Bhyve guests with hardware accelerated graphics

Michael Chiu

EuroBSDCon 2019

Who am I?

Studying Pure Mathematics in San Jose, California

FreeBSD user since 4 years ago

Accidentally started programming

Reason for GPU accelerated Guests?

Possible methods to accelerate graphics applications of Bhyve guests

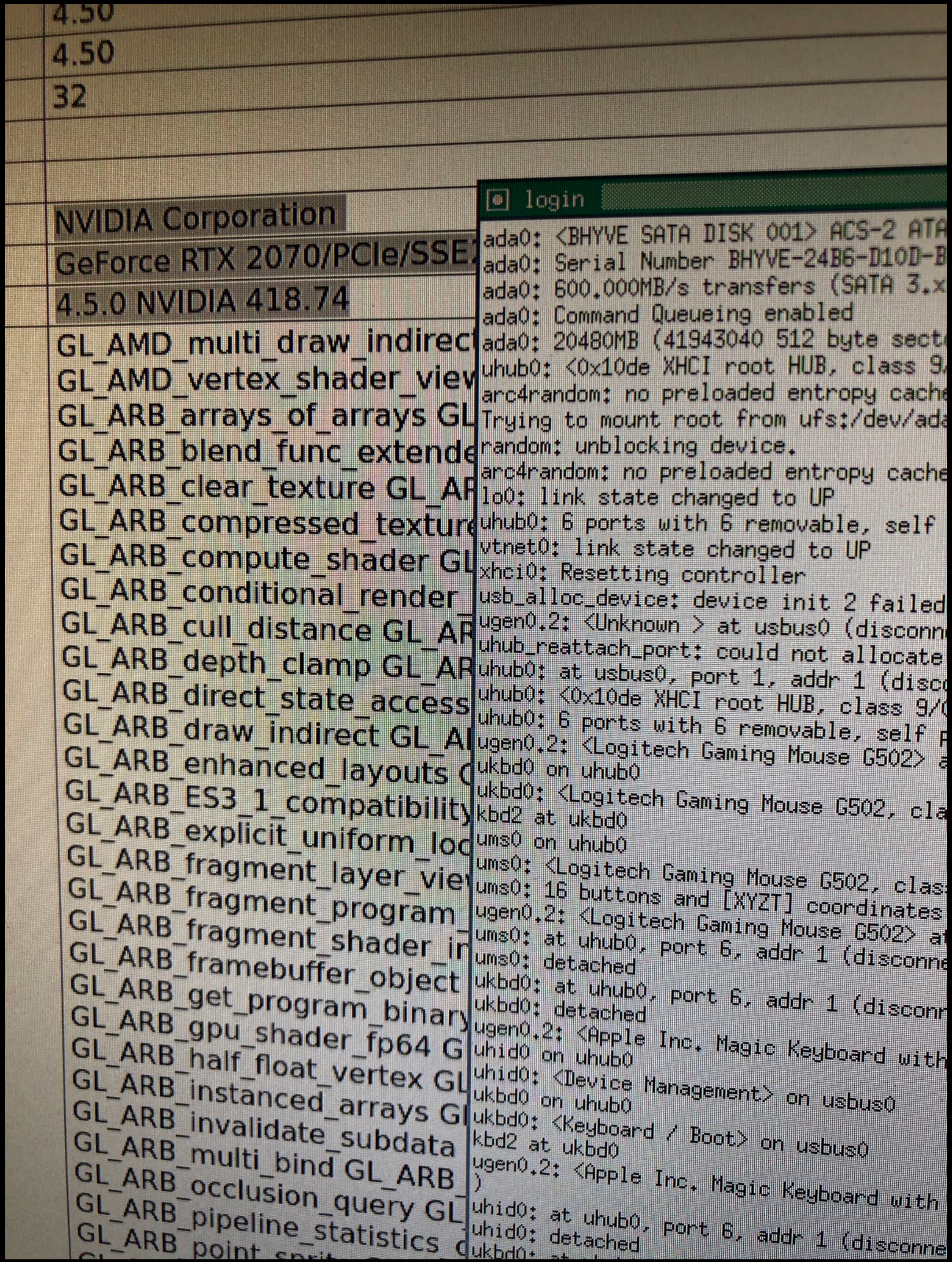
Application Level:

- Offload to another machine / server (for example: VirtualGL, rCUDA)

(Virtualized) Hardware level:

- GPU Passthrough
- vGPU (Intel GVT). ?

