# ICON on its Way to Exascale – Status and Next Steps

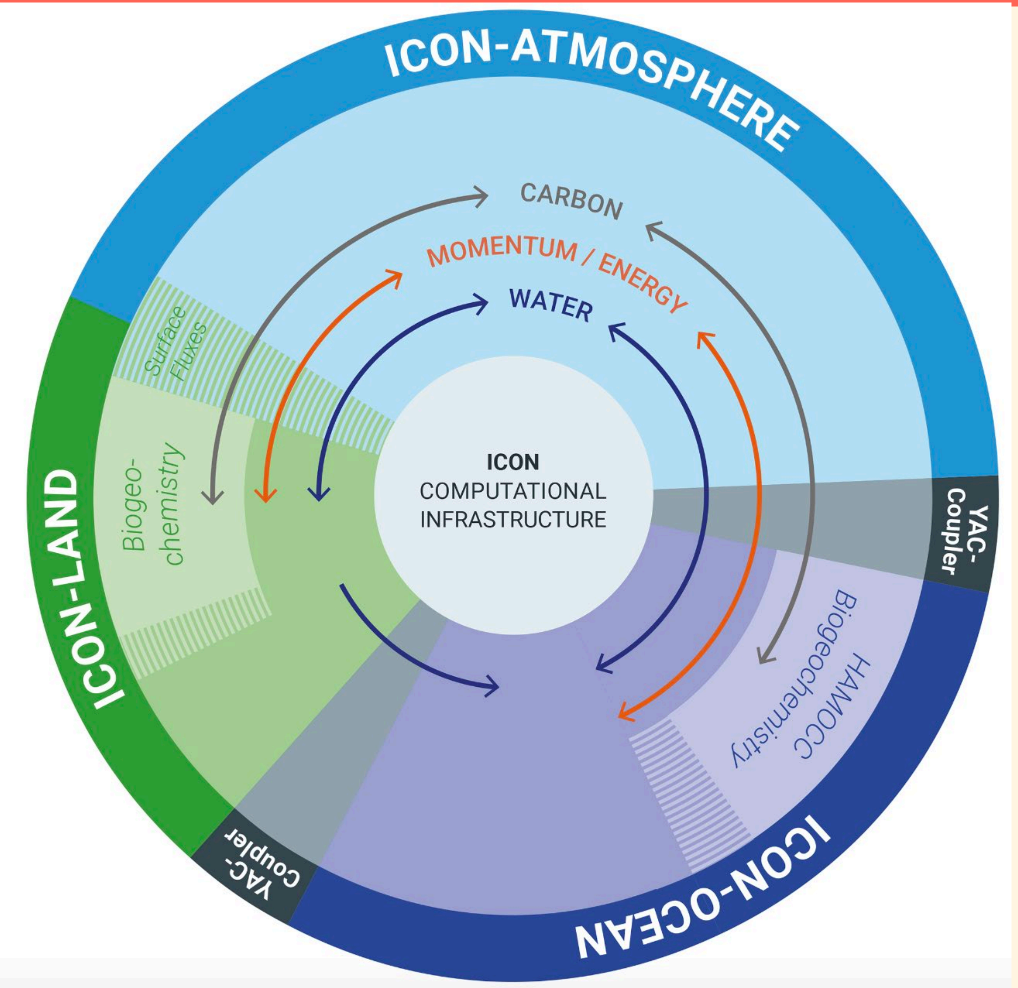
Claudia Frauen[1] (frauen@dkrz.de) and the WarmWorld Faster, ICON-C

and DestinE climate DT teams[2,3,4,5,6,7,8,9]

[1]German Climate Computing Center (DKRZ), [2]Max-Planck-Institut für Meteorologie, [3]Karlsruhe Institute of Technology, [4]German Weather Service, [5]Jülich Supercomputing Center, [6]Alfred-Wegener-Institute Helmholtz Center for Polar and Marine Research, [7]European Center for Medium Range Weather Forecast, [8]Center for Climate Systems Modelling (C2SM), [9]CSC

# The weather and climate model ICON

- Used for different applications, e.g. numerical weather forecasts, CMIP-type climate simulations, km-scale simulations
- ~20 years development by DWD, MPI-M, DKRZ, KIT and C2SM
- Written mostly in Fortran, using MPI/OpenMP for parallelisation




Hohenegger et al., 2023

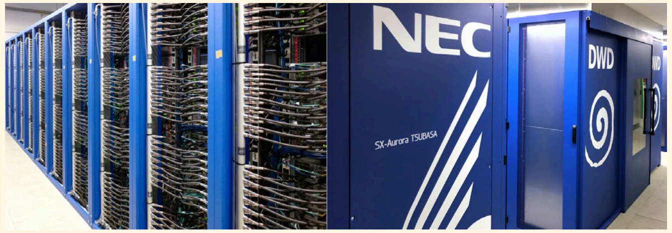
Federal Ministry of Education and Research

