Experiences With Adding SYCL Support to GROMACS

Andrey Alekseenko
Szilárd Páll
Erik Lindahl
andreyal@kth.se
pszilard@kth.se
erik@kth.se
KTH Royal Institute of Technology
Stockholm, Sweden

ABSTRACT

GROMACS is an open-source, high-performance molecular dynamics (MD) package primarily used for biomolecular simulations, accounting for 5% of HPC utilization worldwide. Due to the extreme computing needs of MD, significant efforts are invested in improving the performance and scalability of simulations. Target hardware ranges from supercomputers to laptops of individual researchers and volunteers of distributed computing projects such as Folding@Home. The code has been designed both for portability and performance by explicitly adapting algorithms to SIMD and data-parallel processors. A SIMD intrinsic abstraction layer provides high CPU performance. Explicit GPU acceleration has long used CUDA to target NVIDIA devices and OpenCL for AMD/Intel devices.

In this talk, we discuss the experiences and challenges of adding support for the SYCL platform into the established GROMACS codebase and share experiences and considerations in porting and optimization. While OpenCL offers the benefits of using the same code to target different hardware, it suffers from several drawbacks that add significant development friction. Its separate-source model leads to code duplication and makes changes complicated. The need to use C99 for kernels, while the rest of the codebase uses C++17, exacerbates these issues. Another problem is that OpenCL, while supported by most GPU vendors, is never the main framework and thus is not getting the primary support or tuning efforts. SYCL alleviates many of these issues, employing a single-source model based on the modern C++ standard. In addition to being the primary platform for Intel GPUs, the possibility to target AMD and NVIDIA GPUs through other implementations (*e.g.*, hipSYCL) might make it possible to reduce the number of separate GPU ports that have to be maintained.

Some design differences from OpenCL, such as flow directed acyclic graphs (DAGs) instead of in-order queues, made it necessary to reconsider the GROMACS's task scheduling approach and architectural choices in the GPU backend. Additionally, supporting multiple GPU platforms presents a challenge of balancing performance (low-level and hardware-specific code) and maintainability (more generalization and code-reuse). We will discuss the limitations of the existing codebase and interoperability layers with regards to adding the new platform; the compute performance and latency comparisons; code quality considerations; and the issues we encountered with SYCL implementations tested. Finally, we will discuss our goals for the next release cycle for the SYCL backend and the overall architecture of GPU acceleration code in GROMACS.

CCS CONCEPTS

• **Applied computing** → *Molecular structural biology*; *Physics*; • **Computing methodologies** → **Parallel programming languages**; **Parallel algorithms**.

KEYWORDS

GROMACS, SYCL, Heterogeneous acceleration

ACM Reference Format:

Andrey Alekseenko, Szilárd Páll, and Erik Lindahl. 2021. Experiences With Adding SYCL Support to GROMACS. In *International Workshop on OpenCL (IWOCL'21), April 27–29, 2021, Munich, Germany*. ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3456669.3456690

ACKNOWLEDGMENTS

The authors would like to thank Intel Corporation for the financial support, and in particular Roland Shultz and Heinrich Bockhorst for many stimulating discussions.

IWOCL'21, April 27–29, 2021, Munich, Germany
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9033-0/21/04.
https://doi.org/10.1145/3456669.3456690