Comparing Power Saving Techniques for Multi cores ARM Platforms

Vincent Guittot <vincent.guittot@linaro.org>
Linaro Power Management Working Group

Content

- Why to use CPU hotplug ?
- Tests environment
- CPU hotplug constraints
- What we have / What we want
- How to do ?
- Sched_mc_power_saving
- Why these awaking ?
- Gather or spread
- Conclusion

Content

- Why to use CPU hotplug ?
- Tests environment
- CPU hotplug constraints
- What we have / What we want
- How to do ?
- Sched_mc_power_saving
- Why these awaking ?
- Gather or spread
- Conclusion

Why to use CPU hotplug ?

- cpuidle driver managed only CPU0
  - CPU1 must be **unplugged** to reach deep C-states
- Now most of cpuidle driver manages SMP ARM
  - Both CPUs are managed
  - Reach deep C-States in SMP mode
  - Same power level as CPU hotplug
- But
  - Must synchronize the idle sequence
  - Enter/Leave sequences are more complex

Why to use CPU hotplug ?

- CPU hotplug does better
  - Retention rate
  - State transition
  - Power consumption
- Aggressive power saving use cases
  - Low CPU load use cases
- Background activity
  - Voice call
  - MP3 playback
  - ...

Why to use CPU hotplug ?

- MP3 playback on Linaro Android 12.01
  - Not a power optimized SW
- Snowball
  - Retention rate increases from 70 to 74 %
  - ~1mW over 10mW on ARM power rail
  - CPU1 is not power gated
- Panda
  - ~5mW over 37mW on VDD core power rail
  - Which part is for ARM ?
  - CPU1 is power gated

Content

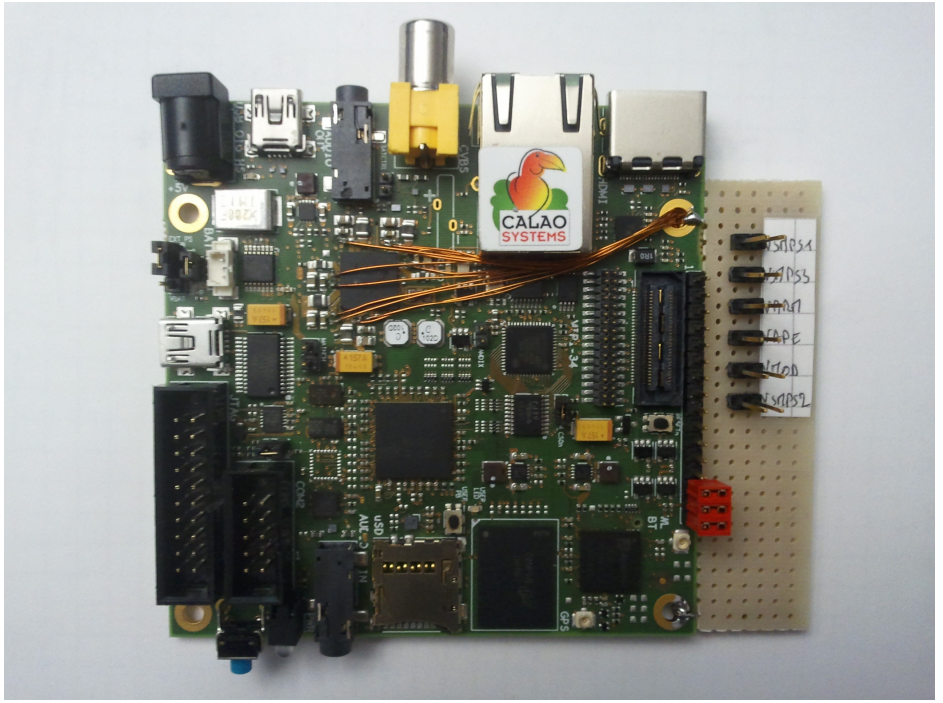
- Why to use CPU hotplug ?
- Tests environment
- CPU hotplug constraints
- What we have / What we want
- How to do ?
- Sched_mc_power_saving
- Why these awaking ?
- Gather or spread
- Conclusion

Tests environment

- We have used
  - Cyclictest
- Simulate multi threaded use case
- Simulate light CPU load use case
- Preferred to low power MP3
  - Easier to tests and reproduce
  - No dependency with optimization
  - No dependency with external devices & firmware

