

基于故障注入的应用软件可靠性测试

姜文,刘立康

(西安电子科技大学 通信工程学院 陕西 西安 710071)

摘要: 随着计算机技术的发展,云计算技术在各行各业的应用普及,基于云计算的应用软件的种类也越来越多。基于云环境的软件可靠性测试有许多问题和技术需要研究和探讨。在可靠性测试技术中,故障注入技术应用十分广泛。结合一个云化通讯软件实例叙述了采用故障注入技术在云环境开展可靠性测试工作的全过程。叙述了可靠性测试过程以及各个角色在测试过程中的职责和任务;叙述了云环境的层级结构、私有云环境的部署和软件产品的安装;叙述了云环境中故障注入方法和可靠性测试环境的构建;详细叙述了该软件实例的测试用例设计、可靠性测试流程和编写自动化测试脚本;最后给出了测试结果分析。工作实践表明,做好软件可靠性测试工作可以提高软件产品的质量,提高上线后软件产品的可靠性,从而更好地满足客户对软件产品的需求。

关键词: 可靠性;故障注入;可靠性测试;云环境;容错能力

中图分类号: TP311.5

文献标识码: A

文章编号: 1673-629X(2019)02-0023-06

doi: 10.3969/j.issn.1673-629X.2019.02.005

Reliability Test of Application Software Based on Fault Injection

JIANG Wen, LIU Li-kang

(School of Telecommunication Engineering, Xidian University, Xi'an 710071, China)

Abstract: With the development of computer technology, cloud computing technology is widely used in all walks of life, and there are more and more kinds of cloud-based application software. Software reliability testing based on cloud environment has many problems and technologies needed to be studied and discussed. Fault injection technology is widely used in reliability testing. The whole process of reliability test in cloud environment using fault injection technology is described with an example of cloud-based communication software. We describe the process of reliability testing and the duties and tasks of characters in the testing process; describe the hierarchy of the cloud environment, the deployment of private cloud and the installation of software products; describe the fault injection method in cloud environment and the reliability test environment construction; describe the software instance of test case design, reliability test process and writing automation test scripts in detail. The test analysis is also given at last. Practice shows that doing well in software reliability test can improve the quality of software products and the reliability of the software product after launch, better meeting customer demand for software products.

Key words: reliability; fault injection; reliability testing; cloud environment; fault tolerance

0 引言

随着计算机技术的发展,云计算技术在各行各业的应用普及,基于云计算的应用软件的种类也越来越多。软件产品在使用过程中出现软件失效,将导致一系列严重的事故与后果。软件测试阶段不仅需要进行软件的功能、性能测试,还需要对软件进行可靠性测试,以确保新开发的软件上线后能够正常运行。

在可靠性测试技术中,故障注入技术应用十分广

泛。与传统可靠性评测技术相比,它具有无需建立和求解复杂的系统模型、实验时间短、结果精度高等优点,已引起众多学者和研究人员的重视。目前对故障注入技术应用非常成功的领域之一是容错系统的可靠性验证。

文中介绍了可靠性测试的基本概念;叙述了可靠性测试过程以及各个角色在测试过程中的职责和任务;结合工作实践叙述了一个云化通讯软件实例采用

收稿日期: 2018-03-18

修回日期: 2018-07-24

网络出版时间: 2018-11-15

基金项目: 国家部委基础科研计划: 国防预研基金项目(A1120110007)

作者简介: 姜文(1986-)女,工程师,硕士,CCF会员(E200032324M),研究方向为图像处理、数据库应用和软件工程;刘立康,副教授,研究方向为数字通信、图像处理和软件工程。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20181115.1048.046.html>

故障注入技术在云环境开展可靠性测试工作的全过程;最后给出了测试结果分析。

1 软件可靠性基本概念

软件可靠性是指软件在所规定的环境条件下和规定的时间内正确地完成任务的能力。

1.1 软件可靠性的定义

1983 年美国 IEEE 计算机学会对软件可靠性^[1]给出如下的定义:

