C++ for Heterogeneous Programming: oneAPI (DPC++ and oneTBB)

2020 International Conference for High Performance Computing, Networking, Storage and Analysis ("SC20")

Full-day tutorial

James Reinders, Michael J. Voss, Pablo Reble and Rafael Asenjo

Abstract

This tutorial provides hands-on experience programming CPUs, GPUs and FPGAs using a unified, standards-based programming model: oneAPI. oneAPI includes a cross-architecture language: Data Parallel C++ (DPC++). DPC++ is an evolution of C++ that incorporates the SYCL language with extensions for Unified Shared Memory (USM), ordered queues and reductions, among other features. oneAPI also includes libraries for API-based programming, such as domain-specific libraries, math kernel libraries and Threading Building Blocks (TBB). The main benefit of using oneAPI over other heterogeneous programming models is the single programming language approach, which enables one to target multiple devices using the same programming model, and therefore to have a cleaner, portable, and more readable code.

In the current heterogeneous era, it is still challenging for developers to match computations to accelerators and to coordinate the use of those accelerators in the context of their larger applications. Therefore, this tutorial's main goal is not just teaching oneAPI as an easier approach to target heterogeneous platforms, but also to convey techniques to map applications to heterogeneous hardware paying attention to the scheduling and mapping problems (how to achieve load balance and which regions of the application are more suitable to each particular device).

1. Detailed Description

Tutorial Goals

By the end of the tutorial, attendees will be familiar with the important architectural features of commonly available compute devices (CPUs, GPUs and FPGAs) and will have a sense of what optimizations and types of parallelism are suitable for these devices. Attendees will also be introduced to oneAPI, including DPC++ and oneTBB, learn about its heterogeneous programming features, and will build and execute a heterogeneous application. Attendees will take part in hands-on exercises to create a small example, port it to oneAPI and evolve it from a host-only shared-memory implementation to a heterogeneous implementation that runs on both the host CPUs and accelerators (GPU and FPGA).

Relevance

Heterogeneous platforms are becoming increasingly common in HPC programming, with compute resources that include a diverse collection of integrated and discrete graphics processors, FPGAs and other domain-specific compute engines. Understanding the tradeoffs in using these accelerators and how to select and optimize computations for offload to these devices is an important and timely topic.

The goal of oneAPI is to augment C++ to create a model that covers all of these devices, without sacrificing performance. C++ continues to grow in importance in HPC programming and the combination of oneAPI's DPC++ and oneAPI's Threading Building Blocks (oneTBB) provides a powerful combination for expressing heterogeneous applications in C++.

Target Audience

Programmers in the field of High Performance Computing that want to better understand heterogeneity and to develop portable C++ applications that unleash the power of multi-core, many-core as well as heterogeneous systems.

Content level

50% beginner:	A survey of heterogeneous architectures and programming models. Data level parallelism, Task based parallelism.
40% intermediate:	Data flow and graph parallelism. Heterogeneous features in oneAPI and oneTBB.
10% advanced:	Developing heterogeneous scheduling and load balancing algorithms.

Prerequisites

Attendees should have an understanding of basic parallel programming concepts such as threads and locks. Attendees should be comfortable with programming in C++. Advanced C++ features such as lambda expressions will be briefly introduced before they are used in the tutorial. No previous experience with oneAPI, DPC++, SYCL, Threading Building Blocks, GPUs or FPGAs is required.

Content

This full day tutorial starts with a survey of heterogeneous architectures and programming models, and discusses how to determine if a computation is suitable for a particular accelerator. Next, oneAPI is presented as a unified, standards-based programming model that includes Data Parallel C++ (DPC++) for the direct programming of devices, libraries for API-based programming (including oneTBB), and advanced analysis and debugging tools. DPC++ is leveraged to exploit data parallel, kernel oriented, programming. Two alternatives, buffers and USM (Unified Shared Memory), to share data between the host and the accelerator are introduced. oneTBB is a part of oneAPI and is a widely used, portable C++ template library for parallel programming on the host. TBB's task orientation and work-stealing load balancing make it an excellent library to orchestrate and schedule computations among the different devices. The tutorial will present recent results and experimental validation of the suitability of oneAPI and oneTBB to make the most out of CPU+Accelerator platforms with less programming effort. Finally, we will discuss some proposed heterogeneous schedulers built on top of oneAPI+oneTBB. These heterogeneous implementations of the parallel_for template automatically distribute the workload between the multicore and the accelerator (GPU or FPGA). We compare performance and programmability metrics of OpenCL+TBB implementations with the oneAPI+oneTBB counterparts.

