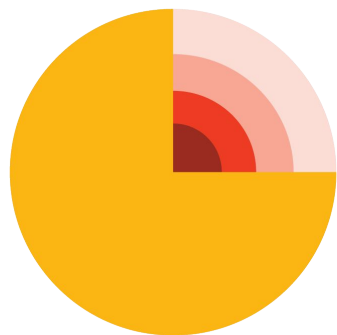




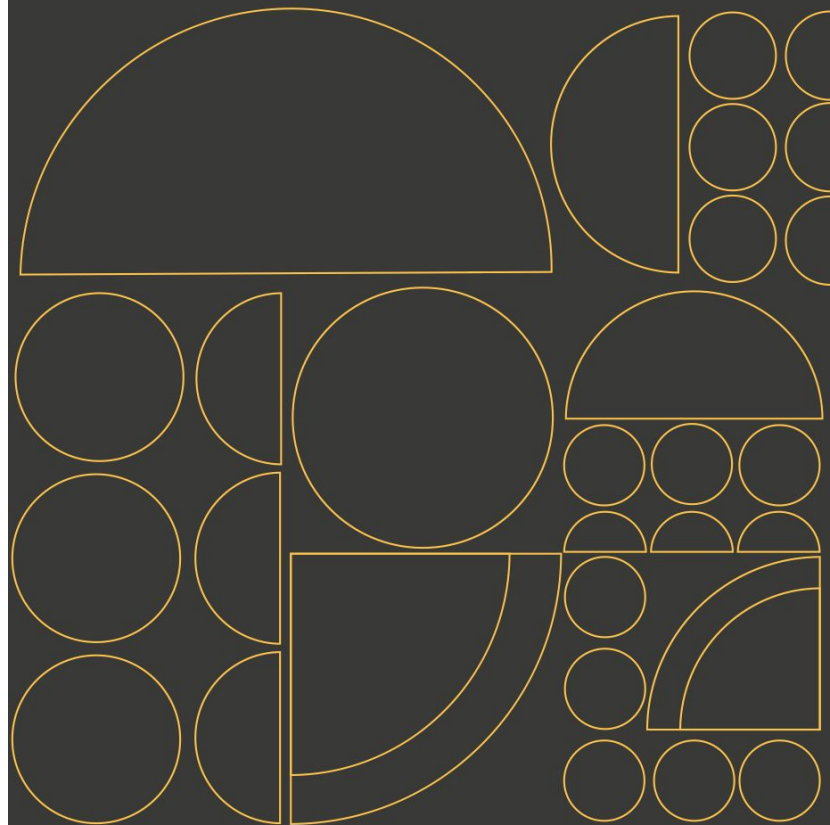
ChEESE

HySEA

Jorge Macías (UMA)



Project funded by EuroHPC under the grant agreement No 101093038.



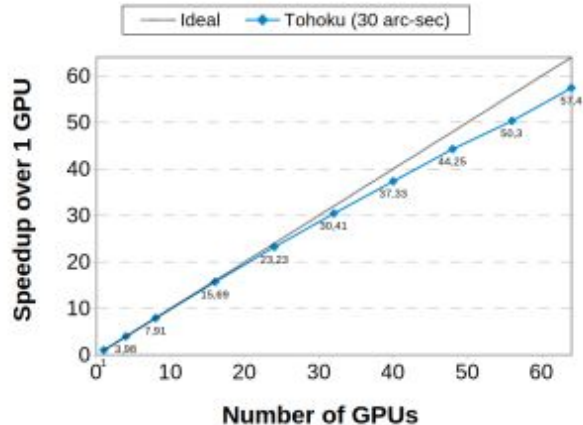
Contributions to WP2. Porting

- The Tsunami-HySEA mini-app works on one GPU and has reduced functionalities.
- CUDA version: <https://github.com/edanya-uma/TsunamiHySEA>
- SYCL version: <https://github.com/edanya-uma/TsunamiHySEA-SYCL>
- A new version has been recently developed adding more features.
- The latest mini-app CUDA version has been integrated into TOAST (Tsunami Observation And Simulation Terminal, <https://www.gempa.de/products/toast>)
- Runtimes in seconds (tsunami in the Mediterranean Sea, 30 arc-sec):

	NVIDIA H100	Intel GPU Max 1550	Intel GPU Max 1100
CUDA	74.4	-	-
SYCL	76.6	53.1	119.0

Contributions to WP2. Optimization

- Test case: Tohoku, 30 arc-sec, 336.84 millions of cells.
- Most relevant observations from the audit:
 - The scaling of Tsunami-HySEA is good (**0.897** using 64 NVIDIA A100 on Leonardo).
 - Main bottleneck: A few CUDA kernels with low load balancing numbers may be considered for improvement.
- The audit-driven development task is ongoing.