# ICON – target systems

DKRZ – Levante:
"Standard" CPU system
(AMD EPYC Milan)


CSCS – Piz Daint:
NVIDIA P100 GPUs
https://www.cscs.ch/publications/news/2013/with-piz-daint-cscs-enters-the-path-towards-petaflop-computing/


CSCS – Alps:
NVIDIA Grace
Hopper
https://www.cscs.ch/computers/alps




DWD - NEC SX Aurora
Tsubasa
https://www.dwd.de/DE/derdwd/it/_functions/Teasergroup/datenverarbeitung.html


JSC – JUWELS Booster: NVIDIA
A100 GPUs
Copyright: Forschungszentrum Jülich / Wilhelm-Peter Schneider


CSC – LUMI: AMD MI250 GPUs
(Image credits: Fade Creative)

SPONSORED BY THE
Federal Ministry
of Education
and Research

# ICON Sapphire

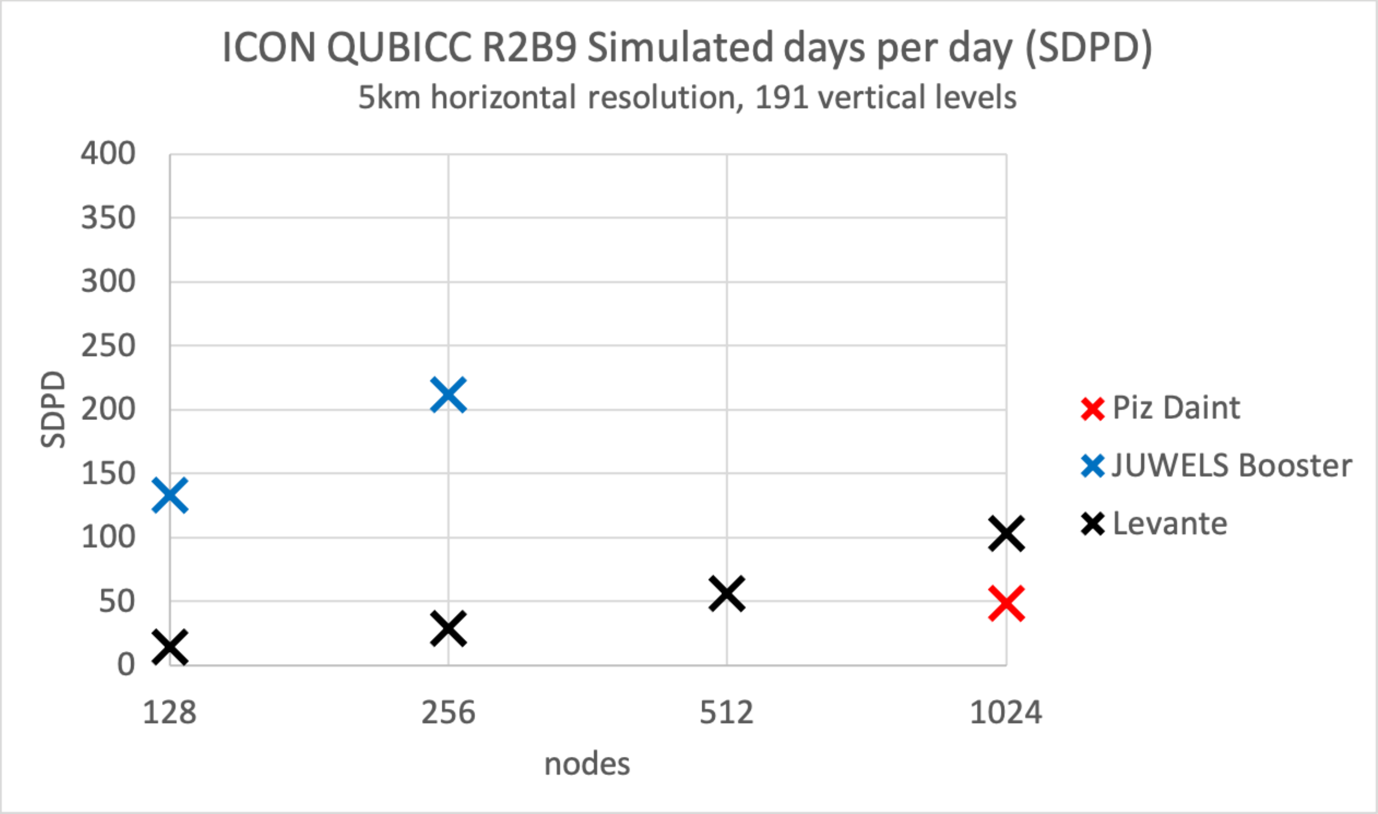
- Coupled model performs and scales well on CPU systems like Levante
- Still far away from goal of 1 SYPD @ 1 km resolution

