

Porting and Measuring the Performance of SPH-EXA on SYCL

Master Project (6 CP)

University of Basel
Faculty of Science
Department of Mathematics and Computer Science
High Performance Computing Group

Advisor and Examiner: Prof. Dr. Florina M. Ciorba
Supervisor: Dr. Osman Seckin Simsek

Author: Yannick Martin
Email: yannick.martin@stud.unibas.ch

January 17, 2025



Contents

1	Introduction	2
2	Background	3
2.1	SPH-EXA	3
2.2	OneAPI	3
3	Methodology	4
3.1	Migration Process	4
3.2	Prepare for Migration	4
3.3	Migrate your Code	5
3.3.1	Compilation Database	5
3.3.2	Source Code Migration	5
3.3.3	CMake Files Migration	6
3.4	Review the Migrated Code	6
3.5	Build the New SYCL Code Base	7
3.6	Validate the New SYCL Application	7
3.6.1	Personal Computer	7
3.6.2	miniHPC Cluster	8
3.6.3	Davinci	8
4	Results and Discussion	9
4.1	Migration Results	9
4.1.1	Metrics	9
4.1.2	Tests	12
4.2	Accuracy and Performance Analysis	13
5	Conclusion	14
	Appendix	16
A.1	List of Compilation Errors	16
A.2	Parallel Programming Models	19
A.3	Lines of Code	20
A.4	Code Structure SPH-EXA Sedov Test Case	21
A.5	List of all warnings	22

Chapter 1

Introduction

Portability is a critical consideration in modern High-Performance Computing (HPC) applications and is becoming increasingly important as hardware architectures grow more diverse. The advantages of portability are clear: software that can operate across hardware from multiple manufacturers enhances accessibility and usability for scientists worldwide. SYCL exemplifies this principle, offering a unified programming model that enables code to run on GPUs from Intel, NVIDIA, and AMD. In contrast, CUDA is limited to NVIDIA GPUs, restricting flexibility and cross-platform deployment.

This report documents the steps I undertook to adapt SPH-EXA to SYCL, the obstacles encountered during the process, and the strategies used to overcome them. It then examines the results of the migration, including an assessment of the implementation's outcomes from migrating SPH-EXA to SYCL.

The source code of the migrated project, as well as additional material, is available on the Bitbucket repository: <https://bitbucket.org/unibasdmihpc/yannick-martin-msc-project/src/main/>

For the report on my master project, I used DeepL Write to improve the language of texts that I had written myself. An AI tool did not create new content. I have checked all the texts and take full responsibility for the result.

Chapter 2

Background

2.1 SPH-EXA

SPH stands for smoothed particle hydrodynamics. It is a Lagrangian and mesh-free method, meaning that it models the system using discrete particles that move with the flow, rather than relying on a fixed grid or mesh. The SPH method is used in different fields of research to understand how fluids and plasma behave under complex physical conditions. Such simulations often use a large number of particles. To be efficient, the code should be parallelized. Parallelizing SPH codes is challenging due to their boundless nature and the absence of a structured grid, leading to continuously changing interactions between fluid elements and mechanical structures at each time step. This complexity makes SPH a challenging application for the computer science community, especially for addressing parallelization and scalability on upcoming exascale computing systems. [5]

The aim of [SPH-EXA](#) was to implement SPH simulations in such a way that they make efficient use of exascale computing systems. SPH-EXA is written in C++20 and is parallelized with MPI, OpenMP, CUDA and HIP. A detailed list of the SPH-EXA components along with their associated parallel programming models ([A.2](#)), as well as the code structure for the Sedov test case in SPH-EXA ([A.4](#)), is provided in the Appendix. [6]

2.2 OneAPI

OneAPI is an open programming model developed by Intel to enable developers to write code that runs on a variety of hardware platforms, including CPUs, GPUs, NPUs, ASICs, FPGAs, and other accelerators, without having to rewrite code for each specific device. The core of this infrastructure is SYCL, a C++-based programming abstraction for parallel computing. It simplifies the writing of code for diverse systems, making it easier to develop complex, parallel applications, and thus increasing the portability of the code. [3]

OneAPI provides tools like the DPC++ Compatibility Tool (DPCCT) and SYCLomatic to assist developers in porting existing CUDA code to SYCL, simplifying the transition to OneAPI. The primary distinction between these tools is that SYCLomatic is open-source, while both offer identical CUDA-to-SYCL conversion functionality. DPCCT claims that 90% to 95% of CUDA code is automatically migrated to SYCL code. The rest can be migrated and improved using the inline comments generated by the tools. [2]

Chapter 3

Methodology

3.1 Migration Process

A CUDA-based program can be migrated to SYCL using the DPC++ Compatibility Tool or SYCLomatic Tool by following these five steps:[3]

1. Prepare for Migration
2. Migrate your Code
3. Review the Migrated Code
4. Build the New SYCL Code Base
5. Validate the New SYCL Application

CUDA[†] to SYCL[‡] Code Migration & Development Workflow

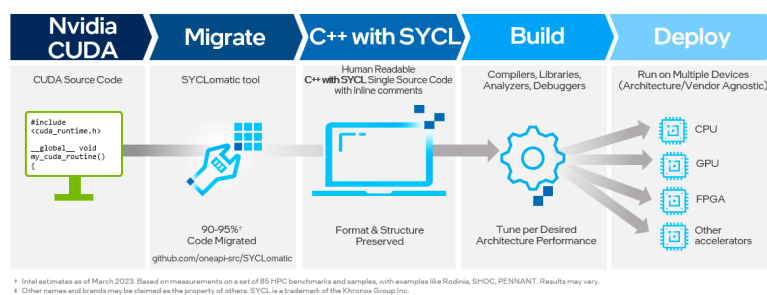


Figure 3.1: Migration Workflow

3.2 Prepare for Migration

Before starting the migration of the program, the OneAPI Toolkit must be downloaded, which includes the DPC++ Compatibility Tool, Syclomatic, and various Intel compilers.

It should also be checked that the program to be migrated has no syntax errors and is Clang compatible, as the compatibility tool uses the latest version of the Clang parser to analyze CUDA source code. [1] The SPH-EXA project has no syntax errors and is already clang-compliant, so there was no need for adjustments here.

Whenever working with oneAPI, it is necessary to initialize the oneAPI environment. This can be done by running the following command:

Listing 3.1: Initializing oneAPI environment

```
source /opt/intel/oneapi/setvars.sh
```

This step ensures that all required environment variables and paths are correctly set for oneAPI tools and libraries to function properly. [1]

3.3 Migrate your Code

3.3.1 Compilation Database

To migrate the project, a compilation database was first created with the following commands:

Listing 3.2: Compilation Database Creation Commands

```
# Create the Compilation Database
# in the directory of the Project
rm -rf build/
mkdir build
cd build
cmake ..
intercept-build make
```

The compilation database is a JSON file that provides detailed build options for each source file in a translation unit. This information can be used by the Intel DPC++ Compatibility Tool to assist in the migration of the code.[1]

It is not always necessary to create a compilation database, but with a project as large and complex as SPH-EXA, migration without a compilation database did not work. It is crucial for the intercept-build tool to locate all the necessary libraries to properly generate the database. In my case, it initially failed to detect OpenMP and the CUDA compiler. Unfortunately, no errors were displayed, which negatively impacted the migration process.

3.3.2 Source Code Migration

The following command can be used to migrate a program from CUDA to SYCL:

Listing 3.3: Migration Command

```
dpct -p=directory_CompilationDatabase/compile_commands.json
      --in-root=directory_Project --out-root=output_directory
      --gen-helper-function
```

This command uses the compilation database specified with the `-p` option. The `-in-root` option specifies the folder to be migrated. The `-out-root` option specifies where to save the migrated files. The `-gen-helper-function` option allows the tool to generate helper functions. There are many other options for influencing the migration. A list of options can be found [here](#) on page 45.^[1]

In principle, it is possible to migrate an entire project with a single command. However, when working on the SPH-EXA project, I repeatedly encountered an internal error. As a result, each subdirectory of the SPH-EXA project had to be migrated individually, and the same was true for the domain subfolder, which required its subfolders to be migrated separately as well. As the system did not provide an error message, it was not possible to identify the cause of the issue or determine a solution to fix the error directly.

