Goals for Today

- **Learning Objective:**
  - Understand how energy usage informs OS design

- **Announcements, etc:**
  - MP4 due **May 6th**
    - Get started ASAP!
  - HW1 available! Due **May 8th**
    - Just an “appetizer” for the final exam
    - Multiple attempts allowed, but first attempt is graded

**Reminder:** Please put away devices at the start of class
CS 423
Operating System Design: Energy and Power Considerations

Professor Adam Bates
Spring 2018
Why care about energy?

Low-end Computers:
- Resource-constrained battery-operated devices (laptops, phones, wireless sensors, ...)
- Processor speed have grown faster than battery capacity: energy becomes a bottleneck

High-end Computers:
- Cost of energy is increasing: The energy bill is the second highest operational expense of data centers (Google, HP, IBM, ...)

Microprocessor Clock Speed

- Question: Why did the speed curve level off in 2005?
Computational Power (per Die)

- Note the exponential rise in power consumption
Moore’s Law:  Transistor count doubles every two years

Named after Intel co-founder Gordon E. Moore, who described the trend in his 1965 paper
Part of UEFI since 2013:

- Exposes different power saving states in a platform-independent manner
- The standard was originally developed by Intel, Microsoft, and Toshiba (in 1996), then later joined by HP, and Phoenix.
- The latest version is "Revision 6.3" published in January 2019!
ACPI Global States

- **G0**: working
- **G1**: Sleeping and hibernation (several degrees available)
- **G2**: Soft Off: almost the same as G3 Mechanical Off, except that the power supply still supplies power, at a minimum, to the power button to allow wakeup. A full reboot is required.
- **G3**: Mechanical Off: The computer's power has been totally removed via a mechanical switch (as on the rear of a power supply unit).
ACPI Global States

- **G0**: working

- **G1**: Sleeping and hibernation (several degrees available)

- **G2**: Soft Off: almost the same as G3 Mechanical Off, except that the power supply still supplies power, at a minimum, to the power button to allow wakeup. A full reboot is required.

- **G3**: Mechanical Off: The computer's power has been totally removed via a mechanical switch (as on the rear of a PSU).
ACPI “Sleep” States

C-States:

- **C0**: is the operating state.
- **C1** (often known as Halt): is a state where the processor is not executing instructions, but can return to an executing state instantaneously. All ACPI-conformant processors must support this power state.
- **C2** (often known as Stop-Clock): is a state where the processor maintains all software-visible state, but may take longer to wake up. This processor state is optional.
- **C3** (often known as Sleep) is a state where the processor does not need to keep its cache, but maintains other state. This processor state is optional.
C-States:

- **C0**: is the operating state.
- **C1** (often known as Halt): is a state where the processor is not executing instructions, but can return to an executing state instantaneously. All ACPI-conformant processors must support this power state.
- **C2** (often known as Stop-Clock): is a state where the processor maintains all software-visible state, but may take longer to wake up. This processor state is optional.
- **C3** (often known as Sleep) is a state where the processor does not need to keep its cache, but maintains other state. This processor state is optional.
P-States:

- **P0** max power and frequency
- **P1** less than P0, voltage/frequency scaled
- **P2** less than P1, voltage/frequency scaled
- ...
- **Pn** less than P(n-1), voltage/frequency scaled
### Terminology
- $R$: Power spent on computation
- $V$: Processor voltage
- $f$: Processor clock frequency
- $R_0$: Leakage power

### Power spent on computation is:
- $R = k_v V^2 f + R_0$
  where $k_v$ is a constant (relates to capacitance)
Energy of Computation

- Power spent on computation is:
  - $R = k_v V^2 f + R_0$

- Consider a piece of computation of length $C$ clock cycles and a processor operating at frequency $f$

- The execution time is $t = C/f$

- Energy spent is:
  - $E = R \cdot t = (k_v V^2 f + R_0) \cdot (C/f)$
- Power spent on computation is:
  - \( R = k_v V^2 f + R_0 \)

- Energy spent is:
  - \( E = R t = (k_v V^2 f + R_0)(C/f) \)

- Question:
  - Does it make sense to operate the processor at a reduced speed to save energy? Why or why not?
Is reducing processor frequency good or bad?

- Does it make sense to operate the processor at a reduced speed to save energy? Why or why not?

Possible Answer:

\[ E = R \cdot t = (k_v \cdot V^2 \cdot f + R_0) \cdot (C/f) = k_v \cdot V^2 \cdot C + R_0 \cdot C/f \]

- Conclusion: \( E \) is minimum when \( f \) is maximum.
  
  → Operate at top speed

- Is this really true? What are the underlying assumptions?
Reducing voltage and frequency:

- In reality, processor voltage can be decreased if clock frequency is decreased
  - Voltage and frequency can be decreased roughly proportionally.
- In this case (where $V \sim f$):

$$R = k_f f^3 + R_0$$

$$E = (k_f f^3 + R_0)(C/f) = k_f f^2 C + R_0 C/f$$
Reducing voltage and frequency:

- In reality, processor voltage can be decreased if clock frequency is decreased
  - Voltage and frequency can be decreased roughly proportionally.
  - In this case (where $V \sim f$):
    \[
    R = k_f f^3 + R_0
    \]
    \[
    E = (k_f f^3 + R_0)(C/f) = k_f f^2 C + R_0 C/f
    \]
  - Question: Does reducing frequency (and voltage) increase or decrease total energy spent on a task?
Dynamic Voltage Scaling (DVS)

- **Answer:** there exists a minimum frequency below which no energy savings are achieved

\[
E = k_f f^2 C + R_0 C/f
\]

\[
dE/df = 2k_f f C - R_0 C/f^2 = 0
\]

\[
f = \frac{3}{\sqrt{2k_f}} \sqrt{\frac{R_0}{2k_f}}
\]

- **Dynamic Voltage Scaling:** We selectively ‘undervolt’ the processor to maximize power savings (or performance).
Linux defines multiple DVS modes (called CPUfreq "governors"):  
- Performance (highest frequency)  
- Powersave (lowest frequency)  
- Userspace ("root" user controls frequency)  
- OnDemand (adaptively change frequency depending on load)
When should we perform dynamic voltage scaling?

