

基于云计算的软件测试系统框架研究

杨本生¹, 袁祥梦², 黄晓光²

(1. 河北工程大学 资源学院, 河北 邯郸 056038; 2. 河北工程大学 信息与电气工程学院, 河北 邯郸 056038)

摘要: 针对传统的软件测试方法中存在的效率低、成本高等问题, 将云测试技术应用到了软件测试中; 介绍了云测试中的相关技术, 并详细阐述了系统总体框架, 使用了一种基于动态优先权的高优先权先调度算法, 设计并实现了系统的调度子模块; Matlab 仿真实验结果为在执行相同任务调度时, 该调度算法比传统 FCFS 调度算法总周转时间短; 当任务增加时, 该调度算法总周转时间增加的慢, 证明该算法可以显著地提高测试效率, 提高资源利用率, 缩减测试成本, 实现软件测试的自动化。

关键词: 云计算; 云测试; 动态优先权; 调度算法; Matlab

Framework of Software Testing Based on Cloud Computing

Yang Bensheng¹, Yuan Xiangmeng², Huang Xiaoguang²

(1. College of Recourses, Hebei University of Engineering, Handan 056038, China; 2. College of Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China)

Abstract: For the problem that efficiency is low and cost is high existing in the traditional software testing method, the paper tested software using cloud testing technology. It introduced related technologies including cloud testing, and described the design of overall architecture of the system in details, designed and implemented the scheduling module using a high priority first scheduling based on dynamic priority. The Matlab simulation results are that the total turnaround time of the scheduling algorithm in this paper is short when the same tasks are scheduled and increase slowly when the number of tasks is increased than traditional FCFS. This shows that this scheduling algorithm can reduce test cost and realize the automation of software testing under the condition of significantly improving test efficiency and resource utilization.

Key words: cloud computing; cloud testing; dynamic priority; scheduling algorithm; Matlab

0 引言

软件测试是为发现错误而执行程序的过程。软件测试作为保障软件质量的重要手段, 在软件工程领域越来越受到人们的重视。可是在传统单点、本地化的测试过程中购买测试工具的费用, 复杂的测试环境搭建和维护等问题严重局限着测试技术的发展。云计算是一种新型的分布式计算模式, 云计算利用虚拟化技术, 通过网络对云中的基础设施资源进行统一的管理和对外服务, 形成了一种以用户为中心的“按需使用、按量付费”的商业服务模式。James A. Whittaker 在《探索式软件测试》一书中展望了“软件测试的未来”, 其中提到了基于云计算的软件测试服务^[1]。将云计算和软件测试结合起来就是云测试。目前对云测试的研究主要围绕以下 3 个方面: (1) 针对云的测试, 涉及到云计算内部的结构、资源配置和功能扩展等多方面的测试问题^[2]; (2) 测试的迁徙, 将传统的测试方法、管理、过程和框架等迁徙到云中^[3]; (3) 利用云对其他软件系统进行测试。本文主要介绍第 3 种, 云测试服务提供商将基于云平台之上的测试以服务的方式有偿地提供给用户使用, 可以很好地解决传统测试方式中存在的问题。近几年, 云测试开始受到学术界的关注, 并取得了很多研究成果。文献[4]提出了一个带错误注入虚拟机的 D-Cloud 云计算测试环境模型。文献[5]提出了一种基于云的并行符号测试引擎 Cloud9。文献[6]

提出了一种可以测试用不同编程语言写的程序的云版本自动随机测试工具 YETI。

基于云计算的软件测试系统的相关研究正在快速地发展并成为一个热点, 在银行应用软件、铁路系统、电信移动增值业务系统等领域已经应用并取得了很好的效果。研究系统中调度策略, 使软件测试具有更好的并行性和自动性, 以提升服务的质量, 解决传统测试方式中存在的问题具有非常广泛的实用意义和应用前景^[7]。本文结合云计算、云测试的思想, 给出了针对软件测试的云测试解决方案, 设计了系统的总体结构和软件体系结构, 分析并实现了系统调度子模块。