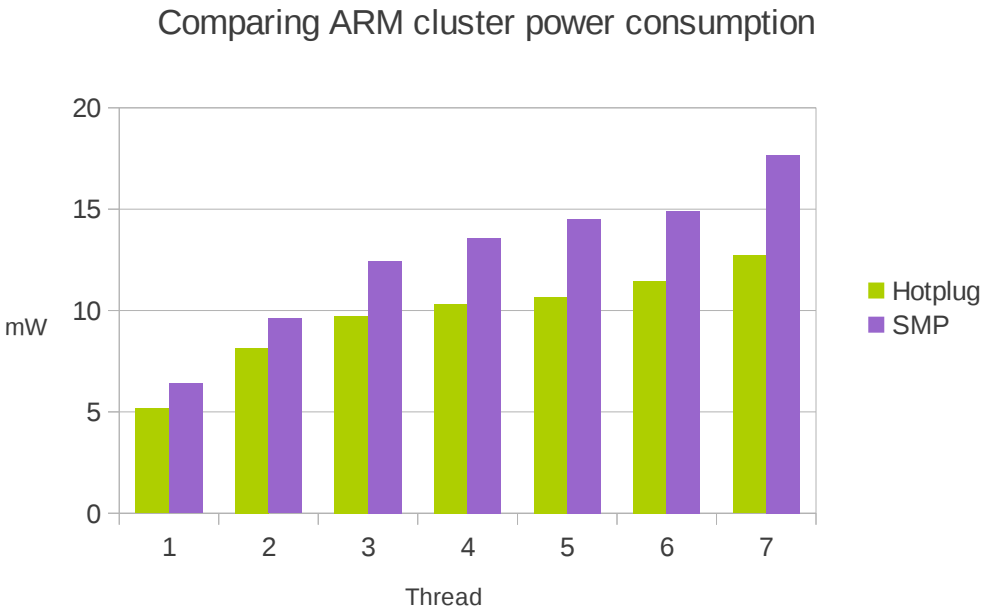
Tests environment

- SW environment
  - Linaro kernel
  - Linaro developer rootfs
  - Cpuidle
  - Cluster retention
- HW environment
  - Dual cortex-A9
  - Snowball board
  - Modified to measure I_{ARM}



Tests environment

- Power consumption figures



Vincent Guittot
Linaro Power Management Working Group

Content

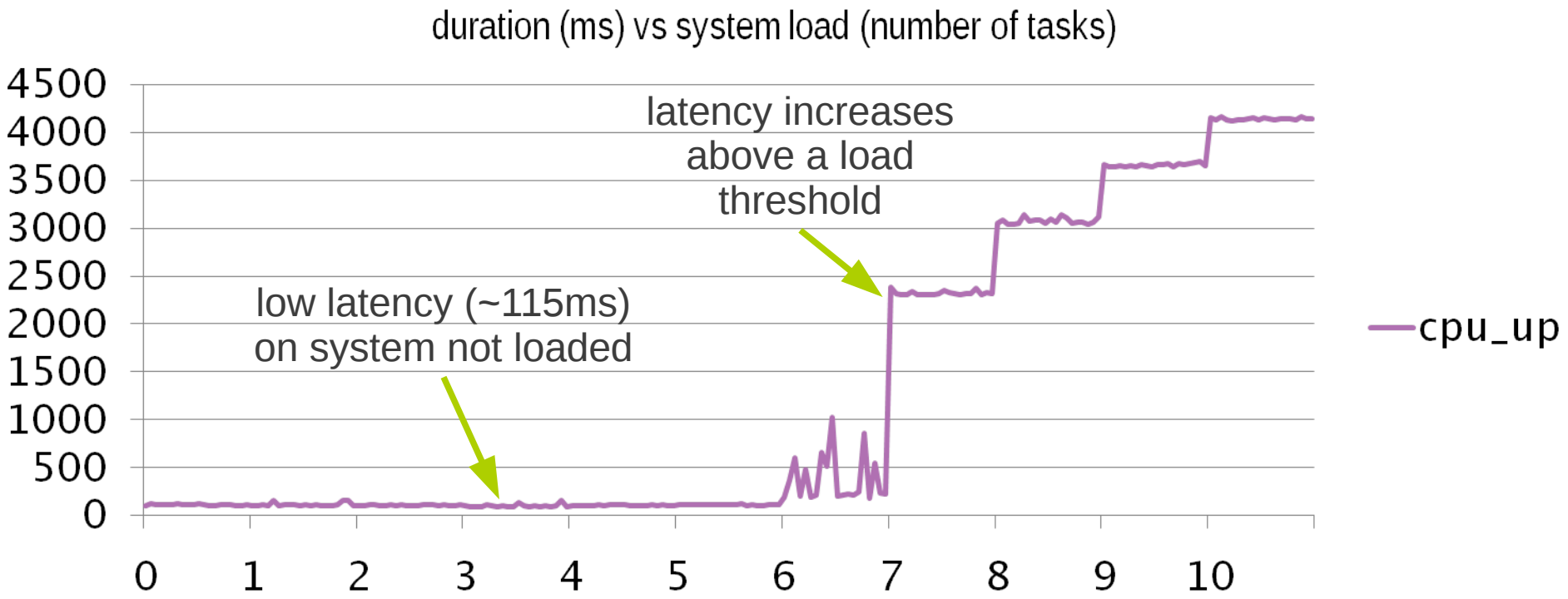
- Why to use CPU hotplug ?
- Tests environment
- CPU hotplug constraints
- What we have / What we want
- How to do ?
- Sched_mc_power_saving
- Why these awaking ?
- Gather or spread
- Conclusion

