HKG15-409: ARM Hibernation enablement on SoCs - a case study

Android Boot time Optimization
Agenda

- Goals
- Preliminary analysis
- Working environment
- ARM hibernation basic support
- Hibernation sequence
- Android kernel - issues
- Debugging
- Time gain from hibernation
- To be done
- References
- code at: https://git.linaro.org/people/grygorii.strashko/p-ti-linux-3.14.y-android-hib.git
Goals

- Reduce Android™ boot time using one of the snapshot save/restore techniques:
  - DMTCP
  - suspend-to-disk (or hibernate)
- Prepare working prototype on TI BeagleBoard-X15 (AM572x) or J6 (DRA7xx) platform using the LSK/Android kernel for further analysis and measurement
- Avoid or minimize changes to Google's AOSP
- security and RPMMsg features descoped
- Target kernel: TI Android kernel 3.14 based on LSK Android 3.14 kernel
Preliminary analysis - DMTCP

- **DMTCP**: Distributed MultiThreaded CheckPointing
  - User space framework
  - AOSP needs to be patched heavily in many places

- DMTCP is out of scope of this presentation
Preliminary analysis - Hibernation - TuxOnIce

- **TuxOnIce** (formerly known as Suspend2, TOI) is an OpenSource implementation of the suspend-to-disk (or hibernate)
- It supports a lot of advanced features
  - eg: “Support for saving a full image of memory”
- It requires basic hibernation support for ARM
- It’s good, !BUT! As is, It was rejected by LKML community see
  - [http://lwn.net/Articles/333007/](http://lwn.net/Articles/333007/)
  
  "So this is an idea to replace our current hibernation implementation with TuxOnIce. Which unfortunately I don't agree with. I think we can get _one_ implementation out of the three, presumably keeping the user space interface that will keep the current s2disk binaries happy, by merging TuxOnIce code _gradually_. No "all at once" approach, please.

  And by "merging" I mean _exactly_ that. Not adding new code and throwing away the old on”

Rafael J. Wysocky [http://lwn.net/Articles/333090/](http://lwn.net/Articles/333090/)
Preliminary analysis - Hibernation - swsusp

- **LKML swsusp/uswsusp** (s2disk) is a combination of Linux kernel and userspace utilities
  - Link: [http://suspend.sourceforge.net](http://suspend.sourceforge.net)
- In general, it provides the same functionality as TOI
- It requires basic hibernation support for ARM

- The final choice for this exercise:
  - **LKML swsusp** + basic ARM-hibernation support for ARM
Working environment

- **HW:** BeagleBoard-X15
- **Android Lollipop:**
  - **manifest:** [http://git.ti.com/cgit/cgit.cgi/android-sdk/platform-manifest.git/](http://git.ti.com/cgit/cgit.cgi/android-sdk/platform-manifest.git/)
  - branch: d-lollipop-release
- **Boot source:** SD card
- **Ref:** [https://cards.linaro.org/browse/CARD-1216](https://cards.linaro.org/browse/CARD-1216)
ARM hibernation basic support 1

- Initially available in LKML 3.16 (19583ca Linux 3.16)
  - Link: https://lkml.org/lkml/2014/3/11/486 (Sebastian Capella/Russ Dill)

- Patches backported from LKML 3.19 (no merge conflicts):
  - 7f8998c "nosave: consolidate _nosave{begin,end} in <asm/sections.h>"
  - 603fb42 ARM: 8011/1: ARM hibernation / suspend-to-disk
  - c7d442f ARM: 8010/1: avoid tracers in soft_restart
  - e26a9e0 ARM: Better virt_to_page() handling
    - This is a preparatory patch to avoid merges, and it is optional. Without this patch, pfn_is_nosave() has to be implemented as in v7 of "ARM hibernation / suspend-to-disk" see https://lkml.org/lkml/2014/3/5/916
ARM hibernation basic support 2

- Minimal kernel configuration with busybox rootfs

- Kernel Kconfig options:
  
  ```
  CONFIG_HIBERNATION=y
  CONFIG_SWAP=y
  CONFIG_PM_TEST_SUSPEND=y
  ```
Test hibernation using swap partition on SD-card:
- create swap partition on SD-card (512M) (can be done on regular PC)
- boot your device
- enable swap
  - `# mkswap <swap partition> -- set up swap area`
  - `# swapon <swap partition> -- enable partition for swapping`
- run smth - memtester can be used
  - `# memtester 256M &`
- hibernate
  - `# echo disk > sys/power/state`
- Add resume parameters to Kernel boot cmd-line in u-boot before 'root=xxx'
  - 'resume=/dev/mmcblk1p2 resumewait' and it has to be done before 'root=xxx'
- boot/reboot/poweron
Android kernel - hibernation sequence

1. power on
2. u-boot
3. kernel
4. Android
5. Lock screen

6. Screen off
7. hibernate
8. power off

9. 10. 11. restore

12. 13. Lock screen

normal boot
hibernate
restore
Some devices do not configure hibernation callbacks properly (especially when dev_pm_ops is used) as result such devices might be kept active during hibernation.

```c
static const struct dev_pm_ops leds_class_dev_pm_ops = {
    .suspend = led_suspend,
    .resume  = led_resume,
};
```

