Sprint documentation #7

HAMOCC Sprint

Enrico Degregori[1] and Fatemeh Chegini[2]

[1] Deutsches Klimarechenzentrum GmbH, Hamburg, Germany
[2] Max-Planck-Institute for Meteorology, Hamburg, Germany

Contact: info@nat-esm.de

Published on 06.03.24 on https://www.nat-esm.de/services/accepted-sprints

1 Summary

This sprint aimed at optimizing the computational efficiency of HAMOCC and ocean tracer transport on heterogeneous architectures. The sprint objectives included porting HAMOCC and tracer transport to GPUs and investigating their performance and scaling in a concurrent setup. Technical approaches involved GPU porting, optimizations, and investigation of communication between components. Insights include the development of asynchronous output, resolution scaling, and SLURM hetjobs configuration. Results demonstrate the feasibility of concurrent execution, with HAMOCC showing reasonable scaling on GPUs, paving the way for high-resolution simulations. We discuss challenges in communication time and future directions for optimization, emphasizing the potential for realistic ocean biogeochemistry simulations at higher resolutions.

2 General information

Start and end date:	07.06.2023 – 23.11.2023
Intended period:	6 months
Responsible RSE:	Enrico Degregori (DKRZ)
Responsible scientist:	Fatemeh Chegini (MPI-M)

The HAMburg Ocean Carbon Cycle model (HAMOCC) as part of the Earth System models at MPI-M simulates ocean biogeochemistry. The processes simulated by HAMOCC include biogeochemistry of the water column and upper sediment, as well as interactions with the atmosphere. In HAMOCC, Marine biology dynamics connects biogeochemical cycles and trophic levels through the uptake of nutrients and remineralization of organic matter, represented by the extended NPZD approach. In the water column currently at least 20 biogeochemical tracers are prognostically calculated. To achieve a consistent evolution of ocean biogeochemistry, the biogeochemical variables are handled as tracers on the three-dimensional grid of the ocean general circulation model. They are transported in the same manner, i.e., using the same numerical methods and time step, as salinity and temperature.

3 Sprint objectives

The aim of the sprint was to have a concurrent production experiment with the ocean running on CPU, and the biogeochemistry (HAMOCC and tracer transport) running on GPU.

A concurrent implementation of the ocean and the biogeochemistry was developed by Leonidas Linardakis (Linardakis et al. 2022). This implementation allows to scale the model beyond the domain decomposition. The benefit of a concurrent implementation is that it also allows to run different components on different architectures, so it can be used as an infrastructure when porting the HAMOCC model (HAMOCC and tracer transport) to GPU.

The goal of the sprint was to continue the previous effort of Linardakis et al. (2022) to port the ocean tracer transport and HAMOCC to GPU, and apply it to a concurrent production run inorder to use the HPC resources efficiently.

4 Procedure and insights

4.1 Technical Approach / procedure

The work of the sprint was separated into two parts. First, the GPU porting of the tracer transport and the HAMOCC model was completed with further optimizations. Then, the performance and scaling of the concurrent heterogeneous setup was investigated in detail at different resolutions up to R2B8.

The porting of the ocean tracer transport and the HAMOCC model is an effort which was already started before the natESM sprint. First of all, all the missing parts of the two components have been ported to GPU and then the following optimizations have been applied.

1. Avoid temporary arrays initialization to zero in order to get rid of some unnecessary kernels.
2. Use an asynchronous queue in the HAMOCC model since no MPI communication is involved.
3. Collapse loops without dependencies in order to achieve a better scaling of the model on GPU.

One critical part of the HAMOCC model is the monitoring which involves MPI reductions every time step. This implementation would perform poorly on the GPU, and it would prevent achieving a reasonable throughput. An alternative approach for the monitoring was implemented based on stages: during each time step the monitoring variables are calculated on the local partition and only at the output step the global monitoring variables are evaluated, involving MPI reductions. This implementation should also improve the scaling on a CPU system and the same approach can be applied to the ocean model when porting it to GPU.

In the second part of the sprint, the concurrent setup was investigated in different experiments using different resolutions (up to R2B8) and different numerical schemes (mlevel and zstar). The aim was to analyze HAMOCC on GPU in terms of both throughput and scaling and to investigate the communication between the two components. During this analysis the output was initially disabled to obtain the scaling plots, but in a second phase the asynchronous output was also enabled.

4.2 General Insights

The sprint allowed us to investigate in detail the YAXT library which is used for the communication between the two components having different domain decompositions. Increasing the experiments resolution, it was noticed an exponential increase in the YAXT initialization. A workaround was implemented by the YAXT developers which shows a significant speed up of the concurrent setup initialization for both homogeneous and heterogeneous setups.

