ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Ecosystem and Cinder OS for Energy Efficient in Mobile Devices

Narra Aravind
B.Tech 4th Year,
Electronics and Communication Engineering,
Sreenidhi Institute of Science & Technology,
Hyderabad-501301.

Yalamanchili Nikhita
B.Tech 4th Year,
Electronics and Communication Engineering,
Sreenidhi Institute of Science & Technology,
Hyderabad-501301.

ABSTRACT:

Besides performance, energy is an important issue for operating systems especially when they are running on a mobile platform. Targeting an optimal lifetime of the system battery, operating systems have to provide mechanisms for energy accounting and energy controlling. This document describes the prerequisites for an effective energy management. It also introduces ECOSystem and the Cinder operating system, two implementations of an energy-aware operating system, along with applications developed for these systems that demonstrate the benefits of energy management on operating systems.

Keywords:

D.4.1 [Operating Systems], Process Management, energy efficiency, Mobile devices.

1.INTRODUCTION:

Since mobile phones the dominating mobile systems on the market are capable of running general-purpose operating systems like Linux, there is a great demand for making operating systems more energy-aware. Improvements have been made to make energy consumption more visible by means of smart device interfaces. In addition, modern devices provide a lot of energy-saving modes to avoid wasting energy. These improvements allow operating systems a coarse control of the platforms' energy behavior. Mobile devices today run a variety of full UNIX-like kernels with user-downloaded applications sitting atop complex software stacks. Apple's App Store and Google's Android Market both provide immediate access to a growing suite of thousands of third-party applications [1].

This explosion of mobile software complexity and application availability makes it difficult to reason about the energy requirements and expenditures of mobile systems. Additionally, the sheer bulk of available software implies that many applications are buggy, malicious, or inefficiently implemented.Unfortunately, users currently have no quantitative means to determine how much energy an application is consuming, nor do they have much control over it, beyond choosing whether or not to run the application at all. Current systems are not in command of their limited energy budgets and, consequently, users have no means of reserving or prioritizing energy for important applications while curtailing others. Although platforms can be extended to track energy for tasks [9, 10, 5], previous research has not adequately ad-dressed dynamic systems, application control of energy budgets (i.e. sub allocation), and policy generation via system feedback.We argue for a new operating system called Cinder. Cinder is designed to answer not only which applications are energy-expensive, but also what behavior we can expect and how they may be controlled by both the user and the applications in order to achieve their desired energy goals. Cinder empowers users and applications - the former via useful system feedback coupled with policy generation, and the latter through delegation of energy resources and the enablement of application-level resource control.

2.Related work:

The ECOSystem has a data-centric security architecture, consisting of a minimal, trusted kernel upon which user-level library operating systems (e.g. a POSIX layer) are built. Cinder is based on ECOSystem and inherits many of its features. Most importantly, Cinder is built atop a minimal set of seven first-class kernel objects. Object containers, one such example, are particularly useful for resource tracking and revocation, enabling fine-grained resource ac-counting using a fundamental, pervasive system primitive. Finally, a fast, simple, and non-reentrant kernel greatly simplifies accounting across the protection boundary. ECOSystem is simple, yet sophisticated; the above are only the salient aspects involved in resource handling.Cinder extends ECOSystem by adding a new fundamental kernel type for energy: capacitors, an abstraction for flexible, dynamic, and hierarchical energy management. Cinder also differentiates itself by maintaining task profiles to aid meaningful policy generation. In Cinder, a capacitor is modeled after its real-world electrical counterpart, but with a special distinction: it contains rights to request consumption of joules, rather than reservations for actual joules.

A capacitor is then a storage pool for potential energy, with additive increase and multiplicative decrease in residual potential energy. Capacitors can naturally place a ceiling on this quantity, which would ensure that it does not grow without bound.Cinder will also maintain task profiles, running statistics of application energy consumption. Since it is unreasonable to expect a user to know or care how many joules their mail reader, for example, consumes over a certain time interval, it is nonsensical to have users express energy policies in units of energy or power, or fractions thereof. Instead, we envision users expressing their desire for the system to ensure that their mail reader will run for a given number of minutes or hours based on previous application behavior. It is then the responsibility of the system to keep the users well-informed, first aiding policy generation and later enforcing them.

