Attempt to enhance the open-mp scalability of the spectral transforms of ALADIN

stay report by

Nuno Lopes
(I.M., I.P.)

under the supervision of

Ryad El Khatib
(METEOFRANCE – CNRM/GMAP)

4 July to 29 July 2011

1. Introduction

Historically, one can see a positive trend regarding model complexity, number of processors, code parallelization and communications overhead, all influencing computational performance. In the framework of the execution of ALADIN's configuration 001, spectral transforms are an important time consuming part of it and subject to continuous code optimization.

In massively parallel systems, code optimization can be checked with scalability. Scalability is “a way of assessing the ability of an application to continue profiting with the use of an increasing number of computing resources” [1] and as that is a useful “criterion to determine the efficiency of parallelism” [1] and consequently of code optimization.

The task addressed during the stay was concerned with direct spectral transforms, in particular with attempting to improve the scalability of spectral transforms of ALADIN.

2. Code changes

In the spectral transforms, we worked only on the Legendre Direct Transforms which are calculated in *subroutine eltdir* [2]. Looking inside the code, in *eltdir_mod.F90*, there is a comment about the existence of a loop split in three distinct parts, with the first and third parts parallelized with OpenMP but not the second part, due to the existence of MPI communications inside *euvtvd_mod.F90*. The existence of this message passing serve the purpose of sending information about mean wind components u and v from a processor to all other processors.

In fact, the information about mean wind components is carried only in the first wave of the spectral model, wave 0, so the communication is done only by the processor who has this information. Since there is a loop for all wave numbers for transforming u,v in divergence and vorticity, with the condition that when we are in the first wave some MPI communication is done, the subroutine was rearranged so that now we split the *euvtvd module* in two modules. One module will be containing only the MPI communications and actually only acting when dealing with the first wavenumber. The other module will be carrying only the OpenMP parallelized part of the code so that we could define a bigger OpenMP cycle. In the new module with the MPI communications, a change was made in *MPL_SEND function*, defining it now with a “non blocking status”, so that the processor can continue to send information without waiting for the confirmation that the previous message sent as arrived it's destiny. For the new module with the code previously inside an OpenMP cycle, the instructions specific to the parallelization were removed as this code is now inside a bigger OpenMP loop, defined inside *eltdir module*.

So before the modifications, there were three OpenMP cycles within *eltdir_mod.F90*, two cycles directly inside the subroutine and a third cycle in *euvtvd module* which was called during execution. After the changes we have now two OpenMP cycles, reducing the resources needed to allocate OpenMP and theoretically beneficing efficiency. On the reverse side, two new files were created to accommodate the repositioning of the code although the programming inside is basically unchanged from the original one.

In figure 1 bellow, we show schematically the actual operational *eltdir module* and afterwards in figure 2, the changed *eltdir module*.

