

# A Tool for Monitoring User Activity on Palm OS Devices

A Thesis Proposal  
presented to the  
Department of Computer Science  
Brigham Young University

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science

by  
Hyrum D. Carroll  
September 2003

## 1 Introduction

To improve the performance of handheld devices, in terms of speed and energy consumption, system designers need to be able to evaluate the consequences of various design alternatives. To do this quantitatively, the data used to evaluate the system must be representative of real workloads. Designers need to understand how real applications, run by actual users, under normal conditions, affect memory access patterns and instruction usage. A tool is needed to produce accurate and relevant data to aid designers and developers. Such a tool is not currently available in industry or academia.

There is a need for a tool that collects a normal user's activity and can replay that information to produce unperturbed, accurate data to characterize real workloads and make design decisions. Previous attempts to collect system activity for handheld devices have not been successful nor performed under normal operating conditions.

Researchers have collected information about the execution of workloads on handhelds for years. Most research in this area has focused on evaluating the energy consumption and efficiency of handheld devices [1, 2, 3, 4, 5]. Flinn and Satyanarayanan [1] collect information about handheld devices using oscilloscopes. External hardware monitors such as oscilloscopes, significantly perturb the system by restricting the user from operating the handheld normally. [2, 3, 4] use cycle-accurate simulators to estimate energy consumption. Simulators still need real input to produce accurate estimations. Cignetti, Komarov and Ellis' work [5] is unique in that they utilize a modified version of the Palm OS Emulator. They limited their studies to energy consumption estimates.

The most related published research to my thesis is Gannamaraju and Chandra's [6] and Rose and Flanagan's [7]. Gannamaraju and Chandra wrote a tool to record the advent of 80% (707 of the 880) of the Palm OS 3.5 system calls. Due to their collection techniques, they are inherently unable to generate a complete list of system activity or events (commonly referred to as a *trace*). Also, they suffer from a large amount of overhead and memory requirements. Using their technique, they generate 1.34 MB of records on the handheld for about one minute of execution. Given current memory limitations for Palm OS devices, only a few minutes of execution can be traced.

Rose and Flanagan are able to produce complete traces of normal workloads. They collect interrupts and replay them on an identical system. During playback, the addresses from the processor can be recorded. Although their methods have proven successful, they limited their study to desktop systems.

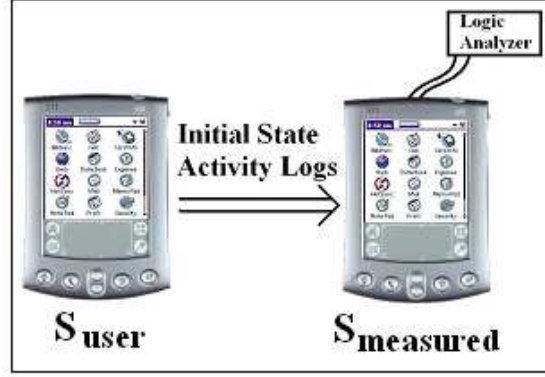


Figure 1:  $S_{user}$ , the system of a participant in the research pool, is used in normal operating conditions. Our tool collects the initial state and activity log on  $S_{user}$ . They are transferred to an equivalent system,  $S_{measured}$ , the system being measured, to be used for playback. Since  $S_{measured}$  has the same initial state as  $S_{user}$ , and is driven with the same inputs, its execution path and final state are also the same. By collecting data about the behavior of  $S_{measured}$ , we are collecting the same data from  $S_{user}$  without perturbation to normal execution.

This thesis focuses on a tool that overcomes previous deficiencies in collecting handheld system activity by recording the initial state and then recording external inputs. These inputs are stored in a database on the handheld under test. The additional execution time is minimal and the necessary memory for the records is realistic for the systems being tested. The collection adds an acceptable amount of overhead because it is imperceptible.

After the activity log is collected, it is transferred to a desktop machine. The log is parsed and prepared for playback by a software emulator running on the desktop. The emulator has been modified to convert an activity log back into inputs, driving the execution of the emulator. Furthermore, the emulator has been instrumented to record memory access patterns and instruction usage. Other statistics can also be recorded.

