STANDARDS IN HPC

Reloaded

TABLE OF CONTENTS

STREAM
High Performance Computing

OPENCL ON THE MOVE

Constantly improving

NOT L'ART POUR L'ART

IMPROVING BIT BY BIT

STREAM
High Performance Computing

- OpenCL SDK serves as a „one-stop shop” for devs
  - Initial batch of native samples & utilities awarded to Stream
- The Khronos OpenCL Work-Group continues to improve this SDK
  - Second round of major improvements also awarded to Stream 
  - Expectation is to land improvements throughout the summer
- Quarterly spec updates
  - Clarification, bug fixes
  - New features

OPENCL ON THE MOVE

NEW EXTENSIONS

STREAM
High Performance Computing

- cl_khr_semaphore
  - A new mutable, reusable sync primitive
- cl_khr_external_semaphore
  - Import/export semaphore sync primitives of/with other APIs
- cl_khr_external_memory
  - Import/export buffers and images of/with other APIs
- cl_khr_command_buffer
  - Record a series of commands for faster replay
  - cl_khr_command_buffer_mutable_dispatch
  - cl_khr_command_buffer_multi_device

OPENCL ON THE MOVE

ADVISORY PANEL

STREAM
High Performance Computing

- Group of OpenCL experts from both industry and academia
- Participation is free of charge
- Members get access to
  - Working drafts of the spec
  - Internal discussion materials
  - Direct channel of communication with the WG
- Infrequent panel meetings

- If interested, reach out to the AP liaison
  - mate[[@]]streamhpc.com

OPENCL ON THE MOVE

CHIP-SPV

- Project of Tampere University, Argonne National Laboratory et al.
- HIP implementation running on SPIRV-enabled OpenCL runtimes
- Project available on GitHub

