IVI Fast boot approach
About Myself

➢ Yuichi Kusakabe (Fujitsu TEN LIMITED)

➢ Software Engineer of IVI about 10 years (for 16-bit and 32-bit architecture)

➢ Linux Software Engineer(2011–2013)
➢ Linux Software Lead Engineer(2013–Now)

➢ BSP Porting/Customizing
➢ Supporting for in-house software developers
Agenda

- IVI Fast boot requirement
- Target Hardware and Software Spec
- Boot time optimization
- Demonstration & Results
- Conclusion
IVI Fast boot requirement
**IVI Requirements**

### IVI Typical Requirements

- **Functional Requirements:**
  - Navigation
  - DTV/DVD/USB-VIDEO
  - HD-RADIO/XM/DAB/RDS
  - Bluetooth/WiFi
  - MirrorLink/Miracast/Carplay
  - Voice Recognition
  - Full Browser
  - Download Apps
  - Back Camera/Image Recognition
  - CAN/MOST/Ether AVB

- **Non-Functional Requirements:**
  - Fast boot (Booting in **2 seconds**)  
  - Protecting system against power outage
  - BSP Update/Security Fix
  - Very Long-Term Support (**7 years**)  

---

*Point: These requirements are quite different from 'Smartphones'.*
Fast boot Requirements

IVI Typical Requirements v.s. Technical Issues (Fast boot)

- Fast boot (Driver/kernel/Middleware/Application)
  - Starting CAN communication: 60 ms
  - Displaying Back Camera Image: 2 sec
  - Playing music: 2 sec
  - Displaying Last Screen Image: 3 sec

Tech Issue

- Starting CAN communications before kernel booting
- Getting ready for user land in 1.5 sec with Cold Start

Renesas R-CAR M2N
- CPU: ARM CA15 (1.5 GHz)
- MEMORY: DDR3 1 GB
- 64 MB QSPI
- SD CARD (SDR50)
Target Hardware and Software Spec
Target Hardware spec

AGL reference Hardware to Renesas R-CAR M2 Porter board

- R-Car M2 SoC
  - ARM® Cortex-A15 Dual Core 1.5GHz
  - Multimedia Engine SH4A 780 MHz
- GPU
  - PowerVR SGX544MP2 (3D)
  - Renesas graphics processor (2D)
- 2 GB DDR3 memory (dual channel)
- Two flash memory chips
  - 4 MB SPI
  - 64 MB SPI
- Debug Ethernet (100 Mbps)
- Storage connection
  - one SATA rev. 3.1 port
  - one SD card slot
  - one microSD card slot
- Analog Video In: ADV7180 Video Decoder
  - RCA jack
  - NTSC/PAL/SECAM autodetection
- Audio codec: AK4643EN
  - Line In 3.5 mm jack
  - LineOut 3.5 mm jack
- Two USB 2.0 ports
  - microUSB port supports host, device and OTG modes
- PCI Express x1 slot
- CAN transceiver

http://elinux.org/R-Car/Boards/Porter

32GB microSDHC

http://panasonic.jp/sd/p-db/RP-SMGB32GJK.html
Target Software Spec

Base soft AGL 2nd release software

Supported Hardware

The AGL Unified Code Base supports a number of reference hardware boards and QEMU emulation. The complete list of supported hardware along with prebuilt binaries and build instructions can be found on this site.

Downloads

- Download the latest source code from git
- Download the latest stable release code
- Pre-built binaries and source tarballs for the latest stable release

Developer Resources

- AGL Wiki Page
- AGL Source Code Repositories

Brilliant Blowfish Features and Benefits

Changes made for Version 2.0 (Brilliant Blowfish) of the AGL UCB include:
- Linux-based distribution updated to use Yocto Project version 2.0
- Common network management as used by GENIVI and Tizen
- Vehicle bus messaging replaced with AGL developed solution for signaling
- Build server moved to faster hardware with Jenkins Job Builder coming on-line in Q3 2016
- A completely revamped test infrastructure utilizing Fuego and Lava to run standard test suites

Beta Features available for testing in the Version 2.0 AGL Staging git tree
- AGL Application Framework with life cycle management and access control that enables application and resource isolation
- Secured automotive bus prevents unwanted intrusions onto MOST, CAN, or other buses
- Audio Routing Plug-in combining the best of GENIVI and Tizen audio management enabling streamlined audio management configurability

https://www.automotivelinux.org/software/download
Target Software Spec

Kernel version is LTSI3.10.31 (Renesas BSP v1.9.2)
Boot time optimization
Fast boot approach

Target kernel and user land boot up time : **1.5 sec**

- **Boot optimization**
  - **Cold boot tuning** ← But I can’t for now
  - Many Hardware dependency
  - **Linux standard Hibernation** ← Adopted
  - **Linux suspend type (disk, mem, ***)**
  - Suspend to **disk**
    - https://www.kernel.org/doc/Documentation/power/swsuspend.txt

- Application Hibernation (hooks) point
  - Process exec and pre-Initialization
  - AGL demo software wake up
### Suspend type “/sys/power/state”