(1) 在规定的条件下,在规定的时间内,软件不引起系统失效的概率。该概率是系统输入和系统使用的函数,也是软件中存在的缺陷的函数。

(2) 在规定的周期内,在所述条件下程序执行所要求的功能的能力。

经美国标准化研究所批准,该定义成为美国的国家标准。1989 年中国国标 GB/T-11457 采用了这个定义。

1.2 软件可靠性测试

软件可靠性测试包括:测试数据、测试环境的准备、测试运行、可靠性数据收集、可靠性数据分析和失效纠正。目的是为了发现软件运行中出现的错误,实现软件可靠性增长;验证软件的可靠性要求,定量评估软件的可靠度。

1.2.1 可靠性测试方法

可靠性测试方法主要有两种类型:

(1) 基于操作剖面开展测试。

按照软件操作剖面^[2]对软件进行随机测试的测试方法。根据软件在客户实际使用过程的各种操作的使用概率选择操作剖面,对软件系统进行可靠性测试。

(2) 基于故障注入开展测试。

故障注入是按照选定的故障模型,采用人工的方法将故障注入到特定的目标系统中,同时采集系统对所注入故障的反应信息,通过这些信息对系统进行可靠性分析。

1.2.2 软件故障注入技术

与其他可靠性评价技术相比,故障注入方法能方便灵活、便捷有效地处理各种可靠性问题,得到了越来越多的关注。故障注入技术^[3-6]有模拟实现的故障注入、硬件实现的故障注入和软件实现的故障注入三种。

软件故障注入是通过修改硬件或软件的状态变量或相关数据来模拟故障的产生,加速系统的失效,分为静态注入和动态注入两种类型^[7-11]。

静态故障注入主要通过程序变异的方法,通过改变原程序,使被测系统文件静态的存在错误,从而使其运行时出现故障。静态注入占用很少的系统资源,能够较好地保持系统原来的时序,这种注入法有很好的

优化性。

动态注入是在被测系统正常运行过程中,实现故障注入。该种方式是根据被测系统的运行状态或条件注入故障的,具有灵活性。

2 可靠性测试

2.1 可靠性测试过程

软件产品在需求分析阶段时,就需要启动针对软件的可靠性分析。可靠性测试过程^[12]主要包括:仔细研读软件需求说明书、分析测试需求、提交测试策略、设计测试用例、搭建测试环境、执行可靠性测试、分析测试结果,如图 1 所示。

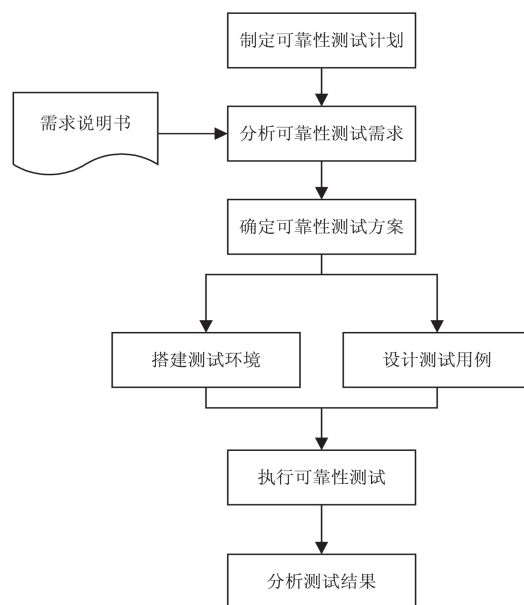


图 1 可靠性测试过程

(1) 制定测试计划:明确可靠性测试的对象、目的、内容、方法、进度要求。

(2) 分析可靠性测试需求:根据需求说明书,确定可靠性测试方案,编写测试策略。

(3) 设计测试用例:设计测试用例设计,编写自动化测试用例脚本。

(4) 搭建测试环境:搭建测试所需的各种软硬件环境,如数据、网络、测试工具的安装和设置等。

(5) 执行可靠性测试:采用手动或者自动化模式执行可靠性测试,记录测试数据。

(6) 分析测试结果:根据可靠性测试数据,分析测试结果,提交测试报告。

2.2 可靠性测试过程中的角色

软件可靠性测试涉及到以下角色:系统工程师、测试架构师、软件开发工程师、软件测试工程师和质量工程师。各角色各司其职,分工协作共同完成软件可靠性测试。

各角色的职责和任务:

