Agència
de Gestió d'Ajuts
Universitaris
i de Recerca

Memòria justificativa de recerca de les convocatòries BCC, BDH, BE, BP, CTP-AIRE, DEBEQ, FI, INEFC, i PIV

La memòria justificativa consta de les dues parts que venen a continuació:
1.- Dades bàsiques i resums
2.- Memòria del treball (informe científic)

Tots els camps són obligatoris

1.- Dades bàsiques i resums

Nom de la convocatòria

BP

Llegenda per a les convocatòries:

BCC	Convocatòria de beques per a joves membres de comunitats catalanes a l'exterior
BDH	Beques i ajuts postdoctorals del Programa DGR-Henkel KGaA
BE	Beques per a estades per a la recerca fora de Catalunya
BP	Convocatòria d'ajuts postdoctorals dins del programa Beatriu de Pinós
CTP-AIRE	Ajuts per accions de cooperació en el marc de la comunitat de treball dels Pirineus. Ajuts de mobilitat de personal investigador.
DEBEQ (Modalitat A3)	Beques de Cooperació Internacional i Desenvolupament
FI	Beques predoctorals per a la formació de personal investigador
INEFC	Beques predoctorals i de col·laboració, dins de l'àmbit de l'educació física i l'esport i les ciències aplicades a l'esport
PIV	Beques de recerca per a professors i investigadors visitants a Catalunya

Títol del projecte: ha de sintetitzar la temàtica científica del vostre document.
Porting and performance analysis of EC-Earth system on the BSC-MareNostrum Supercomputer

Dades de l'investigador o beneficiari

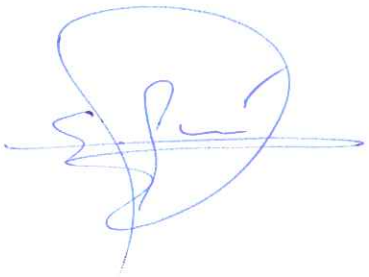
Nom	Cognoms
Maria	Val Martin

Correu electrònic
mvalmart@seas.harvard.edu

Dades del centre d'origen

Earth Sciences Department, Barcelona Supercomputing Center

Nom i cognoms i signatura
del/de la investigador/a



Vist i plau del/de la responsable de la
sol·licitud

Agència
de Gestió d'Ajuts
Universitaris
i de Recerca

Número d'expedient
SG053297

Paraules clau: cal que esmenteu cinc conceptes que defineixin el contingut de la vostra memòria.
climate modeling, high supercomputer, model perfomance, CMIP-5, IPCC-AR5

Data de presentació de la justificació
18-10-2010

Agència
de Gestió d'Ajuts
Universitaris
i de Recerca

Resum del projecte: cal adjuntar dos resums del document, l'un en anglès i l'altre en la llengua del document, on s'esmenti la durada de l'acció

Resum en la llengua del projecte (màxim 300 paraules)

Earth System Models (ESM) have been successfuly developed over the past few years, and are currently being used for simulating present-day climate, seasonal to interannual predictions and projections of climate change. The supercomputer performance plays an important role in climate modeling since one of the challenging issues for climate modellers is to efficiently and accurately couple earth system components on present-day computer architectures.

At the Barcelona Supercomputing Center (BSC), we work with the EC-Earth system model. The EC-Earth is an ESM, which currently consists of an atmosphere (IFS) and an ocean (NEMO) model that communicate with each other through the OASIS coupler. Additional modules (e.g. for chemistry and vegetation) are under development. The EC-Earth ESM has been ported successfully over different high performance computing platforms (e.g., IBM P6 AIX, CRAY XT-5, Intel-based Linux Clusters, SGI Altix) at different sites in Europe (e.g., KNMI, ICHEC, ECMWF).

The objective of the first phase of the project (6 months) was to identify and document the issues related with the portability and performance of EC-Earth on the MareNostrum supercomputer, a system based on IBM PowerPC 970MP processors and run under a Linux Suse distribution. During this 6-month period, EC-Earth was successfully ported to MareNostrum, and a compilation incompatibility of IFS in EC-Earth to the compiler XLF version 12.1 was detected. This incompatibility was solved by a two step compilation approach using the XLF version 10.1 and 12.1 compilers. In addition, the EC-Earth performance was analyzed with respect to scalability and trace analysis with the Paraver software. This analysis showed that EC-Earth performs fairly well in MareNostrum, and that the efficient number of IFS CPUs is 64. Running EC-Earth with a larger number of IFS CPUs (>128) is not feasible at the moment since some issues exists with the IFS-NEMO balance and MPI communications.

