合技术可以发现满足用户需求的服务组合,但是发现结果是一批功能相似的服务组合,加之缺乏功能属性以外的评价参数,因此难以从中选出最佳的服务。为解决这一问题,基于非功能属性的 Web 服务组合方法被提出,基于 QoS(Quality of Service)的 Web 服务组合是其中的重点^[9]。基于 QoS 的 Web 服务组合问题是一个组合优化问题,属于 NP 难问题。服务组合策略包括:局部最优策略^[10-11]、全局最优策略^[12-13]和混合策略^[14-15]。基于这 3 种策略的 Web 服务组合算法有穷举算法^[16]、贪婪算法^[17-18]、遗传算法(Genetic algorithm, GA)^[19-21]、粒子群优化(Particle Swarm Optimization, PSO)算法^[22-23]、蚁群优化(Ant Colony Optimization, ACO)算法^[24-25]和融合算法^[26-28]等。以上算法的核心是基于 QoS 属性的 Web 服务量化,可以满足现有的大规模 Web 服务组合需求,虽然难以直接将这些算法和其量化评估方法直接应用于执行体数量较少的拟态构造 Web 服务器,但是仍可以借鉴这些算法中对服务质量的量化方法。文献[4]提出了拟态构造 Web 服务器的系统构造和设计方法。文献[5]提出了合理且系统地拟态构造 Web 服务器的测试方法。这两个研究都对拟态构造 Web 服务器的性能指

标损耗情况进行讨论,并没有提出评价其服务质量的方法。因此,本文针对缺乏有效的量化评估拟态构造 Web 服务器服务质量的方法的问题,提出了一种有效的量化评估方法,该方法不但适用于拟态构造 Web 服务器,还为量化其他拟态防御系统的服务质量提供了参考。

2 服务质量量化的关键问题

拟态防御技术以软硬件多样性为基础,以异构性最大化为主要目标,综合了冗余、表决等技术,通过 DHR 架构提升安全性。本节论述拟态构造 Web 服务器的防御原理以及其与现有的 Web 服务组合的区别,阐述拟态构造 Web 服务器的服务质量量化的关键问题。

2.1 拟态构造 Web 服务器的架构

定义 1(执行体) 拟态构造 Web 服务器提供的真实 Web 服务的实体,记为 A_i 。

若执行体 A_i 与执行体 A_j 相同,则记为 $A_i = A_j$;若执行体 A_i 与执行体 A_j 不同,则记为 $A_i \neq A_j$ 。

定义 2(执行体集) 拟态构造 Web 服务器同一时刻上线工作,且一起参与表决的执行体组成的集合,记为 $A = \{A_i | A_i \text{ 是一个执行体,且 } i = 1, 2, \dots, n\}$ 。

拟态构造 Web 服务器的系统架构如图 1 所示。拟态构造 Web 服务器依据拟态防御原理,构建功能等价的、多样化的、动态化的非相似 Web 虚拟机池,采用冗余度表决、执行体冗余执行等技术来阻断攻击链,基于负反馈机制的中心调度器利用动态执行体调度来减小同种攻击连续发生的概率,从而增大漏洞后门或病毒等的利用难度^[29]。拟态构造 Web 服务器使用动态、异构、冗余、表决等特性来保证系统的可用性、可靠性和安全性。

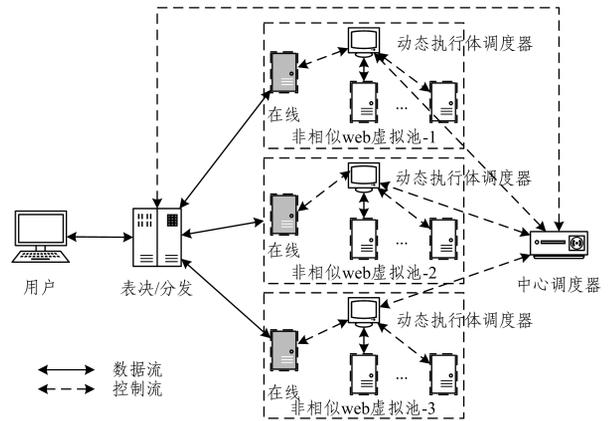


图 1 拟态构造 Web 服务器的系统架构

Fig. 1 System structure of Web server with mimic construction

拟态构造 Web 服务器注重软硬件之间的异构性。异构性以多样性为基础,功能等价的软硬件在设计、实现、运行的过程中不完全相同,使得特定硬件出现的某些错误在其他软硬件中不会发生。拟态构造 Web 服务器的冗余是指对于同一个输入,由 2 个或 2 个以上功能等价的组件并发执行,即冗余执行。拟态构造 Web 服务器的表决特性与冗余是紧密关联的,多个异构、冗余组件的响应结果应当是等价的,通过

表决可以从多个冗余组件的输出结果中得到相对正确的输出,并发现异常组件。拟态构造 Web 服务器的动态特性使得系统具有了动态性,打破了其长期以来的静态性,通过对资源进行优化调度,实现了异构组件的优化组合,避免了相同故障或攻击的再次发生。

2.2 拟态构造 Web 服务器的服务质量量化的关键问题

拟态构造 Web 服务器基于拟态防御的“动态异构冗余”的典型构造,与安全系统、Web 服务组合技术有着本质的区别,其服务质量量化面临以下 3 个关键问题:

(1)难以设计对拟态构造 Web 服务器的合理、有效和客观的测试方法,且难以取得准确、有效、客观的 QoS 数据。首先,Web 服务组合的动态性和分布式特点使得其测试技术和方法与传统的软件测试有很大的区别,因此测试方法的有效性、客观性会直接影响 QoS 数据的有效性、客观性。其次,基于 QoS 的 Web 服务量化仍有一些不足,即忽视了 QoS 内在的不确定性,易导致服务选择的可靠性差。实际环境中,由于受到利益化的市场竞争和资源管理对外的封闭性等因素的影响,不能保证服务提供者发布的服务是客观、真实、可信的^[30]。全球网民分布广泛、素质参差不齐,因此每个人使用服务后反馈的数据不一定相同、真实^[31]。拟态构造 Web 服务器服务质量的量化也面临着同样的问题。

(2)难以对执行体集的 Web 服务质量进行量化。异构的 Web 服务器所组成的执行体集提供了真实的 Web 服务,是用户、攻击者所直接面对的服务实体,因此,拟态构造 Web 服务器的服务质量的量化主体就是执行体集。传统的 Web 服务组合技术的分支结构中只有一个服务工作,如图 2(a)所示;传统的 Web 服务组合技术的并行结构中有两个功能不等价的服务同时工作,如图 2(b)所示;而在拟态构造 Web 服务器中执行体集内功能等价的异构执行体并发工作,如图 2(c)所示。拟态构造 Web 服务器中执行体集内的功能等价的异构执行体的 Web 服务质量是不同的,加之执行体是并发执行的,因此对系统服务质量的量化更加复杂、困难。

