Deliverable 3.1 – SYCLOPS Reference Platform v1.0 Release & Validation

GRANT AGREEMENT NUMBER: 101092877

Project acronym:	SYCLOPS
Project full title:	Scaling extreme analYtics with Cross architecture acceLeration based on OPen Standards
Call identifier:	HORIZON-CL4-2022-DATA-01-05
Type of action:	RIA
Start date:	01/01/2023
End date:	31/12/2025
Grant agreement no:	101092877

D3.1 – SYCLOPS Reference Platform v1.0 Release & Validation

Executive Summary: At M18, we have completed the deployment of v1.0 of SYCLOPS stack. In “*D2.1 – Architecture, interface, and benchmark specification”,* we described the interfaces between various SYCLOPS layers, and detailed the work required to verify compatibility. At M18, we have performed all these activities to verify compatibility across key layers using the deployed EMDC. This deliverable is a summary of these achievements.

WP:	**4**		
Author(s):	Jan Kaštil, Martin Božek, Jaroslav Pavlík, Fred Buining, Uwe Dolinsky, Kumudha Narasimhan, Mehdi Goli, Aksel Alpay, Raja Appuswamy		
Editor:	Raja Appuswamy		
Leading Partner:	HIRO		
Participating Partners:	CSIP, CPLAY, UHEI, EUR		
Version:	1.0	**Status:**	Draft
Deliverable Type:	Other	**Dissemination Level:**	PU - Public
Official Submission Date:	30-06-2024	**Actual Submission Date:**	01-07-2024

Disclaimer

This document contains material, which is the copyright of certain SYCLOPS contractors, and may not be reproduced or copied without permission. All SYCLOPS consortium partners have agreed to the full publication of this document if not declared “Confidential”. The commercial use of any information contained in this document may require a license from the proprietor of that information. The reproduction of this document or of parts of it requires an agreement with the proprietor of that information.

The SYCLOPS consortium consists of the following partners:

No.	Partner Organisation Name	Partner Organisation Short Name	Country
1	EURECOM	EUR	FR
2	INESC ID - INSTITUTO DE ENGENHARIADE SISTEMAS E COMPUTADORES, INVESTIGACAO E DESENVOLVIMENTO EM LISBOA	INESC	PT
3	RUPRECHT-KARLS-UNIVERSITAET HEIDELBERG	UHEI	DE
4	ORGANISATION EUROPEENNE POUR LA RECHERCHE NUCLEAIRE	CERN	CH
5	HIRO MICRODATACENTERS B.V.	HIRO	NL
6	ACCELOM	ACC	FR
7	CODASIP S R O	CSIP	CZ
8	CODEPLAY SOFTWARE LIMITED	CPLAY	UK

Document Revision History

Version	Description	Contributions
0.1	Structure and outline	**EUR**
0.2	Version with CSIP RISC-V solution & OCK	**CSIP, CPLAY, EUR**
0.3	Version with AdaptiveCpp and DPC++	**CPLAY, UHEI**
1.0	Final version	**EUR**

Authors

Author	Partner
Jan Kaštil, Martin Božek, Jaroslav Pavlík	**CSIP**
Fred Buining	**HIRO**
Uwe Dolinsky, Kumudha Narasimhan, Mehdi Goli	**CPLAY**
Aksel Alpay	**UHEI**
Raja Appuswamy	**EUR**

Reviewers

Name	Organisation
Aleksandar Ilic	INESC
Vincent Heuveline	UHEI
Stefan Roiser	CERN
Nimisha Chaturvedi	ACC
Martin Božek	CSIP
Mehdi Goli	CPLAY

Statement of Originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Table of Contents

List of Figures

Executive Summary

At M18, we have completed the deployment of v1.0 of SYCLOPS stack. Our hardware testbed containing (i) GPU servers, (ii) MilkV Pioneer board with Sophon SG2042 RISC-V CPU, and (iii) CSIP FPGA board with A730 RISC-V soft core, has been deployed at EUR. In "*D2.1 – Architecture, interface, and benchmark specification",* we described the interfaces between various SYCLOPS layers, and detailed the work required to verify compatibility. At M18, we have performed all these activities to verify compatibility across key layers using the deployed EMDC. This deliverable is a summary of these achievements.

1 Introduction

Figure 1 shows the SYCLOPS hardware—software stack consists of three layers: (i) infrastructure layer, (ii) platform layer, and (iii) application libraries and tools layer.

