Sprint Documentation #14

ICON-YAC-CLEO

Wilton Jaciel Loch[1] and Clara Bayley[2]

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

Contact: info@nat-esm.de

Published on 07.01.2025 on https://www.nat-esm.de/services/accepted-sprints

1 Summary

This sprint addressed two major, yet largely independent, technical limitations of a model for cloud microphysics called CLEO for the benefit of studying warm-rain formation using large domain ICON Large Eddy Simulations with CLEO's microphysics scheme. The limitations were 1) that CLEO was not capable of advection of thermodynamic quantities because it was not coupled to a dynamical core, and 2) CLEO had limited scalability because it had no distributed memory parallelization.

To resolve the first major limitation of CLEO, the sprint created the infrastructure to couple CLEO to ICON via YAC. The infrastructure is complete for a one-way coupling and most of the ground-work is therefore put in place for the two-way coupling too.

To resolve the second major limitation of CLEO, the sprint implemented distributed memory parallelization via MPI domain decomposition. The parallelization was designed and fully implemented during the sprint but would nevertheless still benefit from a profiling of its performance.

The sprint is viewed as a success by both the RSE and responsible scientist. This is not only because of the objectives from the proposal which were completed, but also because of the byproducts of the sprint, for example in terms of collaboration, good scientific coding practices and the establishment of a new working group for ICON microphysics schemes. We strongly aspire for a follow-up sprint to complete the outstanding objectives of this sprint and for the continued progress of ICON's microphysics schemes.

2 General Information

Start and end date:	January 2024 – August 2024
Intended period:	06 months
Responsible RSE:	Wilton Jaciel Loch, DKRZ
Responsible scientist:	Clara Bayley, MPI-M

Within the last decade, a new model for cloud microphysics called the Super-Droplet Model (SDM) has become increasingly relevant for climate research because in comparison with previous models of cloud microphysics SDM has a much less ambiguous representation of cloud condensates and numerous computational advantages. CLEO is a C++ standalone implementation of SDM being developed to make superdroplet simulations with ICON in large domains (O(100km)) at lower resolution (circa. 100m) computationally feasible. It employs Kokkos as the main performance portability layer, natively allowing multiple methods for shared memory parallelization as well as accelerator offloading. CLEO is divided into libraries dedicated to specific aspects of the simulation, like grid management, and superdroplet movement, all tied together by example drivers. The C++ code has a size of around 10500 LOC.

To progress towards its strategic goals, two of CLEO's major outstanding development objectives were tackled: 1) to couple CLEO to ICON via YAC and 2) to implement MPI domain decomposition in CLEO.

3 Sprint Objectives

The sprint consisted of two primary independent objectives, each with several own sub-steps and criteria for fulfillment. These were:

1. Coupling CLEO to ICON via YAC. The fulfillment of which was to run one-way and two-way coupled ICON cloud bubble test cases.
2. MPI Domain decomposition within CLEO. The fulfillment of which was to run CLEO across mode than one node of Levante and then profile its computational performance.

4 Procedure and Insights

4.1 Technical Approach / Procedure

As already outlined above, the sprint had two primary independent goals: the coupling with ICON and the MPI parallelization. Upon discussion, we decided that as the coupling with ICON was deemed more important than the parallelization, the sprint should start there, and a plan was outlined to have an intermediate stage first. This stage consisted of a Python/YAC infrastructure which would be used to implement an input server for thermodynamic data, with CLEO receiving the data from YAC on one side, and a Python script reading the data and sending it through YAC on the other. To get to this stage a series of even smaller experimental tasks was devised. Given also the lack of experience of the RSE with YAC, this proved very useful for a smoother learning and development process.

The first step was the creation of simple producer and consumer scripts using Python and YAC to exchange dummy data. Python was chosen for its simplicity, and it allowed a quick understanding of YAC concepts such as the grid definition, creation of fields and couplings. Once this was done and the concepts understood, the next step was the creation of a Python script to read the CLEO binary data. This was done simply by translating the already existing C++ reader inside of CLEO. The following stage was to integrate the Python data reader and the Python producer to now exchange data which would be actually meaningful for CLEO. After this step, the Python reader script was virtually finalized, and the focus became replacing the consumer script with CLEO. For this, two interface implementations were created for a new "provider" of thermodynamics data in CLEO, comprising the whole YAC infrastructure of grid, field and coupling definitions. With this the intermediate stage was reached and the next stage was to perform the one-way coupled bubble test case. This involved the replacement of CLEO input data by ICON bubble data, and the addition of the coupling infrastructure into ICON. This was accomplished, meaning the one-way ICON bubble test case worked from the technical perspective.

