SUSE

CPU Isolation 2022..2023

Frederic Weisbecker

CPU Isolation: What is it already?

Brief reminder

- Run a task in userspace on a specific set of CPUs without being disturbed by the kernel:
  - No IRQ
  - No competing task
  - No exceptions / faults

CPU Isolation: What is it already?

Brief reminder

- Run a task in userspace on a specific set of CPUs without being disturbed by the kernel:
  - No IRQ
  - No competing task
  - No exceptions / faults

- Typically configured with “nohz_full=” kernel boot parameter and possibly also “isolcpus=”

Changes since last LPC

2022..2023

VMSTAT

What is it.

- Virtual memory statistics gathering and folding
- Per-CPU workqueue executing every second
- Can be triggered remotely

VMSTAT

Past attempts

- Explicit quiescing through prctl()
- Implicit quiescing on return to user

VMSTAT

Solved

- Simply accept some level of imprecision of vmstat
- Don’t schedule it remotely
- It’s naturally postponed locally as a deferred workqueue
- be5e015d107d (“vmstat: skip periodic vmstat update for isolated CPUs”)
- Courtesy of Marcelo Tosatti

Deferred IPI

In progress

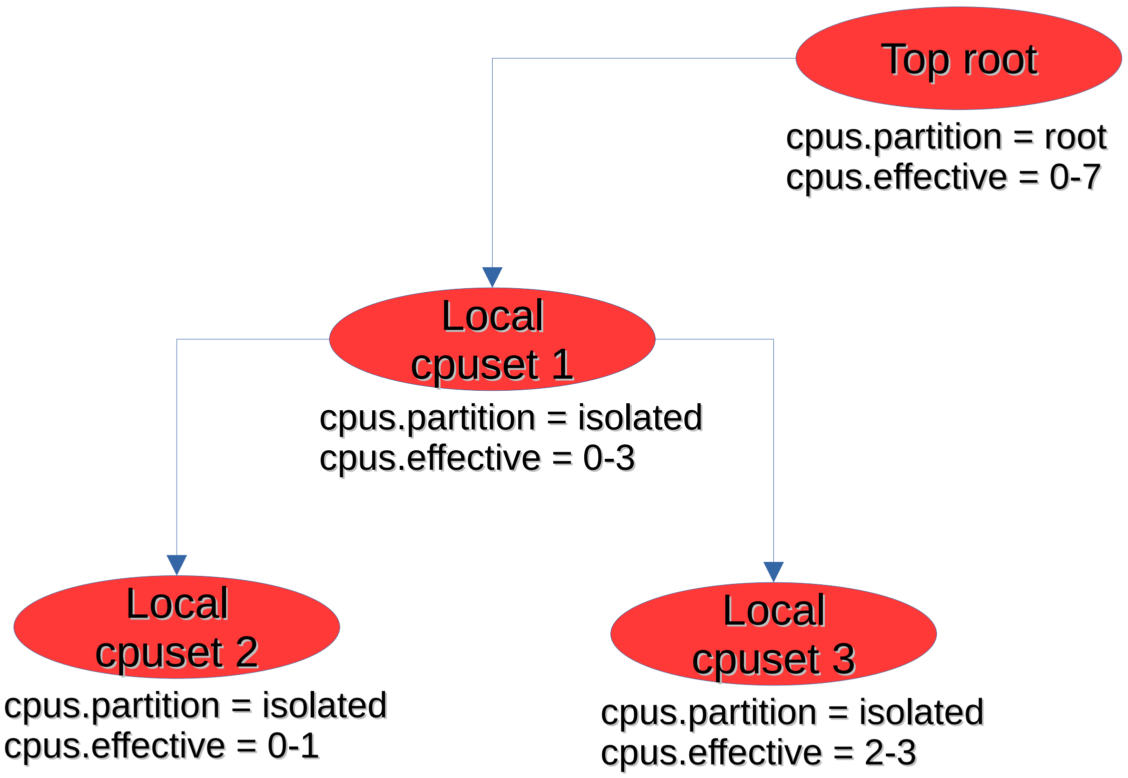
- [RFC PATCH v2 00/20] context_tracking,x86: Defer some IPIs until a user->kernel transition *by Valentin Schneider*

- In progress?

Cpusets v2

New features

- Introduce remote root partitions
- Old style root partition is called **“local”** and has a root partition as a parent
- Newly introduced **“remote”** can have a non-root partition as a parent
- More flexible when containers run several layers down the cpusets tree.
- Courtesy of Waiman Long



