Online Testing of Web-Based Applications

Qianxiang WANG\(^1\), Lining QUAN\(^2\), Fuchen YING\(^1\)

\(^1\) Institute of Software, Peking University, Beijing, China, 100871
\(^2\) Computing Center, Peking University, Beijing, China, 100871
Email: {wqx, ccqin, yingfch} @pku.edu.cn

Abstract

Testing of web-based applications is difficult. Online testing of web-based applications is more difficult. This paper argued the motivation of online testing, especially those of web-based applications. We explained what online testing is, what it concerns, and the difference between online testing and offline testing. We also introduced some key problems, such as how to reduce the interference to the normal services of system, how to improve the efficiency of online testing, etc. In the end, current research works including our primary research work were introduced.

1. Motivation

Nowadays, web-based applications are still in their repaid inflating phase, compared with other kinds of applications. Thus the research works about web-based applications are becoming very popular, such as web-based software development, web-based software maintenance, and web-based software (application) testing, etc.

From Lehman’s view, web-based applications belong to E-type software, which implement real world computer applications, and must be continually evolved to maintain user satisfaction, generally over several years and releases [1]. The following features of web-based applications make the online testing become one of the key technologies to web-based applications.

(1) Evolution. As one typical information system, it is obvious that, the web-based applications has feature of evolution. New technologies are invented continually, new requirement are being proposed, consuming mode is changing, and service flow is under continuously optimizing. This continuously changing environment requires service systems should have ability of durative evolution, such as, increasing service content, adjusting service flow, clearing bugs, and reducing response time, etc. To ensure the correctness of evolution, testing is necessary.

(2) Uninterrupted. Users always have strong interesting to 7X24 style service modes, just like traditional telecommunication services. That means, the services should keep available 7 days every week, and 24 hours every day. That means web-based applications are not only under durative evolution, but should also keep alive when evolving. So, testing of these systems should also be online.

(3) Integration. With the development of web-based applications, these systems are not only being used as information publishing portal, but also being used as business operating portal. These systems provide sufficient operating interfaces, not only let users can access them, but also let other web-based applications can access them also. Those interfaces make multiple web-based applications cooperate together, to provide one-step service to users. Web Services is the representation of those systems. This kind of inter-operating between services embodies the reuse to existing services. Some research works indicate that reuse oriented systems need more and strong testing than other systems [5]. For example, many Web Services have been published today, but some of them do not provide promised services. What’s more, no matter which kind of integrations, it is prone to import new defect to systems. This situation makes testing of web-based applications one very important issue.

Based on the result of online testing, we can compare the testing result with the promises of service provider; and compare the result with related standards, policies, and statutes, to obtain some useful evaluation of those services. To service providers, this result can help themselves to analysis potential defects in system, and improve the QoS of the system. To third-part testing or evaluating organizations, this result can be published, and help potential users select appropriate services with high quality, or provide improvement advices or warning to service providers.

2. What is online testing

As one of the key methods to ensure quality of system, testing gained researchers focus for long time,
and achieved distinct progresses [2]. Most testing works are toward products that offline. Although those kinds of testing are main approached to discover defects in systems, those approaches have some shortages: when doing test work, systems are running in some simulated environment, not real running environment, and data used in testing are simulating data also, not real data. Because of the inherent complexity of web-based applications, many defects can not been exposed under offline cases, but only under online cases.

In fact, as one key method of ensuring quality of system, testing is needed during the whole life cycle of system. When software is delivered, testing is transferred from software engineers to users. Some people even think each using of system is one testing to the system [3]. In recent years, service has been thought as the goal of software. "Software as service" is becoming one important common sense. And focuses on QoS (quality of services) are strengthened. As a result, focuses on online testing emerged and flourish in many areas[4,5,6,7].

Online testing is one kind of testing that is taken when the systems were put in use.

All features of web-based applications listed above make online testing not only necessary, but also frequent, that means, periodic, or driven by event, such as, user reports.

