



**Barcelona  
Supercomputing  
Center**  
*Centro Nacional de Supercomputación*



EXCELENCIA  
SEVERO  
OCHOA

# Autosubmit Tutorial 2021

Hands-on: How to run  
experiments using Autosubmit

The Autosubmit Team

# Autosubmit Introduction



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# Autosubmit - Summary

- **Automatization:** Orchestrating different kind of tasks in homogeneous or heterogeneous environments. No user intervention needed.
- **Provenance and reproducibility:** Unique identifier per experiment, storing all the parameters needed to reproduce it (Autosubmit version, model version, configuration, etc.). Linked with CVS.
- **Failure tolerance:** Automatic retrials and ability to rerun chunks in case of corrupted or missing data, repeating postprocessing and transfers if needed. Recovery capabilities.
- **Versatility:** Different workflows including Auto-Models (EC-Earth, MONARCH, CALIOPE), data downloading (Auto-MARS), machine learning, performance analysis... Providing specialized features for each case (different kind of wrappers, using MPI machine files or masks to handle resources affinity).

# Autosubmit basic usage

## How to create a dummy experiment



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# Before start - Initialization

- **Ensure that you have password less ssh to esarchive.**
  - You can generate a ssh keygen by `ssh-keygen -t rsa -b 4096 -C "email@bsc.es" -m PEM`
  - Then you will need to perform an `scp ~/.ssh/id_rsa.pub user@bscautosubmit02.bsc.es`
  - Repeat the same for **marenostrum4** if not already done.
  - ForwardX11 yes
- **Load autosubmit in two terminals (follow the colors to know where put each**

```
# Terminal 1 (workstation)
```

- `ssh -XY workstation`
- `module load autosubmit`

```
# Terminal 2 (AS machine)
```

- 1) `ssh bscautosubmit02.bsc.es`
- 2) `module load autosubmit`

```
# Terminal 3 (workstation)
```

- Edit files there.

# Basic Workflow - Initialization

- **Create a new experiment** -> Automatically stores fundamental info in a database.

Terminal 1

```
autosubmit expid -dm -H "marenostrom4" -d "AS-Tutorial 2021"
```

```
The new experiment "<expid>" has been registered.
```

- **Creates the folder structure with the basic configuration.**

```
/esarchive/autosubmit/<expid>/conf -> Config files
```

```
/esarchive/autosubmit/<expid>/pk1 -> Workflow
```

```
/esarchive/autosubmit/<expid>/plot -> Visualization
```

```
/esarchive/autosubmit/<expid>/tmp -> Logs, templates
```

# Basic Workflow - Configuration

## conf/expdef

- Default platform (-H).
- Start dates, members and chunks.
- Experiment project source (git, local, svn, dummy).
- Project configuration file path.

## conf/platforms

- Manage cluster, Fat-nodes and Support computers.
- Multiple computers are allowed (even unused).

# Basic Workflow - Configuration(cont.)

## conf/jobs

- Scripts to execute.
- Dependencies between jobs.
- Job requirements.
- Platform and queue to use.

## conf/autosubmit

- Total jobs limitation ( waiting, total).
- Version info, Retrials, Mail notification and storage systems.
- Wrappers, presubmission and migrate.



# Basic Workflow - Create

- Create an experiment using terminal 1 (workstation)

