

Fine Grained Energy Accounting on Smartphones with Eprof

Nishad Gothoskar
ngothosk@andrew.cmu.edu

1: Summary

This paper focusses on the development of a *gprof* for mobile energy usage called *eprof*. Their focus is being able to provide an app developer exactly the information in the right format and context that will be useful to them.

The main problem they faced is linking energy usage to the specific process they are coming from and is it a process, a thread, a subroutine, or a system call. ANother challenge was dealing with asynchronous power behavior, a component consuming power even after its use has been completed. (tail energy like we talked about in a previous paper)

This profiler really gives developers more insight into what kind of consumption their apps have. This helps them optimize and write apps that are more efficient to run. But again, this paper puts most of the blame on apps

I still think the real change needs to come from hardware. But I understand that is much harder to do, as both papers touch on

2: Strengths

- Testing profiling on popular apps gives us insight on how eprof works on a variety of different type
- I feel like integrating this into developing software like Visual Studio and XCode will make it something that developers have open all the time. Raising developer awareness about energy usage is key

3: Weaknesses

- I didn't like how they dealt with asynchronous power. It just attributes it to the last entity. This might say an app is using more energy than it actually is.
- Maybe iOS too.

4: Future Directions

- Recommendations on how to reduce energy usage
- A little bit of a hard to understand paper.