Design of Energy-Adaptive Systems

By
Rajesh Vuppala
Constraints of Mobile Computing:

- Mobile elements are resource-poor relative to static elements.
- Mobility is inherently hazardous.
- Mobile connectivity is highly variable in performance and reliability.
- Mobile elements rely on a finite energy source.
Resource Scarcity or Finite Energy Source:

- The utility of a mobile computer, such as a laptop is largely constrained by battery life.
- The display is a major consumer of battery energy.
- The next major consumers of battery are Hard Disk and Processor (CPU).
Design of Energy-Adaptive Systems
Design of Energy-Adaptive Systems

- The processes running on the system and the display consume energy for their operation.
- So, reducing the energy consumed by display and processor can overcome the constraint of mobile computing to some extent.
Previous approaches to reduce energy consumption:

- Turning off entire display when not in use.
- Using lower-quality or smaller sized displays to minimize power.
Design of Energy-Adaptive Systems

User Study:
- Microsoft Windows environment.
- 17 users.
- Wide range of tasks.

An application level logger program was run on the users’ machines to collect periodic information about:

6. The current window of focus – its size, its location and its title.
7. The size of total screen used.
User Study Results:

- On average, only 59% of the entire screen area is used focused by the user.
- Additional 17% of the screen is used for background windows that are not minimized.
Energy-Adaptive Display System Designs

- Workloads can be classified into:
  2. Access related – web browsing and e-mail.
  3. Personal productivity and code development.
  4. System related and application control windows.
Energy-Adaptive Display System Designs

Proposal based on the following facts:

- Different workloads and users have varying display needs.
- Full screen is not used by the users all the time. Only a part of the screen is usually focused by the user at any point of time.
- Processor is not busy all the time.
Energy-Adaptive System Designs

Current approaches to reduce energy consumption:

- Use of Organic Light Emitting Diodes (OLEDs).
- Use of Software optimizations called *Dark Windows*.
- Use of CPU Scheduling Algorithms.
Energy-Adaptive Display System Designs
HARDWARE LEVEL ADAPTATION:

- At the hardware level, energy consumption can be reduced by use of OLED displays.
- OLEDs consume energy proportional to the overall light output of the display.
Energy-Adaptive Display System Designs

OLEDs are made up of organic molecules which emit light when stimulated by an electric field.

Energy-Adaptive Display System Designs

- The energy consumed by a pixel is dependent on the color being displayed as well as its brightness.
- Different regions of the screen have different energy consumptions.
Energy-Adaptive Display System Designs

SOFTWARE LEVEL ADAPTATION:

• At the software level, energy consumption can be reduced by *Dark Windows* optimization.

• *Dark Windows* enable the windowing environment to change the brightness and color of areas of interest on the screen.

• *Scheduling Algorithms* spread the task in to available time periods so that the task can be completed before its deadline.
Software level optimization can be implemented at several layers:

3. The application level.
4. The window system level.
5. The operating system level.
Energy-Adaptive Display System Designs

Dark Windows Algorithms:

- **HalfDimmed**: Dimmed by 50%
- **FullyDimmed**: Dimmed fully (turned off)
- **GrayScale**: Changed to gray by setting red, green and blue values to the average of three.
- **GreenScale**: Changed to green and its value is set to the average of the three colors and, and the red and blue values are zeroed.
Energy-Adaptive Display System Designs

Prototyping the interface:

Figure: Methodology used to prototype the user interface.
Energy-Adaptive Display System Designs

**Prototyping the interface:** Prototype is implemented on the X Windows System under Linux.

- Open-source VNC (Virtual Network Computing) server.
- VNC provides a virtual representation of the display hardware i.e. a virtual frame buffer.
- Easy to manipulate the pixel values.
- Access to the window server data structures, such as the window of focus.
Energy-Adaptive Display System Designs

Figure 8: Screen shots of the user interfaces with dark windows optimizations.
Energy-Adaptive Display System Designs

Figure: Power benefits from energy-adaptive display designs.
Energy-Adaptive Display System Designs

- Other energy adaptive displays:
  2. Field-Emission Displays
  3. Conventional CRT displays
  4. Hybrid technologies like LCD displays with OLED backlights
  5. “sticky-lamps” at software level adaptation.
Scheduling for Reduced CPU Energy
Scheduling for Reduced CPU Energy

Background:

- Every task (to be precise instruction) requires some fixed number of clock cycles for execution.
- Most of the processors run at maximum speed and execute tasks much before the deadline is reached and enter idle state.
Scheduling for Reduced CPU Energy

- Most of the processors are based on CMOS logic which dissipate an energy (per clock cycle) proportional to the square of the operating voltage.
- Since the processors running at maximum speed require maximum voltage, more energy will be dissipated.
Scheduling for Reduced CPU Energy

Approach:

- Make use of the idle state clock cycles and spread the task to idle state but before the deadline.
- Processor need not run at maximum frequency.
- There by the operating voltage and hence the energy dissipation can be reduced.
Scheduling for Reduced CPU Energy

**Approach:**

- As a result, the operating frequency decreases, tasks take longer time for execution but before the deadline is reached and the energy dissipation decreases.

- **Example:**
  A task has a deadline in 100ms, but takes only 50 ms of CPU time when running at full speed to complete on a normal system.

  If run at half the speed and half the voltage, then the task can be completed at \(1/4^{th}\) the energy consumed by a normal system.
Scheduling for Reduced CPU Energy

- A new metric of interest MIPJ (Millions of Instructions Per Joule) is defined which indicates how many instructions are executed for a given amount of energy.\[\text{MIPJ} = \frac{\text{MIPS}}{\text{WATTS}}\]

- Reducing clock speed causes a linear reduction in energy consumption, but a similar reduction in MIPS. The two effects cancel.
Scheduling for Reduced CPU Energy

Scheduling Algorithms:

- Unbounded-delay perfect future (OPT)
- Bounded-delay limited future (FUTURE)
- Bounded-delay limited-past (PAST)

The above algorithms are simulated for trace data taken from UNIX workstations over many hours of use by a variety of users.
Scheduling for Reduced CPU Energy

Unbounded-delay perfect future (OPT)

- Takes the entire trace, and stretches all the run times to fill all the idle times.
- Impractical: Difficult to predict perfect future.
- Undesirable: Real-time events might not be completed before the deadline.
Bounded-delay limited future (FUTURE)

- Similar to OPT, but peers into the future only a small window.
- Impractical: Depends on future.
- Desirable: No real-time responses are delayed longer than the window.
Scheduling for Reduced CPU Energy

Bounded-delay limited-past (PAST):

- Looks a fixed window into the past and assumes the next window will be like the previous one.
- Practical: Doesn’t depend on future.
- Desirable: No real-time responses are delayed longer than the window.
Scheduling for Reduced CPU Energy

Comparison of all algorithms
Scheduling for Reduced CPU Energy

Dynamic Voltage Scaling Algorithms (DVS):

- These algorithms ensure that the real-time events are executed well before their deadlines.
- These are to be integrated into the popular real-time OS Schedulers like
  - Rate Monotonic (RM)
  - Earliest-Deadline-First (EDF)
References:

References:

- Wilkes, JoIm “Idleness is not Sloth”, to appear proc. of the 1995 Winter USENIX Conf
Design of Energy-Adaptive Systems

Thank You