SAMSUNG

GPU virtualization

2022

Samsung R&D Institute China Xi'an(SRCX) established in 2013
and located in Xi'an High-tech Zone, is the only cutting-edge technology
R&D institute of Samsung Electronics in western China.

SRCX

西安三星电子研究所 | SAMSUNG R&D INSTITUTE CHINA XIAN

目　录 Contents

01. Wayland and virtio-GPU

02. Comparision

SAMSUNG

Wayland and virtio-GPU

01

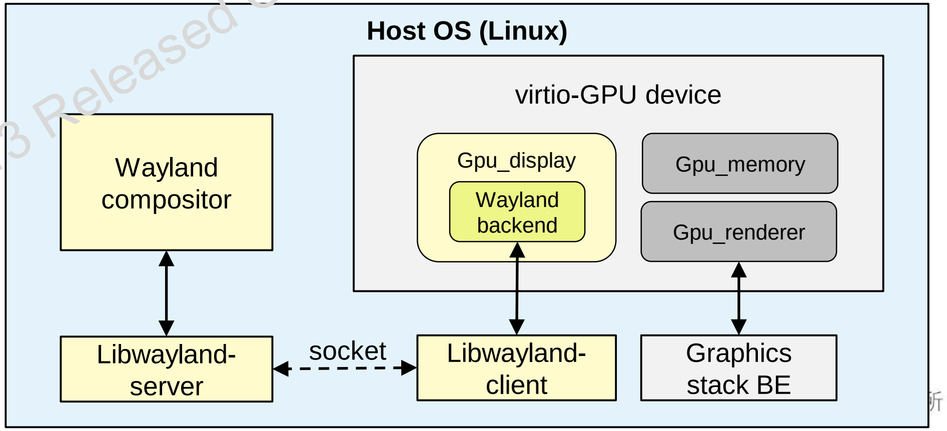
西安三星电子研究所 | SAMSUNG R&D INSTITUTE CHINA XIAN

Virtio-GPU display backends

SAMSUNG Confidential

- virtio-GPU device can be divided into following modules by logic:
  - Gpu_renderer cooperates with graphics stack BE to parse the graphics command transferred from guest OS, and execute them on host native graphics library.
  - Gpu_memory allocate memory for guest OS framebuffer usage
  - Gpu_display is responsible for transfer the target framebuffer of guest OS to the compositor of host OS for display
- There are three types display backend on gpu_display. Which to use is determined in startup phase by trying them in the following order until one of them connects successfully
  - Wayland: a protocol for a compositor to talk to its clients, and intended as a simpler replacement for X11
  - X11: a window system common on Unix-like operating systems
  - VNC: a graphical desktop-sharing system that uses the Remote Frame Buffer protocol (RFB) to remotely display and control another system.
- Currently Wayland display backend is used on Samsung virtio-GPU solution,

