A Survey of Power Management Techniques in Mobile Computing Operating Systems

Gregory F. Welch
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175

welchg@cs.unc.edu

Abstract

Many factors have contributed to the birth and continued growth of mobile computing, including recent advances in hardware and communications technology. With this new paradigm however come new challenges in computer operating systems development. These challenges include heretofore relatively unusual items such as frequent network disconnections, communications bandwidth limitations, resource restrictions, and power limitations. It is the last of these challenges that we shall explore in this paper—that is the question of what techniques can be employed in mobile computer operating systems that can reduce the power consumption of today's mobile computing devices.

1. Introduction

Batteries are typically the largest single source of weight in mobile computing devices. While reduction of the physical dimensions of batteries is certainly a possibility, such efforts alone will reduce the amount of charge retained by the batteries. This will in turn reduce the amount of time a user can use the computing device before being forced to re-charge the batteries. Such restrictions tend to undermine the notion of mobile computing. However if we can succeed in reducing the basic consumption of individual components of mobile computing devices, we have the luxury of either reducing the battery dimensions while retaining the original charge characteristics, or increasing the battery charge time while retaining the original battery dimensions, or some combination of both.

Many physical components are responsible for ongoing power consumption in a mobile computing device. For example, Table 1 below lists the top power-consuming components of the Sharp PC 6785 as presented by Forman et al. [1]. This data is indicative of most of today's mobile computing devices, and highlights the areas requiring attention for power reduction.

Immediately we notice that the top three items are simply the base CPU and memory, operating at three different clock rates. The backlight for the LCD screen is the next largest consumer, and the hard drive motor follows the backlight. While it is certainly reasonable to expect that a mobile operating system might manage the intensity of an LCD screen backlight, we choose in this paper to survey only various power management techniques aimed at controlling the power consumption of the CPU and the hard drive.

Table 1: Power consumption of various components of the Sharp PC 6785 shown in order of decreasing power from top to bottom. (Data courtesy of Forman et al. [1].)

Component	Power (Watts)
base system (2MB, 25 MHz CPU)	3.650
base system (2MB, 10 MHz CPU)	3.150
base system (2MB, 5 MHz CPU)	2.800
screen backlight	1.425
hard drive motor	1.100
math co-processor	0.650
floppy drive	0.500
external keyboard	0.490
LCD screen	0.315

1.1 CPU Power Consumption

In general, the power consumed by the CPU and memory is related to the clock rate, the supply voltage, and the capacitance of the devices being switched (e.g. the transistors). The reduction in CPU and memory power consumption as the clock rate decreases (cf. Table 1) is a result of the switching characteristics of the logic gates in today's typically CMOS* VLSI circuits. When a complementary transistor pair in such VLSI circuits switches states, a brief short-circuit between the power supply and ground occurs, resulting in wasted energy. Unfortunately, most of the gates in a CMOS CPU will switch states on *every* clock cycle. Thus the higher the CPU clock rate, the more frequently the gates are switching, and the more energy or power is wasted.

In addition, the wasted power is related to the operating voltage of the components—the voltage appearing across each complementary transistor pair. The power wasted by logic gates during the brief short-circuit switching transitions is equal to the supply voltage squared divided by the resistance of the short-circuit path, i.e. $P = V^2/R$. Because the switching resistance is generally fixed, the wasted power is proportional to the square of the operating voltage.

* Complementary Metal Oxide Semiconductor circuits typically consist of logic gates that employ pairs of N and P-type transistors connected in series between the power supply and ground in a complementary fashion.

Furthermore, in order to accomplish useful work, transistors are typically connected together to form "chains" of logic, with each transistor or pair driving the gates of succeeding transistors. The larger the gate capacitance of the succeeding transistors, the more energy required to charge them during a transition. Thus the switching power is also proportional to the gate capacitance of any driven transistors.

