Mobile Cloud Computing Testing Review

Conference Paper · December 2013

DOI: 10.1109/ACSAT.2013.42

Ahmad Al-Ahmad
Al-Madinah International University
9 PUBLICATIONS 7 CITATIONS
SEE PROFILE

Syed Aljunid
Universiti Teknologi MARA
20 PUBLICATIONS 40 CITATIONS
SEE PROFILE

Anis Shobirin Abdullah Sani
Universiti Teknologi MARA
2 PUBLICATIONS 5 CITATIONS
SEE PROFILE

Mobile Cloud Computing Testing Review

Ahmad Salah Al-Ahmad
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA
40450 Shah Alam, Selangor, Malaysia
e-mail: 2012230098@isiswa.uitm.edu.my

Syed Ahmad Aljunid
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA
40450 Shah Alam, Selangor, Malaysia
e-mail: aljunid@tmsk.uitm.edu.my

Anis Shobirin Abdullah Sani
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA
08400 Merbok, Kedah, Malaysia
e-mail: anisshobirin@kedah.uitm.edu.my

Abstract— **As mobile cloud computing applications are becoming widely used to mitigate the limitations of mobile devices and exploit cloud services, testing these applications becomes critical. Testing mobile cloud application is complicated further due to dynamic assign of cloud resource to applications that run on mobile devices, due to the increased multiple factors in test case generations and execution. This paper will critically review the three related areas of mobile, cloud and mobile cloud applications testing, in terms of features and models. Furthermore, it will show the necessity of a mobile cloud computing applications testing model due to its uniqueness from both native mobile application testing models and cloud applications testing models. Based on this review, although testing mobile cloud computing applications is critical, there is currently no model to test mobile cloud computing applications.**

Mobile cloud computing, mobile, cloud, testing model (key words)

I. Introduction

The number of clients in mobile and cloud markets is rapidly increasing. In Malaysia alone, 80% of the population between 20-49 years old own mobile phones [1]. On the other hand, in USA during first quarter of 2012 more than two-thirds of companies have implementation over cloud, or planning to have such implementation in the future [2]. As a result of high usage of mobile and cloud, both academics and the IT industry have pushed to improve and merge both mobile computing and cloud computing technologies into Mobile Cloud Computing (MCC) and provide possible solutions to MCC-paradigm challenges.

Thus the MCC paradigm is the intersection between both mobile and cloud computing, as illustrated by Figure 1; it overlaps both technologies yet it presents its own uniqueness over both paradigms. In particular, MCC allows developers to create applications with enhanced cloud services that have fewer limitations over native mobile applications. However, MCC comes with higher complexity that affects the not only the reliability and delivery of its application but also its development lifecycle.

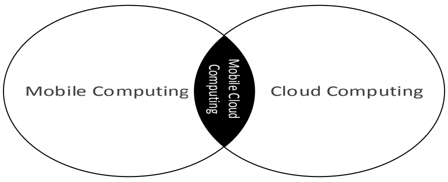

Figure 1. Mobile Cloud Computing

With respect to testing, currently there are a few separate testing frameworks for mobile applications and cloud applications [3] but none on MCC applications. As testing is very expensive in terms of time and resources, it is important to have a model or a framework that can guide testers in their tasks; this testing model would reduce time, cost, number of testers required and helps in providing high quality testing results.

This rest of this paper is organized as follows: section 2 will introduce cloud computing, its testing features and testing models; section 3, on the other hand will delve on mobile computing, its testing features and testing models, while section 4 will discuss testing features and testing models for MCC. Section 5 concludes the paper.

II. Cloud Testing

Cloud testing can be used for testing cloud applications or using the cloud to test applications of different type such as Web application and mobile applications or testing cloud itself in term of management software and infrastructure back-end applications.

Cloud testing can be defined as “a new form of software testing in which web applications that use cloud computing environments seek to simulate real-world user traffic as a means of load testing and stress testing web sites”[4].

A. Cloud Testing Features

Cloud testing would enhance testing applications process, by providing high performance, dynamic resources, less expensive, high availability, centralization and easier to deploy features [5].

This new Cloud Testing technique is restricted to some types of test and to application with special features. Hence, unit testing, high volume automated testing, and performance testing are test types that can be done using cloud testing, while independent, self-contained and programmatically accessible interface suitable for automated testing are features for the applications that can be tested using cloud testing[6].

