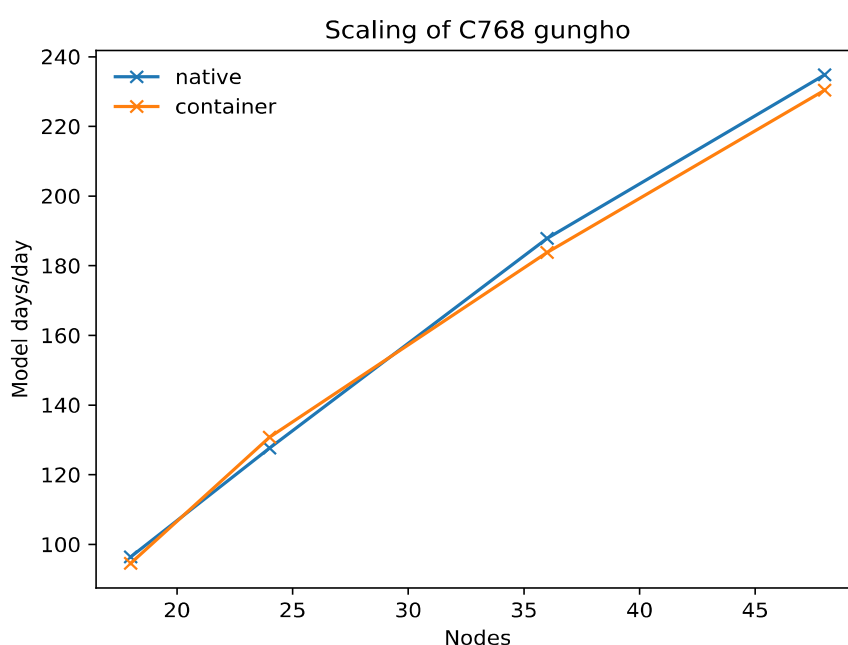




# Final Report on Porting Earth System Models to Containers

## Deliverable D2.9



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.

## ESiWACE2 Deliverable D2.9

### About this document

Work package in charge: WP2 - Establish, evaluate and watch new technologies for the community

Actual delivery date for this deliverable: 28 Feb. 2022

Dissemination level: PU (for public use)

### Lead authors:

Swiss National Supercomputing Centre, ETH Zurich (ETHZ/CSCS): William Sawyer, Theofilos Manitaras

Center for Climate System Modeling, ETH Zurich (ETHZ/C2SM): Jonas Jucker, Matthieu Leclair

### Project internal reviewer:

Max Planck Institute for Meteorology (MPIM): Reinhard Budich

### Contact details

Project Office: [esiwace@dkrz.de](mailto:esiwace@dkrz.de)

Visit us on: [www.esiwace.eu](http://www.esiwace.eu)



Access our documents in Zenodo:

<https://zenodo.org/communities/esiwace>

Disclaimer: This material reflects only the authors view and the Commission is not responsible for any use that may be made of the information it contains.

## Contents

1. Abstract /publishable summary.....	4
2. Conclusion & Results.....	4
3. Project objectives.....	5
4. Detailed report on the deliverable .....	5
4.1    EC-Earth.....	6
4.2    LFRic.....	6
4.3    COSMO .....	8
4.4    ICON.....	9
5. References.....	11
6. Changes made and/or difficulties encountered, if any .....	12
8. Sustainability.....	12
9. Dissemination, Engagement and Uptake of Results.....	12
9.1 Target audience .....	12
9.2 Record of dissemination/engagement activities linked to this deliverable .....	13
9.3 Publications in preparation OR submitted.....	14
9.4 Intellectual property rights resulting from this deliverable .....	14

## 1. Abstract /publishable summary

Atmosphere and ocean models are characterised among other things by complex dependencies, external configurations, and performance requirements. The containerisation of such a software stack helps to provide a consistent environment to ensure security, portability and performance. Since the container is built only once but then can be deployed on multiple platforms, productivity is increased.

ETH Zurich has led a multi-year effort to containerise key models from the climate / numerical weather prediction community. This effort started in 2018 with a “hackathon” (programming session) to help ESiWACE2 community scientists create containers for their Earth-system models. The six teams made fast progress on their containers and their results were reported in the ESiWACE2 deliverable “D2.8 Summary of Container Hackathon Experiences”. Of these teams, three received extended ESiWACE2 funding to refine their containers : LFRic, EC-Earth, and COSMO. Even though they did not receive funding, the ICON team viewed their containerisation as strategic and has offered their results as part of this final deliverable. Therefore, we report here on four model containers – LFRic, EC-Earth, COSMO and ICON – rather than the originally foreseen three.