Top root

cpus.partition = root
cpus.effective = 0-7

Local cpuset 1

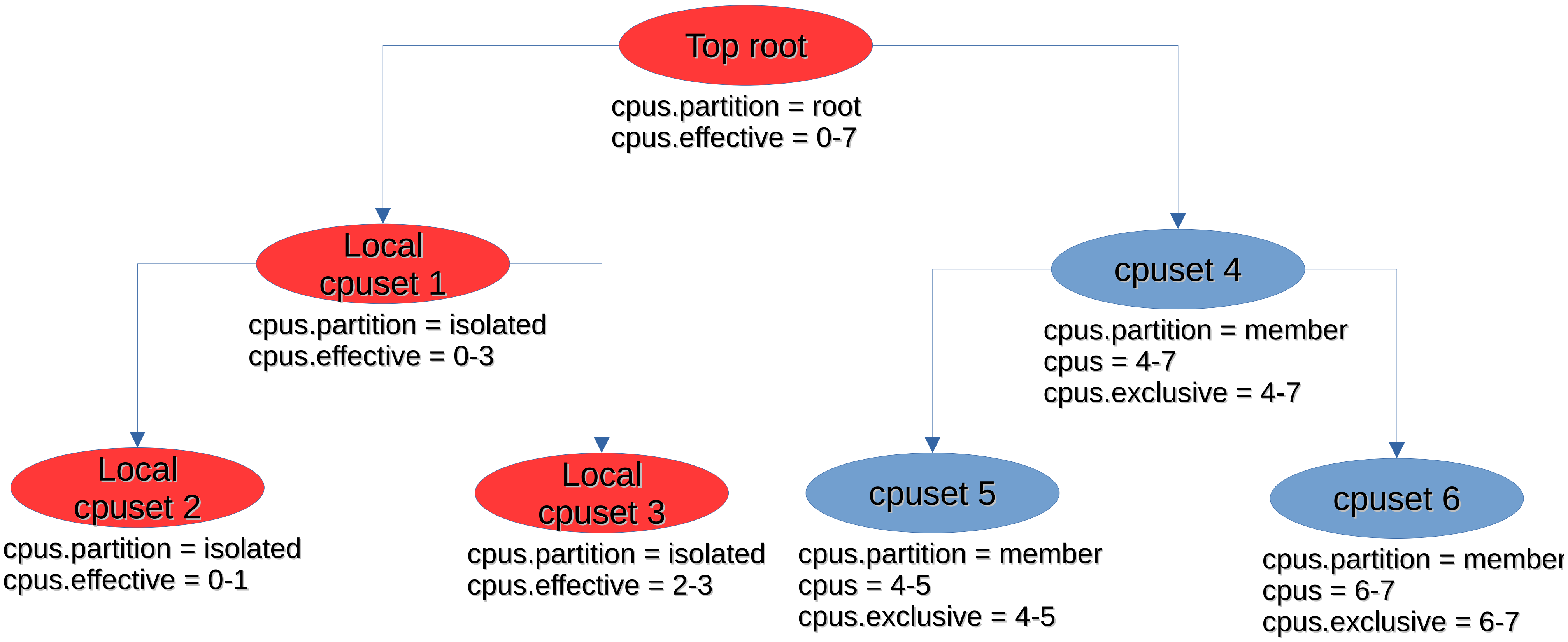
cpus.partition = isolated
cpus.effective = 0-3

