Sprint Documentation 19

Optimization of ICON-XPP for the DWD NEC Aurora vector computer

Sergey Sukov[1], Trang Van Pham[2] and Jens-Olaf Beismann[3]

[1]Jülich Supercomputing Center, Forschungszentrum Jülich GmbH, Jülich, Germany
[2]Deutscher Wetterdienst, Offenbach, Germany
[3]NEC Deutschland GmbH, Düsseldorf, Germany

Contact: info@nat-esm.de

Published on 10 February 2026 on https://www.nat-esm.de/services/accepted-sprints

1 Summary

The objective of the sprint was to improve the performance of the ICON-XPP model on the RCL/DWD HPC system, with a particular focus on the ocean-carbon-cycle module operating within the ocean component. As a result of vectorizing the HAMOCC subroutines, their execution time was reduced by a factor of 40, which, among other things, enabled ICON-XPP experiments on RCL/DWD as part of the CMIP7 project. The changes introduced to the code have been merged into the master ICON branch and will be available in an Open-Source Release of the ICON model soon.

2 General Information

Start and end date:	May 2025 – October 2025
Intended period:	6 months
Responsible RSE:	Sergey Sukov (JSC)
Responsible scientist:	Trang Van Pham (DWD)
Technical consultant:	Jens-Olaf Beismann (NEC)

ICON-XPP (ICON eXtended Predictions and Projections) is the climate-modelling tool used by most of the German Climate Community. It is based on the components of the Deutscher Wetterdienst (DWD) numerical weather prediction model (ICON-NWP), the land model (JSBACH), the ocean-sea ice model (ICON-Ocean), hydrology model (HD), and the coupler YAC (DKRZ). ICON-XPP is extensively used for the climate-forecast tests and climate projection run at DWD on the NEC Aurora Vector Machine, RCL. Peak performance on the RCL/DWD HPC system is achieved when the code meets the vectorization requirements. Optimization of ICON-XPP components for NEC architectures is implemented continuously. Nevertheless, by the start of the sprint, a significant performance imbalance was observed between the two most resource-intensive components, the atmosphere and the ocean components. In a basic experimental configuration, the ICON-Ocean model required a substantially larger number of vector engines for execution than the atmospheric model. In the next stage, upon activation of HAMOCC modules (Hamburg Model of the Ocean Carbon Cycle, which is an unvectorized part of the ocean model ICON-Ocean), ICON-XPP experiments were not feasible on the RCL/DWD as part of the 7th Coupled Model Intercomparison Project (CMIP7).

3 Sprint Objectives

The main objective of the sprint was to improve the ICON-XPP model performance on RCL/DWD, with a particular focus on the ocean-model code. More specifically, the task involved vectorizing HAMOCC subroutines used within the coupled ICON-NWP/ICON-Ocean/HD/YAC model system. During the vectorization process, attention should have been paid to exploring the potential benefits of implementing a "wet"-cell index list. Additionally, in the final test runs, it was planned to formulate a pseudo-optimal algorithm for distributing hardware resources between the atmosphere and ocean components.

4 Procedure and Insights

4.1 Technical Approach / Procedure

The actual sprint schedule consisted of the following stages:

1. Profiling and analysis of the HAMOCC model code structure.
2. Vectorization of HAMOCC subroutines.
3. Final performance runs of the vectorized code, including determination of the optimal execution configuration on RCL/DWD.
4. Merging the updated code into the master branch of the ICON repository hosted at DKRZ (Git).

In the course of profiling and evaluating the performance gains achieved through vectorization, data from the NEC Ftrace profiler, ICON's built-in timers, and additional control parameters implemented in the code by the RSE were collected and analyzed. The NEC Ftrace profiler provides detailed information on the percentage of code vectorization, the average vector length, and the execution times of individual subroutines. Data from ICON's built-in timers allow for assessment of the scaling of parallel computations. The additional parameters introduced by the RSE quantify the load imbalance across MPI processes in terms of the number of grid primitives assigned (surface cells, depth levels, etc.).

Vectorization of HAMOCC subroutines (essentially, vectorization of loops within the subroutines) was carried out step by step, in descending order of their execution times, applying one or more of the following techniques:

- loop reordering.
- loop splitting.
- replacement of scalar functions with their vectorized counterparts.

Switching to vectorized code blocks was controlled via the conditional compilation macro **__LVECTOR__**, which is conventionally used in ICON for this purpose. Due to load imbalance and the nonlinear growth of overheads associated with MPI data exchange and component coupling, the optimal distribution of hardware resources for the final performance runs was determined experimentally. Merging the NEC-optimized code into the ICON master branch involved validating the correct operation of the vectorized subroutines on standard multicore CPUs (Levante, DKRZ).

Discussions of the sprint's technical details among the RSE, scientists, and NEC HPC department team members were carried out primarily via email, the Mattermost channel, and scheduled weekly video conferences.