| Grid spacing | Machine | Nodes | SDPD |
|---|---|---|---|
| 5 km | Mistral | 420 (300 A, 120 O) | 17 |
| ” | Levante | 600, 24A:8O | 126 |
| ” | Levante | 420, 24A:8O | 96 |
| ” | Levante | 400, 24A:8O | 90 |
| ” | Levante | 200, 24A:8O | 48 |
| ” | Levante | 100, 24A:8O | 24 |
| 2.5 km | Levante | 600, 24A:8O | 20 |
| 1.25 km | Levante | 900, 24A:8O | 2.5 |
| 1.25 km (A) | Levante | 908 | 4 |
| 1.25 km (O) | Levante | 1024 | 97 |
| ” | Levante | 2048 | 179 |

Hohenegger et al., 2023

# ICON on (NVIDIA) GPUs

- Huge effort has been made initially especially by CSCS and NVIDIA to port ICON-A to GPUs using OpenACC directives to run on Piz Daint at CSCS and benefit from its large computing power
- Within the QUBICC project first ICON-A simulations were performed initially on Piz Daint and later on JUWELS Booster at JSC (Giorgetta et al., 2022)




Adapted from Giorgetta et al., 2022

SPONSORED BY THE
Federal Ministry of Education and Research

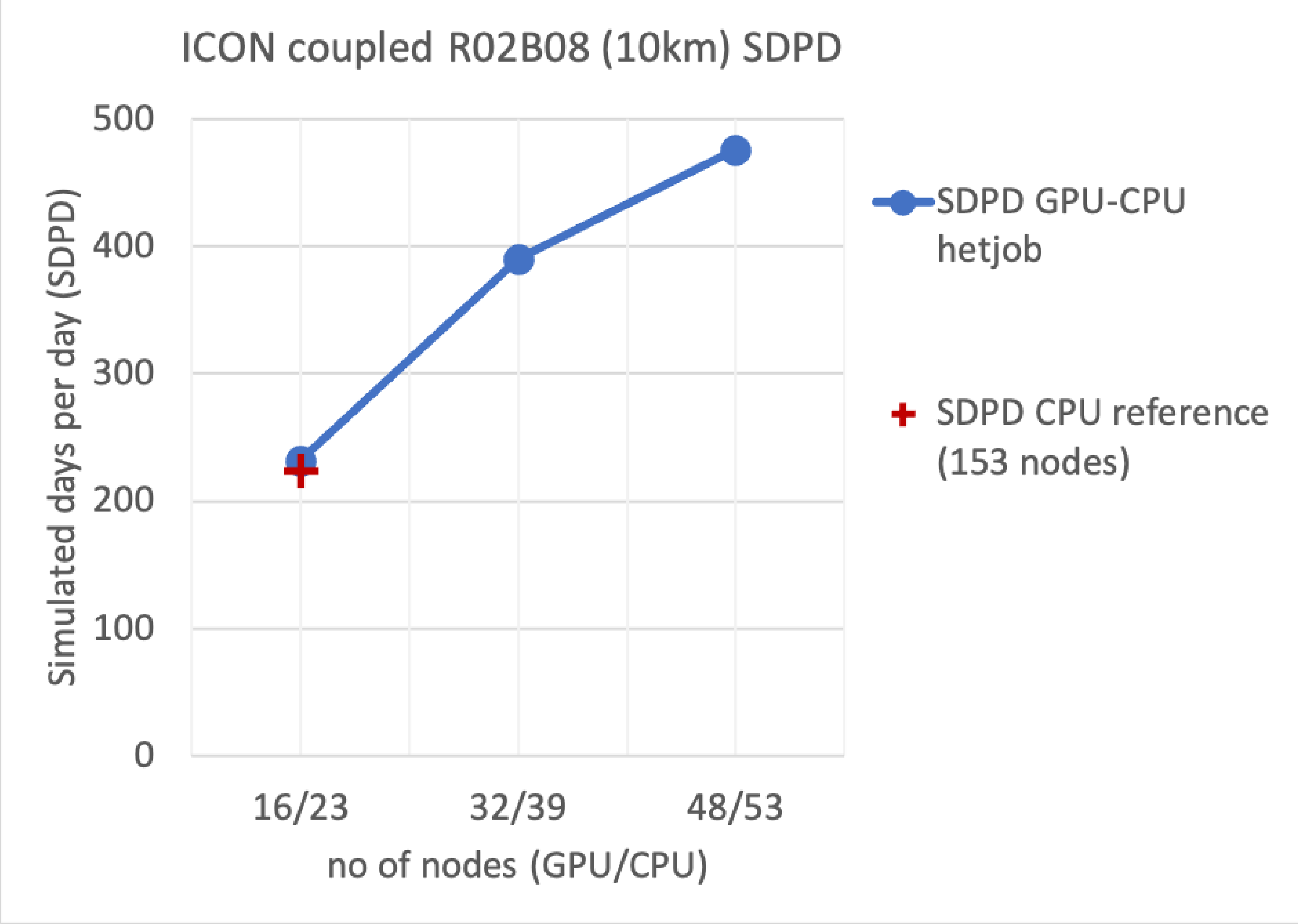
# ICON on (NVIDIA) GPUs

- Land component of ICON (JSBACH) had initially been ported using CLAW DSL in order to avoid refactoring large parts of the code
- End of CLAW development and support forced a rewrite with OpenACC directives after all
- Ports of ocean component (ICON-O), atmospheric chemistry (ART) and ocean biogeochemistry (HAMOCC) using OpenACC directives are in progress

SPONSORED BY THE
Federal Ministry of Education and Research

# Heterogeneous architectures

- Ocean component needs much less computing ressources than atmosphere component
- To run the coupled model we can take advantage of heterogeneous architectures by running the atmosphere component on GPU partitions and I/O and the ocean component on CPU partitions on systems like Levante, JUWELS and LUMI




Federal Ministry of Education and Research

WARM WORLD

# ICON on LUMI

- In autumn 2020 it was announced that the EuroHPC pre-exascale system LUMI will be making use of AMD GPUs as accelerators
- Initial tests with using OpenMP for GPU offloading showed that compiler support was not as good as for OpenACC
- HPE/Cray commited to supporting Fortran and OpenACC in the Cray Compiler
- => Decision was made to stay with OpenACC also for LUMI