The timing of the hackathon – just before the Covid-19 pandemic – was fortunate, for the teams were able to return home to refine their containers and put them into a production framework with only virtual interaction. ETH Zurich provided technical assistance to the teams, which became highly independent with time. The requirement for the final deliverable was not only the containerisation itself but also that the resulting model performance did not suffer adversely. The overall process and an overview of the containerization work is given in the subsequent sections, each of which references the public online documentation which the teams developed during their efforts.

## 2. Conclusion & Results

Work on this deliverable took place intermittently during 2020-2022, with each of the three teams receiving roughly 3 PM for the implementation. ETH Zurich provided container assistance on request through Theofilos Manitaras, as well as the possibility to continue their efforts to deploy and benchmark their containers on the ETHZ/CSCS Piz Daint HPC platform.

The efforts of at least two of the three teams were a success. Containers were successfully integrated into the production workflow for the ICON, LFRic, and COSMO models, and at least one publication is pending from these efforts. The risk assessed at the onset of ESiWACE2 was that significant performance might be lost when running large, complex models using message passing and even Graphics Processing Units (GPUs). Thanks to HPC-aware middleware, this problem did not materialise: results in the detailed report below indicate that the overhead of running in containers is negligible (<10%) and is considerably smaller than the variations in compute time due to changing load on the underlying HPC platform.

We view the containerisation results as a clear indication that containers can increase productivity in the deployment of Earth system models running on HPC systems.

### 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	Yes
Establish new technologies for weather and climate modelling	Yes
Enhance HPC capacity of the weather and climate community	Yes
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

ETH Zurich (ETHZ) coordinated the containerisations by intermittent coordination and periodic participation in the individual projects:

- Theofilos Manitaras
- Jonas Jucker
- Alberto Madonna
- Christopher Bignamini
- Andreas Jocksch
- William Sawyer

After the ESiWACE2 “Container Hackathon for Modellers”<sup>1</sup>, reported in deliverable D2.8<sup>2</sup>, efforts were concentrated on 3 models: LFRic (led by University of Reading), EC-Earth (led by Barcelona

<sup>1</sup> Container Hackathon for Modellers, 3-5 December 2019, Lugano (CH): <https://www.esiwace.eu/events/container-hackathon-for-modellers>

<sup>2</sup> Sawyer, William, & Benedicic, Lucas. (2020). Summary of Container Hackathon Experiences (D2.8). Zenodo. <https://doi.org/10.5281/zenodo.4323261>

Supercomputing Center, BSC), and COSMO (led by MeteoSwiss and ETHZ). In mid-2022 the impending containerisation of ICON model by the ETH Zurich -- needed for the community's production workflow -- led us to suggest this work also for the final deliverable.

The following people made non-trivial contributions to the containerisation efforts:

Model	Contributors
EC-Earth	Pablo Echevaria Julian Berlin (BSC) Uwe Fladrich (SMHI)
ICON	Jonas Jucker (ETHZ/C2SM) Theofilos Manitaras (ETHZ/CSCS) Andreas Fink (ETHZ/CSCS) William Sawyer (ETHZ/CSCS)
LFRic	Iva Kavcic (UKMetOffice) Simon Wilson (UKMetOffice)
COSMO	Jonas Jucker (ETHZ/C2SM) Matthieu Leclair (ETHZ/C2SM) Theofilos Manitaras (ETHZ/CSCS)

#### 4.1 EC-Earth

The BSC team made very quick progress on during the Container Hackathon (Dec. 2019, extensively reported in D2.8), containerising versions of two simplified EC-Earth configurations: one for the current stable version EC-Earth3, and another for the next generation EC-Earth4, which is currently under development. BSC considers the results of the hackathon as sufficiently successful to also satisfy the ESiWACE2 Milestone MS5<sup>3</sup> reporting requirement. Public online documentation for this effort is not available, but the final report<sup>4</sup> is available.

