SYCL for CUDA: An Overview of Implementing PI for CUDA

Alexander Johnston - Senior Software Engineer

LLVM 2020
SYCL in a nutshell

• SYCL is a Khronos open standard interface that enables parallel processor architectures
• Standard C++
• Single Source
• Requires a Runtime Library and Device Compiler
• More at https://sycl.tech
Summary

• Driver Modifications
• PI library plugin mechanism
• PI CUDA backend
• Performance
Driver Modifications

- Modified the Device Compilation stage to enable the driver to perform device compilation for NVPTX
- This requires passing the generated NVPTX through NVIDIA SDK tools during the Device Compilation step
PI Plugin Mechanism

- PI library provides a plugin mechanism for backend libraries
- Each backend must link up to a dispatch table of PI library functions
- An instance of a backend is loaded at runtime when requested
PI CUDA Backend

- Implemented 95 PI library functions with native CUDA calls
- Provided opaque containers to the PI library with CUDA specific information inside
- Implemented all SYCL builtins in libclc as native CUDA calls
Performance

BabelStream FP32 MB/s

- SYCL for CUDA
- CUDA
- OpenCL for CUDA

Copy, Mul, Add, Triad, Dot
Further Resources

• Get the repo at https://github.com/intel/llvm/


• Code examples at https://github.com/codeplaysoftware/SYCL-For-CUDA-Examples
Thank you!

alexander@codeplay.com