The fundamental design of the tool consists of two equivalent deterministic state machines. The first is the handheld under test or  $S_{user}$ . The second is the emulator that replays the generated activity logs or  $S_{measured}$ . Since the two state machines are equivalent, if they start in the same state and have the same external inputs applied, they will follow the same execution path. Hence, the emulator can be used to reproduce the workload executed on the handheld under test (see Figure 1).

## **2 Thesis Statement**

User's inputs to a handheld can be collected with imperceptible perturbation to the system and replayed on an emulator to characterize the workload executed by the user. Furthermore, the emulator can be instrumented to produce statistics about the workload previously executed on the handheld system.

## **3 Methods**

To demonstrate this claim, I will follow the procedures outlined in this section. First, I will instrument a few handhelds to record the users' input and distribute them to participants in a research pool. Next, I will modify an existing emulator to replay the events generated by the pool. Finally, a logic analyzer will be employed to validate the results obtained from the emulator.

### **3.1 Instrumenting Handhelds**

I will instrument a few (two to four) Palm m515 handheld computers running Palm OS 4.1 to record inputs from the user. To accomplish this, three different hacks will be written to record the inputs. Hacks are head patches to the normal system routine calls and will be managed on the handheld with a hack manager such as X-Master [8]. The hacks will record pen and button events, a timestamp and any pertinent data (e.g., x and y coordinates for pen events). The instrumented handhelds will be given to participants in the research pool to produce logs of normal activity. An emulator will replay the generated activity logs.

### **3.2 Modifying Palm OS Emulator**

Once the handhelds have been properly instrumented and activity logs of users' inputs are produced, I will modify the Palm OS Emulator (POSE) [9] to replay them. POSE is a widely used and accepted development tool in the Palm OS community [10]. POSE runs on Windows, Linux and Mac operating systems and accurately emulates Palm OS devices by fetching instructions from a ROM image or an allocated RAM. The instructions are executed cycle by cycle just as the physical handheld does. POSE will be modified to run as follows:

- 1) Load an activity log collected from a handheld device.
- 2) Simulate the next event in the log when the system's clock is greater or equal to the event's timestamp.

- 3) Collect statistical information (e.g., memory references, opcode usage, etc.) from the emulator.
- 4) Repeat steps 2 and 3 until all events have been replayed.

### **3.3 Validation**

Once the emulator accurately replays the collected activity logs, the data generated from the emulator will be validated with a physical handheld and a logic analyzer. Two handhelds will be setup identically. One of them will have a logic analyzer attached to it. A predetermined sequence of events will be performed on both of them. The activity log from the handheld not connected to the logic analyzer will be replayed by the emulator. Memory usage patterns generated during playback and the data collected by the logic analyzer will be compared. Their correspondence or disparity will serve as the final metric to evaluate my thesis statement.

## **4 Contribution to Computer Science**

This research will produce two main contributions to Computer Science. First, it will produce a tool capable of characterizing and analyzing workloads of real users for Palm OS devices. The data collected from this tool will be used to perform some realistic cache simulations for a family of devices that presently do not have caches. Opcode usage frequencies will be another byproduct of the data. Also, the tool will enhance debugging applications. Second, my work will serve as a proof-of-concept to other researchers on other architectures and operating systems.

## **5 Delimitations of the Thesis**

Although the principles and techniques applied in my thesis are applicable to other systems and devices, only the Palm m515 running Palm OS 4.1 will be used. Furthermore, the Palm OS Emulator will be considered to be accurate and reliable and so a hardware emulator is not needed. Also, the participants in the research pool will not be allowed to use expandable memory cards, the infrared and serial ports nor applications that use random numbers (e.g., games). Using of the memory card and the infrared and serial ports is restricted because accurate playback of activity logs with these events would necessitate storing the incoming data to the device. Storing the incoming data would greatly increase the overhead for these events. The restriction of using programs that require random numbers is

hoped to be temporary, until a reasonable solution to seeding the random number generator in the emulator can be developed.

## **6 Thesis Outline**

I will write the methods used and my results in the format of a Paper Thesis. The paper will begin with the usual title page, abstract and signatures pages. Directly following these, the remainder of the thesis will proceed as outlined below.

### **Chapter 1 - Introduction**

The first chapter will resemble an expanded version of the introduction found in this document.