Resum en anglès (màxim 300 paraules)

Earth System Models (ESM) have been successfuly developed over the past few years, and are currently being used for simulating present-day climate, seasonal to interannual predictions and projections of climate change. The supercomputer performance plays an important role in climate modeling as one of the challenging issues for climate modellers is to efficiently and accurately couple earth system components on present-day computer architectures.

At the Barcelona Supercomputing Center (BSC), we work with the EC-Earth system model. The EC-Earth is an ESM, which currently consists of an atmosphere (IFS) and an ocean (NEMO) model that communicate with each other through the OASIS coupler. Additional modules (e.g. for chemistry and vegetation) are under development. The EC-Earth ESM has been ported successfully over different high performance computing platforms (e.g., IBM P6 AIX, CRAY XT-5, Intel-based Linux Clusters, SGI Altix) at different sites in Europe (e.g., KNMI, ICHEC, ECMWF).

Resum en anglès (màxim 300 paraules) – continuació -.

The main objective of the first phase of the project (6 months) was to identify and document the issues related with the portability and performance of EC-Earth on the MareNostrum supercomputer, a system based on IBM PowerPC 970MP processors and run under a Linux Suse distribution. During this 6-month period, EC-Earth was successfully ported to MareNostrum, and a compilation incompatibility of IFS in EC-Earth to the compiler XLF version 12.1 was detected. This incompatibility was solved by a two step compilation approach using the XLF version 10.1 and 12.1 compilers. In addition, the EC-Earth performance was analyzed with respect to scalability and trace analysis with the Paraver software. This analysis showed that EC-Earth performs fairly well in MareNostrum, and that the efficient number of IFS CPUs is 64. Running EC-Earth with a larger number of IFS CPUs (>128) is not feasible at the moment since some issues exists with the IFS-NEMO balance and MPI communications.

2.- Memòria del treball (informe científic sense limitació de paraules). Pot incloure altres fitxers de qualsevol mena, no més grans de 10 MB cadascun d'ells.

Porting and performance analysis of EC-Earth system on MareNostrum Supercomputer

1. Introduction

1.1. EC-Earth System Model

The European Community Earth system model (EC-Earth) is an Earth System Model (ESM) that was developed by a number of European National Weather Services and university groups. It is based on the seasonal prediction system of European Centre for Medium-Range Weather Forecasts (ECMWF) [Hazelenger et al, BAMS].
EC-Earth is under current development and consists of two main components: an atmosphere, chemistry, land and vegetation model and an ocean and sea-ice model. Each of these two main components contains various sub-components, which represent physical processes, and climate-related biological and geochemical processes. These models communicate with each other through a coupler. The used programming languages are FORTRAN and C [Brandt, 2010]. The final software architecture of EC-Earth will consist of five main software codes:

1. Coupler: The coupler software used is OASIS3, which manages a number of operations related with a simulation with two o more interacting models. The interacting models are the atmosphere model together with a land and vegetation module, an atmospheric chemistry model and an ocean and sea-ice model.

2. Atmosphere model: The atmosphere model of EC-Earth is based on cycle 31r1 of the Integrated Forecasting System (IFS) of ECMWF. The basic configuration contains 62 levels in the vertical, with a model top at 5hPa (~37 km). The dynamical part of the model uses a spectral transform approach, with a present horizontal resolution of T159. The physical parametrization schemes of the model (including clouds, rain, radiation, turbulence and land surface processes) are all calculated on a reduced N80 Gaussian grid, which corresponds to a 1.125 degrees spacing (125 km). The atmospheric model uses a two time-level semi-Langrangian scheme for its dynamics with a 1-hour time step.

3. Ocean and sea-ice model: The ocean and sea-ice model is based on version 2 of NEMO (Nucleus for European Modelling of the Ocean). The ocean general circulation model (OGCM) is OPA (Océan Parallélisé). OPA is a primitive equation model which is numerically solved on a global ocean curvilinear grid known as ORCA. ORCA1 has a resolution of 1 degree and the southern pole grid is refined to resolve the circumpolar currents. The vertical grid has 31 levels (the 31st level is below the ocean bottom) with variable layer depth and a constant 10 m step in the top 150m. The Louvain-la-Neuve sea-Ice Model (LIM) is a thermodynamic-dynamic sea-ice model directly coupled with OPA.

4. Atmospheric chemistry model: The global chemistry model of EC-Earth, is called TM5. It describes the atmospheric chemistry and transport of reactive or inert tracers It can be run on a global spatial resolution of 6x4 or 3x2 degrees, with the option to increase the resolution to 1x1 degrees over specific regions, e.g. over Europe, the United States or Asia. It has been included as an online module that evaluates the transport and concentrations of reactive gases and various aerosol types for the meteorological conditions simulated by IFS. The concentrations of the simulated greenhouse gases and the concentrations and relevant optical properties of the aerosols can be fed back to calculate the associated direct and indirect radiative forcings in IFS. Exchange between TM5 and IFS currently takes place on a 3- or 6-hourly basis.