<table>
<thead>
<tr>
<th>state</th>
<th>Dark current (Top priority)</th>
<th>Resume Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>freeze Low-Power Idle</td>
<td>High</td>
<td>Fast</td>
</tr>
<tr>
<td>standby Power-On Suspend</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mem Suspend-to-RAM</td>
<td></td>
<td>△</td>
</tr>
<tr>
<td>disk Suspend-to-disk</td>
<td>Low</td>
<td>Slow</td>
</tr>
</tbody>
</table>

- Freeze Low-Power Idle: High, Fast ×
- Standby Power-On Suspend: ×
- Mem Suspend-to-RAM: △
- Disk Suspend-to-disk: Low, Slow ◎
Suspend to disk detail

- Hibernation image
  - Micro SD card (mmcblk0p2)
    - p1: rootfs area
    - p2: Hibernation image area
  - Image data compression: LZO

- Use command
  - mkswap /dev/mmcblk0p2
  - swapon /dev/mmcblk0p2
  - echo platform > /sys/power/disk
  - echo disk > /sys/power/state
  - resume=/dev/mmcblk0p2
  - U-boot: swsusp mmc 0:2
Normal kernel boot up time (No APL)

Kernel boot up time : 5.6 sec, user land boot up time : 13 sec

[Tue Sep 06 04:56:32.583 2016] Starting kernel ...
[Tue Sep 06 04:56:32.583 2016] Booting Linux on physical CPU 0x0
[Tue Sep 06 04:56:33.550 2016] Linux version 3.10.31-ltsi (kusakabe@localhost.localdomain) (gcc version 4.8.3 20140401 (prerelease) (Linaro GCC 4.8–2014.04) ) #5 SMP PREEMPT Sun Sep 4 02:04:28 JST 2016
[Tue Sep 06 04:56:33.955 2016] Memory: 1417264k/1417264k available, 548816k reserved, 1064960K highmem
[Tue Sep 06 04:56:35.141 2016] sh–sci.6: ttySC6 at MMIO 0xe6e60000 (irq = 184) is a scif
[Tue Sep 06 04:56:35.141 2016] console [ttySC6] enabled
[Tue Sep 06 04:56:35.157 2016] brd: module loaded
[Tue Sep 06 04:56:37.777 2016] ALSA device list:
[Tue Sep 06 04:56:37.777 2016] #0: rsnd–dai.0–ak4642–hifi
[Tue Sep 06 04:56:38.105 2016] kjournald starting. Commit interval 5 seconds
[Tue Sep 06 04:56:38.105 2016] EXT3–fs (mmcblk0p1): warning: mounting fs with errors, running e2fsck is rec
[Tue Sep 06 04:56:38.323 2016] devtmpfs: mounted
[Tue Sep 06 04:56:38.183 2016] Freeing unused kernel memory: 240K (c06c0000 – c06fc000)
[Tue Sep 06 04:56:42.801 2016] dc_linuxfb – Found usable fbdev device ():
[Tue Sep 06 04:56:42.816 2016] [ OK ] Started Automotive Message Broker.
[Tue Sep 06 04:56:46.170 2016] Automotive Grade Linux 2.0.0 porter ttySC6
[Tue Sep 06 04:56:46.170 2016]
[Tue Sep 06 04:56:46.186 2016] porter login: root
Hibernation boot up time (No APL)

Kernel boot up time: 4.2 sec, user land boot up time: 9.7 sec (-3.3 sec)

Add kernel cmdline ‘resume=/dev/mmcblk0p2’

Compressed image size: 50,783,479 byte

kernel boot up: 4.21 sec
Hibernation image load: 3.40 sec (Compressed image size: 50,783,479 byte)
driver resume: 1.456 sec (PM: restore of devices complete after 1457.444 msecs)

user land boot up time: 9.7 sec
Hibernation image load from U-boot (No APL)

Porting “swsusp” command to U-boot, Boot time 12.2 sec (−0.8 sec)

```
[Sun Sep 04 12:15:34.646 2016] => swsusp mmc 0:2
[Sun Sep 04 12:15:40.566 2016] clock = 50000000, buswidth = 4
[Sun Sep 04 12:15:40.582 2016] Allocating 192512 bytes (nr_pfn_pages 47)
[Sun Sep 04 12:15:40.648 2016] Loading image data pages (47126 pages)
[Sun Sep 04 12:15:40.648 2016] Image loading progress:  0%

[Sun Sep 04 12:15:50.952 2016] Image loading progress: 100%
[Sun Sep 04 12:15:50.972 2016] Enabling non-boot CPUs ...
[Sun Sep 04 12:15:50.985 2016] CPU1: Booted secondary processor

U-boot SD read spec (Support HS and DMA)
  => ext4load mmc 0:1 0x40007fc0 boot/zImage
  clock = 50000000, buswidth = 4
  3606064 bytes read in 176 ms (19.5 MiB/s)
```

Hibernation image load: 10.38 sec (Compressed output size: 440,88,359 byte)

user land boot up time: 12.203 sec
Hibernation image load from U-boot (AGL demo)