#### 4.2 LFRic

LFRic<sup>5</sup> is the new weather and climate modelling system being developed by the UK Met Office to replace the existing Unified Model (UM) in preparation for exascale computing in the 2020s. LFRic uses the GungHo<sup>6</sup> dynamical core and runs on a semi-structured cubed-sphere mesh. One of the guiding design principles, imposed to promote performance portability, is the "separation of concerns" between the science code and parallel code. An application called PSyclone<sup>7</sup> developed at the STFC Hartree Centre, can generate the parallel code enabling deployment of a single source science code onto different machine architectures.

<sup>3</sup> William Sawyer, Jonas Jucker, & Simon Wilson. (2022). Containers 2: Status of containerisation of Earth Science Models - Milestone MS5. Zenodo. <https://doi.org/10.5281/zenodo.7462595>

<sup>4</sup> <https://github.com/eth-cscs/ContainerHackathon/blob/master/EC-Earth/Readme.pdf>

<sup>5</sup> <https://doi.org/10.1016/j.jpdc.2019.02.007>

<sup>6</sup> <https://www.metoffice.gov.uk/research/foundation/dynamics/next-generation>

<sup>7</sup> <https://github.com/stfc/PSyclone>

At the Container Hackathon (Dec. 2019), both a Docker and a Singularity container were developed. The singularity container<sup>8</sup> is being actively used for several projects. These include:

- Development of LFRic code on individuals' laptops.
- On ARCHER2<sup>9</sup>, the UK's academic supercomputer, using the Cray interconnect:
  - A C1152 (8.5 km) resolution on 81 nodes and 324 nodes and a C2304 (4.3 km) resolution on 324 nodes. This is the highest resolution we have ever run LFRic (Gungho, dynamics only). Limited from running further by Archer2 config.
  - It was used as part of the XIOS benchmarking study for the Excalidata project<sup>10</sup>.
  - It will form the basis of any future LFRic installations on ARCHER2.
- On Microsoft's Azure: It has been run (so far) on 5 nodes in Azure with much larger runs planned.
- On Jasmin<sup>11</sup>: As part of a proposed student's benchmarking project using OpenMP and the fast ethernet interconnect.
- On Dirac CSD3 system at Cambridge<sup>12</sup>: Preparation for the I/O using both network fabric and advanced burst buffers as part of the Excalidata project, using the Infiniband interconnect.

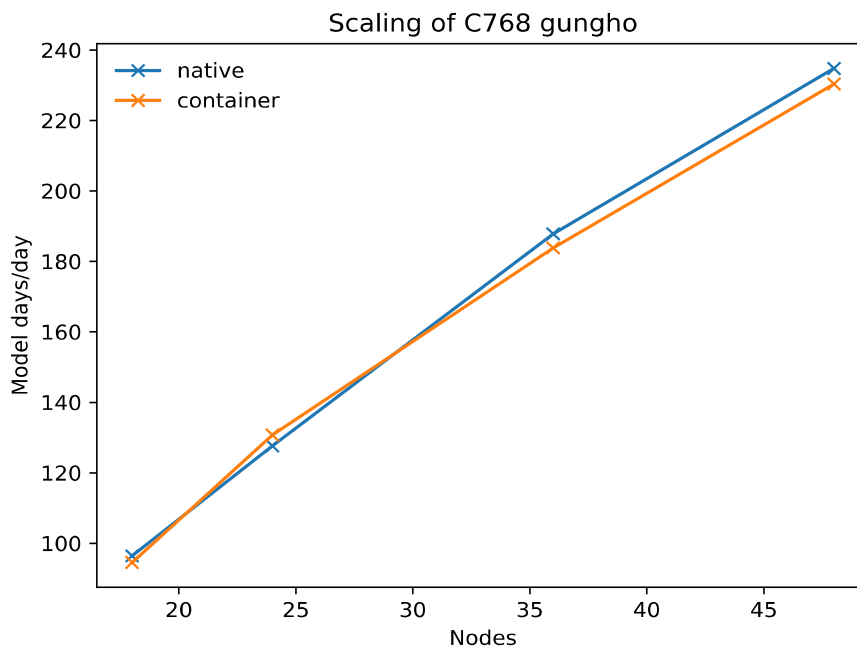


Figure 1: Strong scaling of the Gungho dynamical core for both the native (“bare-metal”) and Singularity container versions.

A performance comparison between natively compiled and containerised LFRic indicates that running in container introduces virtually no overhead, see Figure 1. Containers have thus proved to be an extremely useful tool when the long-term operation of software is required, beyond the longevity of the software stack, which is frequently updated on HPC platforms.