The usage of cloud testing is now growing, as many organizations are now moving to testing over the cloud [7]. Currently, 28% of testing now occurs in a cloud environment, while 78% of organizations are planning to use Cloud testing [8].

In order to find the impact of cloud over software testing, Riungu et al. [9] conducted interviews with industry practitioners from eleven software organizations. They found that cloud testing, as to be used, needs some pilot projects that can be used by others, and more research to be conducted to make it clearer and solve existing issues.

On the other hand, testing cloud application depends on the service delivery model that is used for this application. Subashini at al. [10] discussed the security issues related to each type of delivery method (IaaS, PaaS and SaaS), and concluded that each delivery method has security concerns unique from the other.

B. Cloud Testing Model

Test as a Service (TaaS) is one of the models for cloud testing. A prototype for TaaS over cloud was developed by Yu et al. [11] based on 5 layers. These layers are: 1) test tenant and test service contributor layer, 2) test task management layer, 3) test resource management layer, 4) test layer (and it has three components: testing service composition, testing service pool, and testing task reduce), and 5) test database layer. They defined test tasks into Test Management Task, Modeling Task, Clustering task, schedule task and fault injection task. They found that this model helps tester to setup unit testing environment and implement unit testing.

LaBarge1 et al. [12] implemented penetration testing over a cloud management software package named OpenStack Essex Cloud Management Software. It includes a module that is responsible for managing the cloud starting from virtual machines until the storage. It is important to test the cloud management systems, as to block all possible ways which the attacker will try to gain access to the cloud management system, which in turn allows them to get access for client virtual machines. They built a model to test OpenStack and also implemented the actual testing. Their findings recommended to use secured protocols such as HTTPS to connect with cloud management software package such as OpenStack.

In terms of forms of attack, Web application attacks are still valid on cloud based applications, but the solutions for these vulnerabilities are different when using the cloud due to the nature and complexity of the cloud. For example, by opening too many ports in the cloud, cloud providers may unintentionally make the cloud open for more attacks. Also patches applied may inherently cause security harms as some VMs may not be affected by these patches (that add more security to the cloud) as the previous patch may still be cached and needs time to be cleared; another issue is using too many web services, which may increase the xml attacks. To secure the cloud, it needs to secure each cloud component as below in Figure 2. Among the approaches to secure these cloud components from attacks is penetration testing, which can be used to test all the components below [13].

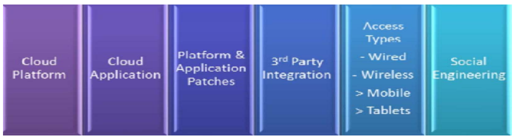

Figure 2. Cloud components

III. MOBILE TESTING

A key feature of mobile application is constant updating. This alone makes it becomes more important to apply rigorous testing to ensure these mobile applications are secure, reliable and consistent. However, testing mobile applications becomes much harder as these applications continue to be more complex over time [14].

However, the basic and main feature of mobile computing is allowing users to access data using mobile devices anywhere and from any wireless network available at any time. To provide this feature, mobile cloud applications needs to be adaptive and responsive to the ever-changing mobile computing architecture and its limitations, such as adaptive to different network, resources, power and architecture [15].

A. Mobile Testing Features

Diverse mobile platform, limitations in simulators, mobile device hardware boundaries, multiple network type and configurations, and rapid application development all make the testing of mobile application a challenging, time consuming and expensive process. [16] proposed a testing model based on cloud to test mobile applications, in order to overcome these limitations by creating multi-mobile platforms for testing purposes over the cloud. This model combines three types of services, mobile testing software as a service, mobile platform as a service and mobile infrastructure as a service. By using these services, testers can emulate a mobile device configuration. Integrating this model with tools to automate testing process allows testing the entire applications in single command.

Applications for mobiles integrate the vulnerabilities of both native and web applications; this makes it open for Cross Site Scripting XSS and Cross Site Request Forgery XSRF beside buffer overflow. Also, these mobile devices introduce more vulnerability such as tap jacking, smudge attack, key stroke caching and automated snapshots [17].

Another source for malicious attacks is the usage of mobile devices for site browsing, which may result in the users not aware of the security level as when they use the browser in full screen mode, which will not allow them to see the security tap in the browser. Malicious URLs for mobile sourced from many different sources other than site browsing, such as crafted URLs from HTML Links, SMS, Whatsapps or even QR Code [18].