The RSE work then proceeded to the MPI parallelization of CLEO. For this goal, unfortunately, the steps to be taken cannot be so easily outlined in the beginning, it is harder to have incremental changes, and many challenges can arise during the development itself. Therefore, the approach was to resort to a more experimental development workflow, simply implementing the parallelization in a naive way first to explore the code and understand which problems would have to be solved. This proved very useful as after a couple of weeks there was a much clearer view on what had to be added and changed for the parallelization to work. Aspects like superdroplet and gridbox initialization, domain decomposition, inter-gridbox superdroplet motion, IO and others had to be touched to implement the distributed memory parallelization. After all the changes were in place, the MPI parallelization could be finalized and tests were made with various numbers of processes, all showing the same output results as the sequential run.

In terms of the sprint's schedule, the work on both goals partially overlapped at times, for example when waiting for the ICON bubble data and scientific verification of the results from the bubble test case. The interaction revolved around weekly meetings and frequent communication via Mattermost. As smaller work packages were finished, they were already merged back into the main branch, to prevent large integration conflicts at the end of the sprint.

4.2 General Insights

It was integral to the success of the sprint to have well-defined sub-goals, even small ones that only needed one/two weeks of work. These sub-goals made the work for the project easier to grasp/tackle, simplified concurrent collaboration and merging, and helped us monitor if we were on track to complete our objectives within the allotted time-frame.

Having an informal channel of communication (Mattermost) was another key foundation for the sprint's success. It enabled small issues/queries to be resolved quickly and also asynchronous communication when we were in different time zones.

The sprint was fortunate that its two major goals were independent of one-another. That enabled more efficient time-management because we could switch tasks when roadblocks occurred. For example, when the RSE was waiting for ICON bubble data in order to run the cloud bubble test case, he started the work on the MPI parallelization concurrently.

5 Results

Regarding the goal of coupling CLEO to ICON via YAC, we successfully created the infrastructure to couple to ICON via YAC, and we partially completed the tasks of running the cloud bubble test cases with one-way and two-way couplings. The one-way coupling completely works for both cell-based and edge-based thermodynamics fields. As a result of the one-way coupling, most of the infrastructure for the two-way coupling is in place and we also have a draft merge request into the ICON repository which fully integrates the one-way coupling of CLEO into ICON. Running the cloud bubble test case with a two-way coupling in ICON would then be possible with relatively small additions compared to the one-way coupling, although further work may then be required to ensure it gives physically reasonable results.

The goal of implementing MPI domain decomposition within CLEO was completed. The sprint successfully created a domain decomposition for CLEO across nodes, a method to write output from many MPI processes, and a method to move superdroplets between nodes. Nevertheless, there is some scope for refinement. CLEO would benefit greatly from making the movement of superdroplets compatible with GPU builds, and from generalizing the domain decomposition so that Gridboxes could have non-uniform dimensions because that would pave the way to enabling the integration of the domain decomposition with the coupling to ICON via YAC. The performance of the MPI communication also still needs to be measured so we can identify and resolve any major bottlenecks.

Another tangential achievement of the sprint was the establishment of a group for discussing the requirements and design of a generalized ICON interface for microphysics models. The goal is that in the future, various different microphysics implementations are able to be used interchangeably within ICON. The group was formed by contacting other members of DKRZ, MPI-M and CSCS which are involved in projects regarding externalization of the microphysics in ICON or the development of completely new models. The group has regular meetings, and continuous developments are being done in ICON. If you would like to learn more about this initiative or get involved, please contact Clara Bayley (clara.bayley@mpimet.mpg.de) and/or follow the merge request (https://gitlab.dkrz.de/icon/icon-mpim/-/merge_requests/504).

6 Conclusions and Outlook

The sprint was hugely beneficial to the development of CLEO—not only because of the rapid progress made towards the goals of the sprint, but also because of the many byproducts of the sprint. For example, the sprint motivated expanding CLEO's documentation and GitHub CI, as well as introducing pre-commit, version management and standard formatting and linting of the code.

The sprint was also a major success for smooth and enjoyable collaboration between a RSE and a research scientist. We were able to communicate well, and Clara learned a lot from Wilton's software engineering expertise and coding (e.g. about YAC and MPI). As a result, the development of CLEO has not only made substantial technical progress, but also resulted in strengthening collaboration and the dissemination of knowledge. No doubt bringing benefits for the ICON and cloud microphysics communities.

As a bottomline we can state that the outlined goals of the sprint have been technically achieved for the major part. There are still some areas left open to development which a follow up sprint would serve well:

On the goal of coupling CLEO to ICON via YAC:

- Implement the two-way coupling and run the two-way coupled ICON bubble test case.

On the goal of implementing MPI domain decomposition within CLEO:

- Make the distributed memory parallelization compatible with the YAC coupling to ICON. Evaluate the performance of the parallelization. Identify and improve bottlenecks where necessary

7 References

The CLEO GitHub project can be found here: https://github.com/yoctoyotta1024/CLEO. Please contact Clara Bayley (clara.bayley@mpimet.mpg.de) if you would like further information and/or access to more documentation about CLEO or the sprint project.