

Celerity: How (Well) Does the SYCL API Translate to Distributed Clusters?

Philip Salzmann
philip.salzmann@uibk.ac.at
University of Innsbruck
Austria

Peter Thoman
peter.thoman@uibk.ac.at
University of Innsbruck
Austria

Fabian Knorr
fabian.knorr@uibk.ac.at
University of Innsbruck
Austria

Biagio Cosenza
bcosenza@unisa.it
University of Salerno
Italy

CCS CONCEPTS

• **Software and its engineering** → *Distributed programming languages*; **Runtime environments**; • **Computing methodologies** → **Parallel programming languages**.

KEYWORDS

distributed memory clusters, cluster computing, MPI, SYCL

ACM Reference Format:

Philip Salzmann, Fabian Knorr, Peter Thoman, and Biagio Cosenza. 2022. Celerity: How (Well) Does the SYCL API Translate to Distributed Clusters?. In *International Workshop on OpenCL (IWOCCL'22), May 10–12, 2022, Bristol, United Kingdom, United Kingdom*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3529538.3530004>

1 INTRODUCTION

As the SYCL[4] ecosystem matures, adoption across both research and industry projects steadily increases[8]. By offering a modern, vendor-agnostic way of programming a wide array of accelerator hardware, SYCL has the potential to become an important player in high-performance computing (HPC) as well. In fact, existing pre-exascale and upcoming exascale machines already officially support SYCL or even recommend it as one of their preferred programming models[3][1]. As such, the question of how well the SYCL programming model translates to distributed computing becomes more prevalent. While traditional approaches such as combining SYCL with the Message Passing Interface (MPI)[2] will undoubtedly remain relevant for years to come, a more forward-thinking approach may be to try and extend SYCL's ease of use for single node systems to a distributed cluster. The first project to explore this in greater detail is Celerity[5], a distributed runtime system and API that heavily leans on SYCL in both its API design as well as its underlying execution engine. The validity of its design is currently being evaluated through the porting of two industry use cases ([6],[7]) for large scale distributed execution as part of the LIG-ATE project. While Celerity is neither a true subset nor superset of

the SYCL API, experienced SYCL users will immediately recognize the familiar structure of its API.

In this talk, we will review the SYCL API from the perspective of Celerity and distributed memory programming in general. We will highlight challenges encountered and opportunities for future improvement of the SYCL API.

2 OVERVIEW OF CELERITY

We will begin our presentation by giving an overview of the Celerity programming model, highlighting its similarities to SYCL and introducing core additions to the API. We will showcase how a typical Celerity program is structured, and how an existing SYCL application can be converted to Celerity. Additionally, we will give a brief overview of how Celerity itself uses SYCL internally to power its distributed execution semantics.

3 SYCL IN DISTRIBUTED ENVIRONMENTS

The main portion of this presentation will concern itself with investigating important features of SYCL and how well they translate to distributed clusters. We will begin by examining core features such as the high-level data-driven APIs of queues, buffers, command groups and accessors in a distributed context. Next, we will highlight newer additions to SYCL such as host tasks and reductions. Finally, we will take a look at APIs that may be considered problematic from a distributed memory perspective, such as unified shared memory (USM).

4 OUTLOOK ON THE FUTURE OF SYCL

We will conclude our presentation with an outlook on what future versions of SYCL could bring to the table to further improve compatibility with distributed memory clusters. We will review HPC use cases that may not yet be fully covered by SYCL and present several potential improvements that would enhance the experience for both us as library developers as well as users of the traditional MPI + SYCL approach.

ACKNOWLEDGMENTS

This project has received funding from the European High Performance Computing Joint Undertaking (JU) under grant agreement No 956137 as well from the Austrian Research Promotion Agency under grant agreement No 879201.

IWOCCL'22, May 10–12, 2022, Bristol, United Kingdom, United Kingdom

© 2022 Copyright held by the owner/author(s).

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *International Workshop on OpenCL (IWOCCL'22), May 10–12, 2022, Bristol, United Kingdom, United Kingdom*, <https://doi.org/10.1145/3529538.3530004>.

REFERENCES

- [1] ANL. 2021. *SYCL and DPC++ for Aurora*. <https://www.alcf.anl.gov/support-center/aurora/sycl-and-dpc-aurora>
- [2] Message Passing Interface Forum. 2021. *MPI: A Message-Passing Interface Standard, Version 4.0*. <https://www.mpi-forum.org/docs/mpi-4.0/mpi40-report.pdf>
- [3] NERSC. 2021. *Transitioning Applications to Perlmutter*. <https://docs.nersc.gov/performance/readiness>
- [4] The Khronos SYCL Working Group. 2021. *SYCL 2020 Specification (revision 4)*. <https://www.khronos.org/registry/SYCL/specs/sycl-2020/pdf/sycl-2020.pdf>
- [5] Peter Thoman, Philip Salzmänn, Biagio Cosenza, and Thomas Fahringer. 2019. Celerity: High-Level C++ for Accelerator Clusters. In *Euro-Par 2019: Parallel Processing*. Vol. 11725. Springer International Publishing, 291–303. https://doi.org/10.1007/978-3-030-29400-7_21 Series Title: Lecture Notes in Computer Science.
- [6] Peter Thoman, Markus Wippler, Robert Hranitzky, Philipp Gschwandtner, and Thomas Fahringer. 2021. Multi-GPU room response simulation with hardware raytracing. *Concurrency and Computation: Practice and Experience (2021)*, e6663.
- [7] Emanuele Vitali, Davide Gadioli, Gianluca Palermo, Andrea Beccari, and Cristina Silvano. 2018. Accelerating a geometric approach to molecular docking with OpenACC. In *Proceedings of the 6th International Workshop on Parallelism in Bioinformatics*. 45–51.
- [8] Michael Wong. 2021. SYCL State of the Union Keynote. Presented at IWOCL'21. <https://www.iwocl.org/wp-content/uploads/k04-iwocl-syclcon-2021-wong-slides.pdf>