B. Mobile Testing Models

Liu et al. [19] figured the need to have a specific model to test mobile applications, keeping in mind the need to make this test reusable and scalable. They designed a testing framework using SOA to test mobile application which focuses on testing applications efficiently. Their methodology is based on (1) Services component division and design (2) Service component interface design. Starting by analyzing the application and defining its functionalities as services, service provider and service registration, they then built the interface that would call the services and run the tests. Their results showed that the proposed mobile testing framework is effective and can be reused.

To cope with the ever-growing complexity of mobile applications, Ridene and Barbier [20] proposed in their model by using a Domain-Specific Modeling Language (DSML) to automate mobile application testing. In this model, a novel Mobile Applications Testing Language (MATeL) had been used to describe scenarios, checkpoints, loops, messages, and sequences. This language is integrated with a tool suite to implement its commands and to automatically run the scenario over mobile applications. Thus, making testing for mobile application easier, visible to be extended and adapted to any other technology.

To apply penetration testing over mobile, Paananen [21] proposed a framework to be the first open source solution framework to apply penetration test over smartphone. They have been motivated as no open source solution has been proposed to apply penetration testing over smartphones. The framework consists of a management console, web based management graphical user interface, management apps, and platform specific payloads or agents. The management console, web based management graphical user interface and management app will be used to launch the attacks while the agents will be responsible for performing commands they got from SMS or HTTP. This would be an initial framework that opens the door for other testers and developers to enhance.

In contrast, Liu et al. [22] proposed the Adaptive Random Testing (ART) model to test mobile apps using the monkey test black box technique in order to overcome limitations of other models that require prior knowledge of the system like record-replay technique or that needs writing and running scripts. Adaptive Random Testing is the model used to randomly generate test cases from event pools and then select the sequence of events from the same selected pool, then execute test cases. ART technique as per experiments showed that it can reduce the number of test cases and time required to expose first failure if compared with other random techniques.

IV. Mobile Cloud Testing

Mobile cloud inherits the issues and characteristics of both cloud and mobile, besides the newly-added issues incorporated because of offloading the processing of applications to cloud, as each process can be run on mobile or offloaded to cloud [23].

A. Mobile Cloud Computing Testing

Chun et al. [24] studied the constraints when developing mobile cloud based application, starting from defining the machine feature needed for each process as some process require access to some machine features such as GPS, states and flags used by a process. So these must be located on the same machine and cyclic prevention as if a machine fail the other machine should know it fails and not requesting the task again. This is beside the fact that profiles and logging tasks are complex [24].

B. Mobile Cloud Testing Issues

“What makes testing Mobile Cloud App unique?” is an important question answered separately by [25, 26, 27, and 28]. Summarized below are the points found in these works:

1) Must test for both mobile and cloud vulnerabilities. [25]
2) -“Keep moving” means keep changing environment factors.[25]
3) If something goes wrong, it is very problematic for a tester to determine exactly where the fault occurred and the reason for the fault to occur.[25]
4) Some parameters in the server change, server software is upgraded, or a new version of the software is placed in the telephone.[26]
5) Unlike desktop browsers, custom security plugins and extensions are not available in mobile browsers yet.[26]
6) Error reporting should go beyond reporting standard and screenshots of the error, to provide a better insight on what is the actual issue and all these must be defined in the test plan.[27]
7) To have independent middleware architecture for mobile devices. Additionally, the middleware will be similar concept of the middleware in web-based development but must be able to support different devices.[27]
8) Mobile browsers are more prone to website vulnerabilities because malicious messages come from more sources. The sources are not limited to emails and instant messages but also come from SMS, social messengers, in-app redirect and even QR Codes.[28]
9) Several types of context event, such as: [28]
   - Events coming from the external environment and sensed by device sensors (such as

temperature, pressure, GPS, geomagnetic field sensor, etc.);

- Events generated by the device hardware platform (such as battery and other external peripheral port, like USB, headphone, network receiver/sender, etc.).
- Events typical of mobile phones (such as the arrival of a phone call or a SMS message).
- Events like the arrival of an e-mail or social networks notifications that are related to the fact that modern mobile phones are more and more Internet connected.

Thus, as Mobile Cloud Application is distinct, then testing these applications should take special care and cover all factors.