20220413 Released Only to Harman Under NDA

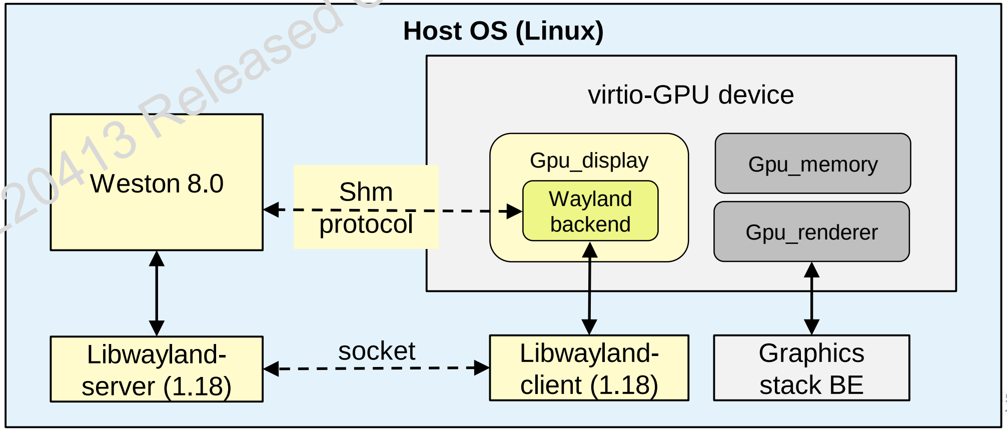


Virtio-GPU Wayland display backend

SAMSUNG Confidential

- Wayland display backend interacts with Wayland compositor as a Wayland client
- Wayland display backend calls Wayland APIs directly, like wl_display_connect(), wl_surface_attach(), wl_surface_commit() etc
- Wayland display backend supports shm buffer sharing protocol:
  - shm protocol: default Wayland buffer sharing protocol, which used to transfer shared CPU memory between Wayland client and server
- There is memory copy from guest target framebuffer into shared CPU memory with shm buffer sharing protocol, and will have some impact on performance.
- There is no close relationship between Wayland display backend and other modules of virtio-GPU device, the main thing is that graphics stack BE will copy guest target framebuffer into shared CPU memory, and Wayland display backend will transfer this shared CPU memory to Wayland compositor for display.
- Samsung virtio-GPU solution choose Weston 8.0 and Wayland 1.18 in host OS, these two are default versions provided by yocto version 3.1

20220413 Released Only to Harman Under NDA

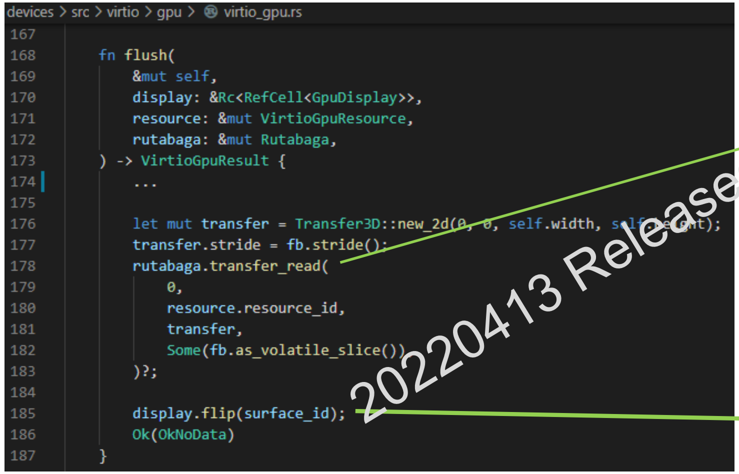


Virtio-GPU display process

SAMSUNG Confidential

- Following illustrates the key part of the display process on virtio-GPU device:
  - Graphics stack BE copy the content of guest OS target framebuffer into shared CPU memory
  - Wayland display backend transfers this shared CPU memory to Wayland compositor for display

Main body of virtio-GPU device

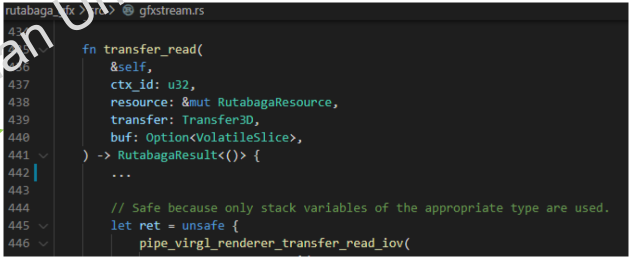

devices > src > virtio > gpu > virtio_gpu.rs

```
    fn flush(
        &mut self,
        display: &Rc<RefCell<GpuDisplay>>,
        resource: &mut VirtioGpuResource,
        rutabaga: &mut Rutabaga,
    ) -> VirtioGpuResult {
        ...

        let mut transfer = Transfer3D::new_2d(0, 0, self.width, self.height);
        transfer.stride = fb.stride();
        rutabaga.transfer_read(
            0,
            resource.resource_id,
            transfer,
            Some(fb.as_volatile_slice())
        )?;

        display.flip(surface_id);
        Ok(OkNoData)
    }
```

1. Copy content from framebuffer to shared CPU memory

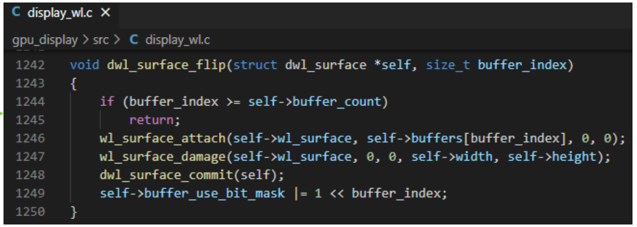
2. Transfer the shared CPU memory to Wayland compositor

