Resilience in an ocean model

Strategy, implementation and validation

TR-CMGC-13-110

Eric Maisonnave
CERFACS-CNRS
Toulouse, France
eric.maisonnave@cerfacs.fr

Vladimir Slavnić
Scientific Computing Laboratory
Institute of Physics Belgrade, University of Belgrade
Belgrade, Serbia
vladimir.slavnic@ipb.ac.rs

Abstract— **Towards Exascale climate model implementations, we propose to explore the possibility of resilience in NEMO ocean model, without the help of spare resources and avoiding to handle the associated on-line data recovery. A simple, non intrusive error repair has been implemented and tested on GYRE idealized basin at 1/12 degree resolution, and failures simulated on PRACE Tier-0 supercomputer. Our solution does not ensure heat content conservativeness nor experiment reproducibility but large scale characteristics of ocean circulation are well preserved, including a case when failures continuously affect calculations over more than 1‰ of the total domain area.**

Keywords: resilience, Exascale, MPI, ocean modeling, meso-scale

I. INTRODUCTION

For decades, climate modeling has been benefiting from the computing progress; now, to follow super-computing hardware and software evolution, legacy FORTRAN programs, made of millions of lines, have to be adapted or partly re-written in order to achieve more efficiency.

Model parallelism has already been improved: by changing the spatial discretization, new dynamical cores avoid costly filtering at pole regions [3]; with better organization of data throughput, new scientific libraries reduce important bottlenecks [10].

New supercomputing facilities are fully exploited to increase model resolution, better represent low scale phenomena and understand scale interactions [4]. Most of the TOP500 machines have been, and presently are, employed by climate modeling groups all over the world (K supercomputer in Japan, Oakridge or Argonne machines in the United States, PRACE Tier-0 facilities in Europe, etc.).

In this context, given that machines gathering million to billion of computing cores are a major target for climate modeling community, emerging issue of the fault tolerance needs to be taken into account.

We can assume that the type of communication within our future Exascale compliant climate models will still be based on MPI paradigm. Assuming that MPI itself will be enhanced [11] to satisfy Exascale constraints [12], an additional work on our FORTRAN programs will be required to adapt their MPI interface to this new MPI behavior [5]. Obviously, this will have an impact on climate model calculations.

We assume that MPI will detect and provide information about failure characteristics (error detection) but, at this point, model modifications will be necessary to start the repair process and/or contain fault propagation (error handling).

II. PURPOSE

In this study, we are proposing to describe a strategy that will lead to implementation of a resilient climate model component (ocean), we are providing details of its first implementation and validating it experimentally. This practical work comes up against different difficulties.

As standards are still a work-in-progress, the different possible behaviors of fault tolerant MPI parallel library need to be foreseen. Indeed, the message passing system is a key tool to articulate a resilience strategy on scientific programs such as climate models: when a failure on a certain MPI process is detected, a repair process can be started on all remaining MPI processes and the calculation can be continued.

The structure of a repair process strongly depends on the kind of preparatory error detection and handling provided by the new MPI implementations, as well as their different performances. They could be suitable or not for the specificity of climate modeling. One of the challenges on resilience strategy efficiency lies in this choice: a high level of error handling (full data recovery, reallocation of missing resources) that requires too much time (comparable to the duration of the whole experiment) cannot be considered as efficient.

Consequently, as a first step, hypotheses on the best suitable MPI error handling have to be made, considering both state-of-the-art standard of fault tolerant MPI and versions of climate models with the highest level of parallelism. In order to validate some of those hypotheses, a repair process has been implemented in one selected module of our climate system and it will be described in this manuscript.

An effective test of a resilient version of our model requires both hardware (Exascale machine) and software (massively parallel climate model) environment that will not be achievable for several years. As a first step, we will choose an existing climate model configuration and mimic conditions that will predominantly exist in an Exascale machine. Our first objective is not to quantitatively validate a fully resilient version of a climate model, but to confirm the first hypothesis on which our implementation strategy is based, that recovery process duration can be achieved in a reasonable time and that resulting errors do not lead to numerical scheme divergence.

Generally speaking, we expect that this experiment will give us a clearer idea on how to design the future version of climate models (massively parallel, using graphics processing units or many integrated core processors, etc.) that will be developed in the coming years, taking into account resilience requirements. This work would be strongly facilitated if an error handling strategy, simple, efficient and compatible with climate modeling requirements, was already tested.

III. METHOD

A. Climate model and Message Passing Library

To the best of our knowledge, there is no existing full or partial climate model implementation that could be considered as resilient. We define full climate model as a complete system necessary to simulate the Earth's climate with enough complexity to address scientific questions such as climate change or seasonal to decadal predictions. CGCMs (Coupled Global Circulation Models), encompassing different modules (atmosphere, ocean, land surface, sea-ice, etc.) and gridded on the whole globe, are mandatory for such experiments. Those components can be assembled in a single executable [6] or launched in several parallel executables, coupled with an appropriate tool [7].

