Per Core Idle Injection

Srinivas Pandruvada
Intel
Use Case

• Clients
  Thermal Control
  Low power mode “Power saver” mode
• Server
  Provide higher frequencies in fewer cores
Requirements

• Should be able to run on a Linux distro
• Per CPU Thermal control
• Allow to keep fewer CPUs active
  Don't break affinity by using online/offline
  Fast transition in and out of mode
Idle Injection in Linux

- Intel Power Clamp driver
  Used for System wide Injection for a while
- Per CPU Idle injection
  Used in ARM based systems for a while
  Not used on Intel x86
Solution

- Implement per CPU idle injection for x86
  Should coexist with system wide idle injection via powerclamp
- Enhance idle-inject for to support powerclamp
  For package idle % compensation
Both can call play_idle* at the same time
Powerclamp driver should use powercap/idle-inject also to avoid this issue

Power Clamp and Per Core Idle injection
Issues

- Soft IRQ issues
  - Warnings
  - Dependency on kernel version
    - In general IRQs should be migrated before idle injection
    - When possible (from user space)
- High timer jitter for pinned timers
- High wake because of interrupts reduces the effect
- To do: NO_HZ_FULL impact
**Warnings**

- Warning for Tick Stop when Soft IRQs is pending [appendix 1]
  
  [147777.095484] NOHZ tick-stop error: Non-RCU local softirq work is pending, handler #08!!!
  
  [147777.099719] NOHZ tick-stop error: Non-RCU local softirq work is pending, handler #288!!!
  
  [147777.103725] NOHZ tick-stop error: Non-RCU local softirq work is pending, handler #288!!!

- Caused by race condition with Idle inject FIFO task and ksoftirqd scheduling
- An IRQ can happen, not finish the softirq in its tail
  
  schedule ksoftirqd
  
  But not get scheduled because idle (injection) wins on priority.
- Either don’t print warning in this path or fix
5.18+

- Dependency on kernel version
  Bug in kernel which prevents warning
  https://elixir.bootlin.com/linux/latest/source/kernel/time/tick-sched.c#L1013
  Tick may stop except when CONFIG_PREEMPT_RT and no_hz_full CPU
  https://elixir.bootlin.com/linux/latest/source/include/linux/bottom_half.h#L36
Solution for warnings/delays

• Give chance to ksoftirqd to run if it is in runnable state
  • Yield for 1 jiffee (test patch from Frederic Weisbecker)
• Sleep timer adjustment because of yield for lost time in soft irqs
• Prevent Soft IRQ storm in idle injection loop
Give chance to run Soft IRQ

- Add additional check in while loop in do_idle()
  
  need_resched() || task_is_running(__this_cpu_read(ksoftirqd));

- In play_idle_precise(), if idle duration timer is not expired and do_idle() loop break
  Call schedule_timeout(1) and return to call do_idle again()
  for the remaining idle duration

- To prevent storm of IRQs
  Introduce a max idle_duration(like usleep_range)
  This way loop will break and return -EAGAIN
Pinned Timers:
3 users in kernel
Mce, ipv4 (2 instances)

Do we care as we are in performance limited scenario?

Platform:
Dell XPS 9310
Linux 5.19 with a sample program using pinned timers with duration 16ms.

Other names and brands may be claimed as the property of others.
Appendix [1]

