NEMO performance optimisation on NEC SX-Aurora TSUBASA

E. Maisonnave

WN/CMGC/21/37

Abstract

In this note, we describe the conformance of our NEMO ocean model to the vector NEC SX-Aurora TSUBASA platform. A light porting effort was necessary to achieve good vector performance and deeper code modifications would certainly make possible a speed up of portions of the vector computations. The main limitation, but less sensible at high resolution, is the constraint on inner loop length (i.e. number of grid points in longitude) that reduces the possibility of 2D spatial decomposition of the grid in MPI subdomain, and the corresponding increase in useless land grid point computations. At the opposite, with a limited amount of resources, the better efficiency of our code on the vector platform is obvious. This superiority, where parallel efficiency is the best, strengthens the assumption that the energy consumption of our code is minimised there. In addition, a mature technology and the modularity given by the coupled configuration of the code, helped to fully exploit the node computing capacity (vector engine for computing + x86 host for disk access), avoiding power waste on the host processor. The overall behaviour of our code on this NEC machine should promote larger tests on production platforms by increasing scalability at intermediate resolution or experimenting larger configurations (ORCA36).

Table of Contents

An SX-Aurora TSUBASA A300-2[1] machine is opened to the CERFACS teams in order to port and optimise the laboratory codes on a vector machine recently chosen by computing centre of our community (Deutscher Wetterdienst, JAMSTEC) . We choose to check the NEMO ocean model [1] vectorisation capacity, remembering the good performance of this code on previous vector architectures [2]. The price of vector processors, more specialised than scalar machines, has diverted most of our computing centres from selecting such architectures, but the fragmentation of the supercomputing market and the end of multi-purpose scalar processor era could finally level off these costs [3].

Despite a continuous occupancy of the successive vector machines by our ocean model [4], the emergence of scalar supercomputers influenced the recent optimisation strategies and shaped the overarching coding rules of the official releases. In particular, one could cite many attempts to limit the communication footprint and maximise scaling, e.g. [5], include OpenMP/OpenACC parallelism [6], exploit single precision processors [7]. The evidence of the CMOS scaling slowdown and the limitation of specialised processors [8] invites to cautiously consider expensive adaptations of the code to possibly unfitted or short-lived technologies and, at the opposite, to better capitalize on software improvements for machines able to easily deliver a good sustained performance.

As already experienced on Intel KNL platform, the SX-Aurora TSUBASA node also includes an x86 host processor. To fully take benefit of this kind of mixed architecture, we must test that our code is modular enough to be adapted.

In this study, we propose to have a better view on the ratio development/performance needed to port NEMO (ocean only, 4.0.4 release) on a vector supercomputer. In chapter 1, we describe how to achieve the best performance on one vector engine. In chapter 2, we extend our study to a larger platform and we finish, in chapter 3, by wondering if the modularity brought by a coupled configuration of our code is able to exploit the extra computing capacity of the machine (hybrid mode).

1- Single VE performance

1.1- Porting

A comprehensive compiling tool set (C, FORTRAN) and necessary libraries (MPI, netCDF) are provided by NEC HPC Europe for use on both Vector Engine (VE) and Intel host. The two compiling are processed from the host, from separated user defined environments. The VE compiler, with appropriate options[2], deliver very comprehensive information on vectorisation

1 including one 10B (first generation) vector engine (VE), 8 cores@ 1.4 Ghz, 48GB of memory, 1.2 TB/s HBM2 memory bandwidth and 2.1 Tflop/s peak performance

2 Compiler options: `-fdiag-vector=3 -report-all` and environment variable `export NMPI_PROGINF=yes`

for each routine and a clear picture of the bottleneck (profiler[3]). This profiling shows more precise information than our internal profiling tool, slightly neglected during the last model developments.

The NEMO model is compiled to be launched on VE only (native compiling). The offload functionality (the program runs on the host, and specific instructions are offloaded to be executed on VE) is considered less appropriate in our case.

In addition to vectorisation listing and profiling options, equivalences to the standard Intel options are prescribed[4]. The `-O2` is the best optimisation option we tested.

To achieve all steps of the executable production in the same NEC VE environment, we noticed that the FORTRAN compilers (`mpinfort`) is not the only specific tool that must be used, but also static library builder (`nar`) et C pre-processor (`ncc -E`).

Two compiling or runtime errors pop up :

- in `nemogcm.F90`, the instruction
  `CALL ctl_opn(numnul, '/dev/null', 'REPLACE', 'FORMATTED', 'SEQUENTIAL', -1, -1, .FALSE. )`
  stops the simulation because the FORTRAN norm for `'REPLACE'` option implies to remove the file before being rewritten, and it is impossible to remove `/dev/null`.

