IS-ENES2 DELIVERABLE (D -N°: 3.4)

Report on NEMO and ICON models re-design

{File name: IS-ENES2_D3.4.pdf}

Author(s): ***G. Aloisio, S. Mocavero, S. Fiore, P. Adamidis, M. Hanke, T. Jahns***

Reviewer (s): ***S. Valcke, M. A. Foujols***

Reporting period: ***01/04 2016 - 31/03 2017***

Date for review: ***31/10/2016***

Final date of issue: ***07/11/2016***

Revision table			
Version	**Date**	**Reviewer**	**Comments**
0.1	22/03/2016	Giovanni Aloisio	First release of the document on NEMO model re-design
0.2	27/07/2016	Panos Adamidis	First release of the document on ICON model re-design
0.3	01/08/2016	Giovanni AloisioSilvia Mocavero	Integration of the two documents
0.4	12/09/2016	Giovanni Aloisio	First formal release
0.5	23/09/2016	Sophie Valcke	Reviewer comments
0.6	10/10/2016	Giovanni Aloisio, Silvia Mocavero, Panos Adamidis,Joachim Biercamp	Revised version after reviewer's comments integrated
0.7	07/11/2016	Marie-Alice Foujols	Reviewer comments
1.0	08/11/2016	Giovanni Aloisio	Final formal release

Abstract

During the first three years of the project, some activities of WP3/NA2 have been focused on the analysis of NEMO and ICON models computational performance in order to identify the main limits to their scalability. Moreover, the technology tracking activity has allowed putting together the main European and global initiatives focusing on the models improvement at exascale as well as bringing this knowledge into the ENES community. These preliminary activities allow defining a strategy for the two models re-design, taking into account the technology trend. The present document aims at proposing the re-design strategy.

Project co-funded by the European Commission's Seventh Framework Programme (FP7; 2007-2013) under the grant agreement n°312979

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants including the Commission Services	
RE	Restricted to a group specified by the partners of the **IS-ENES2** project	
CO	Confidential, only for partners of the **IS-ENES2** project	

Table of contents

1. Executive summary

NEMO[1] is a modeling framework for the study of the oceans and their interaction with other components of the Earth's climate system (such as the atmosphere, marine vegetation, etc.) at different resolutions in both time and space. NEMO is developed by a consortium of six partners (CNRS[2], Mercator-Ocean, Met Office, NERC[3], INGV[4] and CMCC[5]).
The prognostic variables of the model are the three-dimensional velocity field, a linear or non-linear sea surface height, the temperature and salinity. Along the horizontal direction the model uses a curvilinear orthogonal grid of discretization, arranging the variables according to the three-dimensional Arakawa C-type grid. Along the vertical direction, however, the space is discretized using the *z*-coordinates, or *s*-coordinates or a mixture of the two. NEMO is parallelized using MPI with domain decomposition in latitude/longitude [1].
The NEMO ocean model is central to a number of European projects[6]. Portability, longevity and computational performance are therefore highly important to the NEMO users and developers. The next generation computer architectures will enable the resolution and the complexity of climate models to be increased, exploiting the resources' capability. However, for this to be fully realized, the scalability of current models has to be improved, e.g. through the design of new parallel models. Indeed, when the number of parallel processes increases up to hundreds of thousands of cores, the main bottlenecks to scalability are the communications overhead and the memory access. More generally, data movement at all levels becomes the main limiting factor. The current version of the code has not been designed for the high levels of parallelism in computer architectures that is the norm today and will continue to be.

ICON[7] is a next generation earth system model designed to simulate multiple scales of the atmosphere processes, enabling both climate simulations and numerical weather predictions. It is a joint development of MPI-M[8] and the German Weather Service DWD[9]. It provides the option to run locally nested highly refined resolutions, allowing simulations at a very fine scale (more than 4 Million horizontal grid cells with 150 vertical levels). ICON is a non-hydrostatic global model with a local zoom function. Its dynamical core solves the fully compressible non-hydrostatic equations of motion for simulations at very high horizontal resolution [2]. The system of equations is solved in grid point space on a geodesic icosahedral grid, which allows a quasi-isotropic horizontal resolution on the sphere as well as the restriction to regional domains. The primary cells of the grid are triangles resulting from a Delaunay triangulation, which in turn allows C-grid type discretization and straightforward

