# Status of 1 year old full dynticks

(aka nohz_full)

LPC 2014

Frederic Weisbecker <fweisbec@redhat.com>

# Merge steps

- Idle dynticks (2.6.21, 2007)
  - Energy
- (nearly)Full dynticks (3.10, 2013)
  - Real time, HPC

# 1 year later : perf events

- Tick needed :
  - freq and throttling event

- Shutdown on other case

- Mostly useful for lockup watchdog

# 1 year later : sysidle detection

- CPU 0 periodic for timekeeping

  = dynticks forbbiden

- RCU Sysidle : adaptive CPU idle dynticks

- Tricky lockless state machine written by Paul McKenney (who else ?)

# 1 year later : sysidle detection (2)

- Needed if powersaving matters for full nohz users

- Full nohz users...

- Not yet plugged

- Complexity : Boot CPU not always 0 = CPU 0 not always nohz full timekeeper

# 1 year later : RCU nocb

- Thread RCU callbacks, migratable

- Written by Paul + Various fixes / maintainance since v3.10

- Only used by Nohz full

# 1 year later : off case optimization

- Distros want it to be available right away (no need to rebuild kernel)

- Off case optimizations

- Static keys (jump labels) all around :
  - Nohz APIs
  - Context tracking APIs
  - Rcu sysidle detection
  - Rcu nocb

# 1 year later : irq work fixes

- Enforce Nohz full depend on irq work self-IPIs
- Fix some nohz kick callbacks called from the tick (!)

# 1 year later : posix cpu timers

- Fixed off case global kick (workqueue broadcast IPI)
- Fixed missing tick kick on timer rescheduling
- Fixlets

# HPC

- 1000 Hz → full dynticks = +2-3 % perf
- 100 Hz → full dynticks = +0.003 % perf

  = a new CPU every 300
- Benchmark used dummy user loop
- Need real world measurement

# Real time

- Extreme real time (no interruption at all, need more work)

- Residual 1 Hz tick

# Future : workqueue affinity

- Isolate unbound workqueues :
  - https://lwn.net/Articles/599346/
- People advertised taking over patchset...
- Per Cpu workqueues : must be checked case by case

# Future : timers affinity

- Unbound timers : people advertised patchset but never posted
- Per Cpu timers : case by case

# Future : scheduler

- Audit scheduler_tick() and sched_class::task_tick() before removing 1 Hz residual

- Hrtick if full dynticks goes somewhere near long term

# Overall complexity added

- RCU nocb
- RCU sysidle
- RCU User QS
- Tickless cputime accounting (vtime gen)
- Context tracking (+ arch hooks : user_enter()/user_exit(), exception_enter()/exception_exit())
- Nohz core
- Overall : Large and tricky code, sensitive, fragile, very few qualified reviewers

# Question

- Who uses Nohz Full ?
- Is it sensible to maintain this large core codebase for 1 (or none?) users ?
- Wait and see ?

# Special Thanks

- Paul MckKenney for RCU related work
- Peter Zijlstra for regular reviews
- Thomas and Ingo for merging
- ...