

Predicting Power Usage of Android Applications

Ben Bramble and Professor Michael M. Swift

Problem: Mobile devices have limited battery life and charging opportunities are not always available. The increasing complexity (Bluetooth, GPS, WIFI, multicore processors...) of these devices require a more inclusive approach to power modeling.

Solution: Using a software only approach, determine the impact of mobile device subcomponents, measure the performance of application, and combine heuristics with the measurements to predict future application impacts.

End Result: an educated user capable of extending the life of their device through informed decisions.

Step 1: Profiling

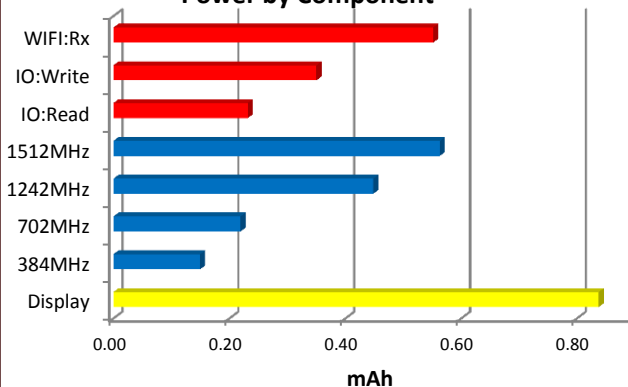
Current Baseline Test of Device Components

- CPU by Core/Frequency
- Display On/Off
- Wifi Bytes Received
- I/O Reads/Writes



- mAh drain per MB
- mAh drain per 1000 CPU Cycles
- mAh drain per 10 seconds

Power by Component



Step 2: App Polling

Determine Heuristics of Running Processes Over Time

- Modified Linux kernel to include counters per processes for CPU statistics by frequency and network bytes received
- Create background service to poll every 30 seconds to gather data on running applications
- Maintain database containing all process information

	Name	Angry Birds	Power Pro	Browser
WIFI (MB)	Rx	0.29	0	2.04
IO (MB)	Read	0	0	0.403
	Write	0.45	2.01	10.05
CPU (cycles)	384MHz	360	12	800
	702MHz	5	0	4016
	1242MHz	6912	212	8201
	1512MHz	36012	800	1001
Display (s)	Foreground Time	2000	0	1509

Future Work

- Extensions
 - Add components such as GPS, 3G, audio, and UI
 - Explore models without custom kernel or super user
- Applications
 - Allow user to dictate policy based off results i.e. contain power usage by application
 - Create visualizations to easily explain where the power is drained such as pie charts and graphs

Step 3: Projection

Apply Process Heuristics to Sub-Component Expressions to Determine:

- Projected Battery Drain per Minute of Using Specific Application
- Identify Power Inefficient Applications

Name	Time Active (s)	Display Cost (%*)	CPU Cost (%*)	IO Cost (%*)	Wifi Cost (%*)	Total cost (%*)
System Services	3786.05	0.000	0.126	0.001	0.142	0.268
Angry Birds Foreground	2168.274	4.598	3.712	0.039	0.058	8.407
Angry Birds Background	1617.776	0.000	0.465	0.001	0.000	0.465
Browser Background	2199.242	0.000	0.041	0.056	0.032	0.129
Browser Foreground	1586.808	3.365	0.128	0.085	0.014	3.592
Power Profiler	3786.05	0.000	0.058	0.078	0.000	0.136
Sum (%*)		7.963	4.529	0.259	0.246	12.998

* % Percent of total Battery Drained

Related Work

- Tools and Applications
 - Power Tutor: Power model for select Android phones based off lab component testing
 - Carat: Cloud-based collaborative effort to identify application power hogs
 - AppScope: Monitors kernel activities for hardware component requests and charges the appropriate application
 - Watts On: Windows 8 developer tool based off lab component testing