# Android Experimentation and Data Collection

Michael Griepentrog, Isaac Hall, Tyler Harter,
Raja Ram Yadhav Ramakrishnan

Email: {mgriepentrog, ichall, harter, rramakrishn2}@wisc.edu

## 1 Introduction

Use of mobile devices, such as Android phones, is becoming increasingly common. Unfortunately, these devices have resource constraints that are much more restrictive than those on typical workstations or servers. Finding ways to get more mileage out of limited memory, processor, storage, wireless, and power resources could significantly impact a large population of Android users.

One approach to circumventing local resource constraints that has been considered is to run some Android processes remotely in the cloud. Not only would the processes be able to run on machines with hardware that is much more powerful than than that of an Android phone, but remote resources could be shared among users. Although this approach seems promising, there are still many questions that need to be answered. For instance, which resources would the cloud best supplement? Are these resources that are consumed by the system itself, or are they resources consumed by processes that could potentially be run remotely? Which processes would an ideal scheduler run locally, and which ones would it run in the cloud?

To answer these questions, one must understand the characteristics and behavior of common Android applications. This is the purpose of our research. We have two goals: create an experimentation framework that allows other researchers to automate application runs, and create a tracing framework that allows us to gather detailed traces of program runs as well as snapshots of resource usage.

## 2 Related Work

The only related articles pertained to general topics of what we are interested in investigating. Because the Android is a relatively new platform, there are few publications pertaining to it.

- Hao, F., Lakshman, T. V., Mukherjee, S., and Song, H. 2009. Enhancing dynamic cloud-based services using network virtualization. In *VISA '09: Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures.*

- Behavioral Detection of Malware on Mobile Handsets. In *Proceeding of the 6th international conference on Mobile systems, applications, and services*, 2008.

# 3 Approach

We plan to use Android 1.6 as the target platform of our research. Android 1.6 was officially released just this last week and may help us accomplish some of our objectives more readily than the 1.5 version. For instance, Mike has already flashed his Android with the 1.6 firmware and discovered that the new version of the system includes utilities that break down power usage into consumption by application.

Initially, we plan to collect data with a Java application that runs in user space and collects basic data about memory and process utilization from the Unix procfs filesystem. To get detailed information about wireless resource utilization, however, we plan to eventually instrument the Android kernel, which would require inserting additional C code. A couple of our team members have had some experience with kernel programming, but none of us have any familiarity with Android internals, so so investigative work will need to be done. Recording and replaying user actions will also likely involve additions to the operating system code, but we are still investigating this. Since the experimentation framework will not need to run on actual devices, another option might be to modify the Android emulator if it is open source.

# 4 Milestones

| ***Date*** | ***Milestones*** |
|---|---|
| Oct 5 | • Write a program that installs a bunch of .apks and starts them<br>• Get process memory and cpu usage snapshots from procfs |
| Oct 12 | • Compile and run Android from source<br>• Get startup stats for 30-40 Android Marketplace apps |
| Oct 19 | • Log keystrokes<br>• Trace solid state storage IO |
| Oct 26 | • Log touch screen actions<br>• Trace WIFI traffic |
| Nov 2 | • Automate keystrokes<br>• Trace Bluetooth traffic |
| Nov 9 | • Automate touch screen actions<br>• Trace cellular traffic |
| Nov 16 | • Write Android daemon that periodically sends process resource snapshot data to a central database server |
| Nov 23 | • Thanksgiving |
| Nov 30 | • Do analysis of collected data: generate stats and plots |

## 5 Deliverables

- Tracing Framework
  - Android user-space application that gathers snapshots of memory/processor usage for running applications.
  - OS additions that log various events to generate traces
  - Statistical data and graphs extracted from traces we collect that characterize 30-40 popular Android applications that we will select.
- Experimentation Framework
  - Tool that launches a given application on an Android device/emulator.
  - A tool that will record user actions (touches/key strokes)
  - A tool that executes recorded or otherwise specified user actions (touchs/key strokes).