3.ENERGY VISIBILITY:

Energy management on operating system level requires at first precise information about energy consumption of the managed devices on the one hand and characteristics of the battery on the other hand. Although modern hardware components may provide interfaces to retrieve the relevant parameters, in most cases measurements are necessary to get accurate results.

Due to the closed nature of mobile systems like smart phones, measuring the power consumption of individual, functional units is difficult. In such cases measuring the total system power and estimating the consumption of single components will lead to inaccurate results. However, there are mechanisms that allow operating systems to work around this kind of error.

Battery Characteristics:

Battery lifetime is an important factor for mobile systems and therefore it is a primary target of energy management to maximize it. The discharge rate of a battery has a nonlinear impact on the usable battery capacity. High discharge rates can lower the capacity to 70%-80%.The Smart Battery interface in the ACPI specification [2] allows operating systems to query all relevant values like current voltage, remaining capacity, and discharge rate. Unfortunately, queries to this interface are slow and return only averaged values of power consumption.

Power Consumption Measurement:

To measure the power consumption of managed devices one has to consider the various states a device could reach. The consumption of a processor, for example, depends on the usage of the built-in functional units and therefore on the current workload. Additionally modern devices offer power saving modes we also have to consider. The transition between the different modes of such devices is also a source of energy consumption that has to be taken into account.

Energy Profiling:

To provide precise energy accounting we have to assign the measured energy consumption to the application that causes it. A simple approach is to sample power consumption and to assign the sample values to the thread currently using the processor. The results will be inaccurate due to the fact that simultaneously active hardware components could have been triggered by two different applications.

A more precise method to account energy online makes use of performance counters embedded in modern processors. Performance counters are implemented as hardware registers. By registering events that imply the consumption of a certain amount of energy at these counters, the operating system is able to change its behavior and therefore to adapt the expected power drain. For example, a high number of main memory references, although only a little number of instructions was executed, could indicate that performance is dominated by the main memory latency. Therefore, throttling the processor speed will improve the energy efficiency without a significant negative impact on system performance.

4.ECOSYSTEM:

ECOSystem is an implementation of the currentcy model, based on the Linux operating system. For handling energy as a first class resource, the kernel was modified to provide mechanisms for allocating and accounting energy. This includes a resource container object for the storage of energy units. The prototype of ECOSystem was evaluated on an IBM Thinkpad T20 laptop with a set of micro benchmarks targeting the CPU, the disk, and the network interface.

Currentcy Allocation:

ECOSystem provides an interface to allow the user to set a target battery lifetime. The maximum discharge rate for the battery depends on the targeted lifetime. The difference between the current and the maximum discharge rate determines the amount of energy that can be allocated in a fixed time interval. This value corresponds to an amount of energy units that can be distributed between all competing tasks. The distribution of energy is performed by a periodic kernel thread that modifies the values of all tasks' resource containers periodically. If a task does not use all of its energy units, it can accumulate these units up to a certain limit.

Currentcy Accounting:

Performing system operations, like the execution of an instruction or requesting data from a disk, requires a certain amount of energy. If a task wants to perform such an operation, the operating system checks the corresponding energy value. If the value is sufficiently high, the operation will be executed. Otherwise, the task is blocked until it accumulated enough energy units for the requested operation.Due to the numerous, different energy characteristics of managed devices, ECOSystem uses energy charging policies with respect to these differences:

•CPU: Execution of instructions on the processor is charged in dependence of a fixed power value and the processing time. A task running out of energy units will be preempted until it accumulates more units.
•Hard Disk: In addition to a fixed amount of energy for every block accessed, all tasks that accessed the disk within the same session share the costs for spinning up and down the disk.
•Network Interface: Sending or receiving of data packets is charged in dependence of the transmit power, the size of the packet, and the bitrate.

Assigning the costs of CPU operations to the correct task is quite simple, whereas tracking the energy consumption of the disk or the network interface is more complex. Files in ECOSystem are accessed through file related system calls. Hereby the container ID of the calling task is stored within the buffer cache entry of the disk. When the cache entry is actually written, the appropriate resource container will be charged for the operation. Source tasks of network operations are identified by their associated source socket.