<sup>8</sup> [https://github.com/NCAS-CMS/LFRic\\_container](https://github.com/NCAS-CMS/LFRic_container)

<sup>9</sup> <https://www.archer2.ac.uk>

<sup>10</sup> <https://excalibur.ac.uk/projects/excalidata>

<sup>11</sup> <https://jasmin.ac.uk>

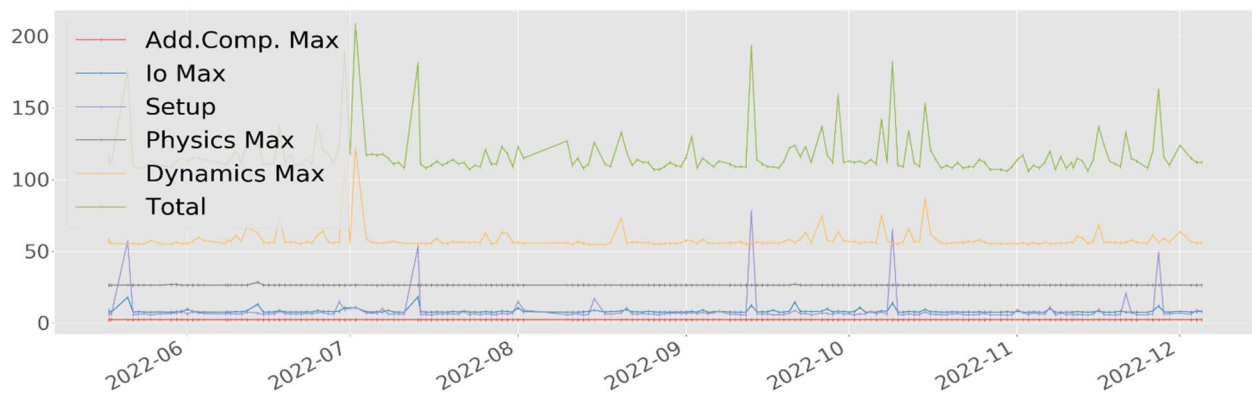
<sup>12</sup> <https://www.hpc.cam.ac.uk/dirac-csd3>

In its investigations, the LFRic team determined there was no meaningful loss in performance when running within a container, assuming the container run-time has the proper HPC support.

### 4.3 COSMO

COSMO has been in production at the German Weather Service (DWD) until 2015. At MeteoSuisse it is used for weather forecasting and will be supplanted by ICON in 2023. Although it was not represented at the Container Hackathon (Dec. 2019), it is an ideal candidate for containers since users require outdated configurations to run after COSMO is no longer officially supported. In this common case, containers can offer long term stability for a given software stack.

Native build cosmo-2e, 1 member, 4h forecast



Container cosmo-2e, 1 member, 4h forecast



Figure 2: The simulation run times of COSMO are depicted in the ETHZ/C2SM Continuous Integration pipeline, which periodically runs both the bare-metal (top) and containerised (bottom) versions of COSMO in a testing mode. The significant run time variations of either version come from the dynamic load of the machine and, particularly, the file system. This dynamic variability is far greater than the difference (roughly 5%) in execution times of the bare-metal and container versions, which run contemporaneously to take the machine load into account.



The Center for Climate System Modeling (C2SM) at ETH Zurich took up the task of containerising COSMO in 2021. The container workflow is documented in <https://github.com/C2SM/container> (public access), which also contains the requisite Dockerfiles.

The resulting COSMO-container is embedded in the Jenkins Continuous Integration (CI) system used by C2SM, where it runs in addition to the COSMO code built natively (without containers). Although runtimes vary due to the load characteristics of the Piz Daint platform at CSCS, the overhead for the containerization is estimated to have less than 10% degradation in performance, as is illustrated in the timings for both the native and container runs of within periodic Jenkins testing, see Figure 2. It should be noted that there is a much larger variation in the timings due to system fluctuations over time (which are often related to file system load) than there are between the native and container timings running on the machine at the same time.

Although it is now embedded into the C2SM workflow, the COSMO container is not currently exploited by users, for the simple reason that they prefer to compile and run with the existing software stack. Since COSMO is no longer maintained by the German Weather Service (DWD), it is highly likely that users will need the containers soon to run on outdated software stacks when COSMO no longer compiles with upgraded stacks.