---

[1] Nucleus for European Modeling of the Ocean
[2] Centre national de la recherche scientifique
[3] Natural Environment Research Council
[4] Istituto Nazionale di Geofisica e Vulcanologia
[5] Centro Euro-Mediterraneo sui Cambiamenti Climatici
[6] http://www.nemo-ocean.eu/About-NEMO/Projects
[7] Icosahedral non-hydrostatic general circulation model
[8] Max Planck Institute for Meteorology in Hamburg
[9] Deutscher Wetterdienst

local refinement in selected areas [3]. Furthermore, the LES configuration of ICON aims at resolving cloud and precipitation processes, so as to reduce significantly the uncertainty in climate change projections and numerical weather predictions. The typical grid resolution for such simulations is 150 m, thus making them computationally and data intensive. For this it is vital to be able to exploit the hardware resources of Exascale-HPC systems in an optimal way. Current and future HPC systems are massively parallel computers consisting of hundreds of thousands of cores. This requires a very good scalability of the model in order to be able to use such architectures efficiently so as to complete the simulations within a feasible time frame.

Starting from the experience on the analysis of the NEMO and ICON models carried out during the IS-ENES2 project, in particular in WP3/NA2 Task 2.2, and from the knowledge provided by other European and global initiatives on the models improvement strategies at exascale, the aim of this document is to trace a roadmap for the NEMO and ICON models re-design, which aims at increasing the model performance by exploiting the characteristics of the new generation architectures.

The document will address the following objectives:

- The need to define a standard methodology for the model performance analysis, which takes into account the system architecture. This kind of analysis has to be as much as possible closest to the architectural level allowing the identification of the main bottlenecks which avoid the model to properly exploit the hardware resources. Performance analysis results are usually affected by the model configuration (e.g. domain of interest, resolution, complexity, etc.) and by the target machine used to perform the benchmarks. So, to answer the question "*what are the main limits to the computational performance?*", we need to define both a set of target configurations and architectures.
- The need to improve the model performance in the short-medium term through evolutionary approaches. Starting from the results of the analysis, an incremental migration of the code to massive parallel systems (as suggested by the ETP4HPC Strategic Research Agenda [4]) can be performed, allowing to reach some benefits in the short-medium term and to deliver intermediate releases of the code to be used by the community without affecting the production activity.
- The need to re-design the model using revolutionary approaches. While ICON is a next generation model and has been designed to solve high-resolution predictions, NEMO has been designed and implemented some decades ago. The re-design from scratch would be needed, especially considering the architectural radical changes in the HPC landscape. Rethinking the model at the different layers of the mathematical stack is a long-term research activity, but preliminary efforts in this direction can be addressed in a shorter timeframe to pave the way for the more ambitious goal.

2. Performance analysis

One of the main aspects of a model re-design is to understand the reasons of its performance limits. NEMO and ICON models have been analyzed during the IS-ENES2 project [5] [6] [7] [8] in order to identify the main bottlenecks to the scalability. Each analysis has been performed on one or more configurations; it has been the preliminary step toward the design of code optimizations and their implementation (i.e. in [9]). The analyses of scalability performed within WP3/NA2 are of interest for the community, but other aspects, such as the single-node performance of the model, have to be taken into account to fully exploit the HPC systems (HPCs).

Moreover, the definition of a standard methodology for the analysis requires (i) deep knowledge of the HPCs, in order to choose the right metrics; (ii) strong accuracy in the selection of the profiling tools, considering also their portability and usability and (iii) careful choice of representative HPCs, taking into account the future technology trends.

The first step regards choosing a set of metrics to measure how the code is able to exploit the hardware resources (e.g. memory bandwidth, Flop/s, instruction throughput, etc.). One of the most effective methods is based on the *roofline model* to evaluate if a specific kernel is memory or computational bounded. The metrics to be used in the roofline model are the bandwidth to main memory and the number of floating point operations per byte loaded from the main memory. However, other metrics allow understanding why the performance of the single kernel is limited. Some examples are (i) the level of efficiency in the exploitation of the memory hierarchy, (ii) the evaluation of the strided memory access level, also with respect to SIMD[10] utilization, and (iii) the number of non-packed and packed load/store instructions. Moreover, some metrics related to the energy consumption have to be included to provide information on the application energy profiling. Primitive and aggregate mathematical operations have to be considered as well as an estimation of the energy cost for each of them and for the load/store operations.

The second step is the choice of the most suitable tools to extract the performance metrics. The main issue is the selection of the proper tools able to be integrated within the code execution in order to provide profiling outputs at runtime on several architectures. The model should integrate a standard profiling (e.g. interfaces to the PAPI[11] libraries) which provides, at each execution, the counters values, in a transparent way for the user. Metrics and tools have to be chosen taking into account their portability on different architectures.

The last step is the choice of a set of heterogeneous systems, also equipped with accelerators/coprocessors.

[10] Single Instruction Multiple Data

[11] Performance Application Programming Interface

3. Evolutionary approaches for the NEMO and ICON performance improvement

The NEMO model is parallelized using MPI with horizontal domain decomposition. The increasing number of nodes on emerging HPCs requires an improvement of the code parallel efficiency in order to exploit as much as possible their capability. At the same time the exploitation of the theoretical single-node performance is very important, taking into account that all large-scale parallel computers now and for the predictable future will be assembled of shared-memory nodes on a high performance interconnection.

The parallelization model followed in ICON is also the message-passing paradigm (MPI). Thereby, horizontal domain decomposition is applied in order to divide the computational domain in subdomains, which are distributed among the MPI-processes. Moreover, thread level parallelism is introduced by employing OpenMP directives to parallelize loops. We have observed that for experiments with very high resolution the major limiting factor to good scaling is due to the memory issues. This means, that with increasing number of MPI-processes the memory consumption per process is increasing as well. Whereas, we need a decrease in memory usage per process with growing number of processes. This is especially important when taking into account that for new machines, the memory per core usually is less than for older generations.

3.1 Scalability

The increase of the number of nodes available on modern HPC systems requires some work to improve the code parallel efficiency with a specific focus on the reduction of data movement overhead. Some techniques could be applied to both NEMO and ICON models to improve the computation/data movement ratio, such as:

- the increasing of the halo size to reduce the communication frequency. This would allow reducing the communications overhead by carrying out additional computation instead of exchanging data; the modification would be almost transparent to the other developers, since the logic of the parallelization is the same. This technique has been already applied in the NEMO SOR[12] [12] solver. An extension to the whole NEMO code and to ICON should be considered;
- the overlap of computation and communication using non blocking communications. This would require a complete revisiting of the code and the full understanding of the new logic of exchanging data, which is based on four steps: (i) computation of the halo zone, (ii) use of non blocking communications to start the data exchange, (iii) computation of the inner domain and (iv) end of communications. The impact on the code developments is not negligible;
- the integration of MPI3 communications (collective neighbors communications) instead of point-to-point communications. The introduction of MPI3 communications should only require the modification of the data exchange routines. In NEMO, the

[12] Successive Over Relaxation

benefits could be appreciable especially on the kernels using a 5-point stencil instead of 9-point. The analysis of the code to define if and where the 9-point stencil is really needed has to be the first step. The integration of the improvement into the code is transparent to other developers and does not require additional code maintenance;

- the minimization of in-memory data movement. During MPI communication many gather-scatter operations are done in order to move data between send/receive buffers and their respective memory locations. In order to optimize MPI communication while reducing in-memory data movement the YAXT[13] library, developed at DKRZ could be integrated. YAXT is designed on top of MPI and makes sophisticated use of MPI Datatypes in order to avoid copying data inside the main memory. Moreover, the aggregation of the messages resulting in fewer, longer messages instead of many short messages would reduce message start-up overheads which is highly beneficial, especially for systems with higher latency interconnect. YAXT uses optimization techniques like message aggregation so as to maximize the utilization of the full bandwidth of the underlying network. The integration in both NEMO and ICON model could limit the communications overhead;
- the limitation of I/O overhead. The concept of asynchronous I/O servers is a feasible solution. The processes are divided into two groups. The one consists of compute processes, which are doing the actual calculations. The other contains I/O processes, which are dedicated to purely I/O tasks and run concurrently to the compute processes. As soon as a compute process has produced data for output the I/O process is collecting data from one or more compute processes and preparing them for output. To this aim, the last NEMO version (3.6) integrates XIOS[14]. A similar solution is designed for ICON (CDI-PIO parallel I/O library), handled by YAXT.
- techniques of in-situ visualization and online diagnostics/feature tracking should be also evaluated to reduce data movement for both NEMO and ICON the models.