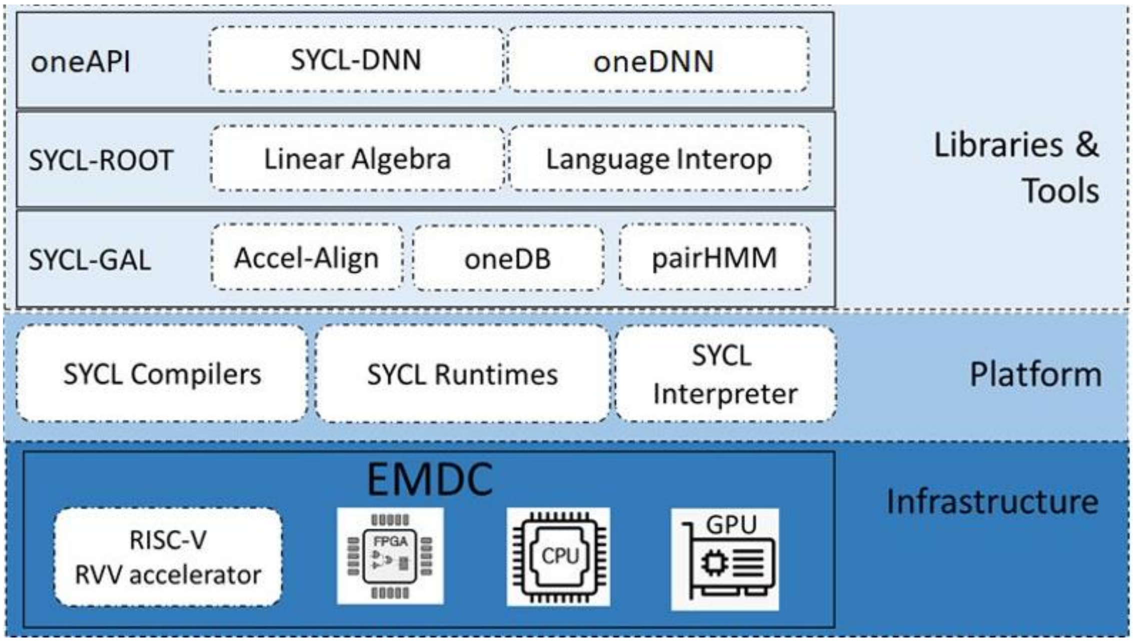

Figure 1: SYCLOPS architecture

Infrastructure layer: The SYCLOPS infrastructure layer is the bottom-most layer of the stack and provides heterogeneous hardware with a wide range of accelerators from several vendors.

Platform layer: The second layer from the bottom, the platform layer, provides the software required to compile, execute, and interpret SYCL applications over processors in the infrastructure layer.

Application libraries and tools layer: While the platform layer described above enables direct programming in SYCL, the libraries layer enables API-based programming by providing pre-designed, tuned libraries for the three SYCLOPS use cases.

In deliverable *"D2.1 – Architecture, interface, and benchmark specification"*, we had defined the key interfaces that would be verified at two key points in SYCLOPS—once at M18 using v1.0 of the SYCLOPS platform, and the second time at M33 using v2.0, the final version, of the SYCLOPS platform. The rest of this deliverable provides a description of activities undertaken in each of the three layers to complete the interim verification at M18. In each case, we describe the interfaces and integration efforts we foresaw and described in *D2.1,* and explain how we achieved our goals.

2 SYCLOPS Infrastructure Layer

In *D2.1*, we had mentioned that the infrastructure layer will consist of two key pieces that will be integrated and verified at M18. We recap these below.

1. An RISC-V application core based on CSIP application class processors will be developed. A precompiled bitstream for a selected FPGA board will be generated. It will be capable of running a full-featured RTOS and offers high levels of customization.
2. A first version of the edge data center (EDC) will be developed. As building a miniaturized comHPC form factor of various boards will be done during the course of the project, EDC v1.0 will developed in standard form factors during COTS hardware with PCIe Gen5. We refer to this as EDC to distinguish it from the v2.0 which will be an Edge Micro Data Center (EMDC).

In Sections 2.1 and 2.2, we describe the work done to build these two goals.

