

服务实时性测试场景研究与设计

于波

(江苏自动化研究所, 江苏 连云港 222061)

摘要: 在面向服务的信息系统服务实体测试研究的基础上, 针对服务实体的运行由服务管理中心进行统一的调度和资源均衡, 从而导致不同的系统运行场景可能会引起等待时间波动的问题, 以定位引起服务实体响应时间波动的因素为目的, 根据信息系统服务实体的业务访问剖面计算顺序、并行、选择和循环四种调用结构下的服务并发访问频率, 设计四种调用结构中服务响应时间计算模型, 提出基于增加压力的服务实体和基于多业务流程并发执行的服务实体两种访问场景设计技术。结合服务并发访问频率、服务响应时间计算模型给出两种测试场景执行结果分析方法, 同时给出了面向服务的信息系统服务实时性测试流程, 为验证软件是否存在时间波动缺陷提供了方法, 为软件实时性测试提供了依据。

关键词: 面向服务; 时间波动; 服务实时性; 服务响应时间; 测试场景

中图分类号: TP311.52

文献标识码: A

文章编号: 1673-629X(2019)03-0198-07

doi: 10.3969/j.issn.1673-629X.2019.03.042

Research and Design of Real-time Service Test Scenario

YU Bo

(Jiangsu Automation Research Institute, Lianyungang 222061, China)

Abstract: On the basis of the service entity test research of the service oriented information system, the service management center carries out unified scheduling and resource balancing for the operation of service entities, which may lead to the waiting time fluctuation in different system operation scenarios. In view of this problem, for the factors of fluctuations in the response time of service entities caused by locating according to the concurrent access frequency of the service under the four invocation structures of order, parallel, selection and loop in business access profile of the information system service entity, we design the service response time calculation model in four invocation structures and propose two access scenario design techniques: service entity based on increasing pressure and service entity based on concurrent execution of multiple business processes. Combined with the service concurrent access frequency and the service response time calculation model, two methods for analyzing the execution results of test scenarios are presented. Meanwhile, the real-time test process of service-oriented information system services is presented, which provides a method for verifying whether the software has time fluctuation defects and a basis for the real-time test of software.

Key words: SOA; time fluctuation; service real-time; service response time; test scenario

0 引言

面向服务的信息系统有很强的实时性要求, 尤其是实时雷达情报、技术侦察情报处理的信息系统时延都小于秒级, 而通用 SOA 实现技术由于其传输协议、消息协议的松耦合性, 消息协议为字符型协议的特点, 无法满足信息系统对信息处理、分发实时性的要求^[1]。目前, 提高实时性是 SOA 信息系统的研究热点。随着 SOA 信息实时系统构建技术的发展, 给测试技术带来了挑战。在 SOA 信息实时系统测试中, 实时性能测试会变得尤其重要, 传统的性能评估方法在 SOA 信息实

时系统中显然已不再适用^[2]。因此, 需要研究满足信息系统实时性的评估技术, 并研制相应的软件, 形成实时 SOA 运行评估系统, 以实现实时雷达情报、技术侦察情报处理等信息系统时延的准确评估。

1 概述

面向服务的信息系统具有松耦合、高互操作性的特点。在面向服务的信息系统中, 服务提供者对其所提供的服务并发响应能力有限的情况下, 如果请求者的请求数量太多, 超过服务提供者并发响应能力, 应当

收稿日期: 2018-04-27

修回日期: 2018-08-08

网络出版时间: 2018-12-20

基金项目: “十三五”重点基础科研项目(41402020101)

作者简介: 于波(1983-), 女, 硕士, 工程师, 研究方向为软件测试、SOA 测试。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20181219.1542.078.html>

如何调度来满足尽量多的请求。这个问题对于实时面向服务的信息系统特别重要,面向服务的信息系统的关键特点就是要保证实时性,因此请求不能无限期等待下去,必须要进行合理调度^[3]。为了保证系统的实时性,传统指火控系统往往将具有实时性要求的任务分配给高优先级线程进行处理,甚至让系统独占多核处理器中的一个内核,而在面向服务的架构设计中,服务实体的运行由服务管理中心进行统一的调度和资源均衡。不同的系统运行场景给服务请求者的等待时间带来了一定程度的波动和不确定性,因此需要对服务实体的访问频率、并发访问概率、响应时间因素进行评估分析,构造出可能存在时间波动的服务实体访问场景,从而找出引起服务实体响应的波动的原因。

面向服务的信息系统实时性测试主要根据信息系统业务剖面模拟出服务运行时的各种场景。首先,根据系统的业务剖面构造服务实体的被访问剖面,根据服务实体的调用关系提取出所有流程及发生概率;然后,预测访问频率、并发访问概率及响应时间,模拟并

