Running ICON on Heterogeneous Architectures

DWD Experiences

Ulrich Schättler

Department for Numerical Modeling (FE1)

Deutscher Wetterdienst, Germany

With contributions from many colleagues!

Contents

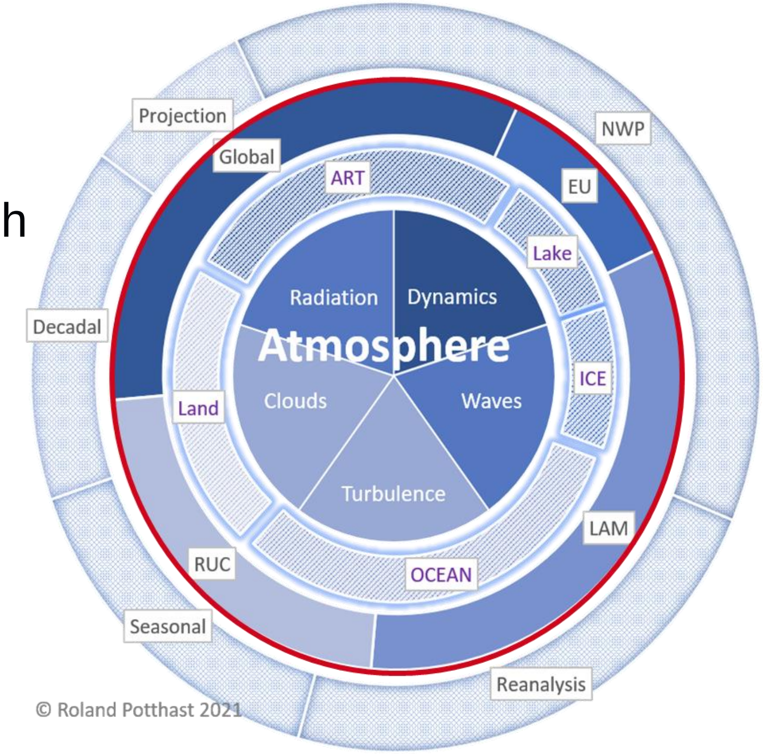
- ➔ Introduction to ICON
- ➔ ICON on NEC SX Aurora Tsubasa
- ➔ ICON on GP-GPUs

Introduction to ICON

ICON Modelling Framework

Started as a project between DWD (NWP) and the Max-Planck-Institute for Meterology (climate)

- ➔ Nonhydrostatic dynamical core on icosahedral-triangular Arakawa-C grid with mass-related quantities at cell circumcenters.
- ➔ Local mass conservation and mass-consistent transport
- ➔ Two-way nesting with capability for multiple overlapping nests.
- ➔ Limited-area mode available.
- ➔ Available components: NWP, climate, ocean, land.
- ➔ At the moment 2 different physics packages for NWP and climate mode.
- ➔ Runs in global NWP mode at DWD since January 2015 and in limited-area mode since February 2021.

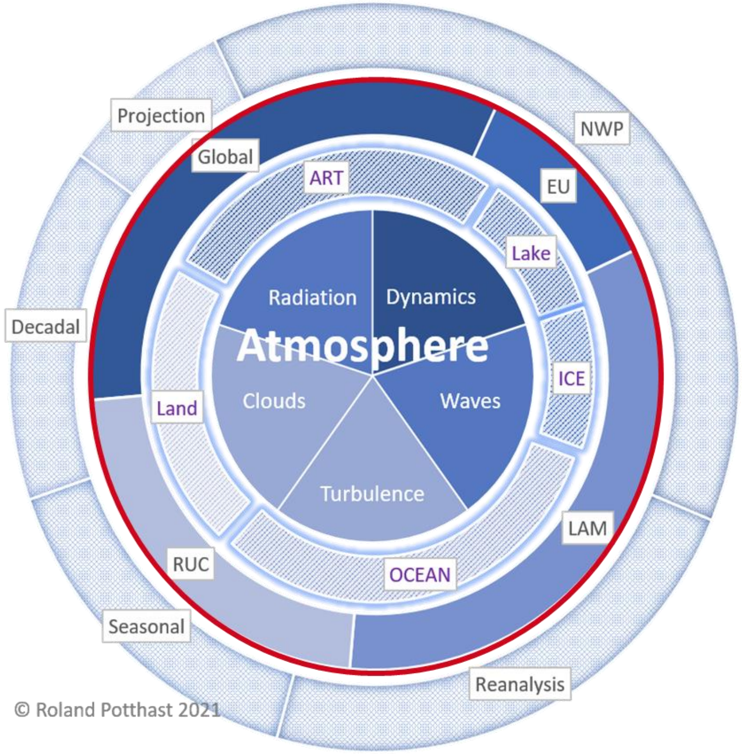


© Roland Potthast 2021

ICON Partners

Other development partners now are

- ➔ DKRZ: German climate computing center
- ➔ KIT: Karlsruhe Institute for Technology: provides the ART module (Aerosols, Reactive Trace gases and pollen)
- ➔ C2SM: Center for Climate Systems Modelling at ETH Zürich