2.1 RISC-V

2.1.1 CSIP FPGA Platform

The RISC-V core developed by CSIP for the SYCLOPS project is based on CSIP's advanced, customizable RISC-V architecture, specifically the Codasip A730. This versatile mid-range 64-bit dual issue in order application core is ideal for a broad range of applications, offering significant improvements over previous generations. The core is compliant with the RVA22 RISC-V profile. The technical specifications are as follows:

- **lArchitecture:** 64-bit in-order dual-issue RISC-V core.
- **ISA:** Supports base integer set, RV64I.
- **Microarchitecture:** In order, 9 stages, dual issue pipeline.
- **Configuration & customization:** Flexible design with options to configure the number of cores (one to four), private L1 caches and shared L2 cache with full coherency, memory protection, branch prediction, and an interrupt controller compliant with the RISC-V standard.
- **Performance:** 2x the performance of previous generations, designed for complex compute tasks in power-constrained devices.

In SYCLOPS, CSIP is working towards enhancing the RISC-V core's capabilities, particularly focusing on Single-Instruction Multiple Data (SIMD) capabilities and effective data processing. To this end, CSIP is improving the CodAL processor description language to support RVV accelerators and working on prioritizing performance, power, and area improvements for the RVV accelerator. Our goal is to complete the development and deployment of the RVV accelerator in EMDC v2.0 by M33. However, in order to ensure integration with other software developed in SYCLOPS and to enable end-to-end testing at the interim stage at M18, we developed an FPGA platform.

The FPGA board used for this platform is the Genesys 2 Kintex-7 FPGA Development Board. Key features and specifications of the board are as follows:

- **Logic Cells:** 50,950 logic slices (4 6-input LUTs & 8 flip-flops each)
- **Block RAM:** 16 Mbits

- **DSP Slices:** 840
- **CMTs:** 10, each with PLL
- **DDR3:** 1800 MT/s with 32-bit data width
- **Internal Clock:** 450 MHz+
- **Quad-SPI Flash:** 16 MB
- **Ethernet:** 10/100/1000 PHY
- **Connectivity and On-board I/O:**
  - **Pmod Connectors:** 5 Pmod ports
  - **VGA:** 1 VGA connector
  - **HDMI:** HDMI Sink and HDMI Source
  - **SD:** microSD card connector
  - **USB:** HID host for mouse, keyboard, and USB MSD host for storage
  - **Audio:** Audio codec with four 3.5mm jacks
  - **FMC Connector:** Fully-populated 400-pin FMC HPC connector with ten GTX lanes
  - **Switches:** 8
  - **Buttons:** 6
- **Power:** 12V ±5% external power only
- **Dimensions:** Width 5.8 in, Length 6.8 in

The FPGA platform defines the set of the features that are implemented in the FPGA design and are accessible from the processor. Not all features of the board are supported in the current platform due to the implementation reasons. The provided FPGA platform instantiated the CSIP A730 core and was prepared for the Genesys 2 board. The platform was distributed as a bitstream ready to be uploaded into the FPGA. The platform is called hobgoblin and contains following components:

- Interrupt controller
- Timer
- UART controller
- SPI Controller connected to the SD Card reader
- On-chip RAM
- 100Mbps Ethernet MAC
- DDR3 Memory controller

The platform allows users to run bare metal applications. To this end, the platform includes board support package for the FPGA board. SDK and tools to develop software using LLVM and GCC compilers is integrated. In addition, we also extended the platform to enable users to run a full-fledged Linux OS. For the SYCLOPS project, the Poky Linux distribution was selected. The Poky distribution was modified to support the hobgoblin platform. The Yocto manifest for the modified distribution can be found on github. The resulting platform can be accessed through the UART console or over the ethernet by SSH protocol.

2.1.2 MilkV

The CSIP FPGA platform provides a reference for testing the RISC-V accelerator offload path, where kernels implemented in SYCL are offloaded to a RISC-V based accelerator. However, the platform does not provide an actual taped-out RISC-V CPU. The compiler and runtime toolchains in SYCLOPS can generate code for RISC-V processors. In order to test them, we needed a real RISC-V CPU in addition to the FPGA emulation from CSIP.