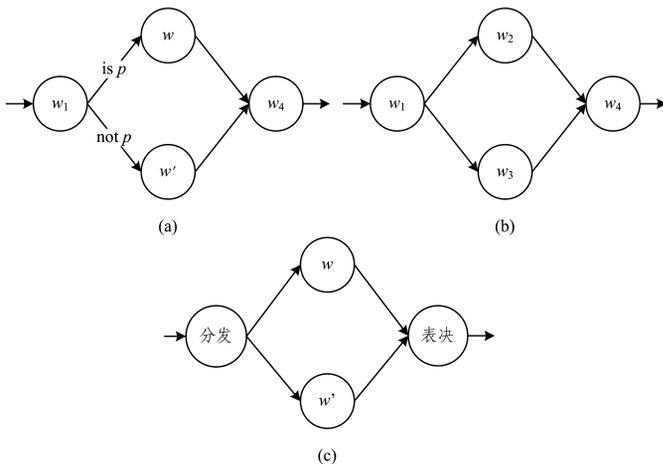


图 2 Web 服务组合与拟态构造 Web 服务器的区别

Fig. 2 Difference between Web service composition and Web server with mimic construction

分发和表决机制是拟态防御技术的关键,分发机制保证了执行体集内的每个执行体能够收到相同的请求,表决机制可以保证系统能够感知不一致的响应结果。同时,分发和表决机制也带来了系统服务质量的损耗,如响应时间增加、吞吐量降低等。因此,在量化拟态构造 Web 服务器的服务质量时,要将分发和表决机制对服务质量的影响加入其中。

3 拟态构造 Web 服务器服务质量的量化

与传统的防火墙、Web 服务器和 Web 服务组合技术等相比,拟态构造 Web 服务器服务质量的量化有其特殊的难点。本节将从拟态构造 Web 服务器 QoS 属性的选取和量化,执行体集 Web 服务的量化、分发和表决机制对 Web 服务质量的影响等几方面展开讨论。

3.1 QoS 属性的选取与量化

定义 3 (Web 服务 QoS) Web 服务 QoS 指软件在以 Web 服务这种形式提供服务时,与其行为相关的质量属性集合。该定义反映了 Web 服务 QoS 的特点,即 QoS 是由一些非功能属性组成,包括服务价格(service price)、响应时间(response time)、可用性(availability)和可靠性(reliability)等^[32]。在 Web 服务组合技术中 QoS 属性可以分为积极属性和消极属性两类。积极属性要求 QoS 最大化,如吞吐量(throughput)、可用性等;消极属性要求 QoS 最小化,如服务价格、响应时间等。

定义 4 (QoS 属性集) QoS 属性集是能够量化 Web 服务质量的 QoS 属性的集合,记为 $P = \{p_1, p_2, \dots, p_i, \dots, p_m\}$ 。其中 p_i 是一个 QoS 属性,如响应时间、吞吐量; m 为 QoS 属性数量。

基于拟态防御原理,结合 Web 服务器、防火墙等网络设备性能测试,拟态构造 Web 服务器选取了能够体现服务性能的客观 QoS 属性来构成其 QoS 属性集,包括每秒事务处理次数(Transactions Per Second, TPS)、最大 TCP 并发连接数(Max Concurrent TCP Connection Capacity, MCTCC)、吞吐量和响应时间。

每秒事务处理次数:每秒钟系统能够处理事务的数量。它是衡量系统处理能力的重要指标^[33],记作 q_{tps} ,单位为 transactions/s。

最大 TCP 并发连接数:穿过网关的主机之间或主机与网关之间能同时建立的最大 TCP 连接数^[34],记作 q_{mctcc} ,单位为 connections。

吞吐量:网络设备的数据包转发能力,即单位时间内成功转发数据的数量^[35],记作 q_p ,单位为 kbps。

响应时间:客户端发出请求到得到响应的整个过程的时间^[36],记作 q_r ,单位为 ms。

以上拟态构造 Web 服务器的执行体的 Web 服务 QoS 属性均可以通过性能测试得到其测量值^[5],由于不同的 QoS 属性测量值可能会相差几个数量级,因此将不同种类的 QoS 属性直接进行量化会因数量级的差异影响量化的公平性。而且,有些 QoS 属性属于积极属性,其值越大,服务的 QoS 度量

(3)难以对分发和表决机制对服务质量的影响进行量化。

值就会越高,如每秒事务处理次数、最大 TCP 并发连接数和吞吐量。而有些 QoS 属性属于消极属性,其值越大,服务的 QoS 度量值就会越低,如响应时间。因此,在测量出执行体集中某一执行体的 Web 服务第 i 项的 QoS 属性测量值 q_i 后,要对 QoS 属性进行归一化处理(如式(1)、式(2)所示),进而得到能够统一量化的无量纲的 QoS 属性值。

$$Q_i = \begin{cases} \frac{q_i}{\max(q_i)}, & \max(q_i) \neq 0 \\ 1, & \max(q_i) = 0 \end{cases} \quad (1)$$

$$Q_i = \begin{cases} \frac{\min(q_i)}{q_i}, & \min(q_i) \neq 0 \\ \frac{\chi}{q_i}, & \min(q_i) = 0 \text{ 且 } \max(q_i) \neq 0 \\ 1, & \max(q_i) = 0 \end{cases} \quad (2)$$

其中, $\max(q_i)$ 和 $\min(q_i)$ 是所有执行体的第 i 项 QoS 属性经过测试得到的最大测量值与最小测量值, χ 为便于计算的一个大于 0 且小于次小测量值的常数。针对 QoS 的积极属性则使用式(1),针对 QoS 的消极属性则使用式(2),并求出第 i 项 QoS 属性的归一化结果 Q_i ,且 $0 \leq Q_i \leq 1$ 。

3.2 执行体集 Web 服务质量的量化

3.1 节已经量化了执行体集的每种 QoS 属性值,为便于量化执行体集的 Web 服务质量,采用了执行体集 QoS 属性矩阵 E 。

$$\text{定义 5} \quad \text{QoS 属性矩阵 } E = \begin{bmatrix} Q_{1,1} & \cdots & Q_{1,m} \\ \vdots & \ddots & \vdots \\ Q_{n,1} & \cdots & Q_{n,m} \end{bmatrix}, \text{ 其中, } Q_{i,j}$$

表示 Web 服务的一个 QoS 属性值, n 为 Web 服务数量, m 为 QoS 属性的数量。

定义 6 执行体的 QoS 属性向量 V_{A_i} 由 QoS 属性矩阵 E 的一行构成,即 $V_{A_i} = (Q_{i,1}, Q_{i,2}, \dots, Q_{i,m})^T$,其中, m 为 QoS 属性的数量。

执行体集作为拟态构造 Web 服务器的服务实体, n 个异构冗余的执行体作为一个整体向外提供服务。针对客户端发出的请求, n 个异构冗余的执行体都会相互独立地作出响应。因此,从“木桶”原理可知^[37],执行体集的某种 QoS 属性值的大小,并不取决于执行体集中执行体的该种 QoS 属性的最大值,而取决于最小值。因此,可以通过式(3)计算执行体集的第 i 种 QoS 属性值 Q_i^A ,其中 n 为执行体集中执行体的数量。

$$Q_i^A = f_{\min}(Q_{1,i}, Q_{2,i}, \dots, Q_{n,i}) \quad (3)$$

定义 7 执行体集的 QoS 属性向量 $V_A = (Q_1^A, Q_2^A, \dots, Q_m^A)^T$,其中, Q_i^A 为执行体集的第 i 种 QoS 属性值, m 为 QoS 属性的数量。

基于式(3),可以将执行体集的 QoS 属性矩阵 E 归约为执行体集的 QoS 向量 V_A 。由于不同的 QoS 属性在量化 Web 服务质量时的权重不一,因此需要针对实际场景确定 QoS 属性的权值。