CPU hotplug constraints

- A power consuming sequence
  - Not only update CPU masks
  - Reset memory structures linked to the CPU
  - Dozens of callbacks for CPU up/down notifications
- Large time scale
  - Compensate the cost of a unplug/plug sequence

CPU hotplug constraints

- A time consuming sequence
  - Create dedicated threads of the CPU



Vincent Guittot
Linaro Power Management Working Group

Content

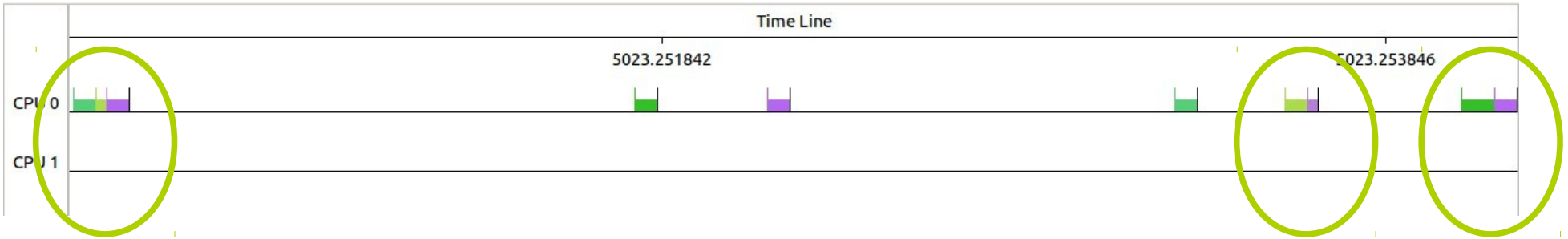
- Why to use CPU hotplug ?
- Tests environment
- CPU hotplug constraints
- What we have / What we want
- How to do ?
- Sched_mc_power_saving
- Why these awaking ?
- Gather or spread
- Conclusion

What we have

- Current SMP behavior
  - Both CPUs are used
  - Not simultaneously most of the time

What we want

- Do not wake up CPU1
  - Let CPU1 in deepest power state
- Do not run on CPU1
  - Neither thread nor interrupt
- Targeted behavior



Vincent Guittot
Linaro Power Management Working Group

Content

- Why to use CPU hotplug ?
- Tests environment
- CPU hotplug constraints
- What we have / What we want
- How to do ?
- Sched_mc_power_saving
- Why these awaking ?
- Gather or spread
- Conclusion

Interruption affinity

- Default interrupt affinity
  - All cores can be used
- ARM gic set affinity
  - cpumask_any_and(mask_val, cpu_online_mask)
  - Take the 1st CPU of the mask
- Unless specific configuration
  - Interruptions raised on CPU0

Scheduler & CPU topology

- Current ARM topology configuration
  - Multi-core topology
- Easy migration on the idle CPU
  - When task wakes up
  - Good for responsiveness/performance
  - Bad for keeping tasks on CPU0

ST ERICSSON

Vincent Guittot
Linaro Power Management Working Group

Linaro

Scheduler & CPU topology

- sched_mc_power_saving mode 2
  - Try to gather tasks in a package
  - Only one package for Cortex-A9 MP !

