

基于 RRT 技术的应用软件集成测试方法

杨 洋,杨庆婷,彭 培

(重庆信息通信研究院,重庆 401336)

摘要: 为提高传统测试方法中识别错误的准确性与效率,针对该问题研究基于 RRT 技术的应用软件集成测试方法。利用软件数据,测试用例距离,生成测试的用例并执行,实现应用软件集成测试方法。实验结果:基于 RRT 技术的应用软件集成测试方法,对错误的识别更具有准确性并且所需要的时间也更短。

关键词: RRT 技术;应用软件;测试;集成

中图分类号:TP311

文献标识码:A

文章编号:2096-9759(2023)06-0063-03

Application Software Integration Test Method Based on RRT Technology

YANG Yang, YANG Qingting, SPENG Pei

(Chongqing Information and Communication Research Institute, Chongqing 401336)

Abstract: In order to improve the accuracy and efficiency of identifying errors in traditional testing methods, an application software integration testing method based on RRT technology was studied to address this issue. Using software data and test case distance, generate and execute test cases to achieve application software integration testing methods. Experimental results: The application software integration testing method based on RRT technology is more accurate in identifying errors and requires less time.

Keywords: RRT technology; Application software; Testing; integrate

0 引言

在实际的环境中,面向对象的测试会存在一定问题,例如,比较常用的一个方式便是随机检测,即在一个系统域随机选择测试用例,然后加以使用。虽然这个技术有着不错的测试性能,

不会忽略所有的程序问题,不过它对进行试验中的测试方法要求较多,所以技术可行性较小。为解决上述问题,本项目提出一种新型的面向对象的软件测试方法,即限制域自适应随机测试算法(RRT)。它是在每一个被执行的真实需求的周边输

收稿日期:2023-03-29

作者简介:杨洋(1994-),男,重庆人,学士,软件测评工程师,研究方向:软件测评领域技能技术与自动化。

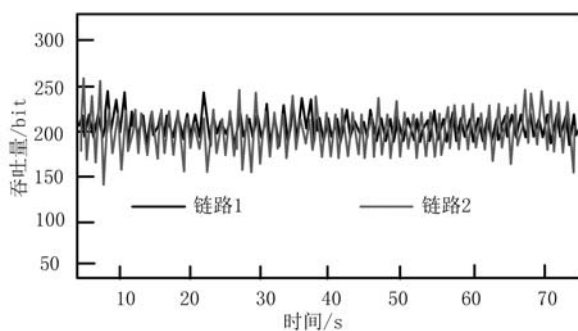


图 2 网络通信双链路热备仿真结果

根据上述图 2 所示的仿真结果,可清晰看出双链路在网络通信过程中对数据信息的吞吐速率,同时,终端所呈现的仿真结果所展示的链路 1 与链路 2 通信传输过程完整、连续,可较为直观地展示当前状态下网络通信双链路热备状态。由此可说明本文此次设计的仿真方法,可实现对双链路服务器通信的仿真,以此方式,可为企业数据信息实时通信、资源高速安全共享提供可靠支撑。

5 结语

网络通信是数据传输、资源共享的基础,为满足用户对信息的需求,通常会采用将互联网作为支撑的方式,进行 IP 数据的线上传输。为提高网络通信传输的可靠性,本文引进 cNSP 平台,通过建立拓扑仿真实验环境、建立双链路通信组网模型、网络通信双链路热备通信认证、双链路服务器通信仿真,设计了针对网络通信双链路的热备仿真实验。完成设计后,以某地区中小型企

业为例,按照本文提出的方法,进行通信双链路热备仿真。仿真结果证明了本文设计的成果可实现双链路服务器通信状态的感知,以此方式,可为市场内各类企业的运营管理、数据安全通信与共享传输提供技术层面的指导与帮助。尽管本次设计的成果在实际应用中的效果良好,但本次实验也仅从仿真成果展示层面进行了通信双链路热备效果的分析,并未从更多的角度对设计成果进行检验。因此,有必要在后续的设计工作中,加大对本次设计的投入,设置更多的指标,对与之方面相关的内容进行测试,旨在通过此次设计,为优化通信双链路热备性能给予支持。

参考文献:

- [1] 黄晓梦,黄永刚,王立群,等. 基于智能传感网络的医疗服务数据采集与管控分析方法研究[J]. 电子设计工程,2023,31(02):29-33.
- [2] 杨瑞,周杰. 基于混沌克隆遗传算法的无线传感器网络低能耗分簇方法[J]. 石河子大学学报(自然科学版),2022,40(05):655-660.
- [3] 赵秋雨,武红玉. 基于 SCD 的继电保护网络通信路径自动规划技术研究及应用[J]. 许昌学院学报,2022,41(05):117-122.
- [4] 胡亨汶,孟祥印,李丹,等. 基于 RESTful Web Services 的云边数据交换设计与实现[J]. 现代制造工程,2022,(08):25-32.
- [5] 雷淑,张孜江. 基于 LoRa 与 NB-IoT 技术的文物保存环境监测系统探索及思考——以四川博物院为例[J]. 文物保护与考古科学,2022,34(04):114-122.
- [6] 马琳. 新时期临建施工中网络通信监控设备的运用——以恒大雅苑工程项目为例[J]. 工程技术研究,2022,7(15):104-106.
- [7] 赵巍,刘宪,李凌,等. 基于 5G 移动通信仿真平台的端到端双栈组网部署方法研究[J]. 长江信息通信,2022,35(04):182-184.

入区内生成一个排除区,并将这个随机生成的真实需求反复进行,直至这个应用范围超出了排除区,这表示这个测试已经结束,接下来就可以进行下一轮的测试了。采用 RRT 方法进行软件测试,能显著减少测试所需用例,提高测试的经济效果。为此,以可信度和节点为主要信息生成的因素,通过综合定义了网络拓扑的特征权图、权值矩阵、数据可信判定指数、可信度等有关要素,本文提出基于 RRT 技术的应用软件集成测试方法。

1 基于 RRT 技术的应用软件集成测试方法

最近几年,愈多的软件使用面向集成测试的开发技术,本文的方法利用结构驱动的整体测试方式来完成对应用程序的测试。通过与已有的试验结果进行比较,结果显示,该试验方法切实可行,试验速度、试验精度均优于已有的试验方法^[2]。

1.1 选取软件数据

RRT 算法属于一种以排斥区域为基础的 ART 算法,设置一个定义集和引用集的定义公式如下:

$$\begin{aligned} Def(s) &= \{o\} \\ Ref(s) &= \{p\} \\ v &\in def(o) \cap ref(p) \end{aligned} \quad (1)$$

公式中:变量 v 满足 $v \in def(o) \cap ref(p)$, 节点 o 和节点 p 的路径上的其他节点表示为 $o' \in a\{o, p\}$, 当 $v \notin def(o')$ 时,其他节点为 $o' \in a\{o, p\}$ 当 o 是 p 的后点,但 p 不是 o 的后点时, o 正相关于 p 。

在选择软件数据的过程中,选择的节点和可信度被用作数据产生的指标,在构建可靠数据时,对有关因子进行了以下的计算:

$$\begin{aligned} G &= (V, E) \\ w_{(o,p)} &= \{LW/W_r, LW/W_r \geq 1 \end{aligned} \quad (2)$$

公式中: E 设定为链路, V 表示一个节点,则 E 的权值 w 表示 o 到 p 之间的链路长度。

程序中虚拟的节点表示为 $NS = \{VN, VL, PN, PL\}$, $VN = \{vn_1, vn_2, \dots, vn_m\}$ 。实际的节点表示为 $PN = \{pn_1, pn_2, \dots, pn_n\}$ 。虚拟的链路表示为 $VL = \{vl_1, vl_2, \dots, vl_n\}$ 。实际链路的反映集合表示为 $PL = \{pl_1, pl_2, \dots, pl_n\}$ 。

设定一个可信判定函数为 $judge - ns - pre(x, y, t)$ 计算公式如下:

$$judge(x, y, t)^M = \begin{cases} 1, NS - PRET \geq T \\ 0, NS - PRET < T \end{cases} \quad (3)$$

公式中: x 是一组要判断的资源, y 是一组资源的可信性, t 是一组数据的信任要求。

其算法思路为:首先,在输入域中随机选择 1 个测试用例,在有 n ($n \geq 1$) 个测试用例的情况下, RRT 将在每一个测试用例的周围定义 1 个排斥区。所有被排斥的面积都是 $R \cdot d \cdot n$, 这里 d 是输入字段的尺寸。 R 是一个已被排除的标度,也就是被测试者设置的标度。选择一个可执行的函数,用来定义一个程序中的 P , 一个变量,这个变量包括了所有可能对变量产生作用的语句。将 RRT 技术应用到软件测试中^[3], 要注重程序中特征路径的兴趣点,从而提高软件测试的效率,选取数据的基本流程:(1)用黑盒子试验来判断输出的结果,并且检验有无出现差错,找出差错的位置,判断差错的影响范围,来排除差错。(2)从多个层面上进行依存性的研究。(3)采用分块(sheet)