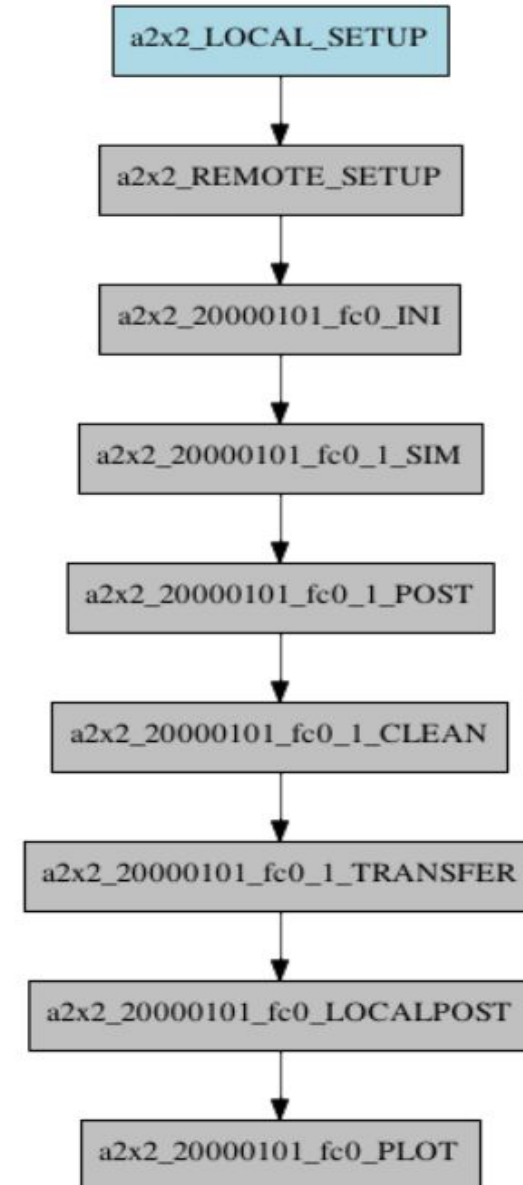
```
autosubmit create <expid>
```

```
#Each one will have a different expid
```

```
Generates plot/expid_timestamp_hour.pdf =>
```

```
Generates tmp/Log_<expid> <= template logs
```

```
Generates tmp/ASLOG <= command logs
```



# Basic Workflow - Config

**conf/platform.conf**

```
[marenostrum4]
TYPE = slurm #scheduler
HOST = mn1.bsc.es #ip or alias
PROJECT = bsc32
USER = bsc32XXX # Your username
SCRATCH_DIR = /gpfs/scratch
ADD_PROJECT_TO_HOST = false
QUEUE = debug
```

open

**conf/expdef.conf**

```
[DEFAULT]
HPCARCH = marenostrum4

[experiment]
DATELIST = 20000101
MEMBERS = fc0
CHUNKSIZEUNIT = month
CHUNKSIZE = 4
MEMBERS = fc0
NUMCHUNKS = 1
CHUNKINI =

[project]
# Select project type. STRING = git,
svn, local, none
PROJECT_TYPE = none
PROJECT_DESTINATION = proj
```