5. Land and vegetation module: The land-vegetation model is the hydrological extension of the Tiled ECMWF Surface Scheme for Exchange processes over Land (HTESSEL) and it is part of the atmosphere model.

The EC-Earth ESM has been ported successfully over different high performance computing platforms (e.g., IBM P6 AIX, CRAY XT-5, Intel-based Linux Clusters, SGI Altix) at different sites in Europe (e.g., KNMI, ICHEC, ECMWF). The current version of EC-Earth consists of three models: IFS, OASIS and NEMO.

1.2. BSC MareNostrum Supercomputer

The Barcelona Supercomputing Center (BSC) hosts MareNostrum, one of the most powerful supercomputer in Europe and the number 87 in the world [Top500 list, June 2010]. MareNostrum was built in March 2004 as a result of an agreement between the Spanish government and IBM. MareNostrum is a node of DEISA2 consortium and BSC is currently a partner in the PRACE project. In this sense, MareNostrum is targeted as one of the first DEISA2 HPC facilities to test ESMs.
The MareNostrum supercomputer is based on processors PowerPC, architecture BladeCenter, Linux operating system and Myrinet interconnection. MareNostrum has 10240 IBM Power PC 970MP processors, 20 TB of main memory and 280 + 90 TB of disk storage. It uses two interconnection networks: Myrinet and Gigabit Ethernet. It is the first supercomputer that runs under a Linux operating system, a SuSe Distribution. The Peak Performance of the system is 94.21 Teraflops.
Marenostrum has 44 racks, 31 of them dedicated to computing tasks. The computing racks have a total of 10240 processors. Each rack is formed by 6 Blade Centers. In total, each rack has 336 processors and 672 Gb of memory; each one has a rough peak performance of 3.1 Tflops.

Each Blade Center has 14 server blades type JS21. Each of these nodes has 2 processors PowerPC 970MP at 2.3 GHz, 8 Gb of shared memory between both processors and a local SAS disk of 36 Gb. Each node has a network card Myrinet type M3S-PCIXD-2-I for its connection to the high speed interconnection and the two connections to the network Gigabit. Each node has a local disk of 36 Gb and works diskless, i.e., the operating system is in the storage racks instead of the local disk and it is loaded through a Gigabit network when each node is initialized.
In addition to the local disk of each node, MareNostrum has 20 storage servers arranged in 7 racks. These servers have a total of 560 disks of 512GB and each one provides a total capacity of 280TB external storage. These disks are working with

Global Parallel File System (GPFS), which offers a global vision of the file system and also allows a parallel access.

Default compilers in Marenostrum are IBM XL C/C++, and IBM XL FORTRAN. In addition, the GNU C and FORTRAN compilers are available.
Several numerical libraries and several application packages are installed in MareNostrum (http://www.bsc.es/plantillaC.php?cat_id=472).

2. Objectives

The main objective of this work is to identify and document the issues related with the portability and performance of the EC-Earth model on the MareNostrum supercomputer. In order to meet the standards for future HPC architectures, it is important to understand the current performance of ESMs on state-of-the-art computing systems. In this sense, we focus on the porting and performance analysis of EC-Earth model on the MareNostrum supercomputer.

In this report, we describe the efforts done to port the coupled EC-Earth model on MareNostrum and the preliminary optimization tasks undertaken to improve the default performance of the modeling system. A description of the execution characteristics of the system is presented and results from a scalability study are discussed. Finally, we present initial analysis of raw performance traces using the Paraver analysis software tool developed at BSC and conclusions for future work.

3. Description of execution and evaluation tests

3.1 Porting

Three versions of EC-Earth have been compiled and run in MareNostrum: version 2.0, 2.1 and 2.2. Detailed instructions for compiling and running the model in its version 2.0 are found in Stefanescu, 2008.

To compile EC-Earth (versions 2.0, 2.1 and 2.2) at MareNostrum some modifications on the code were need. First, we included the flag -qextname for the multiple configuration files and routines that required extra underscoring. The most optimized compiler in MareNostrum is IBM XL compilers since MareNostrum is an IBM machine with PPC970 processors. This type of compilers does not include by default the underscoring in the Fortran routines, which it was required by the C code that calls to Fortran routines in the original EC-Earth code. Second, changes in the source code were required to adapt the program to our architecture, so the IBM AIX structure was selected as base. This approach required some tuning to match the architecture IBM PPC LINUX running at BSC.