The total power required by the CPU and memory is therefore proportional to CV^2F where C is the total capacitance of the wires and transistor gates, V is the supply voltage, and F is the clock frequency. While C can only be affected during chip design, newer chips are beginning to make it possible to vary F and V at run-time, in order to achieve linear and quadratic (respectively) savings in power. Thus a mobile operating system can adjust these parameters in cooperation with scheduling policies to reduce the CPU's consumption of energy. This is the topic of the first paper we will look at in section 2, "Scheduling for Reduced CPU Energy" by Weiser et al.

1.2 Hard Drive Power Consumption

The power consumption of a hard drive in a mobile computing device can also be controlled by judicious scheduling of drive "spindown" events*. Such control is becoming more and more important as other components of mobile computing devices reduce their power consumption. These other reductions are resulting in overall drops in power consumption, with the result being that the relative amount of power consumed by disks has increased from 9% to 31% in some cases [5]. Disk drive power management is the topic of the two papers we will explore in section 3, "A Quantitative Analysis of Disk Drive Power Management in Portable Computers" by Li et al., and "Thwarting the Power-Hungry Disk" by Douglis et al.

1.3 Power and Wireless Communication

As somewhat of a supplement to secondary storage, periodic wireless broadcasts can sometimes be used to simultaneously disseminate information, e.g. a large common database, to a large number of mobile computing devices. While this technique can in a sense provide some of the functionality of a local hard drive, the organization of and access to broadcast data is very different from that of a hard drive. In section 4 we will examine "Energy Efficient Indexing on Air", where the authors present and evaluate two methods for organizing and accessing such broadcast data.

2. "Scheduling for Reduced CPU Energy" [3]

As was discussed in the introduction of this survey, the power consumed by the CPU in a mobile computing device is significant. In fact, if the CPU clock frequency and supply voltage can be controlled, linear and quadratic savings in power can be realized (see section 1.1). This is the main motivation for the work presented by Weiser et al. in their paper on "Scheduling for Reduced CPU Energy". In particular, because lowering the supply voltage results in a quadratic power savings, this is the authors' preferred method for power reduction. As the authors note however, the supply

* Transitions from a drive's *idle* state to its *sleep* or *off* state where it consumes no energy [4].

voltage cannot be lowered (dynamically) without also reducing the clock speed. Therefore the authors consider the CPU clock rate to be linearly adjusted with the supply voltage. Incidentally, it is the authors' assumption throughout their paper that such dynamic adjustments to the supply voltage and clock rate are indeed possible. As this hardware technique is becoming more popular, this is not an unreasonable assumption.

In this paper, the authors present and evaluate three algorithms for adjusting the CPU clock speed under the control of the operating system. The evaluation of the algorithms was accomplished through trace-driven simulations. The trace data was collected from a number of UNIX workstations over several working days, under otherwise normal conditions. During the trace collection, the workstations were used for a variety of applications such as e-mail, simulations, word-processing, etc. The collected traces were then used in off-line simulations of the three algorithms, allowing for flexible but consistent comparisons.

The basic approach of the algorithms is to balance CPU usage between periodic bursts of high CPU utilization and the remaining periods of idle time. For example, assume we have a task that normally results in a 50 millisecond (ms) burst of CPU usage at the full clock rate, followed by a 50 ms idle period. If that task can be replaced by a 100 ms period of CPU usage at half the normal clock rate, without affecting the user adversely, then such a scheme can be used to conserve power. This can be accomplished by dynamically adjusting (slowing) the CPU speed, effectively "stretching" activities from busy periods into subsequent adjacent idle periods.