In order to simplify our problem, we decided to choose the ocean part of the two essential components of the ARPEGE-NEMO CGCM, widely used on the most advanced supercomputers. Indeed, NEMO ocean model [1] combines standard characteristics of climate model components such as Fortran writing and MPI parallelism following two-dimensional spatial decomposition. Moreover, the NEMO code length is compatible with a quick adaptation for an error handling; its MPI interface is isolated in a single file with only 23 routines.

Like climate modeling experiments, standard ocean modeling experiments last for days or months, which means that its three-dimensional prognostic variables have to be regularly checkpointed on disk (every simulated month to every simulated year) and experiment restarted from this point.

The multi-executable structure of our CGCM allows separate adaptation of a single module. The fault tolerant NEMO version will be used in stand-alone mode. But it must be clear that, to fully evaluate performances and reliability of a comprehensive climate model, the other modules and the coupler should also be made resilient, which is out of the scope of this study. Nevertheless, modularity of the system should facilitate incremental modifications. Resilience can be achieved separately on each component and that will eventually lead to a full resilient climate model when these components are assembled with a resilient coupler.

NEMO model has a three-dimensional spatial discretization of the whole planet (spherical coordinates). Its parallelism is based on splitting the longitude/latitude computation domain into several smaller domains and solving the set of equations by addressing independent local problems. Each computing core processes the model equations over a sub-domain of the whole model domain. The MPI library is used to exchange information at sub-domains boundaries, mostly through point-to-point communications, several time per time step, for several variables.

We assume that future implementations of a resilient MPI will detect and begin to handle failures at the communicator level and let the application manage the repair [13]. We suppose that this implementation will be most similar to those previously developed on FT-MPI/OpenMPI.

B. Error handling strategies

1) Possible strategies

In NEMO code, error handling begins when an error is signaled by any MPI library call. We have identified 3 possible error repair strategies ensuing 3 different MPI error handling types:

a) No repair process, MPI is disabled. It implies that model has to checkpoint on disk its current prognostic variables, stop and restart.
b) The failed process is replaced by a new resource. It implies that the model has to recover the lost part of its prognostic variables and resume the experiment at preceding time step.
c) The failed process is not replaced but point-to-point communications are still possible except with the failed process, and a special treatment is done for collective communications. It implies that experiment can go on, but with the missing information.

A simple analysis of NEMO's present behavior on scalar machines already reveals how expensive checkpoint/restart operations are. Their cost is supposed to increase [8] with resolution (Exascale computing will address problems of more and more accuracy) and parallelism (parallel access to disk or cache memory). Climate modeling is already considered as one of the most consuming science disciplines regarding to memory and output requirements. As previously mentioned, the regular duration of a climate simulation experiment is too long to consider the possibility to perform it in a single run. But, to satisfy future Exascale constraints, we think that error treatment only based on a checkpointing-restart strategy has to be avoided; to perform a checkpoint/restart operation as often as failures occur would lead to spending most of the time of simulation in I/O operations rather than in calculation.

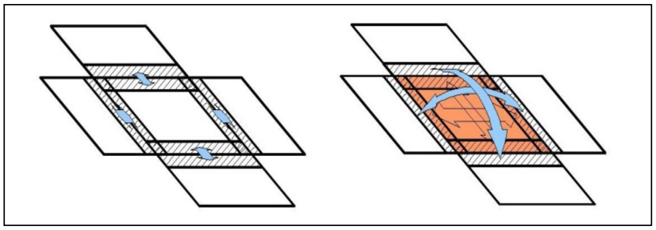

Figure 1. First level of implemented error repair strategy. Shaded arrows represent MPI exchanges necessary to fill extra lines/columns (hatched) at boundaries of sub-domains. Left: a regular situation. Right: a situation after failure of central sub-domain (shaded area). Transparent arrows represent inner communications on pseudo-failed sub-domain.

The same data volume considerations suggest not to deal with data recovery. This operation implies to continuously keep an on-line copy of an important volume of data, certainly uneasy to transfer to the spare memory resources during error recovery. Moreover, this additional cost does not include the additional time needed by MPI management to dynamically reallocate an equally efficient spare resource.

2) Mixed error handling

The error handling strategy we propose is a combination of the last two possible strategies. It is based on the following assumption: when a failure occurs, a numerical experiment can be carried out despite missing data and missing calculations on the sub-domain corresponding to the failed resource.

A clear consequence is the non-reproducibility of experiments. Moreover, data located on the "failed sub-domain" (i.e. the sub-domain covered by the failed resource) are lost and the model no more ensures energy and mass conservation. This bias will have to be evaluated and compared to other sources of non-conservation.