In addition to the changes described above, a particular compilation for IFS was needed for versions 2.1 and 2.2 as a result of an incompatibility of the IFS software and the compiler XLF version 12.1. After extensive testing, a two step compilation approach was used to compile IFS within EC-Earth using XLF version 10.1 and 12.1 compilers. Compiling all IFS with XLF v10.1 was not feasible as other modules within EC-Earth were only available for the new XLF v12.1 compiler.

A platform intercomparison using EC-Earth version 2.1 and a simplified 10-year simulation run showed that EC-Earth ports and performs well in different architectures of European supercomputing centers including MareNostrum.

3.2 Execution

EC-Earth is executed in parallel using MPI and the default setting of 16 CPUs for NEMO and 1 CPU for OASIS. For NEMO, a default number of 16 CPUs was established by the EC-Earth community. For OASIS, one CPU is used as EC-Earth does not use a parallelized version of this coupler. Table 5.1 summarizes the requirements and resources needed to execute EC-Earth in our platform. MareNostrum uses a batch processing support, so all jobs must be run through it. The batch system used in MareNostrum is a combination of two softwares: 1) SLURM, developed at Lawrence Livermore National Laboratory and designed for large clusters, which works as a resource manager and 2) MOAB, developed at Cluster Resources company, which works as a job scheduler. The user then needs to specify the number of CPUs allocated for each task. This is useful for hybrid MPI+OpenMP applications, in which each process spawns a number of threads. The number of CPUs per task must be between 1 and 4, since each node has 4 CPUs (one for each thread). It is important to note that OpenMP settings are globally defined for a whole run. It is also possible to define the number of tasks allocated in each node. When an application uses more than 1.7GB of memory per process, it is not possible to have 4 processes in the same node because of an 8GB memory limit. Due to this special configuration, the submitting batch script to send an EC-Earth run to the MareNostrum queues is different to other platforms (see Table 1 for details).

Table 1: Summary of requirements and resources needed to run EC-Earth.

Platform	
Execution platform	MareNostrum
Requirements	**ksh** - korn shell (ksh93) **perl** - used extensively with generic Makefile to build source libraries **mpi1** - required for parallel execution **Fortran 95** - a fortran 95 compiler that supports an auto-double (or -r8) capability. OASIS3 requires Cray pointers. **OpenMP** - if OpenMP is used then the MPI implementation is required to be thread-safe **netCDF** - the netCDF library for I/O of all NEMO and OASIS3
Libraries	**Oasis:** netcdf **Nemo:** netcdf **IFS:** • blas - a standard ("off the web") blas library • dummy - stubs for routines that are never called • ec- a small number of ecmwf utility routines • emos - library of data manipulation routines, like bufrdc/ bufr

	• (observation l e routines), gribex/ grib (spectral/grid point • fields) and pbio/ simple C based I/O routines • IFS - contains the main IFS source library, has many subdirectories • IFSaux - contains IFS utility routines (in particular MPL * interface to MPI) • lapack - public domain source of LAPACK routines (see also blas) • prepdata - data preparation routines • surf - surface package • trans - IFS transform routines/data distribution
CPUs Used for each component	**IFS:** 64 (NPRTRW =32 and NPRTRV =2) NPROC: Total number of CPUs (NPRTV x NPRTRW) NPRTRW: Number of CPUs used during transform phase in wave space (max 47) NPRTRV: Number of CPUs used during transform phase in vertical direction (max 62) **OASIS:** 1 **NEMO:** 16 (NEMOPROCX=4 and NEMOPROCY=4)
Submit job command	NEMO_nproc=$((NEMO_nprocX*NEMO_nprocY)) minNEMO_nproc=$((IFS_nproc+1)) maxNEMO_nproc=$((IFS_nproc+NEMO_nproc)) total_nproc=$((maxNEMO_nproc+1)) echo "0 ${OASIS3_exe}" > silly.conf echo "1-$IFS_nproc ${IFS_exe} -v ecmwf -e $EXPVER" >> silly.conf echo "$minNEMO_nproc-$maxNEMO_nproc ${NEMO_exe}" >> silly.conf srun -n$total_nproc -l --multi-prog silly.conf

3.3. Optimization

The optimization process mainly focused on adapting the compilation flags in order to improve the efficiency of the code (e.g., inlining of specific functions, test different levels of optimizations and libraries, etc). In addition, we checked the performance of the I/O in the two filesystems available in MareNostrum: 1) /scratch/, which is the local filesystem only accessible from each node and with 36 GB, and 2) /gpfs/, which is globally available from any node and with more than 100 TB of storage space. We chose to run EC-Earth within gpfs/ because the use of the local filesystem did not improve the I/O behaviour, and scratch/ was limited in space and output files were more difficult to recover.

3.4 Scalability

To understand the performance of EC-Earth in MareNostrum, we did a scalability test using EC-Earth version 2.2. The experiment consisted of running 1-month simulations for different numbers of CPUs for IFS (i.e., 4, 8, 16, 32, 36, 64, 96, 128 and 256). In this study, we kept constant setting of 16 CPUs for NEMO and 1 CPU for OASIS. We considered the number of IFS CPUs as multiple of four because MareNostrum

has four CPUs per node and to avoid performance issues due to share memory resources with other applications. A constant number of 16 CPUs was considered for NEMO since that was the default setting established by the EC-Earth community for the Coupled Model Intercomparison Project (CMIP-5).

Figure 1 shows the speedup and the scalability efficiency of EC-Earth for different number of IFS CPUs, and Table 2 summarizes the execution times. Only results from 4 to 128 IFS CPUs are shown because running EC-Earth with 256 IFS CPUs was not feasible due to MPI communication problems (only 2 out of 10 tests were successful). The relative speedup for each number of CPUs is calculated as the ratio of execution time of the base number of CPUs (in this case, 4 IFS CPUs, i.e., a total of 24 CPUs) and the execution time with X CPUs; efficiency is the relative speedup multiply by the ratio of the CPUs (i.e., 24 over X). It is important to note that the standard metric speedup is the ratio over the run with only one CPU. We used the metric speedup ratio in this analysis as our objective was to determine the optimum number of IFS CPUs to efficiently run EC-Earth. These results show that the efficiency of EC-Earth in MareNostrum decreases when a large number (> 64) of IFS CPUs are used, and that the optimum number of IFS CPUs to run EC-Earth in MareNostrum is 64. It is important to note that the efficiency values are above one, which indicates an issue on the EC-Earth performance when using a low number of IFS CPUs (<16). In addition, we tested the EC-Earth performance for different balances between *NPRTRW* and *NPRTRV* and found no significant performance variation.

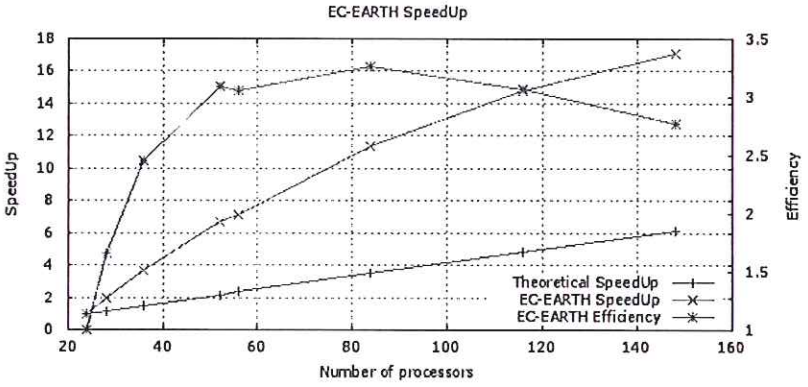


Figure 1: Speedup and scalability efficiency of EC-Earth runs from 24 to 148 CPUs. Numbers reported are the average of 10 runs (excluding the lower and the upper outlier to avoid MareNostrum instability performances).

Table 2: Execution time for each EC-Earth run. Numbers reported are the average of 10 runs (excluding the lower and the upper outlier to avoid MareNostrum instability performances). Total number of CPUs is rounded to the closest multiple of four.

Number of CPUs	Execution Time
24 (1 + 16 + 4)	13:53:58
28 (1 + 16 + 8)	07:13:39
36 (1 + 16 + 16)	03:48:08
52 (1 + 16 + 32)	02:05:11
56 (1 + 16 + 36)	01:57:49
84 (1 + 16 + 64)	01:13:35
116 (1 + 16 + 96)	00:56:35
148 (1 + 16 + 128)	00:49:00

Table 3 shows a summary of computation times within EC-Earth considering coupling and no-coupling times. The computation time is the addition of the time from the log for all time step, the time without coupling is estimated as the product of number of time steps by the mean time of the time steps without coupling stage and the coupling percentage is the fraction of time that the code uses for coupling, based on the coupling and non-coupling times. We find that the coupling percentage increases with the number of CPUs due to a decrease in computation time. From the log files (not shown), we find that there is a significant larger time for processing every 3 hours, likely as a result of this coupling.

Table 3: Example of computing and coupling times extracted from a single EC-Earth run. The total number of CPUs is rounded to the closest multiple of four.

