StarCloud: An Android Cloud Testing Framework

Jared Ravetch† and Kento Aida†‡

1. Introduction

The smartphone has revolutionized communication and the way humans interact with one another. Development and testing for the smartphone, specifically Google's Android[1] platform, is rife with challenge due to the inability to simulate how the application will behave in a distributed environment. As the Android OS has been regularly updated from version 1.5 (Cupcake)[2] to the current version 4.0 (Ice Cream Sandwich)[2], forces developers to test their applications on all iterations, which has no easy solution. Moreover, there is no easily accessible sandbox environment for application developers to test the security and performance implications of their application in the cloud. We present StarCloud, an Android testing framework that emulates a cloud of connected and distributed Android devices utilizing Amazon's Elastic Compute Cloud (EC2)[3] cloud infrastructure.

2. Architecture Overview

StarCloud acts as a middleware between users (or application developer) and EC2 that allows the user to launch a cloud of customized Amazon Machine Image (AMI); each instance running the Android OS utilizing the Dalvik Virtual Machine[4]. A security group is created to allow each device to communicate with one another seamlessly. A master instance is also launched, in order for the user to interact with the virtual Android device cloud to deploy applications, collect device logs and performance data.(**Fig.1**).

---

† Tokyo Institute of Technology
‡ The National Institute of Informatics
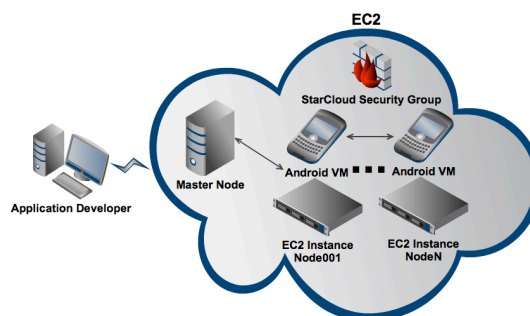
3. Implementation

Fig.1 StarCloud Architecture

EC2 was chosen as the cloud infrastructure platform due to its full-featured API and seamless ability to scale. EC2 has been utilized in a multitude of research and high performance computing projects[5].

The concept of StarCloud grew from the existing tool StarCluster[6], which gives anyone the ability to launch a cluster of EC2 instances, using AMIs specifically customized for scientific research. In order to run the Android OS in the cloud, we referred to the work done by Byung-Gon Chun et al. that created CloneCloud[7], which elastically offloads computation from the Android device to EC2. A snapshot of the running Android OS is used as a template to launch instances running the Android OS in EC2.

Peer-to-peer (P2P) applications are becoming more prevalent on mobile devices as a way to share data, location and stream media. StarCloud is the perfect test bed for Application developers that wish to test P2P applications in in the cloud[8]. Developers can utilize the SIP

protocol that has been ported to Android by the SIP2Peer middleware[9].

4. Conclusion

In this paper we present the novel StarCloud middleware to aid application developers in testing their applications in a virtual cloud of Android devices. Currently, work is focused on building a proof of concept with the various technologies mentioned. Future work will focus on improving the proof of concept for StarCloud to enable the user to launch customized EC2 instances with a random distribution of Android OSs[1], empowering the developer to deploy, test, and benchmark their applications in a more true to life scenario.

References

[1] Google Android. http://developer.android.com.

[2] Google Android Ice Cream Sandwich. http://www.android.com/about/ice-cream-sandwich/.

[3] Amazon Elastic Compute Cloud (EC2). http://aws.amazon.com/ec2/

[4] D. Bornstein. Dalvik Virtual Machine. http://www.dalvikvm.com/.

[5] High Performance Computing on AWS. http://aws.amazon.com/hpc-applications/.

[6] StarCluster. http://web.mit.edu/star/cluster/.

[7] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. 2011. CloneCloud: elastic execution between mobile device and cloud. In *Proceedings of the sixth conference on Computer systems.*

[8] Marco Ughetti, Tiziana Trucco, and Danilo Gotta. 2008. Development of Agent-Based, Peer-to-Peer Mobile Applications on ANDROID with JADE. In *Proceedings of the 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies.*

[9] PeerDroid. http://code.google.com/p/peerdroid/.