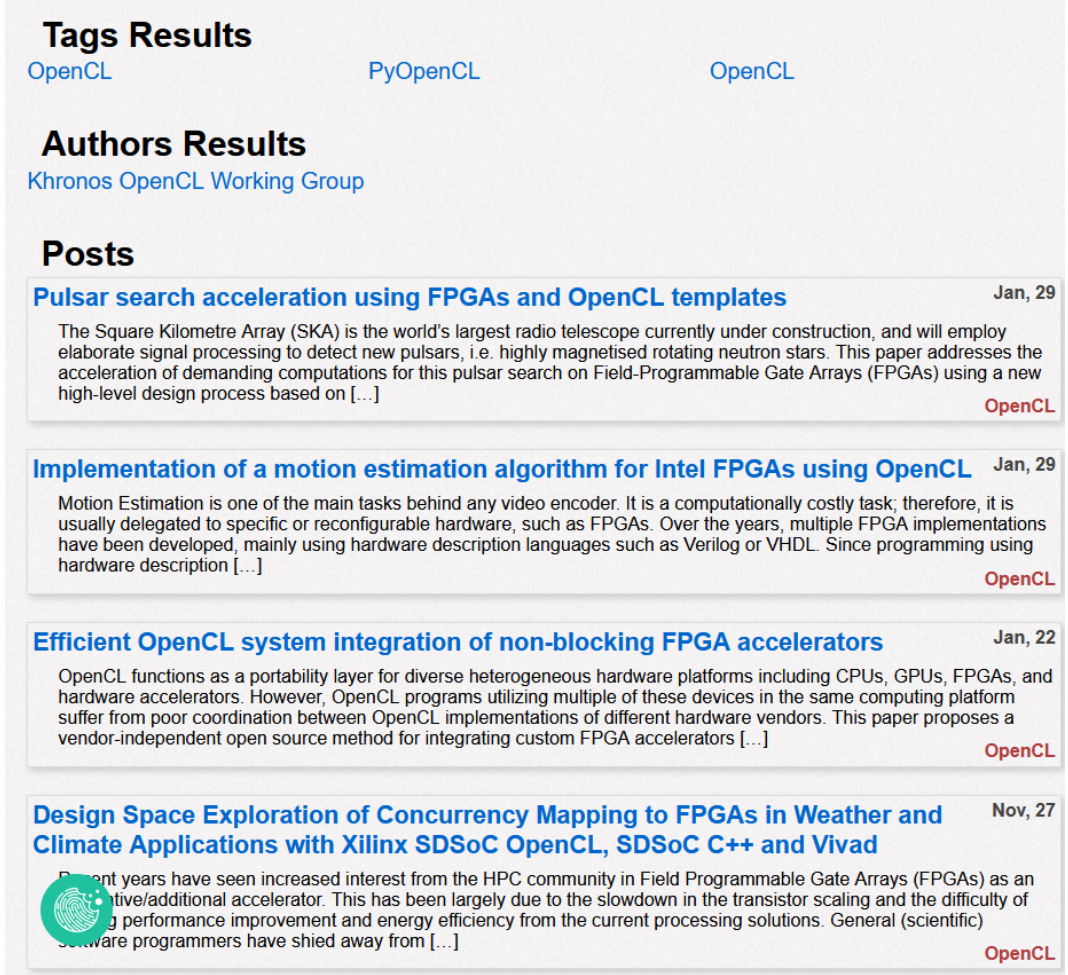
SYCL ON THE MOVE

Rapid progress

THE COOL KID IN TOWN

STREAM
High Performance Computing

- IWOCL 2023 conference program
  - Ctrl+F, SYCL, 92-hits
  - Not 92 talks, but the vast majority
- OpenCL comes in as a distant second
- Interpretation?
  - OpenCL is far more fleshed out, needs less foundational work
  - Discrepancy not visible in cuurrent research



Tags Results

OpenCL PyOpenCL OpenCL

Authors Results

Khronos OpenCL Working Group

Posts

Pulsar search acceleration using FPGAs and OpenCL templates Jan, 29

The Square Kilometre Array (SKA) is the world's largest radio telescope currently under construction, and will employ elaborate signal processing to detect new pulsars, i.e. highly magnetised rotating neutron stars. This paper addresses the acceleration of demanding computations for this pulsar search on Field-Programmable Gate Arrays (FPGAs) using a new high-level design process based on [...]

OpenCL

Implementation of a motion estimation algorithm for Intel FPGAs using OpenCL Jan, 29

Motion Estimation is one of the main tasks behind any video encoder. It is a computationally costly task; therefore, it is usually delegated to specific or reconfigurable hardware, such as FPGAs. Over the years, multiple FPGA implementations have been developed, mainly using hardware description languages such as Verilog or VHDL. Since programming using hardware description [...]