- (1) 系统工程师(SE): 跟客户确认对软件产品的需求, 编写软件特性设计规格文档, 并在特性设计规格文档中给出针对软件可靠性的相关技术方案和要求。
- (2) 测试架构师(TSE): 根据软件特性设计规格文档, 编写可靠性测试策略与测试用例。
- (3) 软件开发工程师: 根据软件特性设计规格文档进行编码、处理软件可靠性的各类故障问题。
- (4) 软件测试工程师: 搭建可靠性测试环境, 编写自动化测试用例脚本, 执行软件可靠性测试。
- (5) 质量工程师(RQA): 负责对软件开发与软件

测试过程中各项流程的监管与审核。

3 基于云网络环境的软件实例

该实例是一个云化通讯软件产品开发项目, 总的代码量大约三百万行。基于云环境进行可靠性测试, 在云网络环境中注入故障, 测试软件产品对于环境的容错能力。

3.1 云环境的层级结构

软件产品实例的云环境的层级结构^[13-15]如表 1 所示。

表 1 软件产品运行的云环境层级结构

序号	类别	层级	配置项	软件产品实例
1	应用软件	产品业务	软件产品	通讯软件产品
2	基础软件平台	网络管理平台	网络操作和管理单元	CGP 平台层(后台 OMU)
3	云环境	虚拟计算节点	虚拟机	16 个虚拟机
4	虚拟设备	虚拟网络设备	私有云(VPC) 网络	单位的私有云网络
5	硬件	硬件设备	服务器、存储设备、网络设备	E9000 刀片服务器、S3900 磁阵、交换机

3.2 云网络环境的部署和软件产品的安装

3.2.1 配置硬件设备

S3900 磁阵(存储设备) 与 11 块 E9000 单板上架与连线; E9000 单板与交换机系统连接, 交换机系统可以连接局域网和广域网; 通过网络登录 E9000 单板的设备管理页面, 将所有的 E9000 单板下电之后的状态设置为 PXE(preboot execute environment, 预启动执行环境) 启动状态。

3.2.2 部署单位私有云

安装 FusionSphere 软件(华为云操作系统, FusionSphere 云操作系统从 5.0 之后的版本全部基于 OpenStack) 其中一块 E9000 单板作为虚拟机软件的控制节点, 将剩余的 E9000 单板通过 PXE 的方式加载到 FusionSphere 的硬件管理页面上; 通过 FusionSphere 软件配置网络平面、对接存储设备、划分管理节点与服务节点等。

3.2.3 创建虚拟机

在管理节点下创建 FusionSphere Openstack 虚拟机; 创建管理节点与服务节点; 创建管理节点与服务节点的用户。

- (1) 登入管理节点用户, 创建管理节点所需要的网络。
- (2) 登入计算节点用户, 创建计算节点所需要的网络。
- (3) 在管理节点用户下创建搭建 CGP 所需要的第三方用户。

创建 16 个虚拟机, 虚拟机的规格数据包括操作系统、CPU、内存、系统盘、数据盘以及网络配置等。在虚

拟机上配置网卡, 在网卡上配置私有云 IP 地址。这些虚拟机组成 VLAN 局域网。

3.2.4 安装 CGP 网络管理平台(安装 OMU 虚拟机)

