



Comparing the Performance of SYCL Runtimes for Molecular Dynamics Applications

Andrey Alekseenko

andreyal@kth.se

andreyal@kth.se

SciLifeLab, KTH Royal Institute of Technology
Stockholm, Sweden

Szilárd Páll

pszilard@kth.se

PDC Center for High Performance Computing, KTH Royal
Institute of Technology
Stockholm, Sweden

ABSTRACT

SYCL is a cross-platform, royalty-free standard for programming a wide range of hardware accelerators. It is a powerful and convenient way to write standard C++ 17 code that can take full advantage of available devices. There are already multiple SYCL implementations targeting a wide range of platforms, from embedded to HPC clusters. Since several implementations can target the same hardware, application developers and users must know how to choose the most fitting runtime for their needs. In this talk, we will compare the runtime performance of two major SYCL runtimes targeting GPUs, oneAPI DPC++ and Open SYCL [3], to the native implementations for the purposes of GROMACS, a high-performance molecular dynamics engine.

Molecular dynamics (MD) applications were one of the earliest adopters of GPU acceleration, with force calculations being an obvious target for offloading. It is an iterative algorithm where, in its most basic form, on each step, forces acting between particles are computed, and then the equations of motions are integrated. As the computational power of the GPUs grew, the strong scaling problem became apparent: the biophysical systems modeled with molecular dynamics typically have fixed sizes, and the goal is to perform more time steps, each taking less than a millisecond of wall time. This places high demands on the underlying GPU framework, requiring it to efficiently schedule multiple small tasks with minimal overhead, allowing to achieve overlap between CPU and GPU work for large systems and allowing to keep GPU occupied for smaller systems. Another requirement is the ability of application developers to have control over the scheduling to optimize for external dependencies, such as MPI communication.

GROMACS is a widely-used MD engine, supporting a wide range of hardware and software platforms, from laptops to the largest supercomputers [1]. Portability and performance across multiple architectures have always been one of the primary goals of the project, necessary to keep the code not only efficient but also maintainable. The initial support for NVIDIA accelerators, using CUDA, was added to GROMACS in 2010. Since then, heterogeneous parallelization has been a major target for performance optimization, not limited to NVIDIA devices but later adding support for GPUs

of other vendors, as well as Xeon Phi accelerators. GROMACS initially adopted SYCL in its 2021 release to replace its previous GPU portability layer, OpenCL [2]. In further releases, the number of offloading modes supported by the SYCL backend steadily increased. As of GROMACS 2023, SYCL support in GROMACS achieved near feature parity with CUDA while allowing the use of a single code to target the GPUs of all three major vendors with minimal specialization.

While this clearly supports the portability promise of modern SYCL implementations, the performance of such portable code remains an open question, especially given the strict requirements of MD algorithms. In this talk, we compare the performance of GROMACS across a wide range of system sizes when using oneAPI DPC++ and Open SYCL runtimes on high-performance NVIDIA, AMD, and Intel GPUs. Besides the analysis of individual kernel performance, we focus on the runtime overhead and the efficiency of task scheduling when compared to a highly optimized implementation using the native frameworks and discuss the possible sources of suboptimal performance and the amount of vendor-specific code branches, such as intrinsics or workarounds for compiler bugs, required to achieve the optimal performance.

CCS CONCEPTS

• **Computing methodologies** → **Massively parallel and high-performance simulations**; *Parallel algorithms*; • **Applied computing** → *Molecular structural biology*; • **Software and its engineering** → **Parallel programming languages**; • **Computer systems organization** → **Heterogeneous (hybrid) systems**.

KEYWORDS

SYCL, GROMACS, heterogeneous acceleration, molecular dynamics, performance-portability

ACM Reference Format:

Andrey Alekseenko and Szilárd Páll. 2023. Comparing the Performance of SYCL Runtimes for Molecular Dynamics Applications. In *International Workshop on OpenCL (IWOCCL '23)*, April 18–20, 2023, Cambridge, United Kingdom. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3585341.3585350>

ACKNOWLEDGMENTS

This work was supported by the Swedish e-Science Research Center, the BioExcel-2 CoE (H2020-INFRAEDI-02-2018-823830), the Swedish Foundation for Strategic Research (SeCI), and the Intel Corporation.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IWOCCL '23, April 18–20, 2023, Cambridge, United Kingdom

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0745-2/23/04.

<https://doi.org/10.1145/3585341.3585350>

REFERENCES

- [1] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C. Smith, Berk Hess, and Erik Lindahl. 2015. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* 1-2 (2015), 19–25. <https://doi.org/10.1016/j.softx.2015.06.001>
- [2] Andrey Alekseenko, Szilárd Páll, and Erik Lindahl. 2021. Experiences With Adding SYCL Support to GROMACS. In *International Workshop on OpenCL* (Munich, Germany) (*IWOCL'21*). Association for Computing Machinery, New York, NY, USA, Article 17, 1 pages. <https://doi.org/10.1145/3456669.3456690>
- [3] Aksel Alpay, Bálint Soproni, Holger Wünsche, and Vincent Heuveline. 2022. Exploring the Possibility of a HipSYCL-Based Implementation of OneAPI. In *International Workshop on OpenCL* (Bristol, United Kingdom, United Kingdom) (*IWOCL'22*). Association for Computing Machinery, New York, NY, USA, Article 10, 12 pages. <https://doi.org/10.1145/3529538.3530005>