Cleaning Up Linux's CPU Hotplug For Real Time and Energy Management

Thomas Gleixner, Paul E. McKenney, and Vincent Guittot
CPU Hotplug: History and Planned Usage

- **Original purpose:** Remove failing hardware
  - For example, cache SRAMs flagged by correctable memory errors
  - Avoid uncorrectable failures by hotplugging corresponding CPU

- **Required properties:**
  - Fast compared to hardware rate of decay
    - A few seconds latency is almost never a problem
  - Reliable compared to failing hardware
    - Some failure rate can be tolerated: Hardware is failing anyway
  - Expected usage frequency: Very rare

- **But CPU hotplug is now being used for other things...**
CPU Hotplug: Current Usage and Issues

▪ Use case #1: Energy management on mobile devices
  – Remove unneeded CPUs from service
  – Requires fast latency and high reliability

▪ Use case #2: Real time CPU conditioning
  – Requires high reliability and minimal disturbance to rest of system
    • Minimal disturbance required for consolidated real-time applications

▪ Summary of CPU-hotplug issues:
  – Too slow: 100s of milliseconds to seconds, need ~5 milliseconds
  – Disturbs rest of system when removing CPU
  – Questionable reliability
    • In part due to poor design and lack of testing
CPU Hotplug: Approach to Solution

- Too slow: need ~5 milliseconds
  - Majority of overhead from per-CPU task creation/migration
  - Approach: Don't destroy per-CPU tasks, park them!

- Disturbs rest of system when removing CPU
  - Approach: Wean CPU hotplug from its use of stop_machine()
  - Large effort, semantics change of for_each_online_cpu()
    - Automated analysis in progress

- Questionable reliability
  - Dependency-defying design: CPU-offline ordering backwards
  - Offline CPUs make one final pass through the scheduler
    - RCU currently kludges around this “interesting” property
  - Approach: CPU-offline notification order reverse of online, offline CPU
    in task context rather than in idle context
Legal Statement

- This work represents the view of the authors and does not necessarily represent the view of IBM, Linutronix, ST-Ericsson, or Linaro.
- IBM and IBM (logo) are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries.
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.