esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Report on services offered on weather and climate benchmarks

Deliverable D3.4

The project Centre of Excellence in Simulation of Weather and Climate in Europe Phase 2 (ESiWACE2) has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 823988.

About this document:
Work package in charge: WP3 “HPC services to prepare the community for the pre-exascale”
Actual delivery date for this deliverable: 30.3.2023
Dissemination level: PU (for public use)

Lead authors:
Deutsches Klimarechenzentrum GmbH (DKRZ): Joachim Biercamp, Ralf Müller
Atos: David Guibert, Erwan Raffin
European Centre for Medium-Range Weather Forecasts (ECMWF): Sam Hatfield

Other contributing authors:
Atos: Loris Lucido
European Centre for Medium-Range Weather Forecasts (ECMWF): Peter Düben

Project internal reviewer:
Netherlands eScience Center (NLeSC): Ben van Werkhoven

Contact details:
Project Office: esiwace@dkrz.de
Visit us on: www.esiwace.eu

Access our documents in Zenodo:
https://zenodo.org/communities/esiwace

Table of contents

1. Abstract / publishable summary

The computational performance of supercomputers is usually expressed in terms of their theoretical peak performance, or measured with rather simple and easy to handle benchmark codes, like the High-Performance Linpack Benchmark, a measure of the floating-point computing power. These metrics allow to easily rank the performance of computers but have only limited value when it comes to the question of how well a computer system is suited for specific complex applications like weather and climate models.
The EU-funded project ESCAPE-2, which ran from 2019 to 2021, therefore developed a benchmark suite called High Performance Weather and Climate Benchmark, in short HPCW, which consists of widely used European weather and climate models and standalone model components (also called dwarfs in the ESCAPE-2 context) that are less complex than the models themselves but represent selected characteristics of the models. After the end of ESCAPE-2, the centre of excellence ESiWACE2 took over the responsibility to establish HPCW as a useful tool for the communities of weather and climate modellers on the one hand, and for computer manufacturers on the other hand. At the end of ESiWACE2 we provide a consolidated and extended version of HPCW that has been installed and tested on some of the largest high performance computing systems in Europe. The work will be continued in the third phase of the centre of excellence, ESiWACE3.

2. Conclusion & Results

In the last year of ESiWACE2, we took over the responsibility for the High Performance Climate and Weather Benchmark (HPCW) from the Horizon 2020 project ESCAPE-2, which ended in 2021. The rationale for the development of HPCW was to prepare weather and climate domain benchmarks that allow a much more realistic assessment of application-specific performance on large HPC systems than traditional benchmarks such as those used for ranking supercomputers in the Top500 lists, High-Performance Linpack and High-Performance Conjugate Gradients.

The benchmark includes the European community climate and weather models IFS, ICON and NEMO. If we want to investigate the suitability of a computer for climate research or weather forecasting, it is important to test and run full applications because of their complexity and flat computational profile. However, the deployment of these very complex codes is time-consuming and difficult, and often needs very large amounts of input data. In addition, access to these codes is often limited because of licence issues. HPCW therefore also includes a number of key functional model components that, even on their own, impose specific computational challenges to the system. The “dwarfs” are publicly available and easier to deploy than full models but allow to assess the impact of central characteristics and properties of a computer system on the performance of weather and climate codes.

While the work in ESCAPE-2 had focused on defining and assembling the benchmark suite, here we concentrated on consolidating the framework and on installing and testing it on a number of different computing systems, including the latest systems of ECMWF and DKRZ, the Swiss ALPS system, and finally LUMI, the only pre-exascale EuroHPC systems which was available in 2022.

A strong focus of our work lay on making HPCW easy to deploy and to monitor on different HPC systems using a continuous integration and testing environment. Setting up a suite of complex and large codes, including all components and their respective test cases, across HPC sites is a huge undertaking, which is not only technically challenging but also time-consuming. In particular, the full models are sensitive to hardware characteristics and often depend on features not well or not at all supported by all compilers. We therefore consider the implementation of the continuous integration and testing environment a big step forward for HPCW.

After the end of ESiWACE2 in March 2023, HPCW will be taken over and disseminated by the newly-funded ESiWACE3 project. Furthermore, we have good reason to hope that we will receive additional funding for HPCW in other EuroHPC JU projects.