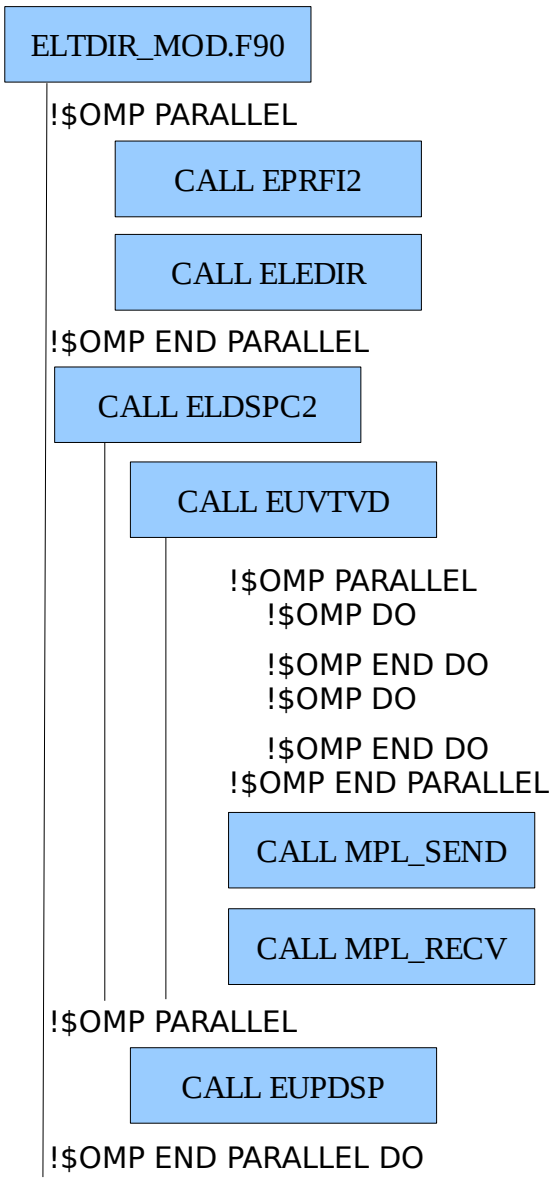

Figure 1 – Scheme of operational *eltdir module* .

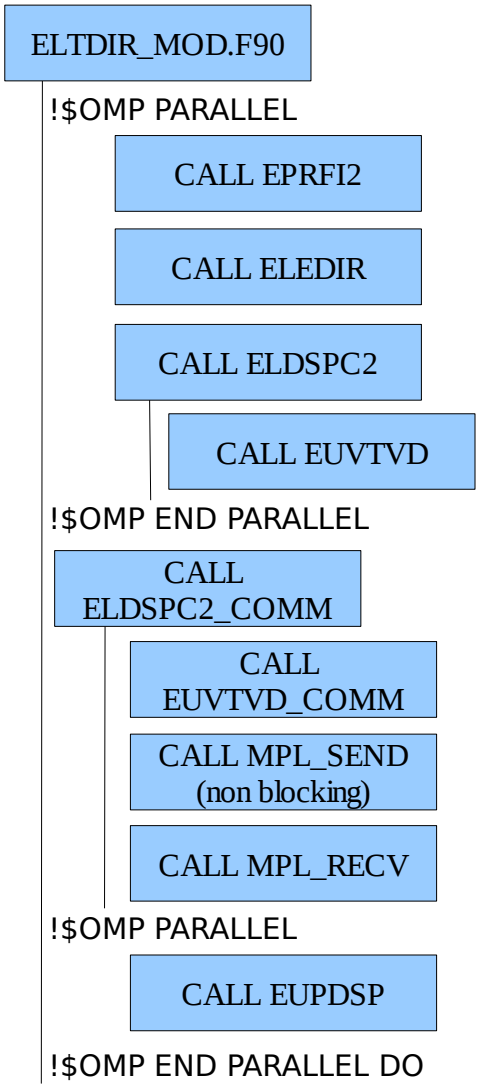

Figure 2 – Scheme of test *eltdir module*.

First tests were made on a local pc with serial and parallel integrations for ALADIN CY37T1. Spectral norms were verified and no differences between the changed pack and the original pack were found in any execution tested. Afterwards, next task was to test scalability on an high-perfomance computing machine, which was done on ECMWF's machine c1a.

In c1a machine, tests were made for AROME CY37T1 in a small domain of 130 x 100 points, evaluating the wall clock time needed for the execution of configuration 001. Those tests were submitted with mtool, a tool developed by Météo-France in order to prepare jobs to be executed in parts, separating the execution of full-pos configurations from the integration of the model.

The results obtained for varying the number of OpenMP threads with fixed MPI tasks, or reversely, for varying the number of MPI tasks with fixed OpenMP threads, showed, in general, small differences of time of only a few seconds between operational code and test code, which most probably resulted from performance issues of the machine c1a and not from a real impact of code changes. Some selected results are shown graphically, in terms of scalability of the model.

