Embedded Linux Conference North America

OpenIoTSummit North America

# Preempt-RT Raspberry Linux

VMware Tiejun Chen <tiejunc@vmware.com>

THE LINUX FOUNDATION

## The declaration of this development

***This is my personal exploration.***

***This is not a roadmap or commitment from VMware.***

## Agenda

- Motivation
- Raspberry Pi platform
- Real Time characteristics & Preempt-RT Linux
- Preempt-RT Raspberry Linux
- Evaluation
- Roadmap

## Motivation

- **IoT** is supposed to be the next big thing.
- **Raspberry pi** is one popular open source hardware.
- **Linux** is more often found on IoT devices and gateways.
- **Linux-based Raspberry pi** is fantastic in the case of IoT.
- But IoT needs to consider Real-Time characteristic.
- A part of my plan of **EPLE** - ***E****nabling* ***P****reempt-RT* ***L****inux* ***E****verywhere*
  - *Preempt-rt rpi Linux*
  - *Preempt-rt Linuxkit*
  - …

## Raspberry Pi overview

- The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards, mice and cases). However, some accessories have been included in several official and unofficial bundles. -- wiki

## Raspberry Pi hardware

- BCM2837: Raspberry Pi 3 and Pi 2 (later)
  - A quad-core ARM Cortex A53 (ARMv8)
- BCM2836: Raspberry Pi 2
  - A quad-core ARM Cortex-A7
- BCM2835: Raspberry Pi 1 and Zero
  - ARM 1176
- **A small and affordable embedded platform that you can do a lot of things, and put into practice in IoT development.**

## Real Time characteristics

- What RTOS needs to be Deterministic
  - Interrupt & Context switch
  - Preemption
  - Latency
  - Jitter
  - Faster ?
- Linux mainline kernel was not designed as RTOS

## Preempt-RT Linux patches

- This series of patches are aimed at converting Linux as the preemptible kernel
  - ~ 100%
  - threaded irq
  - get preempted in critical section
  - quick reaction times!
  - bring latencies down to a minimum **as very possible**
- It’s likely a hard real-time extensions mostly.

## Preempt-RT Linux vs Mainline Linux

- Many features are already merged into mainline.
  - High resolution timers
  - Interrupt threads
  - Mutex
  - …
- Outsider
  - The conversion of spinning locks into sleeping locks

## Raspberry Pi Linux

- raspberrypi/**Linux**
  - Kernel source tree for Raspberry Pi Foundation-provided kernel builds. Issues unrelated to the Linux kernel should be posted on the community forum at https://www.raspberrypi.org/forum
  - https://github.com/raspberrypi/linux

## Raspberry Pi distributions

- **Raspbian**
- Ubuntu Mate, Snappy ubuntu core, Windows 10 IoT Core, …

- Current Linux / BSD kernels for the Raspberry Pi do not provide preempt-rt feature.

## Preempt-RT Linux Source

- The current development -RT tree
  - http://git.kernel.org/pub/scm/linux/kernel/git/rt/linux-rt-devel.git
- The current stable -RT tree
  - https://git.kernel.org/pub/scm/linux/kernel/git/rt/linux-stable-rt.git/
- Released stable -RT patchset
  - https://www.kernel.org/pub/linux/kernel/projects/rt/4.14/

## Raspberry Pi Preempt-RT Linux tree 1/2

- Stage 0: Ready

i. Fork https://github.com/raspberrypi/linux in github

ii. $ git clone https://github.com/TiejunChina/linux

iii. $ git remote add upstream git@github.com:raspberrypi/linux.git

iv. $ git fetch

# Raspberry Pi Preempt-RT Linux tree 2/2

- Stage 1: During development cycle

i. $ git checkout -b rpi-4.14.y-rt rpi-4.14.y

ii. $ git am all preempt-rt patches on rpi-4.14.y-rt  <v4.14.20-rt17>

iii. $ rebase it if any new –rt version is available

- Stage 2: During maintaining cycle

i. $ git am diff between the previous -rt and the new –rt

ii. $ git merge rpi-4.14.y regularly

iii. $ git am any fix specific to rt issue

❖ https://github.com/raspberrypi/Linux/rpi-4.14.y-rt:**v4.14.24-rt19**

## A few issues 1/2

- Rebase some conflicts
  - Like “Revert "softirq: Let ksoftirqd do its job“”
- Lockup
  - “usb: dwc_otg: fix system lockup when interrupts are threaded”
    - Oussama Ghorbel ghorbel@gmail.com

## A few issues 2/2

- Shared IRQ
  - [ 347.337186] Disabling IRQ #59
  - IRQ #59 is shared across the AUX peripherals spi1, spi2 and uart1.

