PREEMPT_RT over the years



Sebastian A. Siewior

Linutronix GmbH

September 25, 2024

Using cyclictest

- **cyclictest -S**

```
policy: other/other: loadavg: 18.22 20.48 22.41 35/1617 70975

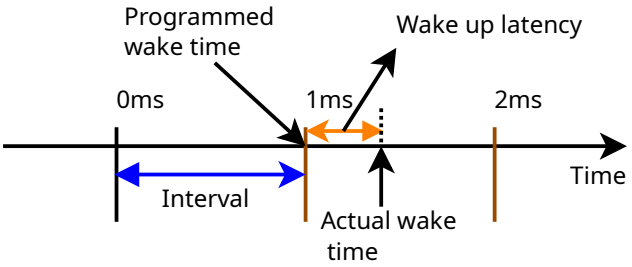
T: 0 (60) P: 0 I:1000 C: 670 Min: 53 Act: 1195 Avg: 172 Max: 11469
T: 1 (61) P: 0 I:1500 C: 457 Min: 53 Act: 104 Avg: 189 Max: 5447
T: 2 (62) P: 0 I:2000 C: 352 Min: 53 Act: 110 Avg: 166 Max: 5948
```

thread (pid) **priority** **interval** **count** **latency in us**

Using cyclictest

- **cyclictest -S**

```
T: 0 (60) P: 0 I:1000 C: 670 Min: 53 Act: 1195 Avg: 172 Max: 11469
```



Using cyclictest + priority

- **cyclictest -S -p 90**

```
policy: fifo: loadavg: 21.57 20.84 22.42 19/1614 70992

T: 0 (77) P:90 I:1000 C: 758 Min: 2 Act: 7 Avg: 26 Max: 101
T: 1 (78) P:90 I:1500 C: 504 Min: 2 Act: 9 Avg: 26 Max: 109
T: 2 (79) P:90 I:2000 C: 379 Min: 2 Act: 9 Avg: 27 Max: 96
```

