Linux Power Management Optimization on the Nvidia Jetson Platform

Merlin Friesen

merlin@gg-research.com
Linux Power Management Optimization on Nvidia Jetson

About You – Target Audience

- The presentation is introductory / intermediate level

- It is intended for any one interested in:
  - Embedded systems
  - System on Chip (SoC) Architecture
  - Linux / ARM power management on the Nvidia Jetson platform
Linux Power Management Optimization on Nvidia Jetson

About Me -- Merlin Friesen

- I have worked for a number of semiconductor companies
  - All developing chips for the cellular / tablet space

- I have lead teams in:
  - Chip validation
    - Pre and Post Silicon
  - System software development

- Currently
  - Founder Golden Gate Research, Inc
  - Linux / wireless consulting
    - cellular / mobile
    - robotics
  - merlin@gg-research.com
Linux Power Management Optimization on Nvidia Jetson

Outline

Overview of the Jetson TX1 Platform

Overview of the Tegra TX1 System on Chip (SoC)

SoC Power Management
- Power Management Unit (PMU)
- Power domains and power islands
- Dynamic Voltage and Frequency Scaling (DVFS)
- Auto clock gating

System Software
- ARM cores
  - cpufreq
  - cpuidle
- Device drivers
  - Power management interfaces

Data Driven Power Management Techniques
Linux Power Management Optimization on Nvidia Jetson

Jetson TX1 Platform

- Tegra TX1 SOM
- TX1 chip
- eMMC Flash 16GB
- Maxim PMU
- DDR4 4GB
- Serial Debug Port

19V Power

- HDMI
- SD/MMC
- Ethernet

- Serial Debug Port
Linux Power Management Optimization on Nvidia Jetson

Jetson TX1 Platform

Jetson ships with Ubuntu installed
- Compilation tools are pre-installed
  - But not recommended
  - Nvidia has a hybrid 32 bit / 64 bit environment
    - The kernel requires both 32 bit and 64 bit tools to compile
  - Compiler differences can make it difficult to get a clean build
- Nvidia has plans to fix this soon

Or you can use your preferred ARM based Linux kernel
Linux Power Management Optimization on Nvidia Jetson

Tegra X1 System on Chip (SoC)

- The Jetson platform is built around the Tegra X1 chip
  - 20nm process
  - 64 bit ARM A57 x 4 with lower power A53 x 4
    - Maximum frequency 1.73 GHz
  - GPU
    - 256 CUDA cores
    - OpenGL 4.5
    - OpenGL ES 3.1
- 4K Video
Linux Power Management Optimization on Nvidia Jetson
Linux Power Management Optimization on Nvidia Jetson

Jetson TX1 Platform

Jetson is a very high end embedded platform
- Compare to other popular embedded platforms
  - Jetson TK1
    - ARM A15 * 5 (32 bit)
  - Raspberry Pi2
    - Cortex A7 * 4 at 900Mhz
  - Beaglebone Black
    - ARM Cortex A8 single core at 1Ghz
Linux Power Management Optimization on Nvidia Jetson

Jetson Platform

It is finding use in high end applications
- Drones
- Vision
- Robotics
Linux Power Management Optimization on Nvidia Jetson

Tegra TX1 System on Chip (SoC)

- Highly integrated cores like this are driving the mobile phone and tablet markets

- The TX1 is in a similar class of mobile devices from:
  - Broadcom
  - MediaTek
  - Qualcomm
  - Samsung

- Given their use in mobile handsets and tablets these devices have state of the art semiconductor power management
Linux Power Management Optimization on Nvidia Jetson

SoC Power Management

Overview: Description of key SoC power Management hardware features

Power Management Unit (PMU)
- The PMU is a discrete Integrated Circuit
- It supplies all the power rails to the SoC
- Jetson TX1 uses the Maxim MAX77620
  - Tegra TK1 communicates with it via I2C bus
    - System software sends commands to it
to change settings on the various power rails
- The device offers us no debug information
  - There are no registers telling us current draw etc.
Linux Power Management Optimization on Nvidia Jetson
Linux Power Management Optimization on Nvidia Jetson