3. Project objectives

This deliverable contributes directly and indirectly to the achievement of all the macro-objectives and specific goals indicated in section 1.1 of the Description of the Action:

Macro-objectives	**Contribution of this deliverable?**
(1) Enable leading European weather and climate models to leverage the available performance of pre-exascale systems with regard to both compute and data capacity in 2021.	YES
(2) Prepare the weather and climate community to be able to make use of exascale systems when they become available.	YES

Specific goals in the workplan	**Contribution of this deliverable?**
Boost European climate and weather models to operate in world-leading quality on existing supercomputing and future pre-exascale platforms	NO
Establish new technologies for weather and climate modelling	NO
Enhance HPC capacity of the weather and climate community	NO
Improve the toolchain to manage data from climate and weather simulations at scale	NO
Strengthen the interaction with the European HPC ecosystem	YES
Foster co-design between model developers, HPC manufacturers and HPC centres	YES

4. Detailed report on the deliverable

HPCW was developed in the H2020 project ESCAPE-2[1]. The general idea and status at the end of the ESCAPE-2 project was documented in a deliverable of ESCAPE-2 (Müller et al, 2021)[2]. HPCW includes not just two of the most advanced Earth system models in Europe, the IFS and ICON, but also a range of standalone model components as well as ocean models. HPCW abstracts away many of the difficult steps required for benchmarking each of the individual components. In this way the user does not need to worry about data downloading, preparation, job scheduling, or validation.
In the present document we describe the work done to maintain and enhance HPCW resourced by ESiWACE2 since November 2021. This includes consolidating the build system, including a number of new components and implementing and testing parts of HPCW on HPC systems that were not yet available to ESCAPE-2. The result is a solid framework ready for exploitation. As a next step we need to evolve the strategy with respect to ownership, licensing, dissemination and the target audience of HPCW to make it a useful product that will be adopted by the community. This will be the focus of the work on HPCW in the first stage of the follow-on project ESiWACE3.

4.1 HPCW v1.0

HPCW v1.0 is described in Müller et al., 2021. It contains the following components:

- Models
  - ICON: (**Ico**sahedral **N**onhydrostatic) ocean and atmosphere model, used in Weather prediction and Climate research
  - ICON atmosphere version with support for GPU
  - IFS: the **I**ntegrated **F**orecasting **S**ystem of ECMWF (RAPS benchmarking version)
  - NEMO: standing for "**N**ucleus for **E**uropean **M**odelling of the **O**cean" is a state-of-the-art modelling framework for research activities and forecasting services in ocean and climate sciences
- “Dwarfs” (mini-apps)
  - IFS-FVM: The IFS atmosphere FV (finite volume) dwarf
  - Radiation dwarf ACRANEB2: A module developed by ECMWF for computing radiation schemes.
  - ICON ocean advection dwarf
  - ecRad: an atmospheric radiation scheme designed for computing profiles of solar (or shortwave) and thermal infrared (or longwave) irradiances from the surface up to the middle mesosphere
- Kronos Workload Simulator

HPCW v1.0 defines the following test cases:

- ICON ocean: Small (160km), Medium (40km) and Big (10km), 3-points on a strong scaling line
- ICON atmosphere: Small (160km)

[1] https://www.hpc-escape2.eu/about
[2] https://www.hpc-escape2.eu/resources/d35-full-hpcw-suite-v10

- ICON atmosphere GPU: Small (160km)
- ICON coupled atmosphere and ocean (160km)
- ICON ocean advection dwarf: Small
- IFS (RAPS): Small (TL159), Medium (TCo639), Big (TCo1999)
- NEMO: Small BENCH1 (ORCA1 like), Medium (ORCA0,25)
- IFS-FVM: Small (O160), Medium (O640), Big (O1280)
- ACRANEB2: Small
- Kronos: Single-serial, single-parallel, multi-serial-events, external-job
- ecRad: Small

4.2 HPCW v2.0

After transferring the responsibility for HPCW from ESCAPE-2 to ESiWACE2, a number of improvements have been made, which we outline below.

New components have been added:

- **ecTrans**: A module developed by ECMWF for performing efficient spectral transforms of global meteorological fields in a distributed memory context. ecTrans is capable of transforming fields between grid point space (e.g., on a Gaussian grid) and spectral space (used internally by the IFS). It consists of compute- and communication-heavy routines and is therefore an excellent stress-test for any machine. ecTrans was recently released as an open source library.
- **CLOUDSC**: A cloud microphysics parameterization scheme developed by ECMWF. CLOUDSC is considered an archetypal physical parameterization scheme, operating on a globe of atmospheric columns with no horizontal dependency, and is therefore a common target for optimisation and porting efforts intended for parameterization schemes as a whole. CLOUDSC contains no communication routines and is therefore a useful test-bed for CPU- or GPU-local optimisation approaches such as vectorisation. CLOUDSC was recently released as an open source library.
- **ecRad**: An atmospheric radiation scheme designed for computing profiles of solar (or shortwave) and thermal infrared (or longwave) irradiances from the surface up to the middle mesosphere was already part of HPCW v1.0 but has been updated to a newer version (9/04/2021 -> 13/07/2022) now.