定义 8 Web 服务 QoS 属性权重向量 $V_w = (\omega_1, \omega_2, \dots, \omega_m)^T$,其中, ω_i 为相应 QoS 属性的权值,表示该 QoS 属性的重要程度; m 为 QoS 属性的数量,并且 $0 \leq \omega_i \leq 1, \sum_{j=1}^m \omega_j = 1$ 。

执行体集 A 的 Web 服务质量可由式(4)计算得出,其中某一执行体 A_i 的 Web 服务质量可由式(5)计算得出。

$$WS_A = V_A \cdot V_w = \sum_{j=1}^m (Q_j^A \times \omega_j) \quad (4)$$

$$WS_{A_i} = V_{A_i} \cdot V_w = \sum_{j=1}^m (Q_{i,j} \times \omega_j) \quad (5)$$

3.3 执行体集 Web 服务质量的损耗量化

由 3.2 节可知,可以根据各个 QoS 属性值来计算出拟态构造 Web 服务器的执行体集的 Web 服务质量 WS_A 和某一执行体的 Web 服务质量 WS_{A_i} 。由此,可以得到执行体集相比其每个执行体的 Web 服务质量损耗值,记为 L_{A-A_i} 。由执行体集的 QoS 属性矩阵 $E =$

$$E = \begin{bmatrix} 0.9 & 1 & 0.8 & 0.9 \\ 0.8 & 0.7 & 1 & 0.9 \\ 0.8 & 0.7 & 0.7 & 1 \\ 1 & 0.8 & 0.7 & 0.9 \end{bmatrix} \text{ 计算出}$$

的 $V_{A_i}, V_A, WS_{A_i}, WS_A$ 和 L_{A-A_i} 如表 1 所列,其中 $V_w = (0.25, 0.25, 0.25, 0.25)^T$ 。

$$L_{A-A_i} = \frac{WS_{A_i} - WS_A}{WS_{A_i}} \times 100\% \quad (6)$$

表 1 服务质量的量化值和损耗值

Table 1 Quality and loss values of service quantization

V_{A_i}/V_A	WS_A/WS_{A_i}	$L_{A-A_i}/\%$	α_{A-A_i}	β_{A-A_i}	γ_{A-A_i}
$V_{A_1} = (0.9, 1, 0.8, 0.9)^T$	0.90	13.89	0.8634	0.9158	0.7907
$V_{A_2} = (0.8, 0.7, 1, 0.9)^T$	0.85	8.82	0.9091	0.9001	0.8183
$V_{A_3} = (0.8, 0.7, 0.7, 1)^T$	0.80	3.13	0.9631	0.9679	0.9321
$V_{A_4} = (1, 0.8, 0.7, 0.9)^T$	0.85	8.82	0.9091	0.9375	0.8523
$V_A = (0.8, 0.7, 0.7, 0.9)^T$	0.775				

虽然通过式(6)能够直观地给出 Web 服务质量的损耗值,但是有时这种量化值并不能准确地反映出异构、冗余机制对 Web 服务质量的影响。如 L_{A-A_2} 与 L_{A-A_1} 相同, V_A 与 V_{A_2} 相比,第 1,2,4 个 QoS 属性相同,第 3 个 QoS 属性降低了 0.3; V_A 与 V_{A_1} 相比,第 3,4 个 QoS 属性相同,第 1,3 个 QoS 属性分别降低了 0.2 和 0.1。 V_A 与 V_{A_2} 相比,第 3 个 QoS 属性降低幅度大;而 V_A 与 V_{A_1} 相比,第 1,3 个 QoS 属性降低幅度相对较小,因此, L_{A-A_2} 大于 L_{A-A_1} 更加符合实际情况。产生这种问题的原因在于:式(6)忽略了向量间角度的差异对服务质量的影响。因此,基于文献^[38]所提出的量化向量相似度方法来更加准确地量化 Web 服务质量的损耗值,本文方法利用向量范数和向量方向来计算向量相似度。向量的相似度越高则 Web 服务质量的损耗值越低,反之向量的相似度越低则 Web 服务质量的损耗值越高。

定义 9 执行体集的 QoS 属性向量 V_A 与执行体的 QoS 属性向量 V_{A_i} 的范数相似度为 $\alpha_{A-A_i} = 1 - \left| \frac{\|V_{A_i}\| - \|V_A\|}{\|V_{A_i}\|} \right|$,其中, $\|V_{A_i}\|$ 为执行体 A_i 的范数, $\|V_A\|$ 为执行体集 A 的范数。

定义 10 执行体集的 QoS 属性向量 V_A 与执行体的 QoS 属性向量 V_{A_i} 的方向相似度为 $\beta_{A-A_i} = 1 - \frac{\theta}{90}$,其中, θ 为 V_A 与 V_{A_i} 的夹角。

定义 11 执行体集的 QoS 属性向量 V_A 与执行体的 QoS

属性向量 \mathbf{V}_{A_i} 的相似度为 $\gamma_{A-A_i} = \alpha_{A-A_i} \times \beta_{A-A_i}$, 即范数相似度与角度相似度的乘积。

由定义 9—定义 11 可知:

- (1) $\|\mathbf{V}_{A_i}\| \geq \|\mathbf{V}_A\|$, 且 $\alpha_{A-A_i} \in [0, 1]$;
- (2) $0 \leq \theta \leq 90$, 且 $\beta_{A-A_i} \in [0, 1]$;
- (3) $\gamma_{A-A_i} \in [0, 1]$, 当 \mathbf{V}_A 和 \mathbf{V}_{A_i} 正交时 $\gamma = 0$, 当 \mathbf{V}_A 和 \mathbf{V}_{A_i} 范数相同且夹角为 0 时 $\gamma_{A-A_i} = 1$ 。

按照执行体集的 QoS 属性向量 \mathbf{V}_A 与执行体的 QoS 属性向量 \mathbf{V}_{A_i} 的相似度的量化方法, 计算出表 1 中的 4 个执行体与执行体集的范数相似度、方向相似度和相似度。由 $\alpha_{A-A_2} = \alpha_{A-A_1}$ 可知, 基于向量长度的范数相似度无法区分出 \mathbf{V}_{A_2} , \mathbf{V}_{A_1} 和 \mathbf{V}_A 之间的差异; 而 $\beta_{A-A_2} < \beta_{A-A_1}$ 且 $\gamma_{A-A_2} < \gamma_{A-A_1}$, 因此再采用向量方向的相似度后可区分出 \mathbf{V}_{A_2} , \mathbf{V}_{A_1} 和 \mathbf{V}_A 之间的差异, 即执行体集 A 的 Web 服务质量比执行体 A_2 的 Web 服务质量降低更多。综上可知, 执行体集 Web 服务质量的量化需要直观的量化值 WS, 当 WS 不能准确量化 Web 服务质量的损耗值时, 就需要利用向量相似度 γ 来更加准确地量化 Web 服务质量的损耗, 向量相似度是执行体集 Web 服务质量量化的有力补充。

3.4 拟态构造 Web 服务器服务质量的量化

分发和表决机制是拟态构造 Web 服务器安全性的关键, 而分发和表决机制必将 Web 服务质量造成影响。如图 3 所示, 分发模块、表决模块和执行体集组成了一个串联结构, 是一个典型的 I-P-O 模型, 分发模块是执行体集的输入, 表决模块是执行体集的输出。

分发和表决机制可能限制执行体集 Web 服务的某些 QoS 属性性能, 进而降低其 Web 服务质量, 如每秒事务处理次数、最大 TCP 并发连接数、吞吐量; 而分发和表决机制会加大执行体集 Web 服务的某些 QoS 属性的性能损耗, 进而降低其 Web 服务质量, 如响应时间延长。

