GSTREAMER-VAAPI

HARDWARE-ACCELERATED ENCODING AND DECODING ON INTEL® HARDWARE

Victor Jaquez

GStreamer Conference 2015 / 8-9 October (Dublin)

VA-API

VIDEO ACCELERATION

—

APPLICATION PROGRAMMING INTERFACE

WHAT IS VA-API?

- An API specification
- A library implementation
- Open Source MIT license

WHAT DOES IT DO?

1. Enables hardware accelerated video decode and encode.

   Entry-points: VLD, IDCT, Motion Compensation, etc.

   Codecs: MPEG-2, MPEG-4 ASP/H.263, MPEG-4 AVC/H.264, VC-1/VMW3, JPEG, VP8, VP9.

2. Sub-picture blending and rendering

3. Video post processing:

   Color balance, skin tone enhancement, de-interlace, scaling, etc.

HOW'S THE IMPLEMENTATION?

libva

http://cgit.freedesktop.org/vaapi/libva/

- It is a front-end
- Opens and registers a backend

WHICH BACKENDS?

- i965_drv_video.so (HD Intel driver)
- ~~vdpau_drv_video.so~~ (VDPAU —noveau/nvidia/s3g— bridge)
- ~~xvba_drv_video.so~~ (XvBA —fglrx— bridge)
- ~~pvr_drv_video.so~~ (PVR bridge)
- gallium_drv_video.so (Gallium bridge!)
- hybrid_drv_video.so

  (another HD Intel driver, does decoding and encoding using either CPU and GPU)

vainfo

```
libva info: VA-API version 0.38.0
libva info: va_getDriverName() returns 0
libva info: Trying to open /opt/gnome/jh/lib/dri/i965_drv_video.so
libva info: Found init function __vaDriverInit_0_38
libva info: va_openDriver() returns 0
vainfo: VA-API version: 0.38 (libva 1.6.2.pre1)
vainfo: Driver version: Intel i965 driver for Intel(R) Haswell Mobile
vainfo: Supported profile and entrypoints
      VAProfileMPEG2Simple            : VAEntrypointVLD
      VAProfileMPEG2Simple            : VAEntrypointEncSlice
      VAProfileMPEG2Main              : VAEntrypointVLD
      VAProfileMPEG2Main              : VAEntrypointEncSlice
      VAProfileH264ConstrainedBaseline: VAEntrypointVLD
      VAProfileH264ConstrainedBaseline: VAEntrypointEncSlice
      VAProfileH264Main               : VAEntrypointVLD
      VAProfileH264Main               : VAEntrypointEncSlice
      VAProfileH264High               : VAEntrypointVLD
      VAProfileH264High               : VAEntrypointEncSlice
      VAProfileH264MultiviewHigh      : VAEntrypointVLD
      VAProfileH264MultiviewHigh      : VAEntrypointEncSlice
      VAProfileH264StereoHigh         : VAEntrypointVLD
      VAProfileH264StereoHigh         : VAEntrypointEncSlice
      VAProfileVC1Simple              : VAEntrypointVLD
```

```
VAProfileJPEGBaseline           : VAEntrypointVLD
VAProfileH264MultiviewHigh      : VAEntrypointVLD
VAProfileH264MultiviewHigh      : VAEntrypointEncSlice
VAProfileH264StereoHigh         : VAEntrypointVLD
VAProfileH264StereoHigh         : VAEntrypointEncSlice
```

vainfo

```
libva info: VA-API version 0.38.0
libva info: va_getDriverName() returns 0
libva info: Trying to open /home/ceyusa/jh/lib/dri//nvidia_drv_video.
libva info: Found init function __vaDriverInit_0_38
libva info: va_openDriver() returns 0
vainfo: VA-API version: 0.38 (libva 1.6.2.pre1)
vainfo: Driver version: Splitted-Desktop Systems VDPAU backend for VA
vainfo: Supported profile and entrypoints
      VAProfileMPEG2Simple            : VAEntrypointVLD
      VAProfileMPEG2Main              : VAEntrypointVLD
      VAProfileMPEG4Simple            : VAEntrypointVLD
      VAProfileMPEG4AdvancedSimple    : VAEntrypointVLD
      VAProfileH264Baseline           : VAEntrypointVLD
      VAProfileH264Main               : VAEntrypointVLD
      VAProfileH264High               : VAEntrypointVLD
```

HOW'S THE API?