测试出不同访问场景下服务实体的平均响应时间指标;最后,根据不同访问场景下的时间波动趋势,对系统的设计缺陷进行分析。构建服务访问频率、并发访问概率及响应时间的评估方法是 SOA 实时系统性能测试中至关重要的一环,是服务实体被访问场景设计的基础和依据。

2 服务的访问频率与响应时间预测

面向服务的信息系统中对不同的业务访问的频率有所不同,部分业务流程被调用的频率非常高,为了保证能快速、高效地完成任务,对被调用频率较高的业务流程中服务的响应时间要求较高^[4]。此外,业务流程中对服务的调用关系错综复杂,存在一个服务被多个业务流程调用的现象,当多个业务流程同时运行时,就会出现并发访问的现象,为满足系统的实时性,对此类服务响应时间要求很高^[5]。面向服务的信息系统的每个业务流程访问频率及业务流程结构可以从业务剖面中获得,面向服务的信息系统的业务流程调用结构如图1所示。

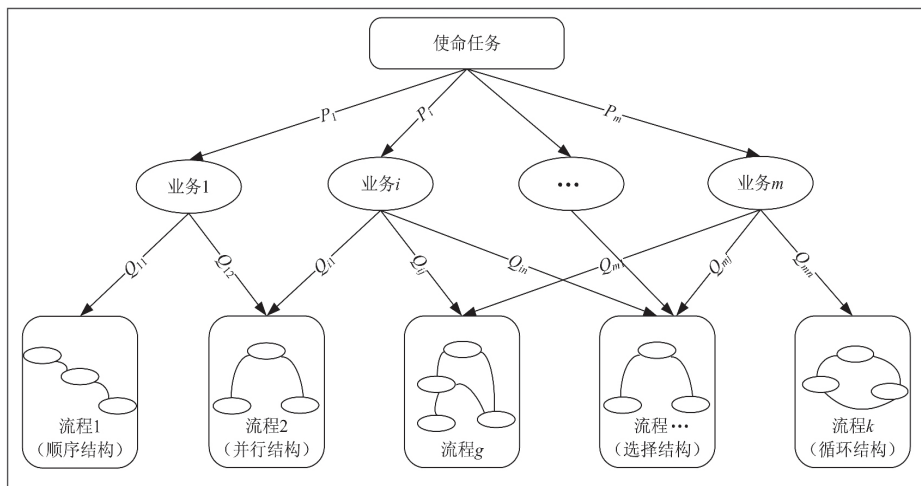


图1 业务流程调用结构

M : 代表系统中的总业务数量;

N : 代表每个业务所调用的总流程数量;

K : 代表系统中的总流程数量;

L : 代表访问每个流程所对应的业务个数;

i : 代表若干个业务之一, $1 \leq i \leq M$;

j : 代表每个业务所需调用的流程之一, $1 \leq j \leq N$;

g : 代表若干个业务流程之一, $1 \leq g \leq K$;

P_i : 代表执行业务 i 的概率;

Q_{ij} : 代表业务 i 调用所有可能调用的流程中第 j 个流程的概率;

$R(nm)$: 代表第 m 个业务调用流程 n 的概率, $1 \leq m \leq L, 1 \leq n \leq K$ 。

因此,在系统接到使命任务后,执行流程 g 的概率

R_g 可表示如下:

$$R_g = \sum_{m=1}^L R(nm) \quad (1)$$

$$R(nm) = P_i \cdot Q_{ij} \quad (2)$$

其中, i 的取值为所有能够调用到流程 g 的业务; j 的取值为流程 g 在业务 i 调用的所有流程中的编号。

2.1 服务访问频率预测

流程编排中服务之间的调用关系存在着顺序、并行、选择和循环四种结构,因此计算在系统接到使命任务后,调用某个服务的概率时,必然要考虑它所处的流程调用结构^[6-7]。

(1) 在顺序、并行和循环调用结构的流程中所有服务的访问概率均为 1,服务在单个业务流程中被访问的频率 P_{sev} 值等于 R_g 。

(2) 选择调用结构的流程中服务的访问具有被选择的概率,因此,访问这一业务流程中所能访问的服务的频率 P_{sev} 等于 $R_g \cdot P_{select}$ 。

并发访问在业务流程调用中是一种常见现象,为完成某一任务,有时会需要调用多个业务流程,多个业务流程同时执行,一个业务流程中的服务可能同时被其他业务流程所访问,从而产生并发访问的现象^[7]。此外,在同一业务流程中也存在同一服务被不同并行分支并发访问的现象。

(1) 当业务流程中存在选择结构,且一个选择分支覆盖并发访问的服务,则并发访问的服务的访问概率 P_{sev} 等于 $\sum R_g \cdot P_{select}$ 。

