Stock-take of status of implementation of NEMO HPC strategy

NEMO HPC Working Group

Version 0.3 15 January 2019

Four main issues and three additional issues were identified in the HPC chapter of the NEMO Development strategy. Progress and plans for these are outlined below. In each case we cover:

- who is working on the issue and which European projects will contribute to it
- what progress has been made
- what progress is expected in 2019
- highlight any problems that we do not know how to solve

The final section describes work to establish benchmarks and information on NEMO's performance on various machines.

1. Inter-node communications

1.1 Who is contributing

Eric Maisonnave (Cerfacs), Seb Masson (CNRS) and Gurvan Madec (CNRS). Eric's time is paid by ESIWACE WP2 and CNRS funds.

Francesca Mele (CMCC), Silvia Mocavero (CMCC). [Need to note funding sources]

Louis Douriez(Bull), David Guibert(Bull) and Erwan Raffin(Bull). Louis and David are really looking and working on the code; Erwan is more on the management side (contact with us, contact with ESIWACE and ESIWACE2). They are paid by ESIWACE and ESIWACE2. In ESCAPE-2 Bull is involved in installation and performance testing of benchmark dwarfs and models (of which one is a dycore of NEMO).

Miguel Castrillo (BSC), Mario Acosta (BSC) and Stella Paronuzzi (BSC) will work on this task during the next year from funding from IS-ENES3.

1.2 What progress has been made

Gurvan removed many communications in the V4.0 (unfortunately we have no number). He rewrote the mpp interface to simplify it and fully integrate the "multi communications". He either used multiple to gather and reduce the number of communications or rewrote parts of the code to simply suppress communications.

mppini has been revisited (merge of mppini and mppini2) which simplifies the code and allows a better understanding and further developments (toward larger halo for example).

Eric, Seb and Rachid have done several things since version 4.0 was moved into the trunk:

1) added a report of which routines is doing how many call to lbc_lnk or global communications. So we know where we have to do something, what is acceptable or not.

2) added a timing specific to the communications: add a report of the % of time spent in communications (sending or waiting) and the computation.

3) updated the timing routine to solved problems when a routine is called by 2 different routines

4) implemented of the BENCH test case [this should go in the final section ?]

5) improved the north pole folding that was identified as the main bottle neck for scalability

6) suppressded of global communications (except in the diagnostics, that should be in fact done through xios)

7) suppression / gather in communications in oce but mainly in si3

8) modified mppin to automatically provide the best domain decomposition, including removal of land sub-domains.

9) we plan to compare BENCH with ORCA configurations by the end of the year [this should go in the final section ?]

All of the above work is in a branch ready to be merged by the end of 2018.

CMCC has made a deep analysis of all of the global communications; several global communications have been identified and safely removed. These optimizations will be included in the NEMO 4.0 official release which will be released in early 2019

Bull has been exploring various modifications of the communications (in consultation with CNRS). There is a list of ideas that could give a potential benefit that has been presented to the HPCWG by David.

1.3 What progress is expected in 2019

CNRS aims to pull successful ideas from Bull into the trunk. The other ideas will have been tried and documented.

Subject to Eric's time being available, Seb and Eric plan to

- explore larger haloes to reduce communications, for example in some critical parts of the code

- do diagnostics with xios2 to suppress blocking global communications

- remove communications needed only to do outputs with the help of xios or by suppressing periodicity columns and band in input/output files

There may be a problem reading forcing files at high frequency on a large number of cores. If that is the case xios should be used for the input file in fldread.

CMCC will carry out some investigations regarding the exploitation of the neighbour collective communications available in MPI3 to reduce the internode communications. ECMWF will investigate GPI/GASPI and MPI3 features to facilitate overlapping of communications and computations as part of EPIGRAM-HS starting in Feb 2019. This work will inform on inter-node communication strategies.

BSC will carry out a set of profiling analyses with NEMO4 using BSC tools to provide feedback on it and to discuss about new improvements. ECMWF and BSC will profile coupled configurations.

1.4 Problems

None identified.

2. Shared memory parallelism

2.1 Who is contributing

Francesca Mele (CMCC), Silvia Mocavero (CMCC) with funding from EsiWACE.

Maff Glover (Met Office), Mike Bell (Met Office) with funding from IMMERSE.

Ioan Hadade (EPiGRAM-HS), Michael Lange (EuroEXA), and (limited) Kristian Mogensen(ECMWF) under the scalability programme at ECMWF will investigate options for overlapping communications and compute (GPI-2/GASPI; MPI+X).

2.2 What progress has been made

CMCC has made investigated coarse-grain parallelization applied to the advection kernel only. The results demonstrated that the intra-node parallel efficiency has improved by 5-11% compared to pure-MPI version, whilst the internode parallel efficiency is increased by about 5-7%. The fine-grain parallelization has been applied to the $1/16^{o}$ global configuration with an improvement of about 7%. Even though the shared memory parallelism exposes some performance improvements, these seem to be not sufficient to justify a radical modification of the programming approach.