The main argument that leads us to move forward in this direction is that increasing parallelism leads to decrease the relative failed sub-domain size, compared to the global domain. The question is: at which parallelism level could the bias and its propagation be considered as a simple perturbation ? Even though this limit can not be found on present machines, a test case configuration of our model has been chosen to represent the problem as realistically as possible and give us a first idea on the acceptable limit.

C. Implementation

To reduce the error impact, our error repair strategy combines two repair levels:

- On-line error handling: After failure detection, calculations are resumed, but no spare resource is needed and no data recovery tackled: model algorithm is modified to keep exchanging boundary conditions at sub-domain limits through the failed sub-domain, using values of the nearest valid sub-domain neighbor.

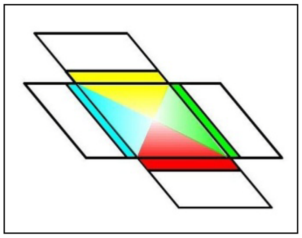

Figure 2. Second level of implemented error repair strategy. During the next regular checkpoint/restart phase, the restart file is repaired: missing grid point values are extrapolated from values of grid point neighbors. Simulation can be resumed on the whole domain.

- Restart file repair: When the next regular checkpoint is performed, values of nearest valid sub-domain neighbor grid points are extrapolated to fill values of prognostic variables over the failed sub-domain. Calculations can then be started again with regular resources number.

1) On-line error handling

When a failure occurs on one resource, our model is able to detect and handle it only when an MPI communication is needed with the failed sub-domain. At this stage, an error repair is initiated: communication pattern between sub-domains is redefined. For the four sub-domains which are neighbors of the failed-domain, boundary condition communications will no longer be done with the failed domain (Fig. 1, left, regular communications) but with its neighbor (Fig. 1, right, modified communications). Communications can then be resumed and calculations will carry on in this "rescue mode" until the next regular checkpoint.

2) Failure simulation

To temporary simulate failures, which cannot be produced by present hardware and handled by the available MPI library, the failed sub-domain calculations must be stopped (Fig 1, right, shaded area). For our ocean model, this could be done by replacing ocean grid points by land grid points. As communications on the boundaries are still necessary on the sub-domain to avoid MPI collective blocking, eastern-western (northern-southern) boundary conditions are internally exchanged (Fig. 1, right, transparent arrows). An advantage of this solution is that collective communications can go on without any particular correction: for example, global mean values can be calculated via an MPI collective communication with a global domain, pseudo-truncated (masked) on the failed sub-domain.

3) Restart file repair

Simulation is resumed in rescue mode until the next regular checkpoint/restart (usually one month of simulated ocean circulation for high resolution simulations, i.e. 30 to 60 wall-clock minutes). For the moment, our implementation excludes the possibility of a second failure in the meantime. After simulation normally stopped at the next checkpoint, the second repair level of our strategy is activated. Within the restart file, the model prognostics values at the boundaries of the failed sub-domain are extrapolated over grid points located in the failed sub-domain (Fig. 2). Several extrapolation schemes have been be tested:

- Extrapolation of boundary points of the 4 neighboring sub-domains,
- Extrapolation of East/West neighboring sub-domains,
- Simple copy of a whole neighboring sub-domain over the failed sub-domain area.

Even though the first strategy seems the best to minimize heat and mass losses, it appears that only the third one was able to provide stable restart conditions. Nevertheless, the three solutions have been implemented and future investigations on gradient smoothing near failed sub-domain boundaries could lead to reconsideration of this choice.

IV. VALIDATION

A. *Exascale compliance*

Several global or regional NEMO configurations are currently available on supercomputers, but resolutions expected to be used at Exascale are impossible to handle on present machines. On the basis of the last considerations on the topic [9], we guess that a code able to fully exploit an Exascale computer should be parallelized on a billion of cores. On these machines, a node could be composed of 10^3 to 10^4 cores. It is still unclear whether the entire node will be affected entirely or partially by failures. Consequently, the ratio f between the global Earth surface area and the area affected by failure could vary a lot depending on these various hypotheses. We chose to fix it to 1000 (10^6 total cores / 1000 cores per node)

Our error handling algorithm is implemented on a recent NEMO version (3.4) and tested with a regional configuration which is 100 times smaller than the global grid. 1024 computing cores of TGCC supercomputer “CURIE” [2] were necessary to lead reasonably fast simulations. With a failure of one of the 1024 sub-domains, f is equal to 1024: our experiment will then be comparable to what we expect on an Exascale machine. Even though we suppose that reaching this f value on an Exascale machine implies finer model resolution and higher parallelism, the present configuration will help us to quickly test if, with the implemented error handling, and despite failure affecting a non negligible part of its global domain, the model is able to withstand resulting perturbations and process the entire experiment.