The CPU speed adjustment decisions made in the simulations were based on the measured amounts of run time and idle time in various segments of the traces. In particular, all three algorithms rely on the assumption that the operating system *sleep* events that normally result in CPU idle time can be classified into two categories, "hard" and "soft" sleep events. Hard sleep events result in idle time during which the CPU speed cannot be reduced. For example, in the case of a disk wait, the sleep is typically issued from within the kernel's biowait() routine. On the other hand, soft sleep events result in idle time during which the CPU speed can be reduced. For example, it should be possible to slow and therefore delay completion of a sleep event that occurs while waiting for the next user keypress. The reason for such classifications was to be "fair" about assessments of which idle times were candidates for use in balancing periods of high and low activity.

Although CPU speed adjustments during small windows might affect job ordering, the authors did not explicitly reorder trace data events in the simulations to "correct" for any such occurrence. Their justification for this was twofold. First, significant reordering will generally only occur if unusually high (unreasonable) loads are offered to the CPU. Second, in their simulations a slowed CPU is ramped back up to full speed with increasing loads, thus restoring the original full speed conditions of the traces during periods of high demand.

The three algorithms presented are named OPT, FUTURE, and PAST. The three algorithms are similar in spirit to those typically used in (evaluating) page replacement policies for virtual memory systems. OPT is an impractical and undesirable algorithm. It is completely optimistic (and impractical) in the sense that it assumes complete knowledge of the future work to be done in an interval. It is undesirable because it adjusts the CPU rate over an *entire* trace, severely impacting the run times of user jobs and ignoring the relative importance of events such as keystroke response or network communications. OPT is however useful as a benchmark for performance. The FUTURE algorithm is similar to OPT in that it peers into the future, but it does so for only short windows of time, and optimizes over only those windows. Like OPT, FUTURE is unrealistic because it peers into the future. Unlike OPT, it is practical because it only optimizes over short windows, hence the impact on time-critical work is minimized. The PAST algorithm is a realistic (practical) version of FUTURE. It is realistic because instead of peering into the future, it looks to a short window in the past for information about CPU usage. It is desirable because like FUTURE it only monitors activity over a short window of the trace. It is less reliable because it assumes the activity in the next (future) window will be like that in the current (past) window. If the adjustments were optimistic (the activity level was higher than expected) and a window ended with work left to be done, that work was added to the work in the following window. The amount of such "carry-over" work was used as a measure of penalty incurred with different window lengths under this scheme.

The authors conclude that it is indeed more profitable (in terms of power savings) to spread out work when possible over periods of slower CPU clock rates, rather than contending with bursts of high-speed activity followed by wasted idle time. They conclude that in their simulations a PAST implementation with a 50 ms window would have saved up to 50% of the power in a 3.3 volt CPU, and up to 70% in a more aggressive (in terms of power savings) 2.2 volt CPU. They note that the energy savings are dependent on the window length, and also the minimum CPU clock rate used. Specifically, the interval between adjustments must not be too small or CPU usage becomes "bursty", and the minimum clock rate used must not be too slow or the assumptions about the future windows become less reliable resulting in higher penalties from excess or carry-over work. On the other hand, windows must not be too large or the interactive response of the system will be impacted. As a compromise between power savings and response time, the authors suggest a shorter 20 or 30 ms window.

3. Disk Drive Power Management

A common scheme for reducing the power consumed by hard drives in mobile computers is to "spindown" or turn off drives during extended periods of non-use. The papers discussed in the following two subsections evaluate schemes for determining when to perform a spindown.

3.1 "A Quantitative Analysis of Disk Drive Power Management in Portable Computers" [4]

It is common for manufacturers of hard disk drives for mobile computers to recommend a fixed time threshold for spinning-down the hard drives, where the threshold is generally on the order of 3-5 minutes. In this paper, the authors perform a quantitative analysis of the pros & cons of hard drive spindown as a power reduction technique. The basic premise of this work is that a fine-grained approach to spinning-down the hard drive is more appropriate than the coarse-grained approach pursued by manufacturers, given what has found to be normally very scattered usage patterns*.

Like the work done by Weiser et al. described in section 2, the authors of this work used a large collection of operating system traces for their evaluations. For this work (paper) the authors col-

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.