Faster Resume For Energy Savings on MeeGo

Yoshiya Hirase (yoshiya.hirase@nokia.com)
April 13, 2011
Agenda

• Goal
• Approach
• Resume time optimizations
• Movie 1
• Early-screenshot for better user experience
• Movie 2
• Summary
Goal

Introduce hibernation to handset devices for more energy saving

[use-cases]
- Turn off to maximize battery life while meeting, sleeping, etc
- Start using instantly when out of the package
- Keep using “simple phone mode” even under battery emergency
- Recover application context after battery run out

- Enable more power-off time
- Three seconds for fast resume
- Platform and application independent
**Approach**

- Use TuxOnIce for hibernation
- Make resume faster
  - Speed up boot sequence
  - Minimize snapshot image size
  - Implement as generic as possible

![Diagram showing the process of resume with stages: Bootloader, Linux kernel start-up, Restore system context and time measurement.](Image)
### Resume Process Breakdown

- **Kernel loading**: 4.0%
- **Kernel uncompressing**: 0.9%
- **Device initialization**: 0.6%
- **Swap device wait**: 17.2%
- **Read header**: 2.2%
- **Restore system activities**: 75.1%

Platform: N900 (256MB SDRAM)
Processor: TI OMAP 3430: ARM Cortex-A8 600 MHz
Linux kernel: 2.6.35 (MeeGO for N900) + patch sets
  - Suspend-To-Disk for ARM
  - TuxOnIce for 2.6.35 kernel

Snapshot image size: 23MB (70 % compression)
Snapshot image compression: LZO
Boot device: eMMC mass storage
Resume time: about 10 seconds
Shrink Snapshot Image Size

- Page out as many pages as possible before suspend process starts
  - Take advantage of asymmetric aspect in user experience
    → Resume time is more important for user rather than suspend time
  - Restored with usual page-in mechanism after resume
- Pros
  - Resume time not proportional to the number of running applications
- Cons
  - Slower suspend process
  - Application responses might degrade after resume

- Removing unnecessary memory area
  - Kernel text, data
  - DSPBridge buffer
  - Framebuffer data
How to shrink snapshot image

- Reclaim pages in suspend process as much as possible
  - Modify mm/vmscan.c in order to reclaim pages aggressively
  - Execute shrink_all_memory() repeatedly certain times
- Remove kernel text & data: 4 MB (MeeGo 1.1 for N900, Linux 2.6.35)
  - Reload in kernel boot-up
- Remove a reserved region by DSPBridge: 6 MB
  - Unload before suspend and load after resume
  - Reserved even when the driver is unloaded
- Remove framebuffer area: 16 MB
  - Redraw after resume

<table>
<thead>
<tr>
<th>Linux kernel start-up</th>
<th>TuxOnIce</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device initialization</td>
<td></td>
</tr>
<tr>
<td>Swap device wait</td>
<td></td>
</tr>
<tr>
<td>Read headers</td>
<td></td>
</tr>
<tr>
<td>Read pageset 1</td>
<td></td>
</tr>
<tr>
<td>Restore system activities</td>
<td></td>
</tr>
<tr>
<td>Read pageset 2</td>
<td></td>
</tr>
</tbody>
</table>
**Improve device initialization**

- Remove omap-mcbsp driver initialization from boot process
  - Compile as a module with `CONFIG_SND_SOC=m`
- Device initialization time: about 1.4 seconds

<table>
<thead>
<tr>
<th>Driver</th>
<th>Description</th>
<th>Device initialization time</th>
</tr>
</thead>
<tbody>
<tr>
<td>omap_dss</td>
<td>Display</td>
<td>12.26%</td>
</tr>
<tr>
<td>Serial</td>
<td>Serial (UART)</td>
<td>43.93%</td>
</tr>
<tr>
<td>onenand</td>
<td>NAND flash</td>
<td>18.48%</td>
</tr>
<tr>
<td>omap-mcbsp-dai</td>
<td>Serial sound I/F</td>
<td>16.37%</td>
</tr>
</tbody>
</table>

- Offloaded

---

FasterResumeForMoreEnergySavings_ELC_v0.5Final.ppt/ 2011-04-13 / Yoshiya Hirase
Improve device restore

- WLAN restore is more than half of Restore System Activities
  - 332 devices are restored during resume operation
- Restore System Activities: about 1.5 seconds

<table>
<thead>
<tr>
<th>Device</th>
<th>Description</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>phy0</td>
<td>Physical layer for 802.11</td>
<td>55%</td>
</tr>
<tr>
<td>omapdss</td>
<td>Display</td>
<td>6%</td>
</tr>
</tbody>
</table>