1 云测试概述

随着软件规模的不断扩大和硬件设施的日益复杂化, 软件测试遇到前所未有的挑战。云计算的资源配置动态化、需求服务自助化、服务可计量化、资源的池化和透明化的特征可以解决这些问题。云测试是基于云环境和云中的基础设施, 利用云技术和解决方案来进行软件测试^[8]。通俗地讲, 云测试就是云计算和软件测试的结合。

云测试是一种新型测试方案。它可以将分布式的测试资源进行有效调用, 以统一平台或服务的形式呈现给用户, 为企业组织提供了一个共享可用的自动化测试环境交付平台。将云测试应用到软件测试中, 企业就不需要购买测试服务器, 购买各类价格昂贵的测试软件, 也不再需要部署复杂的测试环境, 只需要列出测试目的、测试环境要求、虚拟机台数、何时间断租用即可, 实现按需付费, 具有节约成本, 立即可用, 专家服务, 装配完备等优势; 云测试服务提供商提供 24/7/365 的按需自动化测试服务, 实现了资源共享; 更重要的是, 云测试在保证 Web 系统质量的前提

收稿日期: 2013-12-30; 修回日期: 2014-02-25。

作者简介: 杨本生(1956-), 男, 河北省武安市人, 教授, 硕士生导师, 主要从事矿图数字化、矿山压力及其控制、计算机技术在矿山应用方向的研究。

下缩短了产品进入市场投入使用的时间^[9-10]。

并不是所有的应用程序都适合进行云测试，例如，迁徙的成本大于回报^[11]。测试程序的特征决定程序的测试过程是否适合迁徙到云上，影响因素有：(1) 测试任务是相互独立的或是具有简单的相互依赖关系，实现任务的并发执行；(2) 操作环境独立且易识别，实现测试的自动化；(3) 适合自动化测试的可编程方式访问的接口，实现测试的可扩展性。云环境中执行引擎的特性决定适合在云中进行的测试包括单元测试，尤其是大型回归套件；高容量自动化测试和性能测试，其他测试类型在云中执行是非常复杂的^[12]。

一个成熟的云测试系统应该包括以下功能：可扩展的测试环境功能；多租户测试模式；数字化测试管理功能；按需自动化测试和控制服务；测试解决方案集成和组合功能；测试跟踪和监控功能；大规模测试仿真功能；测试承包和计费功能^[13]。

2 系统总体设计

2.1 云测试步骤

在云中对其他软件进行测试的过程中，首先由用户向系统提出测试请求，系统接受测试任务，测试任务调度分配，提供中间件服务，虚拟资源匹配，测试任务执行和监督，最后收集测试结果。云测试步骤如图 1 所示。

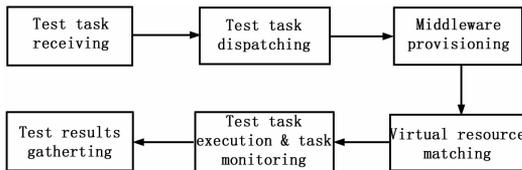


图 1 云测试步骤图

2.2 系统架构设计

云计算有 3 种服务模式，基础设施即服务 IaaS (Infrastructure as a Service)、平台即服务 PaaS (Platform as a Service)、软件即服务 SaaS (Software as a Service)，云测试可以看作是 SaaS 业务模式在软件测试领域中的应用，所以系统以这 3 种服务模式为依据。结合云测试步骤，设计云测试系统架构，如图 2 所示。

(1) IaaS 层：系统利用虚拟化技术，将处理器、存储、网络和其他基础设施资源虚拟成逻辑上的资源池，以服务的形式统一集中的提供给用户，用户根据自己的需求向系统提出各种资源的请求，而无需关心资源是如何分配和调度的，提高了硬件资源的利用率，使测试过程更加智能化、自动化。

(2) PaaS 层：PaaS 是软件研发的业务基础平台，向用户提供进行定制化研发的中间件平台，同时包括对基础设施资源和用户提交的测试任务的管理。该层由虚拟资源管理模块，中间件管理模块和测试任务管理模块组成。其中，在虚拟资源管理模块实现了对虚拟化的基础设施的调度和分配；中间件管理模块中分为安全管理，服务等级协议 SLA (Service Level Agreement)，性能监控和结果收集功能；测试任务管理模块中包括对用户的管理和对测试任务的调度。