#### **Related Work (and their shortcomings)**

A detailed description of the few previous attempts to record system activity will also be treated in this chapter. The researchers' methods, findings and shortcomings will be discussed here.

#### **Background**

Necessary background will be explained. Information about handhelds systems, the Palm operating system and HotSyncing will be treated first. An explanation of hacks and X-Master will follow. The Palm OS Emulator will be covered next. Finally, pertinent information about address buses will be detailed.

### **Chapter 2 - Collection of Activity Logs**

The different elements of my research will be discussed starting with the collection of the external inputs. Inputs commonly found in a handheld device and those tracked in the research pool will be discussed. Next, I will talk about how handhelds are initialized with hacks and X-Master. An explanation of each hack will follow.

### **Chapter 3 - Playback of Activity Logs**

The next chapter will address playback of collected activity logs. It will begin with a review of the Palm OS Emulator. An explanation of why POSE was chosen as an

appropriate emulator will follow. Finally, the major modifications made to the source code of POSE will be detailed.

#### **Chapter 4 - Validation of the Emulator's Results**

The employment of a logic analyzer to validate the emulator's results will be presented. The motivation, setup and methods used to trace a Dragonball processor's address bus will be explained. Results confirming or refuting the authenticity of the emulator's execution will be explained with words, charts and graphs.

#### **Chapter 5 - Playback Data**

The results from the data collected during playback will be explained and analyzed. This chapter will contain a number of graphs and charts illustrating typical histograms of memory usages for the different memory banks, and instruction usages frequencies. Also, a few representative simulations will be given for hypothetical caches.

#### **Chapter 6 - Conclusion and Future Work**

The last chapter of my thesis will reiterate the key points. It will review the motivation for the tool, the different components, the evaluation results and the thesis statement will be revisited. Furthermore, possible expansions and modifications to my research will be covered. The need for a hardware emulator, longer studies with larger research pools and the various analysis that can be performed from the data generated during playback will be detailed.

#### **References**

References cited in the thesis will appear in this section.

#### **Appendix**

Graphs and charts of the data collected from the emulator that is too large or distracting from the Playback Data chapter will be presented in the appendix.

#### **7 Thesis Schedule**

I have already designed most of the mentioned tool. I have instrumented two handhelds and selected a research pool to produce the activity logs. Most of the modifications

to the Palm OS Emulator have also been implemented. Further timing considerations during playback will be addressed in the month following my proposal presentation. Next, I will validate the emulator's results with a logic analyzer the following month. Finally, I will write my thesis and defend it in the first quarter of 2004.

## References

- [1] J. Flinn and M. Satyanarayanan. Powerscope: A tool for profiling the energy usage of mobile applications. In *Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 2–10, February 1999. Flinn and Satyanarayanan explain their tool, PowerScope, which “maps energy consumption to program structure” for mobile applications. Their tool identifies how much power a process or a procedure consumes.
- [2] Narayanan Vijaykrishnan, Mahmut T. Kandemir, Mary Jane Irwin, Hyun Suk Kim, and W. Ye. Energy-driven integrated hardware-software optimizations using SimplePower. In *ISCA*, pages 95–106, 2000. SimplePower is a cycle-accurate simulator for energy consumption estimates. Although it does not include Palm OS devices, it serves as an example of energy studies on mobile devices.
- [3] W. Ye, Narayanan Vijaykrishnan, Mahmut T. Kandemir, and Mary Jane Irwin. The design and use of simplepower: a cycle-accurate energy estimation tool. In *Design Automation Conference*, pages 340–345, 2000. See [2].
- [4] Sheayun Lee, Andreas Ermedahl, Sang Lyul Min, and Naehyuck Chang. An accurate instruction-level energy consumption model for embedded RISC processors. In *LCTES/OM*, pages 1–10, 2001.
- [5] T. L. Cignetti, K. Komarov, and C. S. Ellis. Energy estimation tools for the palm. In *ACM MSWiM 2000: Modeling, Analysis and Simulation of Wireless and Mobile Systems*, August 2000. Cignetti, Komarov, and Ellis explain their tool for estimating power consumption for Palm OS devices. They monitor the transitions between different power states to calculate the steady state power and transient energy levels. Their tool, like mine, is based on the Palm OS Emulator. They also address timing differences between the emulator and a physical device.
- [6] R. Gannamaraju and S. Chandra. Palmist: A tool to log palm system activity. In *Proceedings of IEEE 4th Annual Workshop on Workload Characterization WWC4*, pages

111–119, December 2001. Gannamaraju and Chandra utilize hacks to record the occurrence of 80% of the Palm OS 3.5 system calls. Their work is the most direct ancestor to my techniques on collecting activity logs.