HPCW v2.0 now supports recipes to build HPCW components with SPACK, a package manager that is widely used nowadays. SPACK is developed with HPC in mind and serves major HPC sites around the world (European sites, US sites, as well as Japan (e.g. Fugaku)). It allows HPCW to also benchmark models that are already installed on a cluster, if the model's binary code is compatible with the input data provided by HPCW, or it allows to compare models used in production with models that could be compiled with other compiler flags, dependency versions, etc.

We have also updated a number of dependencies on external libraries.

The HPCW repository contains a file CHANGELOG.md, which summarises the changes, the fixes and the new components that are available.

4.3 Implementations

The HPCW suite has been installed and tested on a number of HPC systems. The final goal of work package 3 of ESiWACE2 was to port and run HPCW benchmarks on the pre-exascale EuroHPC systems. Unfortunately only one of those systems, LUMI, became available towards the very end of the project.
Most of the work was therefore naturally carried out on systems owned by the developers of HPCW, i.e. Atos, DKRZ and ECMWF, all ranking among the largest systems in Europe. We also started to use the Swiss ALPS system as a "stand-in" for the pre-exascale systems, but once it became clear in December 2022 that the use of LUMI was possible for the team, we concentrated on that new system.
The results presented here were run on various architectures and compiler toolchains. New ones can be tested by only adding a small CMake toolchain and compiler shell environment. The best performance can be obtained when the build system used by the project allows customization of compilation options such as vectorization options (AVX-256, AVX-512, etc.).

4.3.1 Atos in-house systems

On the Atos in-house systems, all the models have been compiled and tested, some with Intel compilers, some with NVHPC compilers (where GPU support is required) either with Intel MPI or OpenMPI. For the tests, energy measurements have been obtained with the Atos' Energy Optimizer but it was not available on every compute node.

Depending on which information is printed in the output log, we can obtain interesting information such as performance results (flops) or accuracy results. This, however, needs to be supported and maintained by each model.
HPCW has been fully tested on Atos in-house clusters for different CPU architectures. Table 1 shows results from the AMD Milan 7763 CPU compiled with the Intel toolchain, with the exception of the GPU version of the CLOUDSC dwarf, which ran on Nvidia V100 GPU and was compiled with NVHPC 21.2. Results from AMD Genoa (9654), Intel Icelake (8358) and ARM Ampere (Q8030) with armclang 22.1 are given in Tables 2, 3 and 4, respectively.

These results demonstrate that HPCW v2.0 can be fully deployed on a system, but they also show that customization needs to be done with care to obtain the optimum performance. Indeed, obtaining the optimum performance could lead to a high specialisation of the build and run recipes for a particular component, which can be different for another component. For example, finding the most efficient compiler, compiler flags, libraries, MPI-OpenMP combination, etc. for each component implies a huge optimization space to be explored. This exploration has not been performed for the presented performance figures in Tables 1-4.

project name	revision	status	time (s)	time_app	gflops	error (%)	energy (Wh)
dwarf-p-radiation-acraneb2-lonlev-0.91-small	v2.0	OK	0.17				0.14
dwarf-p-radiation-acraneb2-lonlev-0.9-small	v2.0	OK	9.69				1.04
dwarf-p-cloudsc-fortran	v2.0	OK	6.6		80.76		0.8
dwarf-p-cloudsc-gpu-claw	v2.0	OK	18.95		33.1		2.75
dwarf-p-cloudsc-gpu-scc	v2.0	OK	17.46		124.91		2.49
dwarf-p-cloudsc-gpu-scc-hoist	v2.0	OK	17.09		216.21		2.49
ecrad-small	v2.0	OK	0.45	0.19439			0.14
ectrans-all	v2.0	OK	342.49	342.47			23.92
icon-atmo-small	v2.0	OK	45.35				6.18
icon-coupled-small-n24	v2.0	OK	359.11				61.86
icon-ocean-advection-dwarf	v2.0	OK	9.01				0.85
icon-ocean-big	v2.0	OK	3398.65				26081.9
icon-ocean-medium	v2.0	OK	736.11				1089.03
icon-ocean-small	v2.0	OK	15.26				1.93
icon-atmo-gpu-small	v2.0	OK	41.03				5.5
ifs-fvm-big	v2.0	OK	10988.47				212434.49
ifs-fvm-medium	v2.0	OK	3661.08				35117.33
ifs-fvm-small	v2.0	OK	1086.99				180.46
ifs-tco1999-big	v2.0	OK	1529.81	1483.607		0.89732	21460.31
ifs-tco639-medium	v2.0	OK	363.23	357.879		0.40571	480.01
ifs-tl159-small	v2.0	OK	21.03	18.043		1.1787	2.64
nemo-orca25-medium	v2.0	OK	1727.84				711.55
nemo-bench-orca1-like-small	v2.0	OK	351.13				26.5

Table 1: Results of all HPCW components obtained on AMD Milan CPUs (2x64 cores per node).

