Earth Sciences Department

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EC EARTH

BSC testing protocol and tools

March 20, 2024

Eric Ferrer, Gilbert Montané, Genís Bonet, Rohan Ahmed

Index

Intro: Motivation

Why test the software?

- Find any bugs before production runs start

Intro: Motivation

Why test the software?

- Find any bugs before production runs start
- Test any new implemented features

Intro: Motivation

Why test the software?

- Find any bugs before production runs start
- Test any new implemented features
- Software architecture differences

Intro: Motivation

An easy to fix bug related to how some machines intrepret the number 08 (or 09) in a text as octal numbers leads to some errors like this one*:

```
member_index=$(echo $(($(echo ${MEMBER} | cut -c3-) + 1)))
echo $(($(echo ${MEMBER} | cut -c3-) + 1))
echo ${MEMBER} | cut -c3-
+++ cut -c3-
+++ echo fc08
/home/bsc32/bsc32627/.lsbatch/1630476521.594678.shell: line 55: 08: value too great for base (error token is "08")
+ member_index=
```

* shown running manually the commands

Intro: Motivation

An easy to fix bug related to how some machines intrepret the number 08 (or 09) in a text as octal numbers leads to some errors like this one*:

```
member_index=$(echo $(($(echo ${MEMBER} | cut -c3-) + 1)))
echo $(($(echo ${MEMBER} | cut -c3-) + 1))
echo ${MEMBER} | cut -c3-
+++ cut -c3-
+++ echo fc08
/home/bsc32/bsc32627/.lsbatch/1630476521.594678.shell: line 55: 08: value too great for base (error token is "08")
+ member_index=
```

The fix is just a 3 character addition to the line:

```
member_index=$((10#$(echo ${MEMBER} | cut -c3-) + 1))
```

* shown running manually the commands

Intro: Motivation

An easy to fix bug related to how some machines intrepret the number 08 (or 09) in a text as octal numbers leads to some errors like this one*:

```
member_index=$(echo $(($(echo ${MEMBER} | cut -c3-) + 1)))
echo $(($(echo ${MEMBER} | cut -c3-) + 1))
echo ${MEMBER} | cut -c3-
+++ cut -c3-
+++ echo fc08
/home/bsc32/bsc32627/.lsbatch/1630476521.594678.shell: line 55: 08: value too great for base (error token is "08")
+ member_index=
```

The fix is just a 3 character addition to the line:

```
member_index=$((10#$(echo ${MEMBER} | cut -c3-) + 1))
```

Detecting it in the early test phases is key

* shown running manually the commands

Intro: Motivation

Differences between different HPC's login nodes*:

- Marenostrum4 HPC login node (has default Python and modules):

```
bsc32627@login3:~> python
Python 2.7.13 (default, Jan 11 2017, 10:56:06) [GCC] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
bsc32627@login3:~> module load intel
Set INTEL compilers as MPI wrappers backend
```

* shown running manually the commands

Intro: Motivation

Differences between different HPC's login nodes*:

- Marenostrum4 HPC login node (has default Python and modules):

```
bsc32627@login3:~> python
Python 2.7.13 (default, Jan 11 2017, 10:56:06) [GCC] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
bsc32627@login3:~> module load intel
Set INTEL compilers as MPI wrappers backend
```

- HPC2020 HPC login node (has modules but not default Python):

```
[c3ef@ac6-100 ~]$ python
-bash: python: command not found
[c3ef@ac6-100 ~]$ module load intel
```

* shown running manually the commands

Intro: Motivation

Differences between different HPC's login nodes*:

- Marenostrum4 HPC login node (has default Python and modules):

```
bsc32627@login3:~> python
Python 2.7.13 (default, Jan 11 2017, 10:56:06) [GCC] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
bsc32627@login3:~> module load intel
Set INTEL compilers as MPI wrappers backend
```

- HPC2020 HPC login node (has modules but not default Python):

```
[c3ef@ac6-100 ~]$ python
-bash: python: command not found
[c3ef@ac6-100 ~]$ module load intel
```

- MeluXina HPC login node (doesn't have either default Python nor modules):

```
[u100498@login03 ~]$ python
-bash: python: command not found
[u100498@login03 ~]$ module load intel
-bash: module: command not found
```

