arm

Arm, Cambridge, UK

Experimenting with C++ Libraries in OpenCL Kernel Code

Presented by
Ole Marius Strøhm

Ole Marius Strøhm, Anastasia Stulova

IWOCL 2021, 26–29 April 2021

C++ Libraries in C++ for OpenCL

- A lot of C++ features come from the libraries
  - About 70% of the specification pertains to the standard library
- Not every feature of C++ can be supported by C++ for OpenCL
  - Function pointers - possible divergent execution
  - Virtual methods
- C++ does not have address spaces

Exploration of libcxx

- Open-source, easily accessible and well integrated with clang
- Many headers in libcxx require unsupported features
- Disabled errors for known unsupported features and tried compiling headers
  - Forces compilation to continue when unsupported features are encountered
  - Made no changes to libcxx
  - This exposed errors in supported features e.g. address spaces

Issues encountered

- No address spaces in C++
  - Need to be deduced by the compiler
- Unsupported features in a lot of headers
  - Least issues in type traits

Type traits in C++ for OpenCL

- Why choose type traits?
  - Stand-alone header-only library
  - Compile-time features suitable for any execution model
    - Unsupported features disappear at compile-time
  - Heavily templated - Good test of compiler
- Address spaces deduced very well already
  - Only two small bugs had to be fixed
- Unsupported features are still in type traits

Work done in upstream clang

- Fixed bugs in clang
  - D82781 - Template variable addr space deduction
  - D83665 - Copy constructor missing addr space
- Two extensions added:
  - __cl_clang_function_pointers
    - To support: is_member_function_pointer
  - __cl_clang_variadic_functions
    - To support: result_of, invoke_result, is_invocable, is_nothrow_invocable and is_member_function_pointer

Results for type traits

- libcxx contains 147 tests
- 51% of the test were modified
  - Done through a script
  - Does not limit coverage as it only removes unsupported language features

Pass	XFail	Dep on other headers	C++20 features	Not supported
126	3	8	7	3
129		18		

arm

Example using type traits with OpenCL

```
$ clang -cl-std=clc++ -I<path to libcxx>/include -DN=10 test.cl
// Enable compiler extensions for type traits.
#pragma OPENCL EXTENSION __cl_clang_function_pointers : enable
#pragma OPENCL EXTENSION __cl_clang_variadic_functions : enable
#include <type_traits>
#pragma OPENCL EXTENSION __cl_clang_function_pointers : disable
#pragma OPENCL EXTENSION __cl_clang_variadic_functions : disable
// Find a minimal value in a given non-empty sequence.
template<typename arr_ty,
         // Use C++ type trait functionality in OpenCL kernel code.
         int arr_size = std::extent<arr_ty>::value,
         typename elem_ty = typename std::remove_extent<arr_ty>::type>
auto find_min(arr_ty arr) {
    elem_ty res = arr[0];
    for (auto i = 1U; i < arr_size; i++)
        res = min(res, arr[i]);
    return res;
}
// Example that uses find_min in a kernel with an array of int4.
__kernel void compute(__global int4* input, __global int4* output)
{
    int4 seq[N];
    do_some_work(input, seq);
    output[get_global_id(0)] = find_min<decltype(seq)>(seq);
}
```

Conclusion

- Our evaluation showed feasibility of porting libcxx implementation to OpenCL
- Enabled type traits for users of C++ for OpenCL to experiment in offline compilation starting from clang release 12

Future Work

- Add OpenCL specific functionality for type traits
  - add/remove address spaces, detect vector types, etc.
- Evaluate support for C++20 and other libraries
  - iterator, algorithms, etc.
- Productize C++ libraries for OpenCL via libcxx or libclcxx

Resources

- https://www.iwocl.org/wp-content/uploads/iwocl-syclcon-2020-stulova-13-slides.pdf
- https://www.stroustrup.com/bs_faq.html#big
- https://libcxx.llvm.org
- https://clang.llvm.org/docs/OpenCLSupport.html#opencl-experimental-cxxlibs
- https://github.com/KhronosGroup/libclcxx
- https://godbolt.org/z/5WbnTfb65

arm

Thanks!

Presented by
Ole Marius Strøhm