# Example: how to build

- cross-compiler
  - gcc-linaro-7.2.1-2017.11-x86_64_aarch64-linux-gnu
  - https://releases.linaro.org/components/toolchain/binaries/latest/aarch64-linux-gnu/
- aarch64
  - bcmrpi3_defconfig {*bcm2709_defconfig/bcmrpi_defconfig*}
  - *Phil Elwell committed this*
- make rpm

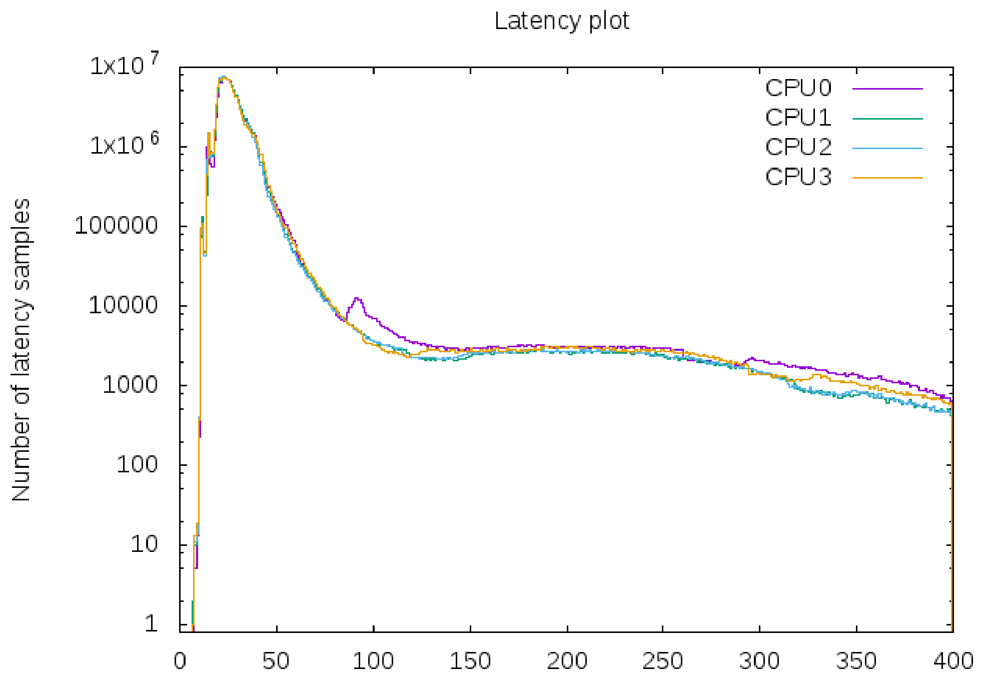
## Evaluation: cyclctest

- Raspberry Pi3 Model B
- Based on openSUSE 64bit
  - http://download.opensuse.org/ports/aarch64/factory/images/
- rt-test
  - https://wiki.linuxfoundation.org/realtime/documentation/howto/tools/rt-tests
  - Cyclictest
  - https://www.osadl.org/uploads/media/mklatencyplot.bash