* shown running manually the commands

Intro: Motivation

These differences between HPC's remark some of the cases when there is an issue

- **No issues:** Version match -> everything works

Intro: Motivation

These differences between HPC's remark some of the cases when there is an issue

- **No issues:** Version match -> everything works
- **Best case error:** No default version -> error shown and program crashes

Intro: Motivation

These differences between HPC's remark some of the cases when there is an issue

- **No issues:** Version match -> everything works
- **Best case error:** No default version -> error shown and program crashes
- **Worst case error:** Version doesn't match -> No crash -> possibility of hidden errors

Intro: Motivation

What parameters and options may affect an EC-Earth experiment that is running with Autosubmit (auto-EC-Earth)?

- HPC (architecture)
- LEGSIZE and #LEGS
- #START DATES
- #MEMBERS
- ACCOUNT (RES/BSC)
- RESTARTS (cold start/restart)
- COMPONENTS (IFS, NEMO[PISCES], TM5, LPJG)
- CMORIZATION (T/F)
- ECE3 POSTPROC (T/F)
- PRODUCTION FLAGS (T/F)
- PRECOMPILED BINARIES (T/F)
- SAVE_IC configurations
- TRANSFER PROCESS to archive
- SAVE RAW OUTPUT (MMA, MMO, DDA, ICMCL...)

Intro: Motivation

What parameters and options may affect an EC-Earth experiment that is running with Autosubmit (auto-EC-Earth)?

- HPC (architecture)
- LEGSIZE and #LEGS
- #START DATES
- #MEMBERS
- ACCOUNT (RES/BSC)
- RESTARTS (cold start/restart)
- COMPONENTS (IFS, NEMO[PISCES], TM5, LPJG)
- CMORIZATION (T/F)
- ECE3 POSTPROC (T/F)
- PRODUCTION FLAGS (T/F)
- PRECOMPILED BINARIES (T/F)
- SAVE_IC configurations
- TRANSFER PROCESS to archive
- SAVE RAW OUTPUT (MMA, MMO, DDA, ICMCL...)

With this many variables, **is impossible to run one experiment with each different configuration available**.

Intro: End-to-End workflows

Why should we use end-to-end, full workflows for testing implementations?

Intro: End-to-End workflows

Why should we use end-to-end, full workflows for testing implementations?

- **Tailored tests** -> can lead to implementing **new bugs**
- **Not use real cases** -> **not testing the current behavior**

Intro: End-to-End workflows

Why should we use end-to-end, full workflows for testing implementations?

- **Tailored tests** -> can lead to implementing **new bugs**
- **Not use real cases** -> **not testing the current behavior**

Summarizing:

- too general -> not useful -> might not check any specifics
- too specific -> not useful -> might not test different cases

Intro: End-to-End workflows

Why should we use end-to-end, full workflows for testing implementations?

- **Tailored tests** -> can lead to implementing **new bugs**
- **Not use real cases** -> **not testing the current behavior**

Summarizing:

- too general -> not useful -> might not check any specifics
- too specific -> not useful -> might not test different cases

Best option: multiple tests, with different real configurations, running in parallel after any new integration

Intro: Autosubmit workflow manager

Using a workflow manager is a "must" with these kind of complex workflows. BSC-ES develops the Autosubmit workflow manager, which is used to run all modelling experiments, including those of EC-Earth *. Some of its features are:

* Autosubmit is an open source Python tool that supports various infrastructures such as Destination Earth or EDITO. (**https://pypi.org/project/autosubmit/**)