(2) 当业务流程中的所有选择分支均覆盖当前并发访问的服务,那么并发访问的服务的访问概率 P_{sev} 等于 $\sum R_g$ 。

2.2 服务响应时间预测

在面向服务的信息系统的软件测试中,单个服务响应时间的测试可直接通过端口输入与输出结果的响应来衡量。响应时间往往更关注的是一组服务的响应时间情况^[8-9]。因此,文中从组合服务的角度出发对不同组合结构的组合服务响应时间进行预测分析。由于面向服务的信息系统是一个实时 SOA 系统,服务数量有限,故调度本身占用时间可忽略不计^[9]。

(1) 顺序结构组合服务响应时间 T_{sev} 等于各个服务的执行时间之和。

(2) 并行结构组合服务响应时间 T_{sev} 等于并行分支中各个服务响应时间之和最大的时间值。

(3) 选择结构组合服务响应时间 T_{sev} 等于被选择的分支中服务执行时间最大值。

(4) 循环结构组合服务响应时间 T_{sev} 等于服务执行时间与循环次数之积。

3 服务实体访问场景构建与时间波动性分析

3.1 服务实体访问场景构建技术

面向服务体系架构下,服务实体被调用的不确定性常常会导致服务的响应时间出现不易复现的波动特性^[10]。访问频率较高的“热点”服务主要存在于两种业务活动场景中,一种是仅被一个业务流程调用,因该业务流程在所有业务流程中被访问频率较高,导致该服务被列入“热点”服务之中,辨别此类服务的依据是通过该业务流程中所涉及的所有服务是否均在“热点”服务行列来判定。另一种是被多个业务流程调用,因多个业务流程中都需要执行此服务,导致该服务被列入“热点”服务之中^[11-12]。因此,针对“热点”服务可能引起访问时间波动的场景设计主要从以下两个方

面进行。

(1) 基于增加压力的服务实体访问场景构建。

基于增加压力的服务实体访问场景是针对因业务流程在所有业务流程中被访问频率较高,导致业务流程中被访问的服务被列入“热点”服务之中而构建的测试场景。找出业务流程中所有服务均被列入“热点”服务的分支,开启该业务流程,通过设置启动延迟、触发次数和触发间隔不断增加用户访问压力的方法,测试业务流程与服务响应时间的最大值、最小值以及平均值。

(2) 基于多业务流程并发执行的服务实体访问场景是针对因多个业务流程中都需要执行此服务,导致该服务被列入“热点”服务之中而构建的测试场景。找出存在多业务流程访问的“热点”服务,开启所有访问业务流程,测试业务流程与服务响应时间的最大值、最小值以及平均值。若不存在时间波动性,则继续增加各个业务流程的访问压力,测试出现服务实体访问时间波动的可能性。基于多业务流程并发执行的服务实体访问场景设计如图 2 所示。

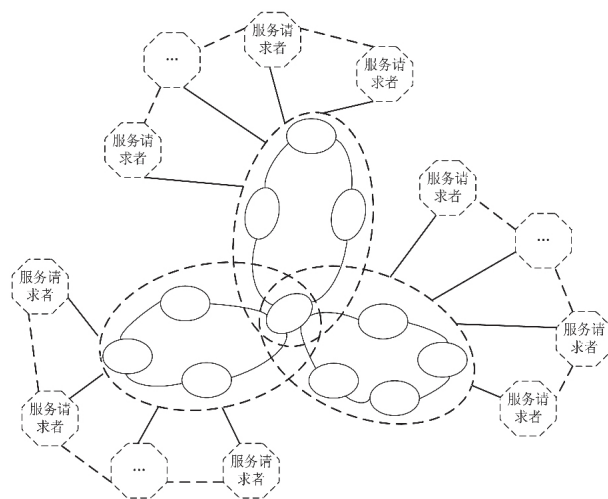


图 2 基于多业务流程并发执行的服务实体访问场景设计图

3.2 服务响应时间波动性分析

服务执行的响应时间受服务本身以及服务所处环境的影响,服务环境主要包括提供服务的服务器主机以及服务器所在的网络两个方面,包括服务器硬件配置、操作系统、网络资源环境配置等。在该研究中,仅从服务本身来剖析时间波动的复杂性。

面向服务的信息系统服务实体访问场景设计为服务实体访问的时间波动性分析提供了测试手段,依据不同的访问场景执行测试用例后,可得到相应业务流程响应时间。对于访问场景中业务流程的响应时间值可以有两方面的参照标准,一方面是性能指标要求值,另一方面是根据业务流程的结构推算出的响应时间预

测值,通过测试所得响应时间值、指标要求值和预测响应时间值可生成各业务流程的响应时间图形,为测试人员提供数据分析依据。测试人员通过对图形的分析比对,找出存在时间波动的业务流程。