- in `stpctl.F90`, the `ISNAN` function is not identified by the compiler and may be replaced by `ieee_is_nan`

A ticket[5] is open in the NEMO website to facilitate the compiling on NEC (and possibly other) machines to the future users of the release.

The whole porting phase was achieved in a couple of minutes, which validates the capacity of the NEC compiler to handle our code.

1.2- Vectorisation

The second phase of the porting starts with checking the level of vectorisation of the code. This can be done:

- for the whole code, by checking the standard output when the `NMPI_PROGINF` environment variable is set. A good vectorisation is reached when the average vector length (AVL) is close to the maximum possible (256) and when most of the simulation time is spent in vectorised sections.

3 Compiler and loader option: `-ftrace`
4 `-fdefault-integer=4 -fdefault-real=8 -fargument-noalias`
5 #2611

- for every routine, by looking at the diagnostic files[6] produced with the `-report-all` compiling option. The vectorisation is detailed for each line of the code, and can be optimised through directives[7].

The BENCH configuration [9] is chosen from the officially maintained ones. This configuration highly facilitates its handling for non-experts (no input files) but is totally similar (from computational or physics point of view) to the global configurations widely used in our community. Sea ice and biogeochemistry are excluded from this study. In this chapter, BENCH computations are performed on the ORCA1 global grid, defined on 362x332x75 points.

A precise timing of the time loop (excluding initialisation and finalisation phases) can be produced by NEMO namelist option[8] . The standard unit (simulated year per day, [10]) facilitates comparisons with past, present and future platforms.

The out-of-the box performance of our BENCH configuration on one NEC VE is comparable to the recent AMD Rome tests made on Météo-France supercomputer[9].

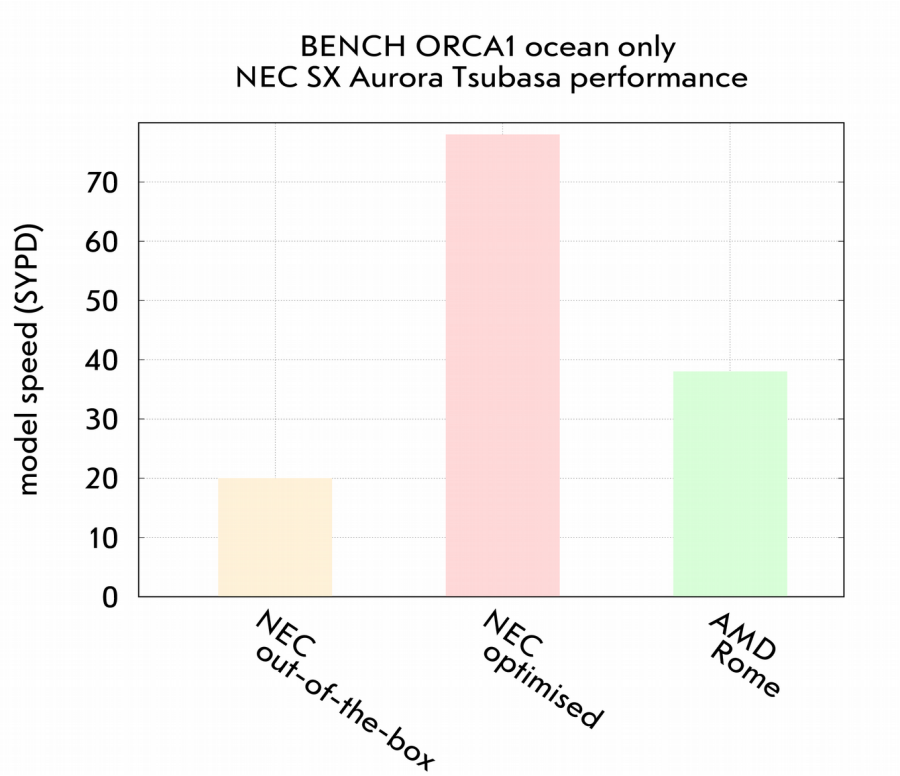

Figure 1: ORCA1 model speed comparison between NEC SX-Aurora TSUBASA (one VE) and AMD Rome (one 128-core node)

The AVL can be easily increased by changing the X/Y sub-domain decomposition, which defines the MPI partitioning of the total grid. We force this decomposition to (1,8), instead of the original (4,2) automatic choice, which extends the index count of most of the innermost loops of the NEMO code to the global dimension of the grid (360, excluding halos). The AVL goes from 82 to 174.