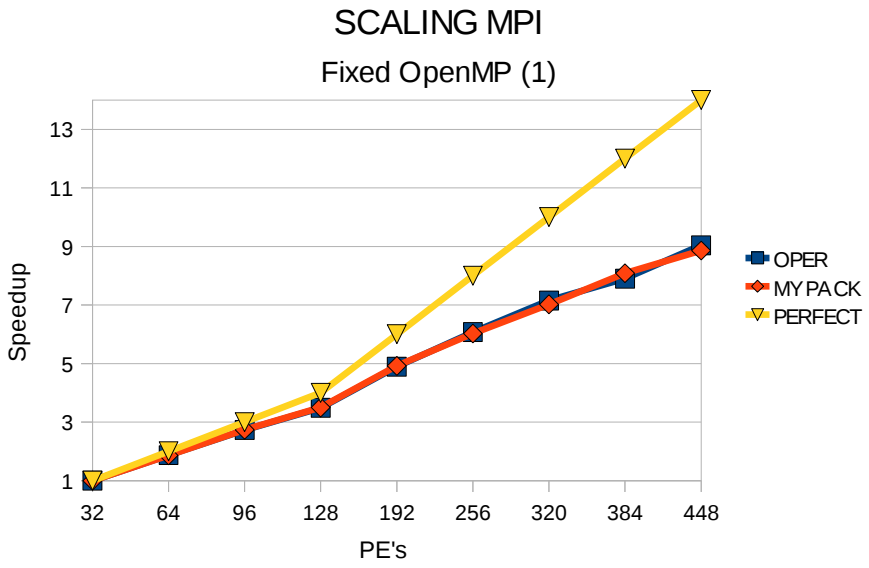

Figure 3 – MPI scalability for AROME.

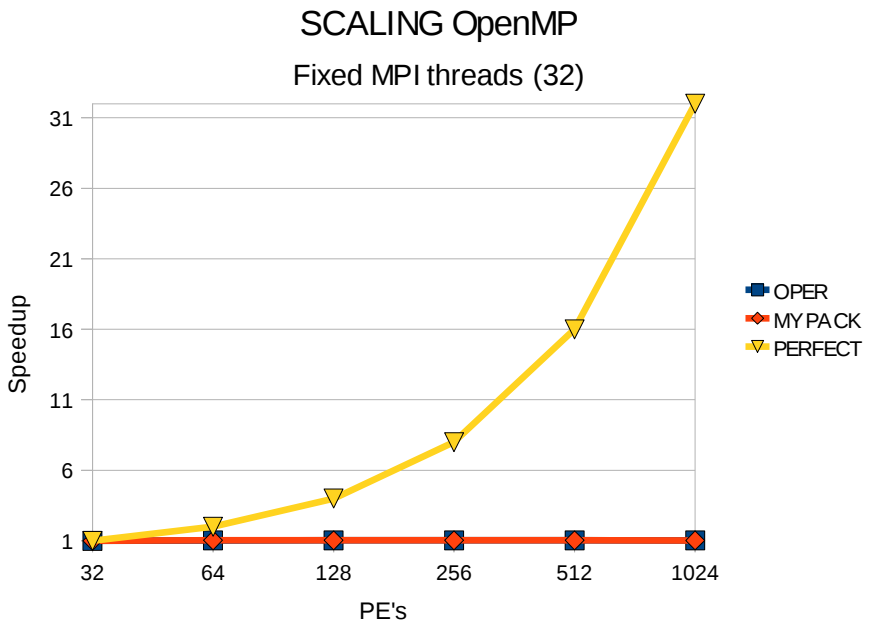

Figure 4 – OpenMP scalability for AROME.

3. Conclusion

No definitive conclusion can be taken from the experiences made. For the small AROME domain tested there seems to be a null impact of the changes. As so, more weight in the spectral transforms part of the model integration is required to better assess the changes impact. That could be accomplished by testing the same domain using an eulerian integration scheme or even simply using a bigger AROME domain.

References

[1] Eric Sevault, “Guide du routard d'un monde parallèele”. Ateliers de Modélisation de l'Atmosphère, February 2011.

[2] Karim Yessad, “Spectral transforms in the cycle 37t1 of ARPEGE/IFS”. ALALDIN internal documentation, May 2011. (http://www.cnrm.meteo.fr/gmapdoc/spip.php?article28&var_lang=en)