算法,对错误的来源进行定位。(4)执行回归测试的排除与修理步骤,直至没有任何错误再次出现。(5)负责模块的自动化数据选择。测试数据生成的框架如下图 1 所示。

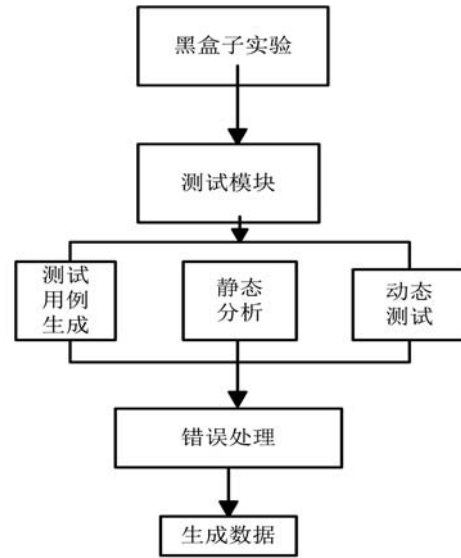


图 1 数据生成框架

1.2 测试用例距离

在 OOP 环境下使用 RRT 算法,其关键是要确定测试用例间的距离。本文通过对国内外相关文献的全面分析,提出了一套测量实验用例距的方法^[4]。当对一个面向对象的程序进行测试时,如果把一个测试用例作为一个对象,那么这个测试用例就包含了属性和方法,在这个两个属性中,一个是静态的,另外一个动态的, RRT 会围绕每个已执行测试用例来定义 1 个排斥区域。假定两个距离分别是 $p=a+bi$ 和 $q=m+ni$, 其中 a, m 代表的是主距离, b, n 代表的是辅助距离,那么 p, q 的四则运算操作是指相应的主要和次要的距离。当要确定两个距离的关系时,分别对主次距离进行判断,对于 2 个距离 p, q , 只有在 $a=m, b=n$ 时,才得出 $q=o$ 。只有在 $a>m$ 的情况下,或 $a=m$ 而 $b>n$ 的情况下, $q<p$ 才会产生;只有在 $a<m$ 的情况下,或 $a=m$ 而 $b<n$ 的情况下, $q>p$ 才存在。

其中,测试用例的间距分为两个部分^[5]:一个是静态的,一个是动态的。静态区段包含自订特性、继承特性和巢状特性。动态模块包含了定制的方法和继承的方法,为便于计算,还将继承的方法与定制的方法相结合。2 个测试用例可以用 p, q 来表达,用 $p.s$ 和 $q.s$ 来表达,用 $p.s.d$ 和 $q.s.d$ 来表达 2 个测试用例的元素,用 $q.s.d.type$ 和 $p.s.d.type$ 来表达测试用例的类型;使用 $q.s.e$ 和 $p.s.e$ 来代表 2 个测试中的嵌套属性;使用 $q.b$ 和 $p.b$ 来描述 2 个测试案例中的一个动态的组成部分。

静态距离加上动态的距离就可以表示两个测试用例之间的距离,计算公式如下:

$$OD(p, q) = SD(p.s, q.s) + BD(p.b, q.b) \quad (4)$$

公式中: OD 为两个用例之间的距离(objectdistance), SD 为静态距离(staticdistance)表示为自定义属性距离加嵌套属性距离, BD 为动态距离(behaviordistance)。distance 函数是计算 2 个属性数值差的绝对值并除以该属性的范围,该函数计算公式如下:

$$\sum_{i=1}^m (D(p.s.d[i], q.s.d[i])) = ED(p.s.d, q.s.d) \quad (5)$$

公式中: ED 为元素距离(elementarydistance)。其中,一些

静态距离是指由对象集的距离所构成的嵌套属性距离。首先,计算两个对象的自定义属性集中成员变量数目的差别,在此取绝对值;接着,计算两个对象的自定义属性集中成员变量的相似性和相异性,在此也取绝对值;最终,将两者相加,便可得出两个对象的类型距离。

1.3 生成测试用例

在使用 RRT 算法应用在软件集成测试程序时,首先要输入待测类的具体信息。实验所采用的原型系统包含一个类图信息输入模块,在此模块中实现了类的建立,再基于类图的构建,利用类的构建方法建立了一个目标,并通过一个随机数产生算法给目标中的每一个成员分配了一个数值。应用软件集成测试的重点是模块之间的界面以及整合后的测试模型。在进行测试时,先通过以下的公式,测量了软件的动态复杂性。

$$f = NS \frac{\delta y(x, y)}{\delta m} + \rho \frac{\delta y(x, y)}{\delta n} \quad (6)$$

公式中: δ 代表了一个软件部件的失败几率; f 代表了一个构件的复杂性之和。用动态复杂性来度量各模块间的动态耦合性,得出了如下公式:

$$k = \frac{q(I_a + I_r)}{f} \quad (7)$$

公式中: q 代表软件部件被利用的频率; I_a 代表频率更高的部件的使用; I_r 代表一个频率更少的组件。该方法通过对各构件的分层和分级,以最常用的构件为起始,对已有的构件进行了综合测试,并对已有的构件进行了排除。

