What's new in GStreamer

FOSDEM
2 February 2013
Brussels

Tim-Philipp Müller <tim.muller@collabora.co.uk>
Quick intro

• who am I?
• what is GStreamer?
• 0.10, 0.11, 1.0 and all that
What is GStreamer?

• set of libraries

• pipeline-based: elements, components, pads, data flow

• plugins, abstract API

• often wrap other libraries (decoders, encoders, filters, etc.)
What is GStreamer? (cont'd)

- low-level API and high-level API
  - playbin, decodebin, encodebin, gst-editing-services, gst-rtsp-server, gst-streaming-server
  - (and farstream, telepathy etc.)

- integration with other frameworks and projects
  - e.g. webkit, clutter, Windows/OSX/Android/iOS

- goal is to adapt to platform/framework (inputs, outputs, decoders, DSPs/GPUs etc.)
So what's new?
So what's new?

• GStreamer 1.0 is finally out!
So what's new?

• GStreamer 1.0 is finally out!
  • released in September 2012
  • followed by 1.0.x bug-fix releases
Versions and all that

- stable API/ABI, backwards compatibility
- old stable API: 0.10.x
- old unstable/development API: 0.11.x
- new stable API series: 1.x.y = 1.0
Versions and all that (cont'd)

• new stable 1.0 API series:

  • New Versioning Scheme
    • 1.0.0, 1.0.1, 1.0.2, 1.0.3... bug fix point releases
    • 1.1.0, 1.1.1, 1.1.2, 1.1.3... pre-releases
    • 1.2.0, 1.2.1, 1.2.2, 1.2.3... bug fix point releases

• More frequent bug-fix releases:

• Cherry-pick fixes into a stable branch
But before we continue, a public service announcement
We need to talk about 0.10
0.10 is dead
(sorry)
0.10 is dead end-of-life
Aaargh!

You should move to the new version.
Understanding the 0.10 → 1.0 transition

• most things are still the same

• evolution, not revolution (no CADT)

• application interfaces mostly the same

• make things more generic (caps, pad functions), i.e. just reshuffling some bits

• split ops that are not related, e.g. buffer alloc + negotiation

• overall complexity has not hugely increased
  • threading model is the same and well-understood
  • scheduling remains the same
  • most complex elements (e.g. demuxers) were barely changed
Major changes

• reworked memory allocation / handling

• simplified and more flexible audio/video API

• renegotiation support for all pipeline configurations [theoretically]

• dynamic pipelines should be much easier [citation needed]

• watch GStreamer Conference 2012 video by Wim Taymans at http://gstconf.ubicast.tv
That sounds great, but does it work?

• much better than expected
Ongoing and Future Work

• new openmax wrapper: gst-omx
• raspberry-pi support (via gst-omx)
• VDPAU/VA-API hardware-accelerated video decoding (this time for real)
• MFC video4linux codec API (Samsung Exynos + some Freescale IMX)
• better, more natural OpenGL integration (generically, but also e.g. eglglessink)
• generic context sharing / distribution
Ongoing and Future Work (cont'd)

• multiview (3D) video

• advanced subtitle features

• DASH (client and server)

• optimisations: use all the nifty new features we've added to GStreamer in 1.0
Ongoing and Future Work (cont'd)

• better device discovery and probing (GstPropertyProbe replacement)

• smooth as butter stream switching, and stream activation/deactivation

• playlist support

• gst-plugins-bad module split? Add gst-plugins-staging?

• bluetooth plugins move to GStreamer
Ongoing and Future Work (cont'd)

• development tools / debugging
  • better error reporting
  • tools collection
  • ideas floating about about something like "dot file" dumps, but more interactive
GStreamer Conference

c.a. 21-22 October 2013

Edinburgh, Scotland

co-hosted with LinuxCon and the Embedded Linux Conference Europe (ELCE)

See you there hopefully!
Questions?
Thank You!