Soft IRQ errors

```<idle>-0 [003] 231.067277: softirq_raise:  vec=1 [action=TIMER]
<idle>-0 [003] 231.067282: softirq_raise:  vec=7 [action=SCHED]
<idle>-0 [003] 231.067282: hrtimer_expire exit: hrtimer=0x00f9a6edccc650 function=tick_sched_timer0x0 expires=229101521504 softexpires=229101521504
<idle>-0 [003] 231.067288: irq_handler_entry: irq=129 name=ksi_ihc
<idle>-0 [003] 231.067305: softirq_raise: vec=6 [action=TASKLET]
<idle>-0 [003] 231.067309: irq_handler_exit: irq=129 ret=handed
<idle>-0 [003] 231.067312: softirq_entry: vec=1 [action=TIMER]
<idle>-0 [003] 231.067312: timer_cancel: timer=0x00f9a6063fccc58
<idle>-0 [003] 231.067314: timer_expire_entry: timer=0x00f9a6063fccc58 function=hooks_post_schedule_work timer_fn now=429494557 basedclk=429494557
<idle>-0 [003] 231.067315: sched_hello_read_work_queue_work: work struct=0x00f9a6063fccc30 function=dump_idle_injection_func worker=0x00f9a6063fccc80
<idle>-0 [003] 231.067316: sched_waking: comm=ksoftirqd/3 pid=1941 priority=48 target_cpu=003
<idle>-0 [003] 231.067319: sched_wakeup: ksoftirqd/3:1941 [49]: CANT FIND FIELD success> CPU:003
<idle>-0 [003] 231.067320: timer_expire_exit: timer=0x00f9a6063fccc58
<idle>-0 [003] 231.067321: softirq_exit: vec=7 [action=SCHED]
<idle>-0 [003] 231.067325: softirq_exit: vec=7 [action=SCHED]
<idle>-0 [003] 231.067327: sched_waking: comm=ksoftirqd/3 pid=33 priority=120 target_cpu=003
<idle>-0 [003] 231.067330: sched_wakeup: ksoftirqd/3:33 [120]: CANT FIND FIELD success> CPU:003
<idle>-0 [003] 231.067339: sched_switch: swapper:[3]:120 R->ksoftirqd/3:1941 [49]
ksoftirqd/3:1941 [003] 231.067341: sched_kthread_work_execute start: work struct=0x00f9a6063fccc30 function=dump_idle_injection_func
ksoftirqd/3:1941 [003] 231.067342: play_idle_enter: state=24000000 cpu_id=3
ksoftirqd/3:1941 [003] 231.067342: hrtimer_init: hrtimer=0x00f9a6064221e40 clockid=CLOCK_MONOTONIC mode=0x0
ksoftirqd/3:1941 [003] 231.067345: hrtimer_expire exit: hrtimer=0x00f9a6064221e40 function=ftime_inject timer_fv0x0 expires=229211664941 softexpires=229211664941
ksoftirqd/3:1941 [003] 231.067358: printk: can_stop_idle_tick.isra.16: NOHZ tick-stop error:3
(Tick will not be stopped because of pending softirq)
ksoftirqd/3:1941 [003] 231.067358: cpu_idle: state=3 cpu_id=3
ksoftirqd/3:1941 [003] 231.071528: cpu_idle: state=4294967295 cpu_id=3
```
Appending [2] 5.19 Soft IRQ

<idle>-0 [007] 2736.421871: softirq_exit: vec=7 [action=SCHED]
<idle>-0 [007] 2736.421871: sched_waking: comm=ksoftirqd/7 pid=58 prio=120 target_cpu=007
<idle>-0 [007] 2736.421872: sched_wakeup: ksoftirqd/7:58 [120]<CANT FIND FIELD success>

CPU:007
<idle>-0 [007] 2736.421873: bprint: do_idle: need_resched loop end:7
<idle>-0 [007] 2736.421875: sched_switch: swapper/7:0 [120] R ==> idle_inject/7:1856 [49]
idle_inject/7-1856 [007] 2736.421876: sched_kthread_work_execute_start: work struct 0xffffda70ffdeb830:
function clamp_idle_injection_func
...
idle_inject/7-1856 [007] 2736.421878: bprint: can_stop_idle_tick.isra.0: report_idle_softirq pending bh
...
idle_inject/7-1856 [007] 2736.421878: tick_stop: success=1 dependency=NULL
...
idle_inject/7-1856 [007] 2736.446231: sched_stat_runtime: comm=idle_inject/7 pid=1856 runtime=16135 [ns]
vruntime=0 [ns]
idle_inject/7-1856 [007] 2736.446241: sched_switch: idle_inject/7:1856 [49] S ==> ksoftirqd/7:58 [120]
ksoftirqd/7-58 [007] 2736.446246: softirq_entry: vec=3 [action=NET_RX]
Appending [3] Timer Delay

```
[007]  6230.039784: sched_waking:  comm=ksoftirqd/7 pid=58 prio=120 target_cpu=007
[007]  6230.039785: sched_wakeup:   ksoftirqd/7:58 [120]<CANT FIND FIELD success>
CPU:007
[007]  6230.039786: cpu_idle:       state=3 cpu_id=7
[007]  6230.039787: cpu_idle:       state=4294967295 cpu_id=7
[007]  6230.039788: softirq_raise:   vec=1 [action=TIMER]
```

```
[007]  6230.043765: softirq_raise:   vec=1 [action=TIMER]
[007]  6230.047806: softirq_raise:   vec=1 [action=TIMER]
[007]  6230.051822: softirq_raise:   vec=1 [action=TIMER]
[007]  6230.056044: softirq_raise:   vec=1 [action=TIMER]
[007]  6230.058575: play_idle_exit:  state=24000000 cpu_id=7
[007]  6230.058577: sched_stat_runtime:  comm=idle_inject/7 pid=56 runtime=7119 [ns]
  vruntime=0 [ns]
[007]  6230.058584: sched_switch:    idle_inject/7:56 [49] S ==> ksoftirqd/7:58 [120]
  ksoftirqd/7-58    [007]  6230.058587: softirq_entry: vec=1 [action=TIMER]
```

6230.058587 - 6230.039788 = 18ms later

Linux Plumbers Conference | Dublin, Ireland Sept. 12-14, 2022
Patch Link

https://github.com/spandruvada/linux-kernel/tree/idle-inject