project name	revision	status	time (s)	time_app	gflops	error (%)
dwarf-p-radiation-acraneb2-lonlev-0.91-small	v2.0	OK	0,65			
dwarf-p-radiation-acraneb2-lonlev-0.9-small	v2.0	OK	8,49			
dwarf-p-cloudsc-fortran	v2.0	OK	7,02		87,8	
ecrad-small	v2.0	OK	1,3	0,16663		
ectrans-all	v2.0	OK	283,56	283,52		
icon-atmo-small	v2.0	OK	49,28			
icon-coupled-small-n24	v2.0	OK	338,79			
icon-ocean-advection-dwarf	v2.0	OK	29,59			
icon-ocean-medium	v2.0	OK	759,27			
icon-ocean-small	v2.0	OK	33,16			
ifs-fvm-small	v2.0	OK	877,48			
ifs-tco639-medium	v2.0	OK	330,11	322,109		0,50357
ifs-tl159-small	v2.0	OK	17,96	13,822		1,37226
nemo-orca25-medium	v2.0	OK	997,35			
nemo-bench-orca1-like-small	v2.0	OK	377,39			

Table 2: Results of HPCW components obtained on AMD Genoa CPUs (2x96 cores per node).

project name	revision	status	time (s)	time_app	gflops	error (%)
dwarf-p-cloudsc-fortran	v2.0	OK	8,08		47,84	
ecrad-small	v2.0	OK	0,59	0,1703		
icon-atmo-small	v2.0	OK	46,98			
icon-coupled-small-n24	v2.0	OK	323,77			
icon-ocean-advection-dwarf	v2.0	OK	7,57			
icon-ocean-medium	v2.0	OK	663,19			
ifs-tco639-medium	v2.0	OK		540,206		0,6667
ifs-tl159-small	v2.0	OK	74,1	49,489		1,09193
nemo-orca25-medium	v2.0	OK	924,66			
nemo-bench-orca1-like-small	v2.0	OK	96,5			

Table 3: Results of HPCW components obtained on Intel Icelake CPUs (2x32 cores per node).

project name	revision	status	time (s)	time_app	gflops
dwarf-p-cloudsc-fortran	v2.0	OK	10,89		42,76

Table 4: Results of HPCW components obtained on ARM Ampere CPUs (1x80 cores per node).

Supporting the ARM compiler is challenging as most models or dwarfs do not support it in their build systems. Most of the models need to be updated before HPCW can support them on ARM nodes. This will probably be addressed in ESiWACE3.

4.3.2 DKRZ

The new HPC system at DKRZ called “Levante” is equipped with AMD Milan CPUs (type 7763) as well as Nvidia A100 GPUs. Due to ongoing stability problems with the GPU software stack, the HPCW components were deployed using CPUs only. Levante offers MPI implementations from Intel and OpenMPI, from which the latter was used due to performance benefits. For compilation the Intel compiler from 2022 was chosen.
Table 5 presents the latest performance numbers for all successful tests. All other components not listed in the table were compiled, but could not be run for technical reasons: Intel compiler-errors for NEMO and network errors working with ICON, IFS and IFS-FVM in medium and high resolution. Solving issues of that type takes a lot of time and in this case we decided against it given the deadline of this deliverable. On the other hand these are typical problems when porting weather and climate codes that can have very specific needs regarding communication. Unlike many other benchmarks, HPCW comes with a large variety of models to cover climate and weather applications. Porting the complete package is a considerable amount of work, but this is a consequence of the domain it is supposed to represent. HPCW offers a big toolbox to test high-performance systems with real world applications. This is what sets it apart from other benchmarks like High-Performance Linpack or STREAM.

project name	revision	status	time (s)	time_app	gflops
dwarf-p-radiation-acraneb2-lonlev-0.91-small	v2.0	OK	0,19		
dwarf-p-radiation-acraneb2-lonlev-0.9-small	v2.0	OK	9,16		
dwarf-p-cloudsc-fortran	v2.0	OK	1,42		6,31
ecrad-small	v2.0	OK	1,12	0,18731	
ectrans-all	v2.0	OK	94,53		
ifs-tl159-small	v2.0	OK	23,26	18,669	
ifs-fvm-small	v2.0	OK	834,57		
icon-auto-atmo-small	v2.0	OK	64,47		
icon-auto-ocean-advection-dwarf	v2.0	OK	3,82		
icon-auto-coupled-small-n24	v2.0	OK	66,51		
icon-auto-ocean-small	v2.0	OK	10,31		
icon-auto-ocean-medium	v2.0	OK	357,92		

Table 5: Results of HPCW components obtained on Levante (AMD Milan CPUs (2x64 cores per node)).