In theory, nearly all types of systemic testing (performance, pressure, fault tolerant, etc.) can be done in running time. But different with traditional offline testing, which tried to activate all defects in systems, online testing should be very careful, to not activate some dangerous bugs. Some testing can only be taken offline. For example, reliability testing should not take online, so as to avoid contrived system failure. Pressure testing should not be taken online too, to avoid unnecessary computing resource wasting. For online testing, its advantage can be revealed in following types of testing: Does service systems under normal states? Does the service is compatible with promised specification? Does service system strong enough to withstand the evil intruding? And so on. These kinds of testing are strongly related with the running context, which is not easy to simulate.

(1) State monitoring

States monitoring is the foundation of online testing. It is the basic monitoring of application systems. By testing systems periodically (usually in minute), we can make sure in which state the system is in. We can also separate some of system states into low level fine-grained states, to grasp the system in detail.

For inner system administrators, states monitoring result can be sent automatically to administrators in time. While system is fund in abnormal state, it can be checked and maintained quickly, so as to let the applications return to normal states.

For third-part tester, states monitoring can be used to help analysis distribution of faulting, such as time distribution, domain distribution, zone distribution, and ripple effect, etc.

(2) Functional testing

Functional testing is the core content of online testing. Because we are in early researching phase now, thus, to make testing simple, it is appropriate to focus on services that have explicit interfaces, which have formal specification, and easy to compare the testing result with the promised functions. Web services are ideal candidates, which have formal specifications of their services.

For service interface, every operation, attribute, and most constrains can be tested. This is one kind of surface level testing, just like black-box testing. But because tester can gain some design information from service provider, so they can carry through some deeper testing, just like grey-box testing, to enhance veracity of testing.

The detailed contents of functional testing should includes

1) Correctness testing. From technical view, correctness of online testing is similar with traditional testing method. For example, generating high efficiency testing cases, focused on boundary testing, etc. But, online testing is carried out under the true environment, so, when testing ended, if the testing operations will store some persistent data to system, those data should be specially treated, to keep the correctness of system states.

2) Integrality testing. This kind of testing want answer the following question: do service systems have implemented all promised function? If some functions have not been implemented, how about the integrality degree of the system?

3) Consistency testing. In this paper, we consider consistency as system’s consistent degree with popular protocol and standards, such as CORBA, J2EE.

To improve efficiency of online testing, testing should focus on the following interfaces: evolved interfaces, newly added interfaces, user frequently accessed interfaces and those interfaces that have defects in user’s reports, and so on. Considering the continually evolving feature of web-based applications, online testing will bring forth strong periodic feature. So, to reuse typical testing cases, similarities between testing cases should be carefully analyzed. For inner system administrators, other traditional testing methods such as, recursive testing, unit testing can be used also for online testing.
Other advanced research topics of online testing include self-testing, build-in testing, etc., which have been very popular in hardware online testing. For example, when size grew and many functions were integrated on a chip, it is difficult for integrated circuits to assure the reliability. The solution for large chips was BIST (built in self-testing).

3. Problems and potential solutions

Although we have limited the research scope of online testing, there are still many challenges that this new emerging topic must face with.

1) How to reduce the interference to the normal services of system. Testing of one running system will inevitably interfere in its normal running state. It is true in the physical world, and it is true also in the running logical world. The negative effects in performance and response time are obvious. But they are not severe. For functional testing, some testing operations will modify some persistent data of system, which will threat the consistency of the systems. If online testing can not solve this problem, servicing systems will refuse online testing, just like their hostile attitude to online monitoring and testing of security wholes. One of those potential solutions is to separate the testing requests from the normal requests in some sensitive executing steps, which can be implemented by expanding some of current interoperating protocols.

2) How to improve the efficiency of online testing. Online testing is one important complement to traditional offline testing. But constrained by the condition that the system should keep running, the testing ability is strictly limited, such as testing scope, testing granularity, testing time, etc. One of the potential solutions is making use of statistical method. By accumulating enough history data, including online and offline test cases, and plenty of user reports, we can get more accurate testing results.