edit

# Basic Workflow - Run

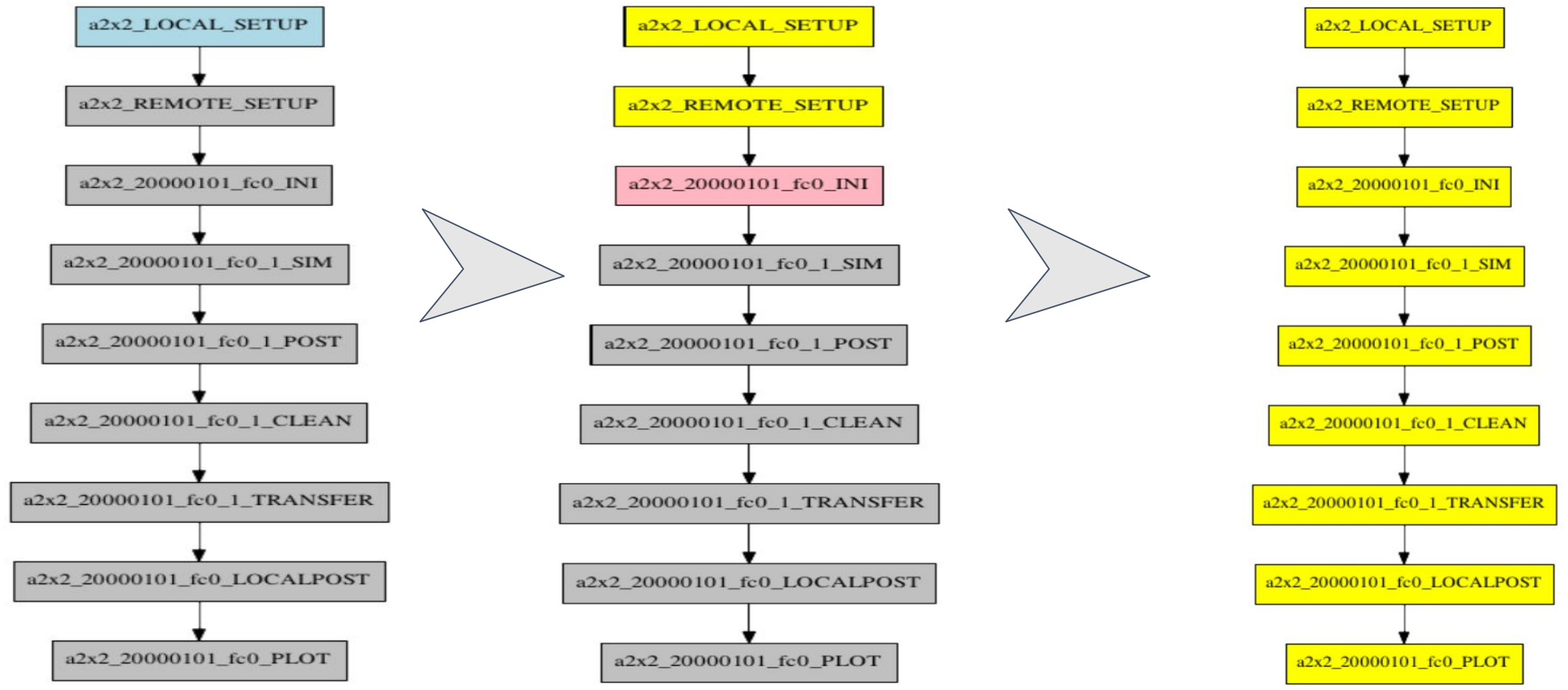
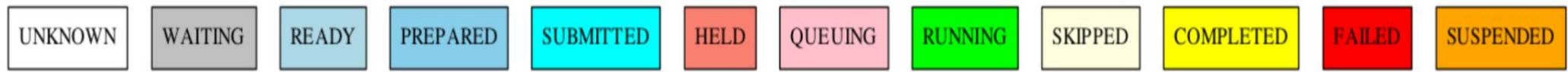
```
autosubmit run <expid>
```

- Generates `tmp/Log_<expid>` `<= template logs` (per each completed job)
- Generates `tmp/ASLOG` `<= command log`
- Runs the experiment.

Run log:

```
[marenostrom4] Connection successful to host mn1.bsc.es  
[marenostrom4] Correct user privileges for host mn1.bsc.es  
[local] Connection successful to host localhost  
[local] Correct user privileges for host localhost
```

# Basic Workflow - Monitor



autosubmit monitor <expid>

# Basic workflow - Common commands

## **autosubmit refresh**

Update project experiment folder.

## **autosubmit inspect**

Generates a preview of the job scripts combined with AS settings (templates).

## **autosubmit recovery**

Synchronizes the status of the remote platform with the local platform.

## **autosubmit setstatus**

Allows to change the status of the workflow.

# Practical Example: Running MONARCH with Autosubmit



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# Monarch - Creation of a new experiment

```
dbeltran@bscesautosubmit01:~$ autosubmit expid --copy t0lk -H marenostrum4 -d "Test for MONARCH & Autosubmit training - DUST mode"
```

The new experiment "**<expid>**" has been registered.  
Copying previous experiment config directories  
Experiment registered successfully

The experiment folder will be created in: `/esarchive/autosubmit`

```
dbeltran@bscesautosubmit01:~$ cd /esarchive/autosubmit/<expid>
```

# Monarch - Platforms.conf

Configure the main HPC platform, and the transfer machine.