© Roland Potthast 2021

Associated to the development groups are „key development partners“

- ➔ The COnsortium for Small-scale MOdelling (COSMO)
- ➔ The Climate Limited-area Modelling Community (CLM)

Some ICON Projects

ICON-Seamless:

- ➔ NWP plus climate forecasting with ICON-A-O-L and Data Assimilation.
- ➔ Initiated by the directors of DWD, MPI-M, KIT and DKRZ in October 2020.
- ➔ Expert Groups (Atmosphere, Ocean, Land, Data Assimilation) started work in 12/2020.

ICON-Consolidated: Software Redesign (Interfaces and Structures)

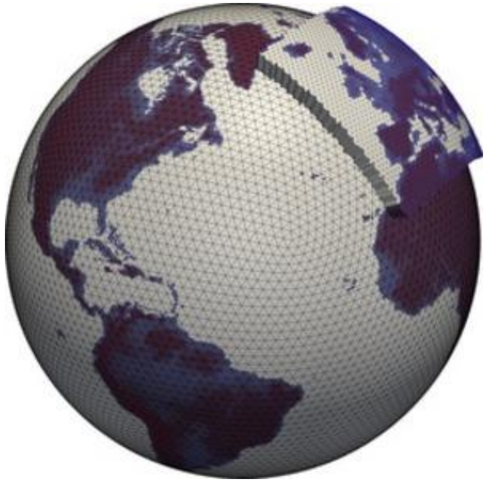
- ➔ Interface Redesign Workshop in Q4/2021
- ➔ Software Concept Design Workshop in Q4/2021

Technical Projects: („Computer Science“ Projects)

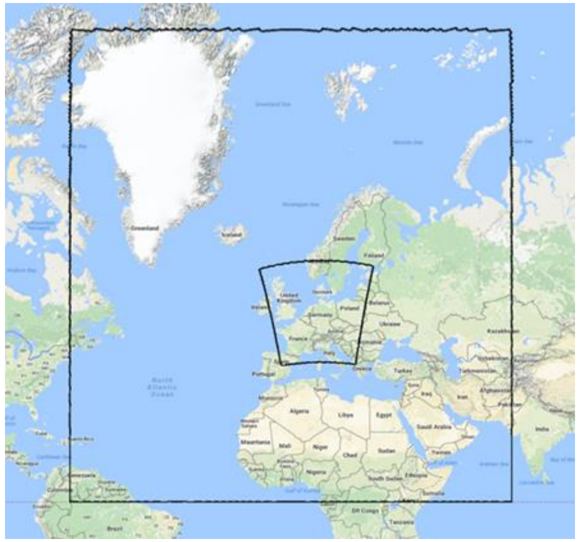
- ➔ PASC-ENIAC: (CH) Enabling the ICON model on heteorogeneous architectures.
- ➔ COSMO PP IMPACT: ICON on Massively Parallel Architectures.
- ➔ IAFE-GPU: DWD project to support ICON and Data Assimilation implementation on GPUs

ICON for NWP at DWD

- ➔ Since 2015 ICON is running at DWD as global model with a nest over Europe.