- VADisplay
  - X11, DRM, Wayland, Android, etc.
- VAConfigID
  - VLD for requested codec.
- VAContexID
  - "Virtual" video processing pipeline.
- VASurfaceID
  - Render targets.
  - Not accessible to the client.
- VABufferID
  - data, parameters, quantization matrix, slice info, etc.

MPEG2 DECODE

INITIALIZATION

```
dpy = vaGetDisplay(x11_display);

vaCreateConfig(dpy, VAProfileMPEG2Main, VAEntrypointVLD,
               &attr, 1, &cfg);
vaCreateSurfaces(dpy, VA_RT_FORMAT_YUV420, w, h, &surface, 1,
                 NULL, 0);
vaCreateContext(dpy, cfg, w, h, VA_PROGRESSIVE, &surface, 1,
                &ctxt);
```

MPEG2 DECODE

FILL DATA

```
vaCreateBuffer(dpy, ctxt, VAPictureParameterBufferType,
               sizeof(VAPictureParameterBufferMPEG2), 1,
               &pic_param, &pic_param_buf);

vaCreateBuffer(dpy, ctxt, VAIQMatrixBufferType,
               sizeof(VAIQMatrixBufferMPEG2), 1,
               &iq_matrix, &iqmatrix_buf);

vaCreateBuffer(dpy, ctxt, VASliceParameterBufferType,
               sizeof(VASliceParameterBufferMPEG2), 1,
               &slice_param, &slice_param_buf);

vaCreateBuffer(dpy, ctxt, VASliceDataBufferType,
               slice_size, 1, slice_data, &slice_data_buf);
```

MPEG2 DECODE

DECODE AND DISPLAY

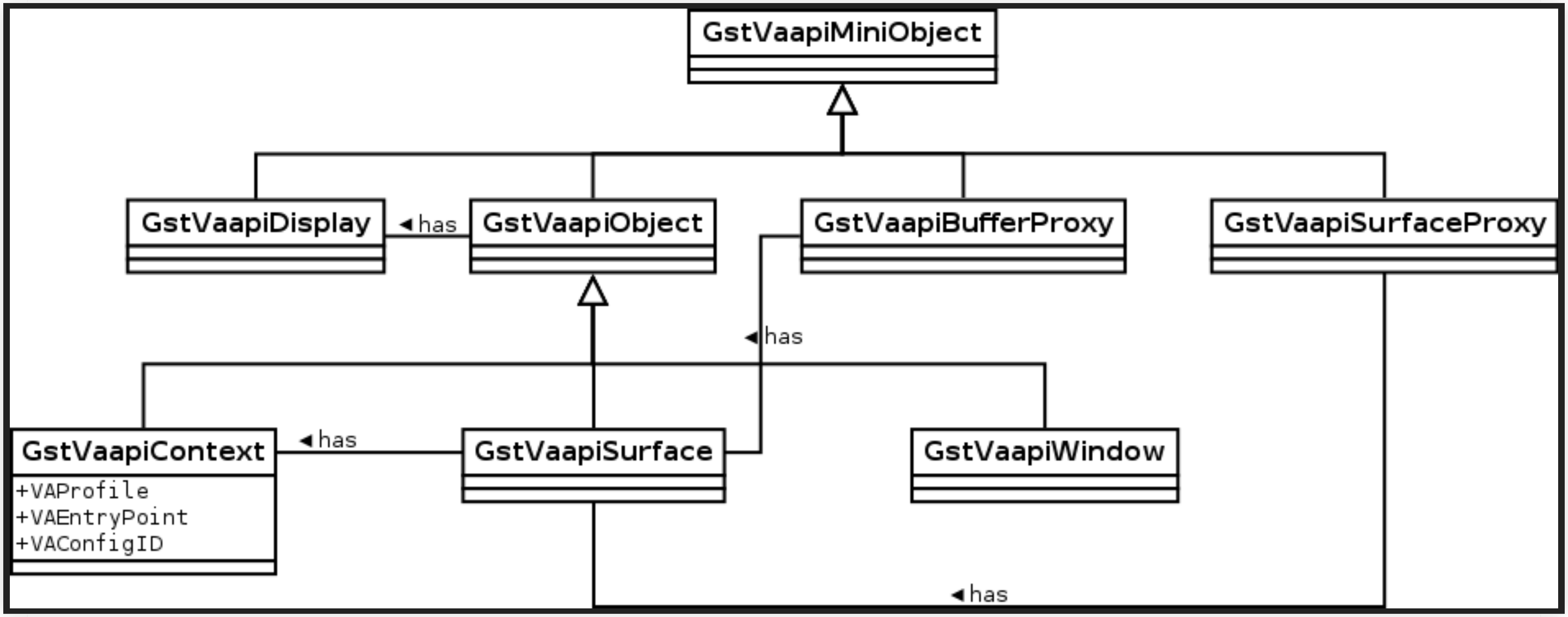
```
vaBeginPicture(dpy, ctxt, surface);
vaRenderPicture(dpy, ctxt, &pic_param_buf, 1);
vaRenderPicture(dpy, ctxt, &iqmatrix_buf, 1);
vaRenderPicture(dpy, ctxt, &slice_param_buf, 1);
vaRenderPicture(dpy, ctxt, &slice_data_buf, 1);
vaEndPicture(dpy, ctxt);

vaSyncSurface(dpy, surface);

vaPutSurface(dpy, surface, x11_window, sx, sy, sw, sh, dx, dy, dw, dh
```