thread (pid) **priority** **interval** **count** **latency in us**

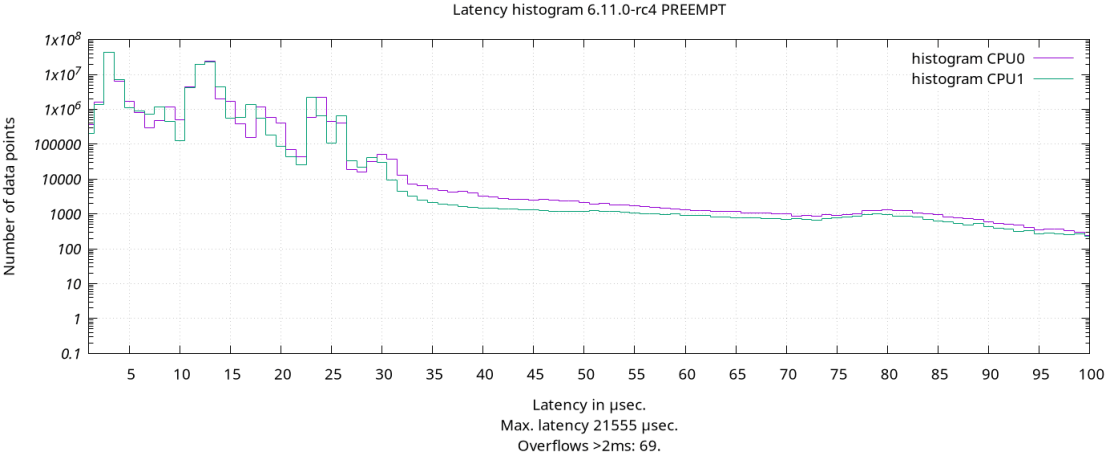
Using cyclictest + priority + interval

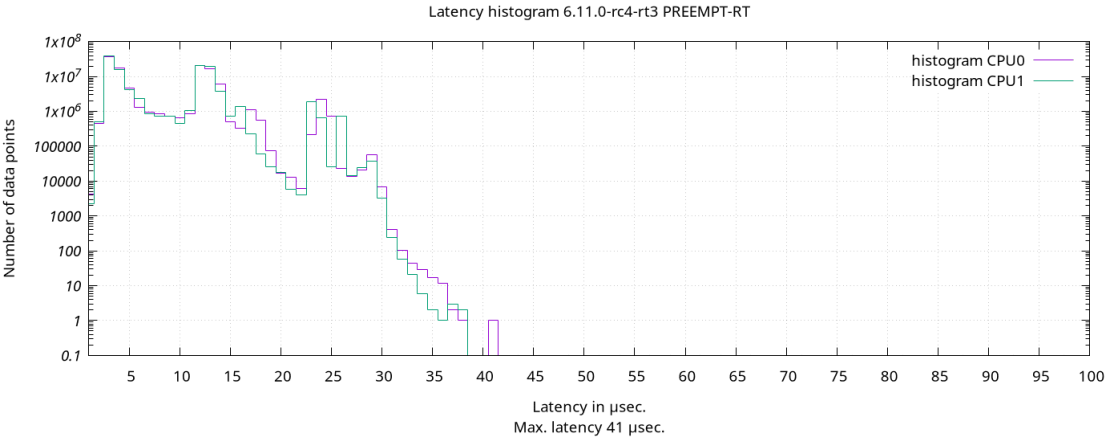
- **cyclictest -S -p 90 -i 250 -d 0**

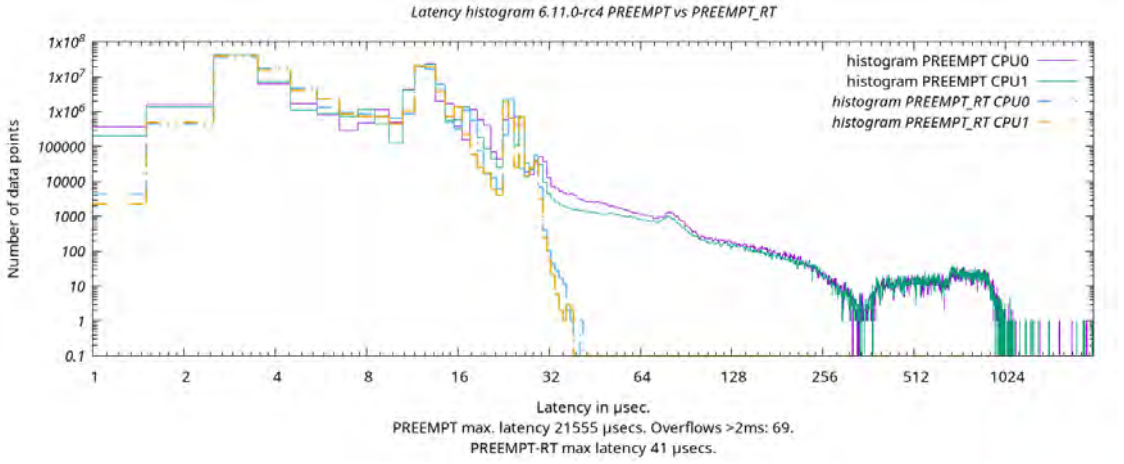
```
policy: fifo: loadavg: 23.28 21.24 22.52 65/1614 71010

T: 0 (95) P:90 I:250 C: 2799 Min: 3 Act: 5 Avg: 6 Max: 40
T: 1 (96) P:90 I:250 C: 2799 Min: 3 Act: 4 Avg: 5 Max: 87
T: 2 (97) P:90 I:250 C: 2799 Min: 3 Act: 6 Avg: 7 Max: 57
```

thread (pid) **priority** **interval** **count** **latency in us**


Latency histogram 6.11.0-rc4 PREEMPT
histogram CPU0
histogram CPU1
Number of data points
1x10^8
1x10^7
1x10^6
100000
10000
1000
100
10
1
0.1
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
Latency in µsec.
Max. latency 21555 µsec.
Overflows >2ms: 69.