在面向服务的信息系统业务流程场景中,融合很多业务流程与服务组合,可根据业务场景的设计有针对性地画出相应的业务流程。随着操作的不同,数据值也会随之改变,可通过脚本录制及回放技术进行不同业务流程、服务组合的清晰显示和应用分析。当测得的响应时间不满足性能指标要求时,首先需要对性能指标要求值的合理性进行分析,即通过性能指标要求值和响应时间预测值的对比,找出存在不合理性的指标要求。

性能指标要求合理且时间波动性较大的业务流程,需要通过插桩的手段测出业务流程中各服务的响应时间,形成单个服务响应时间对比图,从对比图中可以找出影响业务流程响应时间的服务,对时间波动性较大的服务进行分析和度量,验证“热点”服务引起服务实体访问时间波动的可能性。

4 面向服务的信息系统服务实时性测试流程

服务实体访问场景构建是为了找出引起服务访问时间波动的因素而构建的业务访问场景,这些业务场景的执行需建立在单个服务响应时间和每个业务流程响应时间均符合性能指标要求的基础之上。因此,在进行面向服务的信息系统服务实时性测试时,首先,要针对单个服务的响应时间设计测试用例,测得单个服务的响应时间,将之与指标要求进行比较分析,确保单个服务的响应时间满足指标要求。其次,针对每个业务流程设计相应的测试用例,通过执行相应的测试用例测得单个业务流程的响应时间,业务流程的响应时间可依据其结构进行预测,最终通过测试值、指标值、预测值三者的比较进行判定,当测试值与指标要求值不相符时,可参考预测值对指标进行评价,验证指标要求的合理性,在确保指标要求合理的情况下,业务流程响应时间的测试值要满足指标要求。最后,根据针对服务实体访问设计的测试场景进行“热点”服务的压力与负载测试,分析响应时间对比图,查看是否存在时间波动的现象,分析引起时间波动的原因,验证因“热点”服务引起服务实体访问产生时间波动的可能性,为开发方指出会引起时间波动的服务实体。

5 面向时间波动的服务实时性测试实例验证

在实际应用中,选用某舰船火控系统作为应用对象,某舰船火控系统是一种基于面向服务思想而设计的舰船火力控制系统软件,主要由战斗、训练、检

查三种工作模式组成。战斗任务主要通过报文解析服务、通道组织服务、指挥决策服务、目标处理服务、射击服务、火控解算服务来完成;训练任务主要由武器系统训练服务、作战系统训练服务、射击服务、火控解算服务来完成;检查任务主要由设备题检服务、系统题检服务、零飞检查服务、火控解算服务来完成。面向服务的舰船火控系统的服务实体访问剖面如图3所示。

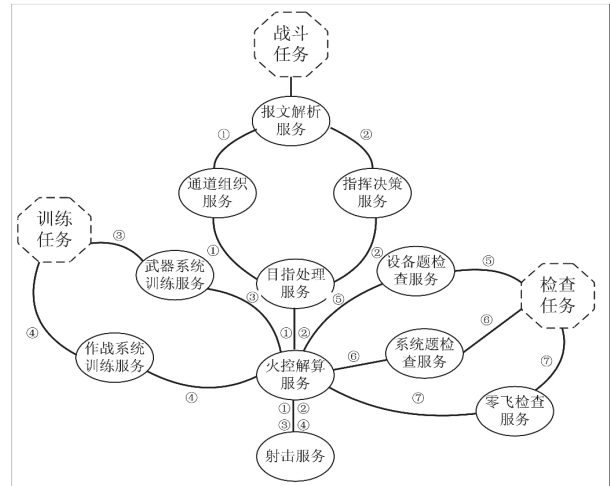
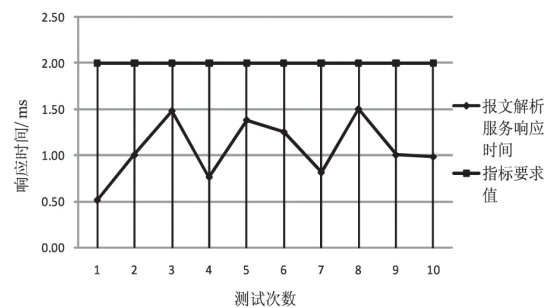
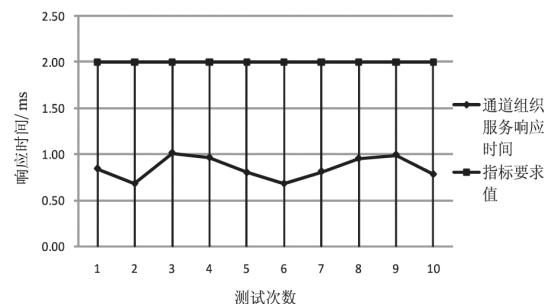