Graphcis stack BE - Gfxstream


rutabaga_gfx > src > gfxstream.rs

```
    fn transfer_read(
        &self,
        ctx_id: u32,
        resource: &mut RutabagaResource,
        transfer: Transfer3D,
        buf: Option<VolatileSlice>,
    ) -> RutabagaResult<()> {
        ...

        // Safe because only stack variables of the appropriate type are used.
        let ret = unsafe {
            pipe_virgl_renderer_transfer_read_iov(
```

Wayland display backend


display_wl.c

gpu_display > src > display_wl.c

```
void dwl_surface_flip(struct dwl_surface *self, size_t buffer_index)
{
    if (buffer_index >= self->buffer_count)
        return;
    wl_surface_attach(self->wl_surface, self->buffers[buffer_index], 0, 0);
    wl_surface_damage(self->wl_surface, 0, 0, self->width, self->height);
    dwl_surface_commit(self);
    self->buffer_use_bit_mask |= 1 << buffer_index;
}
```

20220413 Released Only to Harman Under NDA

SAMSUNG

Comparisons

02

SAMSUNG Confidential

Comparisons - performance

- The below table lists the performance of baremetal (BM) Android and virtio-GPU with following condition:
  - Run on ExynosAutoV910
  - Tested on Mali MP12 GPU
  - 4 CPU for Android domain
- The test case gl_manhattan31_off of GFXBench does not work on virtio-GPU now because virtio-GPU does not support OpenGL ES 3.1 yet, only the performance of gl_manhattan_off list below.
- There is a certain performance gap between virtio-GPU and ARM MPT GPU, and still in improvement

Test case		Baremetal Android (FPS)	ARM MPT GPU in Android		Gfxstream mode of virtio-GPU in Android	
			FPS	% of Baremtal GPU	FPS	% of Baremtal GPU
GFXBench	gl_manhattan_off	147.3	145.5	98.7%	91.5	62.1%

20220413 Released Only to Harman Under NDA

Comparisons - performance

SAMSUNG Confidential

- The below table lists the performance of baremetal (BM) Android, MPT and virtio-GPU with following condition:
  - Run on ExynosAutoV910
  - Tested on Mali MP12 GPU
  - 4 CPU for Android domain
  - Each VM runs Manhattan Offscreen simultaneously
  - Baremetal Score = (Linux Baremetal Score + Android Baremetal Score) / 2
  - VM Score = Linux VM Score + Android VM Score

Test case		Baremetal Linux + Android (FPS)	Linux MPT GPU + Android MPT GPU		Linux PT + Gfxstream mode of virtio-GPU in Android	
			FPS	% of Baremtal GPU	FPS	% of Baremtal GPU
GFXBench	gl_manhattan_off	150.2	145.3 L : 71.8 A : 73.5	96.7%	144.9 L : 84.1 A : 60.8	96.4%

- Time slice is 4ms. And Manhattan 3.0 running time is 60sec = 60000ms. While two Manhattan 3.0 offscreen are running, 15000 times VM handovers are happened. That is, GPU switching time spends very short.
- Virtio-gpu's max performance in Android standalone is not high, but total performance fulfills high performance in the parallel running with Linux.

20220413 Released Only to Harman Under NDA

Comparisons - preemption

SAMSUNG Confidential

- There are different granularities of GPU preemption including following aspects:
  - Preempting GPU between different VMs
  - Preempting GPU between different processes in one OS
  - Preemption between tasks executing on the GPU
- Below table lists the GPU preemptions support status for various GPU scenario:

GPU scenario	Preemption support status	Remark
Pass-through (PT)	Support	
MPT GPU	Support	
Virtio-GPU	Not support	Google officially refused to implement EGL_IMG_context_priority extension with following comments: “neither gfxstream/virgl plumb through the guest priority to the host. That makes sense - we don't want the untrusted guests creating high priority contexts.”

20220413 Released Only to Harman Under NDA

2022

THANK YOU

Samsung R&D Institute China Xi'an(SRCX) established in 2013 and located in Xi'an High-tech Zone, is the only cutting-edge technology R&D institute of Samsung Electronics in western China.

西安三星电子研究所 | SAMSUNG R&D INSTITUTE CHINA XIAN