SPONSORED BY THE

Federal Ministry of Education and Research

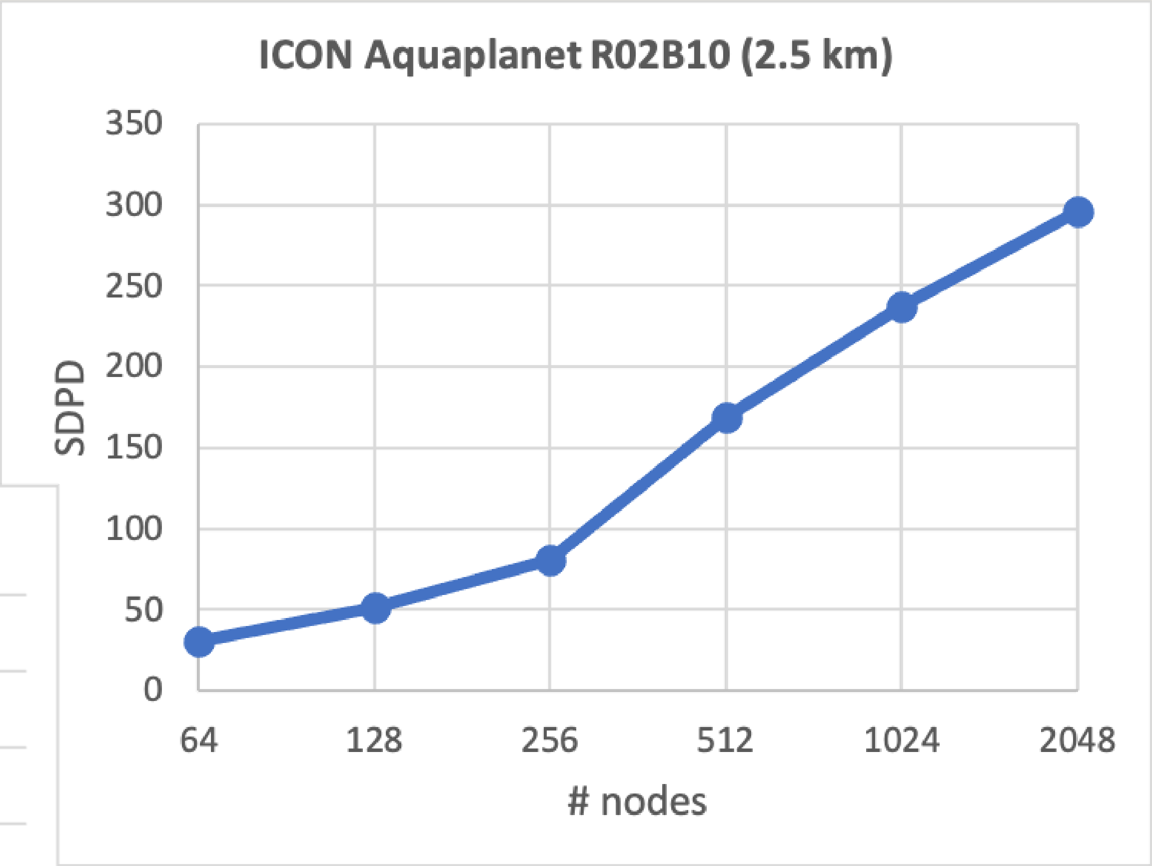
# ICON on LUMI

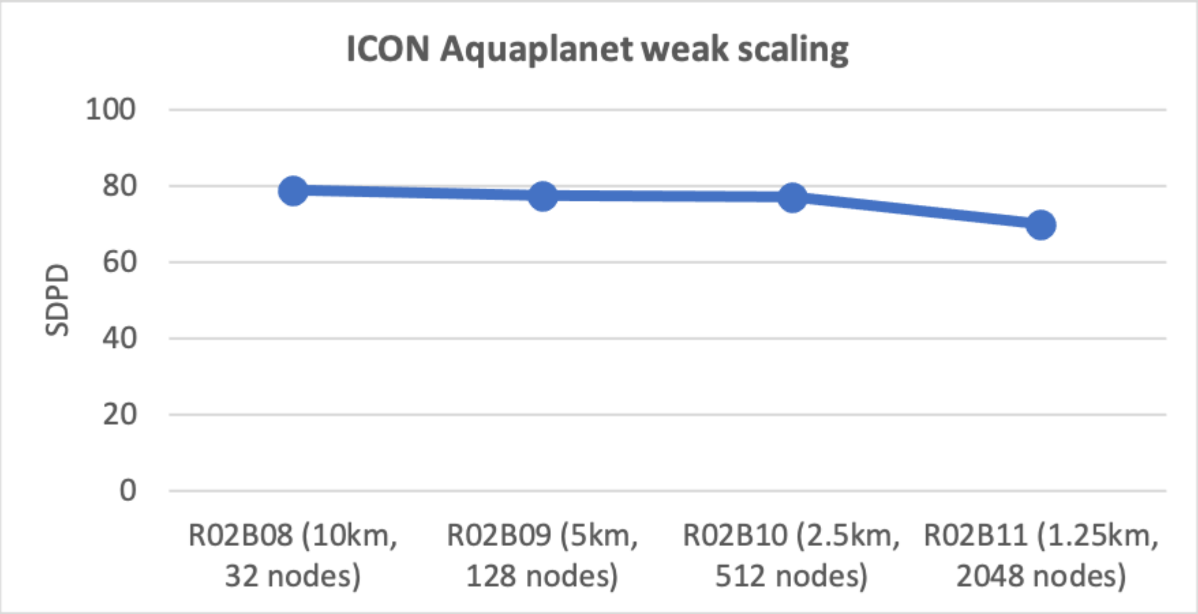
- Earlier version of ICON (2.6.0) was benchmark code for LUMI and was ported to LUMI by HPE/Cray
- Since then a lot more of the ICON code has been ported to (NVIDIA) GPUs, the initial OpenACC implementation has been optimized and parts of the model have been rewritten
- Initial issues:
  - Different OpenACC standard interpretations between NVIDIA and Cray – small rewrites needed
  - Small amount of CUDA code in ICON had to be rewritten in HIP
  - Compiler bugs
- Ongoing issues:
  - Some compiler bugs were fixed in CCE 15 (available on LUMI since April 2023); others still persist. Now working with CCE 16.0.1.1 in a container – waiting for CCE 16 to be officially released and made available on LUMI
  - Performance is still far from where we want to be

SPONSORED BY THE
Federal Ministry of Education and Research

# ICON on LUMI - Aquaplanet

- First experiment to get running: Aquaplanet with resolution as high as 1.25km and using up to 2048 nodes

