EGU21-10970
https://doi.org/10.5194/egusphere-egu21-10970
EGU General Assembly 2021

Porting NEMO diagnostics to GPU accelerators

Maicon Faria[1], Mario Acosta[2], Miguel Castrillo[2], Stella V. Paronuzzi Ticco[2], Sergi Palomas[2], David Vicente Dorca[1], and kim Serradell Maronda[2]

[1]Barcelona Supercomputing Center, Operations, Spain (mfaria@bsc.es)

[2]Barcelona Supercomputing Center, Earth Sciences, Spain (mfaria@bsc.es)

This work makes part of an effort to make NEMO capable of taking advantage of modern accelerators. To achieve this objective we focus on port routines in NEMO that have a small impact on code maintenance and the higher possible overall time footprint reductions. Our candidates to port were the diagnostic routines, specifically *diahsb* (heat, salt, volume budgets) and *diawri* (Ocean variables) diagnostics. These two diagnostics correspond to 5% of the NEMO's runtime each on our test cases. Both can be executed in an asynchronous fashion allowing overlap between diagnostic GPU and other NEMO routines CPU computations.
We report a methodology to port runtime diagnostics execution on NEMO to GPU using CUDA Fortran and OpenACC. Both synchronous and asynchronous are implemented on *diahsb* and *diawri* diagnostics. Associated time step and stream interleave are proposed to allow the overlap of CPU execution of NEMO and data communication between CPU, and GPU.

In the case of constraint computational resources and high-resolution grids, synchronous implementation of *diahsb* and *diawri* show up to 3.5x speed-up. With asynchronous implementation we achieve a higher speed-up from 2.7x to 5x with *diahsb* in the study cases. The results for this diagnostic optimization point out that the asynchronous approach is profitable even in the case where plenty of computational resources are available and the number of MPI ranks is in the threshold of parallel effectiveness for a given computational workload. For *diawri* on the other hand, the results of the asynchronous implementation depart from the *diahsb*. In the *diawri* diagnostic module there are 30 times more datasets demanding pinned memory to overlap communication between CPU and GPU with CPU execution. Pinned memory attribute limits data management of datasets allocated on main memory, therefore makes possible to the GPU access to main memory, overlapping CPU computation. The result is a scenario where the improvement from offloading the diagnostic computation impacts on NEMO CPU general execution. Our main hypothesis is that the amount of pinned memory used decreases the performance on runtime data management, this is confirmed by the 7% increase of the L3 data cache misses in the study case. Although the necessity of evaluating the amount of datasets needed for asynchronous communication on a diagnostic port, the payout of asynchronous diagnostic may be worth given the higher speed-up values that we can achieve with this technique. This work proves that models such as NEMO, developed only for CPU architectures, can port some of their computation to accelerators. Additionally, this work explains a successful and simple way

to implement an asynchronous approach, where CPU and GPU are working in parallel, but without modifying the CPU code itself, since the diagnostics are extracted as kernels for the GPU and the CPU is yet working in the simulation.