6 Files with `.L` extension

7 `!NEC$ directive-name`

8 `ln_timing = .true.`

9 `https://www.top500.org/system/179853/`

A finer analysis of the .L vectorisation reports suggests several modifications or simple vector directive forcing:

- removing `IF` condition involving character chain from the vectorised loop (`tra_nxt_vvl`)
- replacing unvectorised `SIGN` calls with `IF` conditions (`tra_adv_fct`, `bbl`, `ldf_slp`)
- force vectorisation where redirected index (`lbc_lnk`)

A key factor for gaining performance on such machine is the size of the innermost loop in computations. In NEMO, this size is bounded by the number of longitudinal grid points of each MPI decomposed subdomain, i.e.

```
X-dimension of the total grid /  # MPI processes in X
```

The number of grid point in longitude of our ORCA1 grid is 362, which forbids any other MPI decomposition along X but 1, since 256 grid points are needed for an optimum vectorisation.

The current NEMO version still includes the `key_vectopt_loop` pre-processing keyword, which allows to perform computations on the whole domain line, including halos. This operation makes contiguous the whole 2D data arrays and allows, if some conditions are fulfilled, the collapsing of the second loop and a vectorisation of the 2D array. This operation once had beneficial effects on overall performance. Unfortunately, in the current NEMO version, the `-floop-collapse` compiling option has an effect on 71 inexpensive loops only, mainly because memory accesses in the other loops are not contiguous. However, the performance gain is sensible (2%).

At the end of these operations, the AVL of the first 15 most expensive routines is equal to 180 (177 in total) and these routines spend more than 99% of the time in vector sector (more than 95% in total). The detailed analysis is given in Appendix 1. The NEMO ORCA1 configuration, with rough optimisations, is about two times faster than on one 128-core AMD Rome node (Figure 1).

2- Supercomputer performance

Recent tests on new AMD ROME based supercomputers proved that, in addition to good single node results, an efficient network is also necessary to performance at massively parallel scale.

Due to its low number of grid points, the parallelisation of the ORCA1 grid (362x332 horizontal grid points) cannot fully stress the supercomputer inter node communications. Benoît Lodej proposes to increase our problem size to the ORCA025 (1442x1207) and ORCA12 (4322x3147)

dimensions on a bigger machine located in Fuchū (NEC Japan). The machine VE card belongs to the second generation (20B type).

The result of the machine upgrade is the performance increase on a single VE (8 cores) with ORCA1 resolution: from 78 to 125 simulated years per day (SYPD). This further digs the performance gap between a single VE and a single AMD node (39 SYPD).

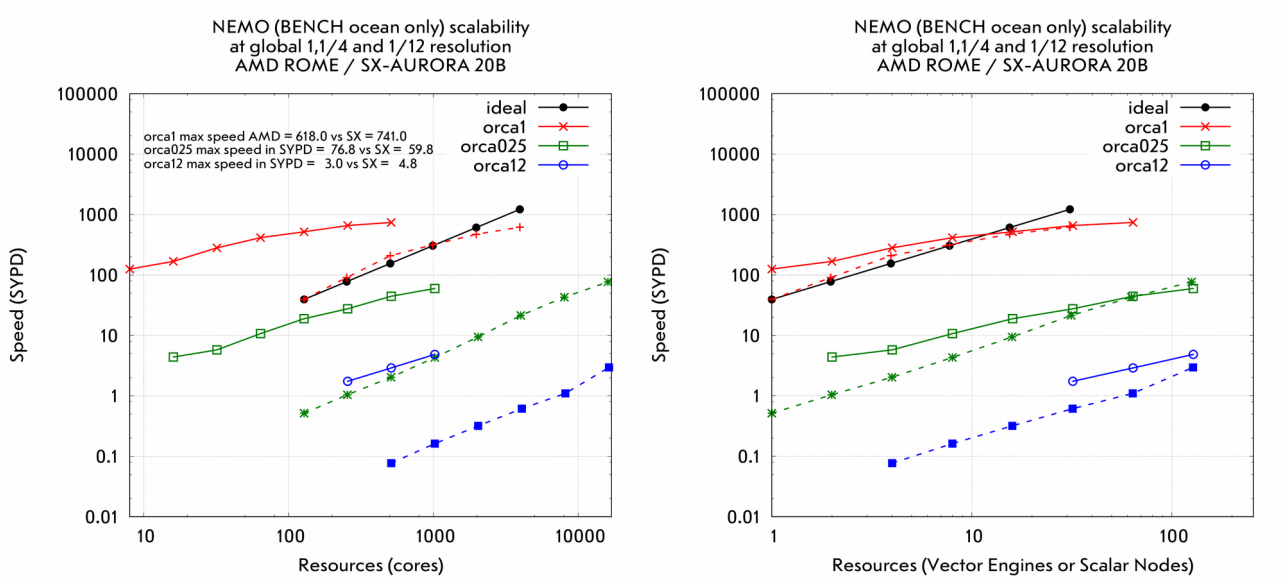

Figure 2a & b: NEMO (several horizontal resolutions) performance (speed in SYPD) on AMD ROME (dashed lines) and NEC SX-Aurora TSUBASA 20B (solid line) based machines, as a function of the number of scalar or vector cores (left) or number of Vector Engines/Scalar Nodes (right)