B. *Model configuration*

1) *Experimental conditions*

To reach the highest possible level of parallelism, one of the most accurate NEMO configurations available is used. This configuration, meant to represent the North Atlantic ocean, consists of an ideal rectangular basin (GYRE), centered at 30°N and rotated by 45°, 3180 km long, 2120 km wide, similar to the one extensively used to understand resolution impact on various represented physical phenomena [4]. Its eddy resolving resolution of 1/12 degree (about 9 km) and 31 vertical levels allows to represent interactions between meso-scale physics (eddies) and large-scale thermohaline circulation. Strength of the inter-gyre current (representing Gulf Stream or Kuroshio) characterizes this large-scale circulation. External forcing (heat, water) has been set to zero and a constant zonal wind stress of 0.1 m/s is prescribed uniformly. Consequently, our model is not able to represent seasonal cycle and cannot be compared with observations. On the other hand, non conservativeness of our model will be simple to estimate: a measure of sea level will directly give the total volume variation due to mass and heat gains and losses.

2) *Error estimation*

Due to the strong non linearity of models such as the NEMO ocean model, it is not possible to quantify the error by simply comparing the final results of an experiment with a control one: an infinitesimal perturbation occurring in the initial state or during the experiment leads to a totally different result. Error estimation is usually made by integrating quantities such as annual mean values of multi-annual experiments, or ensemble mean values if several members of the same ensemble experiment can be processed. In our case, the error produced by our resilience strategy is evaluated by mean values over the whole experiment, which is relevant for large scale quantities we will describe below.

At our level of parallelism, one failure stops calculations on a 30x20 km^2 region. At this spatial scale, meso-scale physics is locally affected. Although the affected area is significantly smaller than the total domain, we propose to evaluate the large scale effect of disturbances created by our resilience strategy, given that the chosen GYRE eddy-resolving configuration is able to represent interactions between meso-scale eddies and large scale thermohaline circulation.

C. *Experimental setup*

1) *Spin-up*

Reaching the long term adjustment with GYRE configuration requires a spin-up experiment of hundred of years. Due to CPU allocation constraints, only a 10-year long spin-up experiment started from rest and temperature/ salinity analytic profile has produced initial conditions for our fault tolerance experiment. Nevertheless, the double gyre circulation that we want to study here had enough time to be established.

2) *Failures characterization*

Three 10-year long experiments are started from the spin-up result: one experiment (CTRL) without any failure and two experiments with two different failure rates, to quantify failure rate impact. In the “resilient” experiments, one sub-domain is switched off at a time, every simulated month (FULL) and every 6 simulated months (SIXT).

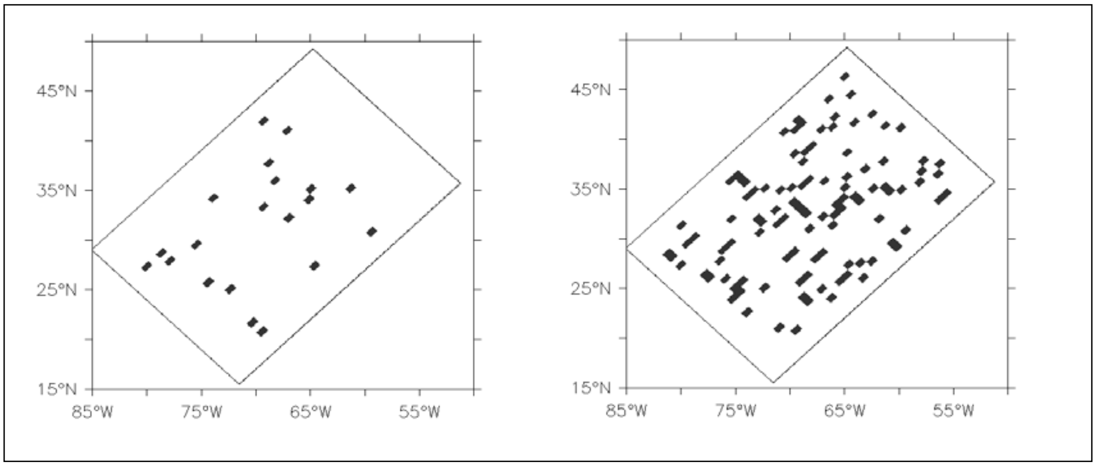

Figure 3. Position of failed sub-domains during SIXT (left) and FULL (right) simulations. The total domain is plotted in longitude/latitude coordinates.