To this end, we expanded our SYCLOPS infrastructure by purchasing and integrating the Pioneer board sold by MilkV. The key specifications of the MilkV board we deployed are as follows:

- SOPHGO SG2042 64-core RISC-V CPU based on T-Head Semiconductor's XuanTie C920 64-bit processor design clocked at 2GHz.
- 128GB 3200 DDR4
- 1x 1TB PCIe 3.0 SSD
- 1x Intel X540-T2 Network Card with 2x 10Gbps RJ45 ports
- 1x AMD R5 230 Graphic Card

By porting our compiler toolchains to this board, we were able to compile and execute SYCL kernels on a real, taped out RISC-V CPU. A detailed description of our compiler porting efforts is outlined in Section 3 of this deliverable and described in detail in deliverable *D4.1.* Section 4 of this deliverable presents performance evaluation using the MilkV board.

2.2 Heterogeneous Edge Data Center (EDC) Server

The Edge Micro Data Centers (EMDC) R&D is executed in two stages, making use of the capabilities of two custom built EMDCs, EMDC v1.0 and EMDC v2.0, and allows SYCL software to be tested on accelerators from different manufacturers (Nvidia, Intel, AMD), RISC-V cores, and allows researching the effect of state-of-the-art fabrics like PCIe gen5 and CXL.

2.2.1 EDC v1.0

While v2.0 of the EMDC will be delivered by M33, we wanted to develop an interim v1.0 of the EMDC to enable M18 verification of key interfaces across the SYCLOPS stack. As this is an interim solution, we planned to use COTS components with commodity server form factors for our v1.0 solution. Hence, we call it an EDC instead of an E**M**DC. This will also help us to integrate the RISC-V boards described earlier and to meet the second requirement we set forth at the beginning of this section.

To this end, we have configured and developed a customized, heterogeneous EDC. Three key requirements of this server were (i) support for PCIe 5.0, (ii) ability to house multiple GPU accelerators from different vendors, and (iii) support integration with the RISC-V boards described earlier. In addition, in collaboration with EUR, we developed a requirement list for the server based on inputs from SYCLOPS partners, especially the use case partners. Based on this, we built a custom EDC for research purposes only, with the collaborative support from various manufacturers (CSIP, HIRO, Intel, Micron technologies, Supermicro). We worked with SuperMicro in customizing a server build, as none of their default servers could meet all our requirements. The resulting EDC was based on SYS-221GE-NR Supermicro server, a multiple-GPU server normally equipped with GPU's from 1 manufacturer. HIRO's hardware lab worked with SuperMicro in specing the server to support GPUs from different manufacturers, and the CSIP FPGA. The key specifications of the final EDC that we developed and deployed are as follows:

- Intel Xeon Platinum CPU
- 64GB DDR5-5600
- 1TB Samsung PCIe SSD
- NVIDIA L40S GPU

- AMD Instinct MI210 accelerator
- Intel GPU Flex 170 GPU

Across this EDC and the RISC-V boards, our SYCLOPS platform contains three types of processing backends (CPU, GPU, FPGA) from four different vendors (Intel, XuanTie, AMD, NVIDIA) making it a truly heterogeneous testbed.

2.2.2 EMDC v2.0

The datacenter market will, the coming decade experience serious disruption from PCIe CXL technology, becoming the dominant fabric for data dense environments where big data and AI workloads are processed, not only in data centers but also edge environments (for example smart car). CXL allows to create composable infrastructure from pools of CPU's memories, GPU's, etc. The RISC-V standard is already provisioned to CXL capabilities.

Currently EMDC v2.0 is being built by HIRO and carries a server Petascale 2U Proof of Concept (POC) unit. This special server will carry an AMD (dual SOC, CXL capable), special cables to connect 2x CZ120 CXL memory (micron technology) from backplane to the board, allowing to extend the CPU's memory with 0.5TB of the extra memory, and an A10 NVidia GPU. We will create a link between 1st Gen EMC and 2nd Gen EMC, test SYCL on the accelerators and research the effect of CXL memory extension on accelerated workloads.

The POC EMDCs that HIRO currently builds for SYCLOPS project are not yet in designed in HIRO's current ultra-compact-built (5.4kW in 3U) EMDCs that are based on the com-express form factor with unprecedented energy efficient PUE 1.03. HIRO has also started the development of a new line of EMDCs with the above-mentioned capabilities, equally compact (based on the Com-HPC form factor) and equally energy efficient. This new line of EMDC will be twice as powerful and CXL capable. For the SYCLOPS project HIRO will demonstrate that this type of high performing hardware can be built in their small form factor and deployed at the edge. The com-HPC component will be either a CXL capable PCIe switch or CXL capable CPU board, or an accelerator. Currently we are discussing components availability with the OEM's.