SoC Power Management

Power Domains

- The chip is divided into 4 Power Domains
  - RTC
    - Always on Domain (AOD)
  - Core
    - Peripherals, etc
  - GPU
  - CPU
    - 4 * ARM A57 cores
    - 4 * ARM A53 cores
Linux Power Management Optimization on Nvidia Jetson

SoC Power Management

Power Islands
- Power Domains are in turn divided into Power Islands
- All cores in a Power Island use the same power rail

- Examples of Power Islands
  - CPU
    - Each CPU (1-8) is in a separate power island
    - All handled by the Flow Controller
  - Video (VE)
    - Includes Camera (CSI), Image Sensor Processor (ISP)
    - Video Decode Engine (VDE)

- To turn an island off all the cores in the island must be idle
Linux Power Management Optimization on Nvidia Jetson

SoC Power Management

**Dynamic Voltage and Frequency Scaling (DVFS)**
- Frequency is decreased when possible to reduce power
- Dynamically changing frequency based on the load allows for fine grained power control
- The Tegra TX1 has predefined Frequency / Voltage pairs
  - For example, the ARM processor complex can be set to the following values:

```
pwd
/sys/devices/system/cpu/cpu0/cpufreq

cat scaling_available_frequencies
102000 204000 307200 403200 518400 614400 710400 825600 921600 1036800 1132800 1224000 1326000 1428000 1555500 1632000 1734000
```

- cpufreq uses this capability to reduce frequency (power)
Linux Power Management Optimization on Nvidia Jetson

SoC Power Management

Auto Clock Gating
- Cores are designed to turn off automatically when there is no work
- When the core clock is shut off power consumption is greatly reduced*
- How does this happen?
  - Chip level RTL design tools look at enable signals
    - When the enable is not present the clock driving a block is automatically turned off
  - eg I2C transfers

Thermal Sensing
- Chips now include thermal sensing and cores will be freq reduced or shut down if temperatures get too high
  - This is done to protect the chip
Linux Power Management Optimization on Nvidia Jetson

System Software

Software Controlling ARM Power Management

cpufreq
- Controls frequency / power to the ARM CPU complex
- Voltage / Frequency pairs are defined by the chip manufacturer
  - They can be found in the Device Tree
- cpufreq has pluggable governors

```
pwd
/sys/devices/system/cpu/cpu0/cpufreq
::
: cat scaling_available_governors
interactive conservative ondemand powersave userspace performance
::
: cat scaling_governor
interactive
```
Linux Power Management Optimization on Nvidia Jetson

System Software

cpuidle
- controls what happens when a CPU has no work to perform
- Two governors are available
  - ladder
  - menu
    - main governor in use

WFI
- ARM assembly instruction
- It is used to put the core to sleep
- To sleep the last instruction executed is WFI

```asm
... # Ensure interrupts are enabled for wakeup
wfi # Wait For Interrupt
... # Code executed when core wakes up
```
Tickless idle
- The kernel can be configured to run without the usual scheduler timer tick
- This reduces power consumption as CPUs are not woken up 'x' times / second
- CONFIG_NO_HZ_IDLE=y is used widely by embedded ARM implementations
- The Nvidia Tegra kernel uses it as well:

```
pwd
./proc
:
zcat config.gz | grep CONFIG_NO_HZ_IDLE
CONFIG_NO_HZ_IDLE=y
```
Linux Power Management Optimization on Nvidia Jetson

System Software

Device Drivers

Static Power Management Interfaces

- These are the legacy interfaces called when specific devices are suspended or resumed

- Standard struct used by all device drivers:

  ```c
  struct dev_pm_ops {
    ..
    suspend()          # entry points called by the kernel
    resume()           # on power up and down
    ..
  }
  ```
Linux Power Management Optimization on Nvidia Jetson

System Software

Dynamic Power Management
Runtime PM
- Controls idle for devices (as opposed to just the CPU)
- pm_runtime_get
  - tell the Power Manager that you want to use the core
- pm_runtime_put
  - tell the Power Manager that you do not need the core
- These interfaces use 'use counts' to decide when to shut down a core
- When the use count goes to 0 the core can be shut down
Linux Power Management Optimization on Nvidia Jetson