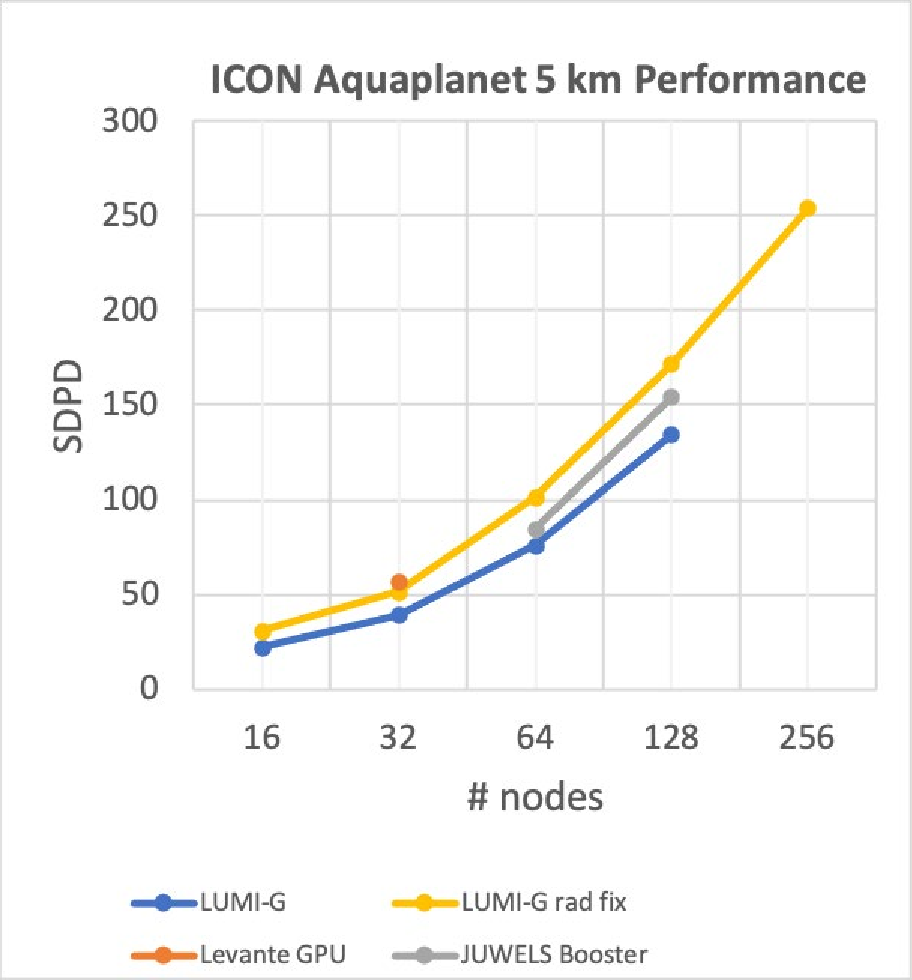



SPONSORED BY THE
Federal Ministry of Education and Research

# ICON Aquaplanet performance

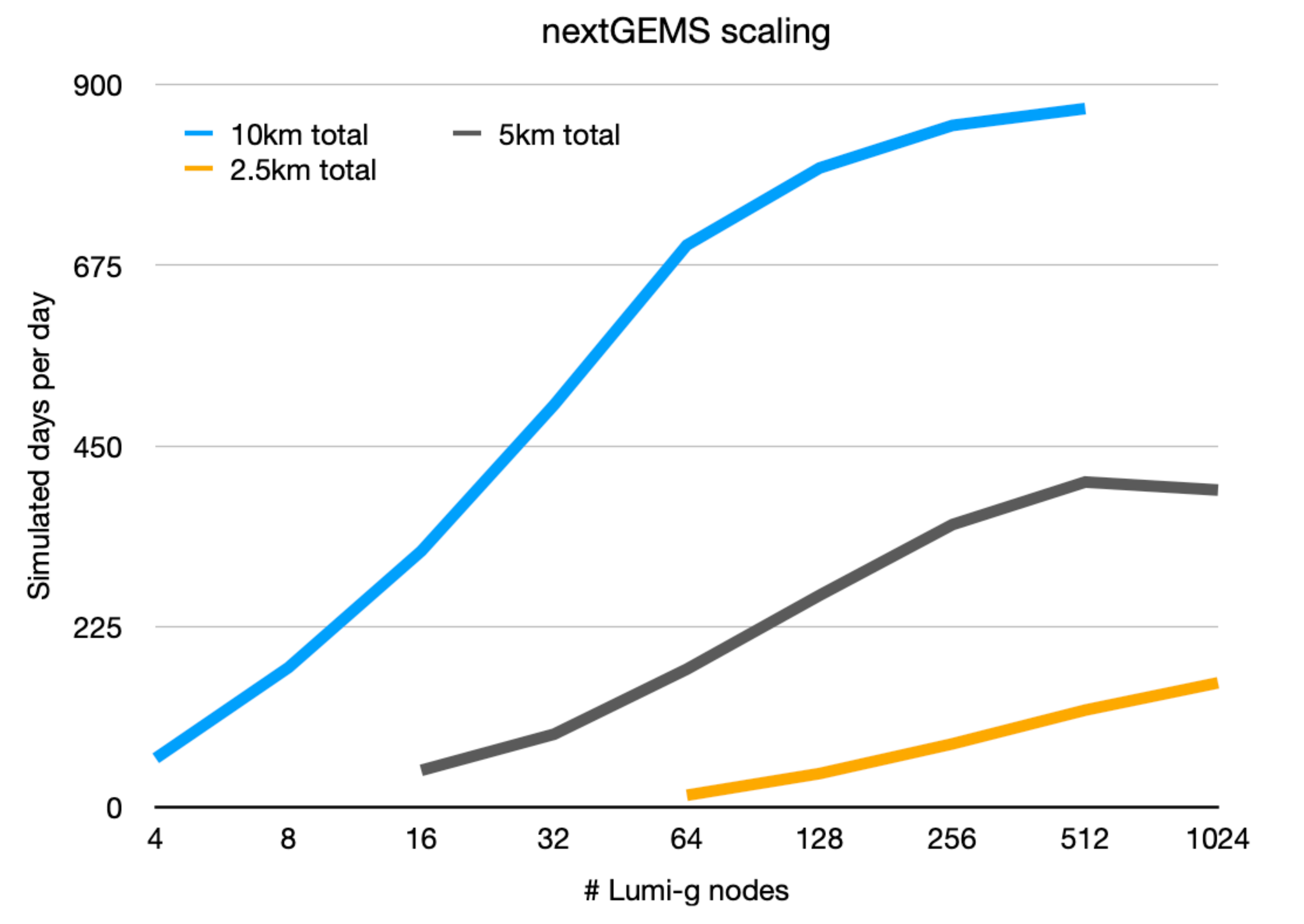
- LUMI-G node:
  - 4x AMD MI250x GPU modules
  - 2x Graphics Compute Die (GCD) per module
  - 64 GB memory per GCD
- Levante-GPU node
  - 4x NVIDIA A100
  - 80 GB memory per GPU
- JUWELS Booster node
  - 4x NVIDIA A100
  - 40 GB memory per GPU