3.2 Memory requirements

Future HPCs are massively parallel computing platforms consisting of several thousands to hundreds of thousands of cores with distributed memory. The new technologies will probably have less memory per core. Some models consume most memory in I/O and domain decomposition. Our analysis of ICON lead to recommendations for improving memory scaling, which were taken on and successfully implemented within the German HD(CP)2 project. ICON initially performed a domain decomposition starting by reading in the grid data from a file during the initialization phase. Each process computes then the decomposition of the global computational domain. Afterwards, all necessary information regarding the subdomain owned by a process, along with the relationships to the neighboring subdomains, which belong to other processes are calculated locally on every process. In this way the communication patterns are set up. Thus, multiple arrays with dimensions proportional to the

[13] Yet Another exchange Tool
[14] XML-IO-SERVER, http://forge.ipsl.jussieu.fr/ioserver/wiki

global grid size are required by the particular domain decomposition algorithm to exist on every process. Till now this had not been a problem, because the grids were small enough so that the memory consumed did not have a significant impact on the total memory footprint. However, with the grids needed for the targeted very high spatial resolution this becomes a problem, because these arrays exceed the resources available to individual processes. For the ICON model new parallel algorithms that already decompose the input data over all processes of the model and hence reduce memory consumption per process were considered and successfully integrated.

Moreover, after domain decomposition, some more fields of the grid file are read in. It also used a single read of the whole field on a single ICON process followed by a broadcast. Afterwards, each process extracted the data associated to its local part of the domain and discarded the rest. As it is not required to have the whole grid data fields to be available on all processes, each processes should read directly the required part of the field. Again, a new algorithm could be implemented which selects a subset of all processes. Each of these processes read in an individual contiguous part of the file. Using a previously generated communication pattern, the data are then redistributed from the “input-decomposition” to the actual domain decomposition in a single communication step. This way, at no time any process needs to store a global grid array in its memory.

This strategy could (and, if scaling to very high core counts is the goal has to) be adopted by models which hit the memory limit. It has to be observed that our experience shows that for a complex climate model such as ICON the necessary restructuring of the code is an extensive work that may take several person years and needs to be planned well in advance.

3.3 Single-node performance

Models single-node performance is always very far from HPCs peak performance. It can be improved (i) reducing the bandwidth requirements by using techniques to optimize memory access and cache reuse and (ii) increasing the computational efficiency of each core (e.g. techniques to increase SIMD instructions) and each node (e.g. techniques to increase the multi-threading exploitation).

The integration of a second level of parallelism (based on the shared-memory paradigm), would allow reducing the memory per core and increasing the number of threads/core, addressing issues related to compute-bound loops (many operations per load from memory) and memory bound loops (where read data is shared, so that cache memory can be used more efficiently).

Moreover, the introduction of a second level of parallelism allows taking advantage of the use of accelerators and coprocessors.

NVIDIA GPUs are currently the most popular accelerator option for scientific computing. Hundreds of lightweight cores can be used in lock-step increasing the SIMD instructions. Moreover, when routines are memory bounded, the high peak memory bandwidth offered by GPUs is an attractive prospect.

The use of GPUs provides also significant improvements in energy performance as compared to traditional multicore CPUs. It is an important factor if we consider that one of the main technological challenges is the adoption of solutions supporting new and emerging applications that require low-energy computing from both an architectural and application perspective.

The development of a NEMO GPU version implies the choice of the parallel programming model [10]. In terms of programmability, the use of OpenACC directives is the easiest one to be implemented and maintained. Indeed, it has a minimum impact on the original code and it is very simple to apply, also for climate scientists who do not have strong expertise in parallelization issues. Moreover adding OpenACC keeps the code portable and makes it still run on CPU if compiled without OpenACC.

On the other hand with OpenACC directives the memory transfers between host (CPU) and device (GPU) might limit the performance when individual subroutines are accelerated. For this purpose data need to be kept on the accelerator between subroutines.

A more efficient solution could be based on the use of CUDA. However, the need to maintain the parallelization strategy in future developments and the limited skills in CUDA of the climate scientists would compromise the maintenance of this solution.

A very similar hybrid parallelization strategy is represented by the introduction of OpenMP directives to improve single-node performance. The hybrid version of the code could be also executed on Intel MIC architectures, in combination or not with offload directives.

Three different strategies for the introduction of the second level of parallelism, based on the OpenMP shared memory paradigm, have been already evaluated on several architectures [11]. The best one was the parallelization on the vertical levels. The impact could be very high on many-core architectures (especially if the SMT is available), while the modification would impact the whole code and its maintenance would require an acceptable effort.

4. Revolutionary approaches for models re-design

The application of incremental improvements to the code allows increasing the computational performance and, at the same time, preserving the roadmap for other model developments, since they are designed taking into account the impact on the code structure and the effort for their maintenance. However, the performance growth due to this kind of optimizations is limited on future architectures and a radical change is needed.