Thanks to the new opportunity to build components with SPACK, it was possible to build all components on the CPU partition of Levante within a 1-day-hackathon. This was a very important exercise because it proved the usability of the new approach to building the components of HPCW and their dependencies. DKRZ uses SPACK for building and distributing the software stack to the users, and HPCW proved to fit into that very well. Major parts of the system installation could be re-used instead of building them once again for HPCW.

4.3.3 ECMWF

HPCW has been deployed on the newly-procured HPC system at ECMWF, an Atos BullSequana XH2000. This system consists of 4 clusters each of about 2,000 nodes. Each node is equipped with two AMD Rome CPUs. This generation is older than those of the in-house Atos and DKRZ machines introduced in previous chapters, so we should expect the performance results to be slightly poorer. Porting the full HPCW suite to the ECMWF Atos machine took a modest time of around 20 person-hours, that is, two days with two people. Compilation of the various models and components was simple, but preparing the large datasets required to run the high resolution model tests took a significant amount of time.

The results are summarised in Table 6. The majority of the tests worked out-of-the-box, which is a good demonstration of the portability of HPCW. Most of the tests that failed were of a medium or large size, and these failed due to a known limitation of the MPI library on the ECMWF system. Further time and tuning by engineers at ECMWF would easily remedy these.

project name	revision	status	time (s)	time_app	gflops
dwarf-p-radiation-acraneb2-lonlev-0.91-small	v2.0	OK	0,76		
dwarf-p-radiation-acraneb2-lonlev-0.9-small	v2.0	OK	8,32		
dwarf-p-cloudsc-fortran	v2.0	OK	1,59		6,66
ecrad-small	v2.0	OK	0,79	0,17778	
ectrans-all	v2.0	OK	64,18		
icon-auto-atmo-small	v2.0	OK	33,91		
icon-auto-coupled-small-n24	v2.0	OK	38,89		
icon-auto-ocean-advection-dwarf	v2.0	KO			
icon-auto-ocean-medium	v2.0	KO			
icon-auto-ocean-small	v2.0	OK	24,6		
ifs-fvm-big	v2.0	KO			
ifs-fvm-medium	v2.0	KO			
ifs-fvm-small	v2.0	OK	974,73		
ifs-tco1999-big	v2.0	KO			
ifs-tco639-medium	v2.0	OK	40,87	0,231	
ifs-tl159-small	v2.0	OK	18,73	0,228	
nemo-orca25-medium	v2.0	OK	1021,11		
nemo-bench-orca1-like-small	v2.0	OK	194,4		

Table 6: Results of HPCW components obtained ECMWF's AMD Rome CPUs (2x64 cores per node). Status: KO means that the test failed.