3 SYCLOPS Platform Layer

In *D2.1*, we had mentioned that the platform layer will consist of two compilers that will be integrated and verified at M18. We recap this below.

1. The DPC++ compiler will be extended to support RISC-V. The oneAPI construction kit (OCK) will be used to support RISC-V accelerators in the SYCLOPS platform
2. The AdaptiveCPP compiler will rely on OCK to target RISC-V devices. To this end, it will support SPIR-V code generation.

The platform layer also has the Cling SYCL interpreter from CERN. However, Cling relies on AdaptiveCpp as its backend. Thus, it does not have any special interfaces for verification, as a functional AdaptiveCpp that can compile to RISC-V and other accelerators in SYCLOPS is all that is required.

3.1 Offload to CSIP RISC-V FPGA platform with OCK

Support for new accelerators is done with the OCK by developing a new custom target. A custom target is made up of three key parts: (i) Runtime code (ComputeMux Runtime), (ii) Compiler code ComputeMux Compiler), and (iii) a HAL (Hardware Abstraction Layer). To support the FPGA, we developed a remote HAL, which is essentially a shim layer to enable offloading kernel code to a device that may be located remotely. Details about the HAL are described in deliverable *“D4.1-RISC-V Compiler Backends”*.

In order to test OCK with the CSIP RISC-V FPGA platform, we booted the platform and configured the Poky Linux distribution running on the RISC-V soft cores to have a static IP address. Following this, we compiled and ran the HAL server as a normal user process on Linux. We ran the HAL client on a x86_64 server with Ubuntu 22 distribution. Using this setup, we tested and verified the offload of SYCLDB kernels to the RISC-V FPGA. Section 4 describes these SYCLDB kernels.

3.2 AdaptiveCpp and DPC++ on EDC servers

We have configured and built the latest version of DPC++ and AdaptiveCpp on both our MilkV and EDC servers. We have also configured and built OCK on the MilkV RISC-V server to test both the host CPU and accelerated offload path. A comparative evaluation DPC++ and AdaptiveCpp under all accelerators available in SYCLOPS platform is presented in Section 5.

4 Libraries & Tools Layer

SYCLOPS libraries layer spans three libraries, namely, SYCL-DNN library of deep learning methods for the autonomous systems use case, SYCL-ROOT library of mathematical operators for scalable High-Energy Physics analysis, and SYCL-GAL library of data parallel algorithms for scalable genomic data analysis. The libraries are being actively developed and are due to be released only at M33 as a part of deliverables *D5.1—D5.4*. As these libraries are entirely developed in SYCL, all that is required to verify compatibility is to ensure that the compilers and runtimes in the Platform layer can compile SYCL code to generate architecture-specific code and execute it for a variety of processor backends.

To this end, in order to demonstrate a complete integration of various interfaces at the infrastructure and platform level, we used SYCLDB[1]—a SYCL-based, vectorized query execution engine that can execute SQL queries on multi-vendor CPUs and GPUs. What makes SYCLDB relevant to our evaluation is that it implements parallel, heterogeneous SQL operators in SYCL that are compact, highly optimized to use tile-based execution, and easy to port across heterogeneous hardware.

4.1 End-to-end evaluation with SYCLDB

In order to test full integration, we chose to evaluate a key kernel in SYCLDB, namely, a projection kernel, which perform a predicate check on each tuple on a database column equivalent to a SQL statement *SELECT 2 * R.a + 3 * R.b AS R.c FROM R WHERE R.a = <p>*. For this experiment, we use columns of integers configured to have a specific cardinality. In this section, we will present evaluation results for various processing backends available in the SYCLOPS platform.