在虚拟资源管理模块中，各种虚拟资源可能处于 2 种状态：空闲状态、运行状态。虚拟资源管理器将各种资源按其状态分到 2 个队列中，对调度好的测试任务，根据资源的状态，分配符合要求的虚拟资源。

(3) TaaS (Testing as a Service) 层：系统根据用户通过

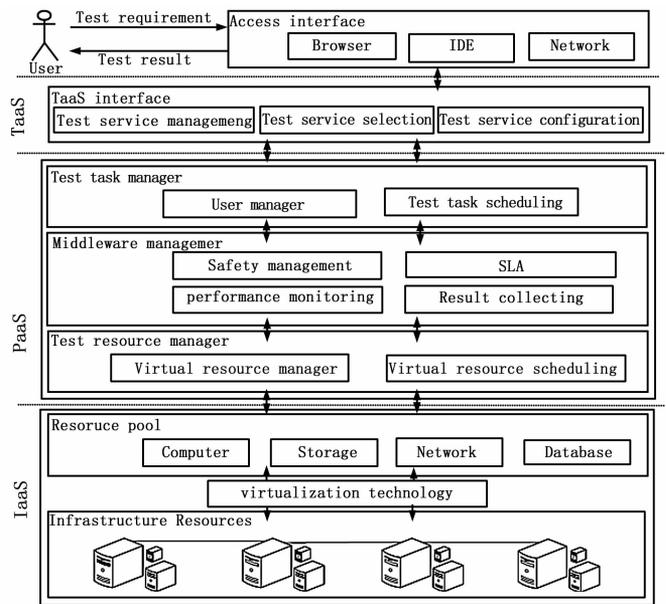


图 2 系统架构图

访问接口传来的测试需求，选择空闲的匹配的软硬件进行安装，配置和维护目标测试环境。此外，用户享有软件使用权，还可以不断升级。

除此之外，用户通过浏览器、IDE (Integrated Development Environment) 或是编程等方式与云测试系统进行交互，提交测试请求，使用系统提供的测试服务，最后得到测试结果。

3 调度子模块的设计与实现

在云测试中，测试任务管理模块中对平行测试任务的部署调度策略非常重要，因为这涉及虚拟设备的部署调度以及测试任务的动态部署调度^[14]。决定了任务的执行顺序及分配给各任务的虚拟资源。本文设计并实现了云测试系统中的调度子模块。

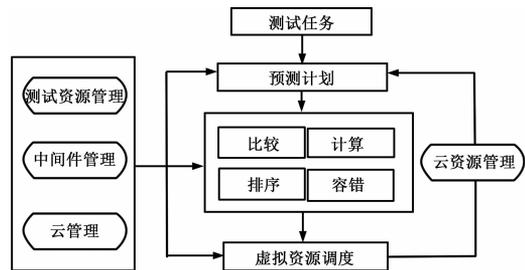


图 3 调度子模块结构图

3.1 调度子模块结构图

用户提交测试任务之后，根据任务的数量以及对测试项目的分析，预测出需要的虚拟机数量是不可缺少的，这样可以减少因请求过多虚拟机而造成的浪费，也可以避免测试执行过程中虚拟机太少重新向云资源管理器申请带来的时间浪费。调度子系统模块结构图如图 3 所示。

3.2 算法过程

适合进行云测试的测试任务之间是相互独立的，在云测试平台中的测试任务调度过程中，不需考虑各任务之间的依赖关系，这对调度过程中使用的算法是一个简化。调度系统使用一

种基于动态优先权的高优先权先调度算法, 即测试任务的优先权会随着任务的执行情况或是任务的等待时间增加而改变, 优先权高的任务先执行。

(1) 在调度过程中, 假设任务数和虚拟机数是有限的。设有 n 个独立的任务 $T = \{T_1, T_2, \dots, T_n\}$, 由 m 个虚拟机 $M = \{M_1, M_2, \dots, M_m\}$ 执行。现约定, 任何任务可以在任何一台虚拟机上执行, 但未完工之前不允许中断处理。任何任务不能拆分成更小的子任务。第一次调度之后的每次调度过程中, 等待执行的任务数赋值给 n , 空闲虚拟机数赋值给 m 。

