



CANONICAL

# Deadline scheduler in the audio domain

Linuxcon Europe, Oct 2015

David Henningsson

Audio Hardware Enablement Engineer



- Introduction
- CPU clocks and capacities
- PulseAudio specials
- RT problems

# Typical audio latencies

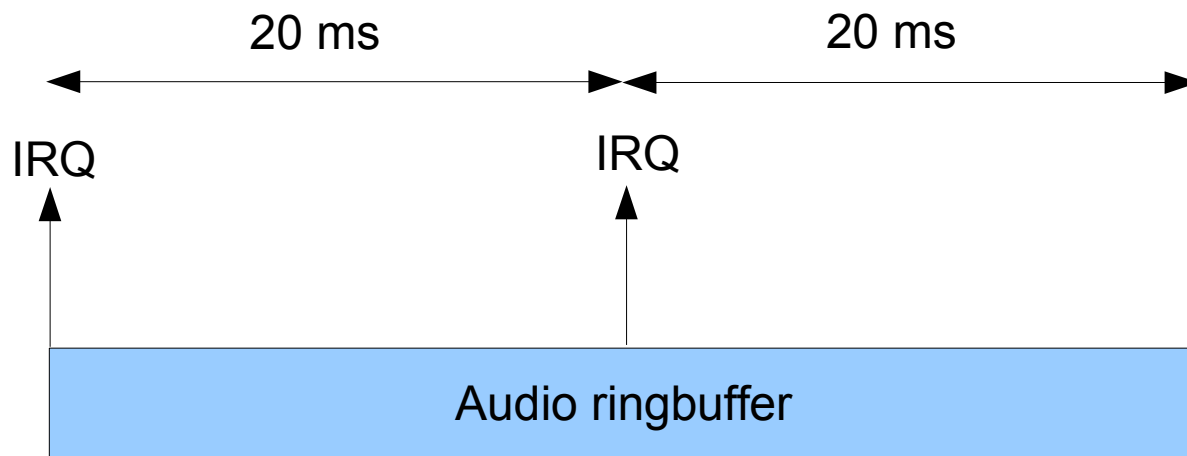
---



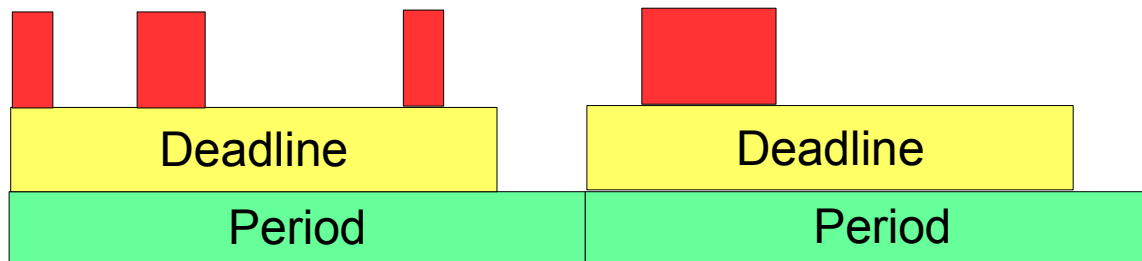
- Music playback: 100 – 2000 ms
- VOIP: 20 – 50 ms
- Gaming: 10 – 30 ms
- Pro audio: 1 – 20 ms