The migration process generated a total of 120 warnings. Details of each warning are provided in the appendix [A.5](#). Additional information about the migration outcomes is provided in the chapter Migration Result.

3.3.3 CMake Files Migration

SPH-EXA is compiled using CMake files. There is also a command option to migrate the CMake files directly with the rest of the code. However, this approach resulted in unexpected errors, leading to the CMake files being migrated separately using this command:

Listing 3.4: Migrating CMake files

```
dpct -p=directory_CompilationDatabase/compile_commands.json
     --in-root=directory_Project --out-root=output_directory
     --migrate-build-script-only
```

Although this migration option is marked as experimental, it worked relatively well. Some adjustments were required manually, such as setting OpenMP targets and changing file extensions from `.cpp` to `.dp.cpp`.

3.4 Review the Migrated Code

In this step, the code is reviewed, and the previously generated warnings are addressed. Some of these warnings must be resolved to avoid compilation errors. Examples include DPCT1000 and DPCT1001, which relate to error handling. In addition, there are numerous warnings that do not cause compilation errors, but provide insights, such as high register pressure (DPCT1110) or suggestions for optimizing performance (DPCT1065). The focus was solely on fixing warnings that caused compilation errors. The remaining warnings were set aside for review once a buildable version was achieved.

3.5 Build the New SYCL Code Base

To build the script file, the standard cmake command is used. As the code involves SYCL, the icpx compiler should be used. However, problems were encountered with OpenMP as it could not set the targets due to being unable to find OpenMP. This required the use of additional flags:

Listing 3.5: Building the new project

```
#Build script file:
cmake -DCMAKE_CXX_COMPILER=icpx -DOpenMP_CXX_FLAGS="-qopenmp"
      -DOpenMP_CXX_LIB_NAMES="libiomp5" -DOpenMP_libiomp5_LIBRARY=
      /opt/intel/oneapi/compiler/2025.0/lib/libiomp5.so ..

#Compile the Project:
make -j3 sphexa-cuda
```

The SYCL project should now compile using the command sphexa-cuda, although the name of the command will need to be updated in a later step.

Once the critical warnings have been addressed, the program should be ready to compile. Numerous compilation errors unrelated to the warnings were encountered. This likely occurred because SPH-EXA is a large and complex program, which may have caused the DPC++ Compatibility Tool to overlook certain elements or introduce errors during the migration process. A partial list of these errors is provided in the Appendix (A.1).

3.6 Validate the New SYCL Application

The final step is to verify that the migrated code produces the same output as the original CUDA code. This involves running and validating the program's tests to ensure accuracy and correctness. [1]

As a large number of errors had to be corrected during the building of SYCL-SPH-EXA, runtime errors are also expected.

3.6.1 Personal Computer

The plan was to address the runtime errors in SPH-EXA on my personal laptop, a ThinkPad L13 Yoga equipped with an Intel Core i5 CPU and an Intel UHD Graphics 9th-generation GPU, before running it on the miniCluster. However, this approach was not feasible because the OneAPI DPC++ Compiler requires accelerators with an Intel GPU from the 11th generation or newer. The full list of hardware and software requirements for the OneAPI DPC++ Compiler can be found [here](#).

As a result, SYCL-SPH-EXA could only run on the CPU. Unfortunately, this limited debugging capabilities for the SYCL kernels, as it was not possible to pause execution within the kernels to inspect and fix memory-related errors. Consequently, debugging was carried out on miniHPC before achieving a fully functional version.

3.6.2 miniHPC Cluster

[miniHPC](#) is a small high-performance computing (HPC) cluster at the University of Basel. Since miniHPC does not have an Intel GPU, the [oneAPI for NVIDIA GPUs](#) software was required to run the SYCL code and debug it using the CUDA debugger.

The cluster operates on CentOS 7.9, which includes Glibc version 2.17. However, oneAPI for NVIDIA GPUs requires an operating system with Glibc version 2.31 or higher. Updating Glibc via the module system was not an option because it would disrupt the entire module system. As a result, it is currently not possible to run or debug SYCL on miniHPC until the operating system is upgraded.

3.6.3 Davinci

For this reason, I switched to Davinci, a computing system used by the [SADA](#) research group at the University of Basel. Davinci is equipped exclusively with NVIDIA GPUs, making the use of oneAPI for NVIDIA GPUs necessary once again. Since Davinci runs on Ubuntu 24.04.01, which includes Glibc 2.39, the installation proceeded successfully. A sample programme was successfully run to test the SYCL kernel functions and confirm that the installation was correctly configured.

The focus was then directed to the SPH-EXA project with the aim of debugging it using the CUDA debugger. Initial efforts were met with challenges in compiling the SYCL version of SPH-EXA due to linking issues with CMake and Glibc. Despite trying various approaches, including manual linking, experimenting with different compilers, and using additional flags, it took me quite a bit of time to resolve these problems. Eventually, the code was successfully compiled, and debugging began. However, before reaching the SYCL code, a segmentation fault occurred with the error message `sycl::_V1::exception: Native API failed. Native API returns: 7 (UR_RESULT_ERROR_INVALID_BINARY)` (Figure 3.2). This suggests an issue with the CUDA backend or the CUDA environment setup.

```
(cuda-gdb) next
104     auto fileReader = fileReaderFactory(ascii, MPI_COMM_WORLD);
(cuda-gdb) next
105     auto simInit    = initializerFactory<Dataset>(initCond, glassBlock, fileReader.get());
(cuda-gdb) next
terminate called after throwing an instance of 'sycl::_V1::exception'
  what():  Native API failed. Native API returns: 7 (UR_RESULT_ERROR_INVALID_BINARY)

Thread 1 "sphexa-cuda" received signal SIGABRT, Aborted.
__pthread_kill_implementation (no_tid=0, signo=6, threadid=<optimized out>) at ./nptl/pthread_kill.c:44
Download failed: Invalid argument. Continuing without source file ./nptl/./nptl/pthread_kill.c.
44     ./nptl/pthread_kill.c: No such file or directory.
```

Figure 3.2: Segmentation fault error message

Due to time constraints, further testing could not be conducted, and it was not possible to address the runtime errors. Consequently, there is currently no functional SYCL-SPH-EXA program.

Chapter 4

Results and Discussion

4.1 Migration Results

4.1.1 Metrics

This chapter presents a detailed analysis of the migration process using various visualizations. The first plot (Figure 4.1) illustrates the number of lines of code (LOC) migrated by the Intel DPC++ Compatibility Tool, sorted by subfolder. As shown, the domain subfolder contains the highest number of migrated LOC, with approximately 1400 lines. In comparison, the sph and ryoanji subfolders have 450 and 550 migrated LOC, respectively.

These proportions closely align with the original distribution of CUDA LOC within the SPH-EXA project’s subfolders. Based on this observation, it is reasonable to anticipate that the domain subfolder will also exhibit the highest number of warnings and errors during migration, reflecting its larger codebase.

