Agenda

• Demand for Mobile Computing Devices
• What is Energy-Efficient Software
• Power Basics
• Saving Energy in the CPU – C-States, P-States
• Designing Software to save Energy
• Resources

Intel®, Intel® Core™ 2 Duo, Intel SpeedStep® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States or other countries

*Other names and brands may be claimed as the property of others.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing.
The Quest for Better Battery Life

- Increased adoption of notebook PCs requires focus on longer battery life

- Battery life is a key differentiator in notebook PCs

- Is there a notebook PC that can go all day (8 hours) on a single standard battery charge?

- Approaches to maximize battery life
  - Make better batteries [[attack source of energy]]
  - Make energy efficient hardware [[attack users of energy]]
  - Effectively manage hardware energy states
  - Make energy efficient software

- What innovations are possible? What marketing possibilities are there? What could you do with an all-day battery?
Energy-Efficient Software

- Any software designed to complement energy saving features in the hardware
  - Uses methodologies designed to consume less energy
  - Does not hinder energy saving features of the hardware
  - Maintains power-awareness and alters behavior based on the system’s power state

- [[Opposite: SW to make energy savings worse]]

- Is often a relative metric
  - Applications can have energy efficient properties
  - Compared to what?
  - There is no absolute metric
Power Basics

• Power is the amount of Energy consumed over a period of time and is measured in Watts

• Battery capacities for notebook PCs are rated in Watt-Hours
  – A standard 60 WHr battery with a 20 W load will drain in 3 hours
  – Typical energy states (varies by Notebook PC):
    • Hibernate: 0 W
    • Sleep/Standby: < 5 W
    • Idle: 15 – 20 W
    • Active: 30 – 50 W
CPU Power: P-States

- **P-State** – A CPU Performance State
  - The frequency and voltage of the processor
  - Controlled by the Operating System and in some cases by the hardware

- ‘Adaptive’ power scheme
  - Can lower frequency as appropriate which leads to power benefits
  - Intel SpeedStep® technology

- **P0** – highest CPU performance state – highest energy usage
- **P1** – limited performance – lower power
- **Pn** – minimum performance – minimum power
C-States: CPU Energy/Power States

C-States – CPU Energy States

- **C0** – Core in Use
- **C1** – Core is Halted
- **C2** – Flush L1, Restrict bus use
- **C3** – Clock Generator is Off
- **C4** – Reduced Voltage, Flush L2
- **DC4** – Further reduced Voltage
- **C6** – Core is off (~0 Volts)
## C-States: A little more detail

<table>
<thead>
<tr>
<th></th>
<th>C0</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>DC4</th>
<th>(C6)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Core</strong></td>
<td>Halt Cores</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Microarchitecture state saved)</td>
<td></td>
</tr>
<tr>
<td><strong>Clocks</strong></td>
<td>Disable / gate core clocking</td>
<td></td>
<td>Turn off clock generators</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Caches</strong></td>
<td></td>
<td>Flush L1</td>
<td></td>
<td>Progressively flush L2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Buses</strong></td>
<td>Restrict front-side bus use; disable components</td>
<td>Only break, interrupt, &amp; APCI events; L2-snoops</td>
<td></td>
<td>All processor processing disabled</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Platform</strong></td>
<td>Lower power state of some components</td>
<td>All event &amp; interrupt sensing moved to platform</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Voltage</strong></td>
<td>Barrier</td>
<td>Barrier</td>
<td>Lowered to where only data retention is possible.</td>
<td>Lowered to where no data retention is possible</td>
<td></td>
<td>(~0V)</td>
<td></td>
</tr>
</tbody>
</table>
Developing Energy-Efficient Software

How do I make my software more energy efficient?

• **Computational Efficiency**
  – Get the job done more quickly

• **Data Efficiency**
  – Reduce the frequency and amount of data movement

• **Context Awareness**
  – Know what’s happening and runtime and respond appropriately
Computational Efficiency via Performance

- Get the job done more quickly and allow the CPU to idle
  - Efficient algorithms
  - Multi-threading
  - Advanced instruction sets

- Total energy when running time-bound app with high CPU usage
- Data indicates that for such apps, finishing faster and then going into idle will save power
Example: Data Efficiency – DVD Playback

Improve Power by:
- Reading and buffering DVD content
- Allowing the optical drive to stop; (>70% energy reduction)
- Overall platform savings of 10%
  - 24 additional minutes of battery life!
Example: Power Awareness -- Timers

Many applications use SetTimer() to invoke particular functionality after a specific time interval

- Smaller intervals increase CPU and platform power consumption
- Applications should use the largest timer interval possible
- Reset small interval timers when required task is complete

Use a timer granularity that matches the task
How do I make my Application Energy Efficient?

• **Computational Efficiency**
  – Improve application performance
  – “Get the job done more quickly and allow the CPU to sleep”

• **Data Efficiency**
  – Reduce the frequency/number of disk accesses
  – Employ buffering
  – Read only what you need

• **Context Awareness**
  – Subscribe to platform Power events
  – Respond appropriately to Power events
  – Don’t employ methodologies that impede hardware power saving features

• **Measure the impact of your changes**
  – Use developer tools to see if your changes have made a difference
Resources

• Energy Efficiency white paper
  – softwarecommunity.intel.com/articles/eng/1458.htm
• Tools – Application Energy Toolkit
  – softwarecommunity.intel.com/articles/eng/1631.htm
• For Linux techniques and tools
  – LinuxPowerTop.org
  – Lesswatts.org
Simplified Power Policy Settings in Windows Vista*

- Windows Vista* provides three default power schemes
  - Maximum Power Savings (lower CPU frequency)
  - Automatic (balanced)
  - Maximum Performance (Max CPU frequency)
- The 3 default power schemes are the personalities for all schemes
  - Personality indicates the power saving behavior of the scheme
  - Active power scheme personality can be broadcast to interested components
- Power schemes can be easily discovered and changed by users

Windows Vista* provides easy configuration of Power Schemes
**Energy-Efficiency Checklist**

**Computational Efficiency**
- Multi-thread applications to take advantage of multiple cores
- Tune with SSE 4 for increased performance

**Data Efficiency**
- Pre-fetch data to allow disk drives to go to low-energy states
- Batch writes to drives

**Context Aware (Power Aware)**
- Handle sleep transitions seamlessly
- Respond to system Power events
- Integrate with Windows Vista Power policy and adapt to active policy
P-states versus C-states

Orthogonal
   meaning P-states are independent of C-states

P-state
   A voltage and frequency mark where the processor is operating, e.g. 2.6GHz at 1.5Vdc.
   Measured in C0

C-state
   Core power saving states
   Independent of P-state frequency and voltage

No matter the P-state, the cores descend into C-states the same way
   E.g. A processor in P0 (2.6GHz @ 1.5Vdc) will descend