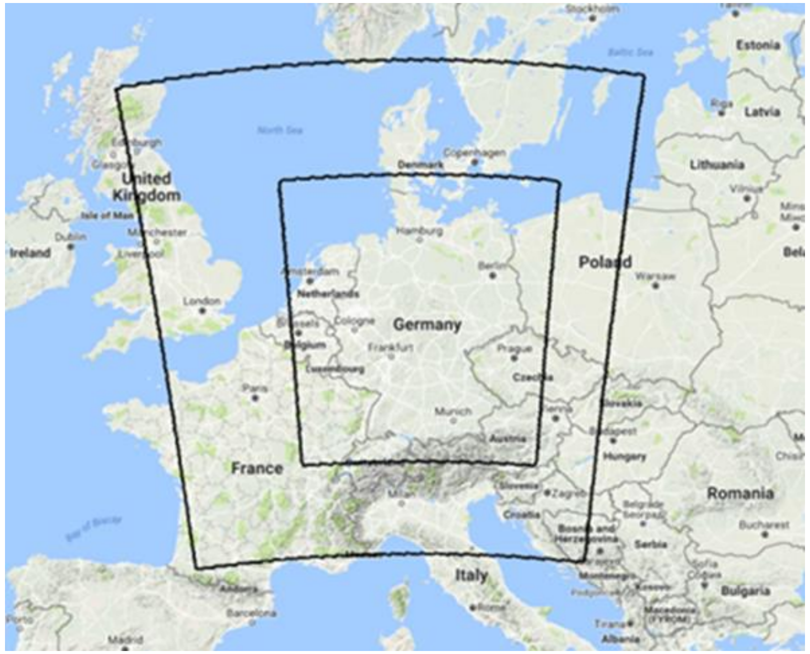
- ➔ The limited-area model of ICON (ICON-LAM) replaced the regional COSMO-Model in February 2021.



Current Work at DWD

Project SINFONY: Seamless INtegrated FOrecastiNg sYstem

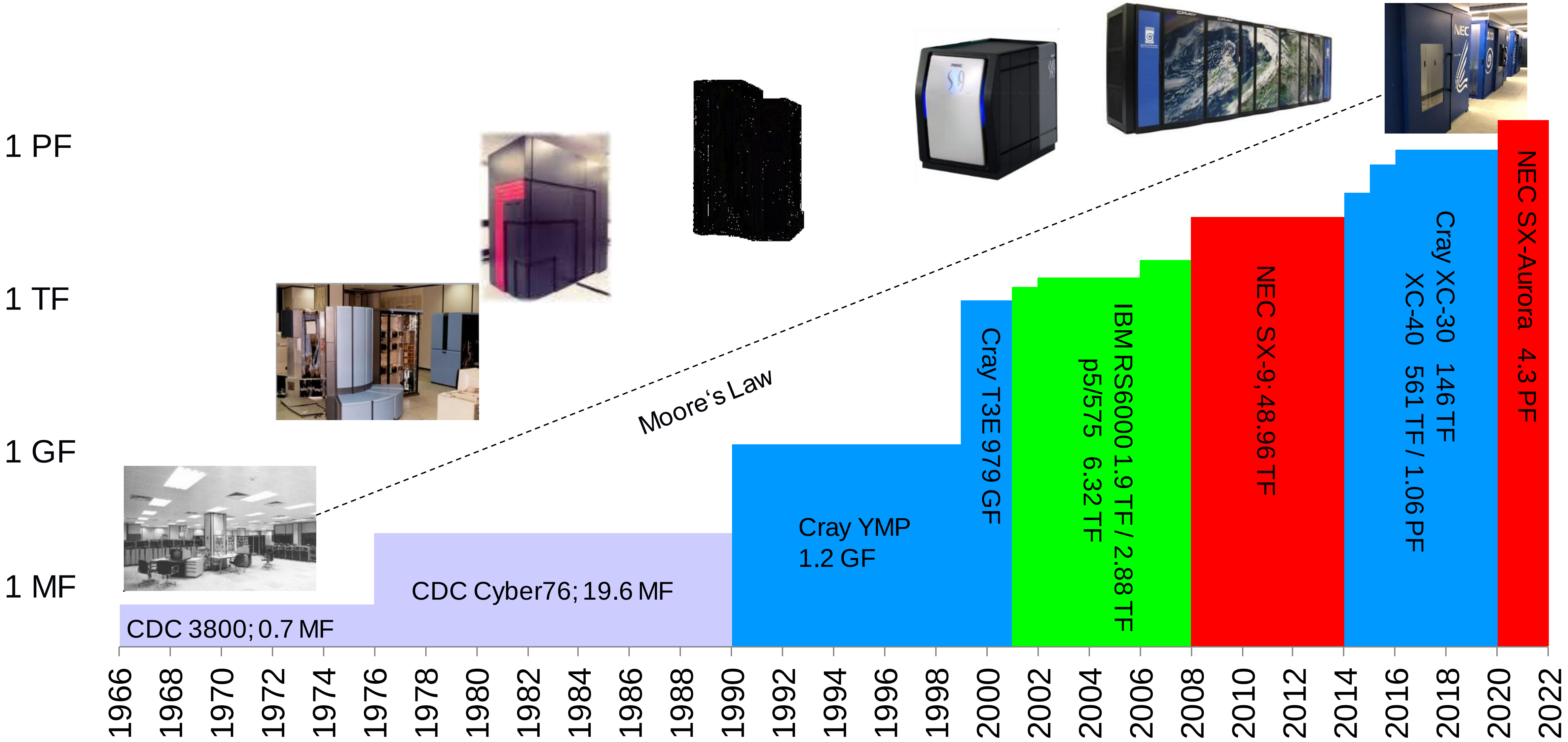
- ➔ DWD developes a forecasting system for the shortest range (0-12 hours). It shall provide a seamless forecast of the atmospheric state and severe weather phenomena (thunderstorms, precipitating cells). (https://www.dwd.de/EN/research/researchprogramme/sinfony_iafe/sinfony_start_en.html).
- ➔ It is planned to develop new combined products between nowcasting and NWP to provide new forecast products for up to 12 hours.
- ➔ The new NWP system will be based on an ICON-LAM ensemble with very high resolution (2 km with a 1 km nest included) and associated data assimilation.
- ➔ The products will be updated hourly by a Rapid-Update-Cycle.



