Providing per-task Quality of Service

Juri Lelli <juri.lelli@arm.com>
Outline

Introduction

Energy Aware Scheduling
   Status update
   Yeah, right.. but what is it?!
   Discussion

Deadline Scheduling
   Status update
   Yeah, right.. but what is it?!
   Discussion
Who am I?

- SSG-Power team @ ARM Ltd.
- working on Linux scheduler
- Energy Aware Scheduler (sched-DVFS in particular)
- SCHED_DEADLINE
Aim

- Briefly introduce API changes/additions enhancing the Linux scheduler towards better energy efficiency and real-time(ish) behaviour
- Start again last year LPC discussion...
- regarding userspace (middleware) vs. kernel space information exchange
Introduction

Energy Aware Scheduling
  Status update
  Yeah, right.. but what is it?!
  Discussion

Deadline Scheduling
  Status update
  Yeah, right.. but what is it?!
  Discussion
Introduction

Energy Aware Scheduling
  Status update
  Yeah, right.. but what is it?!
  Discussion

Deadline Scheduling
  Status update
  Yeah, right.. but what is it?!
  Discussion
Status update

- EASv5 posted July, 7th
- Foundation patches (without RFC tag) posted Aug., 14th
- SchedTUNE posted Aug., 19th
Energy Aware Scheduling

Oh, cool! But, what is it anyway?! Did you/planning to attend EAS miniconf? Thanks/Please do 10/15 min answer
Per-Entity Load Tracking

- Geometric series for
  - Load (amount of time a sched_entity is runnable)
  - Utilization (amount of time a sched_entity is running)

- Utilization gives you a [0..1024] number that represents how big is your task

- Both frequency and µ-arch scaled (i.e., that number is the same wherever you are running and at whatever OPP)
Foundation

Energy Model

- It considers CPUs only, no peripherals, GPU or memory.
- It contains information about OPPs and idle states
Energy Aware Scheduling

- Use the energy model and information about tasks to evaluate implications of scheduling decisions
- The goal is to minimize energy, while still getting “good” performance
- While the default scheduler has a performance-only objective
Energy Aware Scheduling

- Energy-aware scheduling is only active when the system is not over-utilized
- Tipping point: very conservative, one CPU fully utilized at its highest OPP
- When above the tipping point we go for the traditional way (spreading tasks)
- When below, scheduling decisions are taken considering the total energy impact of adding/removing/migrating utilization
Let the scheduler control OPP selection

- New cpufreq governor* that is triggered from scheduler context
- Select an OPP that has enough capacity (spare room) to accommodate tasks that we decide to schedule on a CPU
- Decisions are taken at freq-domain level
  - Per freq-domain kthread (where required) responsible for doing the actual freq change
  - Woken up via IPIs
- CFS only, but should be fairly easy to extend it to the other scheduling classes

* foundation by Mike Turquette
Sched-TUNE

Userspace power/performance tunable knob*

- Stacked on top of Sched-DVFS
- Interacts with CFS scheduler (for the time being)
- Global and CGroup based (per-task) interface
  - global: /proc/sys/kernel/sched_cfs_boost
  - cgroup: /sys/fs/cgroup/stune/performance/schedtune.boost

* Patrick Bellasi
Discussion

- Did you experiment with EAS?
- How about the interface we designed to switch between performance and efficiency?
- What we would need to go forward...
Introduction

Energy Aware Scheduling
  Status update
  Yeah, right.. but what is it?!
  Discussion

Deadline Scheduling
  Status update
  Yeah, right.. but what is it?!
  Discussion
Introduction

Energy Aware Scheduling
- Status update
- Yeah, right.. but what is it?! 
- Discussion

Deadline Scheduling
- Status update
- Yeah, right.. but what is it?! 
- Discussion
Status update

- Mainline since Linux 3.14
- Helping maintaining it (is Peter around?! :-))
- New features are coming, stay tuned!
Deadline scheduling (a.k.a. SCHED_DEADLINE)

Yeah, right.. but, what is it?!

10/15 min answer
Deadline scheduling (a.k.a. SCHED_DEADLINE)

it’s not only about deadlines

- it’s a relatively new addition to the Linux scheduler
- real-time scheduling policy
- higher priority than SCHED_NORMAL and SCHED_FIFO
- allows explicit per-task latency constraints
- enables predictable task scheduling, avoids starvation and
- enriches scheduler’s knowledge about tasks’ QoS requirements
Deadline scheduling (a.k.a. SCHED_DEADLINE)

it’s not only about deadlines

- it’s a relatively new addition to the Linux scheduler
- real-time scheduling policy
- higher priority than SCHED_NORMAL and SCHED_FIFO
- allows explicit per-task latency constraints
- enables predictable task scheduling, avoids starvation and
- enriches scheduler’s knowledge about tasks’ QoS requirements
- it implements
  - Earliest Deadline First (EDF): tasks with earlier deadlines have higher priorities
  - Constant Bandwidth Server (CBS): reservation based scheduling
- CBS it’s the cool thing here
Reservation Based Scheduling

- Concurrent real-time tasks compete for resources (CPU time)
- Resource Reservation mechanism
- A task is allowed to execute for:
  - Q time units (*runtime*)
  - in every interval of length P (*period*)
- Task *utilization* is $U = Q/P$
- Note that this is something that we don’t need to estimate (like with PELT)
- It is enforced by the system
- You need to specify 3 parameters (ns): *runtime*, *deadline*, *period*
Sporadic Task Model

activation/wakeup

start time

computation time

relative deadline

period

absolute deadline

Feelfree to throw whateverkind of task at it!
Sporadic Task Model

Feel free to throw whatever kind of task at it!
RR over Sporadic Task Model

- Average runtime [ns]
- Deadline [ns]
- Period [ns]
SCHED_DEADLINE: some numbers

- MPlayer with HD movie
- QoS metric: IFT, difference between DT of current and previous one
- Variation in IFT → video doesn’t play smoothly
- frame rate = 23.9fps, IFT = 41708μs
- P = IFT (for SCHED_DEADLINE)
- 6 other instances of ffmpeg in background
- CFS QoS highly dependent on system load
- With SCHED_DEADLINE player not affected (with reasonable CPU usage)
Discussion

- Better quality of service provisioning
- Additional information for the scheduler
- How do you like the interface?
- Does it look usable?
- SurfaceFlinger maybe?
Thank You

The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.