- Load balance
  - Default CPU capacity is 1 task

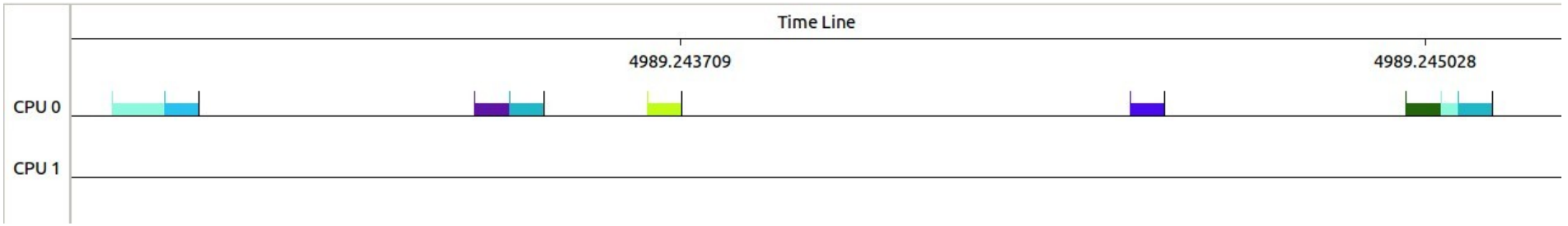
Single core / Multi package

- How to gather tasks on 1 core ?

- Emulate multi-package topology
  - Gather tasks in one package
- Increase cpu_power
  - Keep tasks on one core
- Use asymmetric behavior
  - Use CPU0 in priority

sched_mc_power_saving figures

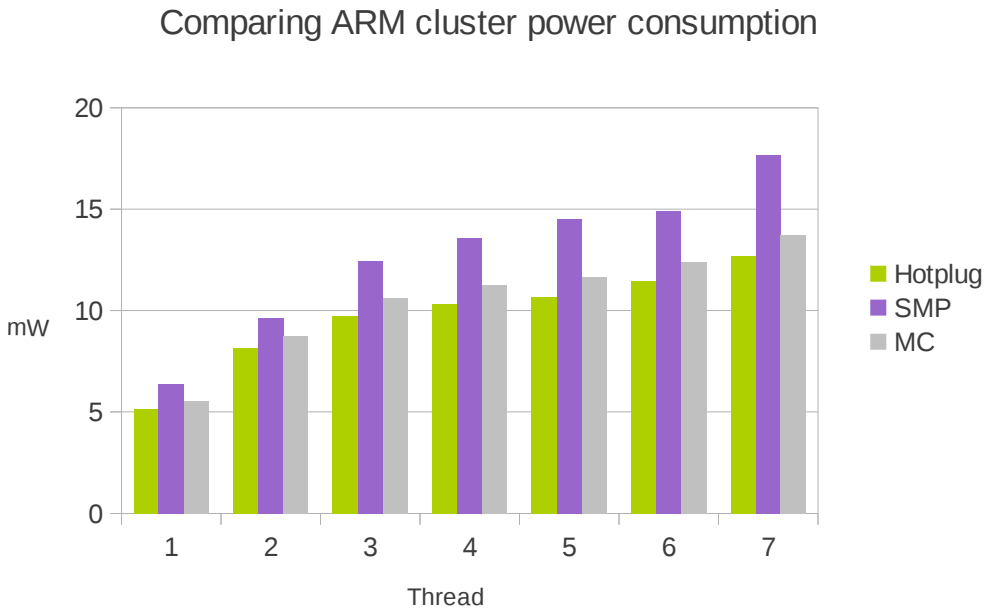
- CPU1 stays idle
  - Nothing is scheduled on CPU1
  - Quite close to CPU hotplug behavior