Latency histogram 6.11.0-rc4-rt3 PREEMPT-RT
histogram CPU0
histogram CPU1
Number of data points
1x10^8
1x10^7
1x10^6
100000
10000
1000
100
10
1
0.1
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
Latency in µsec.
Max. latency 41 µsec.


Latency histogram 6.11.0-rc4 PREEMPT vs PREEMPT_RT
Number of data points
1x10^8
1x10^7
1x10^6
100000
10000
1000
100
10
1
0.1
1
2
4
8
16
32
64
128
256
512
1024
Latency in µsec.
histogram PREEMPT CPU0
histogram PREEMPT CPU1
histogram PREEMPT_RT CPU0
histogram PREEMPT_RT CPU1
PREEMPT max. latency 21555 µsecs. Overflows >2ms: 69.
PREEMPT-RT max latency 41 µsecs.

What are the requirements for real time?

- High resolution ”time” (clocksource)
- High resolution ”delay” (clockevents, ”oneshot”)
- Prioritize user threads over interrupts
- Locking with priority inheritance
- A quick task scheduler
- Maybe debugging infrastructure

What do we have as of v2.6.0?

- None of the above
- O(n) scheduler in v2.4.
- Scalable scheduler in v2.5.1.10
- Ultra-scalable O(1) SMP and UP scheduler 2.5.2-pre6
- [PATCH] Read-Copy Update infrastructure in v2.5.37-mm1 / v2.6 series

The start of real time

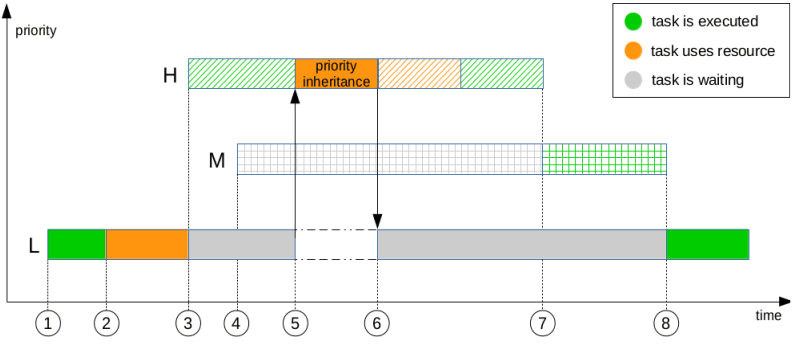
- First announcement: [ANNOUNCE] Linux 2.6 Real Time Kernel by Sven-Thorsten Dietrich 08 Oct 2004 against v2.6.9-rc3.
- The debate began
- Ingo Molnar is working (among other things) on Voluntary Preempt, [patch] VP-2.6.9-rc4-mm1-T5 11 Oct 2004
  - Merged into v2.6.13-rc1 as [PATCH] sched: voluntary kernel preemption
- Ingo started doing realtime preempt [patch] Real-Time Preemption, -VP-2.6.9-rc4-mm1-U0 14 Oct 2004
- Thomas Gleixner picked up [ANNOUNCE] 2.6.15-rc5-hrt2 - hrtimers based high resolution patches 12 Dec 2005, with hrtimers.

The timeline

- lockdep [patch 00/61] ANNOUNCE: lock validator -V1
  - merged into v2.6.18-rc1 as [PATCH] lockdep: core
- Modular Scheduler Core and Completely Fair Scheduler [CFS]
  - against v2.6.21-rc6
  - Merged as ("sched:cfs core code") in v2.6.23
- hrtimer hrtimer - High-resolution timer subsystem
  - merged into v2.6.16-rc1 as [PATCH] hrtimer: hrtimer core code
  - High-Res-Timers (HRT) by George Anzinger in v2.6.13-rc4-RT-V0.7.53-00-realtime-preempt
  - replaced by ktimers by Thomas Gleixner in v2.6.13-rt5
- Futex `LOCK_PI`, priority inheritance
  - appeared first in v2.6.16-rc6-rt1
  - merged into v2.6.18-rc1 as [PATCH] pi-futex: futex_lock_pi/ futex_unlock_pi support

Priority inheritance / PI boost

- **for** `pthread_mutex_lock()` **locks with** `PTHREAD_PRIO_INHERIT`
- **for** `spin_lock()`, `mutex_lock()`, **(not for RW locks)**