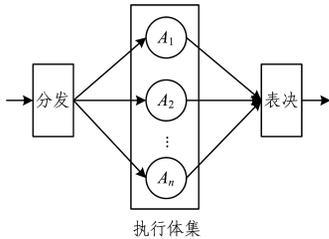


图 3 I-P-O 模型图

Fig. 3 I-P-O model diagram

执行体集 Web 服务质量是拟态构造 Web 服务器的理想值, 而分发和表决机制加入后的 Web 服务质量是拟态构造 Web 服务器的 Web 服务的实际值, 量化拟态构造 Web 服务器的 Web 服务质量的实际值和损耗可以参考 3.2 节和 3.3 节中的方法, 具体步骤如下。

测量出拟态构造 Web 服务器的各项 QoS 属性测量值, 记为 $q_j(\tau_{ws})$; 测量出其中所有执行体的 QoS 属性测量值, 记为 $q_j(A_i)$, 其中 $i = 1, 2, 3, \dots, n, j = 1, 2, 3, \dots, m$ 。

对 $q_j(\tau_{ws})$ 和所有 $q_j(A_i)$ 按照式 (1) 和式 (2) 进行归一化

处理, $q_j(\tau_{ws})$ 归一化后的结果记为 $Q_j, q_j(A_i)$ 归一化后的结果记为 $Q_{i,j}$ 。

$$\text{执行体集 } A \text{ 的 QoS 属性矩阵 } \mathbf{E} = \begin{bmatrix} Q_{1,1} & \cdots & Q_{1,m} \\ \vdots & \ddots & \vdots \\ Q_{n,1} & \cdots & Q_{n,m} \end{bmatrix}, \text{ 执}$$

行体集 A 的 QoS 属性向量 $\mathbf{V}_A = (Q_1^A, Q_2^A, \dots, Q_m^A)^T$, 拟态构造 Web 服务器的 QoS 属性向量 $\mathbf{V}_{WS} = (Q_1, Q_2, \dots, Q_m)^T$ 。

设 QoS 属性权重向量为 $\mathbf{V}_w = (\omega_1, \omega_2, \dots, \omega_m)^T$, 则拟态构造 Web 服务器的服务质量为:

$$WS = \mathbf{V}_{WS} \cdot \mathbf{V}_w = \sum_{j=1}^m (Q_j \times \omega_j) \quad (7)$$

由式 (4) 和式 (5) 得到执行体集 A 的服务质量 WS_A 以及各个执行体的服务质量 WS_{A_i} 。由式 (6) 和 3.3 节的方法得到拟态构造 Web 服务器相比执行体集 A 和各个执行体的 Web 服务质量的损耗值。

4 提高拟态构造 Web 服务器服务质量的方法

基于上节提出的拟态构造 Web 服务器服务质量的量化方法, 本节从理论层面分析并总结了提高服务质量的方法。

4.1 执行体的选取方法

理想状态下, 当所有执行体的各项 QoS 属性值相同时, 执行体集 Web 服务质量最大, 损耗为 0。然而实际应用中, 不存在理想状态, 即使相同的物理主机、相同的 Web 服务应用, 各项 QoS 属性的测量值也不会相同, 因此各项 QoS 属性值也不会相同, 因此总结出方法 1。

方法 1 应选取各项 QoS 属性值相近的执行体, 且执行体的各项 QoS 属性值越相近, 执行体集的 Web 服务的质量就越大, 损耗就越小。

4.2 执行体集冗余度的选取方法

$$\text{设执行体集 } A \text{ 的 QoS 属性矩阵为 } \mathbf{E} = \begin{bmatrix} Q_{1,1} & \cdots & Q_{1,m} \\ \vdots & \ddots & \vdots \\ Q_{n,1} & \cdots & Q_{n,m} \end{bmatrix},$$

执行体集的 Web 服务质量为 WS_A , 则必有:

$$WS_A \leq WS_{A_i} = f_{\min}(WS_{A_1}, WS_{A_2}, \dots, WS_{A_n})$$

将执行体 A_i 从执行体集中去除后, 组成新的执行体集, 记为 A^i 。为便于量化执行体集的 Web 服务质量, 不重新对 QoS 属性进行归一化处理, 则必有:

$$WS_A \leq WS_{A^i} \leq WS_{A_j} = f_{\min}(WS_{A_1}, \dots, WS_{A_{i-1}}, WS_{A_{i+1}}, \dots, WS_{A_n})$$

不断去掉执行体集中的某些执行体, 则至少存在一个执行体 A_k , 在去掉该执行体后执行体集 A^k 使得 $WS_A \leq WS_{A^i} \leq \dots < WS_{A^k}$, 即通过不断去除执行体集中服务较低的执行体, 减少执行体集的冗余度, 新的执行体集的 Web 服务质量会高于原来的执行体集 Web 服务质量。

假设, 执行体集的冗余度为 $N = 2i + 1$, 其中 $i = 1, 2, 3, \dots$, 则当 $N = 3$ 时, 执行体集 Web 服务质量最大^[2]。

由此可以总结出方法 2。

方法 2 当执行体集中执行体的 Web 服务质量存在差异

时,可以通过去除某些执行体来降低执行体集的冗余度,以提升执行体集 Web 服务质量,且当执行体集的冗余度为 3 时,执行体集 Web 服务质量达到最大值。

4.3 分发和表决模块的设计方法

设执行体集 A 与分发和表决模块进行 QoS 属性归一化后,执行体集 A 的 QoS 属性向量为 $V_A = (Q_1^A, Q_2^A, \dots, Q_m^A)^T$, 且 $V_w = (w_1, w_2, \dots, w_m)^T$, 若分发和表决模块的 QoS 属性向量为 $V_D = (Q_1^D, Q_2^D, \dots, Q_m^D)^T$, 则加入分发和表决模块后的拟态构造 Web 服务器的服务质量 WS 为:

$$WS = \sum_{j=1}^m (f_{\min}(Q_j^A, Q_j^D) \times w_j)$$

其中,忽略响应时间对 QoS 属性归一化造成的影响。若分发和表决模块中至少存在一个 QoS 属性使得 $Q_j^A \geq Q_j^D$, 则必有:

$$WS \leq WS_A$$

由此可以总结出方法 3。

方法 3 在设计分发和表决模块时,应提升分发和表决模块的性能,尽量使之不成为拟态构造 Web 服务器服务质量的瓶颈。

5 仿真与实验

本节采用本文提出的量化方法,结合模拟仿真和具体实验,对拟态构造 Web 服务器的服务质量进行量化评估。

5.1 仿真实验

图 4 是由 3 个执行体组成的执行体集的 Web 服务质量的变化情况,颜色由蓝到红表示执行体集的 Web 服务质量由低到高。其中, $V_w = (0.25, 0.25, 0.25, 0.25)^T$, 执行体 A_1 的 QoS 属性向量为 $V_{A_1} = (0.02 * x, 0.02 * x, 0.02 * x, 0.02 * x)^T$, 则 $WS_{A_1} = 0.02 * x, x = 1, 2, 3, \dots, 50$; 执行体 A_2 的 QoS 属性向量为 $V_{A_2} = (0.02 * y, 0.02 * y, 0.02 * y, 0.02 * y)^T$, 则 $WS_{A_2} = 0.02 * y, y = 1, 2, 3, \dots, 50$; 执行体 A_3 的 QoS 属性向量为 $V_{A_3} = (0.02 * z, 0.02 * z, 0.02 * z, 0.02 * z)^T$, 则 $WS_{A_3} = 0.02 * z, z = 1, 2, 3, \dots, 50$ 。由图 4 可知,随着 3 个执行体 Web 服务质量的同步增加,执行体集的 Web 服务质量也增加;而当有执行体的 Web 服务质量低于其他执行体时,执行集的 Web 服务质量也随之降低。