The failed sub-domain is chosen randomly, but failures are avoided in sub-domains located along the domain limit and their neighbors. A sub-domain can be affected only once during the experiment. Failures occur a few time steps after the checkpoint/restart, which means that each failure lasts for one month long run, which is the worst possible case. On SIXT experiment, a failure occurs one sixth of the time. On FULL experiment, at least one failure affects model results all the time. It means that failures affect simultaneously 1‰ of the total sub-domains number all the time: this ratio is the one chosen to fit assumed Exascale conditions. Spatial spreading of failed sub-domains for both SIXT and FULL experiments is shown in Fig 3.

D. Resilience impact on simulation results

The first level of error repair (MPI communication pattern re-definition) is quickly performed (less than 1 second). Rescue mode computations have almost the same restitution time as the regular ones. Consequently, we can conclude that our resilience strategy implementation does not degrade model performance.

Average values for the whole 10-year experiment are calculated for different two-dimension (surface temperature, surface salinity, mixed layer depth, surface height, surface current module) and three-dimension model variables (eddy kinetic energy). The results we present here are based on those quantities. Using the same quantities on two different 5-year long averages did not change the results significantly.

1) Mean state

An inter-gyre current grows from the western boundary at 35°N (Fig. 4, middle). Maximum of its mean value is 1.4 m/s. Correlatively, the mixed layer deepens along the northern boundary (right). The initial zonal temperature gradient is conserved (left), with a slight meridional component along gyres.

2) Meso-scale physics

Ocean kinetic energy can be separated into two parts: the one dissipated by the mean current (large scale) and the turbulent kinetic energy produced by eddies (meso-scale). Our model resolution allows to explicitly represent eddies, whose kinetic energy (EKE) can be directly deduced from the difference between total and large scale kinetic energy. Evaluating values of EKE informs us of the strength of meso-scale processes and how they are affected in "resilient" configurations.

A comparison of EKE averaged for the whole domain and for all vertical levels (Fig. 5) reveals that more energy is dissipated through eddies in “resilient” experiments (in SIXT but, above all, in FULL). It is particularly obvious in the inter-gyre current but can also be noticed north to 35°N in

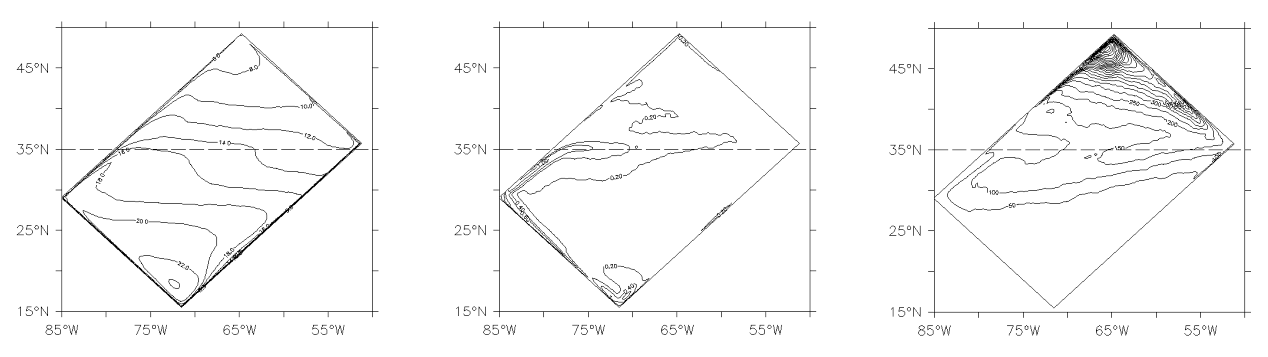

Figure 4. 10-year mean quantities for CTRL simulation. Left: sea surface temperature (C°). Middle: surface current module (m/s). Right: mixed layer depth (m)