Time step 22.5s; ideal would be 45s

SPONSORED BY THE
Federal Ministry of Education and Research

# ICON full atmospheric model setup




SPONSORED BY THE

Federal Ministry of Education and Research

# ICON performance portability - summary

- OpenACC was a relatively easy way to port ICON-A to NVIDIA GPUs
- Other ICON components follow the same strategy
- Experience with AMD GPUs on LUMI shows that it is not an easily portable solution
- What about future HPC systems with AMD GPUs without the Cray compiler or Intel GPUs?

SPONSORED BY THE
Federal Ministry of Education and Research

# WarmWorld Module Faster

- Part of the German BMBF funded project WarmWorld
- Goals are to
  - transform the ICON code base into an open, scalable, modularized and flexible code named **ICON-C** (“ICON-consolidated”).
  - refactor ICON with the goal of scalable development to enable portable performance improvements – ultimately making ICON faster
  - initiate target performance ports to meet throughput (>0.5SYPD on a 2.5km or finer grid) goals
  - progressively redefine the ICON code structure to expose areas of performance improvement for targeted exploration of new programming concepts

SPONSORED BY THE
Federal Ministry of Education and Research

# ICON-Consolidated

- ICON-C is also an initiative of the whole ICON consortium
- Ongoing activities focus on
  - Modularisation
  - Librarisation
  - New language-interoperable memory management
  - New community interface
  - Improved testing strategy

SPONSORED BY THE
Federal Ministry of Education and Research

# Modularisation

- Aims to disentangle the model code into granules with
  - Well-defined interfaces exposed to the ICON driver
  - Stand-alone driver
  - Ability to serialize data from real ICON run and/or contain reasonable test cases
- Examples are new muphys or advection
- Stand-alone setups can also be used to explore different programming paradigms:
  - E.g. current projects use the new muphys to explore the possibilities of Kokkos and SYCL

SPONSORED BY THE
Federal Ministry of Education and Research

# Librarisation

- Aims to extract base bricks of ICON into external libraries
- Does not touch parts of the ICON model directly
- ICON-agnostic libraries, like a Fortran support library or base math utilities
- ICON-aware libraries, like interpolations or math operations

SPONSORED BY THE
Federal Ministry of Education and Research

# Memory Manager

- Aim of the new memory management is to enable language interoperability
- Necessary to be able to incrementally replace parts of the model with non-Fortran code
- E.g. to make use of vendor-native parallel programming frameworks or portability layers