4.1.1 CPU Backends

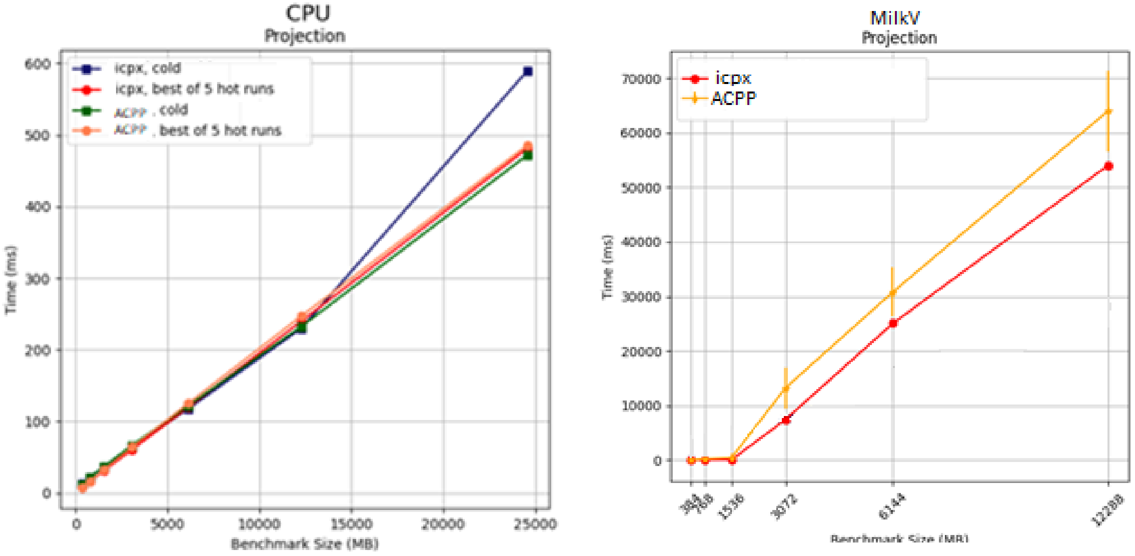

Figure 2: Execution time of SYCLDB kernel under Intel CPU (left) and MilkV RISC-V CPU (right)

[1] https://www.eurecom.fr/publication/6567/download/data-publi-6567.pdf

Figure 2 shows the performance of the projection kernel under an Intel CPU (left) and the MilkV RISC-V CPU (right) when SYCLDB is compiled with either DPC++ compiler (shown as icpx) or AdaptiveCpp. Just for the Intel CPU, we show the execution time of both cold run, where the kernel is compiled and executed for the first time, and hot run which are obtained by running a precompiled kernel again repeatedly. First notice that on the Intel CPU, while there is a difference in performance between cold runs of DPC++ and AdaptiveCpp, hot runs are almost identical. The kernel is able to 25GB of data in under 500 milliseconds, at a rate of 55GB/s. On the MilkV CPU, we see that both compilers are able to generate code for the RISC-V CPU. We found the code generated by AdaptiveCpp to be slightly more optimized than DPC++, but we are looking into optimizations and configurations that could bridge this gap. Our goal in presenting these results are not to compare the performance of the RISC-V CPU and Intel CPU directly. But as can be seen, the Intel CPU provides substantially better performance than the SOPHON SG2024 RISC-V CPU.

4.1.2 GPU Backends

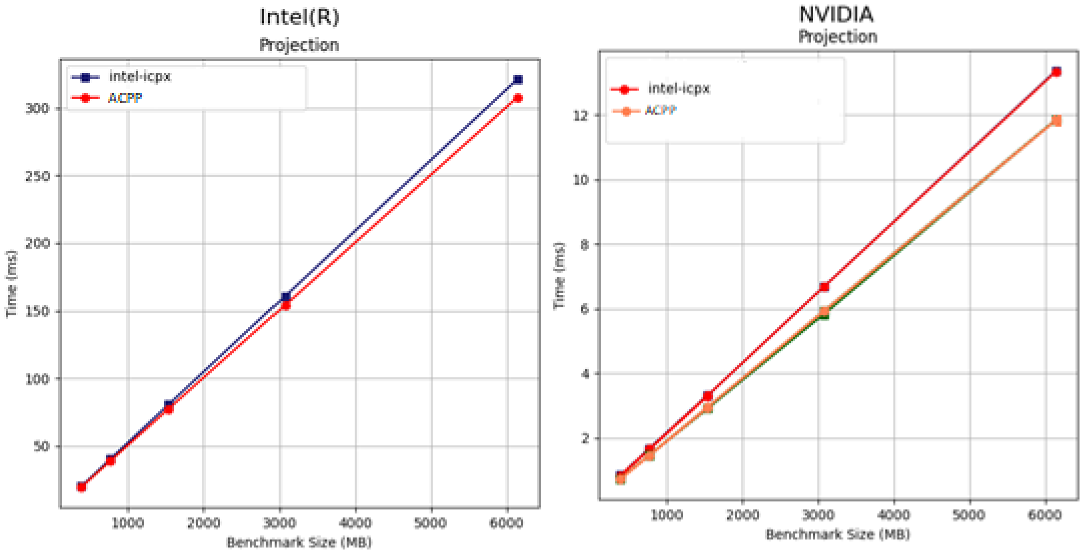

Figure 3: Execution time of SYCLDB kernel under Intel iGPU (left) and NVIDIA dGPU (right)