Contributions to WP3. Status of the mini-app

Code	Validation	x86 profiling status	HBM study	ARM porting	ARM profiling	SVE study	Optimization status + comments
HySEA	Done	Done	Not tested	Not tested	Not tested	Not tested	

Legend
Not tested
Fetch the code
Compile code
Run test cases
Validate results

Environment

- x86 : Intel OneAPI dpc++ 2024.2.1

Compilation

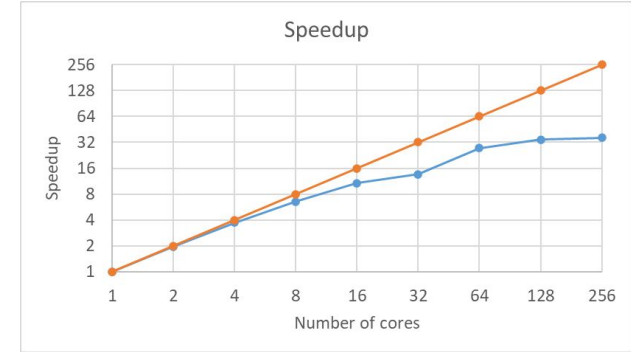
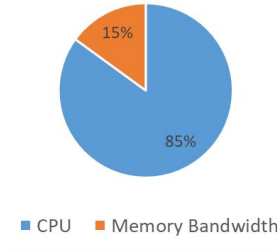
- Fix ".Po" files with hard coded Intel TBB include paths
- Adjust Makefile SYCL flags "-fsycl -fsycl-targets=nvidia_gpu_sm_86" to "-fsycl -nocudalib" to compile for CPU

Run

- Target CPU with ONEAPI_DEVICE_SELECTOR="opencl:cpu"
- Execution on Intel Emerald Rapids 8592, 64 cores per socket, 2 sockets and 2 threads per core.
- Scale number of cores `$>taskset -c 0-$nbcores ./TsunamiHySEA ./mediterranean.txt`
- NB: Hyperthreads are used for the 256 cores run.

Contributions to WP3. Status of the mini-app

- Strong dependency on the CPU (85%)
- HBM shall not provide a large speedup in this case



- The 2 hotspots show low utilization the AVX2/AVX512 vectors
- Mostly scalar FP operations are performed



Top Loops/Functions with FPU Usage by CPU Time

This section provides information for the most time consuming loops/functions with floating point operations.

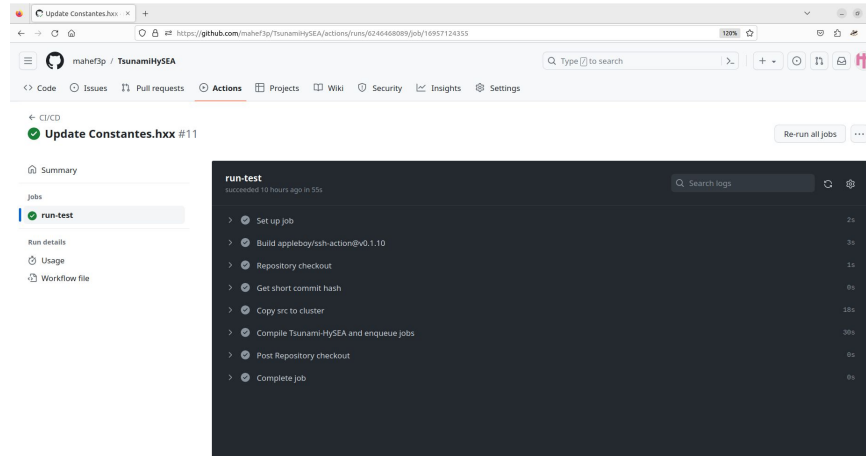
Function	CPU Time	% of FP Ops	FP Ops: Packed	FP Ops: Scalar	Vector Instruction Set	Loop Type
Intel::OpenCL::DeviceBackend::Kernel::RunGroup	36.830s	0.7%	0.0%	100.0%	SSE(128)	
procesarAristasHorNivelOPaso2GPU	27.220s	4.3%	6.8%	93.2%	AVX(128); AVX512DQ_128(128); AVX512F_128(128)	
procesarAristasHorNivelOPaso2GPU	26.830s	4.8%	7.3%	92.7%	AVX(128); AVX512DQ_128(128); AVX512F_128(128)	
__opencl_emutls_get_address	26.715s	7.8%	0.1%	99.9%		
procesarAristasHorNivelOPaso1GPU	21.100s	0.8%	26.6%	73.4%	AVX(128); AVX512DQ_128(128); AVX512F_128(128)	
[Others]	221.620s	13.5%	14.4%	85.6%		