- [7] C. Rose and K. Flanagan. Complete instruction traces from incomplete address traces (CITCAT). *Computer Architecture News*, 24(5):1–8, Dec 1996. The fundamental model used in my thesis for playback was inspired by Rose and Flanagan’s work as presented in CITCAT. They instrumented a R4000 system to notify a hardware monitor when to start tracing the primary memory bus. With each event a timestamp is also recorded. Also, an initial memory image is captured. The event log and initial state are then used by a simulator to produce long and accurate instruction traces.
- [8] LinkeSOFT. *X-Master 1.5 Free Extension (Hack) Manager*. <http://linkesoft.com/xmaster/>. X-Master and information about it are available from this website. X-Master is used in my research to manage the hacks. It is based on Edward Keyes’ HackMaster v0.9 [11].
- [9] PalmSource, Inc. *Palm OS Emulator*. <http://www.palmos.com/dev/tools/emulator/>. I modified the Palm OS Emulator, POSE, to replay activity logs and collect statistics such as memory access patterns and opcode usage frequencies.
- [10] PalmSource Inc. *Palm OS, Palm Powered Handhelds, and 18,000 Software Applications*. <http://www.palmsource.com>. This is the website and definitive source for information on PalmSource Inc., a Palm subsidiary responsible for Palm OS. Their operating system is the only one used in my research.
- [11] E. Keyes. *HackMaster v0.9*. <http://www.daggerware.net/hackmstr.htm>. Keyes is the inventor of hacks and author of the first hack manager. Information about hacks, including how to write one, is available from this website.
- [12] K. Rollin. The Palm OS Emulator. *Handheld Systems*, 6(3), May/June 1998. Rollin is the principal engineer supporting the Palm OS Emulator. In this article he explains PalmSource’s improvements made from Copilot and their intentions for the future of the emulator. Rollin also details how the emulator works.
- [13] Palm, Inc. *Palm Support: HotSync Technology Support Index*. <http://www.palm.com/us/support/hotsync.html>. Palm’s website explaining HotSync technology.

- [14] TealPoint Software, <http://www.tealpoint.com/softmstr.htm>. *TealMaster System Extensions Manager*. TealMaster is an alternative hack manager to HackMaster [11] and X-Master [8]. It is currently the only hack manager to support Palm OS 5.
- [15] G. Hewgill. Copilot: Design and development. *Handheld Systems*, 6(3), May/June 1998. Copilot is Hewgill's creation to emulate the Palm Pilot in Windows. The article is a chronological story of its development.
- [16] Free Software Foundation, <http://www.gnu.org/copyleft/gpl.html>. *GNU General Public License*. The Palm OS Emulator is distributed under the GNU General Public License. Information about the license can be found here.
- [17] C. Bey, E. Freeman, G. Hillerson, J. Ostrem, R. Rodriguez, and G. Wilson. *Palm OS Programmer's API Reference*. PalmSource, <http://www.palmos.com/dev/support/docs/>, July 2001. This is a reference manual for software programmers using Palm OS.

## 8 Artifacts

The following artifacts will be produced as a natural byproduct of researching my thesis statement:

- 1) A tool capable of characterizing and analyzing workloads of real users.
- 2) Data collected from the tool to be used in preliminary cache studies and opcodes usage analysis.

This thesis proposal by Hyrum D. Carroll is accepted in its present form by the Department of Computer Science of Brigham Young University as satisfying the thesis proposal requirement for the degree of Master of Science.

---

J. Kelly Flanagan, Committee Chair

---

Date

---

Charles D. Knutson, Committee Member

---

Date

---

Bryan Morse, Committee Member

---

Date

---

David W. Embley, Graduate Coordinator

---

Date