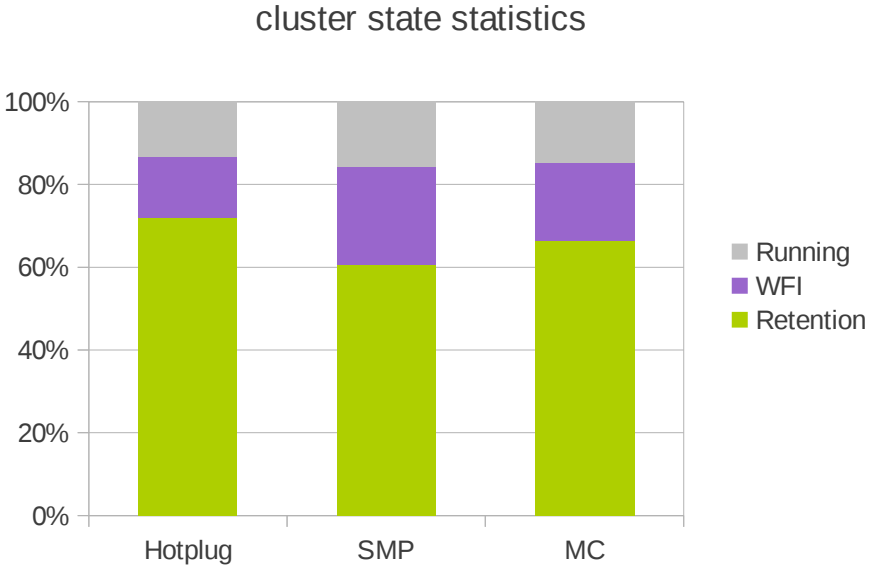


Vincent Guittot
Linaro Power Management Working Group

sched_mc_power_saving figures

- Power consumption figures

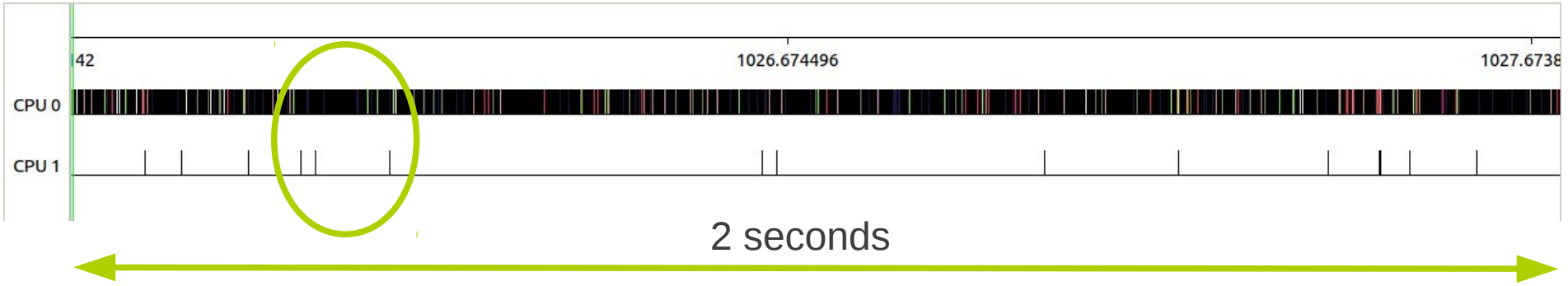




Vincent Guittot
Linaro Power Management Working Group

sched_mc_power_saving figures

- Still differences with CPU hotplug
- Let us look over a bigger period
- Traces show some awaking of CPU1



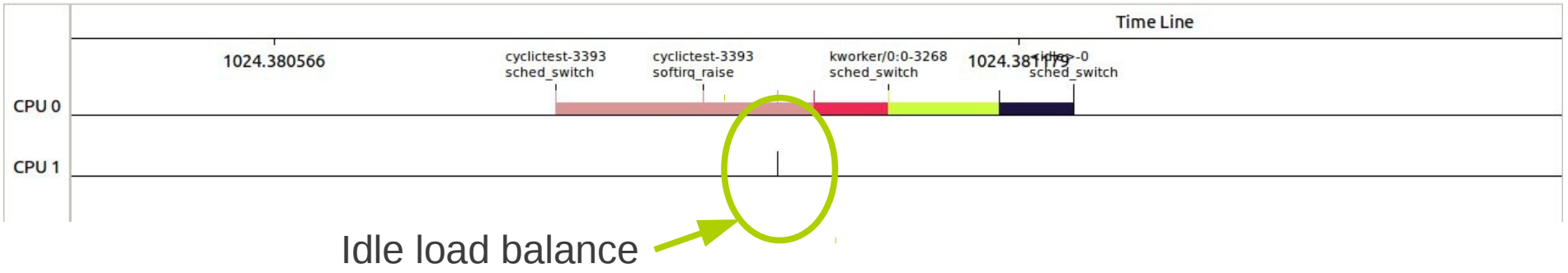
Vincent Guittot
Linaro Power Management Working Group