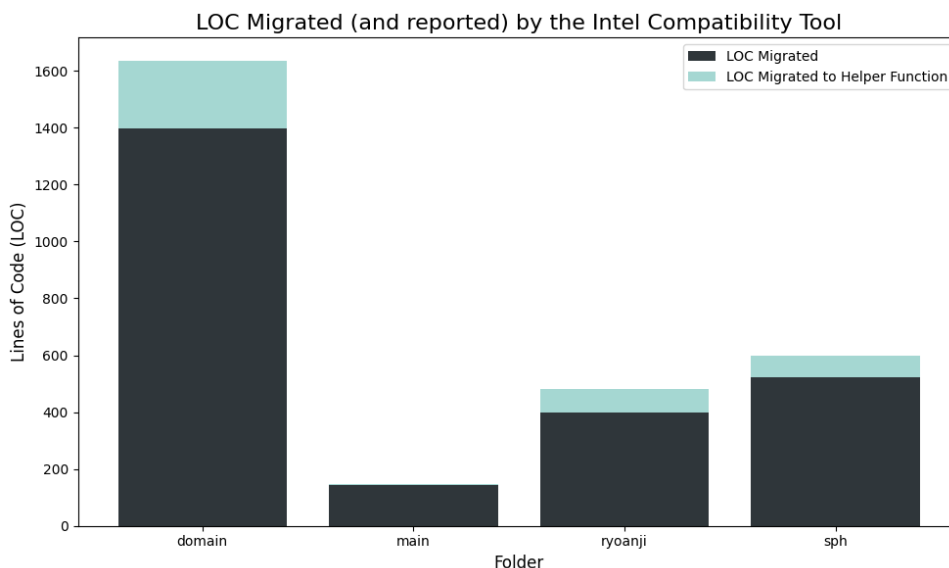


Figure 4.1: migrated lines of code by the DPC++ Compatibility Tool sorted by the four subfolders

This observation is confirmed by the next plot (Figure 4.2), which displays the number of warnings generated during the migration, sorted by subfolder. Additionally, different

warning types are distinguished using various colors in the plot.

As expected based on the first plot (4.1), the domain subfolder shows the highest number of warnings, with just under 60. The sph subfolder follows with slightly more than 40 warnings, which is nearly double the number observed in the ryoanji subfolder. A comparison of this to the ratio of migrated LOC shows that the sph subfolder exhibits a slightly higher density of warnings relative to its size. However, the overall distribution of warnings remains consistent with the pattern seen in the first plot (4.1).

Furthermore, examining the colour-coded warning types reveals that the distribution of different warning types is relatively uniform across all subfolders, with no single type of warning being disproportionately concentrated in any particular subfolder. This suggests a similar level of complexity and challenges across the project.

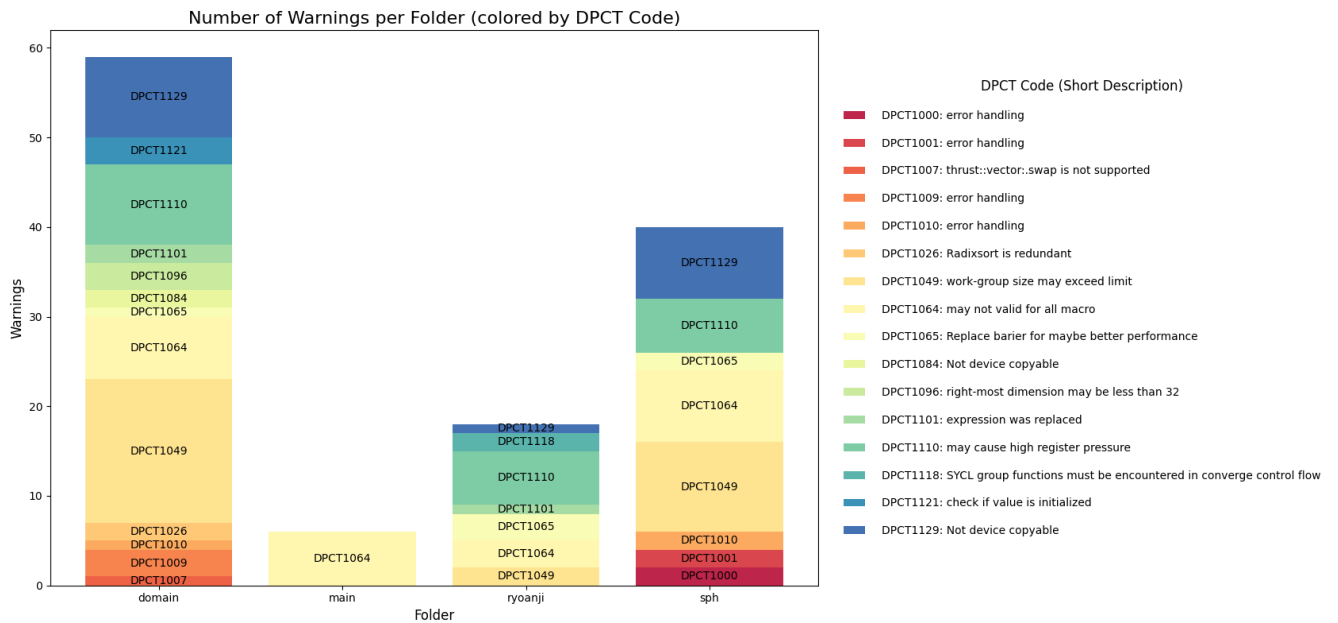


Figure 4.2: Number of Warnings sorted by the four subfolders colored by the different warnings

Figure 4.3 now shows how often the different errors occur. You can see that 4 errors occur relatively frequently, namely DPCT1049, DPCT1064, DPCT1110 and DPCT1129. These 4 warning types account for more than 75% of all warnings. Below is an explanation of each of these warning types:

- DPCT1049: The work-group size provided to the SYCL kernel might exceed the allowed limit. To determine the device’s limit, query `max_work_group_size`. If necessary, adjust the work-group size accordingly.
- DPCT1064: The migrated function call is used in a macro/template definition, and may not work for all uses of the macro/template. The DPC++ Compatibility Tool generates code that works for a specific instance of the macro call, but it may not work for all calls of the macro in the code.
- DPCT1110: The total size of the local variable declared in the device function is more than 128 bytes. This may cause high register pressure.

- DPCT1129: The type is not device copyable. If a type doesn't satisfy the requirements for being device copyable, the tool will attempt to add a specialization of `sycl::is_device_copyable` for this type and will also issue a warning to the user.

From these explanations, it is clear that these warnings do not result in compilation errors but could potentially lead to runtime issues. The only errors that caused compilation failures were related to error handling, and as shown in the plot, such warnings were relatively rare.

If no additional compilation errors had occurred, only a small number of warnings would have required fixing to successfully compile the project. Any potential runtime errors could then be addressed during debugging, should they arise.