YAXT implementation is heavily based on MPI Datatypes, and it is well known that they perform poorly on GPU. For this reason, a new exchanger was implemented by the YAXT developers in the backend which packs/unpacks the data into a buffer before/after the send/recv call. The heterogeneous setup requires to exchange data between CPUs and GPUs, so the new exchanger is (set with an environment variables) and on the HAMOCC side the GPU pointers are provided to the exchange call (using !$ACC HOST_DATA USE_DEVICE). No further changes are needed to support the exchange in the heterogeneous setup

The communication between the two components was investigated in detail during the second part of the sprint because it currently represents the main bottleneck of the heterogeneous setup and possibilities for improvement need to be further explored in the future. For this purpose, the R2B8 experiment was also run at JSC where it will be further investigated in the SCALEXA project.

The heterogeneous setup requires SLURM hetjobs in order to run the two components on different architectures. During the sprint, the SLURM hetjobs functionality was explored because it allows to achieve significant flexibility in running concurrent components efficiently. In particular, the following configuration was run on Levante:

1. Ocean on CPU partition with 4/8 OpenMP threads for each MPI process.
2. HAMOCC on GPU partition with 1 GPU for each MPI process.
3. Asynchronous output on CPU partition with 1 MPI process for each output file.

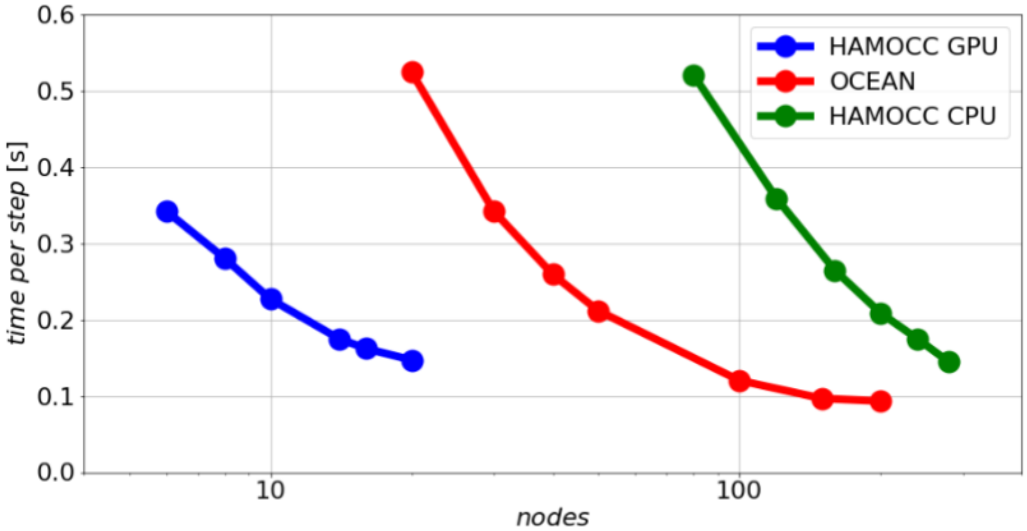
A Modular Supercomputing Architecture concept such as the one at JSC is specifically designed for this approach.

5 Results

The purpose of the sprint was to investigate throughput and scaling of HAMOCC on GPU. The scaling plots for the ocean on CPU and HAMOCC on both CPU and GPU are shown in the Figure below for 10km resolution (R2B8L128).

The different scaling between the two components on CPU explains why HAMOCC is a good candidate to run concurrently with the ocean. The concurrent setup allows us to use more resources on the HAMOCC model efficiently and achieve a higher throughput.

HAMOCC on GPU shows reasonable scaling even if, as expected, it is worse than the one on CPU. However, the ratio between CPU nodes and GPU nodes is around 10-20 to 1 along the scaling curves. This means that at high resolutions the GPU implementation is more energy efficient and at the same time the heterogeneous concurrent setup will allow in the future to simulate the ocean and the biogeochemistry at even higher resolutions.



6 Conclusions and Outlook

The outcome of the sprint shows the possibility to run concurrent components on different architectures. This opens new possibilities to run HAMOCC in higher spatial resolutions and in coupled ocean-atmosphere configurations. The main difference between ocean-atmosphere coupling and ocean-biogeochemistry coupling is that in the first case fields interpolation is needed but only 2D fields are exchanged between the two components. For this reason, in the first case a coupling library is necessary, while in the second case an exchange library is sufficient.

The sprint was useful to show some flaws of the YAXT exchange library which need to be tackled in the future, but it also allowed to show the possibility to easily exchange data between different components on different architectures.

The communication time between ocean and HAMOCC is the main bottleneck at the moment because of the 3D fields exchanged. Possibilities to reduce the communication time need to be explored in the future in order to be able to run concurrent heterogeneous setups in production.

Furthermore, porting HAMOCC to GPU would allow the addition of more tracers, necessary for a more realistic representation of the ocean biogeochemistry processes, while maintaining an acceptable throughput.

7 References

Linardakis, Leonidas, et al. "Improving scalability of Earth system models through coarse-grained component concurrency–a case study with the ICON v2. 6.5 modelling system." Geoscientific Model Development 15.24 (2022): 9157-9176.

A full documentation can be found on the natESM Gitlab wiki: https://gitlab.dkrz.de/natESM/natesm_sprints_documentation/-/wikis/Home/HAMOCC-GPU-porting