I've got a lot of exciting stuff in the pipe now, but for today I'm just going to talk a bit about resource invalidation: what it is, when it happens, and why it's important.

[...]

Resource invalidation can occur in a number of scenarios, but the most common is when unsetting a buffer's data, as in the above example. The other main case for it is replacing the data of a buffer that's in use for another operation. In such a case, the backing buffer can be replaced to avoid forcing a sync in the command stream which will stall the application's processing. There's some other cases for this as well, like glInvalidateFramebuffer and glDiscardFramebufferEXT, but the primary usage that I'm interested in is buffers.

[...]

Currently, as of today's mainline zink codebase, we have struct zink_resource to represent a resource for either a buffer or an image. One struct zink_resource represents exactly one VkBuffer or VkImage, and there's some passable lifetime tracking that I've written to guarantee that these Vulkan objects persist through the various command buffers that they're associated with.

Each struct zink_resource is, as is the way of Gallium drivers, also a struct pipe_resource, which is tracked by Gallium. Because of this, struct zink_resource objects themselves cannot be invalidated in order to avoid breaking Gallium, and instead only the inner Vulkan objects themselves can be replaced.
Intel Compute-Runtime 20.43.18277 is out this morning as the latest version of the company's open-source graphics compute stack for HD/UHD/Iris/Xe Graphics on Linux with OpenCL and oneAPI Level Zero support.

It was the previous Compute-Runtime release two weeks back that brought OpenCL 3.0 for Broadwell through Ice Lake with Gen12/Tigerlake having already seen CL 3.0 support as a new platform. That OpenCL 3.0 support is in good shape with this latest release and the stack remains at a "pre-release" level for its oneAPI Level Zero 1.0 support.

- **llvmpipe is OpenGL 4.5 conformant**, [4]

(I just sent the below email to mesa3d developer list).

Just to let everyone know, a month ago I submitted the 20.2 llvmpipe driver for OpenGL 4.5 conformance under the SPI/X.org umbrella, and it is now official[1].

Thanks to everyone who helped me drive this forward, and to all the contributors both to llvmpipe and the general Mesa stack that enabled this.

Big shout out to Roland Scheidegger for helping review the mountain of patches I produced in this effort.

My next plans involved submitting lavapipe for Vulkan 1.0, it's at 99% or so CTS, but there are line drawing, sampler accuracy and some snorm blending failure I have to work out.
I also ran the OpenCL 3.0 conformance suite against clover/llvmpipe yesterday and have some vague hopes of driving that to some sort of completion.

(for GL 4.6 only texture anisotropy is really missing, I've got patches for SPIR-V support, in case someone was feeling adventurous).

Dave.

- **LLVMpipe Is Now Officially Conformant With OpenGL 4.5** [5]

Beginning with Mesa 20.2 is OpenGL 4.5 support for LLVMpipe, the LLVM-based software rasterizer built as a Gallium3D driver. This succeeded LLVMpipe for years being limited to OpenGL 3.3. While the OpenGL 4.5 support has been enabled for weeks, The Khronos Group
has now officially confirmed its implementation.

Source URL: http://www.tuxmachines.org/node/143846

Links: