

IMPACT : Icon on Massively Parallel ArchiteCTures

Version 0

Project duration: 10.2018 - 9.2022

Total FTE request: 4.26

Project leader: Carlos Osuna (MeteoSwiss)

Summary

Being able to run efficiently on modern high performance computing (HPC) architectures is a critical element to enable forecast quality improvements through increasing resolution, model complexity or the number of ensemble members. In recent years many different hardware architectures have emerged for HPC usage such as graphical processing units (GPUs), many-core processors or field-programmable gate array (FPGA) and performance improvement has been achieved by a dramatic increase in the number of compute units available on all these technologies. With the end of Moore's Law and Dennard Scaling, this trend is foreseen to continue and estimates are that there are more disruptive changes on the horizon. Building on the experience and knowhow from the COSMO POMPA project, the aim of this project is to adapt the ICON model to run on various architectures such as x86 multicore CPUs and GPU accelerators, focusing in the LAM mode for NWP applications. Through the various tasks proposed, the project will investigate and apply methods and tools to achieve performance portability, enabling efficient execution of a single source code on different hardware architectures. Adopting such tools is also a step to get ready for future hardware which will likely require to expose massive parallelism at an even larger scale than today. Software engineering aspects will also be considered to investigate readability, maintainability and safety of the ICON base code.

Motivation

Thanks to the advances in high-performance computing (HPC) systems, recent years have seen significant progress in our capacity to predict weather and climate evolution using numerical models. The latest improvement in computing capacity was achieved by increasing massively the number of compute units on a chip, which requires the software to expose and exploit always more parallelism. This trend will further continue in the future, likely at an even larger scale. An important game changer in the HPC industry is the emergence of artificial intelligence which is now one of the main drivers in hardware design and may further lead to an increase in architecture diversity and will require to adapt our models and change the way we program them (Dongarra et al. 2011; Hu et al. 2010). These developments on hardware technologies offers exciting prospects to improve forecast quality, for instance increasing model resolution would allow to use a model description closer to first principles. In particular, at horizontal resolutions of $O(1 \text{ km})$ and below, the models become cloud permitting and some

physical parameterizations such as convection can be switched off. One could also benefit of more computing capacity by increasing the number of ensemble members for probabilistic forecasting.

From a computer science perspective, the changes in hardware architecture poses major challenges. Harvesting the computational capacity of emerging HPC systems increasingly involves the use of heterogeneous many-core architectures consisting of both CPUs and accelerators (e.g., GPUs). The efficient exploitation of such architectures and future systems requires changes in the way we develop our software. We have to consider new programming models in order to adapt state-of-the art weather and climate models which are typically significant code bases maintained by a large community of domain-scientists.

As part of the POMPA (Performance On Massively Parallel Architectures) project the COSMO model was adapted to run on GPU architecture using a combination of Domain Specific Language (DSL) and compiler directives (OpenACC) (Lapillonne and Fuhrer 2014, Gysi et al. 2013, Fuhrer et al. 2014). Thanks to this work the COSMO code is now run on a GPU system for the operational numerical weather prediction of MeteoSwiss and for regional climate modeling at ETH Zurich.

In the proposed project, we intend to build on the know how acquired while porting the COSMO model to GPU and apply similar approach to the ICON (Icosahedral Nonhydrostatic) model (Zängl et al. 2015). The project will also take advantage of a partial port of the dynamical core of ICON with OpenACC compiler directives which is already integrated in the official code version. The ICON model is already highly optimized for CPU architectures and the project will first focus on running the model on GPU systems which are at the moment a working alternative for such model. Other architectures will be considered at a later stage. This project will be closely coordinated with the ENIAC project which aims at adapting ICON for new architectures for climate applications.

