Rusticl Status update 2023

Karol Herbst

October 18, 2023

Supported drivers

Last year:

- Iris
- llvmpipe
- nouveau
- panfrost

New:

- radeonsi
- ~~r600~~
- asahi
- zink

in Progress:

- v3d
- etnaviv

New features and extensions

- *__opencl_c_subgroups*
- *cl_khr_create_command_queue*
- *cl_khr_device_uuid*
- *cl_khr_expect_assume*
- *cl_khr_extended_versioning*
- *cl_khr_image2d_from_buffer*
- *cl_khr_integer_dot_product*
- *cl_khr_pci_bus_info*
- *cl_khr_spirv_no_integer_wrap_decoration*
- *cl_arm_shared_virtual_memory*
- Proper profiling support

Other Changes

- llvmpipe supports function calls
  - All luxmark benchmarks finally run!
- *PIPE_CONTEXT_COMPUTE_ONLY* now used
  - Allows drivers to disable unneeded 3D functionality
  - Allows to use compute queues for long running jobs

Problems which needs solving

- Long running compute jobs
- Memory mapping API implementation is still bad

Work in progress

- *__opencl_c_program_scope_global_variables*
- *cl_khr_gl_sharing*
- *cl_* * *_device_attribute_query*
- Non uniform workgroups
  - Needs sytem values for enqueued and current block.
  - Asahi has all the system values I need. Great for prototyping.
- Shared Virtual Memory support for Iris

How to implement SVM in Gallium?

- Cutting out driver private allocations blows up VM usage
- Driver could return start address of heap instead?
- *clSVMAlloc* uses *mmap* to allocate from start of driver heap
  - Frontend should request heap allocation at the same address
  - Then frontend needs to keep memory in sync, a.k.a. "memory migration"
  - Frontend makes sure it's all page aligned
- With explicit memory placement (*cl_intel_unified_shared_memory*):
  - Use *resource_from_user_memory* if memory should remain on Host
  - Driver maps resource at same host VM address if memory should remain on GPU

Planned work

- *cl_khr_semaphore*
- *cl_khr_external_semaphore*
- *cl_khr_external_memory*
- Conformance with radeonsi (2 bugs)
- Conformance with Zink (around 5 bugs)
- Features DPCPP (SyCL) and chipStar (HIP) need
- Support buffers bigger than 2GB
- Function call support for radeonsi
- Easier system value lowering in gallium
- CI on GPUs
- Performance optimizations

Enabling devices by default

- Function calls Supported
- Passing the CTS
- Long running compute jobs supported
- Prefer Native driver over Zink (via *device_uuid*)

Thanks

- Antino for gl sharing work
- Nora for working on random CL extensions
- And everybody else filing bugs or submitting MRs

Q&A

- Any questions?

Contact

- Fedi: @karolherbst@chaos.social
- IRC: karolherbst@oftc.net
- Discord: karolherbst