NEMO is discretized on a staggered grid (Arakawa C grid) and the numerical techniques used to solve the PDE are based on the traditional, centered second-order finite difference approximation. The use of finite volume with non-rectangular grid cell could improve the grid refinement needed to use NEMO for coastal applications. However, the switching to unstructured meshes (already adopted in ICON) would increase the need of computational resources due to the use of implicit schemes, generally very expensive from the computational point of view. New solutions at algorithmic level should be taken into account to improve the computational performance. An example could be the integration of communication-avoiding algorithms, aiming at minimizing communications in numerical linear algebra. This approach requires a deep analysis of the computational kernels and a revisiting of its mathematical formulation.

An algorithmic analysis of ICON shows that when decoupling the radiation component from the dynamical core, a new time stepping scheme could be implemented which allows the concurrent execution of the radiation component and the dynamical core in a similar manner as referred to in [13]. In this way, the CPU time spent in the radiation part could be hidden from the dynamical core. Based on the fact that in terms of CPU time one radiation step is equivalent to many time steps of the dynamical core, a performance boost can be expected.

This requires a re-design of the current parallelization concept of ICON, from being based purely on domain decomposition towards a mixed concept, which in addition enables functional/task parallelism. For this we would need to enhance the abstraction level of the communication lib YAXT, to allow component concurrency in MPI (Message Passing Interface) process-space. Furthermore, this new concept would allow exploiting modern hybrid architectures consisting of GPGPU's or other type of accelerators, since it provides the ability to place the separated functional components on different devices, based on the respective performance gain from new the hardware.

One of the main limits of climate applications is the sequential execution along the time direction. For example, NEMO relies on explicit time-stepping algorithms that are subject to the CFL[15] stability criteria; therefore increasing space resolution implies increasing time resolution. Even if the existing software infrastructure of ICON, allows solving the equations on different nested grids, the equations are solved in a sequential order on each grid on every time step. The parallel in time methods, which application to the climate models is still under investigation [14], could both overcome limitations due to the sequential nature of the simulation in the time dimension, and enables both NEMO and ICON to better exploit exascale-class systems with much higher levels of parallelism.

In order to investigate the very appealing parallel in time method, we would need to perform calculations on several grids with different time and consequently spatial resolution, which means that a re-design of the current models is necessary.

In both cases, the model should be completely revised and much of the code rewritten. This would require the cooperation of numerical and computational scientists, such as of technology providers, very close to the hardware features of the next-generation HPCs. The process is not easy and even if different skills have to be conjugated, it is also suitable to decouple it from the scientific development work of the new model.

The experience reported by the UK team during the Gung-Ho and the gOcean projects[16] may be an example of a decoupling methodology.

In this case, the central principle is the division of the code into layers to separate the natural science and computer science aspects. The identification of these layers enables a "separation

[15] Courant-Friedrichs-Lewy

[16] The GungHo project is a joint project between the Met Office and NERC funded under the Joint Weather and Climate Research Programme (JWCRP) and is being developed in association with STFC Daresbury. gOcean is an experiment with the application of a GungHo approach to a subset of the NEMO code.

of concerns" between largely independent components which have different requirements and need developers with very different skill sets.
Using this approach named PSyKAl (Parallel System, Kernel, Algorithm), the model code is re-structured into three layers: natural scientists are responsible for the Algorithm and the Kernel layers while the whole parallelization strategy is contained within the PSy layer, designed and implemented by the computational scientists.
Applying this approach would mean completely re-designing the PSy of the model without worrying about the impact that this process could have on the other layers. Consequently, the maintenance responsibility of the PSy layer would be only of the computational scientists.
A similar approach has been used in the DYNAMICO[17] project with a clear separation of numerical and parallel subroutines. Parallel routines could be optimized for different architectures (e.g. through OpenMP directives, a simple method based on standards).