OpenCL

Efficient OpenCL system integration of non-blocking FPGA accelerators Jan, 22

OpenCL functions as a portability layer for diverse heterogeneous hardware platforms including CPUs, GPUs, FPGAs, and hardware accelerators. However, OpenCL programs utilizing multiple of these devices in the same computing platform suffer from poor coordination between OpenCL implementations of different hardware vendors. This paper proposes a vendor-independent open source method for integrating custom FPGA accelerators [...]

OpenCL

Design Space Exploration of Concurrency Mapping to FPGAs in Weather and Climate Applications with Xilinx SDSoC OpenCL, SDSoC C++ and Vivad Nov, 27

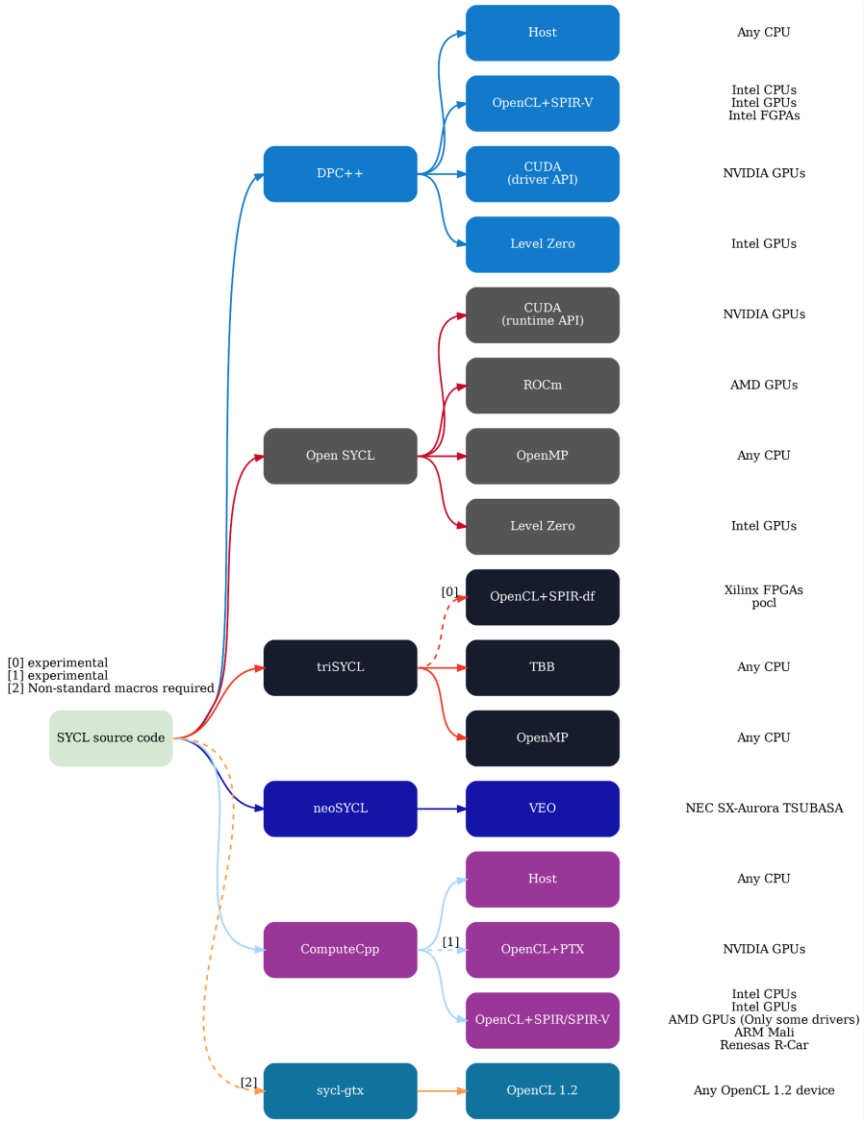
nt years have seen increased interest from the HPC community in Field Programmable Gate Arrays (FPGAs) as an tive/additional accelerator. This has been largely due to the slowdown in the transistor scaling and the difficulty of g performance improvement and energy efficiency from the current processing solutions. General (scientific) ware programmers have shied away from [...]