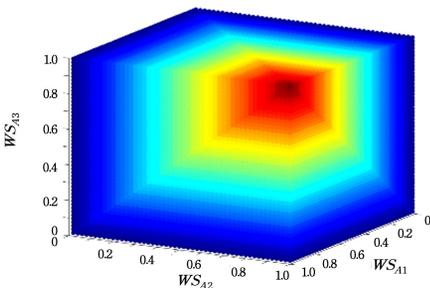


图 4 执行体的 Web 服务质量与执行体集的 Web 服务质量的关系(电子版为彩色)

Fig. 4 Relationship between executors' Web service quality and executor set's Web service quality

随机生成 QoS 属性不同的 20 个异构执行体,由其组成

的 QoS 属性矩阵 E 如式(8)所示,其中, $V_w = (0.25, 0.25, 0.25, 0.25)^T$ 。由 20 个执行体组合得到的执行体集 Web 服务质量如图 5 所示,假设执行体集的冗余度 N 分别为 3, 5, 7, 9, 11, 13, 15, 17, 19^[2]。由图 5 可知,随着某些执行体的去除,以及执行体集的冗余度的降低,执行体集的 Web 服务质量显著提高,且当 $N=3$ 时,执行体集的 Web 服务质量最高。

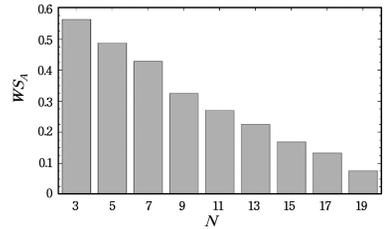
$$E = \begin{bmatrix} 0.5290041 & 1 & 0.5215451 & 0.3808992 \\ 0.6749291 & 0.5552906 & 0.6343438 & 0.4805395 \\ 0.9760769 & 0.5170872 & 0.8482169 & 0.4087858 \\ 0.7414622 & 0.3317269 & 0.5505798 & 0.8205074 \\ 0.410666 & 0.4313605 & 0.209176 & 0.776831 \\ 0.8538831 & 0.4933602 & 0.4698434 & 0.4552176 \\ 0.137893 & 0.0712615 & 0.4429478 & 0.7339639 \\ 0.0620668 & 0.8905442 & 1 & 1 \\ 0.0864763 & 0.0648378 & 0.6418439 & 0.8296883 \\ 0.1682351 & 0.4375632 & 0.7198261 & 0.7464683 \\ 0.3327989 & 0.8292414 & 0.7454713 & 0.1156715 \\ 0.3097106 & 0.3957813 & 0.3590851 & 0.4125318 \\ 0.0119901 & 0.6154133 & 0.5351575 & 0.6251548 \\ 0.5541912 & 0.8212274 & 0.576257 & 0.4860067 \\ 0.0978963 & 0.8890353 & 0.1619945 & 0.0532578 \\ 0.1503917 & 0.9340537 & 0.5818068 & 0.2419428 \\ 0.6478415 & 0.1913875 & 0.7192188 & 0.8825404 \\ 0.8820584 & 0.2593993 & 0.4414413 & 0.0165515 \\ 1 & 0.9007027 & 0.8656571 & 0.9137733 \\ 0.5859431 & 0.5952367 & 0.7570881 & 0.0825941 \end{bmatrix}$$


图 5 执行体集 Web 服务质量与冗余度的关系

Fig. 5 Relationship between executor set's Web service quality and redundancy

随机生成一个执行体集 A , 其 QoS 属性矩阵为 $E =$

$$\begin{bmatrix} 0.8763 & 1 & 1 & 1 \\ 0.2621 & 0.5617 & 0.7683 & 0.7044 \\ 1 & 0.7174 & 0.5708 & 0.6617 \end{bmatrix}, \text{ 设 } V_w = (0.25, 0.25,$$

$0.25, 0.25)^T$, 则执行体集 A 的 Web 服务质量为 $WS_A = 0.5141$ 。设分发和表决模块的 QoS 属性向量为 $V_D = (0.01 * x, 0.01 * x, 0.01 * x, 0.01 * x)^T$, 则分发和表决模块的服务质量 $WS_D = 0.01 * x, x = 1, 2, 3, \dots, 100$ 。若量化响应时间对拟态构造 Web 服务器的服务质量的影响,则将响应时间与其他 3 种 QoS 属性做相同处理,加入分发和表决模块后,拟态构造 Web 服务器的服务质量 WS 与分发和表决模块的服务质量 WS_D 的关系如图 6 所示。由图 6 可知,若 $WS_D < WS_A$, 则 $WS = WS_D$, 分发和表决模块成为了拟态构造 Web

服务器服务质量的瓶颈;若 $WS_D \geq WS_A$, 则 $WS = WS_A = 0.5141$, 分发和表决模块没有成为拟态构造 Web 服务器服务质量的瓶颈。

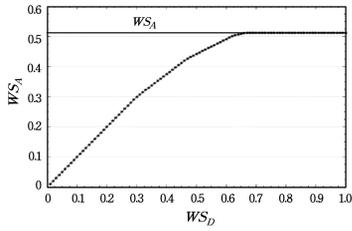


图 6 分发和表决模块对 Web 服务质量的影响

Fig. 6 Impact of dispatch and voter modules on quality of Web services

表 2 拟态构造 Web 服务器的配置

Table 2 Configuration of Web server with mimic construction

	CPU 型号	核心数	线程数	内存/GB	操作系统	Web 服务
物理主机配置信息 1	Intel(R) Xeon(R) CPU x5650	6	12	24	Windows Server 2008	IIS 7.0
物理主机配置信息 2	Intel(R) Xeon(R) CPU x5650	6	12	24	Centos 7	nginx 1.7.4
虚拟主机配置信息 1	Intel(R) Xeon(R) CPU x5650	1	2	2	Windows Server 2008	IIS 7.0
虚拟主机配置信息 2	Intel(R) Xeon(R) CPU x5650	1	1	1	Windows Server 2008	IIS 7.0 等

