Argonne NATIONAL LABORATORY

SYCL: Quick-start

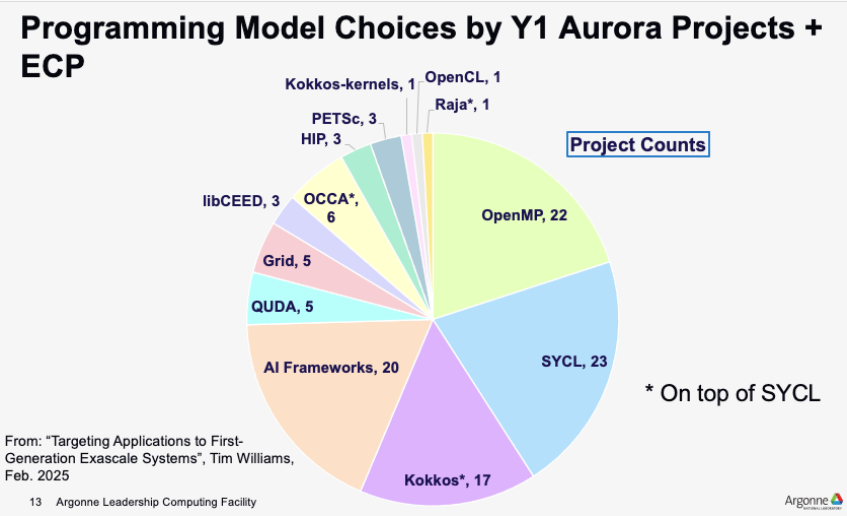
Thomas Applencourt

May 7, 2025

Table of Contents

Overview of SYCL

Is this thing really used?



Code Example

```
1  #include <sycl/sycl.hpp>
2
3  int main(int argc, char **argv) {
4    sycl::queue Q;
5    std::cout << "Running on "
6              << Q.get_device().get_info<sycl::info::device::name>() << "\n";
7
8    // Allocate Device Memory (Nooo raw pointer :( )
9    int *A = sycl::malloc_device<int>(global_range, Q);
10   // Submit blocking kernel who use the memory
11   Q.parallel_for(global_range, [=](auto id) { A[id] = id; });
12   Q.wait();
13   // Allocate Host Memory
14   std::vector<int> A_host(global_range);
15   // Copy the device memory to the host
16   Q.copy(A, A_host.data(), global_range).wait();
17   sycl::free(A, Q);
18 }
```

More in `https://github.com/argonne-lcf/sycltrain/tree/master/9_sycl_of_hell`

More than a API

Of course, API are only the tip of the iceberg

- Have access to math function (sin,cos,sqrt)
- Have access to intrinsic (shuffle, popcount, swizzle)
- MKL Library, and lot of other
- Tool to port cuda to SYCL...

How to Compile and Run

How to Compile?

```
$ icpx -fsycl foo.cpp #JIT -> spirv
$ icpx -fsycl-targets=spir64_gen foo.cpp  # AOT -> genISA
```

How To Run?

Use the default queue

```
1 sycl::queue Q;
```

Then Use `gpu_tile_compact`. Each rank will see one tile
`mpirun gpu_tile_compact ./a.out`. All's right with the world

To be extra fancy

No affinity, you need to understand visibility, hierarchy mode, subdevice, and context.

```
1  int world_rank;
2  MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
3  sycl:: device D = sycl::get_devices()[world_rank];
4  sycl::queue Q(D, sycl::context{D});
```

```
mpirun ./a.out
```

Conclusion

One slide

- SYCL is ”just a portable wrapper C++” on top of lower API / C kernel flavor (CUDA, OpenCL, L0, Hip)
- Can use fancy allocator, mdspan, just ping me if interested (example in the repo posted previously)
- Use `"-fsycl"`
- Don’t hesitate to talk to me or Abhi for any question![1]
- Please, if you hate SYCL/ if SYCL is missing stuff, just open an issue on the specification `https://github.com/KhronosGroup/SYCL-Docs/issues` [2]

[1] We can rant about C++, other framework together, python and how we should all go back to Fortran. 77.
[2] The issue will be most likely ignored, but at least you can sleep peacefully knowing you did your part