The Met Office has adjusted the DO loops in three representative routines (tra_ldf_iso, zdf_tke and tke_tke) so that the calculations are performed over tiles, that is sub-domains of the MPI domains. The calculations within a tile have been written so that they are thread-safe. This introduces an explicit additional level of parallelism within the code that can be exploited at the science control level. OpenMP directives have been inserted in the science control (tra_ldf) routine. This approach results in a small number of OpenMP directives and small number of variables that need to be declared as private. [Need a meaningful comment on the effectiveness of the approach – Performance with 6 threads is 85% of that with 1 thread]

2.3 What progress is expected in 2019

The Met Office will make a branch of the tiling code available. They will also extend the tiling to a larger number of subroutines. The aim is to agree the style of coding to use and to implement a first tranche of code in the trunk by the end of 2019. There will likely be a prototype from ECMWF using GPI-2 or similar technology.

2.4 Problems

3. Single core performance

3.1 Who is contributing

Francesca Mele (CMCC), Silvia Mocavero (CMCC) with funding from IMMERSE. Maff Glover (Met Office) and Mike Bell (Met Office) also with funding from IMMERSE.

3.2 What progress has been made

CMCC have introduced SIMD directives in the routines with the lowest level of vectorization. The use of SIMD demonstrated good improvements (about 10%) and could be extended to other computationally intensive routines. Cache blocking techniques (through the automatic definition of the block size) have also been investigated. The improvement depends on the compiler version and the optimisation level. This kind of technique is not very promising for the NEMO code.

The Met Office has implemented tiling for a small number of subroutines (see section 2.2). The size of the tiles (in each dimension) is tuned using a simulated annealing technique. Results show that the run-time of tra_ldf_iso is reduced by more than a factor of 2 in the main configurations as they are presently run at the Met Office; it is converted from a memory bound routine to a computation bound routine. The run-time of vertical physics is reduced by about 25% [check].

3.3 What progress is expected in 2019

CMCC aim to introduce SIMD directives into the official NEMO trunk. They will also investigate other vectorisation techniques and their impact on the NEMO code readability and other techniques to improve the use of memory hierarchy.

The Met Office aim to start to implement tiling within NEMO (see 2.3).

3.4 Problems

4. Designing a flexible user-friendly code structure

4.1 Who is contributing

Rupert Ford (STFC) and Andy Porter (STFC) with funding from EuroEXA and ESIWACE II. A proposal has been submitted to NERC (UK) call that is solely focused on implementing full support for NEMO in PSyclone.

Italo Epicoco (CMCC) with funding from ESCAPE2 and EsiWACE2.

ECMWF with funding from ESCAPE-2 and EuroEXA.

The work on tiling described in sections 2 and 3 is also relevant to this issue because it arose as a question of how to introduce OpenMP and OpenACC directives into NEMO in a sustainable manner.

Work to re-design NEMO to enable two-level time-stepping schemes to be introduced by Gurvan Madec (CNRS) and Mike Bell (Met Office) is also somewhat relevant to this issue.

4.2 What progress has been made

STFC has an initial branch of PSyclone that uses fparser2 to parse NEMO source. The resulting syntax tree is then processed by PSyclone and converted into PSyclone's internal-representation (PSyIR). It is then possible to add OpenMP directives and re-generate the code. This is working for the traldf_iso and traadv_tvd routines. This branch of PSyclone is in the process of going through code review.

CMCC has made some preliminary investigations into the application of the Psyclone tool to the NEMO code in collaboration with STFC in the context of IS-ENES2.

4.3 What progress is expected in 2019

If the NERC proposal is funded then full-time work on adding NEMO support to PSyclone will commence in March (adding support for OpenACC, loop-blocking and OpenMP). If not, then whatever free effort there is in EuroEXA and ESIWACE will be used to continue the work, aiming initially at adding OpenACC to NEMO.

CMCC will investigate and evaluate the adoption of the DSL toolchain for the advection kernel and provide a benchmark dynamical core model for ESCAPE-2. This will facilitate prototyping and applications of a DSL toolchain. ECMWF will support these efforts as part of ESCAPE-2 towards GPU readiness of NEMO4.

4.4 Problems

It may not be possible to block some NEMO loops at as high a level as has been done manually (due to complex dependencies). This may result in a performance penalty compared to the manual implementation but this is still to be investigated.

5. Mixed precision calculations

5.1 Who is contributing

Oriol Tinto (BSC), Miguel Castrillo (BSC) & Mario Acosta (BSC)

Guidance and management from Peter Dueben (ECMWF) plus 1 FTE (ECMWF) with support from ESiWACE2.

5.2 What progress has been made

A method has been developed to identify which variables can use reduced precision without degrading the quality of the results. Given a test to assess the accuracy of the results, the method uses an emulator to automatically find a solution that minimizes the resources invested in precision without compromising the accuracy.

5.3 What progress is expected in 2019

To have a full implementation of NEMO in which most of the variables have been moved to single precision. There are ongoing discussions about whether different precision levels can be included. The variables will be selected using the methodology developed by the BSC with an accuracy test which is valid for the configurations that the implementation will support. ECMWF will hire a person with funding from ESIWACE-2 to support the creation of a sustainable mixed-precision version of NEMO4.

5.4 Problems

The best way to involve the community in the development and obtain the necessary feedback still has not been found. To proceed with the development ideally a consensus has to be built regarding how the accuracy of the results is assessed and how the implementation must be done.

6. Parallel execution of sea-ice
6.1 Who is contributing
6.2 What progress has been made
6.3 What progress is expected in 2019
6.4 Problems
7. Parallel execution of biology
7.1 Who is contributing
7.2 What progress has been made
7.3 What progress is expected in 2019
7.4 Problems