Embedded Linux Conference North America

OpenIoTSummit North America

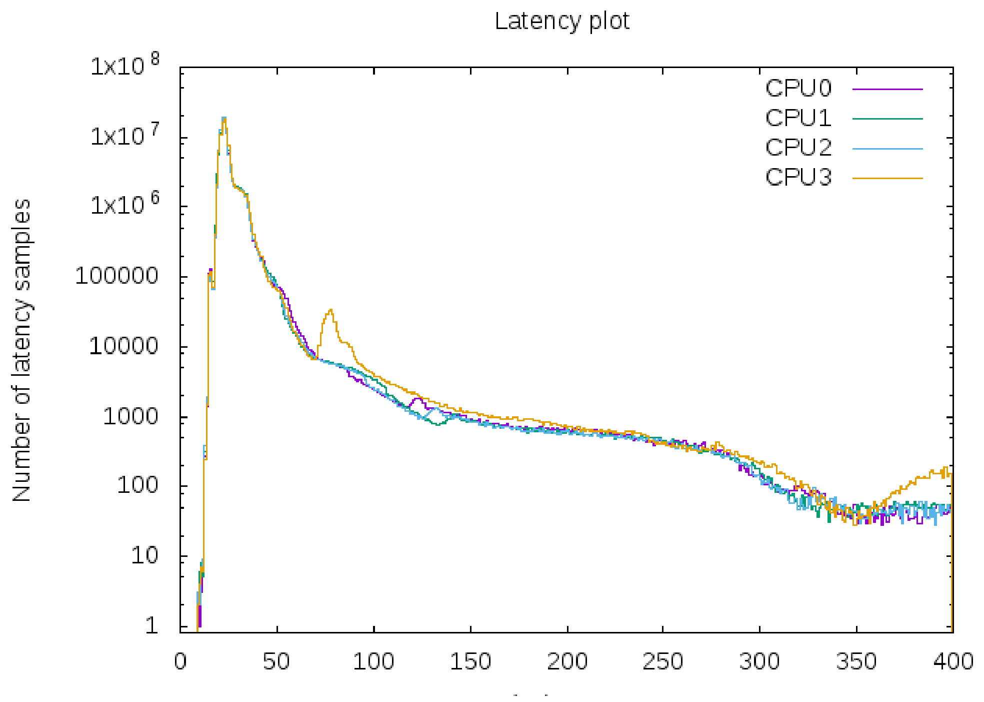
## RPi Preempt-RT Linux: CONFIG_PREEMPT_NONE



- No Forced Preemption
    - There are no guarantees and occasional long delays are possible.
- Latency (us): max 3629 us

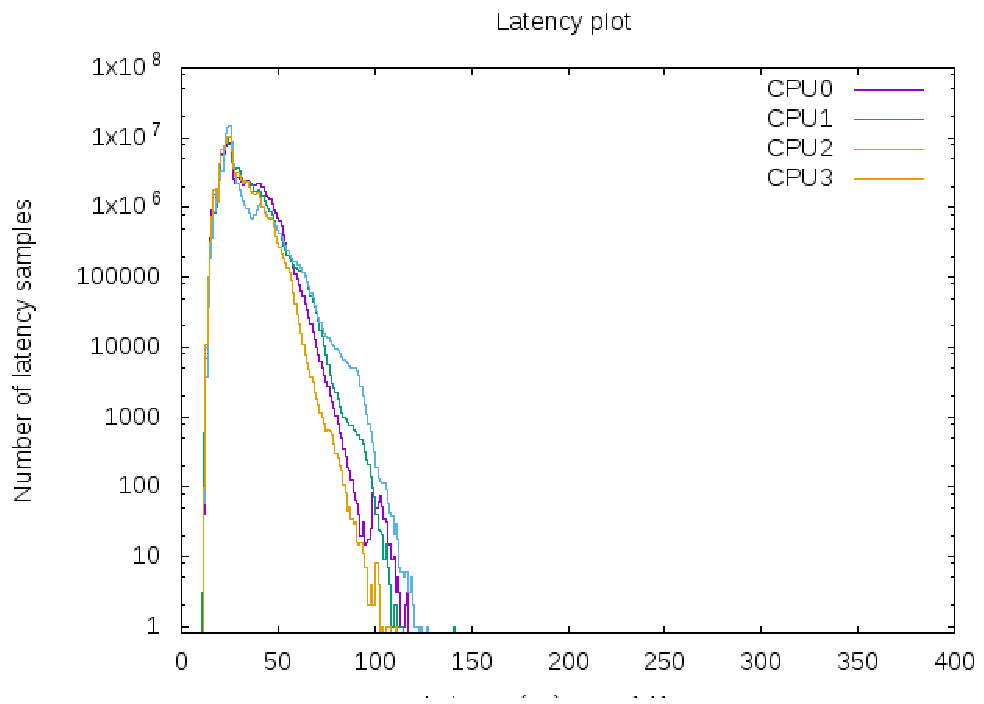
THE LINUX FOUNDATION

## RPi Preempt-RT Linux: CONFIG_PREEMPT__LL



- Preemptible Kernel (Low-Latency Desktop)
  - This option reduces the latency of the kernel by making all kernel code (that is not executing in a critical section preemptible. This allows reaction to interactive events by permitting a low priority process to be preempted involuntarily even if it is in kernel mode executing a system call and would otherwise not be about to reach a natural preemption point.
- Latency (us): max 2582 us