Higher latency = less CPU power, less scheduler pressure

# Audio buffering basics



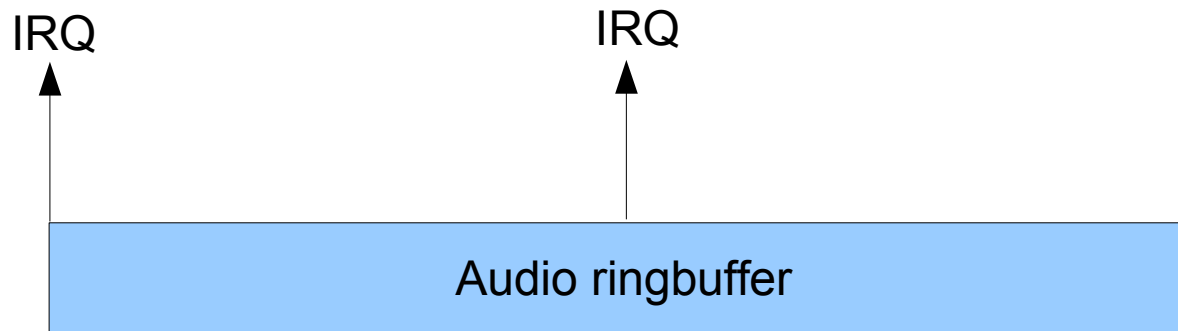
# Deadline scheduler basics



Runtime = 5 ms

Deadline = 17 ms

Period = 20 ms





---

**IN THEORY.**

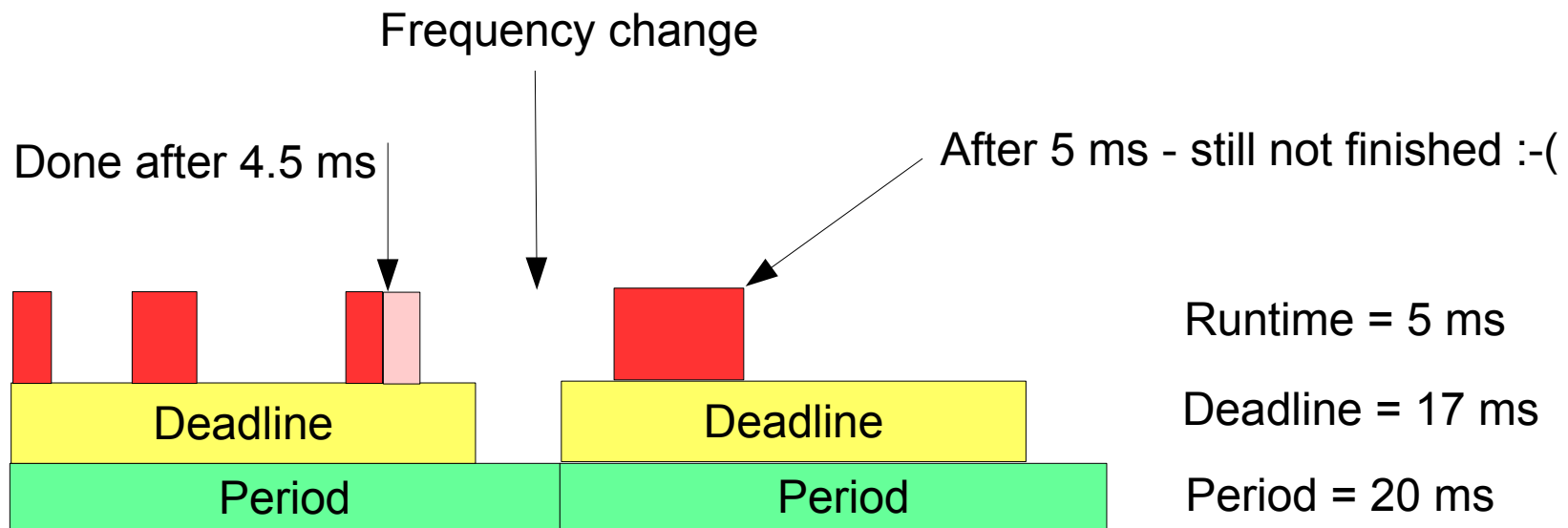


- Two master clocks are never completely synchronised
- Deadline scheduler uses a CPU based clock
- Audio buffer uses its own clock
  - Can even be external (in case of digital inputs)
  
- Workaround: over-provisioning

# CPU capacity change



- CPUs change frequency in order to save power
- Deadline scheduler parameters are all time based
- Amount of runtime depends on the frequency...







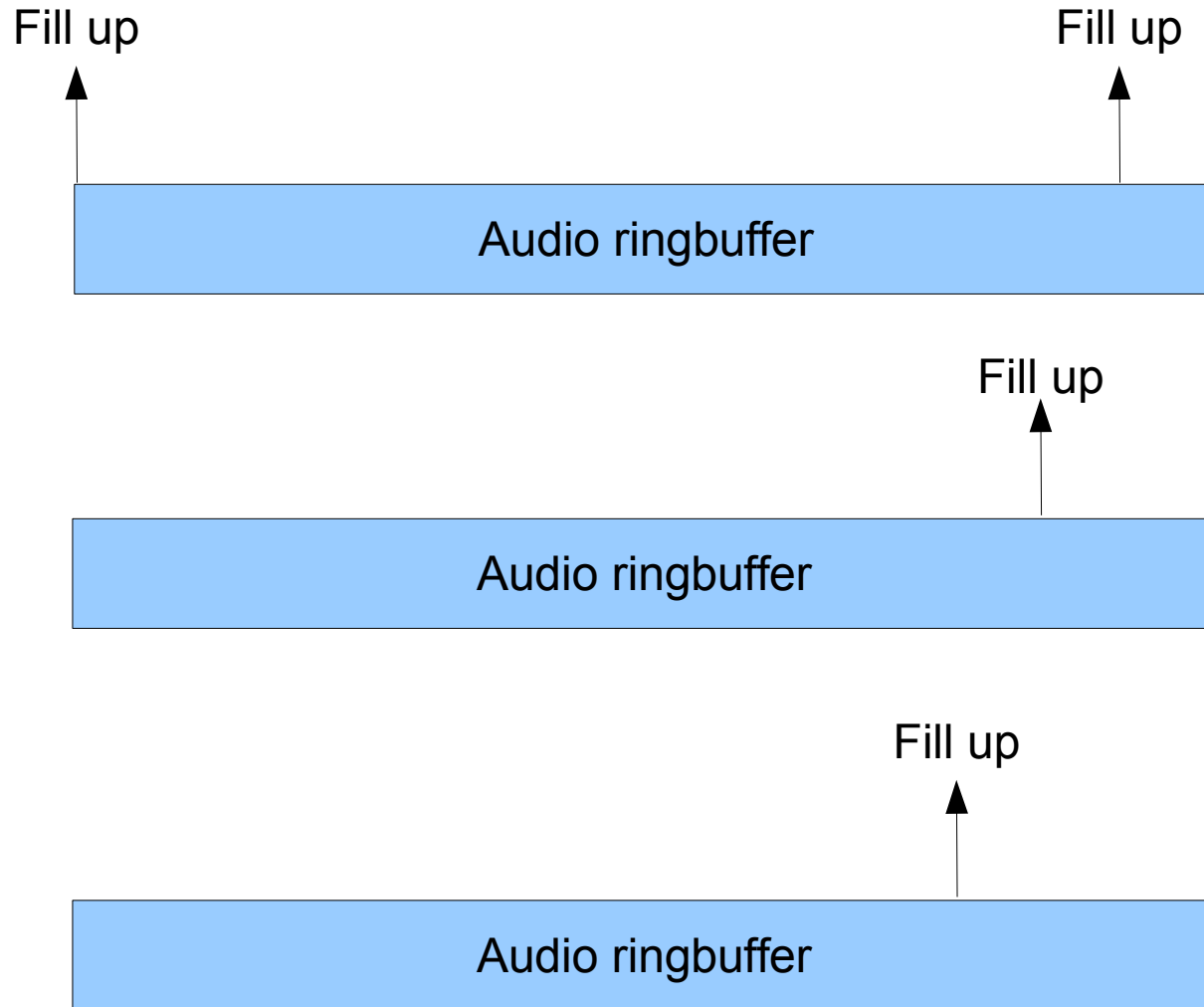
A sound server's CPU consumption depends on:

- Number of streams to mix
- Filters
- Volume / mute
- Latency
- ...etc

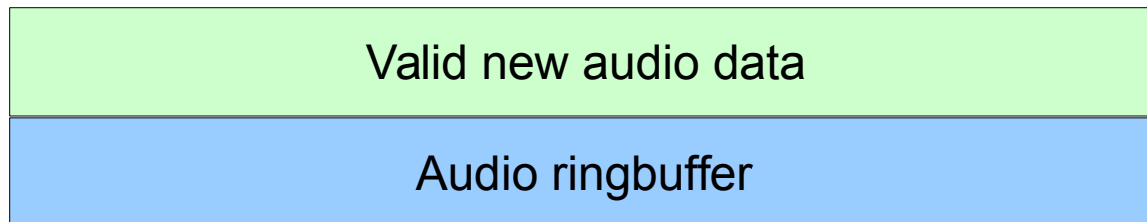
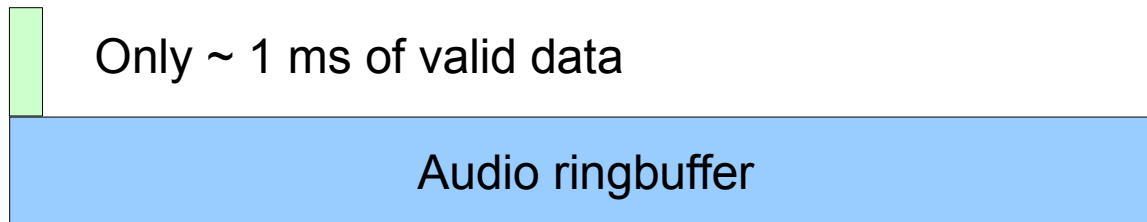
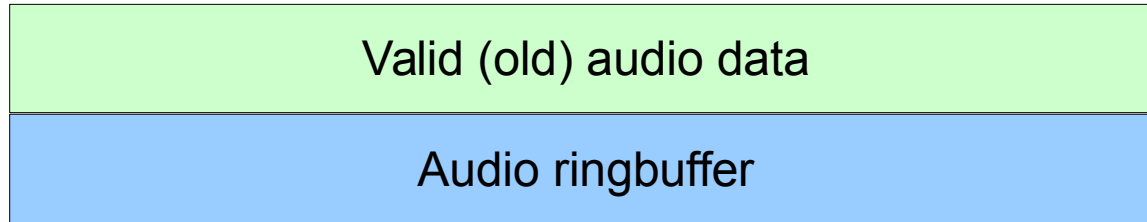


- Desktop software runs on different hardware with different performance
- Maybe measure?
- Maybe just set a fixed number (e g 10%)?
- Requires application discipline

# PulseAudio timer-based scheduling



# PulseAudio rewinds





- No quota system
- A problem with SCHED\_FIFO and SCHED\_RR too
- Current workarounds:
  - PulseAudio uses rtkit, a surveillance daemon
  - JACK encourages people to modify `/etc/security/limits.conf`
- CONFIG\_RT\_GROUP\_SCHED + systemd = no RT for you



- Also a problem with SCHED\_RR and SCHED\_FIFO
- Kernel cannot schedule processes while in an IRQ or when preempt is disabled
- Deadline scheduler cannot take this into account
  - How long time spent in “preempt disable” mode counts as a bug?



CANONICAL

Thank you

Questions?