Energy-Aware Computing Systems

Energiebewusste Rechensysteme

VIII. System Software

Timo Hönig

2018-12-12

Agenda

Preface

Terminology

Accounting and Management
- Abstracting Energy Demand
- Operational Concerns

Energy-Aware Operating Systems
- Currentcy and ECOSystem
- Cinder Operating System
- Linux Energy-Aware Scheduling (EAS)

Summary

Preface: Higher-Level Energy Management

- **motivation** and **origin**
  - lack of feedback on **design decisions** regarding energy demand
  - gap between vision of energy control and reality
    → Milly Watt Project
- use case: *Hiker's Buddy* [3]
  - energy-constraint operations (e.g., GPS)
  - functional design ↔ power state model

Carla Schlatter Ellis
The Case for Higher-Level Power Management
Proceedings of the Seventh Workshop on Hot Topics in Operating Systems (HotOS'99), 1999.

Preface: Higher-Level Energy Management

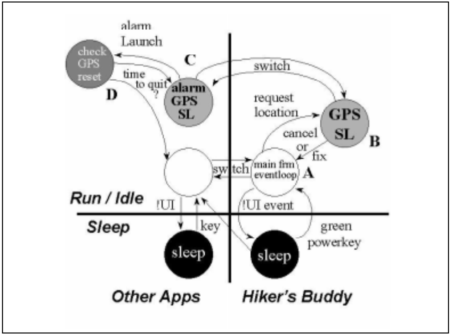


Figure 1. Power State Model

Carla Schlatter Ellis
The Case for Higher-Level Power Management
Proceedings of the Seventh Workshop on Hot Topics in Operating Systems (HotOS'99), 1999.

Preface: Higher-Level Energy Management

- **lower-level** building blocks
  - **energy-management features** at the hardware level (i.e., non-blocking energy management methods)
  - **firmware interfaces** for system controls (i.e., blocking energy management methods)

- **higher-level** abstractions
  - energy **accounting** with energy **models** and **measurements**
  - **resource** management
  - **policies and rights** management → conflict of interests

Abstract Concept: System Software

- **system software**
  - **operating system**
    - program or a set of programs that support (other) programs or applications to **facilitate the programming or operation** of a computer system
    - **monitor** and **control** the execution of programs
    - **operate** the computer system in a specific manner for a particular application
    - implement an **abstract machine**
  - interlocking with **low-level user-space programs** (i.e. system deamons)
- **resource management**
  - {de,}allocation of resources by the system software
  - accounting and enforcement

Abstracting Energy Demand: Resource Peculiarities

- **software resources** as to be used by programs

reusable	
code	critical section/region
data	variable, placeholder

consumable	
signal	notice
message	packet, stream

- **hardware resources** as to be managed by an operating system

reusable	
processor	CPU, FPU, GPU; MMU
memory	RAM, scratch pad, flash
peripheral	input, output, storage

consumable	
signal	IRQ, NMI, trap

- **energy** as a basic **operating-system resource** required to provide hardware and software resources
  - software resources → hardware resources → energy demand
  - energy accounting and management (i.e. resource allocation vs. residual resources)

Abstracting Energy Demand: Accounting

- energy **accounting** at operating system level
  - map resource demand by processes to energy demand
  - exclusive use vs. shared use of resources → attribution of proportions

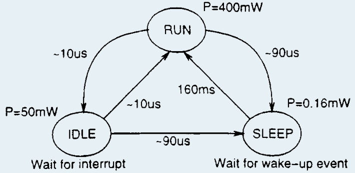
- capturing and tracking energy demand during run-time
  - apply **models**
    - tracking of state and time → device states
    - discrete, logic events → performance counter events
    - ...
  - apply **measurements**

- appropriate metrics for individual capturing methods
  - basic metrics and composite metrics
  - use-case specific granularity (i.e. µW vs. MW)

Abstracting Energy Demand: Accounting

- consideration of power states
- transition delays

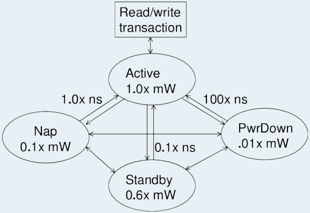
CPU



[2]

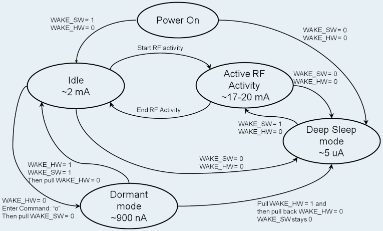
- in isolation...
- ...and in cooperation.

Memory



[4]

Network



Microchip Technology, Inc.