On Figure 2b, we see that this large advantage weakens when the VE/node number increases. For ORCA1 case, this clearly comes from the large overhead required by larger halos (more than two times bigger than the inner domain on more than 32 VE). For the larger resolutions, the weakening is also observed, even though the halo extra weight is less obvious. One possible explanation is the capacity of the network to deal with MPI communication of the NEMO halo exchanges. Due to the limit in time dedicated to this exploratory work, it was not possible to tune more accurately the network parameters that would increase the performance. Bandwidth could also be increased with the network interface controller upgrade (to PCIe Gen 4) coming with the next processor generation.

The AMD/NEC comparison shown in Figure 2 is led with the BENCH configuration that does not include any land grid point. In production mode, the automatic removal of land point only subdomain reduces with the resource number needed when the MPI parallelism increases. The reduction can reach 1/3 on global grids when the subdomain decomposition is fine enough. As seen in Chapter 1.2, the vector length restriction on vector machines limits the subdomain decomposition along longitude, the subdomain areas are longer and the land-only subdomain removal is much less efficient. At scalability limit (10x10 subdomain size), about 1/3 of the subdomains can be switched off on scalar machines, and the corresponding amount of resources saved. This ratio is probably much smaller on vector machine. Consequently, a correction to our numbers (obtain with an ocean only grid) must be applied at scalability limit to really compare vector and scalar performance for configuration including realistic bathymetries.

3- Hybrid mode

In addition to one VE, the NEC SX-Aurora TSUBASA node includes a x86 host processor. We propose to exploit the MPI Infiniband interconnection between host and VE to migrate a crucial part of our model to the host: the output of model result in netCDF format files.

We quickly set up an OASIS [11] based coupled system, by adding a small sequential executable. This process, playing the role of a simplified IO server (IS), only receives via MPI an OASIS coupling field sent by NEMO, . We use the "OUTPUT" OASIS functionality to write this variable on disk, in a configuration similar to the one tested in [12].

We compare two configurations:

- A- in native execution mode, the IS is compiled for a VE execution and launched on one VE core
- B- in hybrid execution mode, the IS is compiled (GNU) for the host and launched on the host

This B-mode imposes that all the necessary libraries (netCDF + OASIS) were both compiled for VE and host. Compiling and launching are easy and the test case can be set up in a few minutes.

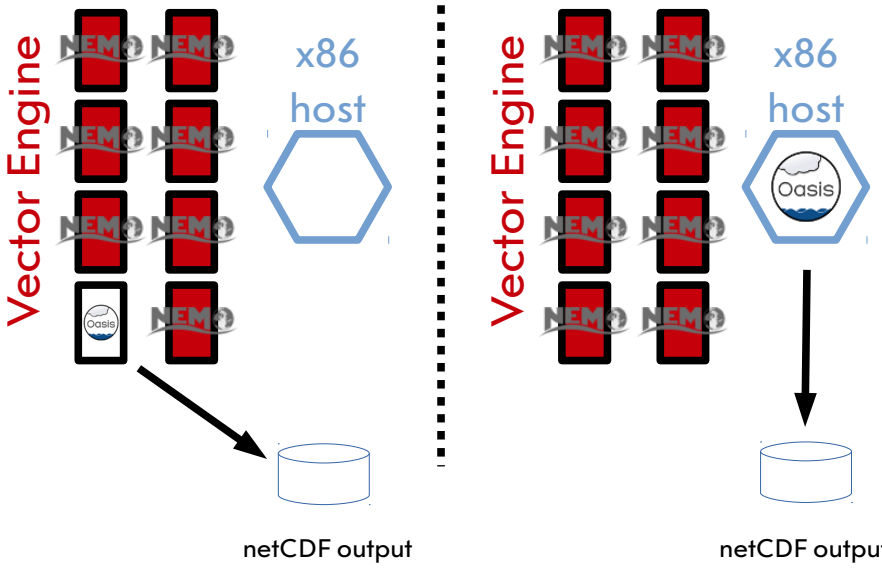

Figure 3: Two execution modes of a NEMO/OASIS output server coupled system on NEC SX-Aurora node, native (left) and hybrid (right)

In native mode, a comprehensive part of the IS execution is spent in output, as can be seen in the Figure 4 ("on Vector" line), produced by the OASIS internal diagnostic tool [13], looking at the red boxes (OASIS "OUT" events)[10]. The amount of output data is not big enough to slow

10 The ENDF event (light green box), occurring during the initialisation period, is excluded from the analysis, since it does not significantly influence the simulation speed in production mode

down the coupled system, as it can be deduced from the visible orange boxes (MPI wait periods), which indicates that the IS is waiting the NEMO coupling field. The simulation speed (67 SYPD) is only bounded by the time needed by the 7 VE cores attributed to NEMO to complete its computations.