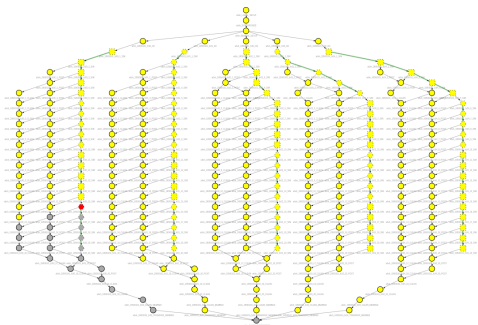
Intro: Autosubmit workflow manager

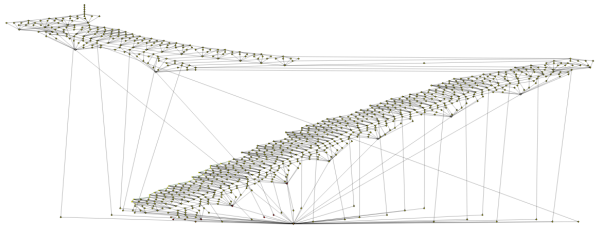
Using a workflow manager is a "must" with these kind of complex workflows. BSC-ES develops the Autosubmit workflow manager, which is used to run all modelling experiments, including those of EC-Earth *. Some of its features are:

- **Automatization**
- **Built-in experiment manager**
- **Multi-platform**
- **Portable and interoperable**
- **RESTful API and web GUI**

* Autosubmit is an open source Python tool that supports various infrastructures such as Destination Earth or EDITO. (**https://pypi.org/project/autosubmit/**)

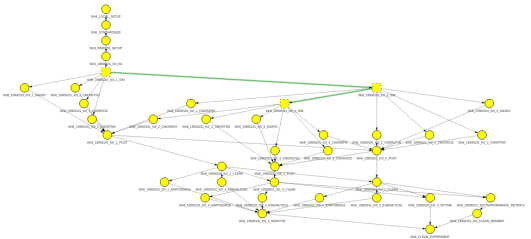
Intro: Example Workflows (Real case)





Multi-member (5), 20-leg (long) experiment. 1 member, 86-leg (long) experiment.

Intro: Example workflows (Test case)



Multi-member multi-startdate (2&2), 2-leg experiment. 1 member, 3-leg experiment.

Testing suite tool

The testing suite (TS) software is a Python project, made at BSC, that allows anyone using Autosubmit to **control a set of test experiments** from the command line and to execute operations for **all at once**.

Testing suite tool

The testing suite (TS) software is a Python project, made at BSC, that allows anyone using Autosubmit to **control a set of test experiments** from the command line and to execute operations for **all at once**.

Main features of the tool:

- **Control multiple experiments**
- **Check performance and output**
- Compare the configuration of all the experiments.
- Generate reports.
- (In development): modularization of the tool.

TS: Protocols

Auto-EC-Earth 3 testing protocols *:

	WEEKLY	**RELEASES**
When (frequency)	After **any merge** to trunk (once a month at least)	Every official release (once or twice a year)
Which branch	**trunk** (continuously changing)	**TAG** X.X.X (frozen after validation)
Compilation	Every test	Precompiled binaries on MN4, and compiled on other HPCs
Duration of test cases	**Shorter**: monthly legs (∼67%) except a few yearly (∼33%)	**Longer**: yearly legs (∼75%) except a few monthly (∼25%)
Validation	**Only Technical**	**Both Technical and Scientific**

* official EC-Earth3 releases are tested too, following a similar procedure to the auto-EC-Earth 3 Releases protocol

TS: Running experiments

Here is the procedure (commands) to run a new set of experiments:

```
./ts.sh --experiments    #to check the list of experiments
./ts.sh --clean          #to clean any output from previous runs
./ts.sh --refresh        #to update the branch from the repository
./ts.sh --create         #to re-set the experiments status to start from scratch
./ts.sh --run            #to run the experiments
./ts.sh --status         #to monitor the current status of the experiments
```