OpenCL

https://hgpu.org/?s=opencl

SYCL ON THE MOVE

- Adoption of SYCL 1.2.1 was held back by the adoption of SPIR-V across the OpenCL ecosystem
- SYCL 2020 introduces
  - Non-OpenCL back-ends via generic interop system
  - Universal Shared Memory to capture host-side memory
  - Improved buffer, accessor, host-task interfaces
  - Built-in device-side algorithms
  - Minimally C++17 conformant



https://github.com/OpenSYCL/OpenSYCL/

SYCL ON THE MOVE

UNIFIED SHARED MEMORY

Table 100. Characteristics of the different kinds of USM allocation

Allocation Type	Initial Location	Accessible By		Migratable To	
`device`	`device`	`host`	No	`host`	No
		`device`	Yes	`device`	N/A
		Another `device`	Optional (P2P)	Another `device`	No
`host`	`host`	`host`	Yes	`host`	N/A
		Any `device`	Yes	`device`	No
`shared`	Unspecified	`host`	Yes	`host`	Yes
		`device`	Yes	`device`	Yes
		Another `device`	Optional	Another `device`	Optional

https://registry.khronos.org/SYCL/specs/sycl-2020/html/sycl-2020.html#sec:usm

- UVA, SVM, USM… all the same
  - Or are they?

SYCL ON THE MOVE

UNIFIED SHARED MEMORY

Table 100. Characteristics of the different kinds of USM allocation

Allocation Type	Initial Location	Accessible By		Migratable To	
device	device	host	No	host	No
		device	Yes	device	N/A
		Another device	Optional (P2P)	Another device	No
host	host	host	Yes	host	N/A
		Any device	Yes	device	No
shared	Unspecified	host	Yes	host	Yes
		device	Yes	device	Yes
		Another device	Optional	Another device	Optional

cu/hipMalloc()

Pinned memory

cu/hipManagedMalloc()

SVMalloc()

- UVA, SVM, USM… all the same
  - Or are they?
- These allocator types unify existing features sets

SYCL ON THE MOVE

https://registry.khronos.org/SYCL/specs/sycl-2020/html/sycl-2020.html#sec:usm

SYCL INTERACTIONS

- GROMACS development team prefers open standards
- AMD prefers HIP back-end
  - Rule out 3rd party defects
- AMD benefits from the SYCL back-end
  - Reduce maintenance cost of HIP back-end
  - Can draw inspiration from for optimization ideas
- Stream HPC improving the HIP back-end
  - Use hipSYCL macros for hip source interoperability
  - Add VkFFT to hipSYCL HIP backend
  - Add CDNA II optimized float3 implementation
- GROMACS improving SYCL back-end based on HIP
  - SYCL listed forces: optimize parameter passing

SYCL ON THE MOVE

SYCL INTERACTIONS

STREAM
High Performance Computing

GROMACS > GROMACS > Merge requests > !3496

SYCL listed forces: optimize parameter passing

Code

Merged **Andrey Alekseenko** requested to merge `aa-gfx90a-do-weird-indexing` into `release-2023` 3 months ago

Overview 12 | Commits 5 | Pipelines 0 | Changes 2 | All threads resolved!