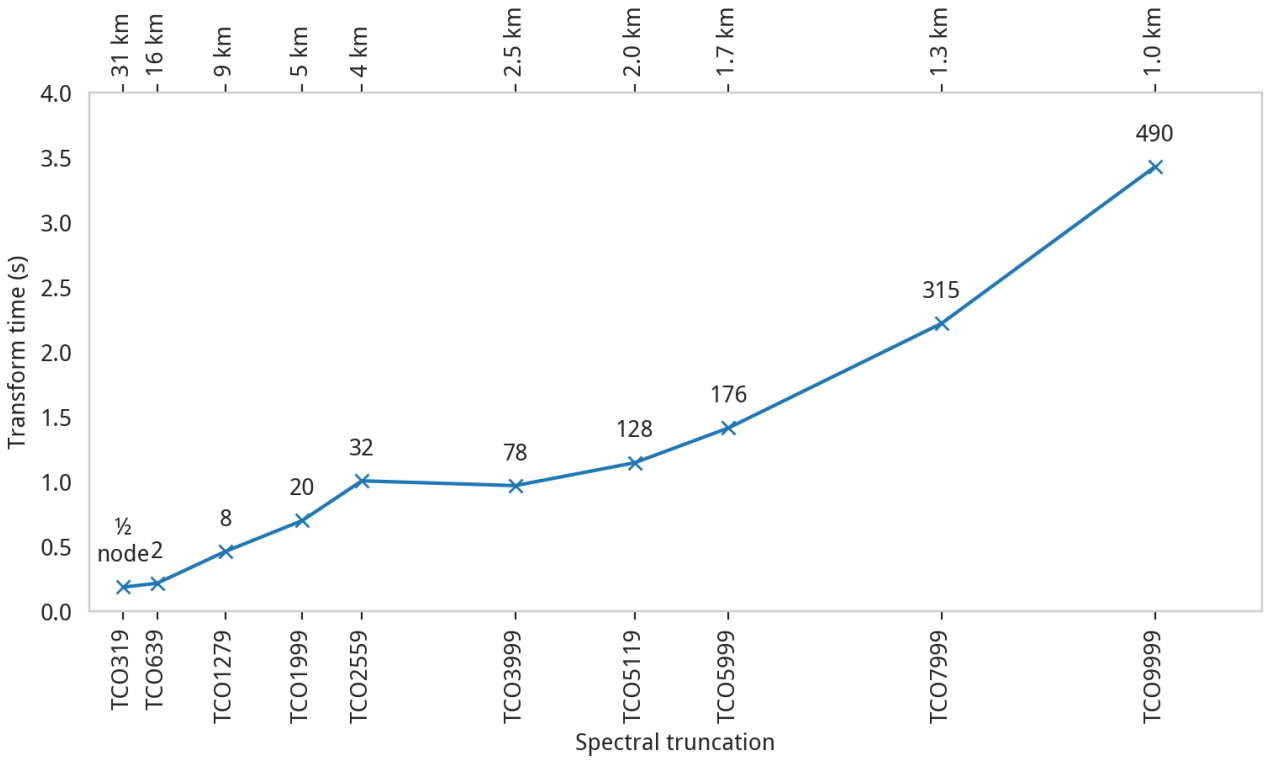

Figure 1: A weak scalability test of ecTrans on the ECMWF HPC system.

A good example for the exploitation of HPCW is the ability to perform scalability tests with defined components. Now that ecTrans has been added to HPCW, as of version 2.0, we are in a position to perform such tests in a weak-scaling manner and Figure 1 gives an example, carried out on the ECMWF Atos system. Here we measure the median transform time (in seconds) for transforming a field between spectral space and grid point space ten times. This mimics the main loop of a full numerical model. We carried this out across a range of resolutions, from TCO319 (about 31 km) to TCO9999 (about 1 km), making sure to scale the allocation of compute nodes such that it is approximately proportionate to the square of the spectral resolution. This, in principle, ensures that the amount of work per node is constant.
As can be seen, the transform time increases with resolution, even though the amount of work per node is intended to be constant. This could be because the square of the spectral truncation is a poor proxy for the amount of work, and that other dimensions not considered also scale with resolution. It could also be because the weak scaling test does not really capture elements such as the communication cost, which is a substantial factor at high resolutions. Consider that at TCO9999, just one of the four MPI transposition routines in ecTrans takes 70% of the total wall clock time. This routine is purely communication. Overall we can be satisfied that ecTrans provides an efficient solution for spectral transforms even at extremely high resolutions. Going forward with HPCW, we expect that ecTrans will be one of the components most easy to port to new systems yet still remaining computationally demanding.

4.3.4 ALPS

ALPS is the upcoming HPC system of CSCS. It is a virtual HPC system offering different backends like a Linux-based system with a Cray software stack and an ARM-equipped one running in the Amazon Cloud. At the time ALPS was chosen for deployment it was unclear if access to LUMI could be granted in time so that the deadline of the deliverable could be met. ALPS' first mentioned platform is very similar to LUMI, so at that time it seemed to be a valuable choice. Luckily, access to LUMI was possible later on and those results are reported in the next section.
Nevertheless it was possible to build and run the Fortran version of the CLOUDSC dwarf on both platforms of ALPS. Unfortunately the scripting necessary to obtain similar benchmark results as on the other systems could not be set up in time.
Virtualization is a new approach bringing HPC to the users, which makes ALPS an interesting and modern system to check HPCW with. Unfortunately Swiss institutions could not become partners of the ESiWACE3 project, but we hope to find other ways to continue the collaboration and use ALPS for HPCW tests.

4.3.5 LUMI

HPCW has been deployed on LUMI. It provides a CPU partition and a GPU partition with AMD MI250X GPUs. LUMI is an HPE Cray EX system and provides software stacks available through the LUMI modules and the CrayEnv module. It is advised to use the Cray programming environment.
HPCW has been updated to support LUMI, mainly by creating customised files (a toolchain file, an environment file and a job launcher).

Behind the Cray Programing Environment (CPE), different compiler suites are accessible: PrgEnv-cray (CCE), PrgEnv-amd (ROCm compilers for GPU partition), PrgEnv-gcc (GNU compilers) and PrgEnv-aocc (AMD compilers).
The first step has been to deploy HPCW using PrgEnv-gcc which had already been tested on the Atos in-house clusters. Table 7 shows the obtained results on some test cases on the CPU partition.

project name	revision	status	time (s)	time_app	gflops
dwarf-p-radiation-acraneb2-lonlev-0.91-small	v2.0	OK	0.49		
dwarf-p-radiation-acraneb2-lonlev-0.9-small	v2.0	OK	8.85		
dwarf-p-cloudsc-fortran	v2.0	OK	1.14		12.16
ecrad-small	v2.0	OK	1.2	0.12491	
ectrans-all	v2.0	OK	54.48		
nemo-bench-orca1-like-small	v2.0	OK	81.56		

Table 7: Results of HPCW components obtained on LUMI's AMD EPYC 7763 "Milan" CPUs (2x64 cores per node).