Porting “swsusp” command to U-boot, Boot time 18.96 sec

Hibernation image load: 17.06 sec (Compressed output size: 72,470,769 byte)  
Very large Image size

user land boot up time: 18.963 sec
Hibernation image size optimization

Hibernation resume speed is dependent on the Hibernation image size

use parameter ‘/proc/sys/vm/drop_caches’ (0, 1, 2, 3)
Default /proc/sys/vm/drop_caches = 0

```
Mem:       1974432     105848    1868584          0       4824      52188
PM: Hibernation image created (32437 pages copied)
PM: Compressed output size: 38431016 [41119744] (imgsize=41115648/swaped size=41119744)
Hibernation image load from SD: 10.61 sec
```

```
echo 1 > /proc/sys/vm/drop_caches
Mem:       1974432     80572    1893860          0         84      31604
PM: Hibernation image created (26078 pages copied)
PM: Compressed output size: 25456529 [27697152] (imgsize=27693056/swaped size=27697152)
Hibernation image load from SD: 7.13 sec
```

```
echo 2 > /proc/sys/vm/drop_caches
Mem:       1974432     104448    1869984          0       4160      51196
PM: Hibernation image created (31797 pages copied)
PM: Compressed output size: 37486193 [40132608] (imgsize=40128512/swaped size=40132608)
Hibernation image load from SD: 10.35 sec
```

```
echo 3 > /proc/sys/vm/drop_caches
Mem:       1974432     80460    1893972          0        104      31592
PM: Hibernation image created (25853 pages copied)
PM: Compressed output size: 25220234 [27447296] (imgsize=27443200/swaped size=27447296)
Hibernation image load from SD: 7.06 sec
```

Best score
Change to Hibernation image to DDR

- Hibernation image
- DDR Hibernation store area
- Kernel: 0x40000000 <-> 0x79000000
- Hibernation: 0x79000000 <-> 0x80000000
- Need to DDR back up

- Use command
  - mkswap /dev/mtdblock3
  - swapon /dev/mtdblock3
  - echo platform > /sys/power/disk
  - echo disk > /sys/power/state
  - U-boot: swsuspmem
Demonstration & Results
Demonstration
Image to SD / DDR
Hibernation (image to DDR) boot up time (No APL)

Kernel and user land boot up time: **2.6 sec (−10.4 sec)**

[Hibernation image load: 0.8 sec (Compressed output size: 44,071,808 byte)]

[Hibernation image load: 0.8 sec (Compressed output size: 44,071,808 byte)]

user land boot up time: **2.6 sec**
Hibernation (image to DDR) boot up time (AGL demo)

Kernel and user land boot up time: 3.23 sec

Hibernation image load: 1.37 sec (Compressed output size: 72,374,707 byte)

Very large Image size

driver resume: 1.499 sec (PM: restore of devices complete after 1499.683 msecs)

user land boot up time: 3.23 sec
## Hibernation image size and Boot time

<table>
<thead>
<tr>
<th></th>
<th>Swap size</th>
<th>Image size</th>
<th>Boot up time (DDR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80,239 pages</td>
<td>72,374,707 byte</td>
<td>3.23 sec</td>
</tr>
<tr>
<td>2</td>
<td>80,179 pages</td>
<td>73,140,769 byte</td>
<td>3.19 sec</td>
</tr>
<tr>
<td>3</td>
<td>80,059 pages</td>
<td>72,854,181 byte</td>
<td>3.22 sec</td>
</tr>
<tr>
<td>4</td>
<td>80,183 pages</td>
<td>73,127,136 byte</td>
<td>3.17 sec</td>
</tr>
<tr>
<td>5</td>
<td>80,362 pages</td>
<td>73,305,227 byte</td>
<td>3.22 sec</td>
</tr>
</tbody>
</table>
Hibernation Results (No APL)

- kernel and user land boot time
  - Normal: 5.6sec(kernel)/13sec(user land)
  - Image to SD: 12.2sec (−0.8 sec)
  - Image to DDR: 2.6sec (−10.4 sec)

- details
  - Image load (SD): 10.4sec (size: 42MB)
  - Image load (DDR): 0.8sec (size: 42MB)
  - Driver resume: 1.4sec
Hibernation Results (AGL demo)

- Kernel and user land boot time
  - Image to SD: 18.9 sec
  - Image to DDR: 3.2 sec

- Details
  - Image load (SD): 17 sec (size: 69 MB)
  - Image load (DDR): 1.4 sec (size: 69 MB)
  - Driver resume: 1.4 sec
Conclusion

- Hibernation is good approach for fast boot
- Hibernation resume speed depends on the image size

Next step
- Fix Hibernation problem
- Optimization of U-Boot SD driver
- Merging
  - Hibernation patch to AGL gerrit
  - Integration AGL software (Goal 2017 CES)
But
I wanna do this!
Other boot time optimize approach

Cold boot: Customizing BSP by SoC Vendor

Starting video playback in 1.6 sec (user space)

https://www.youtube.com/watch?v=N9qLKnLP9pI

R-CAR H2: http://elinux.org/R-Car
Thank you!!!

yuichi.kusakabe@jp.fujitsu.com