Actions proposed

- Investigate together with ICON developers if improvements regarding software engineering aspects can be applied and would benefit to the existing ICON model: make the Fortran code more modular, further improve the existing testing infrastructure by adding threshold based validation.
- Port ICON for NWP applications to GPU using OpenACC directives only: this will serve as the performance baseline for any further steps.
- Investigate-Optimize ICON for x86, GPU, (and possibly additional architectures TBD in close coordination with the ICON developers, e.g. ARM and new Intel accelerator).
- Performance evaluation on HPC systems of COSMO members: run first COSMO with the C++ dycore on CPU (and GPU if available). This will be followed by a comparison with ICON-LAM (limited-area mode of the ICON Modelling Framework).
- Use domain specific language (DSL) and abstractions to achieve better portability of ICON-LAM and prepare the model, in close coordination with the ICON developers, for future changes in hardware. The approaches proposed offer a broad coverage of different levels of abstraction and disruptive changes for later evaluation and possible integration into the Fortran model.
- Explore task parallelism to improve strong scalability.

Deliverables of the project

- Extensions of ICON that allow to run the model (with focus on ICON-LAM for NWP application) on CPU and accelerators (GPUs) achieving a performance improvement for the official ICON version. Required features of the operational configuration of the different COSMO centers will be included in the implementation, e.g. 2 way nesting capability. [code+documentation]
- For the physics, latent heat nudging and organizational code, OpenACC will be used to port the code to GPUs. For the dynamical core, OpenACC as well as the DSL-approaches based on GridTools will be delivered. Comparison of both approaches for the dycore as a basis for insertion in the official ICON code will be reported. [code+documentation and report]
- Comparison of OpenACC with OpenMP 4.5, as well as recommendation for future use [code+documentation and report]
- Evaluation of the use of Fortran base CLAW-DSL single column abstraction for the physics to achieve performance portability with a single source code. Demonstrate it on two parameterizations: microphysics (graupel scheme) and turbulence [code+documentation and report].
- Study on task parallelism based on a literature review and on the evaluation done with the COSMO dynamical core, including recommendations for a general introduction in ICON [report].

Description of individual tasks

The detailed FTEs per task are in the annex 1.

If not otherwise explicitly mentioned, all tasks regarding the ICON model will be done focusing on ICON-LAM for NWP applications. Care will be taken that the implementations are compatible and whenever possible extendable to the other modes of ICON, but no completeness will be guaranteed by the project for these modes.

Task L: Project leadership

Task 1 : Testing and Software engineering

The ICON model is a powerful tool for global and regional weather modelling. Adapting such a large code base, with more than one million lines of code, for different HPC architectures is a great challenge. The code has been highly optimized for CPU architectures and some automated testing is already in place in collaboration with the Max Planck Institute in Hamburg. Based on the experience from COSMO, the maintenance effort of a codebase that has to run in different configurations on different hardware architectures can be substantial. The cost of adapting and maintaining the Fortran code base can potentially be reduced by improving software engineering aspects. In particular, modularity in a sense of being able to run separately different components of the models is an important functionality that will allow to work, develop and test individual components of ICON independently from the rest of the model. Modularity and component testing is a crucial software engineering practice that

decreases the maintenance cost of a model and reduces the complexity of coupled software systems.

In addition the existing testing infrastructure can be extended to further increase the code coverage of the tests. Furthermore most of the current technical testing in place for ICON relies on bitwise-identity. However, bit reproducibility is not guaranteed in general across different compilers and architectures. Therefore, additionally to the current bit reproducibility testing, a threshold based validation – similar to the one used in the COSMO technical testsuite – will be proposed and implemented in this task.

Task 1.1 Guidelines/recommendation for future ICON development

Monitor programming practices in other communities using large Fortran codes. Organize workshops / presentations on software engineering and propose guidelines to the model developers.

Task 1.2 Modularize components of the Fortran code

Provide a solution for applying modularity to the ICON model, where, if adopted, any component of the model could be run in a standalone manner. The approach(es) will be discussed with the ICON developers. A prototype will be implemented for two components or parameterizations and used to demonstrate the component-wise testing that would be applicable to ICON.