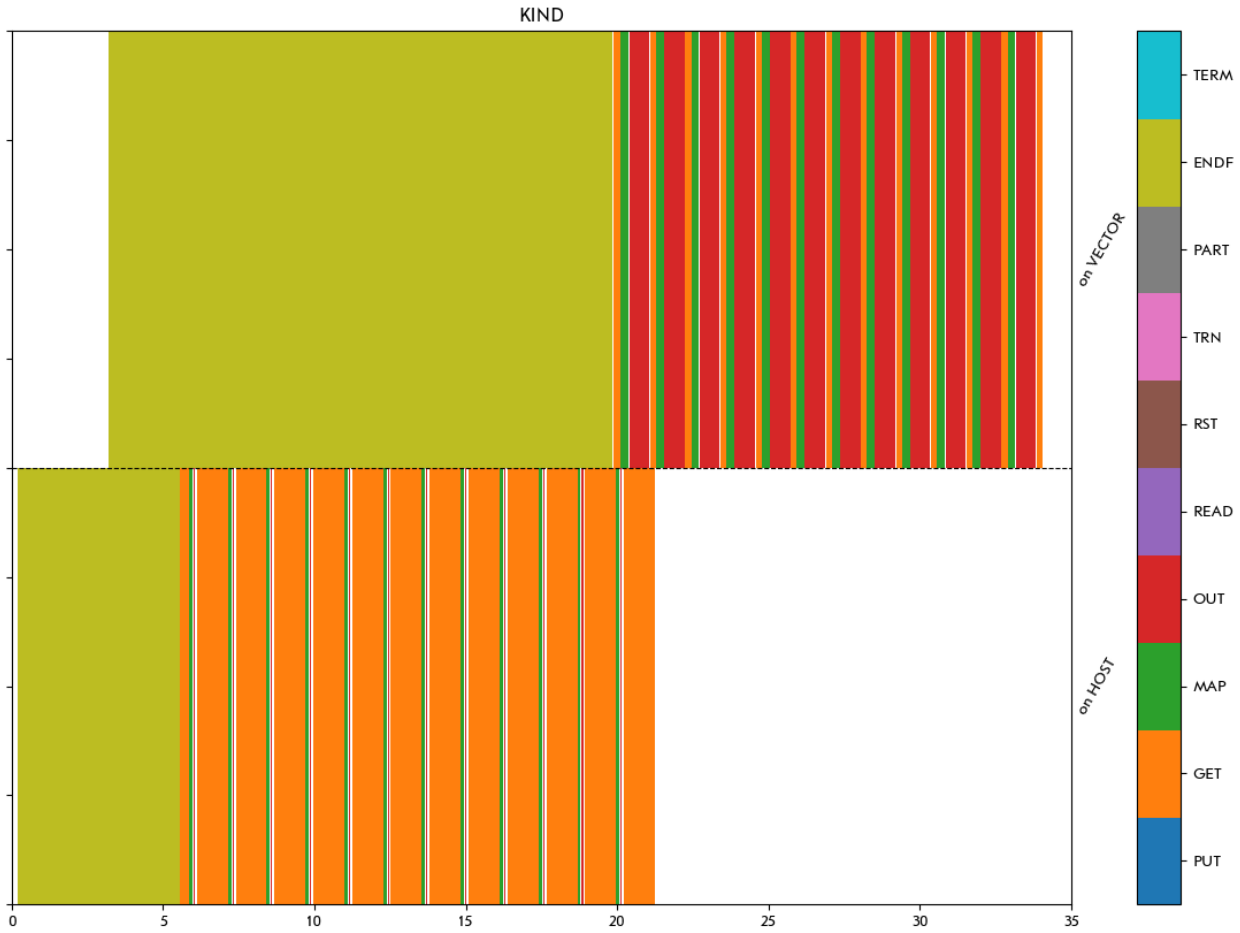

Figure 4: Comparison of two OASIS event timeline (seconds) from two different execution modes (vector native and hybrid) of the IS coupled component. Timeline of NEMO processes are not represented. In vector mode, 7 NEMO processes + 1 IS process are mapped on VE cores. In hybrid mode, 8 NEMO processes are mapped on VE cores + 1 IS process mapped on host. One 2D field output every 8h

In hybrid mode ("on HOST" line), orange is the dominant colour, which means that the IS is quickly performing the output operations and is spending most of the time waiting the MPI message from NEMO. However, despite the fact that NEMO computations are spread on 8 VE cores (instead of 7 in native mode), the NEMO model is slower and the overall coupled system only run at 62 SYPD. The origin of this difference can be identified in the NEMO halo exchange procedure, which suggests that the MPI communications are influenced by the change of network (VE only → VE + x86 host).