图3 某舰船火控系统服务实体访问剖面

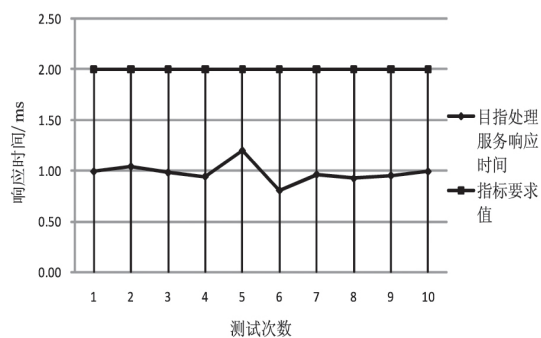
利用SOATest工具对报文解析、通道组织、指挥决策、目标处理、武器系统训练、作战系统训练、设备题检、系统题检、零飞检查、射击服务的响应时间进行测试,并与指标进行对比分析,某舰船火控系统要求火控解算服务的响应时间不大于10ms,其他服务的响应时间不大于2ms;在测试中,对每个服务进行10组输入、输出数据测试,利用SOATest工具设置启动延迟为50ms、触发时长500ms、触发间隔50ms,对应的测试结果分析如图4所示。



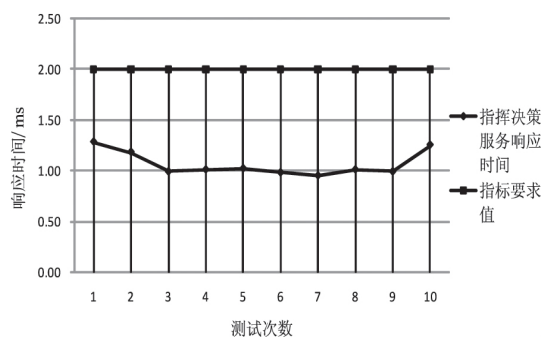
(a) 报文解析服务响应时间



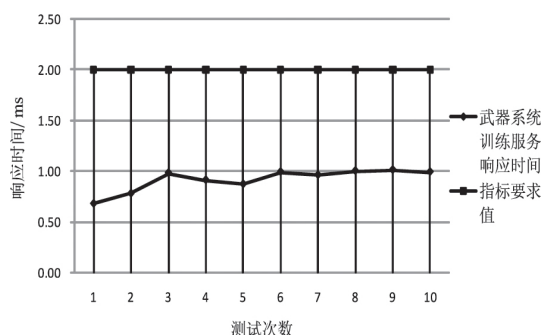
(b) 通道组织服务响应时间



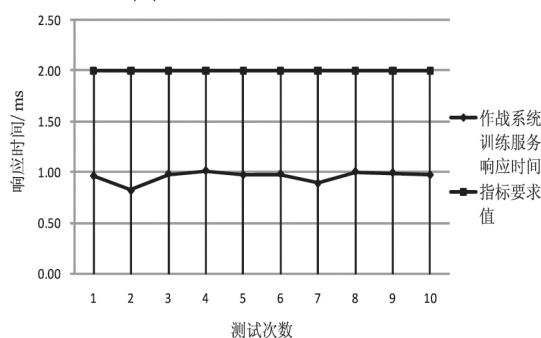
(c) 目标处理服务响应时间



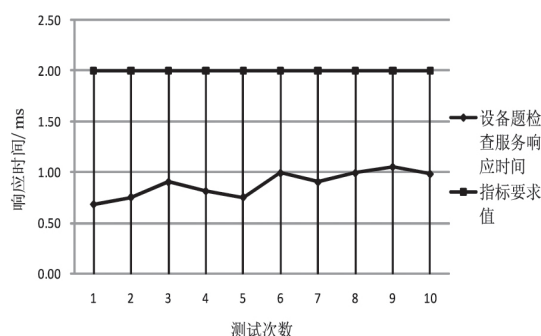
(d) 指挥决策服务响应时间



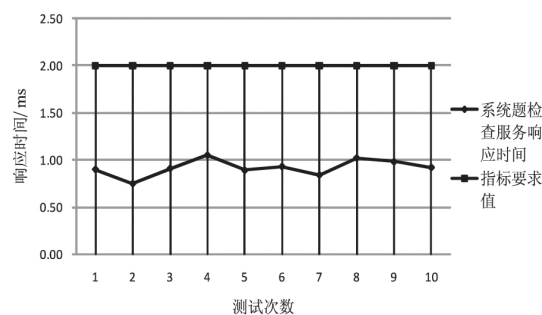
(e) 武器系统训练服务响应时间



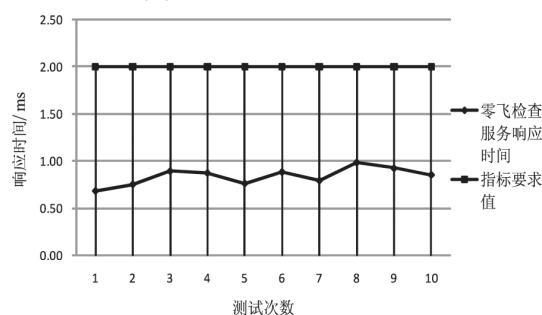