The timeline

- genirq [patch 00/50] genirq: -V3 against 2.6.17-rc4, 17 May 2006.
  - appeared first in v2.6.14-rc3-rt1
  - merged into v2.6.18-rc1 as [PATCH] genirq: core
- clockevents/ HIGHRES High resolution timer / dynamic tick update
  - merged into v2.6.21-rc1 as [PATCH] clockevents: add core functionality
- PREEMPTible RCU Real-Time Preemption and RCU (2005)
  - merged into v2.6.25-rc1 as Preempt-RCU: implementation
  - appeared first in v2.6.12-rc1-V0.7.41-06-realtime-preempt (2005)
  - RFC post RCU: Preemptible RCU (2007)

- Tracing ftrace, v16 (JUN 2008)
  - merged into v2.6.27-rc1 as ftrace: add basic support for gcc profiler instrumentation
  - in RT as ”preemption latency trace” since v2.6.9-rc4-mm1-U6-realtime-preempt
  - initial RFC in 2004 mcount tracing utility
  - follow up in JAN 2008 mcount and latency tracing utility -v7
  - FTRACE appeared first in v2.6.24.2-rt2
- Dynamic ftrace whoopsie.
  - e1000e losses firmware e1000e: 2.6.27-rc1 corrupts EEPROM/NVM
  - duct tape in v2.6.27-rc9: e1000e: write protect ICHx NVM to prevent malicious write/erase
  - Source disable CONFIG_DYNAMIC_FTRACE ... has been merged into v2.6.27.1.
  - LWN article

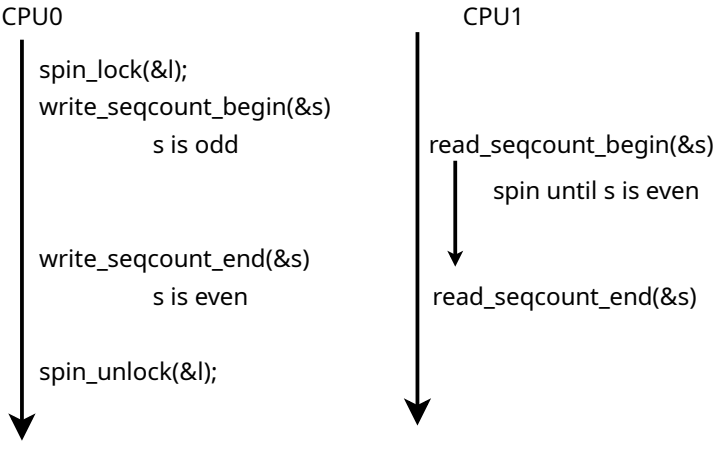
The timeline

- threaded interrupts genirq: add infrastructure for threaded interrupt handlers 01 Oct 2008
  - merged into v2.6.30-rc1 as genirq: add threaded interrupt handler support
  - `git grep request_threaded_irq | wc -l` ⇒ 1129 in v6.11-rc6
- raw_spinlock_t locking: name space cleanup and -rt spinlock annotation
  - merged into v2.6.33-rc1 as locking: Implement new raw_spinlock
- CPU hotplug rework cpu/hotplug: Core infrastructure for cpu hotplug rework
  - merged into v4.6-rc1 as cpu/hotplug: Convert to a state machine for the control processor

