ECE 571 – Advanced Microprocessor-Based Design Lecture 25

Vince Weaver

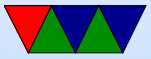
http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

28 April 2016

Announcements

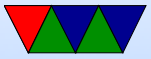
- Presentations next week
- Still looking for volunteer
- Power measurement, more coming in by Monday



Reading

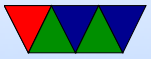
Where is the energy spent inside my app? Fine Grained Energy Accounting on Smartphones with Eprof

by Pathak, Hu, and Zhang



Intro

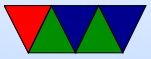
- Apps limited by battery life
- energy consumption is important
- 1. track at level of "program entities" (i.e. thread, routines, etc)
- 2. track various components
- 3. Need to map power used to the entities



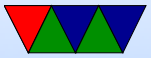
- cell phones do not have built-in power meters.

- asynchrnous power behavior: power draw might not be related to running app:

  1. Tail power (GPS, wifi, sdcard) (ref 4,5)
  2. persistant power wakelocks
     lock keeps phone from sleeping. can extend beyond a routine
  3. "exotic" components: camera and GPS start when switched on by one and continue until switched off (even if app switched in doesn't use it)

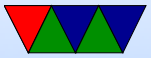


- eprof fine-grained energy profiler "first" for modern
- cellphones on Android and Windows Mobile
- Leverages fine-grained power modeling from [4]
- system call driven finite-state-machine?
- last-trigger policy?
- look at 6 of the most 10 popular apps in android market
- Surprising results:



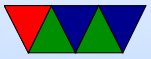
- ad modules consume 65-75%
- clearn termindation of TCP sockets 10-50% of energy
- tracking user data 20-30%
- Actual application only 20-30%
- Majority energy in 3G, Wifi, GPS. I/O energy important CPU optimization might be waste of time.
- asynchronouse power behavior important "I/O energybundles" component stays in high power state Much of I/o energy triggered by small amount of routines

• Accounting granularity

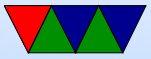


Accounting Granularity

- Granularity: a process, a thread, a subroutine, a system call
- Threads are important as often apps often have third-party threads running
- System calls often trigger different power states, so track them
- They track call stacks when system calls made.

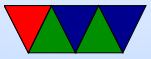


Periodically poll things. And log threadID at context switch



Asynchronouse Power Behavior

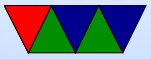
- Lots of components (CPU, mem, sdcard, wifi, nic, 3G, bluetooth, GPS, camera, acceleromteter, digital ompass, lcd, touch sensor, microphone, speakers) and many use as much power as CPU
- Tail power: a routine can start up a high-power component but it does not shut off until long after the component is done
- Wakelocks: some things must keep the phone from



sleeping. i.e sync, etc. source of bugs

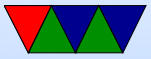
- Exotic components: app can turn on something like CPU and it stays on after context switch, even if other app not using it

- Mearusing power current drawn through battery.



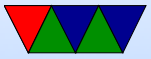
Accounting Policies on Cellphones

- How to attribute energy use to what caused it
- example, send some data. Rmapus up the 3G radio for 2.5s before actually start. 61uAH. But still draws power for 6s after completed
- Why uAH? Batteries often rated in AH (Charge). Not the same as Joules (energy) AH not take into account voltage. Voltage can change while battery discharging.
- If multiple processes use the radio, how split up the cost?

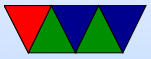


Weights?

- Last-trigger policy: attribute tail to last routine that would trigger.
- If multiple system calls using component, try to split up evenly among them
- High-rate components: not cover RAM or OLED.
- RAM modeled with LLC performance counter. Interesting they try to use perf_event

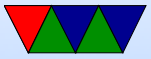


- OLED power can be calculated based on the *colors* on the screen. Can screen-scrape and estimate

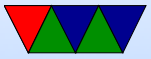


Eprof

- Android and Windows Mobile (only describe Android in this paper)
- SDK Routine tracing: Android Dalvik VM provides runtime tracing
- NDK Routine Tracing: native development kit
- System Call Tracing: use ADB (Android Debugger) logging. Also modify bionic C library

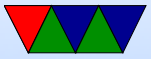


- Modify framework to do syscall tracing without having to modify programs (no source)
- Accounting and Data Presentation



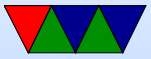
Evaluation/Related Work

- Split-time accounting. Samples, accounts power to whatever running at sample time?
- Accuracy, split-time is worst as it attributes asynch power to pid0
- Overhead, both power and performance



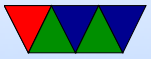
Applications/Benchmarks

- 3G, no room for wifi results
- Android Browser
  Google triggers GPS. 53/31/16 CPU/3g/GPS
  34 threads
- Angry Birds
  time in ads, etc
- Free Chess
  ads



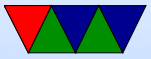
- NYTimes
  Obfuscated code, ads

- Finding energy bugs
  Found wakelock bug which was reported and fixed



Optimizing I/O Energy using Bundles

- I/O consumes most of energy
- I/O energy in a few bundles
- Very few routines perform I/O



Second related work