## RPi Preempt-RT Linux: CONFIG_PREEMPT_RT_FULL



- Fully Preemptible Kernel (RT)
  - Everything
- Latency (us): max 141 us

## Test result of RPi Preempt-RT Linux

- Low latency significantly
  - From 3629 us down to 141 us

## Tips 1/2

- Phenomenon
  - Higher CPU utilization from irq

```
PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
70 root      -51   0       0      0      0 S 25.00 0.000   0:32.80 irq/41-3f980000
71 root      -51   0       0      0      0 S 20.00 0.000   0:33.41 irq/41-dwc2_hso
```

  - If we have too much interrupts?

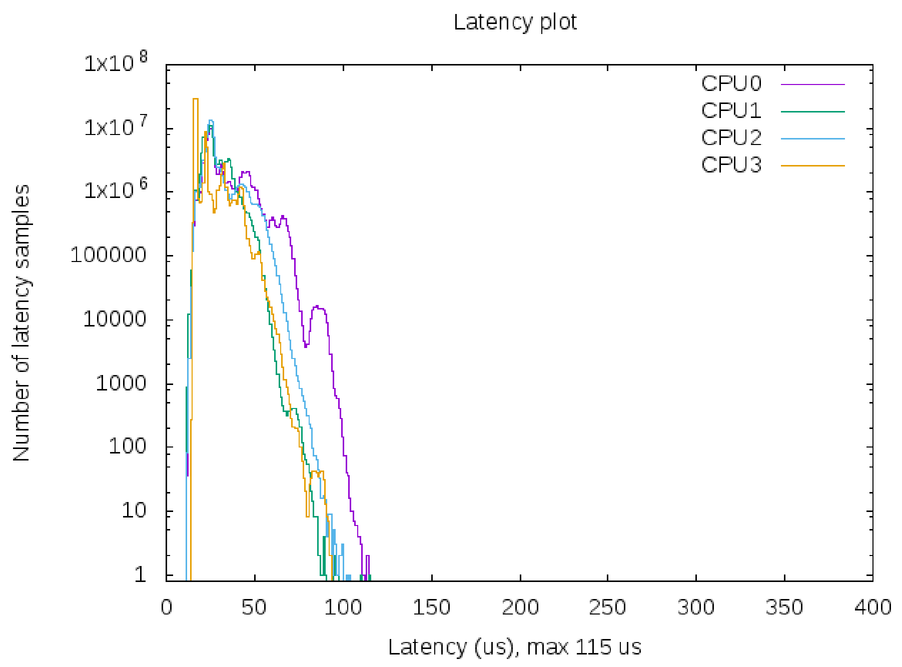
## Tips 2/2

- What we can do now?
  - isolate cpu
    - isolcpus=3
  - dedicate irq to cpu3

```
linux:~ # cat /proc/interrupts
           CPU0       CPU1       CPU2       CPU3
  2:      60182      57004      60988        324  bcm2836-timer   1 Edge      arch_timer
  6:          1          0          0         14  ARMCTRL-level   1 Edge      3f00b880.mailbox
 41:          0          0          0    2017824  ARMCTRL-level  41 Edge      3f980000.usb, dwc2_hsotg:usb1
 71:          0          0          0     237523  ARMCTRL-level  88 Edge      mmc0
```

Embedded Linux Conference North America | OpenIoTSummit North America

## RPi Preempt-RT Linux: CONFIG_PREEMPT_RT_FULL



- Fully Preemptible Kernel (RT)
  - Everything
- Latency (us): max 114 us

## Roadmap

- More platforms
- more tests
  - Ltp, lmbench, posix
- irq affinity dynamically
- virtualization based on preempt-rt Linux
- Unilinux -- unikernelized preempt-rt Linux

Embedded Linux Conference
North America

OpenIoTSummit
North America

THE LINUX FOUNDATION