To use the Cray Compiling Environment (CCE, PrgEnv-cray), the build systems of each model needed to be adapted. ecRad already supported it and HPCW has been adapted to allow building it with the Cray compiler.
Regarding porting HPCW to the GPU partition, it also requires that each model is able to be compiled by either the Cray compiler or the ROCm compilers. The developers of the model and dwarfs need to implement that in the build system and adapt the code to target the AMD MI250X GPUs with approaches like OpenACC or OpenMP directives or HIP. This is a substantial undertaking and so far the only components that support such a hardware and software environment are CLOUDSC, and ecTrans. With the latter, for example, a version using OpenACC-decorated Fortran with HIP kernels to handle the Fourier and Legendre transforms is under development.

4.3.6 Interpreting the results

A comparison of the results in Tables 1-6 demonstrates the difficulty of directly comparing different machines. For example, the coupled version of the small ICON test case ran on the ECMWF machine only slightly longer than the atmosphere only version, which is consistent with the runtime of the ocean only testcase. On the ATOS systems, however, the coupled version was much slower, which hints at a configuration problem.
A naïve interpretation of the results would conclude that the AMD Milan partition of Atos was slower than the ECMWF system, despite that the latter uses the earlier Rome generation of AMD CPUs. This is of course inconsistent. In reality, the differences likely arise due to differences in how much resources were allocated to each benchmark for the two systems. In these two cases, a more consistent comparison is achievable by, perhaps, fixing the number of MPI ranks and OpenMP threads to be equal between the two systems. But if one has idle CPU cores as a result, then that machine would no longer be fully exploited. This would also complicate the comparison with other machines based on, say, ARM CPUs. In general, we therefore suggest caution when comparing results from different systems. Going forward, we will think carefully about how consistent cross-system comparisons can be constructed with HPCW.

It is, however, the number of already supported systems and codes that raises confidence in investing time into the optimization of the most stable components and publishing the numbers for reference. A database like this will need a lot more information about the specific test as well as the environment (node layout, memory, file system).

4.4 Development and Testing

HPCW is developed using a Git repository hosted at DKRZ[3]. Previously it was hosted at ECMWF[4]. Both repositories are still in use and synchronised. The main repository will become the one hosted at DKRZ. An account is required for now. Once obtained, the framework can be downloaded. The input files are hosted at DKRZ and their URLs are provided upon user request.

Some parts of HPCW, like IFS and ICON, are not yet openly available, but require a non-disclosure agreement (NDA). We plan to split HPCW into an open-source core containing all the already open-sourced dwarfs and models like NEMO, and an extension of HPCW with the models or dwarfs under NDA. This aspect will be addressed in ESiWACE3.

HPCW ships a lot of different components, which are supposed to run on a variety of platforms. Before being able to tune or optimise any code it needs to be built and run. Hence portability is HPCW's very first task to solve. This cannot be guaranteed by manual building and testing given the number of possible combinations of hardware and software components. That is why it was decided to set up a continuous integration and testing system for HPCW at DKRZ. It is tightly connected to the hosting GitLab service and includes compile and run tests on different HPC sites. At the moment DKRZ and Atos contribute compute resources to the test runs, and all components are separately tested on both systems.
DKRZ has confirmed the feasibility to add CSCS's ALPS system as well as LUMI and the Japanese Fugaku system. In the context of upcoming projects, these and possibly further HPC centres will be invited to contribute compute resources so that the HPCW benchmark can be tested on a large variety of platforms in a continuous manner. This will ensure a stable development environment for HPCW in the future.
In fact it is essential to consistently add new systems, because the average life cycle of a HPC system is only 3-5 years, whereas the software can exist for decades. Each HPC machine is a unique item combining hardware and software components, so there is no option to test only one to cover them all.

4.5 Outlook

With HPCW v2.0 we now have a stable and easily deployable version of a complex domain-specific benchmarking suite, which is an achievement on its own. As a next logical step, we need to implement mechanisms and define metrics for consistent cross-system comparison.

[3] https://gitlab.dkrz.de/esiwace/hpcw
[4] https://git.ecmwf.int/projects/ESCAPE/repos/hpcw/browse

A useful metric will include parameters relating the use of resources to time-to-solution such as energy cost per simulated year, or simulated years per calendar day per core. For example, by specifying the maximum amount of energy (in Joules) that may be used to simulate a predefined period of time, the wall clock time required to simulate one year can be used as a performance indicator to compare different systems. A detailed discussion of how to assess the computational performance of climate codes is given by Balaji et al, 2017.

In addition to these global numbers, the standard output of HPCW runs will be additional values to characterise the runs and their environment. This should system parameters affecting performance such as memory bandwidth, clock speed and network performance and also configuration parameters such as the number of used cores, the ratio of OpenMP threads to MPI tasks or selected model parameters like NPROMA[5].

