WHAT'S NEW IN GSTREAMER?

FOSDEM

3 February 2018, Brussels

Tim-Philipp Müller < tim@centricular.com >
WHO AM I?

- GStreamer core developer, maintainer, backseat release manager
- Centricular co-founder
WHAT IS GSTREAMER?

Most of you know this, so key points only.
Framework for multimedia processing.
Cross-platform, toolkit agnostic.
Any and all use cases.
Set of libraries and plugins.
Abstract API, very extensible.
We often wrap other libraries.
Low-level API and high-level API: playbin, encodebin, RTSP server, non-linear editing, VoIP etc.
Integration with other frameworks and projects

e.g. WebKit/Blink, Clutter, OpenGL

Windows, OS X, Android, iOS

Goal is to adapt to and integrate with other platforms and frameworks (inputs, outputs, decoders, DSPs/GPUs..)
SO, WHAT HAVE WE BEEN UP TO?
RELEASES!

• 1.12: May 2017
• 1.14: coming up real soon now
SOME THINGS THAT LANDED IN 1.12 ...
VIDEOSCALE + VIDEOCONVERT

Multi-threaded scaling and conversion.
New ttml plugin.

Supports the EBU-TT-D profile for now.

Has potential as general intermediary representation for text subtitles.
SPLITEMUXSINK

Buffer collection and scheduling rewritten.
Splitting should be deterministic now.
New "format-location-full" signal.
Automatically starts new chunk if caps change.
DASH TRICK MODE PLAYBACK
LOTS OF NEW FEATURES AND PERF IMPROVEMENTS FOR EMBEDDED

v4l, dmabuf, zero-copy, openmax etc.

--> Olivier's talk
HARDWARE-ACCELERATED VIDEO ENCODING/DECODING

- New msdk plugin for Intel's Media SDK
- GStreamer VA-API: new features + fixes
  Encoders are now autoplugged.
- nvdec for NVIDIA hardware + nvenc improvements
- v4l + omx for embedded
SOME COOL THINGS THAT WILL LAND IN 1.14 ...
AOMEDIA AV1 SUPPORT

Next-gen royalty-free video codec!
Experimental, but bitstream stabilised (ish?).
Encoding still "a bit" slow, but works.

-> Tim's talk at 5pm today in this room.
IPCPIPELINE

Split pipelines across multiple processes!
Isolate demuxers, parsers, decoders.

--> Olivier's talk!
RING BUFFER FOR DEBUG LOGS

Useful for long running tasks.
Or disk space constrained environments.
Retrieve log lines when things fail.
TRACING FRAMEWORK IMPROVEMENTS

- Leak tracer: stack traces, snapshotting, live objects
- Latency tracer improvements are also in the pipeline
HLSSINK2

Takes elementary streams as input
Uses splitmuxsink internally
Handles transmuxing better, without encoder up front
RTSP

- RTSP 2.0 support in client + server
- ONVIF audio backchannel (coming up)
MISSION: PLUGIN MOVES AND MODULE CONSOLIDATION …

Ongoing effort to move things from -bad into other modules.

Going strong!
MP3 PATENTS HAVE EXPIRED!

mpg123 mp3 decoder -> good
LAME mp3 encoder -> good
twolame mp2 encoder -> good
AC-3 PATENTS HAVE EXPIRED!

a52dec has to stay in -ugly

liba52 is GPL :(
COMING SOON (HOPEFULLY!)

• aggregator base class -> core (done)
• audiomixer + audioaggregator -> base
  incl. audiomixer conversion support
• compositor + videoaggregator -> base
• Next: port muxers, for defined-latency
GSTREAMER OPENGL INTEGRATION HAS MOVED TO -BASE!

Library + plugins

API stable now.
WEBRTC

"How do I stream to my web browser?"

It's going to work everywhere.

It's going to be big.
WEBRTC: WEBRTCBIN

New GStreamer WebRTC plugin and library
Just landed in git master \o/ \o/ \o/
Uses libnice + plugins for ICE.
-> Nirbheek's blog post
-> https://github.com/centricular/gstwebrtc-demos
WEBRTC: EXISTING EFFORTS

- OpenWebRTC
- Kurento
- some proprietary SDKs
- libwebrtc: works, but ..
WEBRTC BIN: ADVANTAGES

- very flexible + full control
- easy to use with hardware encoders/decoders, zero-copy capture/render where supported
- leverage existing GStreamer support on embedded devices
- easy to feed already-encoded content
- no need to deal with or fork libwebrtc
WEBRTCBIN: CAVEATS

- SDP renegotiation is not implemented or incomplete, so dynamically adding/removing streams doesn't work yet
- recvonly streams aren't implemented properly yet, but that's fixable
- internals are sound and follow the spec, so easy to identify clear spaces and fit in missing features.
- help wanted!
Generic per-frame KLV metadata (soon)
Muxing / demuxing KLV meta per frame
PERFORMANCE OPTIMISATIONS..

Everywhere, of course.

Lots of things in the pipeline.
SRT - "SECURE, RELIABLE, TRANSPORT"

Source/Sink elements now merged.

Seems well-placed to replace RTMP.
MESON BUILD SYSTEM STATUS UPDATE:

Mostly complete, just a few things still missing:

- full set of options
- static build untested
- iOS/Android untested
- cerbero integration

MSVC build works. Also: gst-build
RUST

Let's talk about Rust!

("Fast, safe and productive - pick three.")
RUST

No plans to switch to it in the short run.

No plans to make it a hard dep in core components.

We can do a lot without breaking GStreamer backwards compat.

We're playing with it, experimenting, gaining experience.

Something for the longer term.
GSTREAMER RUST BINDINGS

Should be in pretty good shape now.

--> Sebastian's talk tomorrow 11am in the Rust devroom
GSTREAMER C# BINDINGS

Have also been resurrected and updated.
ROOM FOR IMPROVEMENT ?
ADAPTIVE STREAMING (DASH, HLS, ETC.)

Really need to improve our creation story!
MAKE IT EASIER TO WRITE "SIMPLE SERVERS"

- souphtpserversink
- rtspserversink
THAT'S ALL FOLKS
THANK YOU (AND THANKS TO THE ORGANISERS!)
QUESTIONS? COMMENTS?
PS:

- follow us on Twitter @GStreamer
- find us on IRC in #gstreamer on FreeNode
- GStreamer hackfest in spring
- GStreamer Conference in Edinburgh in ~late October