定义 2 个队列: 测试任务等待队列 WaitList; 空闲虚拟机队列 FreeVMList。

(2) 用户通过系统接口提交测试任务, 并按提示给各任务赋权重值, 加入测试任务等待队列 WaitList。

用户在提交测试任务时, 决定每个任务的权重, 根据测试任务的等待时间, 处理时间以及权重得到任务的优先权计算公式如下:

$$p_j = \frac{q_j + d_j}{d_j} + \pi w_j$$

其中: T_j 表示第 j 个任务; M_i 表示第 i 个虚拟机; q_j 表示任务的等待时间; d_j 表示任务处理时间; w_j 表示任务的权重; p_j 表示任务的优先级; π 为引进的参数, 调节 $\frac{q_j + d_j}{d_j}$ 与 w_j , 使两者处于同一数量级。

(3) 根据对要测试的项目需求和用户提交的测试任务数量的分析, 预测虚拟机的数量、配置等, 向云资源管理器发送消息, 申请虚拟机, 然后根据环境配置模板自动部署操作系统、软件系统以及测试系统等完成测试环境的搭建。这样可以在测试执行过程中减少对新资源的请求。当然, 这种根据经验和模板的预测存在一定的误差。

(4) 将 WaitList 队列中任务数赋给 n , FreeVMList 队列中空闲虚拟机数赋给 m 。

(5) 当 $n \leq m$ 时, 释放多余的处理机, 将虚拟机 M_i 分配给任务 T_j 即可, 并在测试任务执行完之后立即释放对应的处理机。

(6) 当 $n > m$ 时, 根据优先权的计算公式, 计算出所有任务的优先权。按优先权排序, 取 WaitList 中前 m 个任务分配给 m 个虚拟机。将任务 $T_1 \sim T_m$ 从队列 WaitList 中删除, 将虚拟机 $M_1 \sim M_m$ 从队列 FreeVMList 中删除。

(7) 将用户添加的新任务和执行过程中出错的任务添加到任务等待队列 WaitList 中, 等到有空闲虚拟机时, 将空闲虚拟机加到队列 FreeVMList 中, 重复执行步骤 (4) 直到所有任务都得到正确执行。

3.3 算法流程图

算法的优点: ①当 $n \leq m$ 时, 不需要考虑每个任务的优先权, 总执行时间为运行时间最长的那个任务的运行时间, 即需要 $O(1)$ 时间。当 $n > m$ 时, 排序耗时 $O(n \log n)$, 循环耗时 $O(n \log m)$, 因此算法所需的时间是 $O(n \log n + n \log m) = O(n \log n)$ 。②每个任务执行完后立即释放虚拟机, 云平台中的资源利用率得到提高。③等待时间长, 执行时间短, 权重大的任务优先执行。当然, 利用该算法时, 每次调度过程之前, 都须先重新计算所有任务的优先权, 会增加系统的开销。算法流程图如图 4 所示。

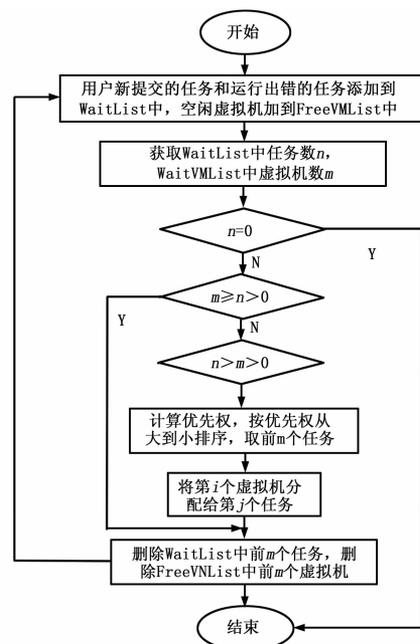


图 4 算法流程图

4 实验验证

在 Matlab 中编码实现本文算法和经典的 FCFS 算法, 仿真实验, 从数学的角度来分析本文算法和 FCFS 算法的运行效率。实验取假设虚拟机数为 250 个, 测试任务数 n 分别取 $\{50, 150, 250, 350, 450\}$, 各测试任务执行时间不相同, 执行时间最长的任务 T_j 执行时间为 12 s, 从仿真结果图 5 可以看出: (1) 当虚拟机数量大于任务数时, 利用本文调度算法运行相同量的功能测试任务和性能测试任务总周转时间是常数, 而 FCFS 算法是增加的; (2) 当虚拟机数量小于任务数时, 本文调度算法的总周转时间增加平缓, 而 FCFS 算法增加快速; (3) 随着测试任务量的增加, 本文算法的执行效率比 FCFS 算法的执行效率高将近 20%, 优势更加明显。