(f) 作战系统训练服务响应时间



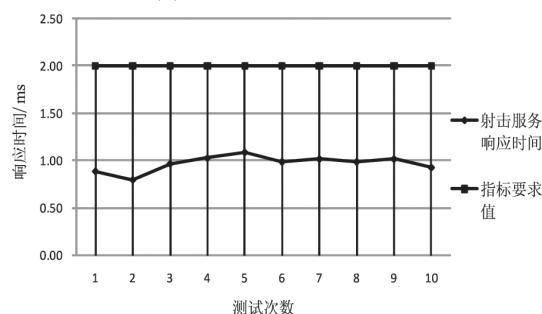
(g) 设备题检查服务响应时间



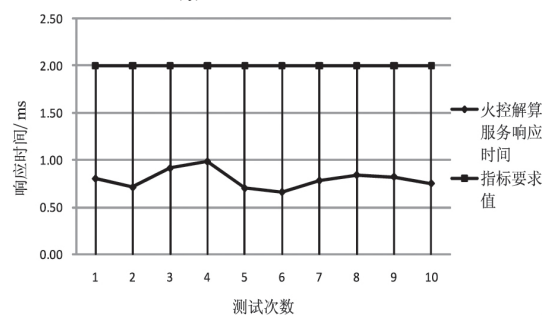
(h) 系统题检查服务响应时间



(i) 零飞检查服务响应时间



(j) 射击服务响应时间

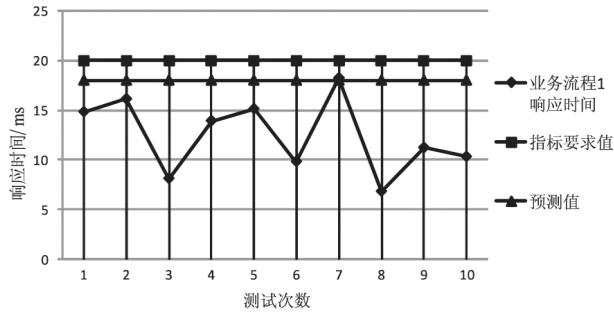


(k) 火控解算服务响应时间

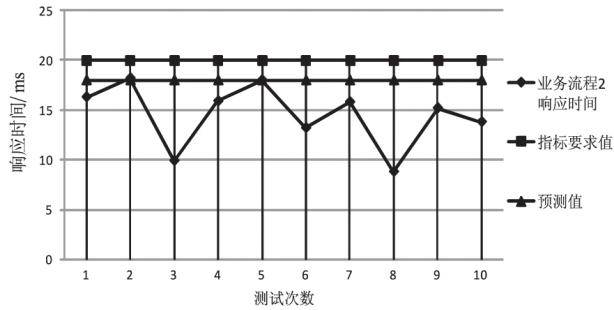
图 4 某舰舰船火控系统单个服务响应时间测试结果

其次,对战斗任务、训练任务、检查任务中的每个业务流程的响应时间进行测试;因每个业务流程的服务时间都是顺序关系,利用顺序结构组合服务响应时间公式计算可得每个业务流程的响应时间应不大于服务个数与单个服务响应时间指标要求值之积;业务流程 1~7 的响应时间预测值分别为 18 ms、18 ms、14 ms、14 ms、12 ms、12 ms、12 ms,指标要求值均为 20 ms,指标要求值大于预测值,从而推断指标要求是合理的;在测试中,针对每个业务流程分别进行 10 次调用测试,利用 SOATest 工具设置启动延迟为 50 ms、触发时长 500 ms、触发间隔 50 ms,对应的测试结果分析