C. Mobile Cloud Testing Model

How these differences will affect testing models? Based on mobile OS MCC testing model will choose which case to generate (Test Case Generator), with multiple versions of test cases in order to cover all versions of OS and mobile types. Also, each test case would be injected in Mobile and in Cloud, alongside with intended environment parameters. Mobile Agent would read environmental variable from mobile and mobile network to simulate intended behavior of real user. Mobile agent also will simulate changing and moving. Mobile agent would record all parameters. During this the Test manager would record all parameters. Beside the fact that input from different types of input resources would be used in test cases (Test Case Generator)

Priyanka et al. [29] in their survey found that 82 researches about Test as Service and 38 researches among them are about cloud based testing. From these 38 researches, there are only two researches about mobile cloud based testing and none of these two researches is about mobile cloud-based applications testing. Meanwhile Incki et al. [30] showed there are only three researches on mobile testing over cloud, and none of these provide a model for testing mobile cloud application. Indeed, based on our literature review, no such model could be found for such purposes.

Instead some models were found for using cloud services to test mobile applications, such as [16] who proposed a testing model based on cloud for testing mobile applications to overcome these limitations by creating multi-mobile platform for testing purposes over the cloud. This model combines three type of services, mobile testing software as a service, mobile platform as a service and mobile infrastructure as a service; by using these services, tester can emulate a mobile device configuration. Integrating this model with tools to automate testing process allows testing the entire application in single command [16].

On the other hand, [31] proposed a model to detect functional and performance issues by mining load testing recorded log using cloud services. This model based on using cloud service for abstracting the log record into categories and examines this log record against performance and functional model and then automatically evaluates it. Thus it would help to reduce complexity but categorizing log records and allow automating log evaluation process.

V. CONCLUSIONS

In this paper, the current researches in the fields of testing cloud, mobile, and MCC applications are presented. Cloud testing alternatively provides testers with services that enhance testing process using multiple types of testing models. Although there are several mobile testing models to make it easier and less expensive to test mobile applications, mobile applications have introduced many difficulties for testers.

Accordingly, MCC testing will face not only more significant problems but also unique problems of its very own paradigm, as it is more complex than either cloud computing and mobile computing; unless specific models and tools for testing MCC applications are created in parallel with the enhancements on mobile cloud computing applications. Indeed there is no known MCC testing model to cater for its specific needs.

REFERENCES

[1] Chun, Byung-Gon, Sunghwan Ihm, Petros Maniatis, and Mayur Naik, “Clonecloud: boosting mobile device applications through cloud clone execution.” arXiv preprint arXiv:1009.3088, 2010.

[2] Babcock., “Cloud's Thorniest Question: Does It Pay Off? - Hardware - Data.”, Retrieved from http://www.informationweek.com/hardware/data-centers/clouds-thorniest-question-does-it-pay-of/240001236?pgno=3, 2013.

[3] Sengupta, S., Kaulgud, V., & Sharma, V. S., “Cloud Computing Security--Trends and Research Directions.”, Services (SERVICES), 2011 IEEE World Congress, pp. 524-531, 2011.

[4] Jun, Wang, and Fanpeng Meng, “Software testing based on cloud computing.”, Internet Computing & Information Services (ICICIS), 2011 International Conference on. IEEE, 2011.

[5] Pawlish, Michael, Aparna S. Varde, and Stefan A. Robila., "Cloud computing for environment-friendly data centers." Proceedings of the fourth international workshop on Cloud data management. ACM, 2012.

[6] Parveen, Tauhida, and Scott Tilley, “When to Migrate Software Testing to the Cloud?.” Software Testing, Verification, and Validation Workshops (ICSTW), 2010 Third International Conference on. IEEE, 2010.

[7] Intel., “Cloud Computing Fun Facts.” Retrieved from http://download.intel.com/newsroom/kits/istcs/pdfs/CloudFunFacts.pdf, 2011.

[8] Murat Aksu and Charlie Li., “World Quality Report-Mobile Testing: Behind the curve”, http://www.uk.sogeti.com/Documents/2012-13_WQR.PDF, 2012.

[9] Riungu, Leah Muthoni, Ossi Taipale, and Kari Smolander, “Research issues for software testing in the cloud.” Cloud Computing Technology and Science (CloudCom), IEEE, 2010.

[10] Subashini, S., and V. Kavitha. “A survey on security issues in service delivery models of cloud computing.” Journal of Network and Computer Applications, 2011.

[11] Yu, Lian, “Testing as a Service over Cloud.”, Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on. IEEE, 2010.