Figure 3 shows the performance of the projection kernel under an Intel iGPU (left) and a NVIDIA dGPU (right) when SYCLDB is compiled with either DPC++ compiler (shown as icpx) or AdaptiveCpp. Once again, these results show that both compilers can target both integrated and discrete GPUs from multiple vendors. As can be seen, SYCLDB kernel provides similar performance under both compilers, with AdaptiveCpp producing slightly better performance. The dGPU outperforms the iGPU as expected. Comparing this with CPU results, we can see that the dGPU provides substantial improvement over the CPU as well, making a case for acceleration of database queries. We also compared the performance of projection kernel written in SYCL with an equivalent one written in CUDA. We found that both AdaptiveCpp and DPC++ compiled SYCL kernels were as fast as the kernels written in CUDA and compiled with NVCC. This makes the case for cross-architecture programming using an open, standard language like SYCL.

4.1.3 RISC-V Accelerator Backend with CSIP FPGA & OCK

For our final evaluation, Figure 4 shows the performance of the SYCLDB projection kernel when using the RISC-V FPGA with CSIP A730 CPU. As a reminder, the RISC-V CPU in this case was treated as an accelerator and not a host. As a result, OCK was used to offload the projection kernel from the x86 host to the FPGA and executed on the RISC-V CPU. The graph shows the execution time in milliseconds of both the first run (marked cold run) and subsequent executions (marked hot run) in the figure. We limited the table size to a few hundred MBs as the Digilent FPGA's on-board memory is limited in size and larger kernels could not be executed.

Clearly, the result shows that the end-to-end integration works for the RISC-V accelerator path. Comparing this with the previous results, we can see that the FPGA execution time is substantially higher for the projection kernel. This is expected as the RISC-V CPU is as a softcore processor clocked at a very low frequency that is running the full Linux OS. In a real accelerator scenario, we would have a taped-out RISC-V accelerator and not the FPGA emulation. But irrespective of this, these results indicate that the integration works and that it is possible to use the OCK to offload kernels to a RISC-V accelerator.

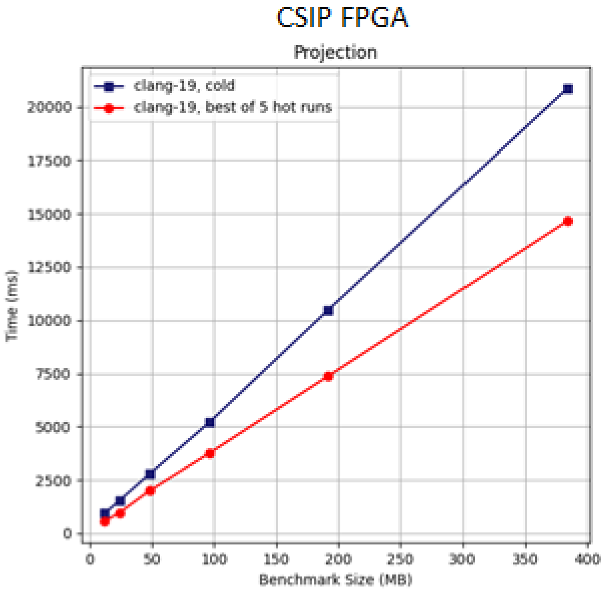

Figure 4: Execution time of SYCLDB kernel offloaded using OCK to CSIP FPGA with RISC-V softcore

5 Conclusion

This deliverable marks the completion of milestone M5 and the release of the first version of of the SYCLOPS stack. We have deployed the SYCLOPS hardware infrastructure at EUR and used SYCLDB to perform end-to-end experiments covering the several different processor backends (x86_64 CPU as host, RISC-V CPU as host, multi-vendor integrated and discrete GPU as accelerator, CSIP FPGA RISC-V soft core processor as accelerator via OCK). Our efforts thus far guarantee that all SYCL libraries that are being developed in the project will be able to run on the hardware in the SYCLOPS infrastructure layer using compilers and runtimes developed in the SYCLOPS platform layer.