Edit the platforms configuration file: `/esarchive/autosubmit/<expid>/conf/platforms_<expid>.conf`

## [marenostrom4]

```
TYPE = slurm
HOST = mn1.bsc.es
PROJECT = bsc32
USER = bsc32NNN
QUEUE = debug
SCRATCH_DIR = /gpfs/scratch
ADD_PROJECT_TO_HOST = false
PROCESSORS_PER_NODE = 48
MAX_PROCESSORS = 768
MAX_WALLCLOCK = 48:00
#DISABLE_RECOVERY_THREADS = FALSE
```

## [marenostrom\_archive]

```
TYPE = ps
HOST = dt02.bsc.es
PROJECT = bsc32
USER = bsc32NNN
SCRATCH_DIR = /gpfs/scratch
ADD_PROJECT_TO_HOST = False
TEST_SUITE = False
```

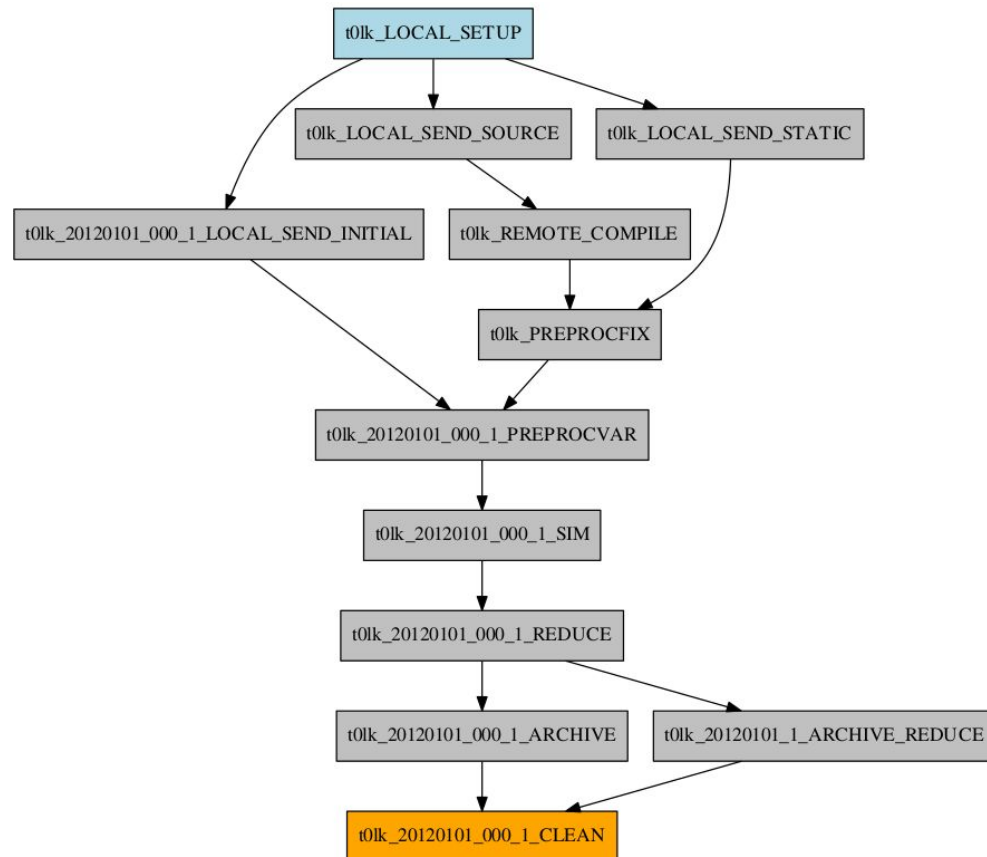
Swap User `bsc32NNN` for your marenostrom user.