Hands-on exercises will be interleaved with the theoretical content from the very beginning. After a brief introduction, attendees will be walked through the process of opening a DevCloud account (free) and compiling one example for CPU-only, GPU-only and FPGA codes. The FPGA compilation will take some time but it should be ready to execute in the afternoon. Emulation mode for the FPGA will also be used for faster compilation. The details of the different implementations and easy exercises will be covered as soon as the required information is covered with slides and examples.

Collaboration

James Reinders is leading the effort, with the Intel DPC++ development team, to write a oneAPI book "Data Parallel C++" that will be available by SC'2020 (open access). Michael Voss, Rafael Asenjo and James Reinders are co-authors of the latest book on TBB: "Pro TBB", Apress 2019 (open access too). Michael Voss and Pablo Reble are members of the engineering team that develops Threading Building Blocks and oneAPI's Data Parallel Library. Both James Reinders and Rafael Asenjo are long-time, and continuing collaborators of the parallel software teams at Intel. The content for this tutorial is being developed collaboratively to create a unified flow and message and will leverage existing content from previous collaborations such as our SC'17, Euro-Par'17, PPoPP'17 and '18 tutorials.

Previous Presentations

Although it is the first time we propose this oneAPI tutorial, James Reinders has recorded several webinars covering oneAPI, and we have previously delivered four TBB oriented tutorials that we list below:

- "An Introduction to Intel® Threading Building Blocks (Intel® TBB) and its Support for Heterogeneous Programming," Rafael Asenjo, Jim Cownie and Aleksei Fedotov, a tutorial at PPoPP'18, Feb 2018, Viena, Austria. https://ppopp18.sigplan.org/track/PPoPP-2018-Tutorials
- "Expressing Heterogeneous Parallelism in C++ with Intel Threading Building Blocks," Michael Voss, James Reinders, Pablo Reble and Rafael Asenjo, a tutorial at SC17, November 2017, Denver, CO, USA. Material available at: https://github.com/oneapi-src/oneTBB/tree/tutorials_sc17/doc/sc17_slides https://github.com/oneapi-src/oneTBB/tree/tutorials_sc17/examples/sc17_hetero
- "CPUs, GPUs, FPGAs: A Tutorial on Heterogeneity and Managing Accelerators with Intel Threading Building Blocks," Michael Voss, Pablo Reble and Rafael Asenjo, a tutorial at Euro-Par 2017.
- "CPUs, GPUs, FPGAs: Managing the alphabet soup with Intel Threading Building Blocks," Michael Voss, Pablo Reble and Jackson Marusarz, a tutorial at the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP) 2017, February 4, 2017, Austin, TX, USA. http://ppopp17.sigplan.org/event/ppopp-2017-tutorials-cpus-gpus-fpgas-managing-the-alphabet-soup-with-intel-threading-building-blocks

2. Tutorial Outline

Part 1: Motivation and background (morning 1st half)

- HW: an introduction to heterogeneous architectures (GPUs and FPGAs)
- SW: oneAPI introduction
  - "Hello oneAPI" on DevCloud (account setup and first example)

Part 2: oneAPI and DPC++: kernel-based approach (morning 2nd half)

- Data parallelism exploited with SYCL and DPC++
- Single source programing model with DPC++
- Data management: Buffers and USM (Unified Shared Memory)
- Hands-On exercises:
  - Offloading computation to the GPU

Part 3: oneTBB: task-based orchestration (afternoon 1st half)

- Introduction to oneTBB: Threading Building Blocks
- Flow graph and its heterogeneous features
- Using async_node to do asynchronous communication
- Hands-On Exercises
  - Using task_group to statically distribute work to CPU and GPU
  - Using async_node to dynamically distribute work to CPU and GPU

Part 4: Putting it all-together: oneAPI+oneTBB (afternoon 2nd half)

- Heterogeneous scheduling, goals and challenges
- Overview of experimental evaluations in terms of performance and programmability
- Hands-On Exercises
  - Targeting a CPU+FPGA on DevCloud

3. Hands-on Part

Intel DevCloud is a development sandbox with access to the latest CPU, GPU and FPGA hardware from Intel. Intel oneAPI software is preinstalled and will be used for the Hands-on part of our Tutorial. Instructions on how to create a free account and setup console access from participants laptops will be provided.

The general outline for the hands-on exercises will be:

- "Hello oneAPI" on DevCloud (account setup and first example)
- Offloading computation to the GPU using Buffers and USM
- Using task_group to statically distribute work to CPU and GPU
- Using async_node to dynamically distribute work to CPU and GPU
- Targeting a CPU+FPGA on DevCloud

Step-by-step instructions will be provided to attendees and walked through the different steps by the instructors.