4.2 General Insights

Regardless of the compiler and hardware platform, vectorization is applied only to data-independent iterations of innermost loops. A second limitation is the inability to invoke external subroutines within the loop body, except in cases where the corresponding procedures can be automatically inlined. Otherwise, to improve performance, the original loop is split into multiple, independently vectorizable subloops, and calls to scalar functions are replaced by their vector counterparts. Such loop fission is accompanied by an increase in the dimensionality of temporary variables, whose values are updated sequentially across the subloops.

Throughout the sprint, the RSE observed that the NEC compiler supports vectorization of more complex code blocks compared to the Intel and NVHPC compilers. Vectorizing ICON-Ocean subroutines on HPC systems with Intel, AMD, or NVIDIA multicore CPUs requires more extensive modifications of the original scalar code. Therefore, within the scope of this sprint, it was assumed that special code branches optimized for the NEC compiler and hardware would not be executed on other platforms. The purpose of running the NEC-optimized HAMOCC subroutines on Levante was only to verify computational correctness. Performance evaluation was not performed.

Approximately a quarter of the HAMOCC subroutines already had source code conforming to vectorization requirements. However, to improve performance, loops over surface elements and depth levels were reordered. A similar approach had previously been applied to optimize ocean model subroutines for NEC and to offload them to the GPU. According to Ftrace reports, this loop reordering resulted in a significant increase in the average vector length.

During the initial profiling stage, it was found that changing the distribution of computational resources between the ocean and atmosphere components led to differences in the results of coupled runs. Therefore, while vectorizing the HAMOCC subroutines, all reference calculations were performed using a fixed number of MPI processes, vector units, and a single hardware distribution between the components.

The task mentioned in the initial sprint plan for introducing a “wet”-cell index list was omitted, as its implementation would have required extensive modifications to the code. Implementing the part of the subroutines that replaced multiple cell-type checks by saving the result of the first operator into a separate logical array did not lead to any performance gains.

5 Results

The main outcome of the completed sprint was the optimization of HAMOCC model subroutine code for the NEC Aurora Vector Machine architecture, RCL. Table 1 presents the average execution times of specific subroutines before and after code vectorization. The data were collected during ICON-NWP/ICON-Ocean/HD/YAC coupled simulations for a one-month model-time period, with ICON-NWP running on 33 vector engines and ICON-Ocean running on 127 vector engines. According to a rough estimate based on the Workload Manager report, the HAMOCC subroutines consumed approximately 65% of the total runtime before the optimization was applied.

Subroutine	Execution time (seconds)		Speedup
	Before vectorization	After vectorization	
calc_dissol	216.36	4.43	48.8
gasex	4.78	0.13	38.2
powach	36.42	0.96	38.1
ocprod	159.98	2.66	60.2
mean_agg_sink_speed	442.243	8.281	53.4
chemcon	4.759	0.068	70.0
settling	17.226	0.225	76.6
cyadyn	4.426	1.206	3.7
update_bgc	3.203	0.566	5.7
swr_absorption	1.957	0.289	6.8
update_icon	1.782	0.268	6.6
set_bgc_tend_output	0.779	0.779	1.0
TOTAL	893.92	19.85	45.0

Table 1. Execution times for the individual HAMOCC subroutines.

It is important to note that these results primarily demonstrate the necessity of thorough profiling and, where required, subsequent modification of the original scalar code prior to execution on

RCL/DWD. The speedup achieved strongly depends on the structure and coding style of the baseline program.

Table 2 presents the performance metrics of coupled runs on RCL/DWD, measured in SYPD (Simulation Years Per Day). The optimal distribution of hardware resources among the components was determined experimentally. The data were provided by Guang Zeng (DWD).

Number of vector engines	Number of MPI processes			SYPD
	Total	Ocean	Atmosphere	
64	512	352	159	9.7
80	640	464	175	11.4
128	1024	752	271	15.2
160	1280	976	303	17.0

Table 2. Hardware resource distribution and performance of coupled runs.

The performance value accounts for the I/O overhead. Preliminary runs with non-vectorized HAMOCC subroutines showed a performance of 4.6 SYPD on 160 vector engines. Consequently, code vectorization reduced the simulation time by more than 3.5 times.

The obtained performance metrics demonstrate the feasibility of running ICON-XPP experiments on RCL/DWD as part of the CMIP7 project.

6 Conclusions and Outlook

This sprint represents one of the sequential steps toward achieving the overall goal of improving the performance of the ICON-XPP model on the RCL/DWD HPC system. Although ICON-XPP in its current state already enables production forecast and projection simulations, the codebase still contains procedures, modules, and components whose vectorization efficiency has not yet been evaluated. Accordingly, a possible scope for a follow-up natESM sprint would be the further optimization of the code for the NEC vector architecture, leveraging the experience obtained in this project.

7 References

https://gitlab.dkrz.de/icon/icon-mpim/-/tree/master?ref_type=heads