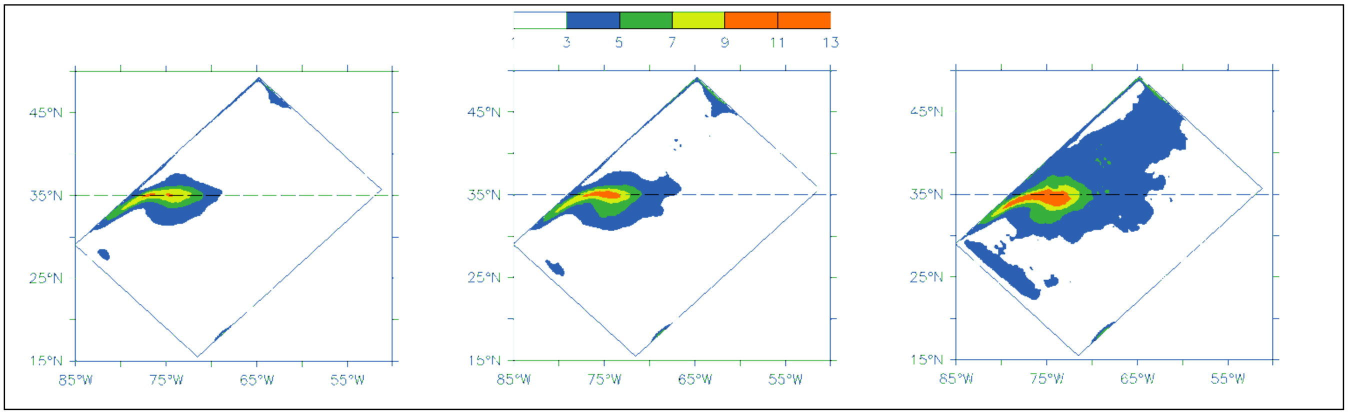

Figure 5. Total EKE (cm^2/s^2) averaged in simulations CTRL (left), SIXT (middle) and FULL (right). The EKE is defined as the total kinetic energy minus the mean kinetic energy.

FULL experiment.

This increase is mainly due to creation of strong and artificial eddies along failed sub-domain boundaries during on-line error handling phase. They disappear after checkpoint-restart repairing phase, even though this phase also contributes to generation of turbulence, after creation of important gradients. Due to the strength of all these eddies, EKE mean value over 10 years is significantly increased.

This effect can be noticed throughout the water column (Fig. 6). SIXT experiment profile is marginally affected. Major normalized anomalies affect lower depth in FULL experiment; the maximum value of EKE over 10 years and all horizontal grid points of the same level is systematically higher.

3) *Large scale physics*

Strength of mean surface current indicates large scale activity of our model. Maximum of experiment mean values of surface current is located at the center of the main inter-gyre current (Fig. 7, left). The additional eddies produced by our repair strategy slightly modify strength and position of the main inter-gyre current. No significant changes can be noticed for SIXT experiment (less than 0.1 m/s compared to the mean value), but FULL simulation exhibits a weaker inter-gyre current (0.2 m/s less), slightly shifted south of 35°N (Fig. 7, right).

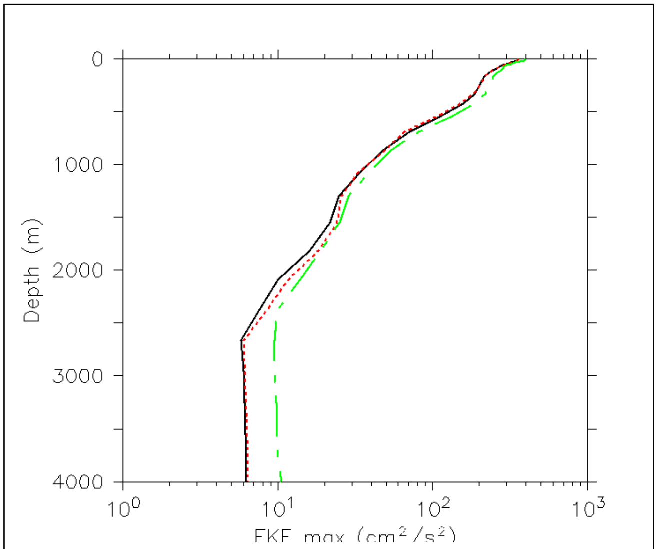

Figure 6. Vertical profile of maximum 5-day mean EKE (cm^2/s^2) during simulations CTRL (solid black line), SIXT (dashed red line) and FULL (dashed-dotted green line)

Modifications of large scale circulation obviously affect its state variables. An analysis is done on sea surface temperature (SST), subtracting and normalizing from CTRL mean values of respectively SIXT (not shown) and FULL (Fig. 8) experiments. We show that decreasing of SST can affect the inter-gyre current zone but also the eastern region where this current leads (northeastern part of the basin). This is more obvious in the high failure rate case. Nevertheless, absolute values (not shown) of these temperature shifts do not exceed a few tenths of a degree.

Non-conservativeness magnitude of our fault tolerance strategy can also be evaluated by checking the evolution during the experiment of mean quantities such as sea surface height (SSH). In CTRL experiment, SSH is constantly zero: NEMO is a conservative model and water and heat energy at it surface is set to zero in our experiment. Gains and losses of energy and mass are randomly endured by our model each time that a failure has to be treated. This happens more often in FULL experiment, which exhibits a final SSH elevation of 8 mm, than in SIXT (elevation of 1 mm) compared to CTRL experiment.

