Scheduling and big.LITTLE Architecture
Scheduling Topics

- Target CPU Selection (Vincent Guittot)
- Sharing Scheduler Information (Vincent Guittot)
- Task Placement for Asymmetric Cores (Morten Rasmussen)
- Scheduling and the big.LITTLE Architecture *(you are here!)*
- Dynamic CPU Core Management (Peter De Schrijver and Antti Miettinen)
- Application of Deadline Scheduling for Power-Saving Strategies (Juri Lelli)

- Main focus of this talk is progress on CPU hotplug
  - See Peter's and Antti's talk for alternatives
CPU Hotplug Primer: Online Process

- CPU_UP_PREPARE
  - Succeeded: CPU_STARTING
  - Failed: CPU_UP_CANCELLED

- CPU_STARTING
  - Failed: CPU_UP_CANCELLED

- CPU_ONLINE

- cpu_hotplug.lock held
- cpu_add_remove_lock held
CPU Hotplug Primer: Offline Process

- **CPU_DOWN_PREPARE**
  - Succeeded
    - **CPU_DYING**
      - Succeeded
        - **CPU_DEAD**
          - **CPU_POST_DEAD**
      - Failed
        - **CPU_DOWN_FAILED**
          - Failed
            - All other CPUs spinning with interrupts off while outgoing CPU is in CPU_DYING notifiers
              - `cpu_hotplug.lock` held
            - `cpu_add_remove_lock` held
CPU Hotplug Is Not Atomic

Valid notifier order for online:
- IPIs
- RCU
- Scheduler
Must reverse order for offline

Reality will intrude...
- RCU depends on scheduler
- Circular dependency!
Must further decompose RCU and scheduler interaction
Valid notifier order for online:
- IPIs
- RCU
- Scheduler
Must reverse order for offline

Reality will intrude…
- RCU depends on scheduler
- Circular dependency!
Must further decompose RCU and scheduler interaction
CPU Hotplug: Valid Online Order (2/3)

Valid notifier order for online:
- IPIs
- RCU
- Scheduler
Must reverse order for offline

Reality will intrude...
- RCU depends on scheduler
- Circular dependency!
Must further decompose RCU and scheduler interaction
CPU Hotplug: Valid Online Order (3/3)

Valid notifier order for online:
- IPIs
- RCU
- Scheduler

Must reverse order for offline

Reality will intrude...
- RCU depends on scheduler
- Circular dependency!

Must further decompose RCU and scheduler interaction
Suppose We Offline In The Same Order...

Valid notifier order for online:
- IPIs
- RCU
- Scheduler

Must reverse order for offline

Reality will intrude...
- RCU depends on scheduler
- Circular dependency!

Must further decompose RCU and scheduler interaction
CPU Hotplug: Invalid Offline Order

Valid notifier order for online:
- IPIs
- RCU
- Scheduler

Must reverse order for offline

Reality will intrude...
- RCU depends on scheduler
- Circular dependency!

Must further decompose RCU and scheduler interaction
CPU Hotplug: Valid Offline Order (1/3)

Valid notifier order for online:
- IPIs
- RCU
- Scheduler
Must reverse order for offline

Reality will intrude...
- RCU depends on scheduler
- Circular dependency!
Must further decompose RCU and scheduler interaction
CPU Hotplug: Valid Online Order (2/3)

Valid notifier order for online:
- IPIs
- RCU
- Scheduler
Must reverse order for offline

Reality will intrude...
- RCU depends on scheduler
- Circular dependency!
Must further decompose RCU and scheduler interaction
Valid notifier order for online:
• IPIs
• RCU
• Scheduler
Must reverse order for offline

Reality will intrude…
• RCU depends on scheduler
• Circular dependency!
Must further decompose RCU and scheduler interaction
Scheduling and big.LITTLE Architecture

CPU Hotplug: Work In Progress

- Reduce task creation/destruction overhead
  - Thomas Gleixner's generic-idle (mainline) and park/unpark (-tip)
  - Tejun Heo's fix for workqueues

- Numerous fixes from Srivatsa Bhat (3.4-3.6)

- Disturbance-free SRCU (Lai Jiangshan, 3.4)

- Wean RCU from __stop_machine() (PEM, 3.5-3.7)

- Wean scheduler from __stop_machine() (in progress)

- Guard for_each_online_cpu() (Silas Boyd-Wickizer, 3.7-3.8)

- Switch hotplug from __stop_machine() (PEM experimental)

- Reverse notifier order, reworking notifier concepts (TBD)

- Apply park/unpark everywhere (TBD)
Scheduling and big.LITTLE Architecture

Other Topics

- Synthetic Mobile Workloads
  - Progress, but no nice solution yet

- Emulating big.LITTLE on commodity systems
  - An embarrassment of riches: http://lwn.net/Articles/501501/
Summary

- Excellent progress on many fronts
- But much work still to be done!
Legal Statement

- This work represents the view of the author and does not necessarily represent the view of IBM.
- 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.
Questions?