Operational Concerns: Energy Management

1. accounting ✓
2. **allocating** energy (e.g. epoch-based)
   - implicit → process analysis (i.e. based on periodicity)
   - explicit → provisioning based on requests
   - avoid overbooking that would conflict with global goal, prevent:
     - thermal breakdown (i.e. by exceed maximum power)
     - too short operating time (i.e. imbalance of power supply and demand)
3. **administering** residual energy (for next epoch)
   - use residual energy as feedback information
     - amount of residual energy depends on accuracy of energy models and measurements, respectively
   - redistribution and reallocation strategies
     - exhaustion control
     - over-provisioning controls

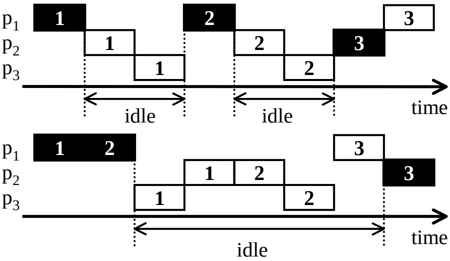
...but OS integration comes with pitfalls...

Operational Concerns: Energy Management

- basic functional requirements
  - accounting ✓
  - allocating ✓
  - administering ✓
- integration causes **conflicts of interest**: process scheduling
  - upon exhaustion of allocated resources
  - reordering of events
  - ...
- pitfalls as to **entering sovereign territory** of the process scheduler
  - priority inversion
  - data dependencies
  - ...

Operational Concerns: Energy Management

- Requester-Aware Power Reduction [5]
  - **track requests** and how they are generated (i.e. by which processes)
  - interaction between **processes** and **power management** facilities at operating system level
  - reordering of requests to reduce overhead and energy demand



black box: **active**
device access (e.g. I/O)

white box: **idle**
no device access (e.g. computation)

challenges:

- violation of priority-based scheduling decisions
- interdependencies between individual processes

Overview

- Currentcy [9] and ECOSystem [8]

  Heng Zeng et al.
  ECOSystem: managing energy as a first class operating system resource
  Proceedings of the 2002 Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '02), 2002.

- Cinder Operating System [6]

  A. Roy et al.
  Energy Management in Mobile Devices with the Cinder Operating System
  Proceedings of the 2011 ACM European Conference on Computer Systems (EuroSys'11), 2003.

- Linux Energy-Aware Scheduling (EAS)

Currentcy and ECOSystem

- Currentcy: A Unifying Abstraction for Expressing Energy Management Policies
  - **Current** → *amount of energy that an application can spend*
  - Curren**cy** → *cf. money as unified abstraction for buying commodities*
  - abstract energy model (1 unit of currentcy is valued at 0.01 mJ)
- **currentcy** is used for...

  ...**energy accounting** and **allocation** across components and processes
  ...**capturing interactions** among energy users in the system

Heng Zeng et al.
Currentcy: A Unifying Abstraction for Expressing Energy Management Policies
Proceedings of the 2003 USENIX Annual Technical Conference (ATC'03), 2003.

Currentcy and ECOSystem

- ECOSystem:
  managing energy as a first-class operating system resource
  - Energy-Centric Operating System
  - motivation: change primary goal of the OS to **energy-efficiency** rather than (speed-based) **performance**
  - primary goal: **user-defined battery life** → determines **amount of currentcy** that can be spent in **each epoch**
  - adaptation of **resource containers** [1]

📄 Heng Zeng et al.
ECOSystem: managing energy as a first class operating system resource
Proceedings of the 2002 Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '02), 2002.

Currentcy and ECOSystem

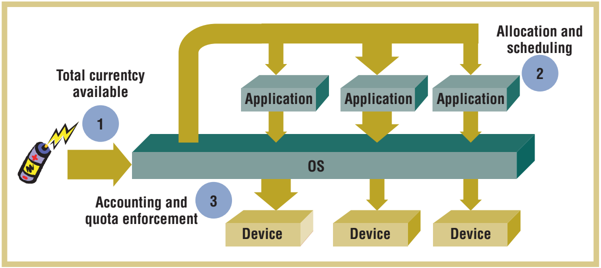
1. query **smart battery** (→ state of charge) prepare for fair allocation of currentcy among processes
2. **allocate** and schedule

→ **block** processes on currentcy depletion

→ processes may decide not to spend their currentcy share during an epoch

3. **accounting**

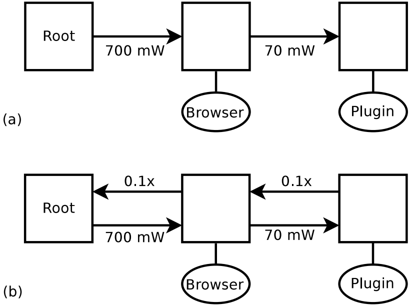
- accumulation of unspent currency is bounded (max. 10x of currentcy per epoch)


Total currentcy available
1
Allocation and scheduling
2
Application
Application
Application
OS
Accounting and quota enforcement
3
Device
Device
Device

[7]