V. DISCUSSION

At meso-scale, turbulence is locally enhanced, throughout the water column, by the fault tolerance strategy that we implemented on NEMO ocean model (on-line error handling and restart repair strategies). Nevertheless, this strong error can be seen as a local perturbation, having in mind that large scale results of our experiments are only slightly modified. Considering the results of two different experiments characterized by two different failure rates, we showed that fault tolerance strategy impact on model's large scale quantities is negligible when failures on 1‰ of the whole domain occur one sixth of the time and becomes measurable when failures on 1‰ of the whole domain occur

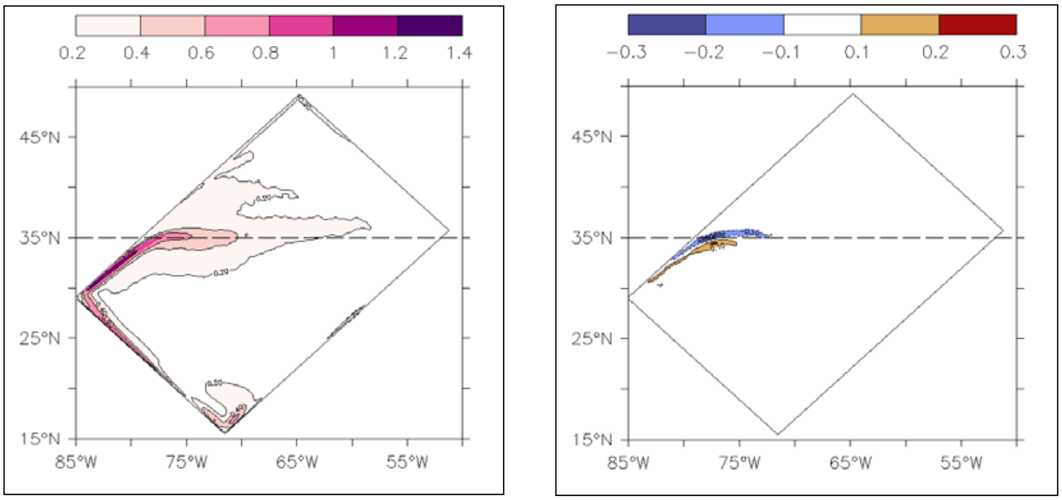

Figure 7. Surface current (m/s) mean values for CTRL (left) and differences FULL-CTRL (right)

all the time (supposed Exascale conditions). In this case, mean values of variables like SST can be modulated. In our test case, this anomaly remains within the range of natural variability or model biases. Its pattern is comparable to any other observed anomaly.

This study confirms however that any future kind of repairing strategy that would favor computation continuation on a degraded mode instead of checkpoint-restart only strategy must be leaded by an estimation of this degradation.

In our case, this estimation would determine which utilization of our model should be avoided: it is clear that, above a certain failure rate that would be necessary to quantify, short-term predictions like seasonal forecast, affected by relatively small large scale anomalies or even strong local gradients, could be significantly slanted by our strategy. But this estimation could also be valuable to improve our implementation: a reduction of artificial eddy production could be set up, increasing viscosity near failed sub-domain boundaries, for example.

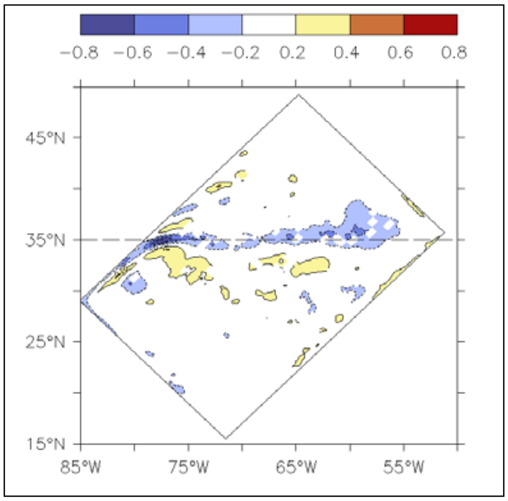

Figure 8. SST composite (standard anomalies) for FULL compared to CTRL experiment

Similarly, model non-conservativeness estimation (in FULL experiment, similar to global warming effect) forbids any utilization of our solution for long term scenarios. Again, substantial improvement of our algorithm will be necessary in such cases. A simple one would consist in evaluating heat and salt balance before and after repairing actions and re-distribution of anomaly to the whole domain.

VI. CONCLUSION

A repairing algorithm has been implemented in the NEMO ocean model. Thus modified, the model is able to face simulated failures of CPU units, carrying on 10-year long calculations despite the lack of one of its sub-domains. The resilient experiment is performed with the same restitution time as a standard experiment. We proved that such simple implementation, which avoids additional allocation of spare resources and associated on-line data recovery handling, is possible on existing HPC compliant models and leads to acceptable results, even though degradations may occur and must be quantified before any scientific study. A strong drawback of this strategy is the non-reproducibility and non-conservativeness, which probably keeps its usage to selected kind of experiments.