Content

- Why to use CPU hotplug ?
- Tests environment
- CPU hotplug constraints
- What we have / What we want
- How to do ?
- Sched_mc_power_saving
- Why these awaking ?
- Gather or spread
- Conclusion

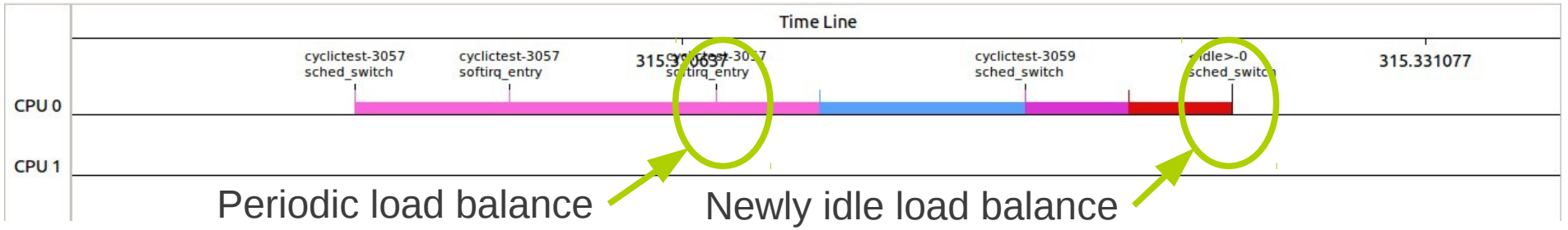
Idle Load Balance

- Idle load balance
  - Run on an idle CPU
  - More than 1 task in the run queue
  - Wake up CPU1



Load balance

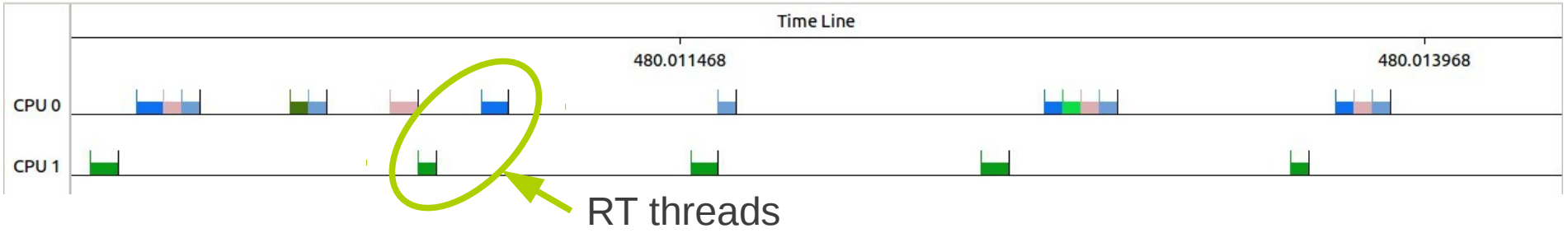
- Load balance
  - Newly idle CPU
  - Periodic (10~256ms)



- Nothing when only 1 CPU is online

Sched_RT

- No power saving policy
- Can break the effort of CFS load balance



- Use cpuset in low power mode
  - Pinned tasks on CPU0 ?