Offloaded
**Improve Memory Transfer**

- Allocate swap space in NAND flash
  - Faster access than eMMC
    - eMMC: < 20MB/s sequential read
    - NAND flash: < 23MB/s sequential read
  - Use mtdblock instead of mtdswap
    - mtdswap is not applicable to hibernation

- Suppress bootloader and kernel messages
  - ‘quiet’ option for Linux
Resume time result

![Graph showing the relationship between system context size (MB) and resume time (s) for TuxOnIce and Optimized. The graph includes two lines: a solid blue line for TuxOnIce and a dashed red line for Optimized. TuxOnIce shows a linear increase in resume time with system context size, while Optimized has a much lower resume time across all context sizes.](FasterResumeForMoreEnergySavings_ELC_v05Final.ppt/ 2011-04-13 / Yoshiya Hirase)
Suspend time result

![Graph showing suspend time result]

- **TuxOnIce**
- **Optimized**

- Linear (TuxOnIce)
- Linear (Optimized)
Snapshot image size shrinking result
Movie: Optimized Snapshot boot
Improve user experience with Early-Screenshot

• Save the last screenshot in suspend process
  • Restore the screenshot immediately in bootloader
• Visual assistance for user
  • Possible to start an action almost right after power-on
• Trade off between user experience and extra cost
  • Overhead in suspend and resume processes
    • Additional 700 ms in bootloader
    • Snapshot image gets bigger (saved as pageset3)
    • Screen data: 750 KB for 800 x 480 pixels, 16bit RGB
• Handover framebuffer settings in resume process
  • Skip part of DSS initialization with CONFIG_FB_OMAP_BOOTLOADER_INIT enabled
    • Not recommended but works
Movie: Snapshot boot w/ early screen image
Summary

- Implement hibernation in a handset device
- Achieved fast resume about 3 seconds
  - cold boot (about 1 minute), TOI (about 10 seconds)
  - Snapshot image size is constant, not proportional to application memory size
- Platform and application independent
THANK YOU
Appendix
TuxOnIce Process Breakdown

75.1% in total resume operation

Swap device wait
Read header
Read pageset1
Restore system activities
Read pageset2

8.25% 0.08% 3.90% 11.42% 29.63% 20.35% 0.05% 0.69% 0.74%

Check snapshot image
Resume
Clean up
Restart system activities
Other

Linux kernel start-up
TuxOnIce
Device initialization in kernel start-up

<table>
<thead>
<tr>
<th>Device</th>
<th>Time / Total (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>usbcore</td>
<td>0.11%</td>
</tr>
<tr>
<td>i2c_omap</td>
<td>0.27%</td>
</tr>
<tr>
<td>twl4030</td>
<td>0.21%</td>
</tr>
<tr>
<td>twl4030_usb</td>
<td>0.17%</td>
</tr>
<tr>
<td>i2c_omap</td>
<td>0.87%</td>
</tr>
<tr>
<td>musb_hdrc</td>
<td>0.08%</td>
</tr>
<tr>
<td>net</td>
<td>0.18%</td>
</tr>
<tr>
<td>dspbridge</td>
<td>0.29%</td>
</tr>
<tr>
<td>omap_dss</td>
<td><strong>12.26%</strong></td>
</tr>
<tr>
<td>acx565</td>
<td>0.11%</td>
</tr>
<tr>
<td>serial</td>
<td><strong>43.93%</strong></td>
</tr>
<tr>
<td>onenand</td>
<td><strong>18.48%</strong></td>
</tr>
<tr>
<td>usbcore</td>
<td>0.45%</td>
</tr>
<tr>
<td>tsc2005</td>
<td>0.85%</td>
</tr>
<tr>
<td>twl4030_wdt</td>
<td>2.01%</td>
</tr>
<tr>
<td>omap-mcbsp-dai</td>
<td><strong>16.37%</strong></td>
</tr>
<tr>
<td>input av jack</td>
<td>0.71%</td>
</tr>
<tr>
<td>alsa</td>
<td>0.47%</td>
</tr>
<tr>
<td>net protocol 17 35</td>
<td>0.60%</td>
</tr>
<tr>
<td>power management</td>
<td>0.92%</td>
</tr>
<tr>
<td>vfp</td>
<td>0.67%</td>
</tr>
</tbody>
</table>

Top four dominates

<table>
<thead>
<tr>
<th>Driver</th>
<th>Description</th>
<th>Operation Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>omap_dss</td>
<td>Display</td>
<td>12.26%</td>
</tr>
<tr>
<td>Serial</td>
<td>Serial (UART)</td>
<td>43.93%</td>
</tr>
<tr>
<td>onenand</td>
<td>NAND flash</td>
<td>18.48%</td>
</tr>
<tr>
<td>omap-mcbsp-dai</td>
<td>MCBSP (serial I/F)</td>
<td>16.37%</td>
</tr>
</tbody>
</table>