Fixed drivers:
- drivers/led-class.c
- net/ethernet/ti/davinci_mdio
- mach-omap2/omap-device
- drivers/net/ethernet/ti/cpsw.c
- drivers/mmc/host/omap_hsmmc.c
- drivers/hwmon/tmp102.c
- drivers/gpu/drm/omapdrm/omap_dmm_tiler.c
- drivers/gpu/drm/omapdrm/omap_drv.c
Android kernel - issues 2

- Hibernation PM events are not processed by PM notifier handlers
  
  ```c
  static int omap_dss_pm_notif(struct notifier_block *b, unsigned long v, void *d)
  {
    switch (v) {
      case PM_SUSPEND_PREPARE:
        DSSDBG("suspending displays\n");
        return dss_suspend_all_devices();
      case PM_POST_SUSPEND:
        DSSDBG("resuming displays\n");
        return dss_resume_all_devices();
      default:
        return 0;
    }
  }
  ```

- Fixed drivers:
  - drivers/video/fbdev/omap2/dss/core.c
  - work in progress
Android kernel - issues 3

- Contexts of DRA7xx/AM572x SoC’s specific HW modules do not saved restored
  - Control registers
  - PRCM - power and clock management registers
  - Clocks state
    - current implementation provided by TI isn’t working and has to be fixed to support DRA7xx/AM572x SoC
    - The question is what is better - restore registers or set frequencies?
  - there are no visible issues, but only because of Step 11 of hibernation sequence - boot kernel sets up SoC’s registers and restore process doesn’t touch them
PIN control registers:
- Now most of TI drivers setup proper PIN states from their suspend/hibernation/PM runtime callbacks.
  ```c
  static int omap_hsmmc_suspend(struct device *dev)
  {
    ...
    pinctrl_pm_select_sleep_state(host-&gt;dev);
    ...
  }
  static int omap_hsmmc_resume(struct device *dev)
  {
    ...
    pinctrl_pm_select_default_state(host-&gt;dev);
    ...
  }
  ```
- Drivers which do not do that should be fixed or "default" PIN states should be added to Pin controller devices.
It's difficult to enable "Platform" hibernation mode if we want to perform SoC specific actions and power off finally, and our Power controller connected through I2C bus:

- `machine_power_off()` will try to access I2C at time when everything is suspended (including System core and timers) and IRQs are disabled as result huge amount of warnings

```c
int hibernation_platform_enter(void) {
    ...
    local_irq_disable();
    syscore_suspend();
    if (pm_wakeup_pending()) {
        error = -EAGAIN;
        goto Power_up;
    }

    hibernation_ops->enter(); --> machine_power_off() --> palmas_power_off() --> i2c
    /* We should never get here */
    while (1);
    ...
}
```

- In general, we don’t need to implement full set platform callbacks, but only those which are needed for saving/restoring of hibernation image and recovery - for discussion.
Debugging

- printk
  - sys/module/printk/parameters/console_suspend
  - no_console_suspend
- registers comparison before and after hibernation
- PM debug features
  - pm_test

[none] core processors platform devices freezer
  - pm_print_times
    - [3701.890011] calling 2-006f+ @ 3775, parent: i2c-2
    - [3701.898243] call 2-006f+ returned 0 after 3138 usecs

^^^^^^ if 0 or low, something is wrong
Time gain from hibernation

- Obtained with SD card (SD-HC-speed class 4)/eMMC
- Cold boot takes ~45 secs on BeagleBoard-X15
- Time gain from hibernation
  U-Boot SPL 2015.01-rc1-00012-g5695f4a (Nov 12 2014 - 11:51:01)

... 0.000000] Booting Linux on physical CPU 0x0
...
... 3.130244] PM: Hibernation image partition 179:3 present
...
... 3.295254] PM: Loading and decompressing image data (135396 pages).
...
SD: [ 15.765606] PM: Read 541584 kbytes in 12.45 seconds (43.50 MB/s)
eMMC: [ 9.602019] PM: Read 567392 kbytes in 6.18 seconds (91.81 MB/s)
...
[ 147.381443] === func(c055e108)
...
[ 150.175764] Restarting tasks ... done.
+ 4-5 sec to enable screen

Total SD: ~ 24-28 secs
Total eMMC: ~18-20 secs
To be done

- Fix all issues ;P
- Collect more tests results and statistics
- Autosuspend support
- RPMsg
Adding support to other ARM SoCs

- Basic support is present already

- suspend-to-mem should be supported with assumption that HW context will be lost completely
References

- Russell Dill “Extending the swsusp Hibernation Framework to ARM” [http://elinux.org/images/0/0c/Slides.pdf](http://elinux.org/images/0/0c/Slides.pdf) or [https://docs.google.com/file/d/0B-JeSjLTapdOFdwVlhVRjNTUU/edit](https://docs.google.com/file/d/0B-JeSjLTapdOFdwVlhVRjNTUU/edit)
  code: [https://github.com/russdill/linux/commits/arm-hibernation-am33xx](https://github.com/russdill/linux/commits/arm-hibernation-am33xx)
- 0xlab - [https://github.com/0xlab](https://github.com/0xlab), [https://gitorious.org/0xlab-kernel](https://gitorious.org/0xlab-kernel)