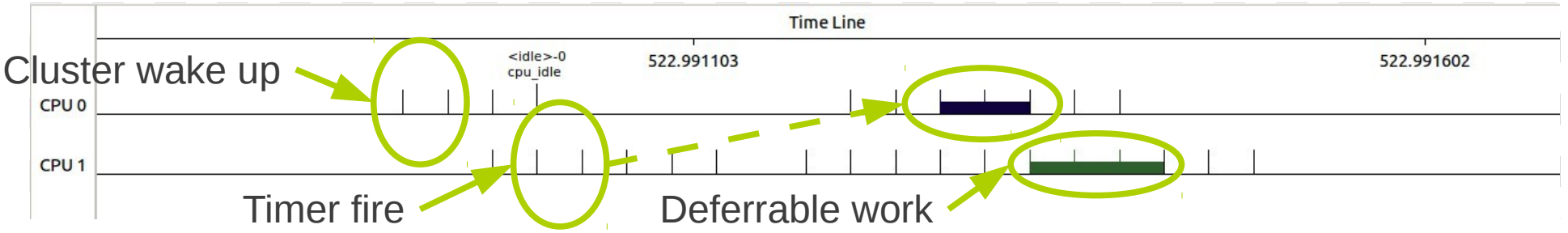
Vincent Guittot
Linaro Power Management Working Group

Deferrable

- Some activities are pinned to CPU1
- Created on cpu_up
- Deferrable
  - Do not wake the CPU
  - Not a problem in itself

Timer

- Timer use current CPU base
  - Unless idle and another CPU is not idle
  - A timer can stay on a CPU



- Force timer migration on CPU0
  - Unplug / plug CPU1

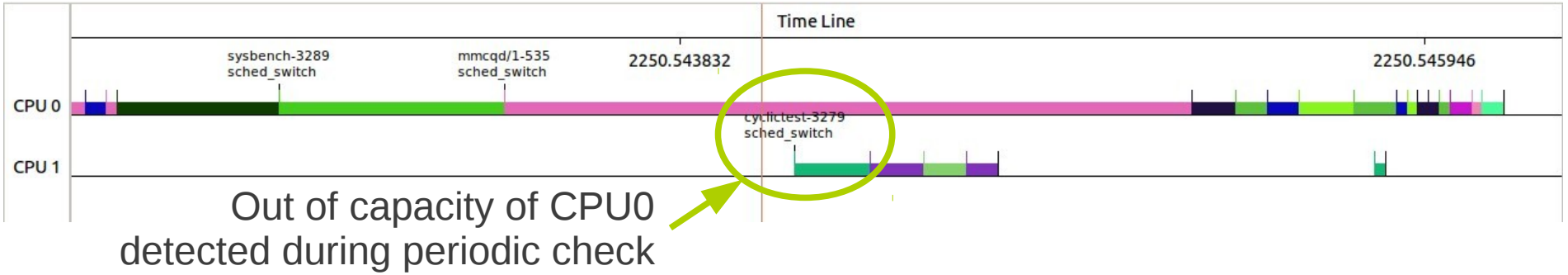
Vincent Guittot
Linaro Power Management Working Group

Content

- Why to use CPU hotplug ?
- Tests environment
- CPU hotplug constraints
- What we have / What we want
- How to do ?
- Sched_mc_power_saving
- Why these awaking ?
- Gather or spread
- Conclusion

Load balance

- Load balance is based on CPU capacity
- CPU0 is out of capacity
  - Some task migrate on CPU1



- CPU0 has capacity
  - Tasks migrate on CPU0

Load balance

- Task migration possibility
  - A newly idle CPU can pull running tasks
  - Idle Load balance triggered by tick
  - Task wakes up on a new CPU
- What about a isolated task
  - Asynchronous to the tick
  - Asynchronous to other tasks

Gather or spread ?

- The most difficult question
- What we have used ?
  - P-state
  - number of running tasks
- May be not enough
  - C-state
  - Others

Content

- Why to use CPU hotplug ?
- Tests environment
- CPU hotplug constraints
- What we have / What we want
- How to do ?
- Sched_mc_power_saving
- Why these awaking ?
- Gather or spread
- Conclusion

Conclusion

- Close to CPU hotplug power consumption
  - On snowball, small overhead on MP3 (<2%)
- Still some issues to solve
  - isolated task
- New mechanisms will help

References

- https://wiki.linaro.org/WorkingGroups/PowerManagement/
- http://git.linaro.org/gitweb
- http://igloocommunity.org/
- https://rt.wiki.kernel.org/articles/c/y/c/Cyclictest.html

QUESTIONS ?

Vincent Guittot
Linaro Power Management Working Group

THANK YOU

Vincent Guittot
Linaro Power Management Working Group

Backup slide

- MP3 sequence