Leveraging P2P Systems to address the Test Scenario Explosion Problem

Andrea Mangion[1], Sebastian Attard[2], Mark Micallef[1], and Conrad Attard[2]

[1] Department of Computer Science, University of Malta
{andrea.mangion.04, mark.micallef}@um.edu.mt
[2] Department of Computer Information Systems, University of Malta
{sebastian.attard.13, conrad.attard}@um.edu.mt

1 Introduction

Modern software development is characterised by a strong customer focus and electronic delivery mechanisms which make it very easy for customers to buy and install a vendor's software. However, it also makes it very easy for customers to buy and install software from competing vendors and as such it is more important than ever for deployed software to be as correct and bug-free as possible.

Whilst certain types of testing can be done in the lab with a high degree of confidence that results will hold when the software is deployed in the wild, in reality software systems are subject to influence from whatever environment they end up being deployed in. Varying factors in customer environments can include operating systems, services packs, device drivers, network connectivity, resource usage by other software, and so on. Any variation or combination of these factors can lead to a situation where a system deviates from its expected behaviour. The problem is amplified even further on mobile devices whereby devices can move between different networks, interrupt apps for phone calls, have varying screen sizes, have user interference in the form of turning features on and off to preserve battery power, vendor-specific operating system code, and so on. A conservative calculation indicates that a software system can be subjected to tens of thousands of different scenarios. Even if one were to execute just one test case against each scenario, obtaining any form of realistic coverage is infeasible for even the most resource-rich organisations.

Cloud and grid infrastructures have been proposed as solutions to improving the scalability of software testing [1–3] since 2008. However, we argue that cloud computing systems are too homogenous and are not representative of real-world usage scenarios. We refer to this problem as the *Test Scenario Explosion Problem.*

2 Peer-to-Peer Systems

Peer-to-peer (P2P) systems are distributed systems which are usually associated with (1) a high degrees of decentralisation, (2) self-organisation and (3) an abundance and diversity of resources [4]. Although they are commonly associated with file sharing applications P2P systems haste been used to leverage the computing resources of cooperating users in order to achieve scalability and organic growth. These characteristics combined with its independence of a dedicated infrastructure and centralised control make P2P an interesting candidate paradigm for the solution of the test scenario explosion problem.

3 Hypothesis and Research Challenges

Our overarching hypothesis is that *P2P systems can be leveraged to achieve a high level of scenario coverage within a feasible amount of time.* That is to say that given a network of peers in which computing resources are made available, a developer can request execution of a test suite whilst placing desirable constraints (e.g. tests can executed by users in Germany who own Android devices). Peers will then propagate this request appropriately such that the workload is shared amongst peers who satisfy the constraints with the developer subsequently receiving appropriate test results.

Whilst the idea is arguably interesting and credible, a number of challenges present themselves. The following is an incomplete but representative list of research challenges which make the problem non-trivial and interesting from both an academic and commercial point of view:

1. How can the problem be effectively reformulated into a distributed search problem such that known P2P algorithms can be reused effectively?
2. What is the best way for the P2P network to self-organise such that test suites can be propagated efficiently?
3. Given a particular network population, can a guarantee be provided that a test suite will be executed with a certain level of scenario coverage within a certain amount of time?
4. What incentive mechanisms can be utilised in order to encourage participation in such a system?
5. What are the security implications of executable code being transferred and deployed between peers?
6. How do peers negotiate obligations, permissions and restrictions between themselves?
7. How does one go about evaluating the effectiveness of such systems?

Whilst we are currently working on developing prototypes for both desktop and mobile devices, we would like to use CSAW 2013 as a discussion platform on the evaluation aspect of this research. Consequently, whilst the talk will consist of discussions about various design decisions that go into building such systems, we will be presenting options with regards to evaluation and soliciting feedback on the topic.

References

1. Jerry Gao, Xiaoying Bai, and Wei-Tek Tsai. Cloud testing-issues, challenges, needs and practice. *Software Engineering: An International Journal*, 1(1):9–23, 2011.

2. H Kothandaraman. Testing applications n cloud. *Computer sciences CorporationLEF*, pages 1–45, 2011.
3. Leah Muthoni Riungu, Ossi Taipale, and Kari Smolander. Research issues for software testing in the cloud. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pages 557–564. IEEE, 2010.
4. Rodrigo Rodrigues and Peter Druschel. Peer-to-peer systems. *Communications of the ACM*, 53(10):72–82, 2010.