上传 CGP(carrier grade platform) 的版本包以及镜像包, 虚拟机上添加网卡与浮动 IP 地址之后, 分别安装主、备 OMU 虚拟机。OMU 安装完成后, 组成基于 Client/Server 结构的网络系统。OMU 的客户端工具为 LMT(local maintenance terminal) 。

LMT 工具通过 MML(man-machine language) 命令, 对网元进行操作和维护。通过网络系统的管理用户(用户名 admin) 负责网络系统管理。

3.2.5 部署安装软件产品

通过 LMT 客户端工具安装软件产品, 上传产品软件的版本包以及镜像包之后, 安装软件产品, 给产品业务虚拟机配置网络环境。软件产品安装完成后, 涉及到故障注入的虚拟机如表 2 所示。

表 2 虚拟机功能描述

产品业务 虚拟机名称	虚拟机功能
OMU(operation management unit)	基于 Client/Server 结构的网管系统, 提供人机接口, 负责管理虚拟机上软件运行状态(主备方式)
CMU(connection maintenance unit)	处理呼叫相关的业务流程, 对其他进行呼叫过程中的连接管理(主备方式)
HRU(high-speed routing unit)	受 CMU 控制, 完成呼叫流程中的信息、消息的转发, 以及媒体消息的处理(负荷分担方式)
VPU(voice&video process unit)	负责语音、视频等媒体数据的处理(负荷分担方式)

4 搭建测试环境

4.1 故障注入方法

在本软件实例中,通过五种方法注入故障。

4.1.1 CFE 工具注入

CFE(common fault-injection entry) 工具是公司自研的 Linux 操作系统下的故障注入工具,可以针对软件产品进行 CPU 消耗故障、内存消耗故障以及多种网络故障的注入。

4.1.2 TC 工具注入

TC(traffic control) 流量控制器^[16],是 Linux 系统下的开源软件,可以为各类数据提供不同的带宽,从而控制它们的传输速度。在测试云化软件的可靠性时,可以使用 TC 工具注入网络丢包、错包以及时延等故障。

4.1.3 通过 MML 命令注入

通过 LMT 客户端登录 OMU,通过执行 MML 命令实现软件模块重启、虚拟机复位、模块主备倒换故障注入。

4.1.4 通过 ifconfig 命令注入

ifconfig 命令是 Linux 操作系统的环境配置命令,通过执行 ifconfig 命令,改变虚拟机网络配置,用于注入虚拟机网络平面端口故障。

4.1.5 通过 openstack nova 命令注入

在云环境中通过 openstack 虚拟化软件管理所有的虚拟机,执行可靠性测试时,可以通过 openstack nova 命令注入故障,故障类型为虚拟机重启与虚拟机迁移。

4.2 性能测试工具 NTE

NTE(network traffic emulator) 工具是公司自研的性能测试工具,可以对通讯软件产品在实际使用环境中的客户呼叫进行计算机仿真。在进行软件产品的可靠性测试时,需要在用户呼叫仿真的背景下,注入各种故障,来获取测试数据。根据测试数据分析软件产品的可靠性。

4.3 搭建测试环境

(1) 部署软件实例的私有云网络环境;

(2) 通过 LMT 客户端登上 OMU 后,执行 MML 命令部署软件实例的业务虚拟机与进程;

(3) 通过 LMT 客户端登上 OMU 后,使用 LMT 上的“文件传输服务”将 CFE 工具和 TC 工具上传到 OMU 虚拟机之后,再通过 scp 命令将这两个工具安装到执行故障注入的虚拟机上;

(4) 将性能测试工具 NTE 与自动化脚本工具 GTR 安装到指定执行机上。

5 基于云环境的可靠性测试

结合软件实例,介绍基于云环境的可靠性测试工作。

5.1 测试用例设计

测试架构师分析软件的功能与性能测试特性之后,选定可靠性测试模型和注入故障类型,对虚拟机或者软件模块注入故障,开展可靠性测试,共设计 40 个可靠性测试用例,如表 3 所示。