5. Conclusions

This document is the result of a deep analysis of the NEMO and ICON models and their limitations from the computational point of view. The proposed solutions to overcome these limits are derived from the knowledge of the model and the analysis of the outcomes of other initiatives on the re-design of climate models at exascale, collected during the IS-ENES2 project.
The complex architecture of Exascale systems makes it hard to develop software which exploits the hardware in an optimal way. There is a big gap between delivered peak performance by a system and applications' sustained performance. The two major bottlenecks are communication in the sense of moving data between the levels of the memory hierarchy inside nodes/cores or among processors and fast parallel I/O.
Avoiding communication by increasing temporal and spatial locality is a key issue. One approach to achieve this is by improving the hybrid MPI/OpenMP parallelization. Achieving better thread scaling is one of our major concerns not only for minimizing communication between MPI processes but also for utilizing efficiently future many core architectures equipped with accelerators as coprocessors. In this regard load balancing is very crucial and it is a problem which can be solved only on the algorithmic level after a thorough analysis.
The proposed solutions have been discussed within the models development teams. Concerning NEMO, the development perspectives will be clearly defined in the update of the "NEMO Development Strategy" document to be published by NEMO Consortium in 2017. Some evolutionary approaches aiming at improving the model scalability and memory issues have been integrated as part of the future development strategy and included as assets of the ESiWACE Center of Excellence; others related to the analysis and improvement of the single-node performance have been included in a EO-3-2016 proposal[18] within the H2020 work programme.

[17] http://forge.ipsl.jussieu.fr/dynamico
[18] https://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/2234-eo-3-2016.html

The most disruptive approaches will have to be scheduled in the medium/long-term and could be the main subject of an H2020 FET-HPC[19] proposal.

6. References

[1] Madec G. and the NEMO team, NEMO ocean engine, Note du Pole de modellisation, Institut Pierre-Simon Laplace (IPSL), France, 27, ISSN 1288-1619, 2008

[2] Gassmann, A., and H.-J. Herzog, Q. J. R. Meteorol. S., 2008. : Towards a consistent numerical compressible non-hydrostatic model using generalized Hamiltonian tools.

[3] Gassmann, A., J. Comp. Phys., 2011 : Inspection of hexagonal and triangular C-grid discretizations of the shallow water equations.

[4] ETP4HPC Strategic Research Agenda achieving HPC leadership in Europe, 2015 update.

[5] The NEMO oceanic model performance analysis – CMCC contribution (https://verc.enes.org/ISENES2/documents/na2-working-documents/the-nemo-oceanic-model-performance-analysis/view)

[6] The NEMO oceanic model performance analysis – BSC contribution (https://verc.enes.org/ISENES2/documents/na2-working-documents/task-2-2-computational-cores-scalability-porting-and-testing-this-documents-reports-a-preliminary-insights-on-nemo-performance-its-scalability/view)

[7] Epicoco I, Mocavero S, Macchia F, Aloisio G (2014). The Roofline Model for Oceanic Climate Applications. Workshop on High Performance Computing for Weather, Climate, and solid Earth Sciences HPCS2014, Bologna, Italy, July 21-25, 2014 pp. 732-737, ISBN: 978-1-4799-5311-0

[8] The ICON atmospheric model, Part 2: Scaling Analysis based on the HD(CP)2 setup (https://verc.enes.org/ISENES2/documents/na2-working-documents/the-icon-atmospheric-model-scalability-analysis/view)

[9] NEMO north-fold optimization (https://verc.enes.org/ISENES2/documents/na2-working-documents/nemo-north-fold-optimization/view)

[10] Porter A., Pickles SM., Ashworth M. Final report for the gNEMO project: Porting the oceanographic model NEMO to run on many-core devices. DL Technical Reports DL-TR-2012-001. 2012.

[11] Epicoco I., Mocavero S., Porter A., Ashworth M., Aloisio G. "Hybridisation Strategies and Data Structures for the NEMO Ocean Model", accepted for publication on the International Journal of High Performance Computing

[12] Epicoco I., Mocavero S., Aloisio G., The performance model for a parallel SOR algorithm using the red-black scheme, Int. J. High Performance Systems Architecture, Vol. 4, No. 2, 2012.

[19] https://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/calls/h2020-fethpc-2016-2017.html#c,topics=callIdentifier/t/H2020-FETHPC-2016-2017/1/1/1/default-group&callStatus/t/Forthcoming/1/1/0/default-group&callStatus/t/Open/1/1/0/default-group&callStatus/t/Closed/1/1/0/default-group&+identifier/desc

[13] Balaji, V., Benson, R., Wyman, B., and Held, I.: Coarse-grained component concurrency in Earth System modeling, Geosci. Model Dev. Discuss., doi:10.5194/gmd-2016-114, in review, 2016.
[14] Martin Schreiber, Adam Peddle, Terry Haut, Beth Winget : A Decentralized Parallelization-in-Time Approach with Parareal, Technical Report: eprint arXiv:1506.05157v2[cs.DC]