WHAT IS GSTREAMER-VAAPI?

- A helper library (libgstvaapi)
- A set of GStreamer elements
- Supports from GStreamer-1.2 to 1.6

LIBRARY

Wraps almost all VA-API concepts



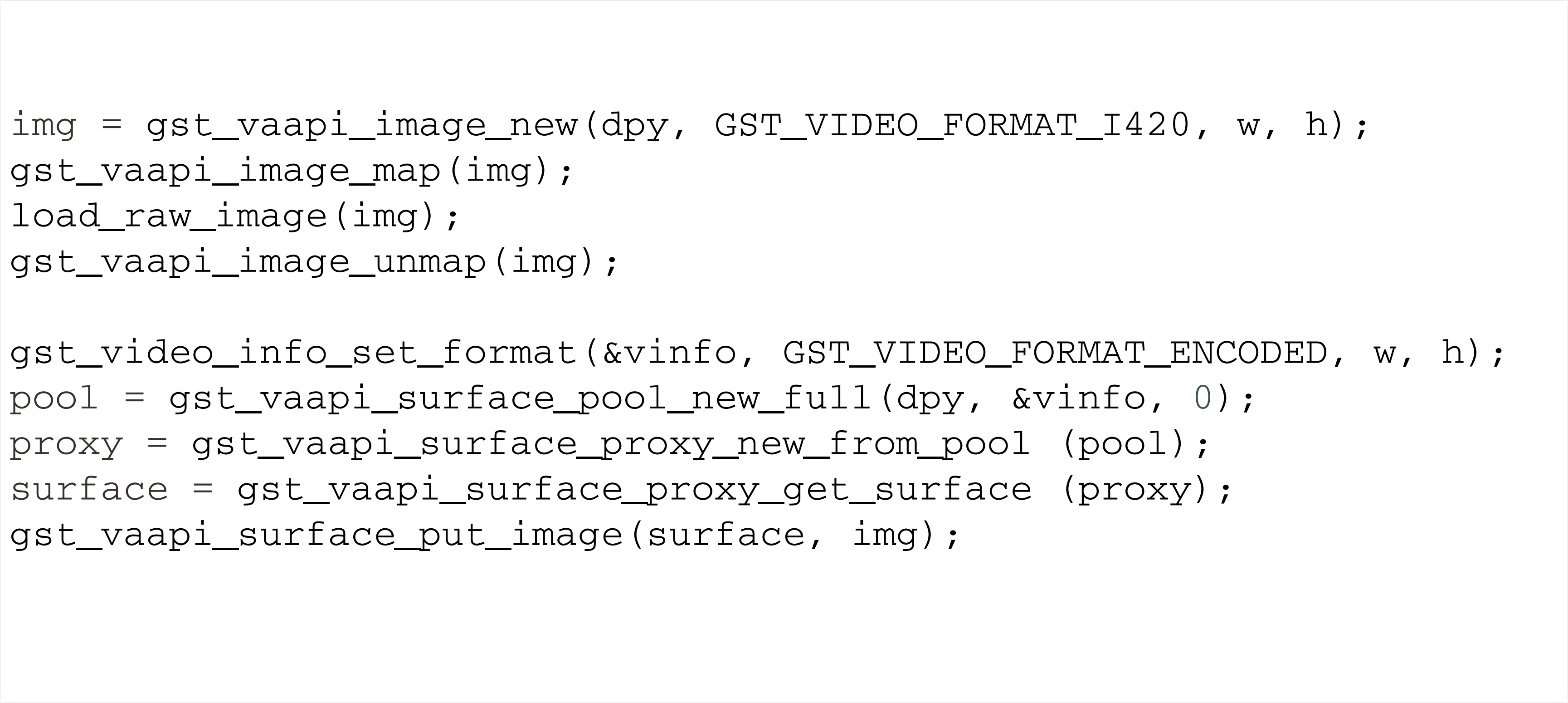
ENCODER H264

INITIALIZATION

```
dpy = gst_vaapi_display_x11_new(NULL);
enc = gst_vaapi_encoder_h264_new(dpy);
st = new_gst_video_codec_state(w, h, fps_n, fps_d);
gst_vaapi_encoder_set_codec_data(enc, st);
```

ENCODER H264

FILL DATA



```
img = gst_vaapi_image_new(dpy, GST_VIDEO_FORMAT_I420, w, h);
gst_vaapi_image_map(img);
load_raw_image(img);
gst_vaapi_image_unmap(img);

gst_video_info_set_format(&vinfo, GST_VIDEO_FORMAT_ENCODED, w, h);
pool = gst_vaapi_surface_pool_new_full(dpy, &vinfo, 0);
proxy = gst_vaapi_surface_proxy_new_from_pool (pool);
surface = gst_vaapi_surface_proxy_get_surface (proxy);
gst_vaapi_surface_put_image(surface, img);
```

ENCODER H264

ENCODE

```
frame = g_slice_new0(GstVideoCodecFrame);
gst_video_codec_frame_set_user_data(frame,
    gst_vaapi_surface_proxy_ref(proxy),
    gst_vaapi_surface_proxy_unref);

gst_vaapi_encoder_put_frame(encoder, frame);

gst_vaapi_encoder_get_buffer_with_timeout(enc, &encbuf, 5000);
buf = gst_buffer_new_and_alloc(gst_vaapi_coded_buffer_get_size(encbuf
gst_vaapi_coded_buffer_copy_into(buf, encbuf);
```

LIBRARY

Implements video codec parsers.

OpenGL helpers (EGL and GLX).

Windowing protocol helpers (DRM, Wayland, X11).

SPLIT LIBRARIES

- libgstvaapi
- libgstvaapi-drm
- libgstvaapi-x11
- libgstvaapi-glx