Conclusion

- Mini-Tsunami-HySEA presents a low dependency on the memory bandwidth
- Efficient AVX instructions are not generated by the compiler

Other relevant technical contributions. CI/CD

- A **EuroHPC project** was carried out on **Vega** to implement a basic CI/CD system.
- When a commit is performed on the repository, the code is copied to the cluster, compiled, and some test jobs are enqueued.
- GitHub workflow has been implemented in YAML using GitHub Actions.
- The CI/CD workflow can be extended to create and upload a container image.





Other relevant information

Pilot Demonstrators:

- PD3. Global tsunami hazard and uncertainty quantification
- PD4. Complex multi-source tsunami modelling

Associated Simulation Cases (SCs):

- SC3.1. The Global Tsunami Hazard Map
- SC4.1. Multi-source local high-resolution scenario with emphasis on the 2022 Tonga case

New Features (we are working on):

- One-layer dispersive/non-hydrostatic version
- Meteo-HySEA (include atmospheric forcing)



New Features

- **One-layer dispersive/non-hydrostatic** version of the standard **Tsunami-HySEA** code
 - A **new formulation** of a one-layer dispersive/non-hydrostatic model in **spherical coordinates**, which includes new terms that are omitted in standard formulations.
 - A **new** robust second-order **numerical scheme** in space and time based on the use of staggered meshes, where the non-hydrostatic pressure, density, and velocity variables are arranged in an Arakawa C-grid type configuration.
 - Solving linear systems based on an iterative Jacobi algorithm, which has been combined with **new strategies** to accelerate its convergence.
 - **Adaptation** of some standard **breaking models** to the new formulation and numerical scheme.
 - **Validation** with examples in the literature have been performed.
 - **Multi-GPU implementation** is also performed, using all the improvements learned from Tsunami-HySEA audits.
- **Meteo-HySEA** (include atmospheric forcing)
 - A **new formulation** in **spherical coordinates** is performed.
 - New reconstruction operators have been developed to minimize the numerical diffusion and they are well-adapted to capture **geostrophic equilibria**.
 - Implementation of single GPU is performed.
 - A preliminary **validation** is carried out using the examples in literature (EGU).
 - A multi-GPU implementation is ongoing work.



Other relevant information

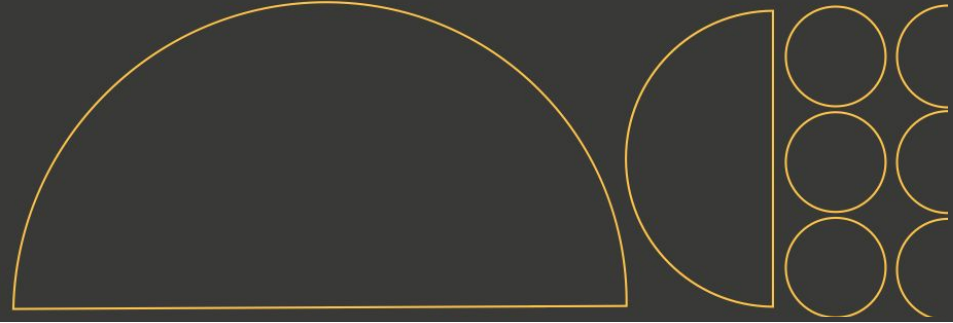
Access to computational resources:

- **RES project** AECT-2023-1-0028 (1/03/2023 - 30/06/2023)
- **EuroHPC Benchmark Access** Call - EHPC-BEN-2023B06-047 - VEGA GPU (15/06/2023 - 14/09/2023)
- **RES project** AECT-2023-2-0033 (1/07/2023 - 31/10/2023)
- Code audit for Tsunami-HySEA ChEESE flagship code - **EuroHPC Access Call** - EHPC-DEV-2023D09-030 (25/09/2023 - 25/03/2024)
- **RES project** AECT-2023-3-0017 (1/11/2023 - 28/02/2024)

Plans for CI/CD strategy:

- Eventually, extend and apply to other repositories/HPC infrastructures

Thank you!



@cheese-coe



<http://cheese2.eu>



@cheese-coe