TS: Running experiments

Most of the commands have an output pre-formatted for gitlab markdown, so it can be directly copy-pasted to any issue, so this terminal output:

```
| test id   | status   | job              | issue          | details | hpc     |
|-----------|----------|------------------|----------------|---------|---------|
| t0ao | Successful |   |   | TAG 3.3.4 test case: ORCA1L75 +  surface restoring - CHUNKSIZE=12 |
marenostrum4 |
| t0ap | Successful |   |   | TAG 3.3.4 test case: T255L91 (IFS only) - CHUNKSIZE=12 |
marenostrum4 |
| t0au | QUEUING | t0au_19900101_fc0_3_SIM  |   | TAG 3.3.4 test case: T255L91-ORCA1L75-LPJG-
PISCES-TM5 co2,co2fb - CHUNKSIZE=12 |  marenostrum4 |
| t0av | Successful |   |   | TAG 3.3.4 test case: OSM-LPJG - RES Account - CHUNKSIZE=12 |
marenostrum4 |
| t0ay | Successful |   |   | TAG 3.3.4 test case: ORCA1L75 nord + surface restoring |  nord3v2 |
| t0b7 | Successful |   |   | TAG 3.3.4 test case: T255L91-ORCA1L75 cold start -
000_SYNC/DT_INTERMEDIATE_STORAGE - CHUNKSIZE=12 |  marenostrum4 |
| t0b8 | Successful |   |   | TAG 3.3.4 test case: ORCA025L75 (NEMO only) + surface restoring -
CHUNKSIZE=12 |  marenostrum4 |
| t0b9 | Successful |   |   | TAG 3.3.4 test case: T511L91 - CHUNKSIZE=12 |  marenostrum4 |
| t0ba | FAILED | t0ba_19930101_fc0_1_POST  |   | TAG 3.3.4 test case: T511L91-ORCA025L75 -
CHUNKSIZE=12 |  marenostrum4 |
```

TS: Running experiments

Becomes this nice table on gitlab:

test id	status	job	issue	details	hpc
t0ao	Successful			TAG 3.3.4 test case: ORCA1L75 + surface restoring - CHUNKSIZE=12	marenostrur
t0ap	Successful			TAG 3.3.4 test case: T255L91 (IFS only) - CHUNKSIZE=12	marenostrur
t0au	QUEUING	t0au_19900101_fc0_3_SIM		TAG 3.3.4 test case: T255L91-ORCA1L75-LPJG-PISCES-TM5 co2,co2fb - CHUNKSIZE=12	marenostrur
t0av	Successful			TAG 3.3.4 test case: OSM-LPJG - RES Account - CHUNKSIZE=12	marenostrur
t0ay	Successful			TAG 3.3.4 test case: ORCA1L75 nord + surface restoring	nord3v2
t0b7	Successful			TAG 3.3.4 test case: T255L91-ORCA1L75 cold start - 000_SYNC/DT_INTERMEDIATE_STORAGE - CHUNKSIZE=12	marenostrur
t0b8	Successful			TAG 3.3.4 test case: ORCA025L75 (NEMO only) + surface restoring - CHUNKSIZE=12	marenostrur
t0b9	Successful			TAG 3.3.4 test case: T511L91 - CHUNKSIZE=12	marenostrur
t0ba	FAILED	t0ba_19930101_fc0_1_POST		TAG 3.3.4 test case: T511L91-ORCA025L75 - CHUNKSIZE=12	marenostrur

TS: Results provided

We can produce a **performance report**, and also a comparison with a previous run of the same experiments. This allows to easily see any regresion in model performance.

test id	status	sim run time avg	sim runs	#PROC	SYPD	ASYPD	CHSY	JPSY	details
t0n2	Successful	0:42:27	1	480	33.92	7.76	339.6	14890000.0	weekly test case: ORCA1L75 + nudgi surface restoring
t0n3	Successful	1:05:19	1	528	22.05	3.0	574.79	29820000.0	weekly test case: T255L91 (IFS only)
t0n4	Successful	0:08:19	8	768	14.41	0.53	1279.55	58793517.41	weekly test case: T255L91-ORCA1L7 restart. ScenarioM ssp245 outclass
t0n5	Successful	1:19:23	2	96	1.51	0.44	1524.77	69207683.07	weekly test case: T255L91-TM5 full c
t0n6	Successful	1:23:58	2	192	1.4	0.48	3302.44	128511404.56	weekly test case: T255L91-ORCA1L7 full chem
t0n7	Successful	2:01:40	3	768	11.84	1.23	1557.4	73273333.33	weekly test case: T255L91-ORCA1L7 LPJG-PISCES
t0n8	Successful	3:13:10	3	528	7.45	0.87	1699.87	82070000.0	weekly test case: T255L91-ORCA1L7 LPJG-PISCES-TM5 co2,co2fb