在应用软件集成测试过程中,以函数型函数为核心,实现了构件的集成。由于软件中各组件的品质及开发周期存在较大的差别,使得集成测试成为一项长期、循序渐进的工作。在此基础上,提出了一种基于分层的集成测试方法。在检测的过程中,不但要人为,而且还要有计算机监控。而在此过程中,工程师还必须对测试过程中所发生的问题种类及其规模加以研究,从而总结出问题的根源和原因,以便于进一步提高应用软件中集成检测结论的正确性^[6]。以上方法使得 RRT 技术能够用于对相同的类进行测试。在对面向对象的软件进行测试时,为了生成测试用例,需要依赖于被测对象的类图信息。

2 实验分析

实验通过识别软件运用的实际过程中发生的错误以及修正问题为基础,验证基于 RRT 技术的应用软件集成测试方法的性能。

实验的环境为 32 位 windows7 操作系统,双核 3.0GHZ,8 G 的存储空间。该系统的基础数据为 124M 字节,在系统更新时,需要对系统进行更新和增加的功能,大约有 10%左右。测试的基本过程是:(1)首先,在 OOP 数据输入模块中,将被测类的具体数据输入到 OOP 中;(2)在每个测试时,在 XML 文件中存储和设置被测试类别的有关信息,这样就可以读取这个类别的设置参数;(3)生成和实施测试案例;从被测试类别的组态参数建立物件,然后将物件的基型别成员指派给物件;4 对测试用例的实施情况进行记录;(5)采用 RRT 与 RT 两种方法进行了测试案例的错误识别;(6)选择两种算法中的测试用例,并将其应用到该软件中。

2.1 错误识别数量实验

实验中对软件中两个模拟场景进行测试,模拟场景一和模拟场景二分别包含 5 个公共方法;两个仿真情景均含有 5 个

资料形态,仿真情景 1 有 3 个被植入,仿真情景 2 有 5 个被植入。下表 1 显示了两种仿真方案的试验结果。

表 1 实验结果

场景名称	RRT	RT
模拟场景 1	3	2
模拟场景 2	5	3

由表 1 可见,与 RT 算法相比,RRT 算法在识别软件程序的错误时,能够识别程序中所有错误。

2.2 错误识别时间实验

实验中在软件中识别错误的速度也是评判集成测试方法的重要标准之一。下图 2 为在相同错误数量情况下两种算法识别错误所需时间的情况。

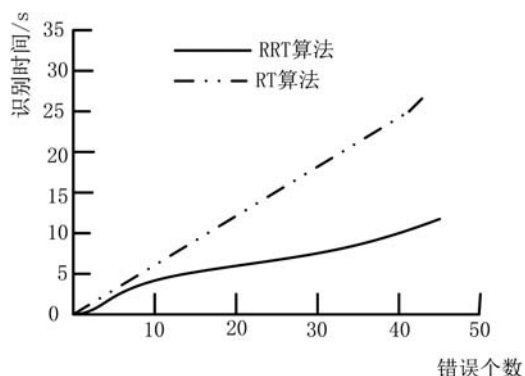


图 2 识别错误时间结果

由图 2 可以看出在识别相同错误数量的同时,RRT 所需要的时间远远小于 RT 算法。

综上所述,基于 RRT 技术的应用软件集成测试方法,识别错误更加具有准确性并且所需要的时间也更短。

3 结语

计算机软件的检测技术,在当前已有了一定的发展,但在其应用中仍需进行一些创新。从目前的面向应用软件集成测试方法开始,对 RRT 算法的有关概念进行了介绍,并逐渐地提出了其在软件集成测试中的应用步骤,从而深化了对相关概念的认识。随着计算机技术的发展,要使软件测试工作更加高效,就必须跟上时代步伐。

参考文献:

- [1] 王全. 基于 RRT 的全局路径规划方法及其应用研究[D]. 国防科学技术大学,2014.
- [2] 黄少芬. 以 RRT 技术为例的面向对象软件测试方法探究[J]. 信息与电脑(理论版),2019(07):102-103.
- [3] 陈锦富,奚家祥,于敏杰等. 一种基于 RRT 技术的面向对象软件测试方法[J]. 江苏大学学报(自然科学版),2019,40(01): 67-75.
- [4] 史传倩. 计算机软件测试技术与深度开发应用研究[J]. 信息与电脑(理论版),2022,34(14):44-46.
- [5] 叶勇健,谭超. 移动互联技术在应用软件安全协同测试中的研究[J]. 沈阳工程学院学报(自然科学版),2021,17(01): 70-75+79.
- [6] 付佳,李宝安. Web 应用软件测试方法研究[J]. 计算机产品与流通,2019(04):22.