Task 1.3 Improve testing infrastructure

Implement threshold based validation for ICON.

Deliverables:

D1.1 (3.2019) Threshold based validation in place in ICON [code+documentation]

D1.2 (3.2020) Modular and standalone version of the microphysics (graupel scheme) and of the tendencies of velocity components of the ICON model [code+documentation, presentation]

Task 2: Baseline performance on CPU and GPU

In this task we propose to implement a baseline version of the ICON model using OpenACC to enable both CPU and GPU execution. The port will be complete and will cover all components of ICON required for LAM NWP simulations. This work will profit and complement the already existing OpenACC implementation of the ICON dynamical core. The performance results will serve as a baseline for other porting approaches (see Task 3) and evaluated in terms of performance, efficiency and time-to-solution. The OpenACC directives are the current working standard for GPU, however another set of directives, OpenMP 4.5 for accelerators, is emerging and may be considered in the future. Currently, OpenMP for accelerators is not yet fully mature and it is not clear at this stage whether this approach will be applicable for codes such as ICON. Some test implementations will be carried out in this project in order to give a recommendation.

Task 2.1 OpenACC port of the ICON model

This includes physics, and other components such as boundary conditions and output. Part of the physics is already implemented with OpenACC for COSMO and can be reused with minor modifications, namely, the microphysics, turbulence, soil and lake modules. Other packages, such as the convection and SSO modules, will be ported. Some components of the dynamics

which are not yet supported with the OpenACC version, such as 2-way nesting, will be ported. The data workflow when running in assimilation mode on a GPU system will be analyzed. We note that, except for latent-heat nudging, data assimilation is not part of the ICON code but is integrated in the data assimilation software DACE. The forward operators needed for the assimilation system KENDA will be called via an online coupling to DACE, which will require data transfer between the GPU and the CPU. In case such transfer would be too costly, some component of DACE may have to be ported as well with OpenACC. In such case work would have to be negotiated and coordinated with responsible group at DWD led by Roland Potthast.

See annex 2 for detailed FTE planning.

Task 2.2 Performance results

In a first step, the performance of only the dynamical core of ICON will be investigated. Performance comparison with results achieved by other porting efforts, namely COSMO and COSMO-Eulag with DSL based implementation on different architectures will be carried out. The tests will be run using the CDIC project test cases. In a second stage, a performance comparison for the full ICON model on CPU and GPU for realistic NWP cases will be carried out.

Task 2.3 OpenMP evaluation

The viability of using OpenMP for accelerator directives will be investigated for the microphysics parameterization, as a prototype. Conclusions extracted from the evaluation of this prototype will be applicable to other parameterizations. Recommendation to replace OpenACC directives with OpenMP directives will be made. The change of the full model from OpenACC to OpenMP is evaluated to about 0.3 FTE (not part of this project).

Deliverables:

D2.1 (08.2021) GPU capable version of the ICON model with OpenACC directives for existing numerical weather prediction applications at DWD and COSMO partners [code+documentation]

D2.2 (08.2020) Port of the microphysics (graupel) as well as velocity tendency code with OpenMP 4.5 and recommendation document regarding this set of compiler directives [code+documentation & report]

D2.3 (9.2019) Performance comparison of the dynamical core between COSMO, and ICON on different architectures. [report]

D2.4 (12.2021) Performance results comparison of the full ICON model on CPU and GPU [report]

Task 3 : Performance portability and abstraction

The OpenACC port of ICON will provide a model version that can run on GPU and CPU. However, from the experience gathered with the port of the COSMO model, it is known that optimizations for CPU and GPU often result in significantly different implementations and structure of the computations. This inhibits retaining a single sources code and will require to accept implementations that will not be optimal in one or more architectures. In addition the OpenACC programming model does not expose fine grain hardware optimizations, such as control over the memory hierarchy, which could have an impact on performance. Finally, introducing OpenACC on top of Fortran+MPI+OpenMP increases the complexity of the code which may result in additional maintenance effort. For the dynamics – because of the horizontal

dependencies which introduce additional complexity for optimization and performance portability – it was shown, that an optimized OpenACC implementation could be up to 50% slower as compared to an optimized code using a hardware specific language. In order to achieve portability retaining a single source code and performance portability as well as higher code maintainability we propose different alternative approaches. The approaches proposed in this task offer different level of abstractions and will also require different levels of disruptive changes in order to adopt them in the Fortran version of ICON.