SPONSORED BY THE

Federal Ministry of Education and Research

# ICON Community Interface ComIn

## What are the aims of ComIn?

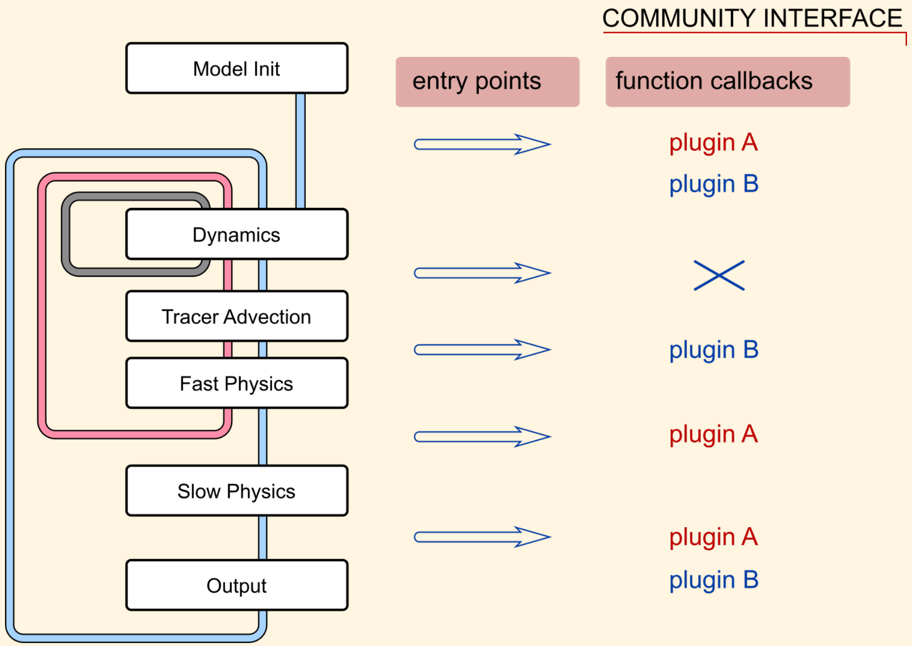
➡ Providing a standardized **public interface** for third party codes (‘**plugins**’) coupled to ICON

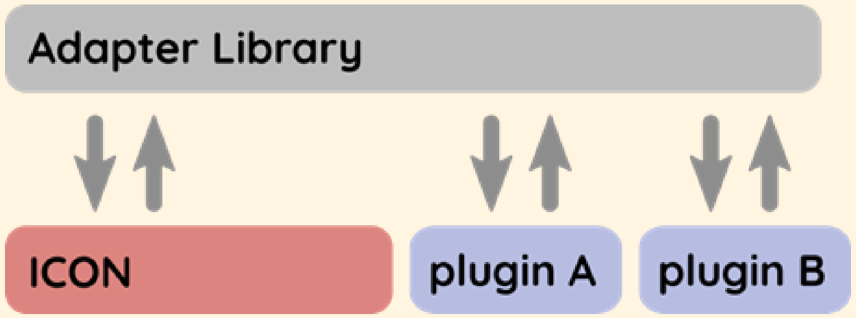
➡ Significantly **reduced maintenance** for ICON as well as for third party code developers

➡ Plugins **easier to migrate** to new ICON releases

➡ Establishing ICON as the core model for applications ranging from **NWP** to **ESM**

➡ Enables **multi-language support** (Fortran, C/C++, Python)




Courtesy F. Prill, DWD

## How does ComIn work in a nutshell?

➡ ComIn organizes the **data exchange** and **simulation events** between the ICON model and multiple plugins.

➡ **ComIn Callback Register**: Subroutines of the plugins are called at pre-defined events during the ICON simulation.

➡ The **ComIn Adapter Library** is included by ICON and the plugins. It contains descriptive data structures and regulates the access and the creation of model variables.

# Conclusion

- ICON-A runs successfully on NVIDIA GPU based systems by using OpenACC directives
- Ports of other ICON components are ongoing following the same strategy
- Getting it running on the AMD GPU based system LUMI is a struggle and shows the limitations of the use of OpenACC for portability
- Work is ongoing to transform the ICON code base into an open, scalable, modularized and flexible code named **ICON-C** ("ICON-consolidated")

SPONSORED BY THE

Federal Ministry of Education and Research

# Outlook

- Fortran and OpenACC seems to be a dead end when it comes to (performance) portability
- ICON scientists are hesistant to leave the Fortran world
- Opening ICON for language interoperability allows to incrementally replace parts of the code by more portable non-Fortran solutions starting with the non-scientific parts of the code

SPONSORED BY THE

Federal Ministry of Education and Research

# References

Giorgetta, M. A., et al.: The ICON-A model for direct QBO simulations on GPUs (version icon-cscs:baf28a514), Geosci. Model Dev., 15, 6985–7016, https://doi.org/10.5194/gmd-15-6985-2022, 2022.

Hohenegger, C, et al.: ICON-Sapphire: simulating the components of the Earth system and their interactions at kilometer and subkilometer scales, Geosci. Model Dev., 16, 779–811, https://doi.org/10.5194/gmd-16-779-2023, 2023.

SPONSORED BY THE

Federal Ministry of Education and Research