Lecture Summary 7: Power as a Finite Resource

Apart from wireless communication, the essential component that make our handsets mobile is the battery. This distinct difference from desktop computers have several important implications on both software and hardware design, mainly towards saving power. All the following may improve the duration of a mobile during a battery cycle (i) Batteries and battery performance (ii) Power consumption of the hardware (iii) How the OS and applications utilise power (iv) Usage-behaviour and patterns.

A main distinction is between power delivery (batteries) and consumption (which some part of the hardware will be responsible for). From a software perspective we need to be aware of the battery capacity and different characteristic of hardware components, and we need to design our software systems looking for optimisations in terms of necessary functionality and consumer behaviour.

We will be looking for such optimisations through the discussion, and investigate techniques that are employed to enable us to find them. We will consider if batteries actually are a significant problem, and if this functional drawback is inhibiting mobiles in any practical way.

Batteries

In order to look for optimisations a natural starting point is investigating the source of power, the battery. Batteries have been an engineering challenge for longer than the computer (first prototypes stem from the early 19th century). They are a different class of technology all-together, performance is related to chemical properties, and is principally a material science. The basic features of a battery are the voltage, total energy and internal resistance. The internal resistance determines the maximum current, which is also related to power by $P = IV$. The internal resistance may vary depending on application and serves as a protection against short-circuit (car batteries for example have low internal resistance). The main properties of interest for a battery to be used in mobile hardware, since we are concerned with weight and size, are the specific energy (energy per unit mass Wh/Kg) and energy density (energy per unit volume Wh/Kg). We are also, for practical purposes, interested in the self-discharge rate (expressed as a small percentage), and cycle durability.

Battery technology has at times improved in marginal leaps with discovery of new materials. Considering a duration of the last 30 years mobile rechargeable batteries have made three such leaps, pertaining to three different materials. Nickel-cadmium batteries in the eighties had an energy density of 30 Wh/Kg, three material advances later, the standard mobile battery today is lithium-ion based of about 90 Wh/Kg. While computational efficiency is improving (by Moore's law), computational demands are also increasing, and predicted to increase further [5]. We will determine more carefully below what these requirements approximately will be.

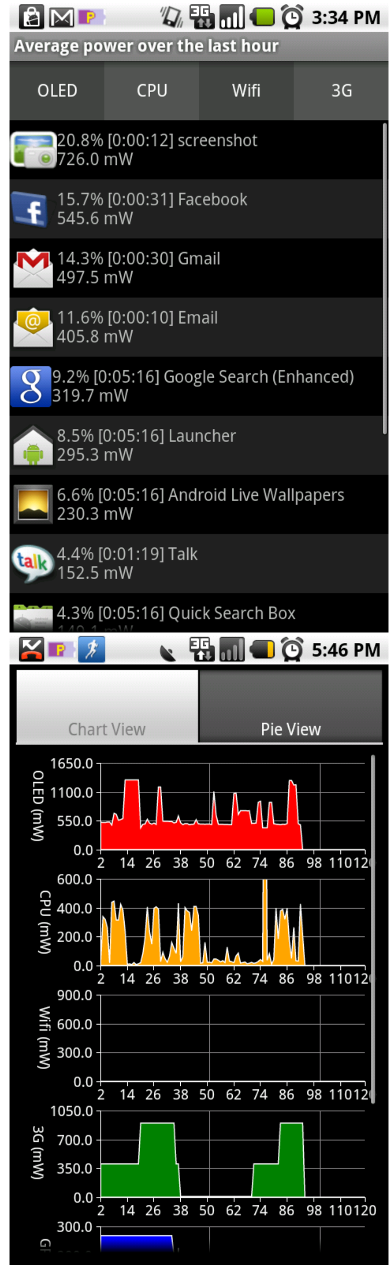

Figure 1: A breakdown of power consumption by the application PowerTutor.

Knowledge of the battery is important in software development for building accurate diagnostic tools, such as the PowerTutor[3] application shown in figure 1. The remaining capacity of a lithium-ion type batteries varies with temperature, number of cycles, age and current. The relevant measurements (temperature, voltage and current) can be combined with a mathematical model of the battery to improve accuracy of power consumption and remaining capacity. Smart batteries will have these systems built-in, and provides an interface to the fuel-gauge. Knowledge of the battery capacity if particularly important for lithium-ion because they may become unstable if they drop to zero capacity, and the sub-system will in these cases cut the battery power at around 90% discharge.

Monitoring Power Consumption

Monitoring power is a useful diagnostic tool, power characteristics may help finding bugs or fault, or help determining where optimisations are possible. In addition to software techniques, an approach that may be more accurate is to connecting a voltmeter between the battery and the terminals on the mobile [2], to obtain time-dependent traces of power consumption. The technique can be used in experiments to investigate the effect on power for any particular aspect. The technique demonstrates, for instance, that passing messages on a wireless network is best done in bursts of data to avoid the power overhead of transitioning from standby. A slight drawback with the technique is that actual draw from a particular hardware component may not always be directly related to the observed measurement in cases where capacitors are buffering power on the device. Another useful insight was the breakdown in power with respect to which pixel colours are switched on. Fully lit Red, Green and Blue displays drew a current of 270mW, 447mW and 1038mW respectively. Combination of these colours are approximately additive, white (since all pixels are turned on) costs the most, and a fully black screen draws close to 0W. An immediate conclusion is that predominantly black user interfaces (like the Window Phone's Metro UI) is a benefit. As a user we can also respond to this effect by choosing a dark wallpaper for the backdrop.