Energy Sharing:

This experiment demonstrates that ECOSystem is able to manage energy sharing between simultaneously running applications. One of the applications is ijpeg, a CPU-intensive graphical application. The other application is netscape, which is mainly using the network interface. A performance indicator for ijpeg is the computation speed and therefore its CPU utilization. The performance of network applications like netscape depends on response times and therefore on page load latencies. The measurement results for various energy ratios between these two applications, whereby the amount of allocated energy for both applications is set to 5 W.The measurements show that both tasks match their targeted allocation. Although netscape makes use of all three functional units, ECOSystem tracks the power consumption across all devices accurately. The CPU utilization of ijpeg, and therefore its performance, increases nearly equivalently to the allocated power, whereas netscape shows a non-linear behavior due to non-linear energy characteristics of the network interface.

5.CINDER OPERATING SYSTEM:

Cinder is an operating system designed for modern mobile phones. It extends HiStar [4], a secure operating system that provides containers and gates in addition to conventional kernel objects.Every object in HiStar has to be referenced by a container including containers ifself. The hierarchical structure of containers is used by the Cinder operating system to deal locate resources and therefore to implement a mechanism for delegating and subdividing resources. Gates provide secure inter-process communication by using security labels.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

for all objects. This security concept of HiStar ensures isolation of containers.The Cinder operating system extends HiStar with two additional kernel objects: reserves and taps. These two extensions to the kernel are explained in the following paragraphs.

Reserves:

Reserves are kernel objects that permit the usage of energy. To determine the amount of energy a task is allowed to use, a power model similar to the currentcy model is used. The value of a reserve is therefore a permission to use a certain amount of energy in a fixed time interval. If the value of a reserve is not high enough for the operation the corresponding task wants to perform, the kernel will prevent execution.Although it is possible to partition a reserve's value and assign it to other reserves, this is in most cases not practical, because threads rarely need to delegate fixed amounts of energy. Usually threads provide other threads with energy by using a fixed rate.

Taps:

In order to guarantee a certain battery lifetime the discharge rate of the battery has to be limited. Transferring energy between reserves with a fixed rate is therefore a better approach than delegating fixed amounts of energy.

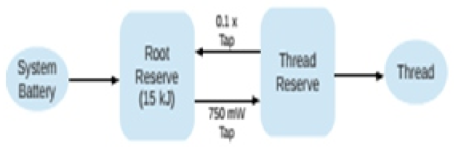


Figure 1: Consumption graph, adapted

Taps are kernel objects that transfer a certain quantity of energy between two reserves with a fixed rate. Therefore a tap contains a rate, a source reserve, and a sink reserve. Instead of transferring a fixed amount of energy, proportional taps transfer a certain fraction of the source reserves' energy to the sink reserve. In practice, taps are implemented as a thread whose job is to transfer energy values between corresponding reserves periodically.

Energy Consumption Graph:

Due to the hierarchical structure of reserves and taps, it is suitable to model it with a directed graph. The root of such a directed graph represents the reserve connected to the system battery; all other reserves are a subdivision of this root reserve.

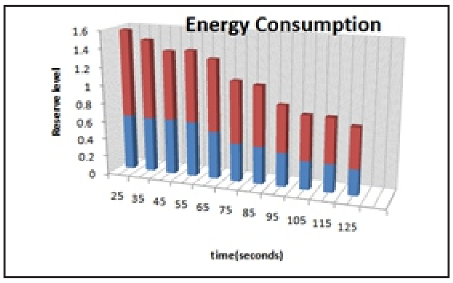
A simple example of such a directed graph is shown in Figure 1. In this example the root reserve is connected to the 15 kJ system battery. A single application draws energy from a separate reserve connected to the root reserve over a 750 mW tap.

Energy Hoarding:

Tasks can accumulate unused amounts of energy to use it for future operations. However, to improve the performance of the whole system, it is good practice to reclaim unused energy and assign it to other tasks. This problem can be solved by using backward taps. A backward tap transfers a fraction of the accumulated energy back to the source reserve. The fraction of the backward tap is also a limitation for the amount of energy the sink reserve is able to hoard. In Figure 1, a backward tap transfers 10% of the reserve's accumulated energy back to the root reserve. When the sink reserve level in the example reaches 700 mW, both taps have the same rate and therefore the level cannot raise anymore.