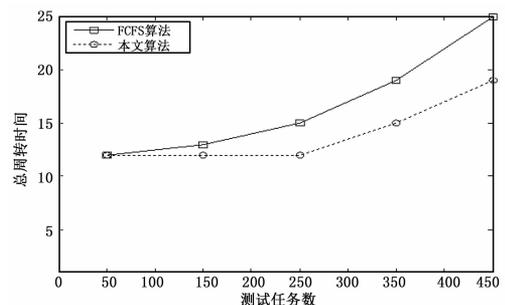


图 5 与 FCFS 调度算法总周转时间的比较

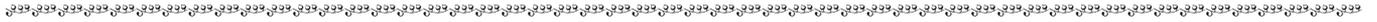
从实验结果分析, 与传统的测试方式相比较, 在本系统中运行相同量的功能测试任务和性能测试任务, 不需要花费大量投资于测试软件, 需要的硬件, 测试人员以及测试环境后期的维护、升级, 只需要向云平台服务提供商支付一定的租赁服务费用即可, 测试成本明显减少。此外, 测试任务提交之后, 只需等待一段时间, 即可得到测试结果报告, 测试效率得到显著提高。

5 结束语

本文结合云计算技术,详细介绍了基于云计算的云测试系统的结构模型框架,使用一种基于动态优先权的高优先权先调度算法实现了系统的调度子模块。通过 Matlab 中的测试实验可得,与经典的 FCFS 调度算法相比该调度算法可以显著提高对软件系统的测试效率,降低测试成本。在云测试系统中,其他模块的实现以及系统的安全性是下一步工作的重点,应对该技术进一步研究。

参考文献

[1] Whittaker J A. 探索式软件测试 [M]. 北京:清华大学出版社,2010.
 [2] Chan W K, Mei L J, Zhang Z Y. Modeling and testing of cloud applications [A]. Proceedings of Services Computing Conference [C]. Singapore: IEEE Press, 2009: 111 - 118.
 [3] Parveen T, Tilley S. When to migrate software testing to the Cloud [A]. Proceedings of Software Testing, Verification, and Validation Workshops [C]. Paris: IEEE Press, 2010: 424 - 427.
 [4] Banzai T, Koizumi H, Kanbayashi R. D- Cloud: design of a software testing environment for reliable distributed systems using cloud computing technology [A]. Proceedings of IEEE/ACM International Conference on Cluster, Cloud and Grid Computing [C]. Melbourne: IEEE Press, 2010. 631 - 636.



(上接第 1672 页)

统、导航系统、空中交通管制系统及飞行驾驶仪表显示、警告系统等不可缺少的信息。大气数据信息的准确性对提高飞行安全和降低寿命周期费用起着重要作用^[10]。运用某型自动测试系统 (Automatic Test System, ATS) 对某型大气数据计算机进行测试,得到某次测试的部分测试结果如表 1 所示。

表 1 某型大气数据计算机的测试结果

序号	测试项目	测试结果
1	REF1	10.1 V
2	REF2	10.2 V
3	REF3	9.9 V
4	离散量 1	高电平
5	离散量 2	高电平
6	离散量 3	高电平
7	气压高度	12 018 m
8	升降速度	56 m/s
9	真空速	2 543 km/h
10	迎角	-10
11	侧滑角	12

假定在自动测试过程中所有的激励条件都能够得到满足,即可拓知识的激励基元都匹配成功,故只需对可拓知识的测试基元进行匹配。将 ATS 得到的测试数据与可拓知识 R_1 、 R_2 、 R_3 和 R_4 进行匹配,代入公式 (3) 得到 $D_1 = -0.367$, $D_2 = 0$, $D_3 = -2.33$ 和 $D_4 = 1$, 所以可拓知识 R_1 、 R_2 、 R_3 匹配成功,即大气数据计算机的电源组合正常、离散量正常、数字量正常;可拓知识 R_4 匹配失败,即无法确认模拟量正常,还需要其它知识对模拟量状态进行进一步诊断。