very high resolution + ensemble + hourly update ⇒ needs enormous CPU power

ICON on NEC SX-Aurora Tsubasa

DWD Computers 1966-2021



Result of the Last Procurement

NEC SX-Aurora Tsubasa

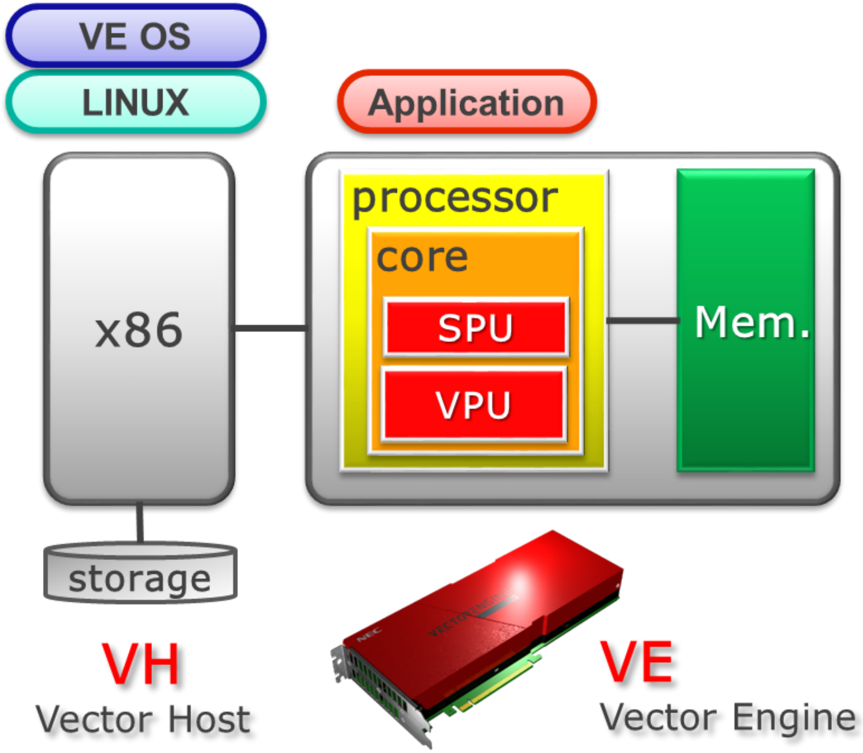
- ➔ Within our financial limits, NEC offered a total upgrade factor of 6.0 compared to former Cray XC 40. Competitors were way below 5 (measured with number of ICON ensemble members).
- ➔ At the same time, SX-Aurora Tsubasa has the lowest power consumption. Typical power expected for Phase 2 is 777 kW. Competitors were between 899 and 1060 kW (with lower upgrade factor).
- ➔ From the latest Green500 list (November 2021; Phase 1): 2 entries for SX-Aurora Tsubasa, NEC, Deutscher Wetterdienst (Research and Operational Cluster)

Rank in Green 500	Rank in TOP 500	#cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)	Efficiency (GFlops/W)
54	119	18,688	3,870.1	5,605.6	648	5.972
55	141	14,336	3,250.4	4,282.1	565	5.752