These configuration parameters in general have a significant impact on performance (in the sense of time to solution) and we plan to enhance the documentation with best practices on how to configure and tune the components for different architectures without changing the source code. We also plan to define reference cases and to set up a platform where results can be published and assessed in a unified way.

Finally, an important task for ESiWACE3 will be to revisit the scope of HPCW and evaluate governance and funding models. This includes discussion on the completeness and possible extension (or reductions) of HPCW. To ease adoption of the benchmark by vendors, licensing issues have to be addressed and the adoption of open source models for as many components as possible needs to be fostered. Currently it is envisaged to bring the dynamical core of the Japanese model NICAM - which is open source - into HPCW and to port HPCW to the top Japanese machine, Fugaku.

5. References

Mueller, Ralf (DKRZ), David Guibert (BULL), Erwan Raffin (BULL), Mario Acosta (BSC), Daniel Beltran (BSC): **D3.5 Full HPCW suite v1.0**. *Deliverable of project ESCAPE-2*, funded from the European Union's Horizon 2020 research and innovation programme under grant agreement No 800987; https://www.hpc-escape2.eu/resources/d35-full-hpcw-suite-v10, 2021.

Balaji, V., Maisonnave, E., Zadeh, N., Lawrence, B. N., Biercamp, J., Fladrich, U., Aloisio, G., Benson, R., Caubel, A., Durachta, J., Foujols, M.-A., Lister, G., Mocavero, S., Underwood, S., and Wright, G.: **CPMIP: measurements of real computational performance of Earth system models in CMIP6**, *Geosci. Model Dev.*, 10, 19–34, https://doi.org/10.5194/gmd-10-19-2017, 2017.

[5] A variable controlling the loop blocking size used by ICON and IFS

6. Changes made and/or difficulties encountered, if any

The benchmarking tool Kronos, which was part of HPCW v1.0 inherited from the ESCAPE-2 project, is used to simulate and analyse the workload generated by HPCW benchmarks. It had been planned to enhance Kronos to be able to deal with production-mode simulations on machines with pre-exascale dimension. However, it became clear that this would not add to the value of HPCW. Instead, ECMWF (in close collaboration with Atos) has extended the Kronos submission mechanism to cope with the new SLURM job scheduler, and also the multi-cluster configuration of the new HPC.

The original plan had been to run benchmarks from ESCAPE-2 onto the pre-exascale EuroHPC systems, diagnose performance and provide advice on how to boost performance on these specific hardware configurations. However, only one of these systems, LUMI, became available, and only towards the end of the project. Thanks to the extension of the project and the delay of this deliverable by three months, we were able to install HPCW on LUMI and obtain preliminary results.

7. How this deliverable contributes to the European strategies for HPC

This deliverable outlines the continuing development of HPCW, which aims to capture the spread of European high-performance codes for weather and climate simulation and bring them together in one place in a consistent manner for benchmarking purposes. Although HPCW is still a work-in-progress, once it is mature it will be an invaluable tool not just for comparing existing EuroHPC machines but also for the procurement of future systems, making sure that these systems will be compatible with the requirements of a class of applications of great scientific and societal relevance. This directly supports the establishment of scientific applications as a central strategic pillar of the European HPC ecosystem.

8. Sustainability

The continuous integration system of HPCW facilitates maintenance and adoption of the benchmarks and thus is the technical basis to enable sustainability of the tool in the long term. We hope and expect that HPCW will be adopted by the community and in particular receive support from EuroHPC sites.

In addition, we have secured some funding for the continuation of the effort within ESiWACE3, the third phase of the ESiWACE CoE, which is funded by the EuroHPC JU. The EuroHPC hosting sites BSC, CSC and JSC are partners of this project. ESiWACE3 is committed to address co-design challenges using HPCW as a common tool. It will maintain and expand HPCW itself and in particular the automatic deployment tools to facilitate and speed-up the adoption of new HPC architectures.

Furthermore we are in the process of setting up a separate cooperation project with Japanese colleagues to include NICAM, one of the world’s leading atmosphere models, in HPCW and to install and test the HPCW suite on the Japanese Fugaku system.

9. Dissemination, Engagement and Uptake of Results

9.1 Target audience

As indicated in the Description of the Action, the audience for this deliverable is:

X	The general public (PU)
	The project partners, including the Commission services (PP)
	A group specified by the consortium, including the Commission services (RE)
	This reports is confidential, only for members of the consortium, including the Commission services (CO)

9.2 Record of dissemination/engagement activities linked to this deliverable

N/A

9.3 Publications in preparation OR submitted

N/A

9.4 Intellectual property rights resulting from this deliverable

The HPCW framework is an open source project hosted on GitLab at DKRZ and on the ECMWF Bitbucket. Note that this does not imply that the weather and climate models themselves are also in the public domain. These codes are used and adapted but not developed within ESiWACE2.