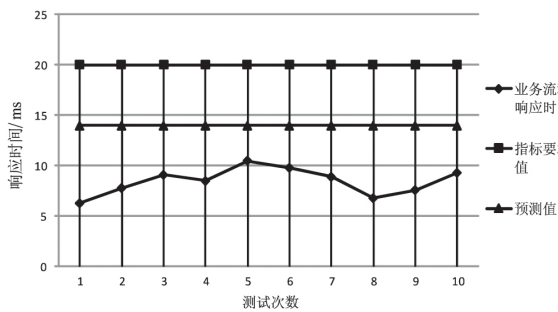
如图5所示。



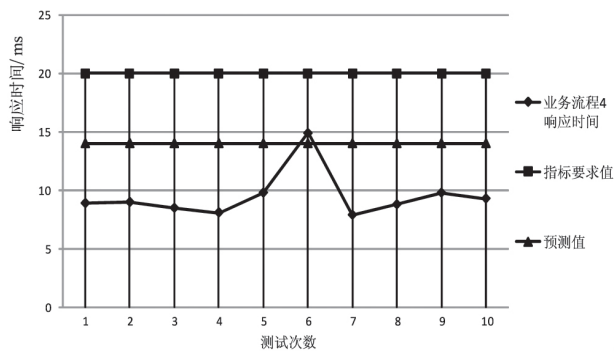
(a) 业务流程1 响应时间



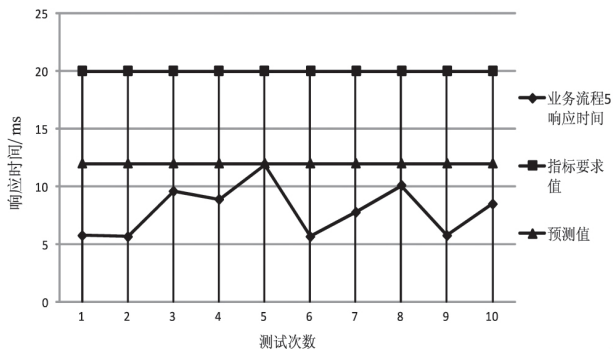
(b) 业务流程2 响应时间



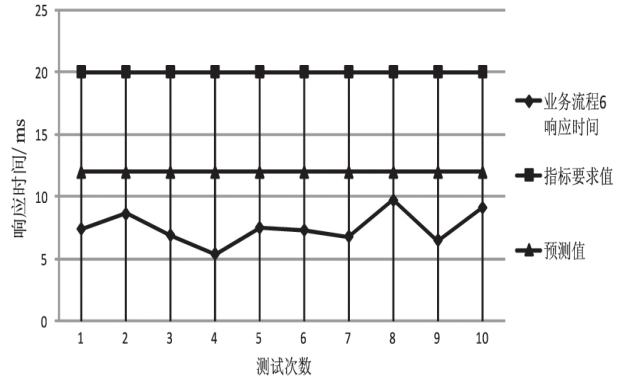
(c) 业务流程3 响应时间



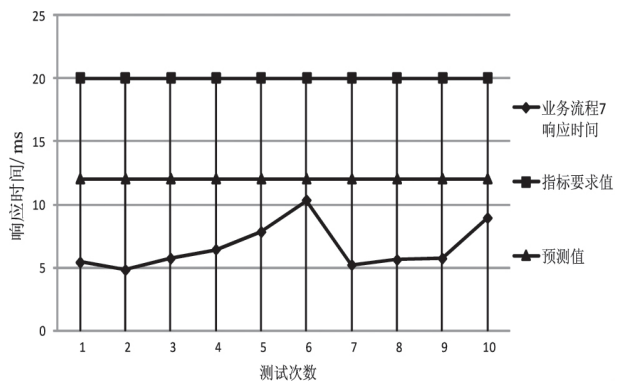
(d) 业务流程4 响应时间



(e) 业务流程5 响应时间



(f) 业务流程6 响应时间



(g) 业务流程7 响应时间

图5 某舰舰船火控系统业务流程响应时间测试结果

再次根据某舰舰船火控系统业务流程应用频率及剖面结构分析确定利用基于多业务流程并发执行的服务实体访问场景对火控解算服务进行场景测试,其服务实体访问场景如图6所示。

根据上述场景执行,开启业务流程1~7,并同时增加放大倍数为5倍,启动延迟为50 ms、触发时长500 ms、触发间隔50 ms,查看每个业务流程的响应情况。测试结果中,发现业务流程2存在时间波动现象,利用插桩技术获得业务流程2中各个服务的响应时间可以看出,服务4的服务响应时间最长,业务流程2中的第四个服务为火控解算服务,由此可得出火控解算这一热点服务会在多用户并发访问的情况下引起业务流程执行缓慢导致的时间波动现象。

6 结束语

文中研究适用于基于SOA的服务实时性测试,服务实体访问场景的构建是基于服务访问频率计算、组合服务响应时间预测推算。

通过组合服务响应时间的计算模型可以得出具体业务流程响应时间的预测值,该值不仅为服务实体访问的时间波动分析提供了依据,同时也为业务流程响应时间的指标要求提供了评判标准。通过服务实体访问场景的构建与时间波动分析的研究,提高了服务性能测试的效率,实现了对服务模型结构的综合评估,指