SX-Aurora Tsubasa

Building block is a node, which consists of

- ➔ a vector host (VH): scalar CPU (24-core AMD Rome; 2.8 GHz; 256 GB memory)
- ➔ 8 vector engines (VE): SX-Aurora 1 TSUBASA Typ 10AE (PCIe card) with
  - ➔ 8 vector cores: 1.5 GHz; 304.1 GF/s (DP); 608.3 GF/s (SP) per core
  - ➔ 48 GB HBM2 3D-stacked memory (6 GB / core; 1.35 TB/s bandwidth)



Source: NEC

Phase	Operational			Research			Avail able
	VH	VE	Cores	VH	VE	Cores	
0	178	1424	11392	232	1856	14848	12/2019
1	224	1792	14336	292	2336	18688	12/2020
2	325	2600	20800	424	3392	27136	12/2022

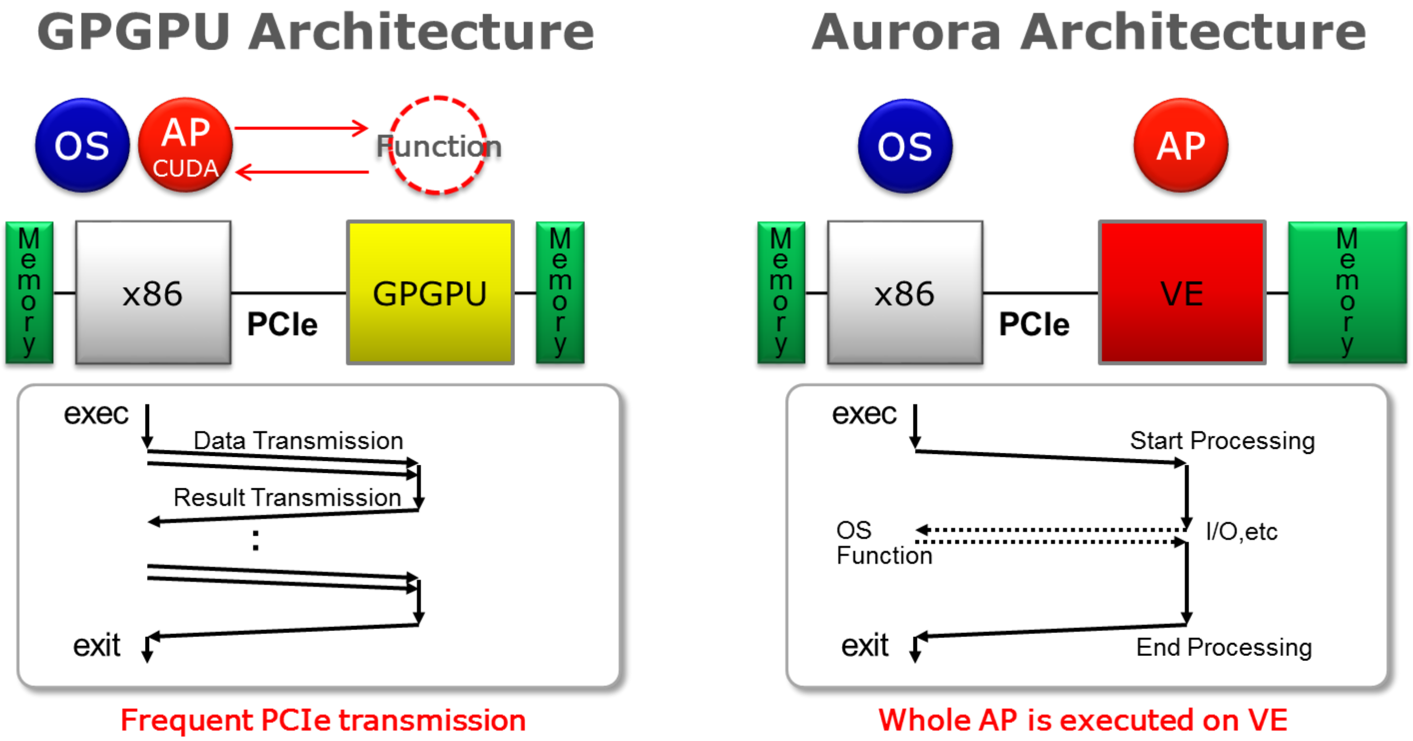
NEC‘s View

According to NEC, the Aurora architecture has these advantages:

- Avoiding frequent PCIe transmissions.
- Standard programming language.

Our assessment:

- Frequent transmissions can also be avoided on GPGPUs, if the whole application can run on the GPGPU (but depends on the application).
- The standard programming language is an advantage, but how can a parallel run on x86 and VE be realized?



Source: NEC

Software Stack and Other Details

- ➔ Standard Linux Distribution (Red Hat) on Vector Host
- ➔ VEOS: Program running on Linux/x86, providing OS functionality for VE applications running on Vector Engine:
  - ➔ utilizes Linux functionality for VE process / memory management, program loading, signal handling, etc.
  - ➔ works completely on Linux/x86: no OS jitter on VE
- ➔ MPI communication via Infiniband is directly executed between VEs, no memory copying to x86 memory. A proprietary NEC MPI implementation is used.
  - ➔ Can handle hybrid x86/VE execution on SX-Aurora, for example to offload I/O processes on x86 cores.
- ➔ Batch System PBS and Queueing System NQSV
- ➔ Compiler:
  - ➔ VE: NEC compiler (nfort, ncc): cannot create x86 code!
  - ➔ VH: Intel, GNU: cannot create VE code!

So how to offload I/O processes or eventually other program tasks?

Offloading vs. Task Parallelism

NEC provides libraries to offload parts of a program running on VE to x86 VH (libvhcall) and vice versa (libveo).

➔ Is used at DWD by some (smaller) programs to offload e.g. eccodes-calls to the x86 cores.

ICON uses task parallelism to run I/O processes on x86 (GNU binary) and compute processes on VE (NEC binary)

➔ MPI implementation of ICON already uses separate I/O tasks

```
mpirun -vh -node 0 -np 1     GNU_binary: \     # read initial data; write to stdout
                   -np 61    NEC_binary: \     # running the model
       -vh -node 0 -np 2     GNU_binary        # 1 for prefetching boundaries + 1 output task
```

ICON-D2, 6h forecast with hourly output:

	61 + 3 VE cores	61 VE + 3 x86 cores
Total Time	345.71	308.83
Model init	57.05	24.37
Output	64.76	6.18

Conclusions (NEC SX)

- ➔ DWD‘s operational NWP suite now is running on an energy-efficient powerful vector architecture.
- ➔ Running I/O tasks of ICON on the x86 vector host is done via task parallelism. The compute tasks are fully executed on the vector engines.
- ➔ Offloading I/O tasks to the x86 vector hosts by using library calls (libvhcall) is done by a few programs. For the assimilation code this is still under investigation.
- ➔ Offloading other tasks (for example non-vectorizable code) to the vector host is not done yet at DWD and might require a major rewrite.

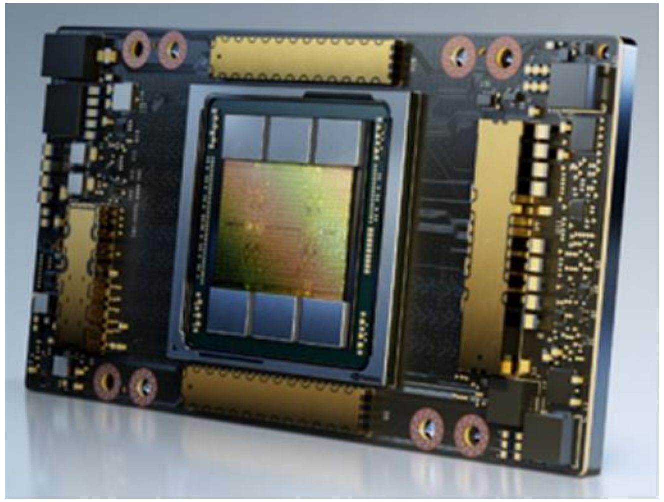
ICON on GP-GPUs

GPU Architecture

General Purpose GPU s have thousands of cores and promise high floating point performance.

➔ NVIDIA A100: 9.7 TF/s for double precision (19.5 TF/s for Tensor Core)

➔ To compare: Traditional Vector processors, as in NEC SX Aurora, have few powerful vector cores: 19.4 TF/s for double precision per node (64 vector cores)



A100 GPU Quelle: NVIDIA

But: GPU architecture is not suited to put MPI tasks / subdomains on a core!

➔ A GPU has not much memory per core, but many registers

Programming GPUs

Thousands of cores now add another level of parallelism:

- MPI: decomposition into subdomains.
- OpenMP: beneficial for blocking (on multi-core CPUs, vectors).
- OpenACC: how to exploit the GPU architecture? On a grid point level?

