OS Noise in the RT Kernel

Clark Williams, Red Hat
Juri Lelli, Red Hat

OS Noise?

- Things that happen in the operating system and/or hardware that interfere with timing behavior of tasks
  - Interrupts/Exceptions
  - Badly performing drivers/hardware
  - Resource contention
- For discussion, we're including something that is not strictly *OS Noise* (but can be caused by it)
  - Starvation
- These things cause our real-time applications to miss timing deadlines

Interrupts

- Mostly a non-issue due to isolcpus/cpuset isolation and interrupt steering
  - Would be nice to have timers respect cpusets
- X86_64 SMIs are still an issue
  - Thinking we may want to publish a paper as guidelines for RT BIOS options
  - Minimize use of SMI to SMM
  - Don't trap MCEs and translate to SMI

Drivers/Hardware

- Just encountered latency spikes that were due to bad tuning and hardware
- Restarting docker services caused hundreds of microsecond latency's
- Traced to VGA adapter bus-locking during scroll operations
- Mitigated with removing 'console=tty0' and with printk settings
- Later realized customer was using 'nomodeset' boot parameter which put adapter in VGA mode
  - Removed 'nomodeset' and bus-locking behavior gone
- Working on documenting this as another "Don't Do This!" moment

Resource Contention

- Mostly caused by invalid/incorrect isolation
- Usually seen with poll-mode applications (e.g. DPDK)
- Per-cpu kworkers failing to run
- Kernel placing kworkers, RCU threads, others, on *isolated* cpus which then contend with RT application
- Almost always means the customer starts using SCHED_FIFO, many times with a polling app
- This leads to...

Starvation

- SCHED_FIFO cpu-hog taking close to 100% cpu time
    - Kworkers starve
    - RCU stalls
- RT Throttling was original attempt to mitigate this
    - Doesn't help with multiple FIFO threads on same cpu
    - Has a poor time resolution/precision (cannot be used on the us base)
    - All or nothing: might leave CPU idle even with RT tasks ready
- Juri/Daniel working on *deadline server* (original set by Peter)
    - Reimplements RT Throttling using DEADLINE entities
    - Detect starved threads and give them a small boost to make progress
- Prototyped in *starvation_monitor* (user-space) program
    - https://github.com/bristot/starvation_monitor

Discussion/Arguments/Snide Remarks