expid	SYPD (previous)	ASYPD (previous)	details	hpc
t0n2	63.13 (+0.27)	5.08 (-17.91)	weekly test case: ORCA1L75 + nudging + surface restoring	marenostrum4
t0n3	29.55 (-0.15)	4.32 (-11.29)	weekly test case: T255L91 (IFS only)	marenostrum4
t0n4	14.76 (-0.24)	3.48 (+1.53)	weekly test case: T255L91-ORCA1L75 (t0nf ICs). ScenarioMIP/ssp245 outclass	marenostrum4
t0n5	1.92 (0.0)	1.26 (+0.15)	weekly test case: T255L91-TM5 full chem	marenostrum4
t0n6	2.03 (+0.02)	1.21 (+0.52)	weekly test case: T255L91-ORCA1L75-TM5 full chem	marenostrum4
t0n7	12.86 (+0.05)	5.69 (+1.45)	weekly test case: T255L91-ORCA1L75-LPJG-PISCES	marenostrum4
t0n8	7.52 (+0.03)	3.25 (+1.44)	weekly test case: T255L91-ORCA1L75-LPJG-PISCES-TM5 co2,co2fb	marenostrum4
t0n9	106.73 (-1.41)	49.53 (+1.34)	weekly test case: OSM-LPJG (res account)	marenostrum4
t0na	27.18 (+0.19)	5.91 (+0.48)	weekly test case: ORCA1L75-PISCES	marenostrum4
t0nb	37.7 (-1.64)	0.54 (+0.11)	weekly test case: ORCA1L75 CMORIZATION FALSE	marenostrum4
t0nd	22.5 (+0.52)	0.32 (-20.16)	weekly test case: ORCA1L75	nord3v2
t0ne	38.5 (+0.1)	0.54 (+0.04)	weekly test case: ORCA1L75 - 000/INTERMEDIATE_STORAGE	marenostrum4
t0nf	18.49 (+0.52)	0.47 (+0.01)	weekly test case: T255L91-ORCA1L75 cold start - 000/INTERMEDIATE_STORAGE	marenostrum4
t0ng	3.32 (-0.04)	0.3 (+0.17)	weekly test case: ORCA025L75 (NEMO only)	marenostrum4

TS: Results provided

We can also **check the output** of the experiments with a provided benchmark experiment (2 checks - one for the files and another for the variables values). This allows for an easy **bit-to-bit reproducibility** check against previous runs before perfoming a more complex statistic reproducibility test.

test id	benchmark	missing files check	output checker	failures	description	hpc
t0n2	t0n2_reference	Successful	Successful		weekly test case: ORCA1L75 + nudging + surface restoring	marenostrum
t0n3	t0n3_reference	Successful	Successful		weekly test case: T255L91 (IFS only)	marenostrum
t0n4	t0n4_reference	Successful	FAILED	siconc tas tos	weekly test case: T255L91-ORCA1L75 (t0nf ICs). ScenarioMIP/ssp245 outclass	marenostrum
t0n5	t0n5_reference	Successful	Successful		weekly test case: T255L91-TM5 full chem	marenostrum
t0n6	t0n6_reference	Successful	Successful		weekly test case: T255L91-ORCA1L75-TM5 full chem	marenostrum
t0n7	t0n7_reference	Successful	Successful		weekly test case: T255L91-ORCA1L75-LPJG-PISCES	marenostrum
t0n8	t0n8_reference	FAILED	FAILED	cLand co2 fgco2 nep siconc tas tos	weekly test case: T255L91-ORCA1L75-LPJG-PISCES-TM5 co2,co2fb	marenostrum
t0n9	t0n9_reference	FAILED	FAILED	cLand nep	weekly test case: OSM-LPJG (res account)	marenostrum
t0na	t0na_reference	Successful	Successful		weekly test case: ORCA1L75-	marenostrum

TS: Results provided

And we can finally **generate reports**, using templates in html format to define what should be included (also can be directly copy-pasted into gitlab markdown). Here is the "simple" report (with all experiments reportes in one table with some predefined variables):