3) Service implementation technologies that support online testing. Well designed system, especially the well-organized structure, outputting of some important inner states, etc, all of these considerations provide great convenience to testing, especially to online testing. Supports to web-based applications online monitoring and testing can come from application software, from system software (especially, middleware), and more possible, from these two kinds software together.

1) Support from software component design. Although the main purpose of software component is for reuse, in recent years, with the emerging of implementation-oriented component specification, such as J2EE, COM, support to software maintenance and evolution are impressive. So, when designing software component, we should consider not only reuse, but also online monitoring and testing.

2) Support from bottom level common services. For example, how to make use of reserved space in communication protocols? How to make use of implementation technologies of data access technologies, to separate data generated by online testing from data generated by normal operation, so as to reduce interfere of online testing to services of running system?

3) Support from log information. Although online testing can be periodical, the number of testing should be limited. So we must make good use of large number of log information, which record interaction information of normal operation (can be considered as passive testing gained data), and then design online testing cases, to analyze the testing results.

4. Current research works

Nowadays, there are some projects dedicated to monitoring and testing web-based applications.

Works in [8] considered a mission-critical subscription-based remote website, web server and web transaction monitoring service, providing accessibility, availability and performance monitoring with real time alert and reporting services for Internet-connected Web Systems. To monitor network services of high availability and quality, networks offering services of high availability and quality need to be carefully monitored. In order to decentralize static centralized systems, previous research has resulted in computationally expensive algorithms that require a global centralized network view. Works in [13] propose a much simpler distributed algorithm and show that it performs as well as existing near-optimal but expensive, centralized algorithms. Works in [14] introduced one technique called system status evaluation that the humanoid uses to detect and localize failures, and then help humanoid robot to be accepted in society and perform as an intelligent human assistant or companion.

For web-based application, the correctness of services provided by each component should be the most important thing. And the integrality and consistency ensure the entire system to work properly. As faults can occur anywhere in a distributed system, online validation addresses the activity of evaluating the behavior of services and service environments in the production phase in order to determine functional faults immediately. In [12], the authors designed and implemented an online validation platform, and
describe a generic architecture, which can be instantiated for implementing validation systems for any type of distributed environments. They also present their experiences in using TTCN-3 to implement an online validation system for active network environments.

The use of components in running systems produces a lot of useful information about the components' behavior that could be used to check for the compatibility between different components and between components and embedding systems. Paper [15] proposed a new technique, called Behavior Capture and Test (BCT), which takes advantage of this information to automatically reveal incongruence of components that either replace existing ones or are added to existing systems to extend their functionality. Their technology based on the behavioral invariants that represent the interaction of the component with the system.

In order to perform testing in production field without interfering to the network, paper [11] introduced the passive testing. The passive testing checks the protocol implementation through online observation. This test method was realized in an OnLine Test System (OLTS).

When online evolution are implemented in our PKUAS [6], we began considering the online testing, which can be used to verify whether the evolved software is correct. Just like researching on online evolution, we referenced online testing method that taken by hardware systems, such as network hardware, traditional telecommunication system, etc. As to the testing objects, we select Web Services as the first testing targets. Web services gained many researchers focuses, and the specification of web services are well defined, which make testing and evaluation easier. Although till now the number of published web services is limited, many of those services have been found that they can not provide promised functions. For system software that supporting online testing, application servers that compatible with J2EE specification are considered mainly. Recently, we are cooperating with China Software Testing Center, to test and evaluate some of network based services that supporting Digital Beijing and Digital Olympic.

Acknowledgement: This effort is sponsored by the National High-Tech Research and Development Plan of China under Grant No. 2003AA115430, 2001AA113060, the National Grand Fundamental Research 973 Program of China under Grant No. 2002CB31200003, and the National Natural Science Foundation of China under Grant No. 60233010, 60125206.

5. References