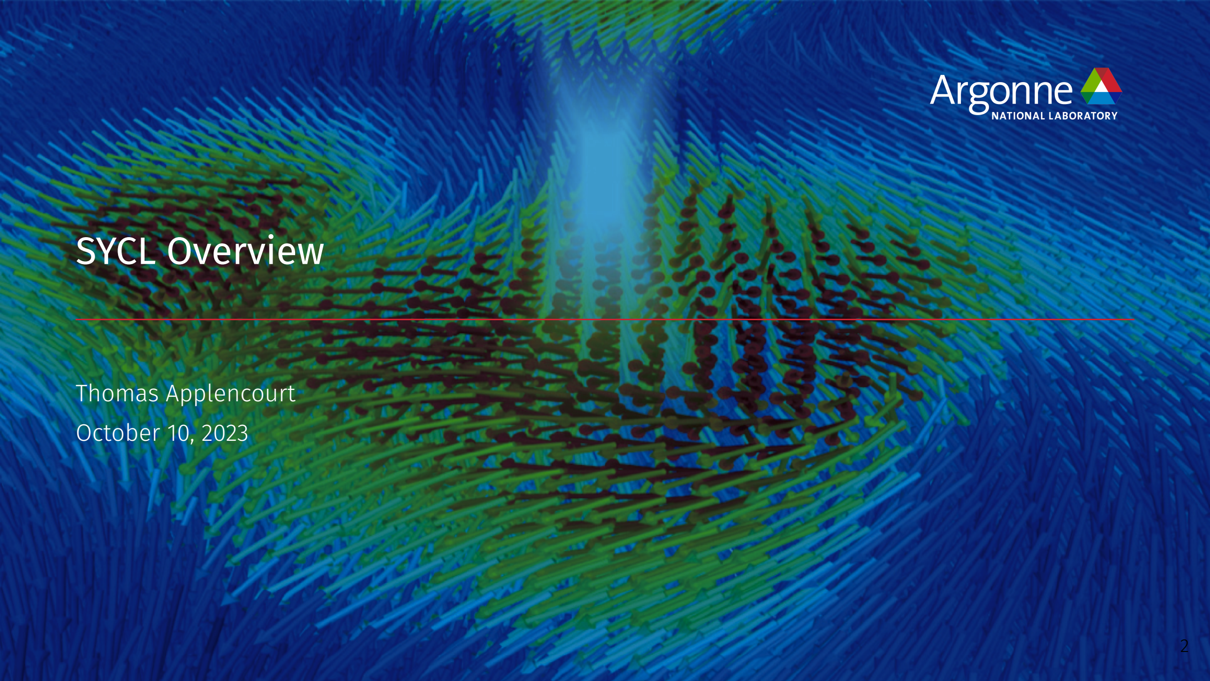
October 10-12, 2023

ALCF Hands-on HPC Workshop

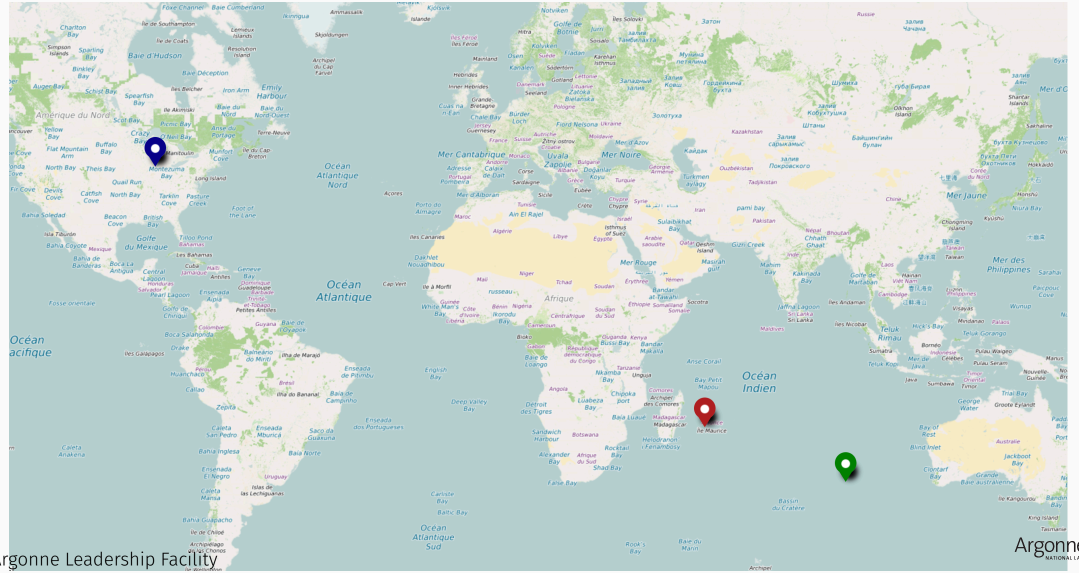


SYCL Overview

Thomas Applencourt

October 10, 2023

Who I am?



Who I am?

- Member of the Performance Engineering Team
- ”Specialist” on Runtime[1]
- Argonne Representative to the SYCL committee

[1]At least I’m interested by them

What is SYCL?