Several steps are then needed to deliver a fully operational resilient climate model. Repairing algorithm has to be enhanced to reduce the numerical turbulence created around failed sub-domains. Other possible but less obvious improvements must be implemented to ensure mass or energy conservation. With realistic configurations (not simple rectangular basins but ocean cut along the observed coast line), our repair strategy creates bathymetry gradients that induce strong instabilities. Considering that the present work has been done simulating failures, an additional implementation will be required to be sure that a real failure, detected by a complete fault tolerant MPI-3 library (when available) could be handled by our system. Finally, the same work would be necessary in an atmosphere model and in the ocean-atmosphere coupling framework.

These ample issues would be tackled more easily on ongoing developments than on legacy codes like NEMO. That’s why further improvements of our resilience system would preferentially be done, on a long term perspective, in models with clear Exascale target.

ACKNOWLEDGMENT

The authors would like to thank Ala Rezmerita, Sophie Valcke, Luc Giraud, Olivier Marti, Thomas Hérault, George Bosilca, Sébastien Masson, Rachid Benshila, Christophe Cassou, Romain Bourdallé-Badie, Gabriel Jonville and Jeffrey Blundell for their help. This work has been supported in part by the ANR project SPADES (08-ANR-SEGI-025). V. S acknowledges support by the Ministry of Education, Science and Technological Development of the Republic of Serbia, under projects No. ON171017 and No. III43007, and EU FP7 project PRACE-2IP. We acknowledge that the results in this report have been achieved using the PRACE Research Infrastructure resource CURIE based in France at TGCC.

REFERENCES

[1] Madec G.: "NEMO ocean engine". Note du Pôle de modélisation, Institut Pierre-Simon Laplace (IPSL), France, No 27 ISSN No 1288-1619 (2008)

[2] Lippert, T., Eickermann, T., Erwin, D., “PRACE: Europe's Supercomputing Research Infrastructure” in Applications, tools and techniques on the road to exascale computing, Advances in Parallel Computing, vol 22, De Bosschere K, D’Hollander E, Joubert GR, Padua D, Peters F, Sawyer M, Eds., Amsterdam, The Netherlands: IOS press, Inc.; pp 7-18 (2012)

Article in a journal:

[3] Satoh, M., Matsuno, T., Tomita, H., Miura, H., Nasuno, T., Iga, S.: Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloud resolving simulations, Journal of Computational Physics, Volume 227, Issue 7, 20, pp. 3486-3514 (2008)

[4] Lévy, M., Klein, P., Tréguier, A.-M., Iovino, D., Madec, G., Masson, S., Takahashi, K.: Modifications of gyre circulation by sub-mesoscale physics, Ocean Modelling, Volume 34, Issues 1-2, pp. 1-15 (2010)

[5] Gropp, W., Lusk, E.: Fault tolerance in MPI programs. Special issue of the Journal High Performance Computing Applications (IJHPCA) 18(3): pp 363–372 (2004)

[6] Gent, P.R., Danabasoglu, G., Donner, J.L., Holland, M.M., Hunke, E.C., Jayne, S.R., Lawrence, D.M., Neale, R.B., Rasch, P.J., Vertenstein, M., Worley, P.H., Yang, Z.-L., Zhang, M.: The Community Climate System Model, version 4. J. Climate, 24 (19), pp 4973-4991. (2011)

[7] Valcke, S.: The OASIS3 coupler: a European climate modelling community software, Geosci. Model Dev., 6, pp. 373-388, doi:10.5194/gmd-6-373-2013 (2013)

[8] Schroeder, B., Gibson, G. A.: Understanding Failures in Petascale Computers, J. of Physics: Conference Series, 78, doi:10.1088/1742-6596/78/1/012022, (2007)

[9] Dongarra, J., Beckman, P. et al.: “The International Exascale Software Roadmap," Volume 25, Number 1, International Journal of High Performance Computer Applications, ISSN 1094-3420 (2011)

Article in a conference proceedings:

[10] Li, J., Keng Liao, W., Choudhary, A., Ross, R., Thakur, R., Gropp, W., Latham, R., Siegel, A., Gallagher, B., Zingale, M.: Parallel netcdf: A high-performance scientific i/o interface. SC03, 0:39 (2003)

[11] Bosilca, G., Herault, T., Rezmerita, A., Dongarra, J.: On Scalability for MPI Runtime Systems, IEEE Cluster 2011 International Conference (Cluster'11), Austin, TX, USA (2011)

[12] Balaji, P., Buntinas, D., Goodell, D., Gropp, W., Kumar, S., Lusk, E., Thakur, R., Traff, J.L.: MPI on a Million Processors, 16th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface. pp. 20-30. Springer-Verlag, Berlin, Heidelberg (2009)

[13] Fagg, G. E., Dongarra, J.:FT-MPI: Fault tolerant MPI, supporting dynamic applications in a dynamic world. 7th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface, pp. 346–353. Lecture Notes in Computer Science 1908, Springer-Verlag, Berlin, Heidelberg (2000)