6.Results & Discussion:

Improving the battery lifetime of a mobile system is just one goal of energy management. Another target is the performance the user expects. A common use case of modern smartphones is to play music while browsing the internet. In this case user experience depends on the delay of network requests and a disruption-free playback of the music file. Although a user may accept varying delays, a disruption of the playback will have an unacceptable impact on his experience.They describe applications that were developed on the Cinder operating system using reserves and taps. They represent typical use cases operating systems have to handle and will show the ability of this operating system to control energy on mobile systems.The platform on which Cinder is implemented and the following applications are evaluated is an HTC Dream, which is also known as Google G1. Because of its closed, integrated structure measuring becomes a challenging task.



ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Fig 2: The energy consumption graph for the background application adapted

Unfortunately the ARM11 core processor provides no performance counters. Therefore energy accounting is based on offline measurements of device power states.

CONCLUSION:

The performance of a mobile platform depends highly on the lifetime of the system battery. Improving energy management is therefore an important topic of research, especially considering that energy is a more and more limited and expensive resource.

Abstracting energy to a first class resource and providing mechanisms to isolate, delegate, and subdivide it enables operating systems to manage energy on a fine-grained level. The evaluation of the presented applications developed on the Cinder operating system and ECOSystem shows that these systems are able to reach a given battery lifetime. Furthermore, these systems allow developers to implement energy aware applications that will improve the user experience by adapting the applications' behavior to the current energy level.

However, it is uncertain if this approach improves the overall energy efficiency, since there were no measurements made to compare energy consumption, performance and battery lifetime of the evaluated systems with standard systems. Providing the ability to account and control energy on such a fine-grained and accurate level requires a lot of additional computation time that causes higher energy consumption and a lower system performance. Consequently, it is doubtful if managing energy as a first class resource is a promising approach to improve energy efficiency on mobile platforms.

REFERENCES:

[1]BELLOSA, F. The benefits of event-driven energy ac¬counting in power-sensitive systems. In Proceedings of the 9th ACM SIGOPS European Workshop (2000), pp. 37-42.
[2]INTEL CORPORATION AND MICROSOFT CORPO¬RATION AND TOSHIBA CORPORATION. Advanced Configuration and Power Interface Specification. http://www.teleport.com/acpi, 1996.

[3]ROY, A., RUMBLE, S. M., STUTSMAN, R., LEVIS, P., MAZIERES, D., AND ZELDOVICH, N. Energy manage¬ment in mobile devices with the Cinder operating sys¬tem. In Proceedings of the European Conference of Com¬puter Systems 2011 (2011), pp. 139-152.

[4]ZELDOVICH, N., BOYD-WICKIZER, S., KOHLER, E., AND MAZIERES, D. Making information flow explicit in HiS- tar. In Proceedings of the 7th Symposium on Operating Systems Design and Implementation (2000), pp. 263¬278.

[5]ZENG, H., ELLIS, C. S., LEBECK, A. R., AND VAHDAT, A. ECOSystem: Managing energy as a first class operat¬ing system resource. In Proceedings of the 10th Interna¬tional Conference on Architectural Support for Program¬ming Languages and Operating Systems (2003), pp. 123¬132.

[6]P. Menage. cgroups, Oct. 2008.http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2-.6.git;a=blob;f=Documentation/cgroups/cgroups.txt;hb=b851ee7921fabdd7dfc96
c4e9609f5062bd12.

[7]C. A. Waldspurger and W. E. Weihl. Stridescheduling: Deterministic proportional-share resourcemanagement. Technical report, 1995.

[8]N. Zeldovich, S. Boyd-Wickizer, E. Kohler, andD. Mazieres. Making information ow explicit inHiStar. In Proceedings of the 7th Symposium onOperating Systems Design and Implementation, pages263{278, Seattle, WA, November 2006.

[9]H. Zeng, C. S. Ellis, A. R. Lebeck, and A. Vahdat.Ecosystem: Managing energy as a firrst class operatingsystem resource. pages 123{132, 2002.

[10]H. Zeng, C. S. Ellis, A. R. Lebeck, and A. Vahdat. Currentcy: A unifying abstraction for expressingenergy management policies. In In Proceedings of theUSENIX Annual Technical Conference, pages 43{56,2003