Local cpuset 2

cpus.partition = isolated
cpus.effective = 0-1

Local cpuset 3

cpus.partition = isolated
cpus.effective = 2-3



Top root

cpus.partition = root
cpus.effective = 0-7

Local cpuset 1

cpus.partition = isolated
cpus.effective = 0-3

cpuset 4

cpus.partition = member
cpus = 4-7
cpus.exclusive = 4-7

Local cpuset 2

cpus.partition = isolated
cpus.effective = 0-1

Local cpuset 3

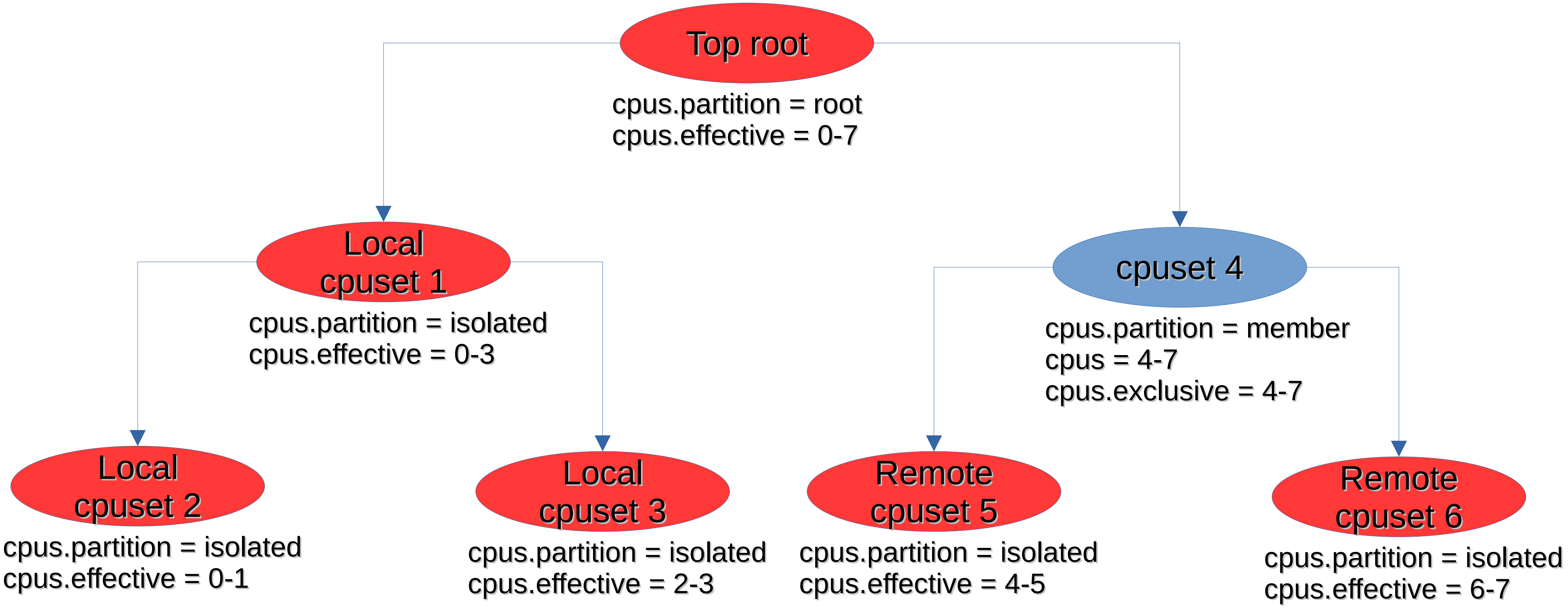
cpus.partition = isolated
cpus.effective = 2-3

cpuset 5

cpus.partition = member
cpus = 4-5
cpus.exclusive = 4-5

cpuset 6

cpus.partition = member
cpus = 6-7
cpus.exclusive = 6-7



Top root

cpus.partition = root
cpus.effective = 0-7

Local cpuset 1

cpus.partition = isolated
cpus.effective = 0-3

cpuset 4

cpus.partition = member
cpus = 4-7
cpus.exclusive = 4-7

Local cpuset 2

cpus.partition = isolated
cpus.effective = 0-1

Local cpuset 3

cpus.partition = isolated
cpus.effective = 2-3

Remote cpuset 5

cpus.partition = isolated
cpus.effective = 4-5

Remote cpuset 6

cpus.partition = isolated
cpus.effective = 6-7

Misc

New features

- 6a792697a53a (memcg: do not drain charge pcp caches on remote isolated cpus, 2023-03-17)
- 8a237adf213d (fs/buffer.c: disable per-CPU buffer_head cache for isolated CPUs, 2023-06-27)
- 0f8b916bc5b5 (hwmon: (coretemp) avoid RDMSR interrupts to isolated CPUs, 2022-12-16)
- Courtesy of Marcelo Tosatti and Waiman Long

Nohz_full cpuset interface

Fate

Nohz_full cpuset interface

Old long term plan

- “nohz_full= “ kernel boot parameter. Can’t be changed at runtime.

- For ten years, plan has been to bring a runtime toggle interface (cpuset)

Nohz_full cpuset interface

Cost 1/3

- When the cpuset is modified, we must make sure that no CPU can queue work to the new nohz_full set. Therefore:

- Need to introduce/maintain a per-cpu rwsem (or an RCU read side) to be used from all noise sources:
  - Core kthread creation (due to affinity setting)
  - All possible kthread affinity setting (RCU, networking, …)
  - Vmstat queue
  - Per-cpu cache drain queue
  - per-CPU buffer_head cache invalidation
  - Etc…

Nohz_full cpuset interface

Cost 2/3

- When the cpuset is modified, we must now (de-)isolate the CPUs:
  - Change workqueue cpumask
  - Migrate unbound timers (though the new pull model might partly solve that)
  - Change many different types of kthread’s affinity
  - Create per-CPU buffer_head
  - Create/remove remote ticks
  - Various networking tweaks
  - Etc....

Nohz_full cpuset interface

Cost 3/3

- Then RCU must (de-)offload its callbacks (RCU_NOCB)

- That one is already upstream but it’s hundreds of lines of very complicated code to maintain

- I’ll be more than happy to remove it!

Nohz_full cpuset interface

I’m giving up

- Adding this interface implies it must be maintained forever

Nohz_full cpuset interface

I’m giving up

- Adding this interface implies it must be maintained forever

- Expect the list of requirements and code to protect within a per-cpu rwsem to ever grow

Nohz_full cpuset interface

I’m giving up

- Adding this interface implies it must be maintained forever
- Expect the list of requirements and code to protect within a per-cpu rwsem to ever grow
- We better have a _real_ incentive to introduce this

Nohz_full cpuset interface

I’m giving up

- Adding this interface implies it must be maintained forever
- Expect the list of requirements and code to protect within a per-cpu rwsem to ever grow
- We better have a _real_ incentive to introduce this
- Because of that I’m giving up this feature, unless people come with very good reasons to proceed and contributors to help.