In consequence of these two effects (hybrid mode has slower MPI communication but speeds up disk writing), the best configuration will depend on the amount of data written on disk. On Figure 5, the same timelines plotted for a 4 times more frequent disk writing frequency shows better hybrid performance. In this configuration, IS is the slower coupled component in native mode (no more orange box), which output slows down the whole simulation.

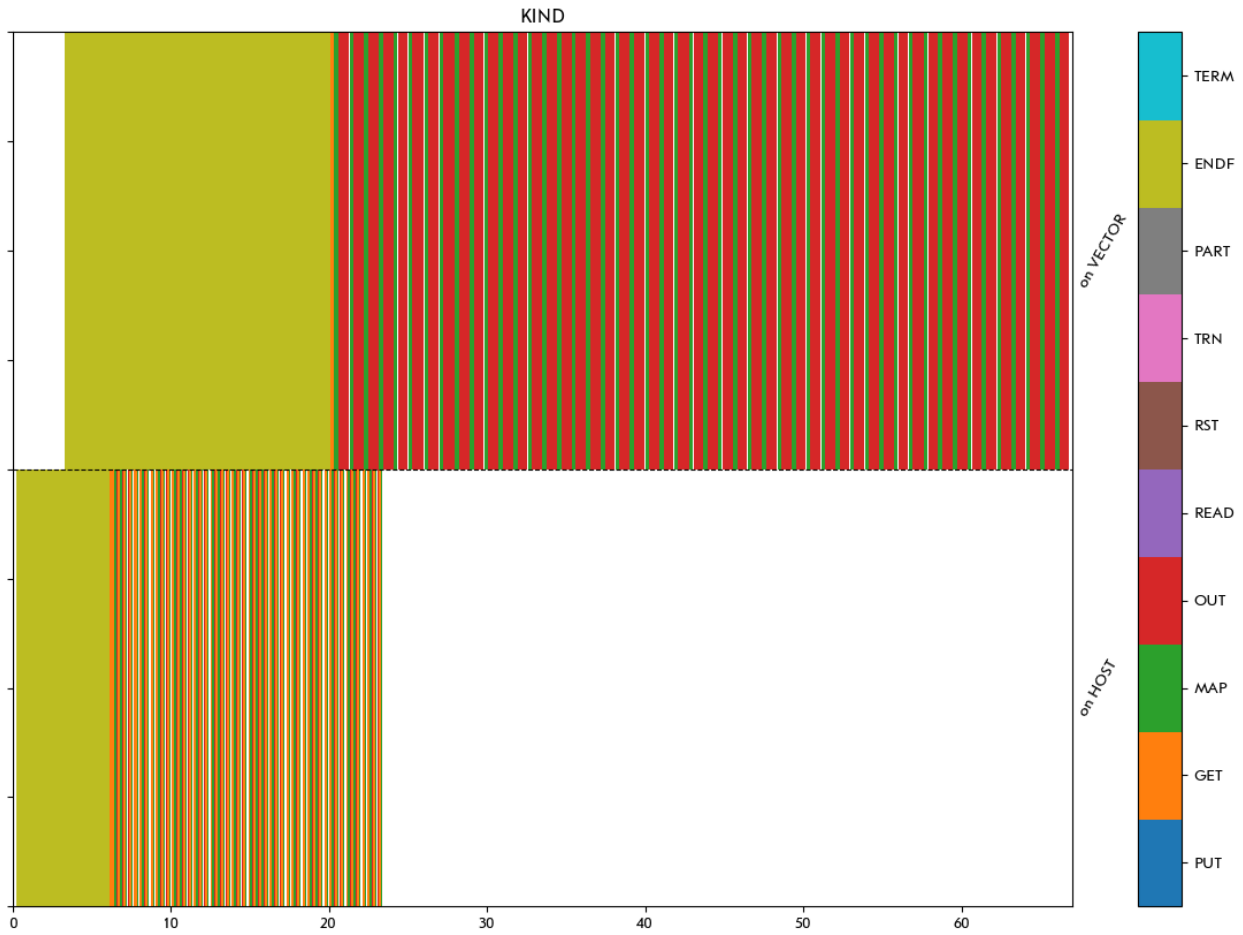

Figure 5: same than figure 3, but higher output frequency (2h)

4- Perspectives

We can clearly conclude from this study that there is a good conformance of our NEMO ocean model to the vector NEC SX-Aurora TSUBASA platform. A light porting effort was necessary to achieve good vector performance and deeper code modifications would certainly make possible a speed up of portions of the vector computations. The main limitation, but less sensible at high resolution, is the constraint on inner loop length (i.e. number of grid points in longitude) that reduces the possibility of 2D spatial decomposition of the grid in MPI subdomain, and the corresponding increase in useless land grid point computations. At the opposite, with a limited amount of resources, the better efficiency of our code on the vector platform is obvious.

