Timing issues in desktop audio playback infrastructure

Alexander Patrakov

April 11, 2015

About myself

- I am not working for any audio or open-source company
- I have submitted some PulseAudio patches
- I wrote dcaenc
- I added a high-quality resampler to Wine

Primary references

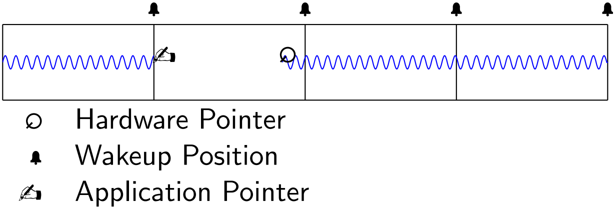
- http://0pointer.de/blog/projects/pulse-glitch-free.html
- https://wiki.freedesktop.org/www/Software/PulseAudio/Backends/ALSA/Issues/

ALSA architecture

- Raw hardware (hw:) devices
- Plugins
  - resampling, format conversion, channel remapping
  - volume attenuation, mixing
  - output to pulse, cras, . . .
- Common API
- `.asoundrc` to glue pcm names with plugin chains

Traditional scheduling

- Buffer, divided into periods
- Sound card tells the kernel when a period elapses
- One period = one application wakeup



Latency Requirements

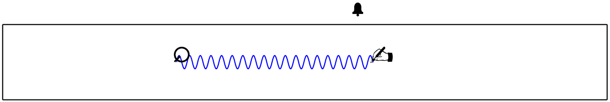
- Latency = buffer size
- Wakeup interval = period size
- Too much latency is bad for games and VoIP
- Low latency $\Rightarrow$ more dropouts
- Too low wakeup interval eats battery

Conflict!

- Consider mixing with dmix
  - Period size is common
  - Period size is not reconfigurable at runtime
  - $\Rightarrow$ Fixed low wakeup interval for the worst case

Timer-based Scheduling

- Soundcard interrupt period is not reconfigurable ☹
- We can use a timer instead ☺



Hardware Pointer

Wakeup Position

✍ Application Pointer

Loop

- Query application & hardware pointer difference
- Write sound data
  - low latency $\Rightarrow$ just some data
  - high latency $\Rightarrow$ a LOT of data
- Schedule a timer that fires just before it plays out
- Sleep

Implementations

- PulseAudio
- CRAS

We've got

Dynamic latency ☺

We've got

Corner cases ☹

On stream start

- To process (resample, mix, encode): 2000 ms of sound
- Budget: 200 ms of real time (due to rtkit)
- Not easy:
  - On a weak CPU (ARM), or
  - With software DTS encoder, or
  - Under valgrind, or
  - ...
- Result: **Killed**

On stream start

- To process (resample, mix, encode): 50 ms of sound
  - `load-module module-udev-detect tsched_buffer_size=50000`
- Budget: 200 ms of real time (due to rtkit)
- Easy!

Wakeup timing

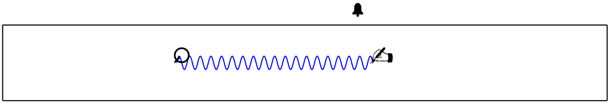
- PulseAudio goal: wake up as late as possible
- Adaptive watermark-based scheduling algorithm
  - Reacts to underruns, near-underruns or absence of them
  - Needs timestamp conversion

Wakeup timing issues

- Xonar DX eats first 5 ms of audio in no time
  - Already worked around in PulseAudio:
    - Cut sleep time in half until one buffer is played
- Imprecise hardware pointer reports
  - Adaptive watermark-based scheduling algorithm gets fooled
  - Worst case: double-buffered (batch) audio transfers
    - PulseAudio switches to period-based scheduling on batch cards

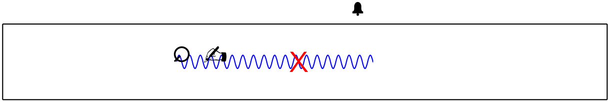
Reacting to unexpected events

- External events
  - New streams
  - Volume changes
- Need to react quickly
  - Even if a high-latency stream is playing
- Solution: rewinds!
  - ???



Reacting to unexpected events

- External events
  - New streams
  - Volume changes
- Need to react quickly
  - Even if a high-latency stream is playing
- Solution: rewinds!
  - ???