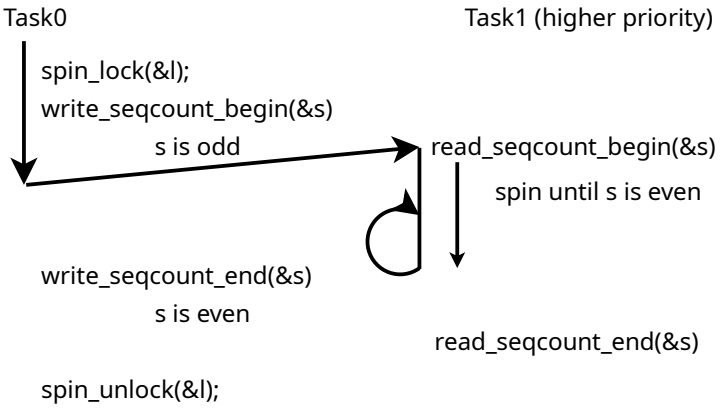
The timeline

- **Decouble** `preempt_disable()` **from** `pagefault_disable()` **[PATCH v1 00/15] decouple pagefault_disable() from preempt_disable()**
  - **merged into v4.8-rc1 as sched/preempt, mm/fault: Decouple preemption from the page fault logic**
- **Non-cascading timer wheel [patch 00/20] timer: Refactor the timer wheel**
  - **merged into v4.8-rc1 as timers: Switch to a non-cascading wheel**
  - **Not in RT first, lowered IRQ-off time.**
- **seqcount_t rework [PATCH v1 00/25] seqlock: Extend seqcount API with associated locks**
  - **merged into v5.9-rc1 as seqlock: Extend seqcount API with associated locks**

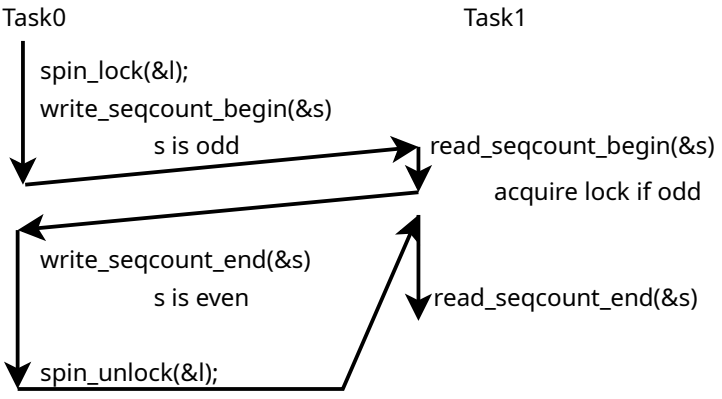
seqcount_t, non-PREEMPT_RT



seqcount_t with PREEMPT_RT



seqcount_t with PREEMPT_RT + rework



The timeline

- migrate_disable() [PATCH 0/9] sched: Migrate disable support
  - merged into v5.11-rc1 as sched: Add migrate_disable()
  - needed due this_cpu: Introduce this_cpu_ptr() and generic this_cpu_* operations since v2.6.33-rc1
  - Not a problem in v2.6.33-RT due to low number of users.
  - in RT since v3.0-rc7-rt0
- local_lock_t [PATCH v3 0/7] Introduce local_lock()
  - merged into v5.8-rc1 as locking: Introduce local_lock()
  - in RT since v3.0-rc7-rt0
- Any context printk ringbuffer printk: replace ringbuffer
  - merged into v5.10-rc1 as printk: add lockless ringbuffer
  - first appeared in v5.0.3-rt1 (using a recursive cpu-sync-lock)
  - first appeared in v5.9.1-rt18 (lockless, as is in mainline today)

Who is using PREEMPT_RT over the years?



WAGO PLC on a DIN-rail



Accessing I/Os over IEC. Max. cycle 150ms.
wago.com

KEBA (injection molding)



Accessing I/Os, Max cycle 500us

KePlast-i-Serie

Engel Victory as a model by Lego (40502)



plastverarbeiter.de

Engel Victory



Mould change

Keba KeMotion/ robotics



Keba cycletime up to 4ms, communicate with I/Os and drive motor

Dürr robots



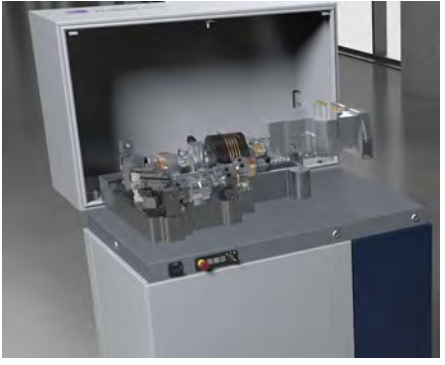
Interior painting at VW Wolfsburg
Kässbohrer PistenBully