The superiority of vector processing at low resource allocation, where parallel efficiency is the best, strengthens the assumption that the energy consumption of our code is minimised there. In addition, a mature technology and the modularity given by the coupled configuration of the code, helped to fully exploit the node computing capacity (vector engine for computing + x86 host for disk access), avoiding power waste on the host processor. A direct measurement of the total power consumption of our code was performed by Benoît Lodej on 8 nodes (64 VE). He shows a 2kW per node (200W per VE) consumption. The same measurement would

be necessary to definitely compare consumption on several machines, but AMD vendor data for the EPYC 7742 nodes installed at Météo-France are higher (450 W).

The overall behaviour of our code on the NEC machine should promote larger tests on production platforms (like the next Earth Simulator, 5,472 vector engines) by increasing scalability at intermediate resolution or experimenting larger configurations (ORCA36).

Acknowledgement: The porting issues could not have been solved without the help of Benoît Lodej and Laurent Gatineau from NEC HPC Europe, neither optimisations and measurements on the HPC cluster. Thanks to Nicolas Monnier for including this study in the CERFACS' codes porting action and Gurvan Madec (LOCEAN) for sharing his NEMO vectorisation strategies. The author wishes to acknowledge Thomas Williams and Colin Kelley for the development of the Gnuplot program, which analysis and graphics are displayed in this report, in addition to graphics from Matplotlib, a Sponsored Project of NumFOCUS, a 501(c)(3) non profit charity in the United States. This project did not received funding from the European Union's Horizon 2020 research and innovation programme. "Least Bee that brew / A Honey's Weight / Content Her smallest fraction help / The Amber Quantity "

References

[1] Madec, G. & NEMO System Team, 2019: "NEMO ocean engine", *Scientific Notes of Climate Modelling Center (27)* – ISSN 1288-1619, Institut Pierre-Simon Laplace (IPSL)
[2] Masson, S., Foujols, M.-A. , Klein, P., Madec, G., Hua, L., Lévy, M., Sasaki, H., Takahashi, K., & Svensson, F., 2008: OPA9 French Experiments on the Earth Simulator and Teraflop Workbench Tunings , proceedings of *High Performance Computing on Vector Systems 2007*
[3] Thompson, N., & Spanuth, S. , 2018: The decline of computers as a general purpose technology: why deep learning and the end of Moore's Law are fragmenting computing. Available at SSRN 3287769
[4] Oouchi, K., Tomita, H., Iga, S., Miura, H., Noda, A.T., Yamada, Y., Kodama, C., Hara, M., Seiki, T., Nakano, M., Chen, Y.-W., Miyakawa, T., Yashiro, H., Ikeda, M., Takigawa, M., Matsui, H., Doi, T., Maisonnave, E., Tatebe, H., Suzuki, T., Komuro, Y., Arakawa, T., Inoue, T., Fukutomi, Y., & Taniguchi H., 2015: Study for Seamless Prediction of Weather and Climate Using Atmosphere-Ocean Coupled Global Cloud-System Resolving Model , Annual Report of the Earth Simulator Center, Yokohama, Japan
[5] Maisonnave, E. & Masson, S., 2019: NEMO 4.0 performance: how to identify and reduce unnecessary communications , Technical Report, **TR/CMGC/19/19**, CECI, UMR CERFACS / CNRS No5318, France
[6] Porter, A. R., Appleyard, J., Ashworth, M., Ford, R. W., Holt, J., Liu, H., & Riley, G. D., 2018: Portable multi- and many-core performance for finite-difference or finite-element codes – application to the free-surface component of NEMO (NEMOLite2D 1.0), Geosci. Model Dev., **11**, 3447–3464, https://doi.org/10.5194/gmd-11-3447-2018, 2018.
[7] Tintó Prims, O., Acosta, M. C., Moore, A. M., Castrillo, M., Serradell, K., Cortés, A., & Doblas-Reyes, F. J.: How to use mixed precision in ocean models: exploring a potential reduction of numerical precision in NEMO 4.0 and ROMS 3.6, Geosci. Model Dev., **12**, 3135–3148, https://doi.org/10.5194/gmd-12-3135-2019, 2019.
[8] Fuchs A. & Wentzlaff, D., 2019: The Accelerator Wall: Limits of Chip Specialization, 2019, IEEE International Symposium on High Performance Computer Architecture (HPCA), Washington, DC, USA, 2019, pp. 1-14, doi: 10.1109/HPCA.2019.00023
[9] Irrmann, G., Masson, S., Maisonnave, E., Guibert, D., Douriez, L. & Raffin, E., MPI communication optimizations achieved between NEMO v3.6 and v4.0, to be published
[10] Balaji, V., Maisonnave, E., Zadeh, N., Lawrence, B. N., Biercamp, J., Fladrich, U., Aloisio, G., Benson, R., Caubel, A., Durachta, J., Foujols, M.-A., Lister, G., Mocavero, S., Underwood, S., & Wright, G., 2017: CPMIP: measurements of real computational performance of Earth system models in CMIP6, Geosci. Model Dev., **10**, 19–34, https://doi.org/10.5194/gmd-10-19-2017
[11] Craig, A., Valcke, S. & Coquart, L., 2017: Development and performance of a new version of the OASIS coupler, OASIS3-MCT_3.0, Geosci. Model Dev., **10**, pp 3297-3308, https://doi.org/10.5194/gmd-10-3297-2017
[12] Maisonnave, E., 2013: PoCO, Post-processing coupled with OASIS,Technical Report, **TR/CMGC/13/70**, SUC au CERFACS, URA CERFACS/CNRS No1875, France
[13] Maisonnave, E., Coquart, L., & Piacentini, A., 2020: A better diagnostic of the load imbalance in OASIS based coupled systems, Technical Report, **TR/CMGC/20/176**, CECI, UMR CERFACS/CNRS No5318, France