- SYCL is a specification developed by the Khronos Group (OpenCL, SPIR, Vulkan, OpenGL)
- C++ 17 API
  - No language extension, No pragmas, No attribute
  - Lot of lambda, lot of template
- Borrow lot of *concept* from battle tested OpenCL (platform, device, work-group, range)
- Single Source
- Portable (HIP, Cuda Driver, L0 Backend exists)

Hello World

```cpp
#include <sycl/sycl.hpp>
int main() {
  // Create a "stream" aka object used to submit work to the device
  sycl::queue Q;
  // Some instrospection!
  std::cout << "Running on "
            << Q.get_device().get_info<sycl::info::device::name>()
            << std::endl;
  const int size = 10;
  // Allocate memory who can be access from the host and the device
  int *A = sycl::malloc_shared<int>(size,Q);
  // Submit a kernel to the GPU, lambda! <3
  Q.parallel_for(global_range, [=](sycl::id<1> idx) { A[idx] = idx; }).wait();
  for (size_t i = 0; i < size; i++)
    std::cout << "A[ " << i << " ] = " << A[i] << std::endl;
  return 0;
}
```

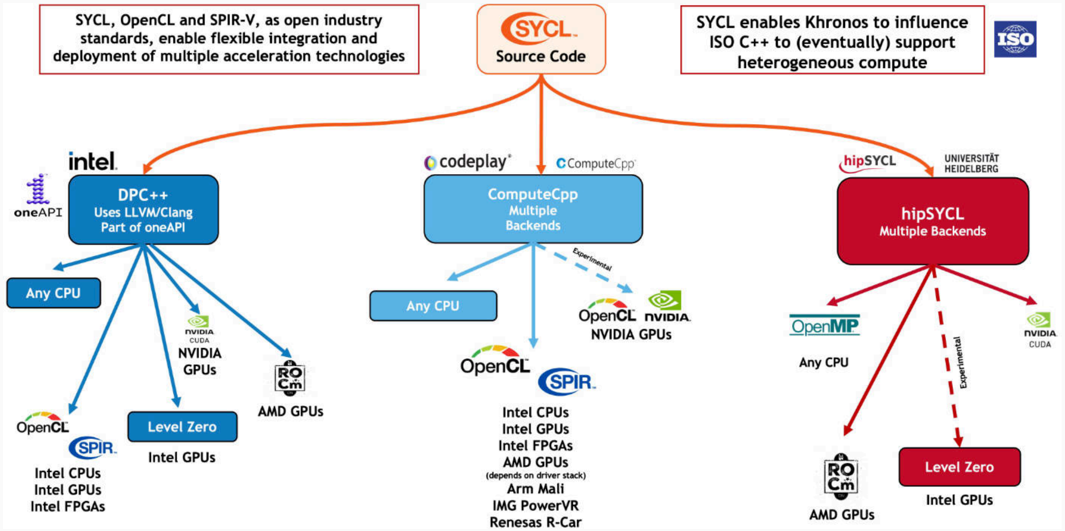
Beyond Hello World

- Reduction, Atomic, Linear Algebra (via oneMKL)

```
sycl:: Q;
[...] // Allocated a left an an exercise to the reader
oneapi::mkl::blas::gemm(Q, oneapi::mkl::transpose::nontrans,
                           oneapi::mkl::transpose::nontrans,
                           m, n, k, alpha, A, ldA, B, ldB, beta, C, ldC).wait();
```

- Backend Interopt! (i.e can get the cuda::stream associated with a sycl::queue, or create a sycl::queue from a cuda::stream)
- Can mix OpenMP and SYCL

Implementer of SYCL



Porting your code for CUDA

- Easy peasy: *`https://github.com/oneapi-src/SYCLomatic`*
- Can translate full projects (Abhi is our home expert)

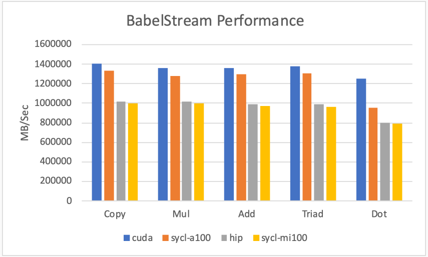
Performance

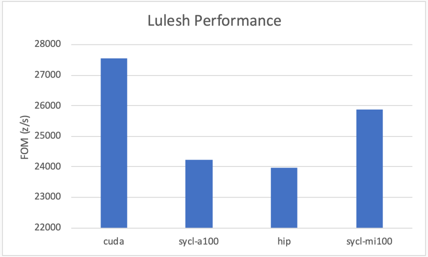
Controversial Take:

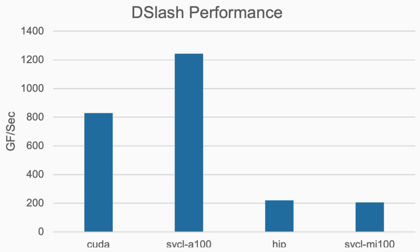
- SYCL and CUDA Runtime sit at the same level in the ladder abstraction: No fundamental reason performance should be different[2]
- Code gen is less important than a good runtime

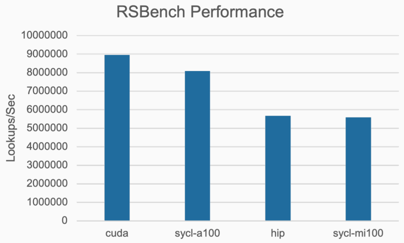
[2]Minus event creation, batching...

Data (we will add yours at the of the day)









Pro and Cons of SYCL

Pro

- Close to C++ (familiarity to C++ programmer)
- Backed by Industry (a lot of man power)
- Portable (deployed at ALCF, NERSC and other)
- Simple
- Performant (please repport bugs if not)

Cons

- Close to C++ (lambda are intelligible and errors 1000 lines long)
- Backed by Industry (will they abandon it?)

What you will learn?

- All of SYCL, aka enough to get your started!
- Know enough to know if SYCL if a good fit for your project
- *https://docs.alcf.anl.gov/polaris/programming-models/sycl-polaris/*
- */eagle/projects/fallwkshp23/SYCL*