Is GPU Passthrough possible in bhyve?



```
4.50
4.50
32

NVIDIA Corporation
GeForce RTX 2070/PCIe/SSE2
4.5.0 NVIDIA 418.74
GL_AMD_multi_draw_indirect
GL_AMD_vertex_shader_view
GL_ARB_arrays_of_arrays GL
GL_ARB_blend_func_extende
GL_ARB_clear_texture GL_AR
GL_ARB_compressed_texture
GL_ARB_compute_shader GL
GL_ARB_conditional_render
GL_ARB_cull_distance GL_AR
GL_ARB_depth_clamp GL_AR
GL_ARB_direct_state_access
GL_ARB_draw_indirect GL_A
GL_ARB_enhanced_layouts G
GL_ARB_ES3_1_compatibility
GL_ARB_explicit_uniform_loc
GL_ARB_fragment_layer_view
GL_ARB_fragment_program_
GL_ARB_fragment_shader_in
GL_ARB_framebuffer_object
GL_ARB_get_program_binary
GL_ARB_gpu_shader_fp64 G
GL_ARB_half_float_vertex GL
GL_ARB_instanced_arrays GL
GL_ARB_invalidate_subdata
GL_ARB_multi_bind GL_ARB
GL_ARB_occlusion_query GL
GL_ARB_pipeline_statistics
GL_ARB_point_sprit
```

login

```
ada0: <BHYVE SATA DISK 001> ACS-2 ATA
ada0: Serial Number BHYVE-24B6-D10D-B
ada0: 600.000MB/s transfers (SATA 3.x
ada0: Command Queueing enabled
ada0: 20480MB (41943040 512 byte sect
uhub0: <0x10de XHCI root HUB, class 9/
arc4random: no preloaded entropy cache
Trying to mount root from ufs:/dev/ada
random: unblocking device.
arc4random: no preloaded entropy cache
lo0: link state changed to UP
uhub0: 6 ports with 6 removable, self
vtnet0: link state changed to UP
xhci0: Resetting controller
usb_alloc_device: device init 2 failed
ugen0.2: <Unknown > at usbus0 (disconn
uhub_reattach_port: could not allocate
uhub0: at usbus0, port 1, addr 1 (disc
uhub0: <0x10de XHCI root HUB, class 9/
uhub0: 6 ports with 6 removable, self p
ugen0.2: <Logitech Gaming Mouse G502> a
ukbd0 on uhub0
ukbd0: <Logitech Gaming Mouse G502, cla
kbd2 at ukbd0
ums0 on uhub0
ums0: <Logitech Gaming Mouse G502, clas
ums0: 16 buttons and [XYZT] coordinates
ugen0.2: <Logitech Gaming Mouse G502> at
ums0: at uhub0, port 6, addr 1 (disconne
ums0: detached
ukbd0: at uhub0, port 6, addr 1 (disconn
ukbd0: detached
ugen0.2: <Apple Inc. Magic Keyboard with
uhid0 on uhub0
uhid0: <Device Management> on usbus0
ukbd0 on uhub0
ukbd0: <Keyboard / Boot> on usbus0
kbd2 at ukbd0
ugen0.2: <Apple Inc. Magic Keyboard with
)
uhid0: at uhub0, port 6, addr 1 (disconne
uhid0: detached
```

It is possible but not perfect

The GPU must not used by host or any other virtual machines prior to passing through to guest.

The only way to use the graphic card again is to reboot

Does not automatically work and behave live physical graphics card for all OSes.

Prerequisites to Passthrough a GPU to guest

0) Any standard bhyve pci-passthrough requirements

1) The GPU must not be initialized yet

2) The Topology of the GPU needs to be trivial
Counter example: Nvidia Optimus

~~3) Donate to one of the BSD projects~~

Attempts to work around the GPU initialization problem

Nvidia RX 2070 seems to lie about supporting FLR (function level reset)

Modified and make ppt to ignore FLR and force `pci_power_reset` devices listed in pptresetdevs in loader.conf

Nope, that does not help.

Real World Experiments

CPU: AMD Threadripper 1950X
RAM: 64GB 2933Hz DDR4

Motherboard: Gigabyte X399 Designare EX

OS: FreeBSD 13 CURRENT

GPU0 host, 1st PCIe Slot: Nvidia GTX 1050TI
GPU1 guest, 3rd PCIe Slot: Nvidia RTX 2070
GPU2 guest, 3rd PCIe Slot: AMD RX 580

All Guests installed on Samsung T5 SSD

Windows 10 Guests

Nope, won’t even boot properly

Possible reasons for failed passthrough to Guests (for example Windows)

According to Debian VGAPassthrough Wiki,

“The PCI geometry they create by default may violate guest OS graphics drivers' expectations by placing the GPU directly on the PCI root bus, or behind a PCI (rather than PCI Express bridge)”

But in order to Passthrough the GPU as X:0:0 in guest,
we need to pass -Y to bhyve, but this usually crash bhyve as a result of

```
Assertion failed: (pi->pi_bar[baridx].type == PCIBAR_IO),
                 function passthru_read
```