Cinder Operating System

- Energy Management in Mobile Devices with the Cinder Operating System
  - **exokernel-based** operating system built on top HiStar OS
  - concept of reserves and taps
  - **reserve** (energy) → available energy resources
  - **taps** (power) → connection between (hierarchic) reserves

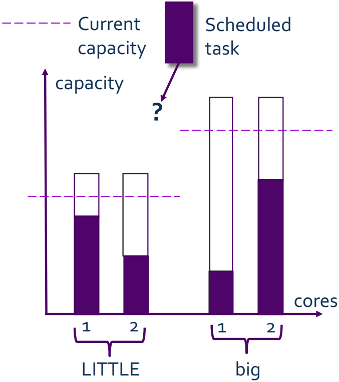


A. Roy et al.
Energy Management in Mobile Devices with the Cinder Operating System
Proceedings of the 2011 ACM European Conference on Computer Systems (EuroSys'11), 2003.

Linux Energy-Aware Scheduling

- Linux Energy-Aware Scheduling (EAS)
  - motivation: exploit heterogeneity for **peak performance** but **low average power** → software counterpart to ARM big.LITTLE architecture
  - energy-aware scheduling for heterogeneous multi-core systems
  - per-CPU energy model necessary

- EAS goals
  - process-dependent core pinning → reliable per-process predictions
  - adaptations of process scheduler
    - adapt to heterogeneous cores
    - energy-awareness → models + estimation
    - integration with DVFS subsystem



elinux.org

- Linux upstream: work in progress

Considerations and Caveats

- **system software**
  - abstraction of energy demand at operating-system level
  - identify interrelationships from higher-level perspectives
- managing energy as a **basic system resource**
  - accounting, allocation, and administering
  - capture and track power states → processes and devices
  - reduce energy demand by reordering
- **energy-aware operating systems**
  - holistic, system-wide resource management
  - use lower-level building blocks (i.e. energy management functions)
  - challenging integration for legacy operating systems

Paper Discussion

- paper discussion
  - Rolf Neugebauer and Derek McAuley
    Energy is just another resource: Energy accounting and energy pricing in the Nemesis OS
    Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS'01), 2001.

Subject Matter

- **system software** is the pivotal element for the **operation of energy-aware computing systems**
- „***energy is just another resource***", its management is a challenging endeavour
- **high-level perspectives** are essential for holistic, system-wide energy management techniques

- reading list for Lecture 9:
  - R. Pereira et al.
    Energy efficiency across programming languages: how do energy, time, and memory relate?
    Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering (SLE'17), 2017.

Reference List I

[1] Banga, G. ; Druschel, P. ; Mogul, J. C.:
Resource Containers: A New Facility for Resource Management in Server Systems.
In: *Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI'99)*, 1999, S. 45–58

[2] Benini, L. ; Bogliolo, A. ; Micheli, G. D.:
A survey of design techniques for system-level dynamic power management.
In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 8 (2000), June, Nr. 3, S. 299–316

[3] Ellis, C. S.:
The case for higher-level power management.
In: *Proceedings of the 1999 Workshop on Hot Topics in Operating Systems (HotOS '99)* IEEE, 1999, S. 162–167

[4] Fan, X. ; Ellis, C. ; Lebeck, A. :
Interaction of Power-aware Memory Systems and Processor Voltage Scaling.
In: *Proceedings of the 2003 Workshop on Power-Aware Computer Systems (PACS'03)*

Reference List II

[5] Lu, Y.-H. ; Lu, Y.-H. ; Lu, Y.-H. ; Lu, Y.-H. ; Benini, L. ; De Micheli, G. ; De Micheli, G. ; De Micheli, G. :
Requester-aware Power Reduction.
In: *Proceedings of the 13th International Symposium on System Synthesis (ISSS'00)*, 2000, S. 18–23

[6] Roy, A. ; Rumble, S. M. ; Stutsman, R. ; Levis, P. ; Mazières, D. ; Zeldovich, N. :
Energy Management in Mobile Devices with the Cinder Operating System.
In: *Proceedings of the 2011 ACM European Conference on Computer Systems (EuroSys'11)*, 2011, S. 139–152

[7] Zeng, H. ; Ellis, C. S. ; Lebeck, A. R.:
Experiences in managing energy with ecosystem.
In: *IEEE Pervasive Computing* (2005), Nr. 1, S. 62–68

[8] Zeng, H. ; Ellis, C. S. ; Lebeck, A. R. ; Vahdat, A. :
ECOSystem: managing energy as a first class operating system resource.
In: *Proceedings of the 2002 Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '02)* ACM, 2002, S. 123–132

Reference List III

[9] Zeng, H. ; Ellis, C. S. ; Lebeck, A. R. ; Vahdat, A. :
Currentcy: A Unifying Abstraction for Expressing Energy Management Policies.
In: *Proceedings of the 2003 USENIX Annual Technical Conference (ATC'03)*, 2003, S. 43–56