Trumpf TruControl



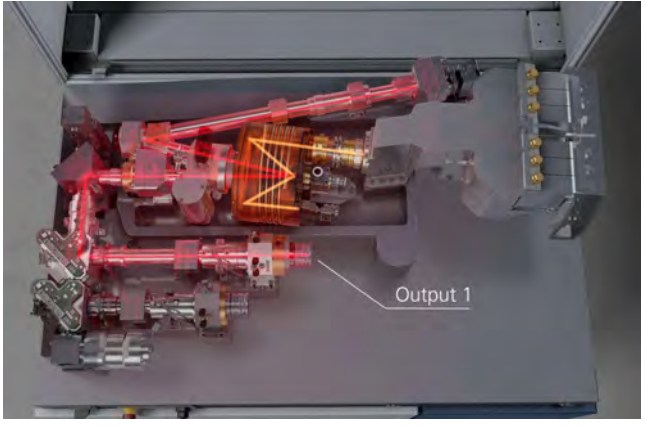
Trumpf

Trumpf TruControl



Trumpf

Trumpf TruControl



Trumpf

Trumpf TruControl (welding)



Trumpf

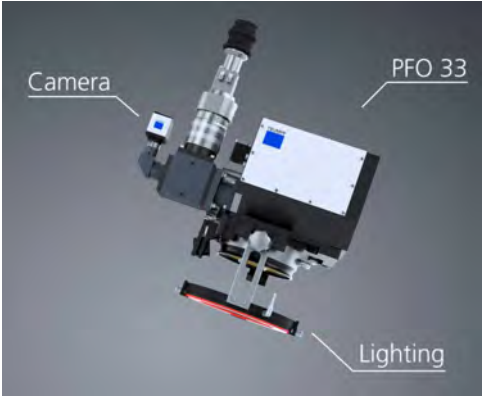
Trumpf TruControl (welding)



Trumpf

Communication over network, less than 2ms to exchange data.