#### 4.4 ICON

The Icosahedral Non-hydrostatic model is essentially the successor of COSMO for numerical weather forecasting but can also be used for climate simulations. A prototype container for ICON was already the subject of Milestone MS4<sup>13</sup> (Dec. 2020). While that prototype was largely successful, it was still a one-off: the ICON software and build system quickly moved on, and the Dockerfiles were no longer maintained. The maintainability question for containers is a central consideration and needs to be addressed to make containers useful in the long term.

The success of inserting the COSMO into the production workflow (mentioned above) led C2SM to propose a similar approach for ICON. This workflow requires that ICON is built using the Spack<sup>14</sup> package manager to manage ICON's many dependencies. Spack vastly simplifies the current ICON build system (which also includes the construction of the software stack), and thus would increase maintainability. Although this Spack-based build system is not yet in the official ICON release, the discussion about its inclusion is ongoing. The resulting Dockerfile becomes essentially a call to spack install.

The Dockerfiles are currently maintained in <https://github.com/C2SM/container>, however the ICON team is in the process of incorporating these into the ICON repository, which will then become available in a public release.

MeteoSuisse is in the process of completing its transition from COSMO to ICON as its production software, and the ICON containers will be fully incorporated into the workflow, allowing any ICON run to be executed through a container, even on Graphics Processing Units (GPUs), on which the

---

<sup>13</sup> Samiento, Rafael, Sawyer, William, Kosukhin, Sergey, Dietlicher, Remo, & Walser, Andre. (2020). The Containerisation of the ICON model for quasi-biennial oscillation simulations. Zenodo. <https://doi.org/10.5281/zenodo.4322638>

<sup>14</sup> <https://spack.io/>

production model typically runs. The containers are tested through the CSCS Alps Continuous Integration (CI) testing system, which validates the container workflow.

MeteoSuisse has provided the benchmark **mch\_bench\_r19b07\_dev**, which illustrates the current operational system under development. This benchmark was run within public release of the icon-exclaim repository (the latest benchmarks require the ICON version in the release candidate), and therefore it is possible to compare performance between the native-built and the container versions. The benchmark is designed to run on a 4xA100 GPUs on the MeteoSuisse production platform, but has been run for the sake of this comparison on the CSCS Daint system with single-P100 GPU nodes. As one can see in the subsequent strong scaling graph, the “bare-metal” (blue) timings are obscured by the red container timings, meaning that there is no major overhead in running the containerised code. The strong scaling is in Figure 3 below.

