CPU Isolation
How to tune it on runtime?
Current state

• Boot defined (nohz_full, rcu_nocbs, isolcpus)

• Housekeeping API
  • Bunch of isolation features (HK_TYPE_*)
    • TIMER, RCU, MISC, SCHED, TICK, DOMAIN, WQ, MANAGED_IRQ, KTHREAD

  • Designed to be individually tunable through cpusets

  • Some of them may be grouped: TIMER | WQ | KTHREAD = UNBOUND
First attempt

• Start with rcu_nocbs=

• Core RCU nocb work to allow that during the last 2 years

• [PATCH 0/4] rcu/cpuset: Control RCU_NOCB offloading through cpusets
Why using cpusets and not sysctl?
- Not hierarchical
- But integrates with load balancing tuning (cpuset.sched_load_balance)

Do we need to tune individual isolation bits at all?
- YES: rcu_nocbs can be used individually for power saving (but didn’t know at the time)
- Suggestion from Peter Zijlstra: start with an all feature switch and add on top more finegrained features
- But Tejun Heo is reluctant to adding kernel internals switch to cgroups
Sysctl alternative?

• Sysctl is attractive:
  • allows for more kernel internal tuning
  • much more simple
  • Then CPU isolation doesn’t need to depend on cgroups

• kernel.isolation.default = [cpulist] (overrides nohz_full boot param: tick, workqueue, timer, etc…)

• kernel.isolation.rcu_noCB = [cpulist] (overrides only the noCB part)

• Etc…