Energy Management in Mobile Devices with the Cinder Operating System

Nishad Gothoskar
ngothosk@andrew.cmu.edu

1: Summary

This paper steps a little bit away from what the last discussed about 2-way interaction between OS and application. In this Cinder OS scheme, the OS is the "boss". A better term would be that the OS is the "bank" and delegates the resources it properly and efficiently to the applications/processes that need it.

The two defining terms of their design are *reserves* and *taps*. On a thread level, the OS assigns reserves of energy accordingly to what it believes the thread needs and deserves. The clever part is that an application itself can do what it likes with its given reserve whether to split it up, assign it to other spawned threads.

The tap is basically the tube that moves energy between reserves. It has a defined fixed rate at which it transfers energy, giving the OS the ability to restrict the rate of energy consumption. This schema for enegy control and allocation is relatively simple, making it easier for application developers to properly deal with and efficiently comply with the OS guidlines.

Specifically the Cinder OS puts the burden of modification and compliance on the application. The application must be coded to deal properly with these energy reserves and taps or else things will not go smoothly. The CPU will not process requests if your app has emptied its reserves, which will cause your app to bug, lag, and maybe even crash.

But in the end the OS is basically "managing" energy in similar ways it managers CPU and memory. It gives space (malloc), allows to get processing power (pthread), and reap/recover its resources (wait/kill)

2: Strengths

- The good thing about this is it will force application developers to be more considerate and aware with their energy usage therefore reducing the overall everyday energy use of mobile devices.
- Actually entirely developed a research mobile Operating System
- "Section 4 shows that the radio has a high initial cost and a much smaller amor- tized price for bulk transfers" - we talked about the costliness of the radio, so the take they have is to just let it "build up" and then do it in bulk
- It might actual deter apps from using ads! Because they are aware they are wasting their own resources.

3: Weaknesses

- The possible downside is that it might be unattractive to application developers to have know how to deal with the Cinder Interface
- I think there should be internal SIGnals sent to the application if they run out of resources so they can handle it properly. Not just not process their requests, because that will just be entirely too buggy.
- The improvements aren't entirely as significant as the first paper.

4: Future Directions

- The obvious extension would be to extend this to PCs because as I said earlier, I believe the computer already does the same thing with its other resources... why not with energy
- Port more common apps to be compatible with this interface so possible people will download and install on their devices (more usage helps with advancements)