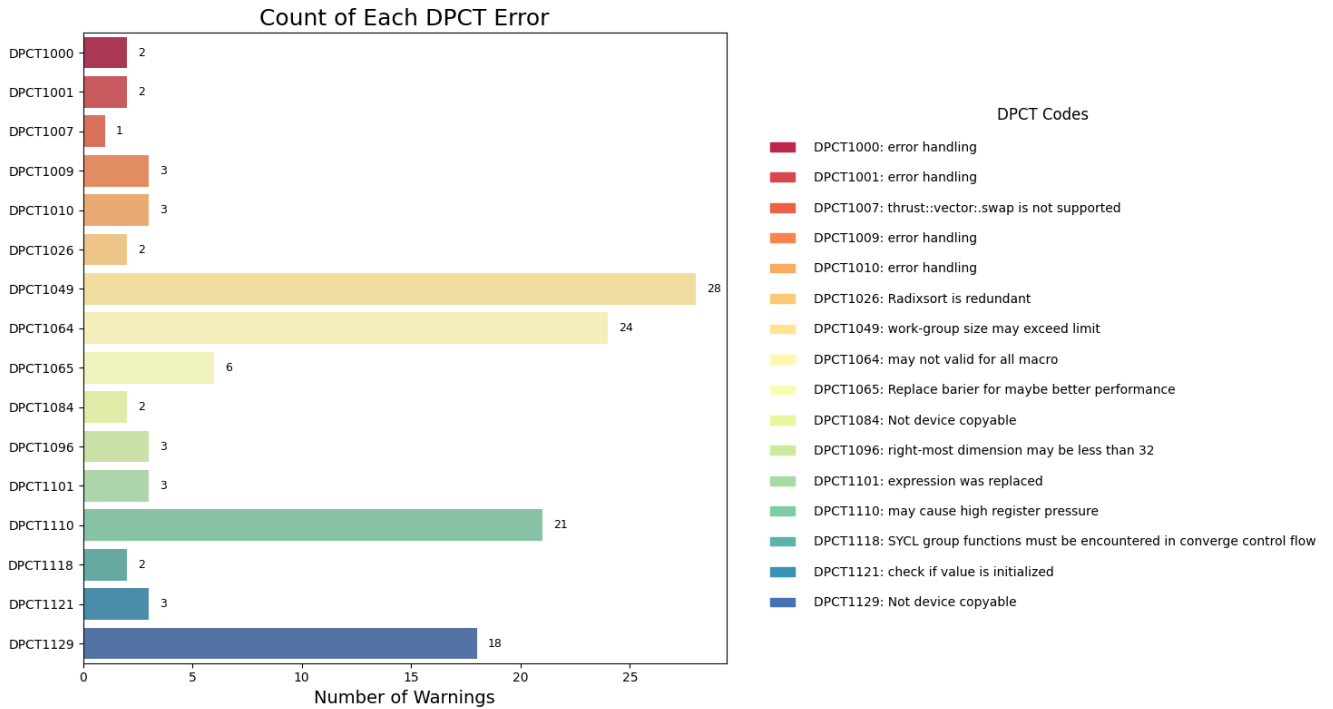


Figure 4.3: Count of the different Warnings

The last plot (Figure 4.4) shows a comparison of the number of LOC for different languages in the folders of the two projects. It reveals that the overhead in LOC from SPH-EXA to SYCL-SPH-EXA is not very significant, except for the domain folder. However, the overhead in the domain folder mainly comes from the header files. This is because each subfolder within the domain folder had to be migrated individually due to an internal DPCT error. Through this process, many helper functions were created for each subfolder, resulting in this overhead, which should normally be generated only once for the whole project.

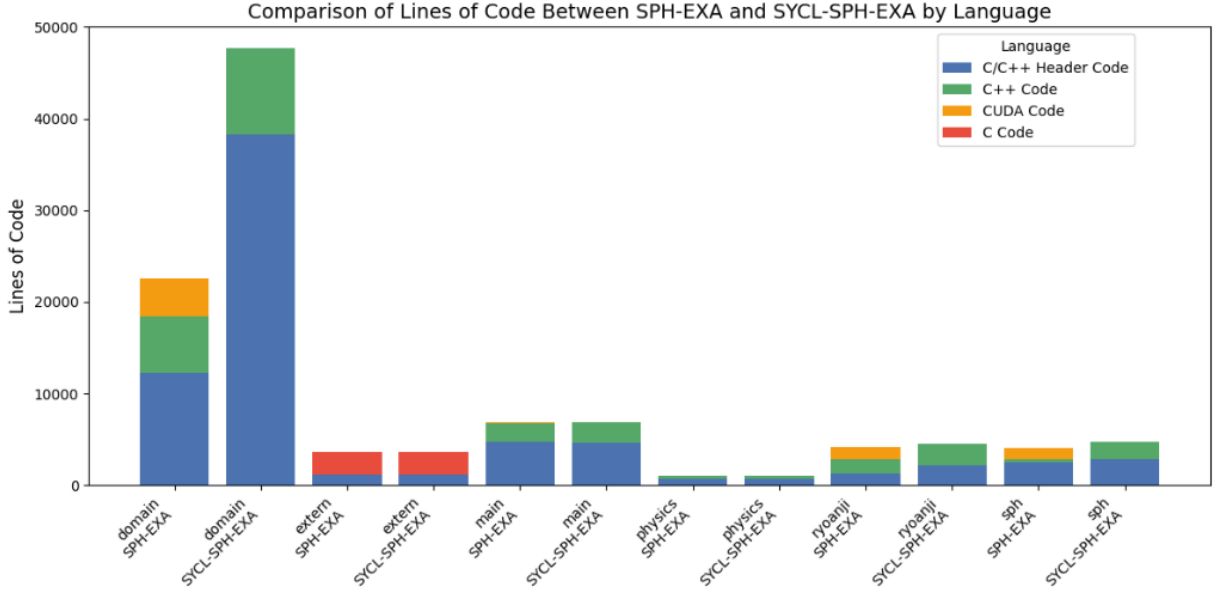


Figure 4.4: LOC Comparison between SPH-EXA and SYCL-SPH-EXA

4.1.2 Tests

Although a fully functional SYCL-SPH-EXA project is not yet available, the tests could still be performed. The results are as follows:

Type	Name	Result
CPU	Integrator.timeReversal	passed
CPU	Integrator.timeEnergyReversal	passed
CPU	SphKernelTestsStd.IAD	passed
CPU	SphKernelTestsStd.MomentumEnergy	passed
CPU	KernelTable.simpsonSine	passed
CPU	KernelTable.simpson3DK	passed
CPU	SphKernelTests.AVSwitches	passed
CPU	SphKernelTests.Divv_Curlv	passed
CPU	SphKernelTests.IAD	passed
CPU	SphKernelTests.MomentumEnergy	passed
CPU	SphKernelTests.VeDefGradh	passed
CPU	SphKernelTests.XMass	passed
GPU	TimestepGpu.Acc	passed
GPU	TimestepGPU.Divv	failed

Table 4.1: Test Results

Out of the 14 tests, 13 were successful. Although the majority of these tests are designed for the CPU, it is reassuring to see that the existing CPU code remains unchanged. Among the two GPU-specific tests, one failed. The error occurred in `TimestepGPU.Divv`

due to a discrepancy between `probe[0]` and `Krho / 119`. The absolute difference between these two values is $1.1641532182693481 \times 10^{-10}$, which exceeds the allowable threshold of 1×10^{-10} .

This accuracy issue could not be resolved and may have implications for the final computational accuracy.

4.2 Accuracy and Performance Analysis

As previously mentioned, not all runtime errors could be resolved. Consequently, it was not possible to perform an accuracy or performance analysis. It would have been particularly interesting to examine how the before mentioned difference impacts the final results. While the difference noted above is minimal for a single calculation, these calculations are repeated frequently, which could cause the difference to accumulate and significantly influence the overall outcome.

The results of a performance analysis would also have been highly valuable. It is anticipated that the SYCL version would be slower than the CUDA version. This is because CUDA is explicitly designed and optimized for NVIDIA GPUs, whereas SYCL is intended to support multiple GPU vendors and is therefore less tailored to any specific hardware. A performance analysis would have confirmed whether this assumption is correct and, if so, quantified the extent of the performance difference.

Chapter 5

Conclusion

Migrating SPH-EXA to SYCL is crucial for enhancing its portability, as SYCL enables the code to run not only on Intel GPUs but also on a variety of other accelerators. To facilitate this migration, the Intel Compatibility Tool provides support and simplifies the process, allowing the entire project to be migrated with just a few command lines.

However, this report highlights that with a large and complex project like SPH-EXA, the tool can generate unexpected errors, complicating the migration process. As a result, the claim that 95% of the code can be successfully migrated does not fully hold true in this context.

Additionally, it was found that installing OneAPI is not straightforward, as it cannot be loaded as a module and requires a direct installation. This process comes with significant dependencies on other software, which is suboptimal for a framework whose primary objective is portability.

Due to time constraints, the migration could not be fully completed. Nevertheless, I believe it is essential to continue this effort to ensure SPH-EXA can run on a wide range of accelerators and truly achieve cross-platform compatibility.