表 3 可靠性测试用例分类

注入故障类型	测试用例个数	故障注入描述	故障注入方法
核(CPU) 离线	2	在故障注入执行机上,执行./cfe “inject rCPU_Pendings(time ,cpuid) values (20 ,1) ”注入虚拟机核离线故障	CFE
虚拟机内存消耗故障	2	在故障注入执行机上,执行./cfe “inject rmem_low (total) values (6G) ”注入虚拟机内存消耗故障	CFE
错包	2	在故障注入执行机上,执行 tc qdisc add dev eth2 root netem corrupt 10%注入虚拟机错包故障	TC
丢包	2	在故障注入执行机上,执行 tc qdisc add dev eth2 root netem loss 10%注入虚拟机丢包故障	TC
时延	2	在故障注入执行机上,执行 tc qdisc add dev eth2 root netem delay 1 000 ms 100 ms 10%注入虚拟机时延故障	TC
模块进程倒换故障	2	使用 MML 命令 SWP MODULE,对指定模块执行故障注入	MML
模块复位故障	4	使用 MML 命令 RST MODULE,对指定模块执行故障注入	MML
虚拟机重启故障	3	使用 MML 命令 RST VM,对虚拟机执行故障注入	MML
虚拟机在 I 层重启故障	5	使用 PUTTY 工具登录虚拟化软件后台,执行“nova reboot 虚拟机 ID”,对指定虚拟机执行故障注入	nova
虚拟机迁移故障	4	使用 PUTTY 工具登录虚拟化软件的后台,执行“nova reboot 虚拟机 ID”,对指定虚拟机执行故障注入	nova
虚拟机网络平面故障	12	通过 OMU 虚拟机,登录到待注入故障的虚拟机的后台,执行 ifconfig 命令,down 掉指定端口的方式执行故障注入	Linux 操作系统 命令: ifconfig

5.2 可靠性测试流程

云化通讯软件实例的可靠性测试流程如图2所示。在测试环境中启动业务软件 and 性能测试工具后,通过注入故障,开展测试工作。

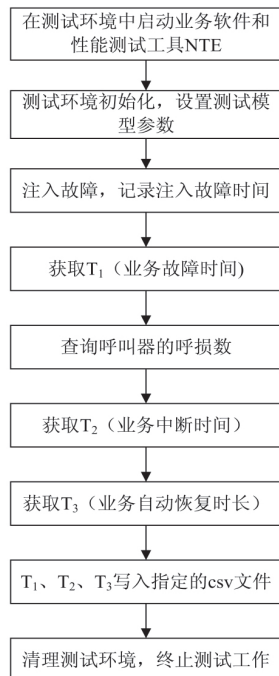


图2 基于故障注入的云化软件可靠性测试流程

图2中专业术语注解:

CAPS(call attempt per second, 每秒呼叫次数): 这里所指的呼叫次数包括成功呼叫(即呼叫接续成功, 双方进行通话)次数和不成功呼叫(呼损)次数。

T_1 (业务故障时间): 故障注入时间到故障发生时间之间的时差, 目前采用的故障注入时间到故障告警出现时间的差值。

T_2 (业务中断时间): 不成功呼叫(呼损)占用的时间; 每隔固定的时间间隔查询读取呼叫器上业务呼损数 sn ; $t = sn / CAPS$, 计算 t 的数值并且写入 $fail_caps$ 数组中; 最后选择数组中的最大值作为 T_2 。

T_3 (业务自动恢复时长): 开始出现呼损的时间到不再出现呼损的时间之间的时差; 在呼叫器上查询呼损数, 一直到呼叫器上不再出现新的呼损(也就是呼损数 sn 不再增加), 然后取两个时间差值作为业务自动恢复时长。

5.3 自动化测试

该软件实例的可靠性测试采用自动化测试, 脚本语言选择 TCL, 通过调用 spider 库中 NTE 的自动化的函数编写脚本和测试套, 通过 GTR 工具编辑、调试、运行 TCL 脚本和测试套。