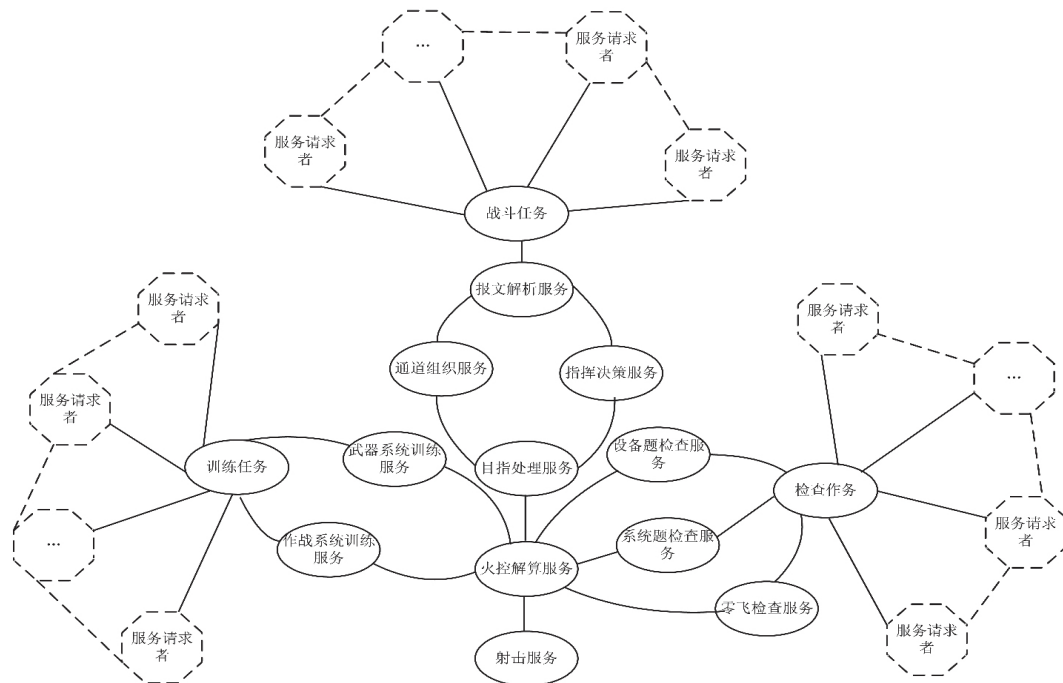


图6 某舰舰船火控系统性能测试服务实体访问场景图

出服务模型构建存在的不足、指标要求的不合理之处，为后续的开发工作提供参考。

参考文献:

- [1] 王 晓, 聂 凯. 基于 SOA 和 MDA 的指挥信息系统构建新方法[J]. 舰船电子工程 2014, 34(6): 56-59.
- [2] 陈颖慧, 邱雪松, 刘益畅, 等. 基于模型的 Web Service 性能测试方法[J]. 北京邮电大学学报 2009, 32: 44-48.
- [3] 王 伟. 武器装备系统测试过程建模与管理系统研究[D]. 北京: 北京交通大学 2012.
- [4] LI Haifeng. A queue theory based response time model for web services chain [C]//2010 international conference on computational intelligence and software engineering. Wuhan, China: IEEE 2010: 1-4.
- [5] 李 乔, 秦 锋, 郑 啸. Web 服务响应时间测试[J]. 计算机工程与设计 2007, 28(19): 4670-4673.
- [6] 谢春丽, 李必信, 王喜凤, 等. Web 服务可靠性的阶段模型[J]. 东南大学学报: 自然科学版 2012, 42(1): 40-44.
- [7] 陆 文, 徐 锋, 吕 建. 一种开放环境下的软件可靠性评估方法[J]. 计算机学报 2010, 33(3): 452-462.
- [8] GOMATHY C K, RAJALAKSHMI S. A software quality metric performance of professional management in service oriented architecture [C]//2nd international conference on current trends in engineering and technology. Coimbatore, India: IEEE 2014: 130-138.
- [9] 姜皇勤. 面向 SOA 多域环境的 Web 服务访问控制技术研究[D]. 郑州: 解放军信息工程大学 2013.
- [10] 顾 军, 罗军舟, 曹玖新, 等. 考虑失效恢复的组合服务性能建模与分析[J]. 软件学报 2013, 24(4): 696-714.
- [11] 明 仲, 尹剑飞, 杨 巍, 等. 一种 Web 系统性能测试框架及其混合建模过程[J]. 计算机研究与发展 2010, 47(7): 1192-1200.
- [12] LEE B C, BROOKS D M, DE SUPINSKI B R, et al. Methods of inference and learning for performance modeling of parallel applications [C]//Proceedings of the 12th ACM SIGPLAN symposium on principles and practice of parallel programming. San Jose, California, USA: ACM 2007: 249-258.