- libgstvaapi-egl
- libgstvaapi-wayland

GStreamer elements

- vaapidecode: VA-API decoder
- vaapisink: VA-API sink

- vaapipostproc: VA-API video post-processing
- vaapidecodebin: VA-API decode bin

GStreamer elements

Encoders

- vaapiencode_h264: VA-API H.264 encoder
- vaapiencode_mpeg2: VA-API MPEG-2 encoder
- vaapiencode_jpeg: VA-API JPEG encoder
- vaapiencode_vp8: VA-API VP8 encoder
- vaapiencode_h265: VA-API H.265 encoder

GStreamer elements

Parsers

- vaapiparse_h264: H.264 parser
- vaapiparse_h265: H.265 parser

CHALLENGES

THOU SHALL NOT PARSE TWICE

- 691712 — codecparsers: add GstMetas to pass parsing results downstream
  - 704865 — Add mpeg2 slice header information to GstMpegVideoMeta

THOU SHALL AUTO-PLUG HW VIDEO FILTERS (DE-INTERLACERS)

- 687182 — playbin: autoplugging s/w and h/w accelerated deinterlacers
- Remove vaapidecodebin

THOU SHALL REVERSE PLAY-BACK

- 747574 — videodecoder: reverse playback in non-packetized decoders
- Handle when the number of buffers in the GOP is bigger than the maximum buffers in the video buffer pool.

THOU SHALL SUPPORT DMABUF

- 755072 — vaapi: expose memory:dmabuf capsfeature

THOU SHALL SUPPORT OPENGL / OPENGL3 / OPENGL-ES

- 755406 — [metabug] GL bugs

THOU SHALL NOT CACHE THE DISPLAY

- 747946 — Remove display cache
- 754820 — Using multiple instances of vaapisink in one application cause problems

QUESTIONS?

THANK YOU!

- Twitter: @ceyusa
- Mail: vjaquez at igalia dot com
- Blog: http://blogs.igalia.com/vjaquez