ID	DESCRIPTION	HPCARCH	LEGS (MONTHS)	RES	#PROC	SYPD	ASYPD
t06a	ORCA1L75 + nudging + surface restoring	marenostrum4	2 (12)	LR	480	33.06	15.83
t06b	T255L91	marenostrum4	1 (12)	LR	528	22.21	13.6
t06g	T255L91-ORCA1L75. ScenarioMIP/ssp245 outclass	marenostrum4	2 (1)	LR	768	15.38	3.27
t06h	T255L91-TM5 full chem	marenostrum4	2 (1)	LR	96	1.52	1.42
t06i	T255L91-ORCA1L75-TM5 full chem	marenostrum4	2 (1)	LR	192	1.4	0.94
t06j	T255L91-ORCA1L75-LPJG-PISCES	marenostrum4	3 (12)	LR	768	12.74	1.33
t06k	T255L91-ORCA1L75-LPJG-PISCES-TM5 co2,co2fb	marenostrum4	3 (12)	LR	528	7.5	1.74
t06l	OSM-LPJG	marenostrum4	2 (12)	LR	144	100.41	37.62
t06p	ORCA1L75-PISCES	marenostrum4	2 (1)	LR	432	10.87	8.53
t06s	ORCA1L75 CMORIZATION FALSE	marenostrum4	2 (1)	LR	480	22.22	19.43
t06u	T255L91-ORCA1L75	ecmwf-xc40	2 (1)	LR	504	6.9	6.33
t07b	ORCA1L75	marenostrum4	2 (1)	LR	480	19.86	18.39
t07h	T255L91-ORCA1L75 cold start	marenostrum4	2 (1)	LR	768	18.03	4.3

TS: Results provided

Here is the **default template report** (generates separated tables for each experiment with some predefined variables):

t06a - ORCA1L75 + nudging + surface restoring					
VERSION	CHUNKS	MODEL_RES	PRODUCTION_EXP	NUMPROC	STARTDATES
trunk	2 (12 month)	LR	FALSE	480	19930101

OUTCLASS	reduced

COMPONENTS	PROCS	INI
LIM3 NEMO	- 336	a2mq a2mq

SYPD	ASYPD	RSYPD	JPSY	CHSY
33.0591	15.8285	13.1547	15430000.0	348.465

TS: Results provided

And last the **custom template** (generates the customized tables and variables for each experiment):

t0bd - T255L91-ORCA1L75-LPJG-PISCES-TM5 co2,co2fb - SD=18500101 - NFXYR=1850 - COMPILATION/CHUNKSIZE=12 /ECE3POSTPROC=TRUE

VERSION	CHUNKS	MODEL_RES	PRODUCTION_EXP	NUMPROC	STARTDATES
v3.3.4	3 (12 month)	LR	TRUE	528	18500101

OUTCLASS	CMIP6/CMIP/EC-EARTH-CC/cmip6-experiment-CMIP-esm-piControl

COMPONENTS	PROCS	INI
IFS	256	a2t4
LIM3	-	a2t4
LPJG	8	a2t4
NEMO	192	a2t4
PISCES	-	a2t4
TM5:CO2	4	-
XIOS	2	-

SAVE_IC	OFFSET	CONDITION
end_leg		true

ATM NUDGING	REFERENCE	OCE NUDGING	REFERENCE	SURFACE RESTORING	REFERENCE	MASK	MEMBER
FALSE		FALSE		FALSE	s4_surfresto	DEFAULT	

SYPD	ASYPD	RSYPD	JPSY	CHSY
8.1558	2.3139	3.9567	0	1553.7367

TS: Validated releases

From some time ago, we started running our testing suite protocol to also validate the EC-Earth releases, with the following ones being the ones finished:

- 3.3.3.2 (#991)
- 3.3.4 (#1075 & #1081, where an issue with TM5 compilation was found and solved before the release)
- 3.3.4.2 maintenance branch release (#1206)
- 3.4 (#1205)

And we expect to continue validating the future releases (3.5 coming up next).

TS: EC-Earth 4 plans

We expect also to use the TS tool in EC-Earth 4.
For the moment, we already have a small initial auto-EC-Earth 4 workflow, using Autosubmit 4, with the experiments used there being able to run from the TS tool.
But this is an initial setting and there isn't that much of a wide variety of experiments yet.

Earth Sciences Department

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EC EARTH

Questions?

Eric Ferrer, Gilbert Montané,
Genís Bonet, Rohan Ahmed

eric.ferrer@bsc.es