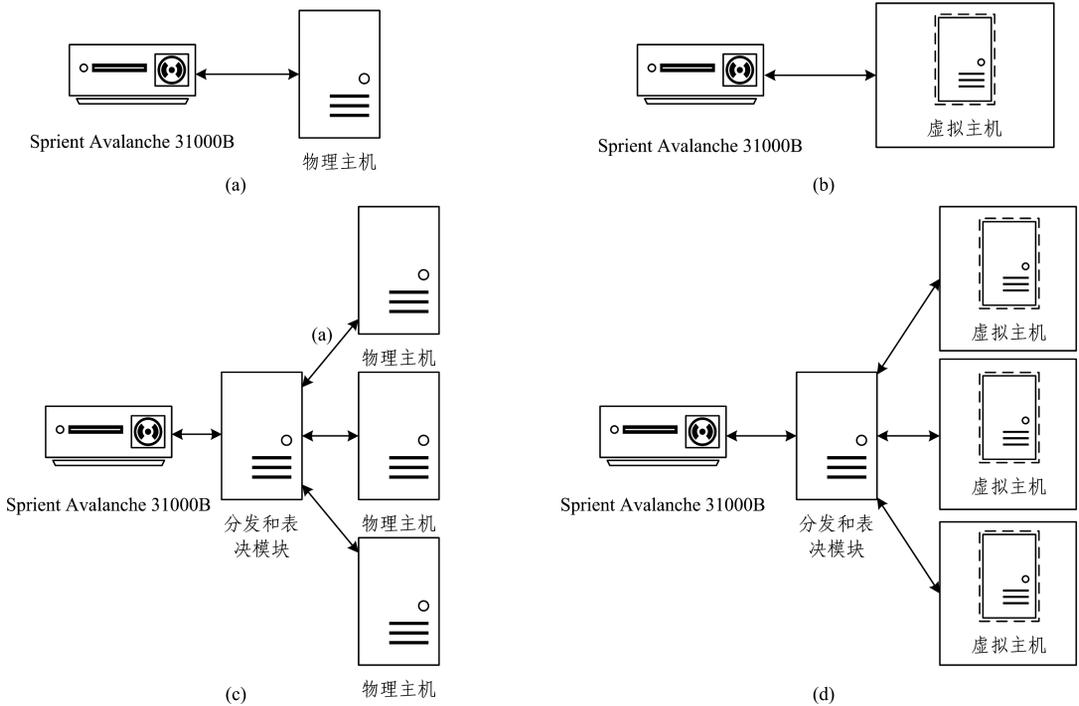


图 7 4 种测试对象

Fig. 7 Four test objects

首先,按照图 7(a)和图 7(b),分别测出以物理主机承载 Web 服务的执行体 A_{c1}, A_{c2}, A_{c3} , 以及虚拟主机承载 Web 服务的执行体 A_{d1}, A_{d2}, A_{d3} 的 QoS 属性测量值。其中, A_{c3} 的配置如表 2 中物理主机配置信息 1 所列, A_{d1}, A_{d2}, A_{d3} 的配置如表 2 中虚拟主机配置信息 1 所列。依据方法 1, 选择同构的执行体来提高拟执行体集的 Web 服务质量, 如图 7(c)和图 7(d)所示。

设 $V_w = (0.25, 0.25, 0.25, 0.25)^T$, 依据本文提出的量化方法, 分别计算出图 7(c)和图 7(d)中的执行体集 A_c 和 A_d 的

5.2 实验验证

本节基于拟态构造 Web 服务器的性能测试数据, 对拟态构造 Web 服务器的服务质量进行量化评估, 配置信息如表 2 所列, 4 种测试部署如图 7 所示。图 7(a)和图 7(b)的测试对象分别为物理主机和虚拟主机承载 Web 服务, 其测量值作为执行体的 QoS 属性测量值; 图 7(c)和图 7(d)的测试对象分别为采用物理主机和虚拟主机承载 Web 服务来实现的拟态构造 Web 服务器, 其测量值作为拟态构造 Web 服务器的 QoS 属性测量值。采用 Spriant Avalanche 31000B 对每秒事务处理次数 (q_{tps})、最大 TCP 并发连接数 (q_{mtcc})、吞吐量 (q_{tp}) 和响应时间 (q_r) 4 个 QoS 属性进行测试。

Web 服务质量。测试和量化结果如表 3 所列, 两种执行体集 Web 服务质量损耗值很低。从测试和量化结果中可以得出以下结论:

- (1) 同构的执行体的相同 QoS 属性的测量值可能存在差异, 因此, 同构的执行体的 Web 服务质量也可能存在差异;
- (2) 选择同构的执行体组成的执行体集, 其 Web 服务质量也存在损耗;
- (3) 选择的执行体的服务质量越近, 则执行体集的 Web 服务质量越高, 且服务质量损耗越低。

表3 服务质量的量化值和损耗值

Table 3 Quality and loss values of service quantization

A_i/A	q_{tps}	q_{mctcc}	q_{tp}	q_{rt}	WS	$L_{AA_i}/\%$	α_{AA_i}	β_{AA_i}	γ_{AA_i}
A_{c1}	34584	792214	977000	0.002	0.9937	1.01	0.9900	0.9929	0.9829
A_{c2}	35192	781372	985000	0.002	0.9966	1.30	0.9871	0.9908	0.9780
A_{c3}	34800	787351	982000	0.002	0.9871	0.35	0.9964	0.9977	0.9942
A_c	34584	781372	977000	0.002	0.9836				
A_{d1}	5250	272871	414000	0.082	0.9867	1.45	0.9854	0.9929	0.9784
A_{d2}	5409	265667	419000	0.080	0.9844	1.23	0.9876	0.9915	0.9792
A_{d3}	5238	266228	412000	0.083	0.9818	0.95	0.9902	0.9900	0.9803
A_d	5238	265667	412000	0.083	0.9723				

然后,按照图 7(b)进行部署,测出表 4 中列出的以虚拟主机承载 Web 服务的拟态构造 Web 服务器的 QoS 属性测量值。其中各个执行体的配置如表 2 中虚拟主机配置信息 2 所列。设 $V_w = (0.25, 0.25, 0.25, 0.25)^T$, 计算出执行体集的冗余度 N 为 3, 5, 7, 9, 11 时的执行体集的最大 Web 服务质量, 结果如图 8 所示。从测试结果中可以得出以下结论:

(1) 异构执行体的 QoS 属性性能明显不同, 且异构执行体的 Web 服务质量存在显著差异;

(2) 异构冗余机制会对拟态构造 Web 服务器的服务质量造成损耗;

(3) 使用异构的执行体组成的执行体集 Web 服务质量, 随着执行体集的冗余度的降低而增大, 且当冗余度 $N=3$ 时,

执行体集的 Web 服务质量最大。

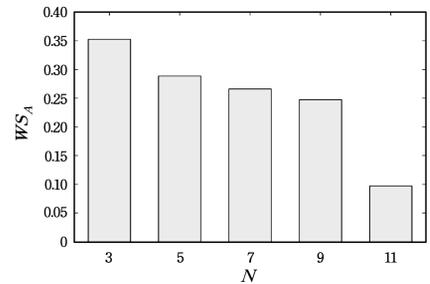


图8 执行体集 Web 服务质量与冗余度的关系图

Fig. 8 Relationship between executor set's Web service quality and redundancy

表4 QoS 属性测量值

Table 4 Measurement of QoS attribute

A_i	q_{tps}	q_{mctcc}	q_{tp}	q_{rt}	WS
Windows server 2003+nginx 1.7.4	8409	1024	322000	0.036	0.4792
Windows server 2003+apache 2.3.9	6937	265	276000	0.002	0.6397
Windows server 2003+IIS 6.0	3021	102517	304000	0.260	0.4692
Windows server 2003+lighttpd 1.4.36	5718	39492	89000	0.012	0.2853
Windows server 2008+nginx 1.7.4	2091	1184	305000	0.081	0.2726
Windows server 2008+apache 2.3.9	2165	254	289000	0.009	0.3120
Windows server 2008+IIS 7.0	5226	145754	322000	0.034	0.6336
Windows server 2008+lighttpd 1.4.36	1285	285	88000	6.608	0.0971
Ubuntu server 14.10+nginx 1.7.4	3052	2967	377000	0.013	0.3843
Ubuntu server 14.10+apache 2.3.9	2248	1467	302000	0.011	0.3151
CentOS 7+nginx 1.7.4	2809	1155	317000	0.007	0.3671
CentOS 7+apache 2.3.9	2167	1185	263000	0.003	0.4075