Compared to 2023.0, using ADH cubic. Speed-up of "Bonded" kernels:

	F-only	FV
hipSYCL 0.9.4, ROCm 5.2.5, bundled Clang, gfx90a	2.67	2.33
hipSYCL 0.9.4, ROCm 5.4.1, Clang 15.0.7, gfx1034	1.94	1.56
hipSYCL 0.9.4, CUDA 11.8, Clang 15.0.7, sm_86	1.00	1.00
IntelLLVM nightly 2023-02-06, CUDA 11.8, sm_86	4.09	1.55
oneAPI 2023.0, Intel Arc 770	1.29	1.01
oneAPI 2023.0, Ponte Vecchio	1.10	~1

Based on AMD/StreamHPC optimization.

Refs #3928 (closed), #4593

https://gitlab.com/gromacs/gromacs/-/merge_requests/3496

SYCL ON THE MOVE

OPENCL & SYCL

What is the fundamental difference?

WHAT IS THE BIG DIFFERENCE?

Host side

Device side

OPENCL & SYCL

WHAT IS THE BIG DIFFERENCE?

Host side

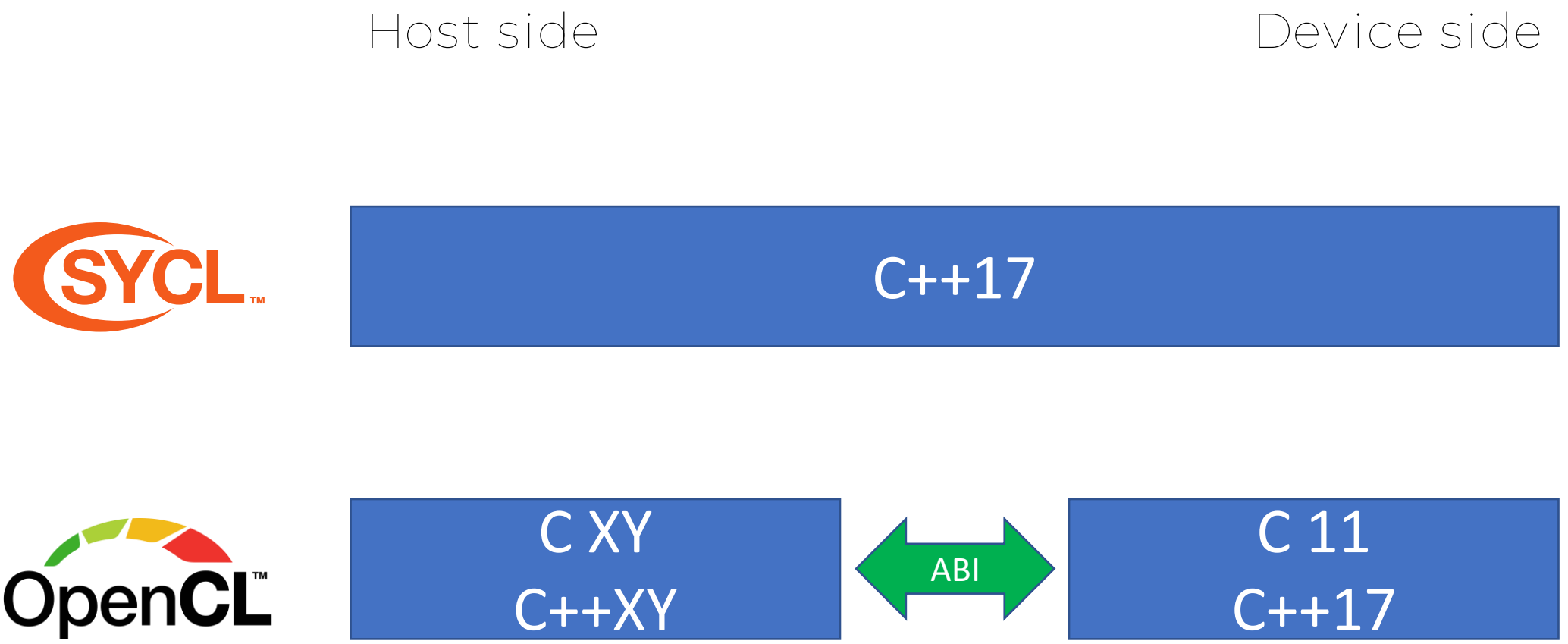
Device side

C++17

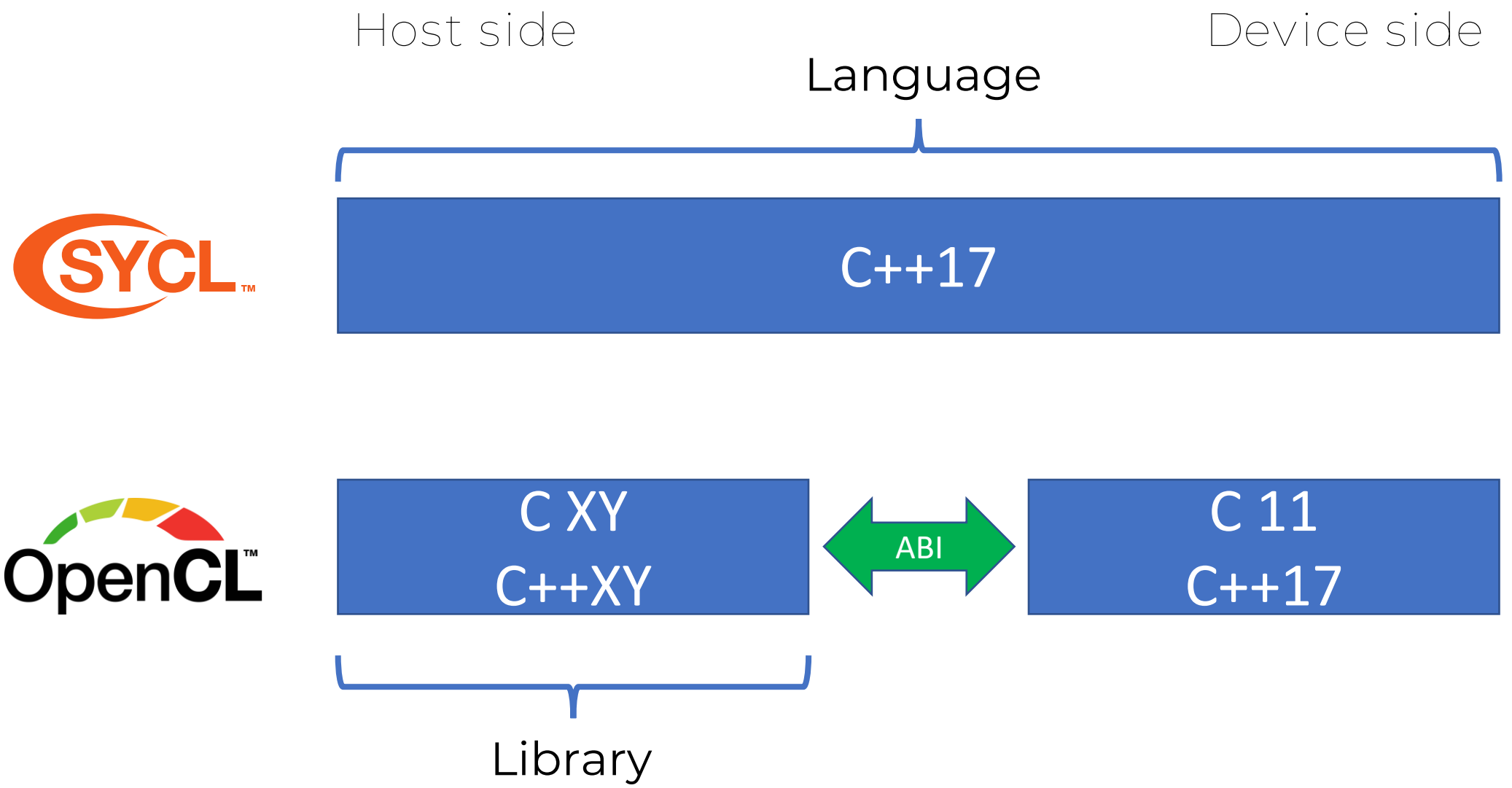
WHAT IS THE BIG DIFFERENCE?

	Host side	Device side
SYCL	C++17	
OpenCL	C XY C++XY	C 11 C++17

WHAT IS THE BIG DIFFERENCE?



WHAT IS THE BIG DIFFERENCE?



WE'RE HIRING

https://streamhpc.com/jobs/

THANK YOU FOR YOUR ATTENTION