# MONARCH - Create Workflow

```
autosubmit create <expid> -np
```

```
autosubmit setstatus <expid> -t SUSPENDED -ft CLEAN -s
```



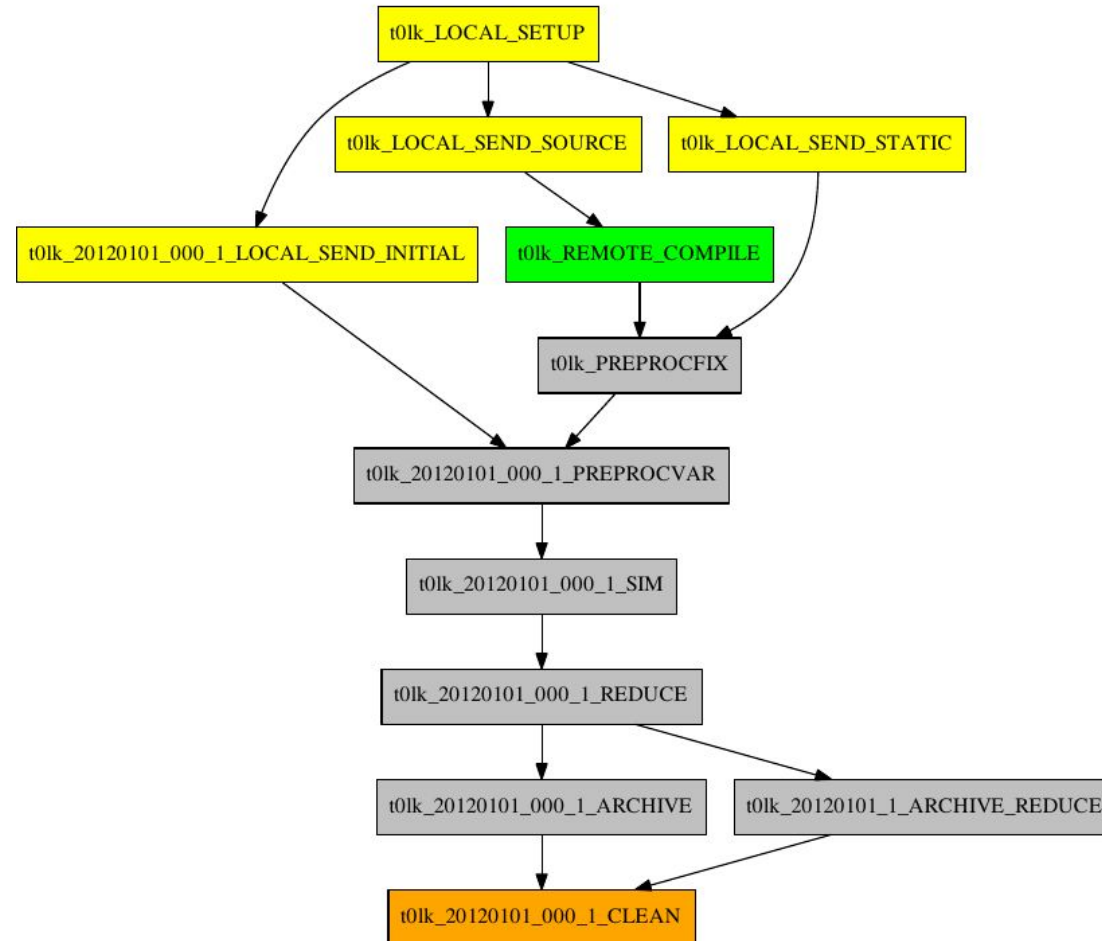
# Monarch - Run and monitor

autosubmit run <expid>

autosubmit monitor <expid>



[marenostrom4] Connection successful to host mn1.bsc.es  
[marenostrom4] Correct user privileges for host mn1.bsc.es  
[local] Connection successful to host localhost  
[local] Correct user privileges for host localhost  
[marenostrom\_archive] Connection successful to host dt02.bsc.es  
[marenostrom\_archive] Correct user privileges for host dt02.bsc.es



# Monarch - Jobs.<expid>.conf

## [LOCAL\_SETUP]

```
FILE = templates/local_setup.sh
PLATFORM = LOCAL
RUNNING = once
NOTIFY_ON = FAILED
```

## [LOCAL\_SEND\_INITIAL]

```
FILE = templates/00_local_send_initial.sh
PLATFORM = marenostrom_archive
DEPENDENCIES = LOCAL_SETUP
#because different dates (chunk = day):
RUNNING = chunk
NOTIFY_ON = FAILED
```