Trumpf VisionLine



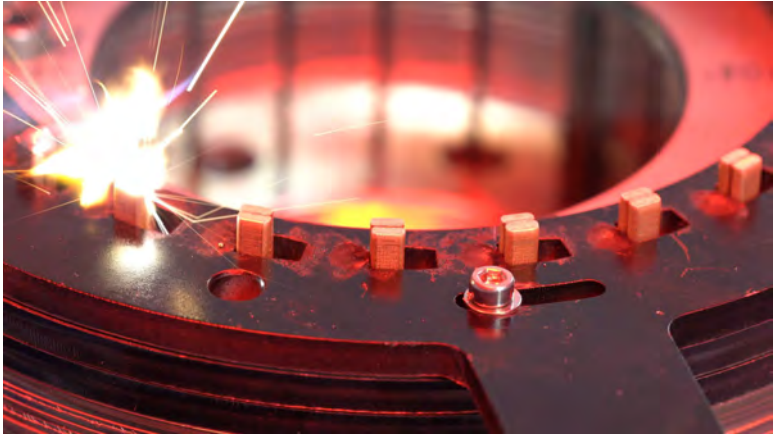
Trumpf

Trumpf VisionLine



Trumpf

Trumpf VisionLine



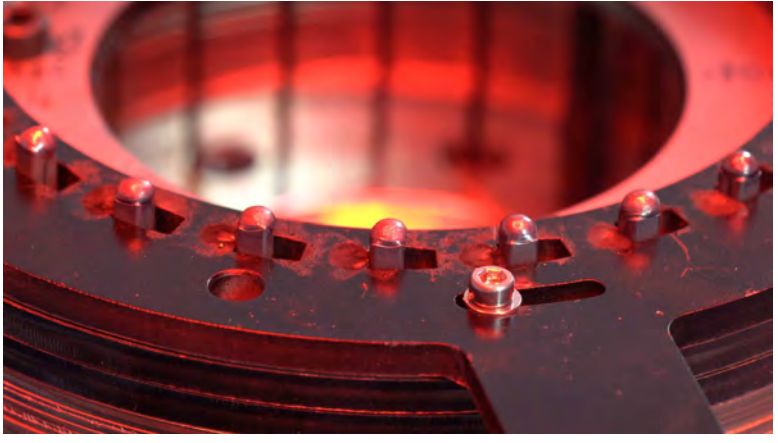
Trumpf

Trumpf VisionLine



Trumpf

Trumpf VisionLine



Trumpf

L-Acoustics L-ISA Processor II, spatial audio for Live



l-acoustics

128 in/ out 96kHz, 3ms round trip latency

L-Acoustics L-ISA Processor II, spatial audio for Live



l-acoustics

L-Acoustics L-ISA Processor II, spatial audio for Live



Running High Channel Count Audio Applications on Linux RT - Olivier Petit - ADC23



Bon Iver

Returns to Live in L-ISA Immersive

Ellips - Wilhelm Weyers



YT

Ellips - Wilhelm Weyers



YT

Ellips - Wilhelm Weyers



YT

Ellips - Wilhelm Weyers



YT

Ellips - Wilhelm Weyers



YT

Ellips - Wilhelm Weyers



YT

Ellips - Wilhelm Weyers



YT

Ellips - Wilhelm Weyers



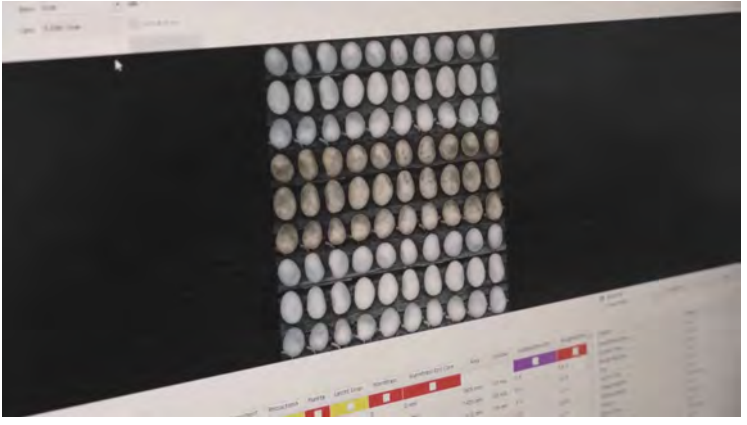
YT

Ellips - Wilhelm Weyers



YT

Ellips - Wilhelm Weyers



YT

Ellips - Wilhelm Weyers



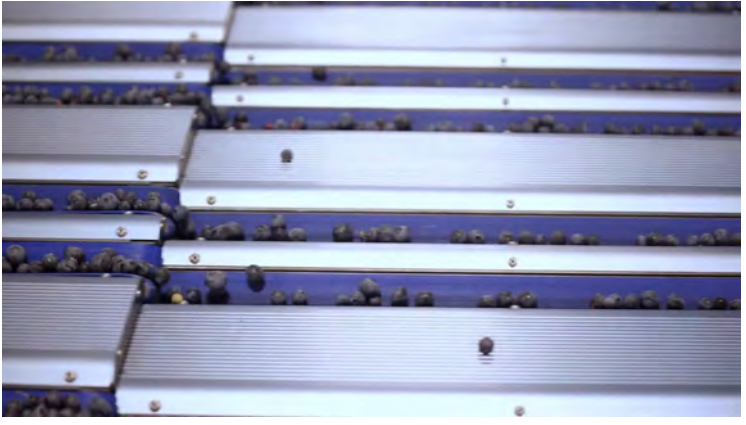
YT

Ellips - Dr B's



YT

Ellips - Dr B's



YT

Ellips - Dr B's



YT

Ellips - Dr B's



YT

Ellips - Dr B's



YT

Ellips

- Vegetables and fruit, from tiny blueberries up to melons.
- 10th gen of Intel processors, Nvidia GPU
- Up to 9 cams per unit and a spectrometer, 10 Gbit ethernet, XDP
- Unit controls up to 40 lanes $\times$ 72 exits $\approx$ 3000 actors on the machine
- Speeds of up to 50 cups per second, accuracy of at 1/10th of a cup, millisecond accuracy in real time threads.
- Missed deadline: missed camera images, bad exit. A few rotten apples in an apple warehouse can make the entire warehouse go bad.