Bibliography

- [1] Intel. Intel® dpc++ compatibility tool developer guide and reference, 2020. Available at: https://cdrdv2.intel.com/v1/dl/getContent/834754?fileName=dpcpp-compatibility-tool_developer-guide-reference_2025.0-768918-834754.pdf Accessed: 01.01.2025.
- [2] Intel. Data parallel c++: the oneapi implementation of sycl*, 2024. Available at: <https://www.intel.com/content/www/us/en/developer/tools/oneapi/data-parallel-c-plus-plus.html>. Accessed: 01.01.2025.
- [3] Intel. oneapi - what is it?, 2024. Available at: <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>. Accessed: 01.01.2025.
- [4] Khronos Group. Sycl::nd_item, 2025. Available at: https://github.khronos.org/SYCL_Reference/iface/nd_item.html. Accessed: 01.01.2025.
- [5] HPC DMI University of Basel. Sph-exa: Optimizing smoothed particle hydrodynamics for exascale computing, 2021. Available at: <https://hpc.dmi.unibas.ch/research/sph-exa/> Accessed: 01.01.2025.
- [6] HPC DMI University of Basel. Sph-exa, 2025. Available at: <https://github.com/sphexa-org/sphexa> Accessed: 01.01.2025.

Appendix

A.1 List of Compilation Errors

This is a list of errors caused by the DPC++ Compatibility Tool and compiling errors. This list is not comprehensive. It includes errors that are either very common or particularly challenging to debug, often requiring significant time to resolve.

CUDA Header

In many cases, the CUDA header files in a source file were not replaced with the corresponding migrated header files. Fortunately, this is easy to resolve; simply replace `.cuh` with `.dp.hpp`.

Tuple Arguments Deleted

During migration, the DPC++ Compatibility Tool removed arguments from tuple constructions. Specifically, whenever a tuple with more than two arguments was created in the original code using `thrust::make_tuple()`, it was incorrectly migrated to `std::make_tuple()` with only two arguments. So `thrust::make_tuple(x, y, z)` became `std::make_tuple(x, y)`. To correct the error, simply add the missing arguments.

This error was difficult to find because the error message did not occur during instantiation but later for example during calculations that tried to access the missing arguments.

Function Call mismatch

A common error that frequently occurred was a function mismatch. The primary cause of this mismatch was the addition of `const sycl::nd_item<3>` in the function definition, while the corresponding argument was often omitted when calling the function.

`const sycl::nd_item<3>` is a type used in SYCL to provide information about the execution environment for kernels running on GPUs or other accelerators. It encapsulates details about the current thread in a 3-dimensional grid of threads, as well as the hierarchical structure of work-items, work-groups, and ranges.^[4]

There were also instances of function miscalls where multiple arguments were missing, or in some cases, no arguments were passed to the function at all.

Global Variables

There were multiple problems related to global variables.

- The first issue involved static variables stored in global memory, intended for use by multiple workers. These variables could not be referenced directly; access was only possible by first creating pointers to their addresses.
- However, this did not completely resolve the issue, as it was not possible to create the pointers within a kernel. Instead, the pointers had to be created outside the kernel and passed as function arguments, necessitating the rewriting of some functions.

The issue arises from the SYCL memory model, which requires explicit memory management for heterogeneous devices. In SYCL, all data used by a kernel must be explicitly passed through memory objects. Direct access to global variables is not allowed. Unlike CUDA, where device variables can be accessed directly by kernels, SYCL enforces a clear separation between host and device memory, preventing such direct access.

Proper handling is demonstrated in the following example from SYCL-SPH-EXA:

Listing A.1: Accessing Global Variables

```
//define and initialize variable in a file
static dpct::global_memory<unsigned, 0> targetCounterGlob;

...
//create a pointer of the global variable
unsigned* targetCounterGlob_ptr =
    estone::targetCounterGlob.get_ptr();
q_ct1.parallel_for(sycl::nd_range<3>(sycl::range<3>(1, 1, 1),
    sycl::range<3>(1, 1, 1)), [=](sycl::nd_item<3> item_ct1)
    {
        //and give it to the functions as an argument
        estone::resetTraversalCounters(ncStats_ptr,
            *targetCounterGlob_ptr);
    });
```

- The required change from directly accessing variables to referencing them via pointers may cause some functions to behave unexpectedly. This occurs when the functions do not accept pointers as arguments, necessitating the use of alternative functions to address the issue.

This issue occurred with the `atomic_fetch_add()` function. The problem was resolved by first creating an atomic reference and then performing the atomic operation. Below, you can see the previous implementation and how the issue was addressed:

Listing A.2: Atomic Add Function

```

//before:
targetIdx = atomic_fetch_add<unsigned,
    sycl::access::address_space::global_space>
    (targetCounterGlob_ptr, 1, sycl::memory_order::relaxed);

//fixed:
sycl::atomic_ref<unsigned, sycl::memory_order::relaxed,
    sycl::memory_scope::device,
    sycl::access::address_space::generic_space>
    atomicCounter(*targetCounterGlob_ptr);

targetIdx = atomicCounter.fetch_add(1);

```

Shared Memory

This error also falls into the category of "function call mismatches". However, because it is related to memory management, it has been given its own category.

In the traverse function, defined in `traversal.dp.hpp` within the `ryoanji` folder, a buffer called `sharedPool` is used. This buffer serves as a storage space for shared memory, which allows threads within the same workgroup or thread block to communicate and share data during computation. Shared memory is significantly faster than global memory and therefore used in this function.

However, this buffer was not created by the DPC++ Compatibility Tool during the migration process, resulting in its absence in the function calls. In SYCL, shared memory is implemented differently compared to CUDA, requiring the use of a local accessor. For instance, shared memory in SYCL can be declared as follows:

Listing A.3: Creating Shared Memory

```

sycl::local_accessor<int, 1> sharedPool_acc_ct1(
    sycl::range<1>(smSize * mSizeRatio), cgh);

```

To resolve the compilation error, the `sharedPool_acc_ct1` object was added to the function call. This addition ensured proper memory allocation and resolved the issue effectively.

Deleted DPCT Warning

During the migration process, the tool reported an error in the `errorcheck` file located in the `domain` folder. The warning, identified as `DPCT1009`, states: "SYCL uses exceptions to report errors, not error codes. The call was replaced with a placeholder string. This code needs to be rewritten."

Typically, such warnings are added as comments in the code. However, in this case, both the comment and the placeholder string were not included. Additionally, the original code within the function was removed, requiring the function to be migrated manually. The reason for this is unclear.

A.2 Parallel Programming Models

FUNCTIONALITY



		OMP	CUDA	HIP	SYCL
domain	cuda		☒	☒	
	domain	☒	☒	☒	
	fields				
	focus	☒	☒	☒	
	halos	☒	☒	☒	
	primitives	☒	☒	☒	
	sfc	☒	☒	☒	
	traversal	☒	☒	☒	
	tree	☒	☒	☒	
	util		☒	☒	
extern	grackle				
	h5part				
main	init	☒			
	io				
	observables	☒	☒	☒	
	propagator				
	sphexa	☒			
	util	☒			
physics	cooling	☒			
ryoanji	interface		☒	☒	
	nbody	☒	☒	☒	
sph	hydro_std	☒	☒	☒	
	hydro_turb	☒	☒	☒	
	hydro_ve	☒	☒	☒	
	sph	☒	☒	☒	
	util		☒	☒	