5.3.1 编写测试套

测试用例评审之后, 测试工程师需要根据测试用例特点编写测试套完成测试环境的初始化, 提供通用

的测试模型参数, 测试结束之后测试套进行测试环境清理。目前使用的呼叫模型参数有2种:

(1) 透传模型: G711-G711, 每秒20个呼叫(caps = 20), 呼叫在线时长 10 000 ms;

(2) 转码模型: G711-G729, 每秒5个呼叫(caps = 5), 呼叫在线时长 10 000 ms。

测试套的放置在指定路径下, 供 TCL 自动化脚本调用。

5.3.2 编写自动化测试脚本

根据图2编写 TCL 脚本, 主要对以下测试场景编写 TCL 脚本。

(1) 调用测试套进行测试环境初始化, 设置相关测试模型参数, 启动性能呼叫;

(2) 在 MEID=0 的 CGP 网元下, 执行 DSP TIME 命令, 并将查询到的时间存入变量 $time_1$, 根据测试用例中的故障类型开始执行故障注入;

(3) 在 MEID=0 的 CGP 网元下, 执行 LST ALM 命令查询到告警产生的时间存入变量 $time_2$;

(4) 将变量 $time_2$ 与 $time_1$ 的差值存入变量 T_1 中, 作为业务故障时间;

(5) 执行故障注入之后, 每隔固定的时间间隔, 查询 NTE 上呼叫器的呼损数, 呼损数除以 caps, 将结果写入数组 $fail_caps$ 中。最后从数组 $fail_caps$ 中取出最大的数值, 存入变量 T_2 中, 作为业务中断时间;

(6) 不断查询 TTE 工具上呼叫器的呼损数, 一直到呼叫器上不再出现新的呼损(也就是呼损数不再增加) 将开始出现呼损到呼叫器上不再出现呼损之间的时长存入变量 T_3 中, 作为业务自动恢复时长;

(7) 测试结束之后, 统计到的数据 T_1 、 T_2 、 T_3 以添加的方式写入指定路径下的 csv 文件中;

(8) 清理测试环境, 终止测试工作。

5.3.3 可靠性自动化脚本连跑

测试工程师完成自动化脚本编写之后, 脚本经过测试架构师审查通过后, 所有的脚本提交到配置库 Git 中, 并将脚本同步提交到软件产品的自动化工厂。每一轮迭代开发结束之后, 通过全部脚本自动化连跑, 对软件产品进行可靠性测试, 这样提高了测试效率。

5.4 测试结果分析

由测试工程师根据软件产品的可靠性规格基线与可靠性自动化测试的结果进行分析。表4给出了软件实例在云环境中, 进程复位、倒换以及虚拟机复位这些场景对于 T_2 (业务中断时间) 的测试分析结果。

从测试的结果可知, 复位与倒换故障注入时, 待测进程与虚拟机上的业务中断时间在允许范围内, 产品的可靠性较好。

其他测试数据由于篇幅所限, 文中不再介绍。在

可靠性测试过程中,没有达成软件可靠性基线指标的用例,测试工程师将测试的情况反馈给软件开发工程师,并和开发工程师一起定位解决。

表 4 T_2 (业务中断时间)的可靠性测试分析结果

测试项	可靠性基线指标/s	实际测试数值/s	分析结果
进程复位	≤ 3	2	达标
进程倒换	≤ 3	2	达标
CMU 虚拟机复位	≤ 300	50	达标
HRU 虚拟机复位	≤ 300	70	达标
VPU 虚拟机复位	≤ 300	64	达标
OMU 虚拟机复位	≤ 300	207	达标