Number of CPUs	Computation time	Coupling percentage	Time without coupling
24 (1 + 16 + 4)	13:53:23	13,27%	12:02:50
28 (1 + 16 + 8)	07:11:45	13,73%	06:12:27
36 (1 + 16 + 16)	03:46:52	14,56%	03:13:51
52 (1 + 16 + 32)	02:03:46	22,01%	01:36:25
56 (1 + 16 + 36)	01:57:24	25,16%	01:27:51
84 (1 + 16 + 64)	01:11:50	29,14%	00:50:54
116 (1 + 16 + 96)	00:55:41	34,21%	00:36:38
148 (1 + 16 + 128)	00:47:22	39,20%	00:28:48

3.5 Traces

We used Paraver to further test the performance of EC-Earth in MareNostrum. Paraver is an open source performance visualization and analysis tool developed in BSC. Figure 2 shows an example of one of the Paraver visualization modes: colors indicate the CPU states (e.g., running, waiting for communication, synchronization, group communication, communication send, I/O), the xaxis represents time, and the

yaxis represents the different CPUs (e.g., 1 for OASIS, 4 for IFS and 16 for NEMO in our example).

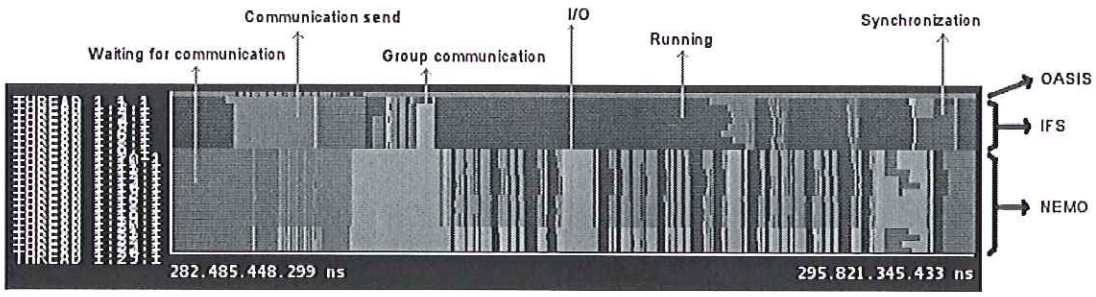


Figure 2: Example of a Paraver graphical view. OASIS, IFS and NEMO are run with 1, 8 and 16 CPUs, respectively. .

In addition from visualizing the CPU behaviour, performance statistics can be extracted from Paraver. We analyzed some of these parameters: the "load balance" is the average percentage of time each thread is executing over the maximum of that percentage (one for an ideal perfect balanced code and one/CPUs# for a totally unbalanced code); the "communication factor" is the maximum of the percentages of time any thread is executing, the "code replication" reflects the increased work the model will have to do, due to non-parallelized code repeated over several CPUs and the cost of the computations needed to perform the communications, i.e., the total number of instructions of that executions over the number of instructions executed for the 24 CPUs run; the "average instruction per cycle (IPC)" describes the internal behaviour of the code; the "speedup" is the execution time of the base number of CPUs (in this case, 24), over the execution time with X CPUs; the "efficiency" is the ratio of speedup over the ratio of the CPUs.

For a reference, the range obtained from load balance, communication factor, code replication and average IPC parameters gives an idea of the parallelization performance of the code in a specific aspect (> 0.9 well, 0.8-0.9 fairly well and <0.8 can be improved). It is important to note that values below 0.8 do not mean that there is a problem within the code, as this regular performance might be inherent to the model.

As a preliminary test, we created traces for 1-day simulation runs. Figure 3 shows results for the tests performed with 16, 32 and 64 IFS CPUs. This test shows the existence of a load imbalance between IFS and NEMO (see red areas in NEMO compared to IFS, i.e., NEMO is waiting for communication while IFS is still processing). However, this load imbalance improves as the number of IFS CPUs increases. In addition, we find that somehow there is a serial processing at the startup and some other periods (see highlighted regions in green), and that there is a significant larger time for processing every 3 hours due to the coupling (not shown).