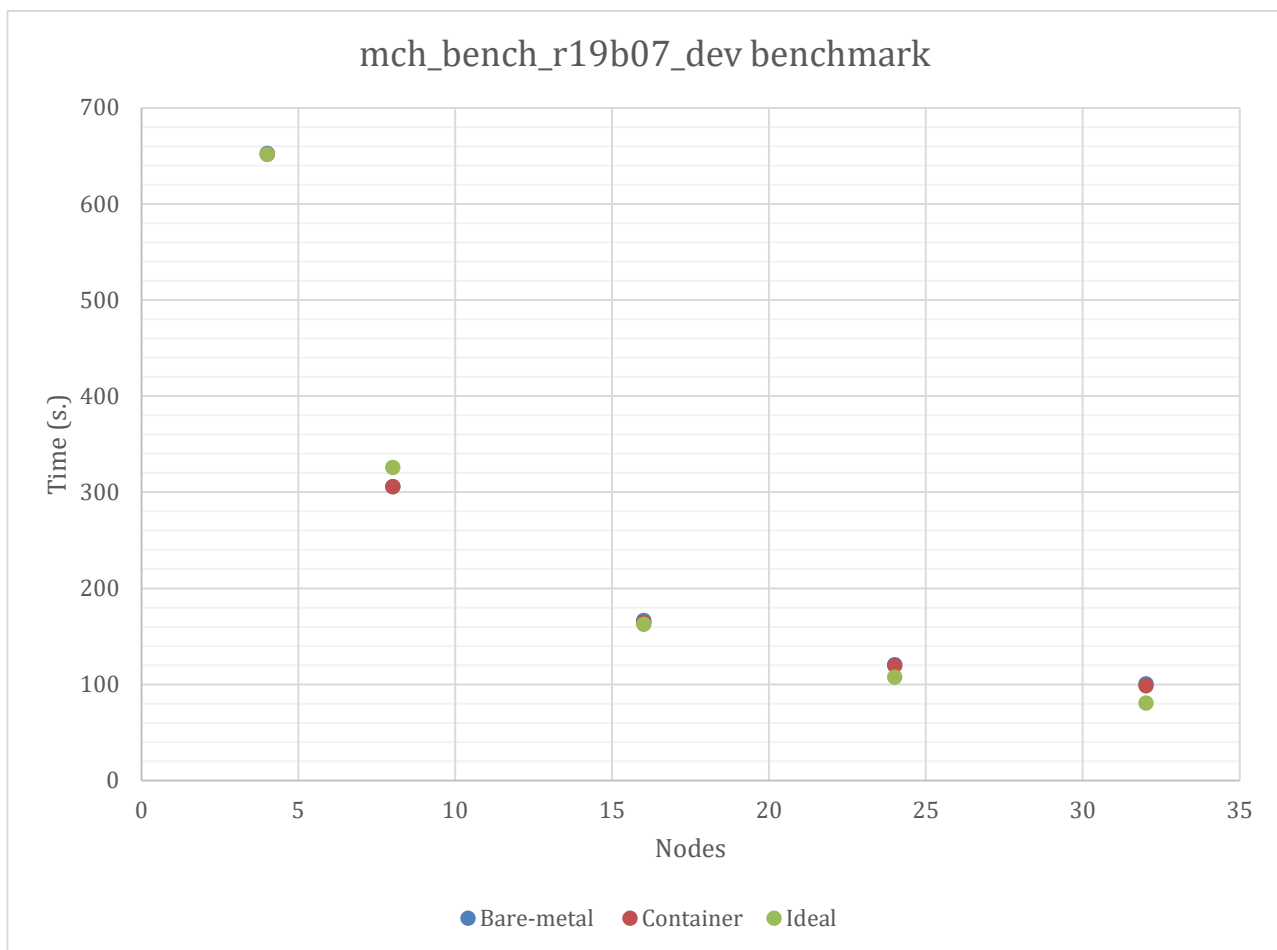


Figure 3: The strong scalability of both the “bare-metal” (blue) and containerised (red) versions of the MeteoSwiss regional 2km benchmark are given, along with the “ideal” linear scalability (green) normalised for the 4-node run. The strong scaling is reasonable from 4 to 32 nodes, and the execution times of the two versions coincide, such that the red dots largely obscure the blue.

In addition, the Max Planck Institute for Meteorology (MPI-M) has contributed the **atm\_qubicc\_r2b7** benchmark, a medium (20 km) resolution test case for a “thick atmosphere” (191

vertical levels) designed for the investigation of the Quasi-Biennial Oscillation<sup>15</sup>. Due to its memory requirements, this benchmark must be run on more than 50 P100 nodes of the ETHZ/CSCS Piz Daint platform. Figure 4 depicts the strong scaling of both the “bare-metal” and containerised versions, along with the “ideal” run times normalised to the time for 64 nodes. As for the MeteoSwiss benchmark, the overhead for running in a container is minimal, and the absolute scaling in the realm 64-256 nodes is reasonable for both versions.