## [LOCAL\_SEND\_SOURCE]

```
FILE = templates/01_local_send_source.sh
PLATFORM = marenostrom_archive
DEPENDENCIES = LOCAL_SETUP
RUNNING = once
NOTIFY_ON = FAILED
```

## [LOCAL\_SEND\_STATIC]

```
FILE = templates/01b_local_send_static.sh
PLATFORM = marenostrom_archive
DEPENDENCIES = LOCAL_SETUP
RUNNING = once
NOTIFY_ON = FAILED
```

# Monarch - Jobs.conf

## [REMOTE\_COMPILE]

```
FILE = templates/02_compile.sh
DEPENDENCIES = LOCAL_SEND_SOURCE
RUNNING = once
PROCESSORS = 4
WALLCLOCK = 00:50
CUSTOM_DIRECTIVES = ["#SBATCH -p interactive"]
NOTIFY_ON = FAILED
```

## [PREPROCFIX]

```
FILE = templates/03_preproc_fix.sh
DEPENDENCIES = REMOTE_COMPILE LOCAL_SEND_STATIC
RUNNING = once
WALLCLOCK = 00:30
NOTIFY_ON = FAILED
```

## [PREPROCVAR]

```
FILE = templates/04_preproc_var.sh
DEPENDENCIES = LOCAL_SEND_INITIAL PREPROCFIX
#In Autosubmit nomenclature chunk will be our days
#(because days is used for start days)
RUNNING = chunk
PROCESSORS = 8
WALLCLOCK = 00:30
NOTIFY_ON = FAILED
```

## [SIM]

```
FILE = templates/05b_sim.sh
DEPENDENCIES = PREPROCVAR SIM-1
RUNNING = chunk
PROCESSORS = 284
WALLCLOCK = 00:10
NOTIFY_ON = FAILED
```

# Monarch - Jobs.conf

## [ARCHIVE]

```
FILE = templates/06_archive.sh
DEPENDENCIES = REDUCE ARCHIVE-1
PLATFORM = marenostrom_archive
RUNNING = chunk
WALLCLOCK = 00:10
NOTIFY_ON = FAILED
```

## [REDUCE]

```
FILE = templates/07_reduce.sh
DEPENDENCIES = SIM REDUCE-1
RUNNING = chunk
WALLCLOCK = 01:00
NOTIFY_ON = FAILED
```

## [ARCHIVE\_REDUCE]

```
FILE = templates/06c_archive_reduce.sh
DEPENDENCIES = REDUCE
PLATFORM = marenostrom_archive
RUNNING = chunk
SYNCHRONIZE = member
WALLCLOCK = 01:00
NOTIFY_ON = FAILED
```

## [CLEAN]

```
FILE = templates/08_clean.sh
DEPENDENCIES = REDUCE+1 ARCHIVE ARCHIVE_REDUCE
PLATFORM = marenostrom_archive
RUNNING = chunk
WALLCLOCK = 00:10
NOTIFY_ON = FAILED
```

# Monarch - project folder

Inside the **proj/auto-monarch** directory we can find the scripts, configuration files and source code to run the model:

```
$ ls -l /esarchive/autosubmit/<expid>/proj/auto-monarch
drwxr-xr-x 2 dbeltran Earth 4096 Nov  6 18:08 plugins
drwxr-xr-x 5 dbeltran Earth 4096 Nov  6 18:08 sources
drwxr-xr-x 4 dbeltran Earth 4096 Nov  6 18:08 templates
drwxr-xr-x 4 dbeltran Earth 4096 Nov  6 18:08 utils
```

```
$ ls -l /esarchive/autosubmit/<expid>/proj/auto-monarch/sources
drwxr-xr-x 3 dbeltran Earth 4096 Nov  6 18:08 monarch-DA
drwxr-xr-x 2 dbeltran Earth 4096 Nov 13 13:50 MONARCH_Reduce
drwxr-xr-x 5 dbeltran Earth 4096 Nov  6 18:08 nmb-monarch
```