[5] Ciortea L, Zamfir C, Bucur S. Cloud9: A Software Testing Service [J]. ACM SIGOPS Operating Systems Review, 2010, 43(4): 5 - 10.
 [6] Oriol M, Ullah F. YETI on the cloud [EB/OL]. Proceedings of the International Workshop on Software Testing in the Cloud; Paris: IEEE Press, 2010: 434 - 437.
 [7] 方锦明. 云计算中虚拟资源调度的决策系统 [J]. 计算机测量与控制, 2011, 19 (12): 3145 - 3148.
 [8] Wikipedia. Cloud testing [EB/OL]. http://en.wikipedia.org/wiki/Cloud_testing. 2013. 8.
 [9] Priyanka, Chana I, Rana A. Empirical evaluation of cloud-based testing techniques: a systematic review [Z]. ACM SIGSOFT Software Engineering Notes, 2012; 37 (3) .1 - 4.
 [10] Gao J, Bai X Y, Tsai W T. Cloud Testing— Issues, Challenges, Needs and Practice [EB/OL]. Software Engineering: An International Journal (SEIJ), 2011, 1 (1): 9 - 23.
 [11] Tilley S, Parveen T. Software testing in the cloud: migration and execution [M]. Springer, 2012.
 [12] 李 乔, 柯栋梁, 王小林. 云测试研究现状综述 [J]. 计算机应用研究, 2012, 29 (12): 4401 - 4406.
 [13] Gao J, Bai X Y, Tsai W T. Testing as a service (TaaS) on clouds [EB/OL]. Proceedings of Service - Oriented System Engineering. IEEE press, 2013; 212 - 223.
 [14] Pinedo M L. Scheduling: theory, algorithms and systems [M]. Third Edition. Springer, 2008. 118 - 130.

5 结论

为克服经典知识表示方法的缺点,将基元理论与经典知识表示方法相结合,实现了可拓知识表示;针对故障诊断专家系统知识的共性,提出了故障诊断专家系统可拓知识的一般模型,通过对模型的分析得出了可拓知识匹配的一般步骤,并构建了属性精确值和区间值混合的匹配度计算公式;最后以 ATS 的测试数据为例对某型大气数据计算机的可拓知识进行匹配,结果表明该方法匹配速度快、匹配效率高、操作简便易行。

参考文献:

[1] Yang C Y, Cai W. Knowledge representations based on extension rules [A]. Proceedings of the 7th World Congress on Intelligent Control and Automation [C]. Chongqing: 2008, 1455 - 1459.
 [2] 周德新, 谢晓敏. 基于可拓规则的故障诊断专家系统推理机的研究 [J]. 计算机测量与控制, 2011, 19 (2): 266 - 272.
 [3] 张海涛, 董春游. 基于 Web 的可拓专家系统研究 [J]. 智能系统学报, 2009, 4 (2): 175 - 179.
 [4] 赵燕伟, 苏 楠. 可拓设计 [M]. 北京: 科学出版社, 2010.
 [5] 王玉刚, 杨建新. 航电设备故障诊断专家系统 [J]. 计算机系统应用, 2012, 21 (12): 26 - 28.
 [6] 李 欣, 乔 颖, 李 想, 等. 基于 ECA 规则推理的故障诊断技术 [J]. 计算机工程与设计, 2011, 32 (3): 1023 - 1028.
 [7] 曹立军, 王兴贵, 秦俊奇, 等. 融合案例与规则推理的故障预测专家系统 [J]. 计算机工程, 2006, 32 (1): 208 - 210.
 [8] Yang C Y, Wang G H, Li Y, et al. Study on knowledge Reasoning based on extended formulas [A]. International Conference on AIAI [C], New York: Springer Verlag, 2005.
 [9] 吴明强, 李霁红, 曹爱东, 等. 故障诊断专家系统综合智能推理技术研究 [J]. 计算机测量与控制, 2004, 12 (10): 932 - 934.
 [10] 马银才, 张兴媛. 航空机载电子设备 [M]. 北京: 清华大学出版社, 2012.