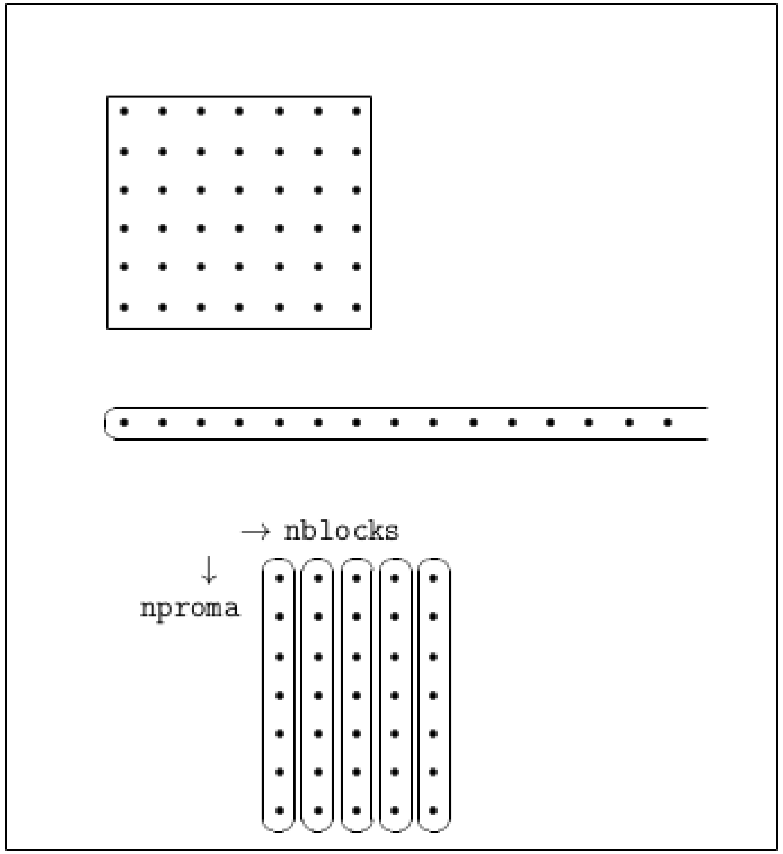
  (also possible since some time: OpenMP)

Data structures are important!

- Classical: Store a two-dimensional field as a two-dimensional array.
- ICON is different. It stores fields in an un-structured way, in principle all grid points in a long one-dimensional vector.
- But for cache efficiency a „blocking“ is introduced, resulting again in a two-dimensional structure: `field(nproma,nblocks)` with

  `nproma:` number of grid points in one block

  `nblocks:` number of nproma-blocks



Mapping to the Architecture

`nproma` can be chosen at the start of the program:

➔ Use a small value (8, 16, 32) for cache-based x86 architectures,

➔ Higher values (> 512) for vector processors.

➔ For GPUs you even need a longer vector!

  ➔ Best choice: `nblocks=1; nproma=#grid points`

```
!$OMP PARALLEL DO
!$acc parallel
!$acc loop seq
DO jb = 1, nblocks
  !$acc loop gang vector
  DO jv = 1, nproma
    field(jv,jb) = …
  ENDDO
ENDDO
!$acc end parallel
```

With the blocked data structure, ICON is suited for vector as well as for GPU architectures.

Le diable est dans les détails!

Acknowledgements for the GPU Work

Investigating GP-GPUs for NWP models has been started in Switzerland with the COSMO-Model about 10 years ago. DWD and other COSMO partners contributed only little to that work. Because of the „classical data structure“ of COSMO, the following actions had been taken:

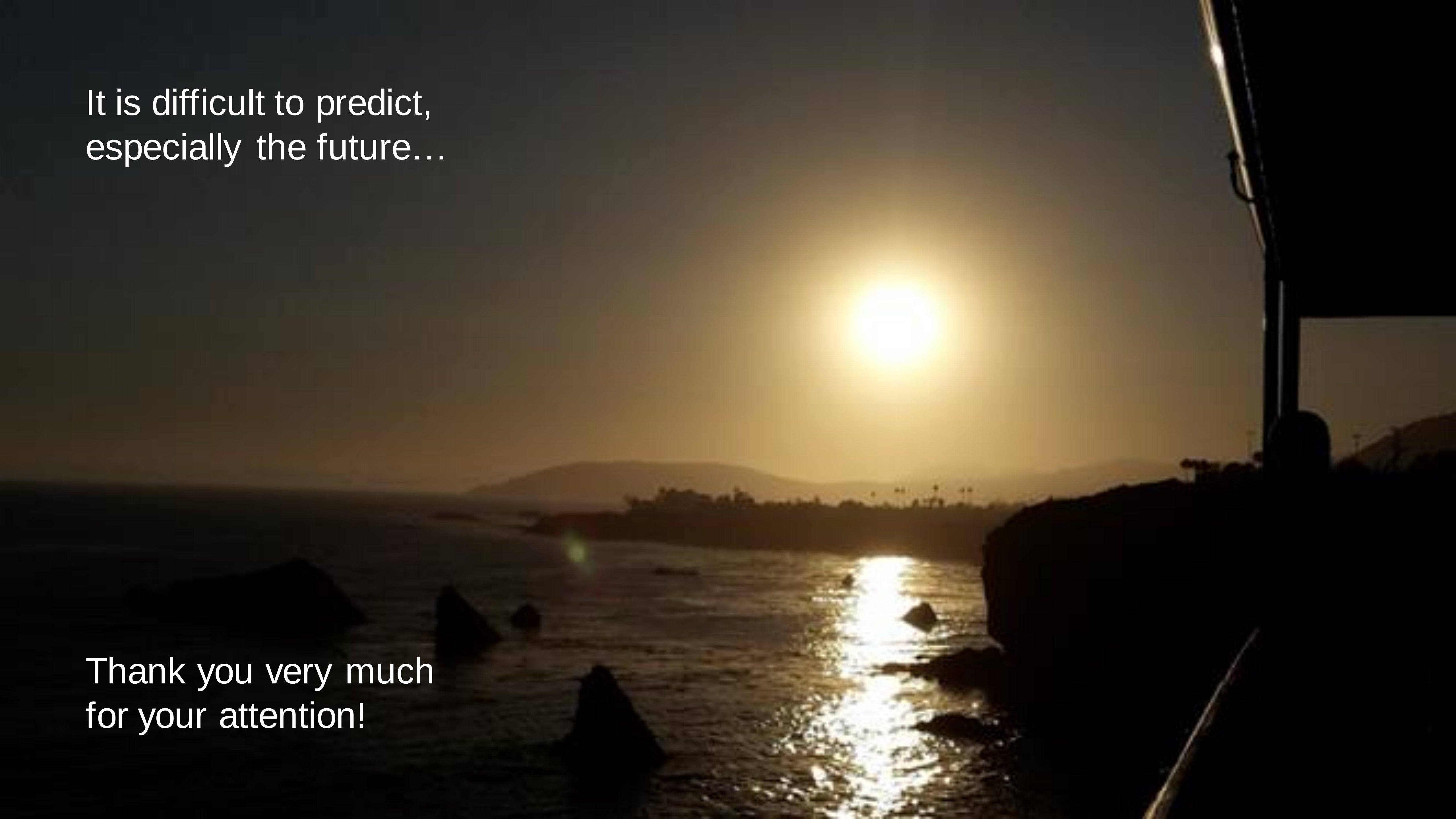
- ➔ Rewrite of the dynamical core using a new DSL GRIDTOOLS. Implementation of the DSL introduced blocking.
- ➔ Introducing a blocked data structure only for physical parameterizations (in this way we could use the same code for COSMO and ICON). These were ported to GPUs using OpenACC.
- ➔ The work was initiated by the Swiss Supercomputing Center (CSCS), which still supports MeteoSwiss.

Porting ICON to GPUs:

- ➔ Porting the dynamics started some years ago at CSCS, using OpenACC.
- ➔ Now, CSCS, MeteoSwiss, MPI-M and DWD are working together on the ICON port, but still only using OpenACC.
- ➔ Usage of GRIDTOOLS is under development at CSCS and MeteoSwiss (see EXCLAIM).

Conclusions (GP-GPUs)

- ➔ ICON‘s data structure is suited to port ICON to GP-GPUs.
- ➔ Right now, OpenACC is accepted by the domain scientists. But most of them are not yet able to do the porting. This is done by a special group now.
- ➔ Experiences, how this can work in the future, still have to be gathered.
- ➔ The DSL approach in COSMO was not really accepted by the main developers of COSMO.
- ➔ Now we are curious, how the new approach for ICON (see Xavier‘s presentation) will look like.



It is difficult to predict,
especially the future…

Thank you very much
for your attention!