最后,按照图 7(a)、图 7(c)和图 7(d)所示,分别测出以物理主机承载 Web 服务的拟态构造 Web 服务器 M_c 、以虚拟主机承载 Web 服务的拟态构造 Web 服务器 M_d 和分发/表决模块 D 的 QoS 属性测量值。分发和表决模块的 QoS 属性测量值以 CentOS 7+nginx 1.7.4 为 Web 服务器的 QoS 属性测量值作为参考值,其配置如表 2 中物理主机配置信息 2 所列。物理主机的配置如表 2 中物理主机配置信息 1 所列,虚拟主机的配置如表 2 中虚拟主机配置信息 1 所示。

由 3.4 节的分析可知,拟态构造 Web 服务器的响应时间必然会高于其任意一个执行体,且由表 5 可知,拟态构造 Web 服务器的响应时间明显高于任意一个执行体,因此这里不讨论分发和表决模块的响应时间参考值对拟态构造 Web 服务器的服务质量的影响。图 7(d)的拟态构造 Web 服务器的 QoS 属性测量值均高于图 7(c)的拟态构造 Web 服务器。分发和表决模块的每秒事务处理次数、最大 TCP 并发连接数和吞吐量 3 个 QoS 属性均低于图 7(c)中的 3 个执行体,而每秒

事务处理次数和吞吐量高于图 7(d)中的 3 个执行体,因此,图 8 的拟态构造的服务器的服务质量损耗值低于图 7(c)的拟态构造 Web 服务器的服务质量损耗值。从测试结果中可以得出以下结论:

(1) 分发和表决机制会影响拟态构造 Web 服务器的服务质量;

(2) 分发和表决模块的性能越高,则拟态构造 Web 服务器服务质量的损耗值越低。

实验结果表明,两种情况中拟态构造 Web 服务器的服务质量分别降低了 60.11% 和 47.91%。相比文献[4-5]只能计算拟态构造 Web 服务器单一 QoS 属性的损耗,本文提出的方法不仅能够直观、准确地呈现其服务质量,而且能够精确地计算服务质量的损耗。

测试中的拟态构造 Web 服务器主要用于验证拟态防御原理,在性能方面没有做过多优化,因此拟态构造 Web 服务器的性能优化质量将影响其实际应用。而且,本次测试只采

用了 4 种 QoS 属性对拟态构造 Web 服务器的服务质量进行量化评估,若将安全性、可靠性、可维护性等 QoS 属性也加入

量化评估中,则拟态构造 Web 服务器的服务质量损耗值必会显著降低,其服务质量甚至会高于其执行体的 Web 服务质量。

表 5 服务质量的量化值和损耗值

Table 5 Quality and loss values of service quantization

	q_{tps}	q_{mctcc}	q_{tp}	q_{rt}	文献[4-5]方法				本文方法		
					q_{tps} 损耗/%	q_{mctcc} 损耗/%	q_{tp} 损耗/%	q_{rt} 损耗	WS	$L_{MA_i}/\%$	γ_{MA_i}
A_{c1}	34584	792214	977000	0.002	-48.77	-91.91	-0.51	+0.224	0.9937	60.25	0.2800
A_{c2}	35192	781372	985000	0.002	-49.66	-91.80	-1.32	+0.224	0.9966	60.37	0.2817
A_{c3}	34800	787351	982000	0.002	-49.09	-91.86	-1.02	+0.224	0.9949	60.30	0.2809
A_c	34584	781372	977000	0.002	-48.77	-91.80	-0.51	+0.224	0.9902	60.11	0.2819
M_c	17716	64086	972000	0.226					0.3950		
D	17910	64547	975000	0.001							
A_{d1}	5250	272871	414000	0.082	-0.82	-76.43	-23.43	+1.046	0.9836	48.51	0.3807
A_{d2}	5409	265667	419000	0.080	-3.73	-75.79	-24.34	+1.048	0.9934	49.01	0.3805
A_{d3}	5238	266228	412000	0.083	-0.59	-75.84	-23.06	+1.045	0.9728	47.93	0.3876
A_d	5238	265667	412000	0.083	-0.59	-75.79	-23.06	+1.045	0.9723	47.91	0.3880
M_d	5207	64327	317000	1.128					0.5065		
D	17910	64547	975000	0.001							

结束语 针对难以量化评估拟态防御对 Web 服务器服务质量造成影响的问题,借鉴 Web 服务组合技术,提出了量化执行体集 Web 服务质量的方法,在此基础上,将分发和表决机制纳入量化评估范围,最终提出了一种拟态构造 Web 服务器服务质量的量化评估方法。该方法可以有效地量化异构、冗余、分发和表决机制对 Web 服务器服务质量造成的影响和损耗。在该方法的基础上,提出了 3 种提升拟态构造 Web 服务器服务质量的方法,对拟态防御的工程实践具有指导意义。相比现有的量化方法,本文提出的量化方法能够更加直观、准确地呈现拟态构造的 Web 服务质量。

下一步将针对拟态防御的动态机制对 Web 服务质量造成的影响进行分析和讨论,改进和完善拟态构造 Web 服务器的动态调度策略。今后将研究拟态构造的 Web 服务器的可用性、可靠性、安全性、可维护性、成本、用户体验、信誉度等 QoS 属性的量化评估方法,并在此基础上改进和完善其 Web 服务质量的量化评估方法,研究更加合理的 Web 服务 QoS 属性权重确定方法,以更加全面地评估拟态构造 Web 服务器的服务质量。

参 考 文 献

[1] 中国互联网络信息中心. 中国互联网络发展状况统计报告[R]. 北京:2018.

[2] 邬江兴. 网络空间拟态防御导论[M]. 北京:科学出版社,2017.

[3] WU J X, ZHANG F, LUO X G. Mimic computing and Mimic Security Defense[J]. Communications of the CCF, 2015, 11(1): 8-14. (in Chinese)

邬江兴, 张帆, 罗兴国. 拟态计算与拟态安全防御[J]. 中国计算机学会通讯, 2015, 11(1): 8-14.

[4] TONG Q, ZHANG Z, ZHANG W H, et al. Design and implementation of mimic defense Web server[J]. Journal of Software, 2017, 28(4): 883-897. (in Chinese)

全青, 张铮, 张为华, 等. 拟态防御 Web 服务器设计与实现[J]. 软件学报, 2017, 28(4): 883-897.

[5] ZHANG Z, MA B L, WU J X. The Test and Analysis of Prototype of Mimic Defense in Web Servers[J]. Journal of Cyber Security, 2017, 2(1): 13-28. (in Chinese)

张铮, 马博林, 邬江兴. Web 服务器拟态防御原理验证系统测试与分析[J]. 信息安全学报, 2017, 2(1): 13-28.

[6] PETRIE C J. Web Service Composition[M]. Cham: Springer International Publishing, 2016.

[7] HWANG S Y, HSU C C, LEE C H. Service Selection for Web Services with Probabilistic QoS[J]. IEEE Transactions on Services Computing, 2017, 8(3): 467-480.

[8] JIANG W, HU S, LIU Z. Top K Query for QoS-Aware Automatic Service Composition[J]. IEEE Transactions on Services Computing, 2014, 7(4): 681-695.