A.3 Lines of Code

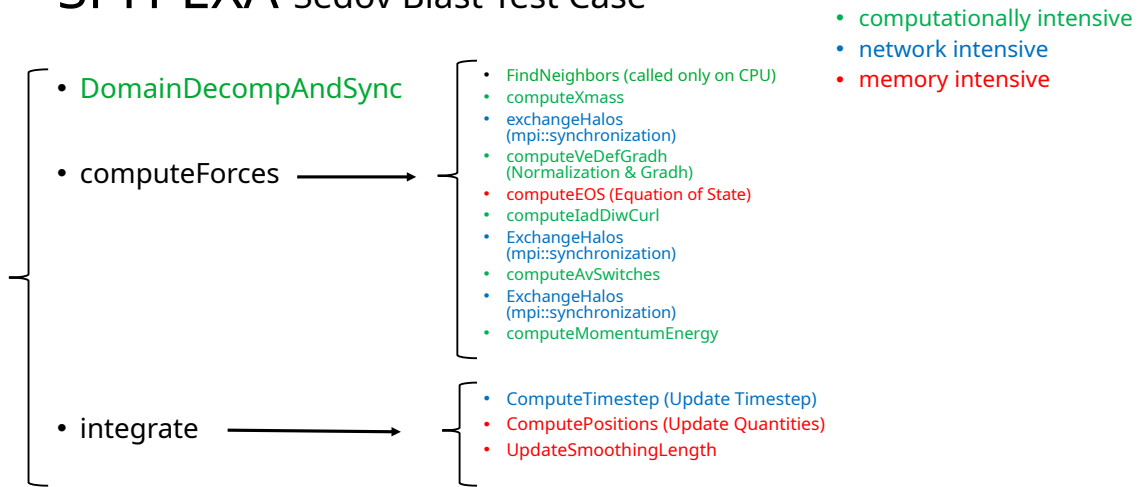


Lines of Code

		C++	C	C/C++ Header	CUDA/ (HIP)	C++ (SYCL V.)	C/C++ Header (SYCL V.)
domain	cuda	0	0	107	239	153	24035
	domain	0	0	1349	244	0	25435
	fields	0	0	290	0	0	24904
	focus	0	0	1416	321	472	25239
	halos	0	0	266	180	108	24186
	primitives	0	0	591	564	296	24752
	sfc	0	0	951	40	63	24781
	traversal	0	0	592	654	121	24964
	tree	0	0	1130	344	432	25042
	util	0	0	4854	28	0	28694
extern	grackle	0	70	0	0	0	70
	h5part	1121	2433	0	0	1121	2433
main	init	103	0	1733	0	103	1683
	io	313	0	534	0	313	534
	observables	44	0	600	164	215	602
	propagator	0	0	1265	0	0	1265
	sphexa	172	0	34	0	172	34
	util	0	0	195	0	0	195
physics	cooling	360	0	653	0	360	653
ryoanji	interface	0	0	29	493	593	101
	nbody	0	0	1238	628	0	2022
sph	hydro_std	0	0	308	162	301	314
	hydro_turb	0	0	390	45	59	396
	hydro_ve	0	0	740	427	754	754
	sph	0	0	1071	463	331	1299
	util	0	0	0	72	0	82

A.4 Code Structure SPH-EXA Sedov Blast Test Case

SPH-EXA Sedov Blast Test Case



A.5 List of all warnings

DPCT Nr.	Warning
DPCT1110	/home/yannu/Desktop/SPH-EXA/ryojanji/src/ryojanji/nbody/ewald.hpp:226:1: warning: DPCT1110:0: The total declared local variable size in device function computeEwaldRealSpace exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1129	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/ve_def_gradh_gpu.cu:91:5: warning: DPCT1129:1: The type "const cstone::Box<typename Dataset::RealType>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1110	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/av_switches_kern.hpp:44:1: warning: DPCT1110:2: The total declared local variable size in device function AVswitchesJLoop exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1129	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/av_switches_gpu.cu:92:5: warning: DPCT1129:3: The type "const cstone::Box<typename Dataset::RealType>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1129	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_std/iad_gpu.cu:117:5: warning: DPCT1129:4: The type "const cstone::Box<typename Dataset::RealType>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1049	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/eos_gpu.cu:68:5: warning: DPCT1049:5: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1049	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_turb/stirring_gpu.cu:73:5: warning: DPCT1049:6: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1049	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_std/eos_gpu.cu:61:5: warning: DPCT1049:7: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1110	/home/yannu/Desktop/SPH-EXA/ryojanji/src/ryojanji/nbody/traversal.cuh:199:1: warning: DPCT1110:8: The total declared local variable size in device function traverseWarp exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1110	/home/yannu/Desktop/SPH-EXA/ryojanji/src/ryojanji/nbody/traversal.cuh:412:1: warning: DPCT1110:9: The total declared local variable size in device function traverse exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1118	/home/yannu/Desktop/SPH-EXA/ryojanji/src/ryojanji/nbody/direct.cuh:80:9: warning: DPCT1118:10: SYCL group functions and algorithms must be encountered in converged control flow. You may need to adjust the code.
DPCT1118	/home/yannu/Desktop/SPH-EXA/ryojanji/src/ryojanji/nbody/direct.cuh:91:9: warning: DPCT1118:11: SYCL group functions and algorithms must be encountered in converged control flow. You may need to adjust the code.
DPCT1049	/home/yannu/Desktop/SPH-EXA/ryojanji/src/ryojanji/nbody/direct.cuh:118:17: warning: DPCT1049:12: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1110	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/momentum_energy_kern.hpp:66:1: warning: DPCT1110:13: The total declared local variable size in device function momentumAndEnergyJLoop exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1129	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/momentum_energy_gpu.cu:130:5: warning: DPCT1129:14: The type "const cstone::Box<typename Dataset::RealType>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1110	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_std/momentum_energy_kern.hpp:131:1: warning: DPCT1110:15: The total declared local variable size in device function momentumAndEnergyJLoop exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1129	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_std/momentum_energy_gpu.cu:115:5: warning: DPCT1129:16: The type "const cstone::Box<typename Dataset::RealType>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1049	/home/yannu/Desktop/SPH-EXA/sph/include/sph/ts_groups.cu:44:5: warning: DPCT1049:17: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1049	/home/yannu/Desktop/SPH-EXA/sph/include/sph/ts_groups.cu:78:5: warning: DPCT1049:18: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.

DPCT1049	/home/yannu/Desktop/SPH-EXA/sph/include/sph/ts_groups.cu:102:5: warning: DPCT1049:19: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1129	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/xmass_gpu.cu:109:5: warning: DPCT1129:20: The type "const cstone::Box<typename Dataset::RealType>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1049	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/xmass_gpu.cu:162:5: warning: DPCT1049:21: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1110	/home/yannu/Desktop/SPH-EXA/ryoanji/src/ryoanji/nbody/kernel.hpp:460:1: warning: DPCT1110:22: The total declared local variable size in device function P2M exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1110	/home/yannu/Desktop/SPH-EXA/ryoanji/src/ryoanji/nbody/kernel.hpp:621:1: warning: DPCT1110:23: The total declared local variable size in device function normalize exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1129	/home/yannu/Desktop/SPH-EXA/ryoanji/src/ryoanji/interface/multipole_holder.cu:151:9: warning: DPCT1129:24: The type "Vec3<Tc>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1049	/home/yannu/Desktop/SPH-EXA/sph/include/sph/update_h_gpu.cu:59:5: warning: DPCT1049:25: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1110	/home/yannu/Desktop/SPH-EXA/ryoanji/src/ryoanji/nbody/ewald.hpp:107:1: warning: DPCT1110:26: The total declared local variable size in device function ewaldEvalMultipoleComplete exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1049	/home/yannu/Desktop/SPH-EXA/ryoanji/src/ryoanji/interface/ewald.cu:84:5: warning: DPCT1049:27: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1110	/home/yannu/Desktop/SPH-EXA/sph/include/sph/positions_gpu.cu:47:1: warning: DPCT1110:28: The total declared local variable size in device function driftKernel exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1049	/home/yannu/Desktop/SPH-EXA/sph/include/sph/positions_gpu.cu:102:5: warning: DPCT1049:29: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1110	/home/yannu/Desktop/SPH-EXA/sph/include/sph/positions_gpu.cu:119:1: warning: DPCT1110:30: The total declared local variable size in device function computePositionsKernel exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1129	/home/yannu/Desktop/SPH-EXA/sph/include/sph/positions_gpu.cu:177:5: warning: DPCT1129:31: The type "const cstone::Box<Tc>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1049	/home/yannu/Desktop/SPH-EXA/sph/include/sph/positions_gpu.cu:177:5: warning: DPCT1049:32: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1110	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/divv_curlv_kern.hpp:44:1: warning: DPCT1110:33: The total declared local variable size in device function divV_curlVLoop exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1129	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/iad_divv_curlv_gpu.cu:99:5: warning: DPCT1129:34: The type "const cstone::Box<typename Dataset::RealType>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1001	/home/yannu/Desktop/SPH-EXA/sph/include/sph/util/pinned_allocator.cuh:130:22: warning: DPCT1001:35: The statement could not be removed.
DPCT1000	/home/yannu/Desktop/SPH-EXA/sph/include/sph/util/pinned_allocator.cuh:130:9: warning: DPCT1000:36: Error handling if-stmt was detected but could not be rewritten.
DPCT1001	/home/yannu/Desktop/SPH-EXA/sph/include/sph/util/pinned_allocator.cuh:149:22: warning: DPCT1001:37: The statement could not be removed.
DPCT1000	/home/yannu/Desktop/SPH-EXA/sph/include/sph/util/pinned_allocator.cuh:149:9: warning: DPCT1000:38: Error handling if-stmt was detected but could not be rewritten.
DPCT1064	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/av_switches_kern.hpp:123:58: warning: DPCT1064:39: Migrated std::abs call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.