DVS?

Can reduce Frequency, but Voltage is Fixed

When processor is idle, it has option to sleep

No. Run at max frequency.

Can reduce Frequency and Voltage

When processor is idle, it must stay awake

Yes. Run at minimum frequency.

When processor is idle, it has option to sleep

Yes. Find Critical Frequency that minimizes energy...
In the preceding discussion, we assumed that task execution time at frequency $f$ is $C/f$, where $C$ is the total cycles needed.

In reality some cycles are lost waiting for memory access and I/O (Off-chip cycles).

- Let the number of CPU cycles used be $C_{cpu}$ and the time spent off-chip be $C_{off-chip}$.
- Execution time at frequency $f$ is given by $C_{cpu}/f + C_{off-chip}$.
Question

DVS throttles P-States... but how do we know when to sleep?
The Cost of Wakeup

- Turning Processor off...
- Energy expended on wakeup, $E_{\text{wake}}$
- To sleep or not to sleep?
The Cost of Wakeup

- Turning Processor off...
- Energy expended on wakeup, $E_{\text{wake}}$
- To sleep or not to sleep?
  - Not to sleep (for time $t$):
    \[ E_{\text{no-sleep}} = (k_v V^2 f + R_0) t \]
  - To sleep (for time $t$) then wake up:
    \[ E_{\text{sleep}} = P_{\text{sleep}} t + E_{\text{wake}} \]
The Cost of Wakeup

- Turning Processor off...
- Energy expended on wakeup, $E_{\text{wake}}$
- To sleep or not to sleep?
  - Not to sleep (for time $t$):
    \[ E_{\text{no-sleep}} = (k_v V^2 f + R_0) t \]
  - To sleep (for time $t$) then wake up:
    \[ E_{\text{sleep}} = P_{\text{sleep}} t + E_{\text{wake}} \]
  - To save energy by sleeping: \( E_{\text{sleep}} < E_{\text{no-sleep}} \)

\[
t > \frac{E_{\text{wake}}}{k_v V^2 f + R_0 - P_{\text{sleep}}}
\]
The Cost of Wakeup

- Turning Processor off...
- Energy expended on wakeup, $E_{\text{wake}}$
- To sleep or not to sleep?
  - Not to sleep (for time $t$):
    $$E_{\text{no-sleep}} = (k_v V^2 f + R_0) t$$
  - To sleep (for time $t$) then wake up:
    $$E_{\text{sleep}} = P_{\text{sleep}} t + E_{\text{wake}}$$
  - To save energy by sleeping: $E_{\text{sleep}} < E_{\text{no-sleep}}$

$$t > \frac{E_{\text{wake}}}{k_v V^2 f + R_0 - P_{\text{sleep}}}$$

Minimum sleep interval
Dynamic Power Mgmt

- DPM refers to turning devices off (or putting them in deep sleep modes)
- Device wakeup has a cost that imposes a minimum sleep interval (a breakeven time)
- DPM must maximize power savings due to sleep while maintaining schedulability
Question

How does dynamic power management affect scheduling?
The Problem with work-conserving scheduling:

Task 1 (C=2, P=12)

Task 2 (C=1, P=16)
The Problem with work-conserving scheduling:

Task 1 (C=2, P=12)

Task 2 (C=1, P=16)

Minimum sleep period
The Problem with work-conserving scheduling:

Task 1 (C=2, P=12)

Task 2 (C=1, P=16)

Minimum sleep period

No opportunity to sleep! : ( }

Minimum sleep period
The Problem with work-conserving scheduling:

Task 1 (C=2, P=12)

Task 2 (C=1, P=16)

Solution: Must batch!

Minimum sleep period
From the perspective of minimizing energy, is it always a good idea to use up all processors?
How many proc to use?

- Consider using one processor at frequency $f$ versus two at frequency $f/2$
  - Case 1: Total power for one processor
    - $k_f f^3 + R_0$
  - Case 2: Total power for two processors
    - $2 \{k_f (f/2)^3 + R_0\} = k_f f^3/4 + 2 R_0$
How many proc to use?

- Consider using one processor at frequency $f$ versus two at frequency $f/2$

  - Case 1: Total power for one processor
    - $k_f f^3 + R_0$

  - Case 2: Total power for two processors
    - $2 \{k_f (f/2)^3 + R_0\} = k_f f^3/4 + 2 R_0$

- The general case: $n$ processors
  - $n \{k_f (f/n)^3 + R_0\} = k_f f^3/n^2 + n R_0$
How many proc to use?

- The general case: \( n \) processors
  - \( Power = n \left( k_f \left( \frac{f}{n} \right)^3 + R_0 \right) = k_f f^3 / n^2 + n R_0 \)
  - \( dPower/dn = -2 \frac{k_f f^3}{n^3} + R_0 = 0 \)

\[
 n = \sqrt[3]{\frac{2k_f f^3}{R_0}}
\]

- What if \( n \) is not an integer?
Example:

- A processor uses $10 \text{mW}$ when running at full speed and $3 \text{mW}$ when running at half speed. How much energy is saved, if any, at half speed? (If energy, in fact, increases, use a negative sign to indicate “negative savings”)

Dynamic Voltage Scaling (DVS)