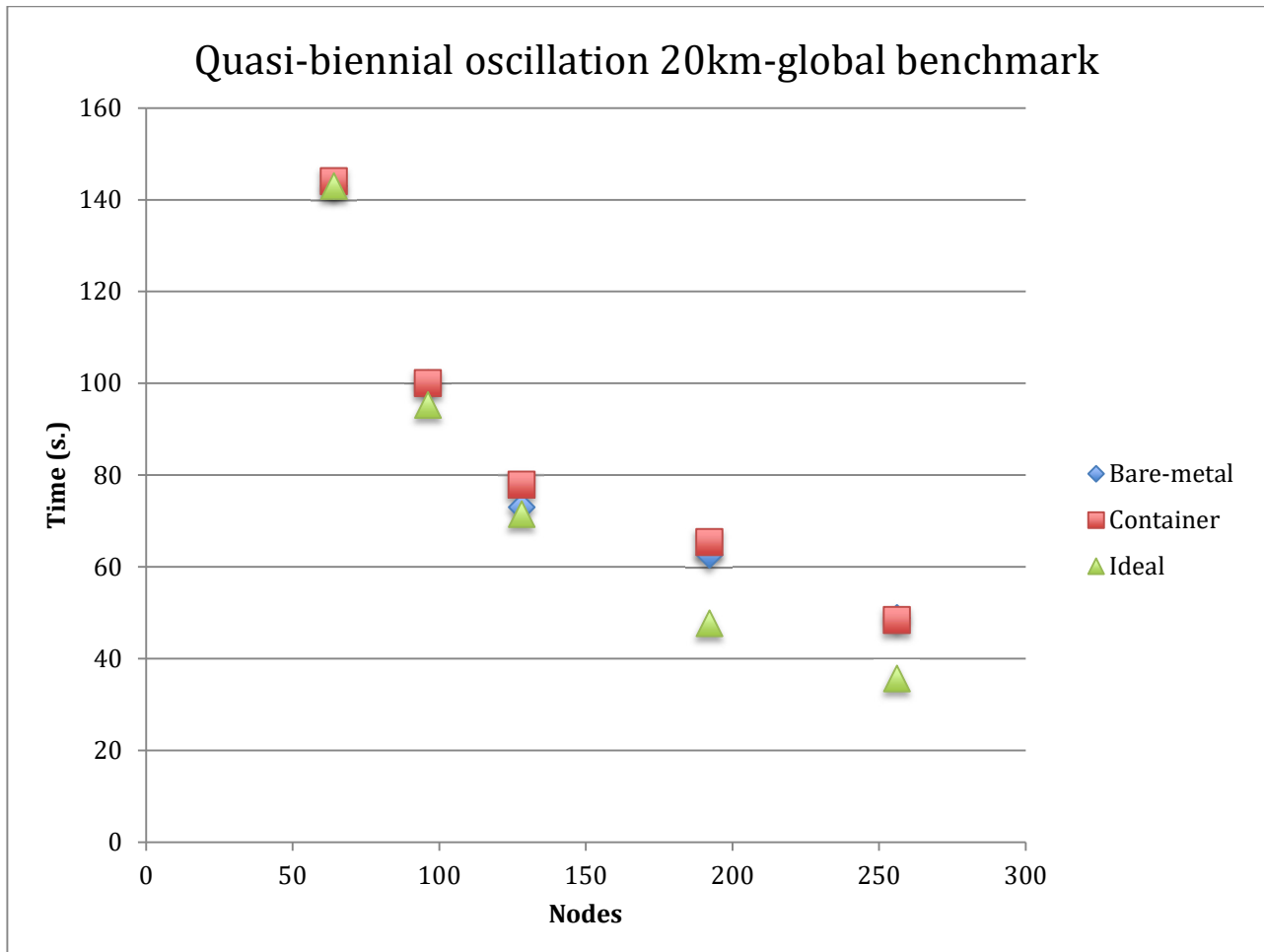


Figure 4: The strong scalability of both the “bare-metal” (blue) and containerised (red) versions of the Max Planck Institute for Meteorology (MPI-M) Quasi-Biennial Oscillation (QBO) 20km benchmark are given, along with the “ideal” linear scalability (green) normalised for the 64-node run. The strong scaling is reasonable for both. The performance of blue and red is so close that the red square often obscures the blue diamond.

## 5. References

All references in this document have been added as footnotes.

<sup>15</sup> <https://www.metoffice.gov.uk/weather/learn-about/weather/atmosphere/quasi-biennial-oscillation>

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

The ICON container was added to the deliverable, since it requires three production-quality applications with real-world benchmarks, and the work on EC-Earth does not meet this standard. We were fortunate that the ICON container was foreseen within the EXCLAIM<sup>16</sup> project in any event.

All the teams seemed to agree that containerisation was conceptually easy, but there were many subtleties. Some of these were related to HPC system issues, such as security requirements and the management of SSH keys within the container. And a major challenge is debugging the container version, which requires expertise in entering the container. In the case of ICON, these issues could only be resolved with the help of container experts at ETHZ/CSCS.

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

The advantages of containers within HPC where the software stack is frequently upgraded are well known. Frequent the upgraded stack causes the models to fail or to give incorrect answers, while the stack which produced correct answers may no longer be supported on the operational HPC platform. Containers can incorporate the software stack which is known to work and give longevity to the working model version.

This deliverable illustrates clearly that weather and climate applications can be built and run inside containers with little loss in performance and adding only minimal development overhead. We feel this is a crucial step forward for HPC strategy, not only in Europe but worldwide.

