GStreamer State of the Union

GStreamer Conference, Berlin
10 October 2016

Tim-Philipp Müller <tim@centricular.com>

Centricular

Introduction

- GStreamer core developer,
  co-maintainer, backseat release manager

- have been using and hacking on
  GStreamer for more than 10 years,
  both for fun and profit

- Centricular co-founder

What is GStreamer ?

- most of you know this, so key points only

What is GStreamer ?

- framework for multimedia processing

- cross-platform, toolkit agnostic

- any and all use cases

- set of libraries and plugins

- abstract API, very extensible

- often wrapping other libraries
  (for decoders, encoders, filters, etc.)

What is GStreamer ? (cont'd)

- 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, Cogl, 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!

- settling on a nice 6-monthly(ish) release schedule

- 1.8 was released in March 2016

- 1.10 is scheduled to be released Really Soon Now ™

- release notes

New API - smaller bits

- GstController: can specify absolute values (not just 0-1 range)

- add API to get property change notifications via bus messages

- bin: add "deep-element-added" and "deep-element-removed" signals

- playbin: new "element-setup" signal

- error messages: can add additional structured details, e.g. HTTP status codes

- official API for redirect messages

New API - cont'd

- tracer framework
  - leaks tracer - works on windows etc., but limitations

- audio conversion/resample library, all done in one step (but elements still separate)

- SMPTE timecode support: GstVideoTimeCode API plus utility functions and GstVideoTimeCodeMeta for buffers

- GstPlayer now available in -bad

  - maybe into -base next cycle

  - try it out!

New features - Stream API

- GstStream: type, caps, tags, etc.

- GstStreamCollection: collection of streams

- plus message/event API to go with all that

- used in parsebin / decodebin3 / playbin3

New features - playbin3

Components:

- parsebin

- decodebin3 / playbin3

  - dynamic streams

  - fast switching

  - only once decoder per stream

  - multiple external inputs (e.g. subs)

- works fine even if plugins (demuxers, parsers, etc.) don't support the new streams stuff yet

New elements and plugins

- webrtcdsp: echo cancellation element

- FDK (Fraunhofer) AAC encoder and decoder

- lv2 audio effects and generators (supersedes LADSPA)

- kmssink (fbsink from this century)

New module: gstreamer-vaapi

- now maintained upstream with GStreamer

- released in lockstep with rest of GStreamer, tighter integration

- driver whitelist: only intel and mesa by default

- runtime feature checking

- perf and stability improvements

- vp9 encoding is ready for when the hardware for it comes out

RTP/RTSP improvments

- much improved retransmission (rtx) support

- jitterbuffer fixes and performance improvments

- RTSP server fixes

- RFC 7273 clock and timing info distributed over SDP

- rtspclientsink, a new RTSP RECORD sink element in gst-rtsp-server

- vp9 rtp payloader/depayloader

- H265 payloader/depayloader fixes

OpenGL improvments

- qmlglsink now works on wayland and windows and the RPi

- gl memory was rewritten
  - enabled eglmemory
  - iosurface memory
  - PBOs
  - generally more extensible

- egl does gl3 now

- gl based colorbalance, videoflip

- gldeinterlace now works with GL ES

- affine transformation meta

Other improvements

- vulkan support

- bluez a2dp metadata + a2dpsink works again

- DVB-T2 support works now

- adaptive streaming improvements (clientside)

  - HLS alternative renditions

  - DASH trick modes

- vp8/vp9 encoder now uses multiple threads

- improved jpeg2000 support across the board

Cross-platform support: Android

- JNI java bindings coming up

- Android zerocopy decode/render

- Android camera source and sensor source

- tinyalsasink

Cross-platform support: macOS/iOS

- OS/X: GstDeviceProvider support for querying available devices

- iOS: "zerocopy" encoding

Cross-platform support: Windows

- many smaller fixes, ksvideosrc

- MSVC toolchain builds

- Visual Studio support coming up

- Cerbero bits should land in the next cycle

Future

- dmabuf support for waylandsink + gst-vaapi (zerocopy)

- EBU-TT-D TTML subtitle renderer + meta

Future: move more things out of -bad

- people worry

- we provide no indication what's "good bad" and "bad bad"

  - (maybe we should?)

e.g.

- move gl lib to base

- move aggregator to base

- rewrite muxers on top of aggregator

- move more plugins to base/good (e.g. tsdemux?)

Future: Documentation

- old gstreamer.com SDK tutorials + docs have been made available under CC license by Fluendo

- are being updated for 1.x and changes

- will land on gstreamer.freedesktop.org very very soon

Future: Documentation - II

- switch off (redirect) gstreamer.com

  - was transfered to GStreamer by Fluendo

- merge all docs into one docs module

Future: Examples

- new gst-examples repository

- we'll populate it slowly

Future: GStreamer Foundation

- is being set up

- low key role, purely administrative

- proper org might also enable things like dev certificates to sign binary releases

Future: Build System

- experimental support for Meson has been merged

- will enable better dev/deployment experience

  - like MSVC/Visual Studio support

- will take some time to mature, but strong momentum

  - "gst-all" will land soon (name tbd), try it!

- goal is to retire autotools

  - there will be a transition period

Room for improvement?

- adaptive streaming (DASH, HLS, etc.)

    - fairly good client-side story, but lousy creation story

- make it easier to write "simple servers"

    - souphttpserversink

    - rtspserversink

    - etc.

- windows, ios, android: more sink elements that work with the native UI toolkits like qmlglsink/gtksink etc.

What else should we do better?

- Developer Tools (old story)

    - slow progress, but some are appearing

- Daily builds (soon hopefully)

Also: we still need better docs !

- more docs

  => new docs

- different docs (tutorials, howtos)

  => new docs

- demo code/apps written as examples

  => gst-examples

- GStreamer book

Summary

We made lots of progress on a wide range of topics we found in need of improvement last year, but of course there's always a lot more to do!

That's all folks

Thank you and enjoy the conference !

Questions or Comments ?