[9] KUMAR S, MISHRA R B, SUGUMARAN V. A Hybrid Model for Service Selection in Semantic Web Service Composition[J]. International Journal of Intelligent Information Technologies, 2017, 4(4): 55-69.

[10] WEI L, YIN J, LI Y, et al. Efficient Web service QoS prediction using local neighborhood matrix factorization[J]. Engineering Applications of Artificial Intelligence, 2015, 38(2): 14-23.

[11] LIAN-YONG Q I, DOU W C. Web service composition method based on local QoS optimization in cross-organizational cooperation[J]. Computer Integrated Manufacturing Systems, 2011, 17(8): 1647-1653.

[12] HAMMAS O, YAHIA S B, AHMED S B. Adaptive Web service composition insuring global QoS optimization[C]// International Symposium on Networks, Hammamet, IEEE, 2015: 1-6.

[13] KANG G S, LIU J X, TANG M D, et al. Global Optimal Web Service Selection Model for Multiple Service Requests[J]. Journal of Computer Research and Development, 2013, 50(7): 1524-1533. (in Chinese)

康国胜, 刘建勋, 唐明董, 等. 面向多请求的 Web 服务全局优化选择模型研究[J]. 计算机研究与发展, 2013, 50(7): 1524-1533.

[14] RODRIGUEZMIER P, MUCIENTES M, LAMA M. Hybrid Optimization Algorithm for Large-Scale QoS-Aware Service

- Composition[J]. IEEE Transactions on Services Computing, 2017, 10(4): 547-559.
- [15] LIU Z Z, CHU D H, JIA Z P, et al. Two-stage approach for reliable dynamic Web service composition[J]. Knowledge-Based Systems, 2016, 97(4): 123-143.
- [16] ZENG L, BENATALLAH B, NGU A H H, et al. QoS-Aware Middleware for Web Services Composition[J]. IEEE Transactions on Software Engineering, 2004, 30(5): 359-364.
- [17] YU Q, REGE M, BOUGUETTAYA A, et al. A two-phase framework for quality-aware Web service selection[J]. Service Oriented Computing & Applications, 2010, 4(2): 63-79.
- [18] ZENG L, XU S, WANG Y, et al. Toward cost-effective replica placements in cloud storage systems with QoS-awareness[J]. Software Practice & Experience, 2017, 47: 813-829.
- [19] CHEN J, ZHOU H, ZHANG N, et al. Service-Oriented Dynamic Connection Management for Software-Defined Internet of Vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(10): 2826-2837.
- [20] SHAHROKH P, SAFI-ESFAHANI F. QoS-based Web service composition applying an improved genetic algorithm (IGA) method[J]. International Journal of Enterprise Information Systems, 2016, 12(3): 60-77.
- [21] DING Z J, LIU J J, SUN Y Q, et al. A Transaction and QoS-Aware Service Selection Approach Based on Genetic Algorithm[J]. IEEE Transactions on Systems Man & Cybernetics Systems, 2017, 45(7): 1035-1046.
- [22] WEN T, LI Y Q, SHENG G J, et al. Improved PSO-based Web service selection under uncertain information[J]. Journal of Jilin University, 2014, 44(1): 129-136. (in Chinese)
温涛, 李迎秋, 盛国军, 等. 不确定信息下基于改进粒子群算法的 Web 服务选择[J]. 吉林大学学报, 2014, 44(1): 129-136.
- [23] SILVAA S D, MA H, ZHANG M. A graph-based Particle Swarm Optimisation approach to QoS-aware Web service composition and selection[C]// 2014 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2014: 3127-3134.
- [24] LI G, BOUKHATEM L, WU J. Adaptive Quality-of-Service-Based Routing for Vehicular Ad Hoc Networks With Ant Colony Optimization[J]. IEEE Transactions on Vehicular Technology, 2017, 66(4): 3249-3264.
- [25] WANG D, HUANG H, XIE C. A Novel Adaptive Web Service Selection Algorithm Based on Ant Colony Optimization for Dynamic Web Service Composition[C]// International Conference on Algorithms and Architectures for Parallel Processing. New York: Springer, 2014: 391-399.
- [26] CAO T F, FU Y Q, ZHONG M Y. Based Web Service Composition with Genetic Algorithm and Ant Colony Optimization[J]. Computer Systems & Applications, 2012, 21(6): 81-85. (in Chinese)
曹腾飞, 符云清, 钟明洋. 融合遗传蚁群算法的 Web 服务组合研究[J]. 计算机系统应用, 2012, 21(6): 81-85.
- [27] DAI Y, LOU Y, LU X. A Task Scheduling Algorithm Based on Genetic Algorithm and Ant Colony Optimization Algorithm with Multi-QoS Constraints in Cloud Computing[C]// International Conference on Intelligent Human-Machine Systems and Cybernetics. IEEE, 2015: 428-431.
- [28] MA H, WANG A, ZHANG M. A Hybrid Approach Using Genetic Programming and Greedy Search for QoS-Aware Web Service Composition[M]. Transactions on Large-Scale Data- and Knowledge-Centered Systems XVIII. Springer Berlin Heidelberg, 2015: 180-205.
- [29] WU X. Meaning and Vision of Mimic Computing and Mimic Security Defense[J]. Telecommunications Science, 2014, 30(7): 1-7. (in Chinese)
邬江兴. 专题导读——拟态计算与拟态防御的原意和愿景[J]. 电信科学, 2014, 30(7): 1-7.
- [30] ALRIFAI M, DOLOGP, BALKE W T, et al. Distributed Management of Concurrent Web Service Transactions [J]. IEEE Transactions on Services Computing, 2009, 2(4): 289-302.
- [31] WANG S G, SUN Q B, YANG F C. Reputation evaluation approach in Web service selection[J]. Journal of Software, 2012, 23(6): 1350-1367. (in Chinese)
王尚广, 孙其博, 杨放春. Web 服务选择中信誉度评估方法[J]. 软件学报, 2012, 23(6): 1350-1367.
- [32] ZHANG C W, SU S, CHEN J L. Genetic Algorithm on Web Services Selection Supporting QoS[J]. Chinese Journal of Computers, 2006, 29(7): 1029-1037. (in Chinese)
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.
- [33] TUYA J, YOUNAS M. A Framework to Test Advanced Web Services Transactions[C]// Fourth IEEE International Conference on Software Testing, Verification and Validation. IEEE Computer Society, 2011: 443-446.
- [34] NEWMAN D. Benchmarking Terminology for Firewall Performance[S]. RFC 2647, 1999.
- [35] SUN H B, CHEN M, CAI Y B, et al. Research on Benchmarking Methods of IPv4/IPv6 Transition Gateway[J]. Computer Engineering, 2006, 32(24): 93-95. (in Chinese)
孙红兵, 陈沫, 蔡一兵, 等. IPv4/IPv6 转换网关性能测试方法研究[J]. 计算机工程, 2006, 32(24): 93-95.
- [36] MENASCE D. Response-time analysis of composite Web services[J]. Internet Computing IEEE, 2004, 8(1): 90-92.
- [37] PETER L J, Hull R. The peter principle[M]. London: Souvenir Press, 1969.
- [38] JIAO L M, YANG J L. A New Method for Calculating Weights [J]. Command Control & Simulation, 2006, 28(1): 94-97. (in Chinese)
焦利明, 杨建立. 一种确定指标权重的新方法[J]. 指挥控制与仿真, 2006, 28(1): 94-97.