Appendix 1: Timing of NEMO ORCA1 BENCH on one VE of NEC SX-Aurora TSUBASA for the 15 most expensive routines

```
FREQUENCY  EXCLUSIVE       AVER.TIME     MOPS   MFLOPS  V.OP  AVER.    VECTOR L1CACHE CPU PORT VLD
LLC PROC.NAME
           TIME[sec](  % )    [msec]                    RATIO V.LEN      TIME    MISS     CONF HIT
E.%

      100     1.219(  9.4)    12.189  67032.2  45167.0  99.11 180.2     1.209   0.007    0.000
34.18 TRALDF_ISO::TRA_LDF_ISO
      100     1.081(  8.3)    10.807  61299.5  40666.9  99.26 180.7     1.074   0.005    0.000
42.47 LDFSLP::LDF_SLP
      100     1.009(  7.7)    10.091  80619.1  59103.1  99.11 181.7     0.994   0.003    0.000
30.43 ZDFIWM::ZDF_IWM
     1309     0.657(  5.0)     0.502  57429.6  28085.9  98.53 180.8     0.650   0.006    0.000
27.75 DOMVVL::DOM_VVL_INTERPOL
      100     0.626(  4.8)     6.261  44568.8  19686.2  98.06 180.7     0.576   0.049    0.000
43.62 DYNSPG_TS::DYN_SPG_TS
    11718     0.550(  4.2)     0.047    630.6      0.0  92.39  41.5     0.462   0.064    0.000
62.54 LIB_MPP::MPPRECV
      100     0.511(  3.9)     5.112  81554.2  60014.9  99.19 180.1     0.509   0.002    0.000
25.21 ZDFTKE::TKE_TKE
      100     0.498(  3.8)     4.981  71933.6  52435.7  99.23 180.1     0.496   0.002    0.000
26.36 DYNZDF::DYN_ZDF
      100     0.473(  3.6)     4.731  46205.8  20905.4  97.98 181.0     0.468   0.004    0.000
32.00 DYNNXT::DYN_NXT
      100     0.426(  3.3)     4.257  76457.4  56465.3  99.26 180.0     0.424   0.001    0.000
19.20 TRAZDF::TRA_ZDF_IMP
      100     0.384(  2.9)     3.840  84525.7  57843.3  99.24 181.0     0.382   0.002    0.000
33.47 ZDFDDM::ZDF_DDM
      100     0.332(  2.6)     3.322  51912.0  28951.7  98.64 180.0     0.328   0.004    0.000
10.65 TRAADV_CEN::TRA_ADV_CEN
      200     0.320(  2.5)     1.602 131369.2 116996.7  99.70 181.0     0.320   0.001    0.000
2.26 EOSBN2::RAB_3D
      200     0.306(  2.3)     1.530  66318.0  46461.2  99.21 180.0     0.305   0.001    0.000
26.16 TRAADV_FCT::INTERP_4TH_CPT
       99     0.305(  2.3)     3.084  85621.7  59290.4  99.18 181.8     0.304   0.001    0.000
42.42 TRANXT::TRA_NXT_VVL
```