Data Driven Power Optimization Techniques

Overview

With the hardware and system software ground work laid out we can look at ways to monitor and improve power consumption

- Tools to help us view performance and power

- Interfacing to Jetson TX1 on board power monitors

- Real world examples of power monitoring
Linux Power Management Optimization on Nvidia Jetson

Tools to help us view performance / power
Linux Power Management Optimization on Nvidia Jetson

Tools to help us view performance / power

- ARM Streamline
  - a graphical tool from ARM
  - It is designed to help view ARM performance
  - It collects and displays data, near real time, on a wide variety of system parameters
Linux Power Management Optimization on Nvidia Jetson

Tools to help us view performance / power

- Modified kernel
- gatord daemon
- gator.ko kernel driver
Linux Power Management Optimization on Nvidia Jetson

Tools to help us view performance / power

- Kernel changes required
  - The Nvidia kernel is not configured to run gator
  - CONFIG_PROFILING is not enabled
  - To use the TI Power Monitors
    - I2C needs to be configured as a module
    - Device tree entries required for power monitor chip (TI INA3221)

- A cross compilation environment is recommended
  - Both 32 bit and 64 bit compile tools are required
  - I have used the kernel source on the platform
  - Created the Image and dtb files
    - And a secondary boot configuration
  - No changes to rootfs.

- Specifics are in the backup slides
Linux Power Management Optimization on Nvidia Jetson

Tools to help us view performance / power

- gator
  - The gator driver and the gator daemon run on the target
  - gator collects data near real time & sends this to Streamline
  - Streamline connects to gator via the ethernet port

- gator is open source and available on github
Linux Power Management Optimization on Nvidia Jetson

Tools to help us view performance / power

Onboard power monitors

Overview of the TI INA3221 chip
  - I2C interface
  - Chip has 3 power rail interfaces

- On the SOM board these are monitoring
  - VDD_IN  Tegra X1 main power rail
  - VDD_GPU  GPU power rail
  - VDD_CPU  CPU power rail
Linux Power Management Optimization on Nvidia Jetson

TI INA3221 Power Monitor sysfs interface

```bash
> pwd
/sys/bus/i2c/devices/1-0040/iio:device0
> ls -a
.
..
crit_current_limit_0  in_power0_input  rail_name_1
crit_current_limit_1  in_power0_trigger_input  rail_name_2
crit_current_limit_2  in_power1_input  running_mode
dev
in_current0_input  in_power1_trigger_input  subsystem
in_current0_trigger_input  in_power2_input  uevent
in_current1_input  in_power2_trigger_input  ui_input_0
in_current1_trigger_input  in_voltage0_input  ui_input_1
in_current2_input  in_voltage1_input  ui_input_2
in_current2_trigger_input  in_voltage2_input  warn_current_limit_0
>  
```
Linux Power Management Optimization on Nvidia Jetson

TI INA3221 Power Monitor sysfs interface

```bash
> pwd
/sys/bus/i2c/devices/1-0040/iio:device0
>
> cat name
ina3221x
>
> cat running_mode
1
> 
```
Linux Power Management Optimization on Nvidia Jetson

TI INA3221 Power Monitor sysfs interface

```bash
> pwd
/sys/bus/i2c/devices/1-0040/iio:device0
> cat rail_name_0
VDD_IN
> cat rail_name_1
VDD_GPU
> cat rail_name_2
VDD_CPU
>
```
Linux Power Management Optimization on Nvidia Jetson

TI INA3221 Power Monitor sysfs interface

```bash
> pwd
/sys/bus/i2c/devices/1-0040/iio:device0
> cat in_current0_input
116
> cat in_current1_input
1
> cat in_current2_input
8
>
> pwd
/sys/bus/i2c/devices/1-0040/iio:device0
> cat in_power0_input
2254
> cat in_power1_input
19
> cat in_power2_input
152
> 
```
Linux Power Management Optimization on Nvidia Jetson

Tools to help us view performance / power

- Modified kernel
- gatord daemon
- gator.ko kernel driver
- Streamline annotation task
Linux Power Management Optimization on Nvidia Jetson

