# Pro Audio is Easy, Consumer Audio is Hard

## Linux Audio Confernce 2010

Lennart Poettering
lennart@poettering.net


May 2010

# Who Am I?

Software Engineer at Red Hat, Inc.

Developer of PulseAudio, Avahi and a few other Free Software projects

`http://0pointer.de/lennart/`

`lennart@poettering.net`

IRC: `mezcalero`

Not musician

Not musician

Didn't graduate in signal processing

Why?

Why?

Correct misconceptions

Why?

Correct misconceptions

Improve cooperation, share code and interfaces

Why?

Correct misconceptions

Improve cooperation, share code and interfaces

It might just be interesting

# General Background

## General Background

## Nobody makes money off Linux Audio

General Background

Nobody makes money off Linux Audio

Only 3 people payed to hack on general purpose infrastructure by the big distributors

However, we have allies, particularly in RT, Mobile, also V4L

Pro Audio: knowledgable user

Pro Audio: knowledgable user

Consumer Audio: laymen users

Pro Audio: good hardware

Pro Audio: good hardware

Consumer Audio: crappy hardware (Both the PC, and the Audio HW, Feature Monsters)

Pro Audio: Audio is primary purpose of the machine

Pro Audio: Audio is primary purpose of the machine

Consumer Audio: Audio is one thing among many

Pro Audio: User owns the machine, full configurability, user == administrator

Pro Audio: User owns the machine, full configurability, user == administrator

Consumer Audio: user != administrator

Different Optimization Targets:

Different Optimization Targets:

Pro Audio: low latencies, functionality

Different Optimization Targets:

Pro Audio: low latencies, functionality

Consumer Audio: low power usage, dynamically changing environments, robustness

# Latency vs. Power Usage

In Pro Audio: minimize latency at price of power usage, and dynamic configuration changes

In Pro Audio: minimize latency at price of power usage, and dynamic configuration changes

In Consumer Audio: choose latency as high as possible, and as low as necessary, allow configuration changes

Result for consumer: we need to change latencies

Result for consumer: we need to change latencies

Unfortunately, most hardware/drivers cannot do that

Result for consumer: we need to change latencies

Unfortunately, most hardware/drivers cannot do that

Also: hardware/drivers limited in there range: ideally we have one wakeup every 1s

Hence: schedule audio with system timers

Hence: schedule audio with system timers

That's difficult, because system timers deviate from sound card timers, and sound card timers are crap

# Time interpolation

Time interpolation

Linear Regression + Spline interpolation

Time interpolation

Linear Regression + Spline interpolation

Exponentional backoff

Time interpolation

Linear Regression + Spline interpolation

Exponentional backoff

Everything twice

To make long latencies useful: support seeking

To make long latencies useful: support seeking

Requires history, for remixing

Dropouts, require dynamic adjustments of sleep times

# Timing and transfer block granularity

Outlook, moving timing into the kernel?

Non-interpolated timing has advantages?

Non-interpolated timing has advantages?

Sound card IRQs are accurate!

Algorithm to use? DLL?

We cannot trust clients, one misbehaving client may not be allowed to stall the entire pipeline

RT CPU time calculations change, need to preemptively increase sleep times when streams are moved, removed and added

RT CPU time calculations change, need to preemptively increase sleep times when streams are moved, removed and added

Why? Never drop out

Unreliable latencies, dynamically changing

Compromises in RT code, IO loops are not constant time

Multiple encoded formats, conversion/resampling required, but ideally zero-copy

Multiple encoded formats, conversion/resampling required, but ideally zero-copy

Need to support S16LE and FLOAT32, and a lot more

Large latency means exchanging of large blobs

Large latency means exchanging of large blobs

Zero copy (because sizeof(metadata) ¡ sizeof(payload))

Cached audio samples

Cached audio samples

Means ref counting

Cannot expose low-level hw controls

Cannot expose low-level hw controls

Need to merge and extend in software

Cannot expose low-level hw controls

Need to merge and extend in software

Difficult: Routing in the Sound HW seldom known (USB!)

Common: unreliable dB data

Common: unreliable dB data

If the mixer controls even exist

Machine not owned by the user and user not knowledgable

Machine not owned by the user and user not knowledgable

Means no low-level control on RT =¿ rtkit

No memory locking

Compromises, compromises, compromises

# Cooperation PA, JACK

Use more standard desktop infrastructure. Tear down the seperating walls.

Cooperation on APIs?

Cooperation on Timing?

Cooperation on Protocols? RTP?

Come to LPC!

Come to LPC!

Meet RT folks, V4L folks, MM folks

That's all, folks.

That's all, folks.

Any questions?