Are we done?



The pull request



The pull request



The pull request



The pull request



The pull request



Next on PREEMPT_RT (printk)

- new thread/atomic (nbcon) console infrastructure
  - first appeared in v5.0.3-rt1
  - atomic part: wire up write_atomic() printing
  - threaded part: add threaded printing + the rest
  - printk for 6.12
- new nbcon drm_log graphic console driver
  - Review drm/log: Introduce a new boot logger to draw the kmsg on the screen
- new nbcon imx uart console driver
  - RFC serial: imx: Switch to nbcon console

Next on PREEMPT_RT

- ARM and PowerPC are still out of tree.
- Finding bad lock constructs. Such as
  - Arnaldo Carvalho de Melo reported 'perf test sigtrap' failing on PREEMPT_RT_FULL Jul 2023
  - Finally addressed perf: Make SIGTRAP and __perf_pending_irq() work on RT. Jul 2024
  - merged into v6.11-rc1 perf: Enqueue SIGTRAP always via task_work.
- Continue on removal of the per-CPU lock in `local_bh_disable()`
  - Work started as locking/local_lock: Add local nested BH locking infrastructure.
  - Avoiding per-CPU locking. Networking is largerst stakeholder.

Trace force-threaded interrupts preempted

```
irq/40-eno0-2034 D...2  681 softirq_raise: vec=3 [action=NET_RX]
irq/40-eno0-2034 ..s.2  681 softirq_entry: vec=3 [action=NET_RX]
irq/40-eno0-2034 d.H.3  690 irq_handler_entry: irq=35
irq/40-eno0-2034 dNH33  692 sched_wakeup: irq/35-ahci prio=44
irq/40-eno0-2034 d.s23  694 sched_switch: prio=49 R+->irq/35-ahci prio=44

irq/35-ahci-837  d..31  696 sched_pi_setprio: irq/40-eno0 prio 49 -> 44
irq/35-ahci-837  d..21  699 sched_switch: prio=44 D->irq/40-eno0 prio=44

irq/40-eno0-2034 d.s34  715 sched_wakeup: iperf3 prio=120
irq/40-eno0-2034 d..21  736 sched_switch: prio=49 R+->irq/35-ahci prio=44

irq/35-ahci-837  D..13  740 softirq_raise: vec=4 [action=BLOCK]
irq/35-ahci-837  ..s.2  740 softirq_entry: vec=4 [action=BLOCK]
```

Trace force-threaded interrupts preempted, patched

```
irq/38-eno0-2006 D...1  032 softirq_raise: vec=3 [action=NET_RX]
irq/38-eno0-2006 ..s.1  032 softirq_entry: vec=3 [action=NET_RX]
irq/38-eno0-2006 d.H.1  033 irq_handler_entry: irq=35 name=ahci
irq/38-eno0-2006 dNH31  034 sched_wakeup: irq/35-ahci prio=44
irq/38-eno0-2006 d.s21  035 sched_switch: prio=49 R+->irq/35-ahci prio=44

irq/35-ahci-842  D..12  038 softirq_raise: vec=4 [action=BLOCK]
irq/35-ahci-842  ..s.1  039 softirq_entry: vec=4 [action=BLOCK]
irq/35-ahci-842  d.s32  041 sched_wakeup: grep prio=120
irq/35-ahci-842  ..s.1  042 softirq_exit: vec=4 [action=BLOCK]
irq/35-ahci-842  d..2.  043 sched_switch: prio=44 S->irq/38-eno0 prio=49

irq/38-eno0-2006 ..s.1  044 softirq_exit: vec=3 [action=NET_RX]
irq/38-eno0-2006 d..2.  051 sched_switch: prio=49 S->swapper/2 prio=120
```

Thank you for your attention

Special thanks to the Linux Foundation for supporting our efforts to bring PREEMPT_RT mainline.



<bigeasy@linutronix.de>