DPCT1065	/home/yannu/Desktop/SPH-EXA/ryoanji/src/ryoanji/nbody/direct.cuh:80:9: warning: DPCT1065:40: Consider replacing <code>ycl::nd_item::barrier()</code> with <code>ycl::nd_item::barrier(ycl::access::fence_space::local_space)</code> for better performance if there is no access to global memory.
DPCT1065	/home/yannu/Desktop/SPH-EXA/ryoanji/src/ryoanji/nbody/direct.cuh:91:9: warning: DPCT1065:41: Consider replacing <code>ycl::nd_item::barrier()</code> with <code>ycl::nd_item::barrier(ycl::access::fence_space::local_space)</code> for better performance if there is no access to global memory.
DPCT1010	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/momentum_energy_gpu.cu:140:20: warning: DPCT1010:42: SYCL uses exceptions to report errors and does not use the error codes. The call was replaced with 0. You need to rewrite this code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/momentum_energy_kern.hpp:179:21: warning: DPCT1064:43: Migrated <code>std::abs</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/momentum_energy_kern.hpp:193:26: warning: DPCT1064:44: Migrated <code>pow</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/momentum_energy_kern.hpp:194:26: warning: DPCT1064:45: Migrated <code>pow</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1065	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_ve/momentum_energy_gpu.cu:115:5: warning: DPCT1065:46: Consider replacing <code>ycl::nd_item::barrier()</code> with <code>ycl::nd_item::barrier(ycl::access::fence_space::local_space)</code> for better performance if there is no access to global memory.
DPCT1064	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_turb/create_modes.hpp:134:41: warning: DPCT1064:47: Migrated <code>std::abs</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_turb/create_modes.hpp:190:47: warning: DPCT1064:48: Migrated <code>ceil</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/sph/include/sph/positions.hpp:50:20: warning: DPCT1064:49: Migrated <code>std::abs</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/sph/include/sph/ts_global.hpp:93:21: warning: DPCT1064:50: Migrated <code>std::abs</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1010	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_std/momentum_energy_gpu.cu:123:20: warning: DPCT1010:51: SYCL uses exceptions to report errors and does not use the error codes. The call was replaced with 0. You need to rewrite this code.
DPCT1065	/home/yannu/Desktop/SPH-EXA/sph/include/sph/hydro_std/momentum_energy_gpu.cu:100:5: warning: DPCT1065:52: Consider replacing <code>ycl::nd_item::barrier()</code> with <code>ycl::nd_item::barrier(ycl::access::fence_space::local_space)</code> for better performance if there is no access to global memory.
DPCT1064	/home/yannu/Desktop/SPH-EXA/main/src/init/isobaric_cube_init.hpp:98:16: warning: DPCT1064:53: Migrated <code>std::abs</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/main/src/init/isobaric_cube_init.hpp:99:16: warning: DPCT1064:54: Migrated <code>std::abs</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/main/src/init/isobaric_cube_init.hpp:100:16: warning: DPCT1064:55: Migrated <code>std::abs</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/main/src/init/isobaric_cube_init.hpp:136:14: warning: DPCT1064:56: Migrated <code>std::abs</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/ryoanji/src/ryoanji/nbody/upwardpass.cuh:174:20: warning: DPCT1064:57: Migrated <code>max</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/ryoanji/src/ryoanji/nbody/traversal.cuh:521:32: warning: DPCT1064:58: Migrated <code>max</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/ryoanji/src/ryoanji/nbody/ewald.hpp:294:40: warning: DPCT1064:59: Migrated <code>sqrt</code> call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1065	/home/yannu/Desktop/SPH-EXA/ryoanji/src/ryoanji/interface/ewald.cu:55:5: warning: DPCT1065:60: Consider replacing <code>ycl::nd_item::barrier()</code> with <code>ycl::nd_item::barrier(ycl::access::fence_space::local_space)</code> for better performance if there is no access to global memory.

DPCT1064	/home/yannu/Desktop/SPH-EXA/main/src/observables/time_energy_growth.hpp:59:63: warning: DPCT1064:61: Migrated std::abs call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/main/src/observables/time_energy_growth.hpp:60:45: warning: DPCT1064:62: Migrated std::abs call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1101	/home/yannu/Desktop/SPH-EXA/ryoanji/src/ryoanji/nbody/direct.cuh:70:34: warning: DPCT1101:64: 'DirectConfig::numThreads' expression was replaced with a value. Modify the code to use the original expression, provided in comments, if it is correct.
DPCT1110	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/traversal.hpp:69:1: warning: DPCT1110:0: The total declared local variable size in device function singleTraversal exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1121	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/primitives/warpscan.cuh:77:12: warning: DPCT1121:1: Make sure that the "value" which is used in the SYCL group function/algorithm is initialized.
DPCT1121	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/primitives/warpscan.cuh:99:12: warning: DPCT1121:2: Make sure that the "value" which is used in the SYCL group function/algorithm is initialized.
DPCT1129	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/groups.cuh:293:5: warning: DPCT1129:3: The type "const Box<Tc>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/groups.cuh:293:5: warning: DPCT1049:4: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/groups.cuh:306:5: warning: DPCT1049:5: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1121	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/primitives/warpscan.cuh:110:12: warning: DPCT1121:6: Make sure that the "value" which is used in the SYCL group function/algorithm is initialized.
DPCT1110	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/find_neighbors.cuh:201:1: warning: DPCT1110:7: The total declared local variable size in device function traverseWarp exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1110	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/find_neighbors.cuh:392:1: warning: DPCT1110:8: The total declared local variable size in device function warpBbox exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1110	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/find_neighbors.cuh:436:1: warning: DPCT1110:9: The total declared local variable size in device function traverseNeighbors exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1110	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/groups.cuh:86:1: warning: DPCT1110:10: The total declared local variable size in device function findSplits exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1110	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/groups.cuh:197:1: warning: DPCT1110:11: The total declared local variable size in device function groupSplitsKernel exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/test/unit_cuda/traversal/groups.cu:259:9: warning: DPCT1049:12: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/test/unit_cuda/traversal/groups.cu:269:9: warning: DPCT1049:13: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/halos/gather_halos_gpu.cu:69:5: warning: DPCT1049:14: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/halos/gather_halos_gpu.cu:121:5: warning: DPCT1049:15: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/tree/csarray_gpu.cu:196:5: warning: DPCT1049:16: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/tree/csarray_gpu.cu:218:5: warning: DPCT1049:17: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.

DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/tree/csarray_gpu.cu:241:5: warning: DPCT1049:18: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1110	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/macros.hpp:82:1: warning: DPCT1110:19: The total declared local variable size in device function computeVecMacR2 exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/focus/source_center_gpu.cu:68:5: warning: DPCT1049:20: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1129	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/focus/source_center_gpu.cu:136:5: warning: DPCT1129:21: The type "const Box<T>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/focus/source_center_gpu.cu:136:5: warning: DPCT1049:22: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1129	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/focus/source_center_gpu.cu:163:5: warning: DPCT1129:23: The type "const Box<T>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/focus/source_center_gpu.cu:163:5: warning: DPCT1049:24: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1129	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/focus/source_center_gpu.cu:191:5: warning: DPCT1129:25: The type "const Box<T>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/focus/source_center_gpu.cu:191:5: warning: DPCT1049:26: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/focus/source_center_gpu.cu:219:5: warning: DPCT1049:27: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1026	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/primitives/primitives_gpu.cu:268:5: warning: DPCT1026:28: The call to cub::DeviceRadixSort::SortKeys was removed because this functionality is redundant in SYCL.
DPCT1026	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/primitives/primitives_gpu.cu:300:5: warning: DPCT1026:29: The call to cub::DeviceRadixSort::SortPairs was removed because this functionality is redundant in SYCL.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/primitives/primitives_gpu.cu:122:5: warning: DPCT1049:30: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1049	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/primitives/primitives_gpu.cu:230:5: warning: DPCT1049:31: The work-group size passed to the SYCL kernel may exceed the limit. To get the device limit, query info::device::max_work_group_size. Adjust the work-group size if needed.
DPCT1129	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/collisions_gpu.cu:102:5: warning: DPCT1129:32: The type "const Box<T>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1110	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/collisions_gpu.cu:118:1: warning: DPCT1110:33: The total declared local variable size in device function markMacsgpuKernel exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1129	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/collisions_gpu.cu:159:5: warning: DPCT1129:34: The type "const Box<T>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1129	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/sfc/sfc_gpu.cu:55:5: warning: DPCT1129:35: The type "const Box<T>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1110	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/findneighbors.hpp:96:1: warning: DPCT1110:36: The total declared local variable size in device function findNeighbors exceeds 128 bytes and may cause high register pressure. Consult with your hardware vendor to find the total register size available and adjust the code, or use smaller sub-group size to avoid high register pressure.
DPCT1129	/home/yannu/Desktop/SPH-EXA/domain/test/performance/neighbor_driver.cu:152:5: warning: DPCT1129:37: The type "OctreeNsView<Tc, KeyType>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.
DPCT1129	/home/yannu/Desktop/SPH-EXA/domain/test/performance/neighbor_driver.cu:152:5: warning: DPCT1129:38: The type "const Box<Tc>" is used in the SYCL kernel, but it is not device copyable. The sycl::is_device_copyable specialization has been added for this type. Please review the code.

DPCT1084	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/boxoverlap.hpp:111:84: warning: DPCT1084:39: The function call "isfcKey" has multiple migration results in different template instantiations that could not be unified. You may need to adjust the code.
DPCT1084	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/boxoverlap.hpp:112:96: warning: DPCT1084:40: The function call "isfcKey" has multiple migration results in different template instantiations that could not be unified. You may need to adjust the code.
DPCT1009	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/cuda/errorcheck.cuh:34:31: warning: DPCT1009:41: SYCL uses exceptions to report errors and does not use the error codes. The call was replaced by a placeholder string. You need to rewrite this code.
DPCT1009	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/cuda/errorcheck.cuh:35:31: warning: DPCT1009:42: SYCL uses exceptions to report errors and does not use the error codes. The call was replaced by a placeholder string. You need to rewrite this code.
DPCT1009	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/cuda/errorcheck.cuh:49:59: warning: DPCT1009:43: SYCL uses exceptions to report errors and does not use the error codes. The call was replaced by a placeholder string. You need to rewrite this code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/domain/test/unit/../../include/cstone/util/array.hpp:261:13: warning: DPCT1064:44: Migrated std::abs call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/domain/test/unit/../../include/cstone/traversal/macros.hpp:140:17: warning: DPCT1064:45: Migrated std::abs call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1007	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/util/reallocate.hpp:57:9: warning: DPCT1007:46: Migration of thrust::device_vector::swap is not supported.
DPCT1064	/home/yannu/Desktop/SPH-EXA/domain/test/unit/../../include/cstone/primitives/gather.hpp:181:38: warning: DPCT1064:47: Migrated std::abs call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/domain/test/unit/../../include/cstone/traversal/macros.hpp:115:17: warning: DPCT1064:48: Migrated std::abs call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/primitives/gather.cuh:135:38: warning: DPCT1064:49: Migrated std::abs call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/halos/exchange_halos_gpu.cuh:71:20: warning: DPCT1064:50: Migrated cudaMalloc call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1064	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/groups.cuh:239:22: warning: DPCT1064:51: Migrated min call is used in a macro/template definition and may not be valid for all macro/template uses. Adjust the code.
DPCT1010	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/sfc/sfc_gpu.cu:56:20: warning: DPCT1010:52: SYCL uses exceptions to report errors and does not use the error codes. The call was replaced with 0. You need to rewrite this code.
DPCT1065	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/focus/rebalance_gpu.cu:145:5: warning: DPCT1065:53: Consider replacing <code>ycl::nd_item::barrier()</code> with <code>ycl::nd_item::barrier(ycl::access::fence_space::local_space)</code> for better performance if there is no access to global memory.
DPCT1101	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/traversal/find_neighbors.cuh:453:20: warning: DPCT1101:54: 'TravConfig::numThreads' expression was replaced with a value. Modify the code to use the original expression, provided in comments, if it is correct.
DPCT1096	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/primitives/warpscan.cuh:88:12: warning: DPCT1096:55: The right-most dimension of the work-group used in the SYCL kernel that calls this function may be less than "32". The function "dpct::permute_sub_group_by_xor" may return an unexpected result on the CPU device. Modify the size of the work-group to ensure that the value of the right-most dimension is a multiple of "32".
DPCT1096	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/primitives/warpscan.cuh:110:12: warning: DPCT1096:56: The right-most dimension of the work-group used in the SYCL kernel that calls this function may be less than "32". The function "dpct::shift_sub_group_left" may return an unexpected result on the CPU device. Modify the size of the work-group to ensure that the value of the right-most dimension is a multiple of "32".
DPCT1096	/home/yannu/Desktop/SPH-EXA/domain/include/cstone/primitives/warpscan.cuh:77:12: warning: DPCT1096:57: The right-most dimension of the work-group used in the SYCL kernel that calls this function may be less than "32". The function "dpct::select_from_sub_group" may return an unexpected result on the CPU device. Modify the size of the work-group to ensure that the value of the right-most dimension is a multiple of "32".
DPCT1101	/home/yannu/Desktop/SPH-EXA/domain/test/unit_cuda/primitives/warpscan.cu:155:20: warning: DPCT1101:58: 'GpuConfig::warpSize' expression was replaced with a value. Modify the code to use the original expression, provided in comments, if it is correct.

Declaration on Scientific Integrity



University
of Basel

Faculty of Science



Declaration on Scientific Integrity

(including a Declaration on Plagiarism and Fraud)
Translation from German original

Title of Thesis: Porting and Measuring the Performance of SPH-EXA
on SYCL

Name Assessor: Prof. F. Ciorba

Name Student: Yannick Martin

Matriculation No.: 2020-050-753

I attest with my signature that I have written this work independently and without outside help. I also attest that the information concerning the sources used in this work is true and complete in every respect. All sources that have been quoted or paraphrased have been marked accordingly.

Additionally, I affirm that any text passages written with the help of AI-supported technology are marked as such, including a reference to the AI-supported program used. This paper may be checked for plagiarism and use of AI-supported technology using the appropriate software. I understand that unethical conduct may lead to a grade of 1 or "fail" or expulsion from the study program.

Place, Date: Basel, 13.01.2025 Student: Y. Martin

Will this work, or parts of it, be published?

No

Yes. With my signature I confirm that I agree to a publication of the work (print/digital) in the library, on the research database of the University of Basel and/or on the document server of the department. Likewise, I agree to the bibliographic reference in the catalog SLSP (Swiss Library Service Platform). (cross out as applicable)

Publication as of: 17 January 2025

Place, Date: Basel, 13.01.2025 Student: Y. Martin

Place, Date: Basel, 13.01.2025 Assessor: F. Ciorba

Please enclose a completed and signed copy of this declaration in your Bachelor's or Master's thesis.

September 2023