## 8. Sustainability

As mentioned in Section 4, the container knowledge has been passed to the application teams at UK Met Office, ETHZ/C2SM, MeteoSuisse, BSC, SMHI, MPI-M and DWD, which have now partially or completely integrated containers into their HPC workflows. These teams are largely independent of the container specialists at ETHZ/CSCS, although the latter is still available to assist through user tickets. We at ETHZ/CSCS believe that the step to utilising containers for weather and climate on HPC platforms has now been enabled for the long term.

We have learned that while some complexity is added by the containerisation, it is largely a porting exercise based on the existing build system. It is computationally expensive to build containers, but the effort can be automated to avoid the need for user oversight.

## 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)
X	The project partners, including the Commission services (PP)

<sup>16</sup> <https://c2sm.ethz.ch/research/exclaim.html>

	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)

**This is how we are going to ensure the uptake of the deliverables by the targeted audience:**

The containerisation of LFRic, COSMO and ICON is documented in publicly accessible GitHub repositories<sup>17 18</sup>. The value of containers for climate and weather applications on HPC platforms has been underlined at conferences such as PASC21<sup>19</sup>. Furthermore, in the realm of ESiWACE2 Work Package 6, containers were a topic at the summer schools in ESiWACE2 summer schools on “Effective HPC for Weather and Climate” in 2020<sup>20</sup> and 2021<sup>21</sup>.

In the future ETHZ/CSCS will encourage its users to adopt containers as the mechanism of choice to run their applications on the new Alps<sup>22</sup> HPC platform, offering educational opportunities to smooth the transition to containers, as well as extensive user documentation. In fact, containers are now a requirement to utilise the CSCS Continuous Integration system<sup>23</sup> for the Alps architecture.

## 9.2 Record of dissemination/engagement activities linked to this deliverable

Type of dissemination and communication activities	Details	Date and location of the event	Type of audience	Zenodo Link	Estimated number of persons reached
Container hackathon	Hackathon (intensive programming session)	Dec. 3-5, 2019 at CSCS in Lugano, Switzerland	Public	<a href="https://zenodo.org/record/3685888">https://zenodo.org/record/3685888</a>	16
Summer school	Summer School on Effective HPC for Weather and Climate	Aug. 23-29, 2020, virtual due to Covid-19	Public	<a href="https://zenodo.org/record/5795447">https://zenodo.org/record/5795447</a>	90+
Summer school	Summer School on Effective HPC for Weather and Climate	Aug. 23-27, 2021, virtual due to Covid-19	Public	<a href="https://zenodo.org/record/5795447">https://zenodo.org/record/5795447</a>	65+

<sup>17</sup> [https://github.com/NCAS-CMS/LFRic\\_container](https://github.com/NCAS-CMS/LFRic_container)

<sup>18</sup> <https://github.com/C2SM/container>

<sup>19</sup> Minisymposium “Sarus: Highly Scalable Docker Containers for HPC Systems”, PASC21, (July 201), [https://pasc21.pasc-conference.org/program/schedule/index.html%3Fpost\\_type=page&p=10&id=msa125&sess=sess130.html](https://pasc21.pasc-conference.org/program/schedule/index.html%3Fpost_type=page&p=10&id=msa125&sess=sess130.html)

<sup>20</sup> <https://www.esiwace.eu/the-project/stories/summer-school-on-effective-hpc-for-climate-and-weather>

<sup>21</sup> <https://www.esiwace.eu/events/summer-school-2021>

<sup>22</sup> <https://www.cscs.ch/computers/alps/>

<sup>23</sup> [https://gitlab.com/cscs-ci/ci-testing/containerised\\_ci\\_doc](https://gitlab.com/cscs-ci/ci-testing/containerised_ci_doc)

### 9.3 Publications in preparation OR submitted

In preparation OR submitted?	Title	All authors	Title of the periodical or the series	Is/Will <u>open access</u> be provided to this publication?
In preparation	Addressing portability and performance of a next generation weather and climate code using a Singularity container	Christopher Maynard, Simon Wilson, Bryan N. Lawrence	Concurrency and Computation: Practice and Experience	Open Access

### 9.4 Intellectual property rights resulting from this deliverable

All the docker files created for this deliverable are in the public domain on the websites specifically mentioned in Section 4. However, this does not imply that the weather and climate models themselves are also in the public domain since these clearly predate ESiWACE2. As such, no new IP rights have resulted from this deliverable.