In the case of software, the PowerTutor uses knowledge of battery discharge behaviour, as mentioned above, and monitors the built-in battery-gauge. The estimated accuracy is given as 5%, it knows how to make the breakdown of power between the components (CPU, LCD, GPS, Wi-Fi,cellular, and audio component) by explicitly controlling their activity states and maintaining hardware models that pertain to a specific cellular model.

Extracting results from the power-diagnostics we can see in the example while the screen is on we it uses between 550 to 1000mW (as suggested above will vary depending on the image shown and set of particular pixels). The CPU will varies widely, but draws approximately upwards of 1W, and 3G is also expensive costing in bursts between 0, 350 and 800mW. Another resource which can draw substantial power (around the 300mW range) is the graphics processor [1]. For an application developer, minimising the use of any of these resources will have benefits on the battery-cycle of the phone. An example of an application that saves power is a car-navigation system, it may shut down the screen and use voice synthesis to communicate instructions to the user, audio only uses a fraction of the power compared to graphics and the durations of this approach are much shorter.

Usage Patterns

In a study of power usage that included user charging routines [6], we can note that most people rely on a inter-day charging cycle. It is beneficial to keep in mind that applications should not necessarily need to bring the charging cycle out of sync with the period of the day. Although generally it is worthwhile to minimise power consumption because of continuously increasing demands. In growing number of applications that we are adopting, and in more applications that have constant processing requirements, and for predicted increases in requirements by improvements in hardware and bandwidth (see below).

Discussion

Monitoring power consumption, as illustrated in figure 1 shows that for most type of usage the three most energy demanding parts of the phone are the central processing, wireless communication and display, this is also verified by Carrol *et al* [1]. Of the three, power used on the display may decrease as we invent more efficient solutions, for example reflective LCDs such as E-Ink[1].

A power-saving feature that the Android SDK provides are the possibilities for applications to register intents with the operating systems. If an application doesn't depend on real-time information and intent can given for a resource, such as a request for geographical location. When a location becomes available, the application will be signalled. In this way several applications can share power dependent operations simultaneously.

Demand on power for wireless bandwidth and central processing will certainly increase in the future. 4G wireless technology are proposing significant bandwidth increases[2]. The increased bandwidth and increased demands of future applications, such as high-definition video processing and interactive video conferencing, may increase computational requirements by several orders of magnitude, one estimation of near-future requirements are about 1000 Mops/W [5] compared to about 10 today. Some problems of power consumption may be solved at the system architecture level. Some ASICs today have efficiencies of about 1000 Mops/W, so we may see more custom ICs designed for specific processing tasks such as for high definition video. Baseband signal processing which is computationally expensive is done by specialised hardware such as Signal-processing On-Demand Architecture [4], which maintains programmability.

[1] http://en.wikipedia.org/wiki/E_ink
[2] www.ieee802.org/secmail/pdf00204.pdf

Improved charging techniques may to some extent alleviate the problem of power consumption. Inductive charging[3] may become more ubiquitous around the home and workplace which could reduce the need to design for whole-day charging cycles. Another solution is to use multiple devices, if all our data is in the cloud and we only depend on the thin-client model, we can swap devices as they run out of power.

Research on batteries indicates that future batteries might be able carry several times more power than today, this would virtually solve the problem all-together (at least for now)[4] [7].

In summary, there are many directions of improvement and we will inevitably be moving along all of them. Power-considerations is one of the main drawbacks of mobiles, but it doesn't look like it will hold us back in any serious capacity.

References

[1] Carrol A. and Heiser G. An analysis of power consumption in a smartphone. *USENIX conference*, 2010.

[2] Rice A. and Hay S. Decomposing power measurements for mobile devices. *IEEE PerCom*, Mar 2010.

[3] Zhanh L. and et al. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, 2010.

[4] Yuan Lin, Hyunseok Lee, Mark Woh, Yoav Harel, Scott Mahlke, and Trevor Mudge. Soda: A low-power architecture for software radio. *Proceedings of the 33rd annual international symposium on Computer Architecture*, 2006.

[5] Woh M. and et al. Mobile supercomputers for the next-generation cell phone. *Computer*, 43:81–85, Jan. 2010.

[6] Vallina-Rodriguez N, Hui P., Crowcroft J., and Rice A. Exhausting battery statistics. *SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, 2010.

[7] Zhao X., Hayner C.M., Mayfair C., and Kung H.H. In-plane vacancy-enabled high-power sigraphene composite electrode for lithium-ion batteries. *Advanced Energy Materials*, 1:1079–1084, Oct 2011.

[3] http://en.wikipedia.org/wiki/Inductive_charging

[4] http://www.northwestern.edu/newscenter/stories/2011/11/batteries-energy-kung.html