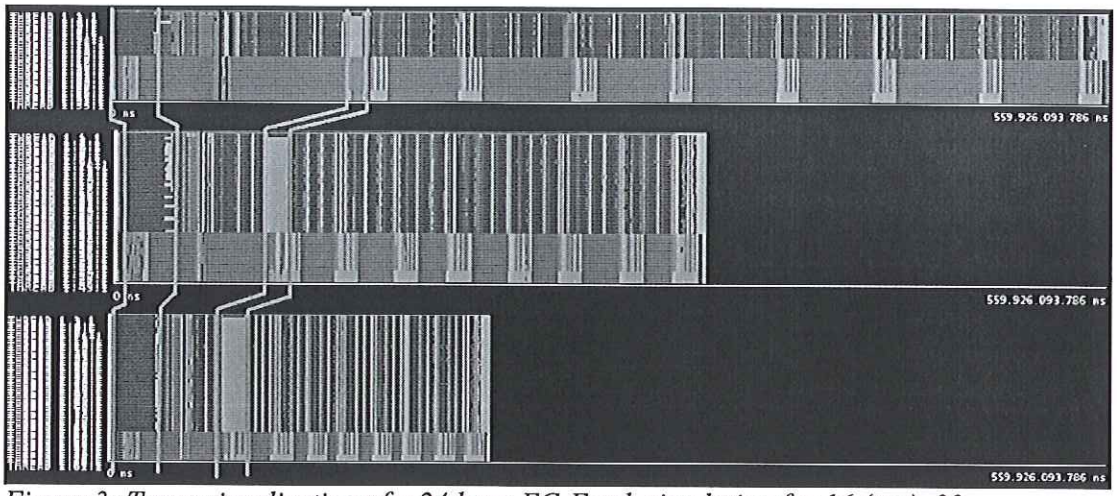


Figure 3: Trace visualization of a 24 hour EC-Earth simulation for 16 (top), 32 (middle) and 64 (bottom) IFS CPUs. NEMO and OASIS were run with 16 and 1 CPUs, respectively. Colors indicate running (blue), waiting for communication (light red), synchronization (dark red), group communication (orange), communication send (pink) and I/O (green). Green vertical lines indicate the initialization part of the trace that does not scale.

Figure 4 shows a comparison of a 3-hour run, including one coupling step, for EC-Earth using 16, 64 and 128 IFS CPUs (please, note the time axis scale is different in each plot). This Paraver analysis shows clearly that OASIS acts as a bottleneck and is always an important limiting factor independently of the number of IFS CPUs used because during coupling both IFS and NEMO are waiting for communication. Based on the Paraver analysis, we find that the coupling time takes about 2 seconds, confirming that the coupling time calculated in Table 5.2 corresponds in fact to the coupling stage.

In addition, we observed the existence of an extreme load imbalance between IFS and NEMO on the 16 IFS CPUS and a large imbalance for 64 IFS CPUs, indicating that IFS acts as a bottleneck and another important limiting factor in EC-Earth. The load imbalance improves with 128 IFS CPUs and IFS and NEMO can both act as limiting factor depending on the time step.

Overall, this analysis shows that IFS spends most of the time running, whereas NEMO spends an important fraction of time doing data transfer. This implies that, at MareNostrum, NEMO may not scale well beyond 16 CPUs. Previous studies showed the significant scalability of NEMO using MPI-OpenMP and recommend running NEMO using this parallel technique. However, we use only MPI parallelization for NEMO at MareNostrum because IFS is run with MPI. MareNostrum is a distributed memory system and does not support a hybrid parallel execution, i.e., one software running with MPI parallelization and another with OpenMP+MPI within the same submit job.

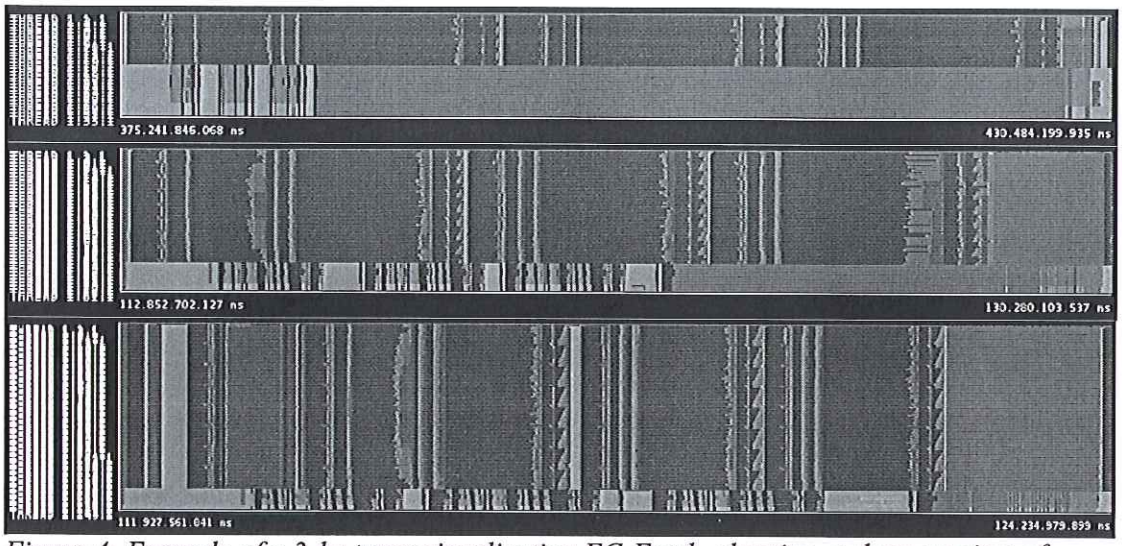


Figure 4: Example of a 3-hr trace visualization EC-Earth, showing a close-up view of the coupler communication for 16 (top), 64 (middle) and 128 (bottom) IFS CPUs. NEMO and OASIS were run with 16 and 1 CPUs, respectively. Colors indicate running (blue), waiting for communication (light red), synchronization (dark red), group communication (orange), communication send (pink) and I/O (green).