For the physics, where individual vertical columns are independent, the use of the CLAW-DSL (Domain Specific Language) will be prototyped and evaluated. The CLAW-DSL is a Fortran based DSL for the physics which allows to implement computations in the code as single column, where the horizontal loops and directives are generated automatically. The generated code is Fortran code with compiler directives.

For the dynamics we propose to further achieve separation of concern between the user code and the hardware implementation by using high level domain-specific language (DSL). This task will benefit from development of a new DSL language as part of the ESCAPE-2 (Energy-efficient Scalable Algorithms for Weather Prediction at Exascale) project led by ECMWF. The ICON developers will be involved in the design of the new language and this DSL will be applied to ICON. The backend of the DSL will likely be based on the GridTools library. The GridTools DSL is an extension of the STELLA library used for the COSMO dynamical core and which has proven its efficiency on CPU and GPU architecture. The new DSL-ICON dynamical core will be attached to the official ICON code as contributed code, and the additional maintenance will be the responsibility of MeteoSwiss. After the end of the project, the Fortran version will continue to be the main version, and the DSL-based dynamical core will not require substantial extensions or modifications of the main version. The new dycore will retain the existing flexibility for grid nesting and MPI domain decomposition.

Task 3.1 Apply CLAW DSL to the most relevant parameterizations

The CLAW-DSL can be applied incrementally as the resulting CLAW optimized code can directly run within an existing OpenACC code. This approach will be applied to the microphysics and radiation code. An evaluation of the results will be done and recommendations regarding a general use of CLAW will be reported.

Task 3.2 Participate in design and implement DSL based dynamical core

The ICON developers will participate in the design of the DSL and the DSL approach will be applied to the complete dynamical core.

Deliverables:

D3.1 (06.2022) Microphysics and radiation parameterization implemented with CLAW-DSL and evaluation report [code+documentation & report]

D3.2 (03.2019) DSL Design workshop with the ICON developers [workshop]

D3.3 (06.2022) Performance portable dynamical core implemented using DSL [code+documentation]

Task 4: Strong scalability

There is a trend of hardware architectures and programming models to move towards a data-movement centric and task oriented approach. This task aims at achieving better strong scaling by using task parallelism. Currently only the parallelism in the spatial dimension is exploited, however with the continuous increase of computing units available per chip in modern accelerators, our model configurations are not providing enough data parallelism to efficiently use all the computing units, limiting the scalability of the model. One way of improving the strong scalability is to introduce task parallelism to different components of a model that can be run concurrently. Exploratory work has shown that our models exhibit a considerable degree of parallelism that can be applied to multiple components. This however requires extensive changes to the code structure to be applicable. In this task we will explore and evaluate the impact of introducing task parallelism using an already available version of the COSMO dynamical core implemented with a new prototype DSL. This prototype DSL allows for exploring automatically the potential gain obtained from task parallelism. This will give insight into for a future implementation strategy in ICON.

Task 4.1 Monitor other projects applying task parallelism in weather and climate models

Watch other projects that are applying task parallelism to weather and climate models to improve scalability. The goal is to evaluate different solutions and technologies being applied and their impact on performance on modern accelerators.

Task 4.2 Small prototype out of COSMO

Use the C++ dynamical core of COSMO as a vehicle to explore task parallelism. A DSL-based dynamical core is ideal for this purpose, since the DSL allows for a high-level of abstraction of the user code and the information exposed in the DSL can be used in order to automatically generate data flow graphs and a schedule of how to run tasks.