FreeBSD Guests

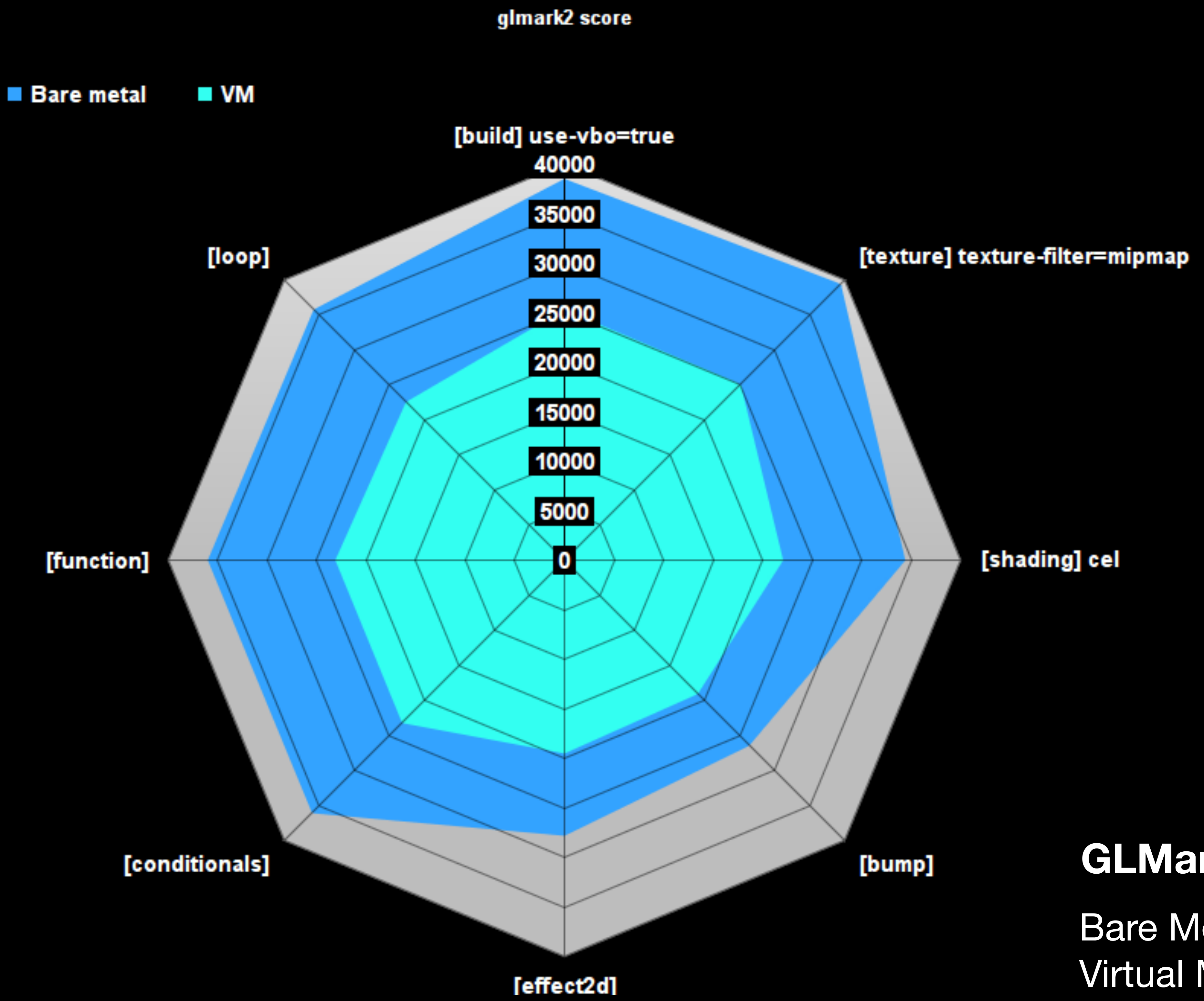
FreeBSD 12.0 Guest (Nvidia RTX 2070)

- Use Nvidia official driver for FreeBSD
- Must use uefi loader to enable vt console
- Console does not show on screen (as expected)
- Need to explicitly add BusID to xorg.conf
- Remove nvidia*_load="YES" from loader.conf
- Add nvidia and nvidia-modeset to kld_list instead
- Bonus: the USB-C port on the GPU works too (as xhci)

FreeBSD 12.0 Guest (RTX 580)

- Used drm-kmod driver
- bootable with uefi loader, bhyveload not tested
- Console show on screen after amdgpu loaded
- Using DRM, so does not require BusID in xorg.conf

Performance compare to bare metal



GLMark2 Score

Bare Metal: 26437
Virtual Machine: 17482

Intel GVT?

Intel technology that allow to create virtual GPU from iGPU, that can pass through to virtual machines.

Most of the code are already in i915 DRM driver

Future work/Work in progress