5.5 测试过程中遇到的典型问题

云化软件在可靠性测试过程中,遇到的一些典型问题,举例如下。

(1) 测试过程中无法正常切换 root 用户导致脚本执行失败。

测试虚拟机迁移的场景时,需要切换为 root 用户,调用 nova 命令执行虚拟机迁移。测试过程中发现登录 FusionSphere 虚拟机化软件后台之后,脚本执行失败。经过定位发现,完成 FusionSphere 软件安装之后,需要修改/etc/ssh/sshd_config 文件,去掉对 root 用户加密限制;tel 自动化测试脚本才能正常切换到 root 用户,执行虚拟机迁移的 nova 命令。

(2) 测试过程中未在脚本中加入环境恢复步骤。

测试虚拟机网络平面故障的场景时,执行 ifconfig ethX down 命令 down 掉虚拟机上的指定网口来实现故障注入;测试工作完成之后没有对 down 掉的网口进行恢复,直接进行下一个相关用例测试,导致测试脚本执行失败。定位脚本运行失败的原因时,发现了这个问题,在测试脚本中增加了恢复环境的场景,以后没有出现这种情况。

在云环境中开展可靠性测试工作,有许多问题需要进一步进行深入探讨和研究。

6 结束语

随着云计算技术的高速发展,基于云化环境的软件在各个领域的应用不断普及和拓展。许多传统环境下的软件产品需要推出云化的软件版本,不仅需要关注软件的各种功能测试与性能测试,可靠性测试的重要性也逐步提升。基于云环境的可靠性测试是软件测试过程中的重要环节,尤其对于大型云化软件显得尤

为重要。工作实践表明,做好基于云环境的软件可靠性测试工作,在软件开发过程中不断解决存在的各种问题,有助于提高软件产品的质量,提高上线后软件的安全性及稳定性,以更好地满足客户对软件产品的需求。

参考文献:

- [1] 陆民燕.软件可靠性工程[M].北京:国防工业出版社,2011.
- [2] 张玲玲,王林章.基于故障剖面的安全关键系统可靠性测试与评估[J].计算机与数字工程,2014,42(12):2304-2310.
- [3] 佟双军.基于 S698 芯片的软件故障注入技术的研究[D].哈尔滨:哈尔滨工业大学,2014.
- [4] 潘庆和.软件故障注入关键技术研究[D].哈尔滨:哈尔滨工业大学.
- [5] 江铁成.软件故障注入方法在星载系统中的应用[J].安庆师范学院学报:自然科学版,2010,16(1):39-42.
- [6] 范文豪,马捷中,孙姜燕.一种针对 VxWorks 系统的通用软件故障注入方法[J].测控技术,2011,30(4):100-103.
- [7] 王立荣.基于软件故障注入的嵌入式系统鲁棒性测试[J].微型机与应用,2011,30(22):14-16.
- [8] ARLAT J,AGUERA M,AMAT L,et al.Fault injection for dependability validation: a methodology and some applications[J].IEEE Transactions on Software Engineering,2003,16(2):166-182.
- [9] HSUEH M C, TSAI T K, IYER R K. Fault injection techniques and tools[J].Computer,1997,30(4):75-82.
- [10] KIM C H, QUISQUATER J J. Faults injection methods, and fault attacks[J].IEEE Design & Test of Computers,2007,24(6):544-545.
- [11] 陈锦富,卢炎生,谢晓东.软件错误注入测试技术研究[J].软件学报,2009,20(6):1425-1443.
- [12] 史玉帅.基于虚拟机架构的云平台可靠性验证方法研究[D].武汉:华中科技大学,2014.
- [13] 姜文,刘立康.基于云环境的大型应用软件联调[J].计算机技术与发展,2018,28(2):14-18.
- [14] 陈国良,朋仲,冯禹洪.云计算工程[M].北京:人民邮电出版社,2016.
- [15] WANG Jun, MENG Fanpeng. Software testing based on cloud computing[C]//International conference on internet computing and information services. Hong Kong, China: IEEE,2011:176-178.
- [16] 龙恒.Linux 流控工具 TC 的原理及实用案例分析[J].计算机与现代化,2010(11):80-84.