Deliverables:

D4.1 (06.2020) Literature/Project review document regarding task parallelism [report]

D4.2 (06.2021) Prototype based on the COSMO dynamical core using task parallelism [code]

Task 5: Coordination activities with ICON development

A good coordination with the official developments of ICON will be crucial in order to integrate outcomes of the IMPACT project into the official code. In this task we propose activities in order to have close interaction with the main developers of ICON in order to facilitate possible future integration of IMPACT into the official code as well as to integrate feedback of the ICON developers in the technologies and porting efforts being developed.

Task 5.1 We will organize joint sessions with the ENIAC project at the ICON developers meetings. The progress and main outcomes of the project will be discussed during those sessions. Feedback gathered will be essential and considered for further developments.

Task 5.2 Introduce code review tools for the main developments of IMPACT that are offered to the main ICON developers. Key developments of IMPACT, like OpenACC port of physical parameterizations and organizational code will be open for code review in the github platform of ENIAC, or on the reviewing infrastructure of DWD/MPI once available, where changes to the official code are discussed.

Deliverables:

D5.1 (-) Regular meeting sessions at the ICON developers meetings, jointly with ENIAC [workshop]

Links to other projects or work packages

The C2I project is related to the migration of COSMO members to ICON and is therefore strongly linked with this project as some member may only transition to ICON once it is GPU capable.

A strong collaboration with the PP CEL-ACCEL which also make use of the GridTools library is required. It is planned to have shared workshop and parallel sessions at the COSMO meetings.

The PASC ENIAC project aims at adapting ICON for climate on heterogeneous architecture and will be strongly coordinate with this project.

The PASC PASCHA project will explore task parallelism in the context of COSMO and is therefore key for Task 4.

The ESCAPE-2 project will provide new DSL suitable for dynamical cores on icosahedral grids with a focus on high productivity and ease of use for scientific developers of the model.

The ESIWACE-2 project will demonstrate the use of recent DSLs for model components of the ICON model.

Risks

The main risk will be to integrate the changes required for this project with the main development line of the ICON model. Since the ICON code base is very large, and is continuously being developed by many different contributors, the work will need to be carefully coordinated. At the start of the project, the governance rules for criteria to accept code changes for the COSMO consortium into ICON are being defined and are not yet fully established. A close coordination between the main ICON developers and this project will be crucial. The risk will be mitigated within the project by means of establishing regular discussions with main ICON developers (Task 5) and reviews of the developments proposed.

Additionally maintenance of the code might be a challenge, since (based on COSMO experience) it is expected to increase when adapting the model for running on multiple architectures. This risk is mitigated in the project by the exploration and use of possible alternatives like the CLAW-DSL and GridTools DSL proposed in the project (Task 3) that offer different levels of abstractions and the possibility to retain single source codes and implementations. At the end of the project the different approaches will be evaluated taking into consideration the maintenance cost for the model and performance portability in different architectures.

Acceptance of new code and technologies may be challenging and regular exchange with the main developers will be organized to mitigate this risk (Task 5). Tools and technologies will be adapted and developed based on user feedback and requirements.

References

- Dongarra, J., P. Beckman, T. Moore, and co-authors. The international exascale software project roadmap. *Int. J. High Perform. Comput. Appl.*, 25(1):3–60, February 2011.
- Gysi, T., O. Fuhrer, C. Osuna, M. Bianco, T. C. Schulthess, 2013: STELLA: A domain-specific tool for structured grid methods in weather and climate models. Proceedings of the international conference for high performance computing, networking, storage and analysis, No. 41, doi: 10.1145/2807591.2807627
- Fuhrer, O., Osuna, C., Lapillonne, X., Gysi, T., Cumming, B., Bianco, M., Arteaga, A., & Schulthess, T. (2014). Towards a performance portable, architecture agnostic implementation strategy for weather and climate models. *Supercomputing Frontiers And Innovations*, 1(1), 45-62. doi: 10.14529/jsfi140103
- Hu, X.S., R.C. Murphy, S. Dosanjh, K. Olukotun, and S. Poole. Hardware/software co-design for high performance computing: Challenges and opportunities. In *Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2010 IEEE/ACM/IFIP International Conference on, pages 63–64, Oct 2010.
- Lapillonne, X., and O. Fuhrer, 2014: Using compiler directives to port large scientific applications to GPUs: An example from atmospheric science. *Parallel Processing Letters*. 24, 1450003, doi: 10.1142/S0129626414500030
- Zängl, G., D. Reinert, P. Ripodas and M. Baldauf, 2015: The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatic dynamical core. *Q. J. R. Meteorol. Soc.* (2015) DOI:10.1002/qj.2378.