Table 3 summarizes the statistical parameters calculated from the Paraver analysis. This table shows that load balance tends to improve as the number of IFS CPUs increases. Communication performance decreases with the number of IFS CPUs, as a result of a reduction on the percentage of time that is used for calculations while the amount of communication remains the same. Similar behaviour is observed for code replication due to an increase in the calculations needed to distribute the workload and to non-parallelization of parts of the code. Not surprisingly, the average IPC improves as the number of IFS CPUs increases. Based on a further analysis of the average IPC from Paraver (not shown), we find that IFS has a significant larger IPC than NEMO and OASIS. Similarly to results found in section 3.4, the Paraver analysis confirms that the efficiency of EC-Earth in MareNostrum decreases when a large number (> 64) of IFS CPUs is used, and that the optimum number of IFS CPUs to run EC-Earth in MareNostrum is 64.

Table 3: Statistic metrics obtained with Paraver excluding the initialization stage (i.e., 21-hour run). Total number of CPUs is reported to the closest multiple of four.

# of CPUs	24	28	36	52	56	84	116	148
Load balance	0,21	0,36	0,54	0,71	0,73	0,84	0,84	0,64
Communications	0,97	0,95	0,91	0,84	0,84	0,76	0,72	0,62
Code replication	1	0,82	0,92	0,69	0,61	0,57	0,55	0,43
Avg. IPC	0,31	0,37	0,38	0,5	0,59	0,63	0,64	0,66
Speedup	1	1,92	3,63	6,65	7,39	11,63	15,22	12,61
Efficiency	1	1,65	2,42	3,07	3,17	3,32	3,15	1,22

4. Summary and Conclusions

The EC-Earth ESM was successfully ported to the BSC MareNostrum supercomputer, a system based on IBM PowerPC 970MP processors and run under a Linux Suse distribution. We detected a compilation incompatibility of IFS in EC-Earth v 2.1 and 2.2 to the compiler XLF version 12.1. This incompatibility was solved by a two step compilation approach using the XLF version 10.1 and 12.1 compilers.

The EC-Earth performance was analyzed with respect to scalability and trace analysis with the Paraver software. The performance analysis was done using the standard configuration of EC-Earth version 2.2, which uses 1 CPU for OASIS3 and 16 CPUs for NEMO, and modifying the number of CPUs allocated for the IFS model.
Our analysis showed that EC-Earth performs fairly well in MareNostrum, and that the efficient number of IFS CPUs is 64. Running EC-Earth with a larger number of IFS CPUs (>128) is not feasible at the moment since some issues exists with the IFS-NEMO balance and MPI communications. We detected a negligible load imbalance within IFS, which is typical in models with complex physical calculations, and an important performance loss of EC-Earth in the coupling stage, as OASIS acts as a bottleneck since the serial version of OASIS is used.

5. References

Hazeleger, W. et al., 2009. EC-Earth: A Seamless Earth System Prediction Approach in Action, accepted, Bull. Amer. Meteor. Soc.

Martijn Brandt, EC-EARTH- the European Community Earth system model, March 2010 [http://ecearth.knmi.nl/EC-Earth_model_documentation.pdf]

Stefanescu S., Standalone environment for compiling and running the EC-EARTH system, Technical Note, April 2008 [http://ecearth.knmi.nl/ecearth2.pdf].

6. Acknowledgments

We wish to thank Xavier Abellán from the BSC Operations team, and Pedro Jiménez from Universidad de Murcia for their help porting and running EC-Earth in Marenostrum, and Judit Giménez of Computer Sciences Department of BSC for her help on the analysis of Paraver traces. We appreciate Simona Stefanescu from ECMWF and Tido Semmler from Met Eireann for technical help with the EC-Earth scripts.

Publications and Presentations:

G. Aloisio, C. Basu, A. Caubel, I. Epicoco, O. Jorba, E. Maisonnave, F. Martínez, S. Mocavero, M. Val-Martin, S. Valcke, D. Vicente, Report on the Description of the Evaluation Suite and Base-case Results, Project Number GA: 228203, IS-ENES, Infrastructure for the European Network for Earth System Modelling.

P. Jiménez-Guerrero, M. Val-Martin, J.P. Montávez and J.M. Baldasano, Evaluation of a regional model climatology in Europe using EC-Earth V2.0 data. EC-Earth International Annual Meeting, Barcelona, May 31-June 1, 2010.