[12] LaBarge, Ralph, and Thomas McGuire. “Cloud Penetration Testing.” arXiv preprint arXiv:1301.1912, 2013.

[13] Karnad, Kiran, and Saravanan Nagenthram, “Cloud security: Can the cloud be secured?.”, Internet Technology and Secured Transactions, 2012 International Conference for IEEE.pp. 208-210. IEEE, 2012.

[14] Ridene, Y., & Barbier, F., “A model-driven approach for automating mobile applications testing.”, Proceedings of the 5th European Conference on Software Architecture: Companion Volume (p. 9), ACM, 2011.

[15] Al-Bar, Adnan, and Ian Wakeman, “A survey of adaptive applications in mobile computing.”, Distributed Computing Systems Workshop, 2001 International Conference on, pp. 246-251. IEEE, 2001.

[16] Baride, Srikanth, et al. “A cloud based software testing paradigm for mobile applications.”, ACM SIGSOFT Software Engineering Notes 36.3 : 1-4, 2011.

[17] MCAFEE,”Penetration testing of iPhone/iPad applications”, http://www.mcafee.com/us/resources/whitepapers/foundstone/wp-pen-testing-iphone-ipad-apps.pdf, 2013.

[18] Sans Org, “Use offense to inform defense. Find flaws before the bad guys do.” http://pen-testing.sans.org/resources/papers/gwapt/robotstxt-108867,2013.

[19] Liu, Zhi-fang, Bin Liu, and Xiao-peng Gao. “SOA based mobile application software test framework.”, Reliability, Maintainability and Safety, 2009. ICRMS 2009. 8th International Conference on, pp. 765-769. IEEE, 2009.

[20] Ridene, Y., & Barbier, F. , “A model-driven approach for automating mobile applications testing.”, Proceedings of the 5th European Conference on Software Architecture: Companion Volume (p. 9). ACM, 2011.

[21] Paananen, Timo., “Smartphone Cross-Platform Frameworks: A case study.”, publications.theseus.fi, 2011.

[22] Liu, Zhifang, Xiaopeng Gao, and Xiang Long, “Adaptive random testing of mobile application.” Computer Engineering and Technology (ICCET), 2010 2nd International Conference on, vol. 2, pp. V2-297. IEEE, 2010.

[23] Costea, C., “Applications and Trends in Mobile Cloud Computing.”, Journal of Electronic and Computer Engineering, 2012.

[24] Chun, Byung-Gon, Sunghwan Ihm, Petros Maniatis, and Mayur Naik. “Clonecloud: boosting mobile device applications through cloud clone execution.” arXiv preprint arXiv:1009.3088, 2010.

[25] Muccini, Henry, Antonio Di Francesco, and Patrizio Esposito. “Software testing of mobile applications: Challenges and future research directions.”, Automation of Software Test (AST), 2012 7th International Workshop, pp. 29-35. IEEE, 2012.

[26] Kirubakaran, B., and V. Karthikeyani, “Mobile application testing—Challenges and solution approach through automation.”, Pattern Recognition, Informatics and Medical Engineering (PRIME), 2013 International Conference, pp. 79-84, IEEE, 2013.

[27] Ping, Tan Ping, Hamizan Sharbini, and Wee Bui Lin. “DESIGNING A MOBILE APPLICATION TESTING MODEL.”, The International Conference on Computing, Networking and Digital Technologies (ICCNDT2012), pp. 255-260, 2012.

[28] Amalfitano, Domenico, Anna Rita Fasolino, Porfirio Tramontana, and Nicola Amatucci, “Considering Context Events in Event-Based Testing of Mobile Applications.”, Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth International Conference, pp. 126-133. IEEE, 2013.

[29] Chana, Inderveer, and Ajay Rana, “Empirical evaluation of cloud-based testing techniques: a systematic review.”, ACM SIGSOFT Software Engineering Notes 37.3, 2012.

[30] Inçki, Koray, Ismail Ari, and Hasan Sozer, “A Survey of Software Testing in the Cloud.”, Software Security and Reliability Companion (SERE-C), 2012 IEEE Sixth International Conference, IEEE, 2012.

[31] Jiang, Z. M., “Automated analysis of load testing results.”, Proceedings of the 19th international symposium on Software testing and analysis, ACM, 2010.

ACKNOWLEDMENT

This work is partially funded by the Malaysian Ministry of Education RAGS research fund 600-RMI/RAGS 5/3 (86/2012)