Annex 1

Task	Contributing scientist(s)	FTE Total	FTE 2018-2019	FTE 2019-2020	FTE 2020-2021	FTE 2021-2022	Start	Deliverables	Date of delivery (*)
L	C. Osuna (MCH)	0.4	0.1	0.1	0.1	0.1			
1.1	X. Lapillonne , N.N. (MCH)	0.05	0.05				09.2018	document for ICON developer	08.2019
1.2	N.N (MCH)	0.2		0.2			09.2019	version of a the microphysics and	03.2020
1.3	N.N (MCH)	0.1	0.1				09.2018	Threshold based validation in place in ICON	03.2019
Total T1		0.1	0.05	0.2	0	0			
2.1	X. Lapillonne, N.N. (MCH) U. Schaettler, D. Reinert, G. Zaengl (DWD)	2.2 0.4	0.65 0.2	0.65 0.15	0.9 0.05		12.2018	GPU capable version of ICON for NWP	08.2021
2.2	X. Lapillonne (MCH), U. Schaettler (DWD), Riccardo (ReMet), Andrea Montani (ARPA), Cosmin Barbu (NMA), Zbigniew Piotrowski (IMGW) (0.02 FTE per contributing center)	0.1	0.05	0.05			06.2018	Performance comparison of the Dynamical core	06.2019
2.2	X. Lapillonne (MCH), U. Schaettler (DWD), Riccardo (ReMet), Andrea Montani (ARPA), Cosmin Barbu (NMA) (0.01 FTE per contributing center)	0.05				0.05	09.2021	Performance evaluation of the full ICON model	06.2022
2.3	N.N (MCH)	0.05		0.05			12.2019	Microphysics with OpenMP for accelerator and recommendation document	08.2020
Total T2		2.8							
3.1	N.N (MCH)	0.15		0.05	0.05	0.05	09.2019	Microphysics and radiation parameterization implemented with CLAW	06.2022
3.2	C. Osuna (MCH), G. Zaengl (DWD)	0.05		0.05				DSL Design workshop with the ICON developers	02.2019
3.2	N.N (MCH)	0.5			0.25	0.25	09.2020	Performance portable dycore implmented using DSL	06.2022
Total T3		0.7							
4.1	N.N (MCH)	0.01		0.01			09.2019	Literature/Project review document regarding task parallelism	06.2020
4.2	N.N (MCH)	0.09		0.05	0.04		06.2019	Small protoype based on the cosmo dynamic using task parallelism	06.2021
Total T4		0.1							
5.1	C. Osuna (MCH)	0.12	0.03	0.03	0.03	0.03			
5.2	C. Osuna (MCH)	0.04	0.02	0.02					
Total T5		0.16							
Total FTE		4.26							

Annex 2

Openacc Port	
Part of the Dynamics not yet supported with OpenACC such as support for 2-way nesting	0.3
Physics shared with COSMO already ported : Graupel microphy., ice microphy., Turbulence, soil, flake	0.4
Physics which need to be ported with OpenACC: Convection, sso, radiation	0.8
Data assimilation : Latent Heat Nudging, evaluation of the workflow with DACE assimilation code and potentiel port of component if required and possible	0.8
Organization code (boundary conditions, communication, output ...)	0.6
Total OpenACC port	2.6