Rewinds in ALSA

- `snd_pcm_rewind()`
  - Please let me overwrite the last N samples!
- `snd_pcm_rewindable()`
  - How much can be rewound now?
- `snd_pcm_forward()`, `snd_pcm_forwardable()`
  - Undo a rewind
- PulseAudio assumes that full rewinds work

Rewinding hw devices

- Rewinding is easy!
  - Just move the application pointer
- Telling how much to rewind is not easy ☹
  - Problem: imprecise pointer position
  - Problem: interference with DMA controller
  - Workaround: static 256-byte or 1.33 ms “safeguard” in PulseAudio

Testing rewinds

- Use a buffer with four periods
- In a loop, after filling the buffer with silence:
  - Rewind one period
  - Write one period of silence
  - Write one period of square waves
- Correct output: silence
  - hw devices pass the test

Rewinding plugins

- Callbacks in `snd_pcm_fast_ops_t`
- Default implementations in `src/pcm/pcm_generic.c` and `src/pcm/pcm_plugin.c`
  - Forward the request to slave
  - Move application pointer

Rewinding plugins

- Callbacks in `snd_pcm_fast_ops_t`
- Default implementations in `src/pcm/pcm_generic.c` and `src/pcm/pcm_plugin.c`
  - Forward the request to slave
  - Move application pointer
- Also one needs to restore state

Rewinding plugins

- Callbacks in `snd_pcm_fast_ops_t`
- Default implementations in `src/pcm/pcm_generic.c` and `src/pcm/pcm_plugin.c`
  - Forward the request to slave
  - Move application pointer
- Also one needs to restore state
  - No state, no problem

Rewind support status

Good: hw, alaw, asym, copy, empty, hooks, linear, lfloat, mmap_emul, mulaw, multi, route, softvol (if nobody changes volume)

Dmix bug

- Look at this old bug:

```
if (dmix->state == SND_PCM_STATE_RUNNING ||
   dmix->state == SND_PCM_STATE_DRAINING)
       return snd_pcm_dmix_hwsync(pcm);
```

- Net result: `return 0;` and do not rewind
- Introduced in 2008 (patch adds 459 lines)
- Noticed and fixed in 2014
- Still there are other bugs (yet undiagnosed) ☹

iec958 plugin

- Needed on old cards for adding preambles and various auxiliary bits
- Preamble sequence:
  `ZYXYXYXYXYXYXY....ZYXYXYXYXYXYXY....` (period $= 384$)
- State: position in that sequence

adpcm plugin

- Software adpcm codec
- State: `snd_pcm_adpcm_state_t`
  - Needs to be stored for past samples
  - Is now stored past the last sample only
  - Problem with testing the change

Rewind support status

Good: hw, alaw, asym, copy, empty, hooks, linear, lfloat, mmap_emul, mulaw, multi, route, softvol (if nobody changes volume), iec958 (1.0.28)
Bad but fixable: dmix, dshare, file, adpcm

Interfacing with the world

- ioplug
  - pulse, bluetooth (old), cras, a52
- extplug
  - upmix, vdownmix
  - dca, alsaequal
- ladspa

ioplug

- `struct snd_pcm_ioplug_callback`
- has `.transfer` callback
- has no rewind-related callbacks

ioplug

- `struct snd_pcm_ioplug_callback`
- has `.transfer` callback
- has no rewind-related callbacks
  - They wouldn't be implementable anyway!
    - Think about unsending Bluetooth packets ☹
    - External libraries are not rewindable

ioplug

- `struct snd_pcm_ioplug_callback`
- has `.transfer` callback
- has no rewind-related callbacks
  - They wouldn't be implementable anyway!
    - Think about unsending Bluetooth packets ☹
    - External libraries are not rewindable
  - They aren't needed if `.transfer` does nothing irreversible
  - `jack` plugin has no `.transfer` callback and is rewindable ☺

Rewind support status

Good: hw, alaw, asym, copy, empty, hooks, linear, lfloat, mmap_emul, mulaw, multi, route, softvol (if nobody changes volume), iec958 (1.0.28), ioplug (without `.transfer`)
Bad but fixable: dmix, dshare, file, adpcm
Unfixable: ioplug (with `.transfer`), extplug, ladspa

Rewind support status

Good: hw, alaw, asym, copy, empty, hooks, linear, lfloat, mmap_emul, mulaw, multi, route, softvol (if nobody changes volume), iec958 (1.0.28), ioplug (without `.transfer`)
Bad but fixable: dmix, dshare, file, adpcm, rate (in principle)
Unfixable: ioplug (with `.transfer`), extplug, ladspa, rate (library-based or with current set of ops)

Relevant results

- a52 (ioplug)
  - already worked around (hackishly)
  - `max_rewind = 0`
- dca (extplug)
  - patch rejected
  - ALSA changes are wanted

ALSA changes

- `snd_pcm_hw_params_can_rewind()`
- Added, but then removed in favour of `snd_pcm_rewindable()`
  - Works only of the buffer size is already set
  - Returns 0 for an empty buffer
  - Verdict: unusable for PulseAudio purposes

Internal processing in PulseAudio

- Resampling
  - `https://bugs.freedesktop.org/show_bug.cgi?id=50113`
- Virtual sinks (echo cancellation, virtual surround)
  - Same problem with state
- Software crossover for LFE channel extraction
  - Took four attempts
  - Provoked a “how to test” question from devs
  - Works now ☺

`pulse` ALSA plugin issues

- Does not tell PulseAudio about rewinds
- Blindly agrees to “impossible” buffer metrics

Conclusions

- Timer-based scheduling works in simple cases
- In other cases, PulseAudio needs/has workarounds
- CRAS doesn't have any of the discussed workarounds
  - Self-inflicted problems?