ARM Streamline Annotation Task

```c
fdIn = fopen("/sys/bus/i2c/devices/1-0040/iio:device0/in_current0_input", "r");
fdGpu = fopen("/sys/bus/i2c/devices/1-0040/iio:device0/in_current1_input", "r");
fdCpu = fopen("/sys/bus/i2c/devices/1-0040/iio:device0/in_current2_input", "r");

ANNOTATE_SETUP;
ANNOTATE_ABSOLUTE_COUNTER(0, "VDD_IN", "I - Current (ma)");
ANNOTATE_ABSOLUTE_COUNTER(1, "VDD_GPU", "I - Current (ma)");
ANNOTATE_ABSOLUTE_COUNTER(2, "VDD_CPU", "I - Current (ma)");

clock_gettime(CLOCK_MONOTONIC, &ts);

for (;;) {
    // vdd in
    fread(&curr_st_vddin, sizeof(char), 8, fdIn);
    fseek(fdIn, SEEK_SET, 0); // Set to the beginning of the file
    curr_vddin = atoi(curr_st_vddin);

    // vdd gpu
    fread(&curr_st_vddgpu, sizeof(char), 8, fdGpu);
    fseek(fdGpu, SEEK_SET, 0); // Set to the beginning of the file
    curr_vddgpu = atoi(curr_st_vddgpu);

    // vdd cpu
    fread(&curr_st_vddcpu, sizeof(char), 8, fdCpu);
    fseek(fdCpu, SEEK_SET, 0); // Set to the beginning of the file
    curr_vddcpu = atoi(curr_st_vddcpu);

    ANNOTATE_COUNTER_VALUE(0, curr_vddin);
    ANNOTATE_COUNTER_VALUE(1, curr_vddgpu);
    ANNOTATE_COUNTER_VALUE(2, curr_vddcpu);

    ts.tv_nsec += 10000000;
    if (ts.tv_nsec >= 1000000000) {
        ts.tv_nsec -= 1000000000;
        ++ts.tv_sec;
    }
    clock_nanosleep(CLOCK_MONOTONIC, TIMER_ABSTIME, &ts, NULL);
}
```

Linux Power Management Optimization on Nvidia Jetson
Linux Power Management Optimization on Nvidia Jetson

Suspend

```bash
# cd /sys/power
# echo lp1 > suspend/mode
# echo mem > state
```

The term window will now lock up – the K1 is in Suspend state.

The power drops to 16ma.

I then pulled the fan power and it dropped to 0ma.

The fan draws about 16ma
Linux Power Management Optimization on Nvidia Jetson

Low Power States

LP1 or Suspend
- Low Power 1
- VDD_CPU is off
- DRAM memory controller is off
- The DRAM state is maintained using self refresh mode

LP0 or Deep Sleep
- Low Power 0
- VDD_CPU is off
- VDD_CORE is off
  - separate power rail supplied by the PMU
- DRAM memory controller is off
- The DRAM state is maintained using self refresh mode
Linux Power Management Optimization on Nvidia Jetson

Deep Sleep

```bash
# cd /sys/power
# echo lp0 > suspend/mode
#
# echo mem > state
```

Term will now lock up

- To Resume
  - generate an interrupt
  - eg insert SD/MMC card. This will wake CPU up.

- Alternately start a timer
  Which will generate an interrupt
Linux Power Management Optimization on Nvidia Jetson

- memtester running
- 1 process

> memtester 1 &
Linux Power Management Optimization on Nvidia Jetson

- memtester queued up
- 4 processes
Linux Power Management Optimization on Nvidia Jetson

DVFS Example (memtester x 1 running)
Linux Power Management Optimization on Nvidia Jetson

Recap

- We have reviewed the Jetson platform
  - Tegra TX1 capabilities
  - Tegra TX1 power management features

- Linux on Tegra
  - Kernel and device drivers

- We looked at some tools and techniques to monitor and improve power consumption
Linux Power Management Optimization on Nvidia Jetson

Questions ?
Linux Power Management Optimization on Nvidia Jetson

Thank you!

Contact: merlin@gg-research.com