# Monarch - proj conf and folder

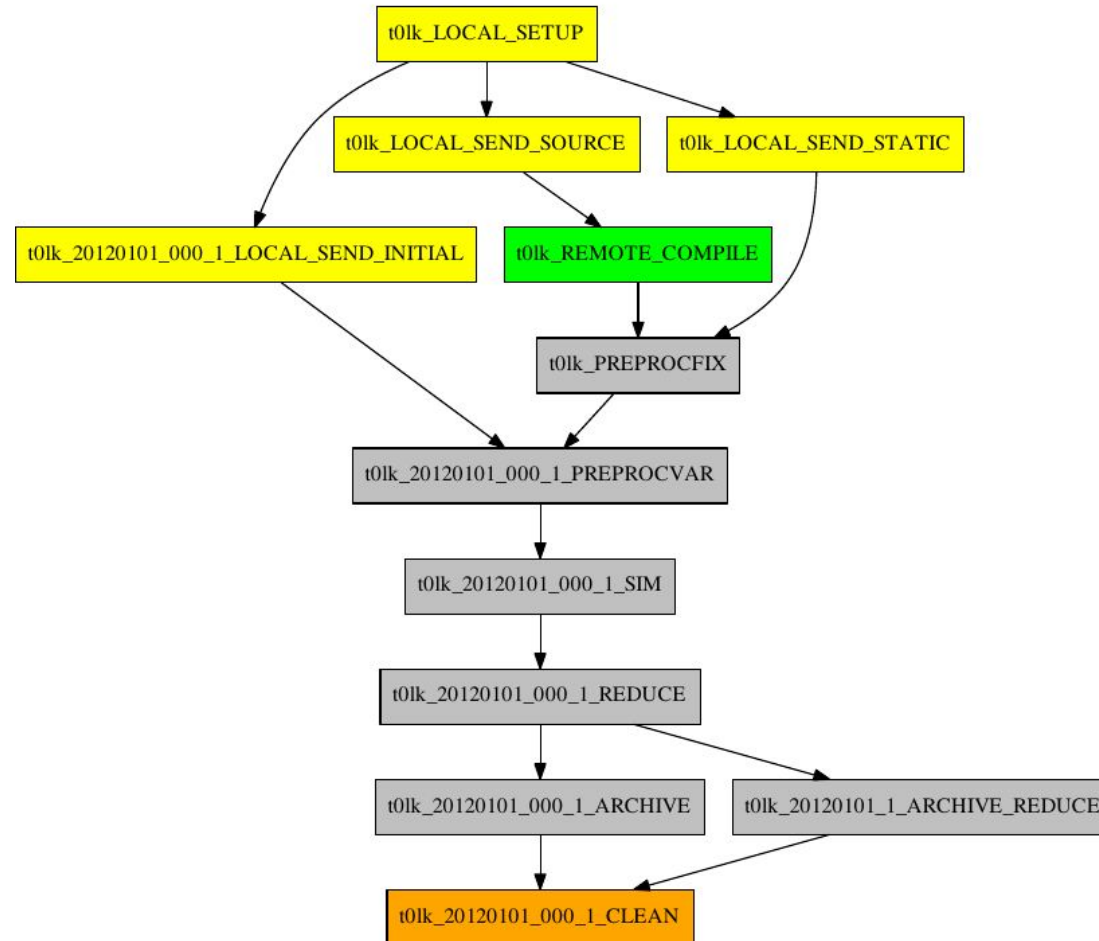
Inside the **proj/auto-monarch** directory we can find the scripts, configuration files and source code to run the model:

```
$ ls -l /esarchive/autosubmit/<expid>/proj/auto-monarch/templates
-rw-r--r-- 1 dbeltran Earth 5.1K Nov  6 18:08 clean.sh
-rw-r--r-- 1 dbeltran Earth 12K Nov  6 18:08 compile.sh
drwxr-xr-x 2 dbeltran Earth 4.0K Nov  6 18:08 conf
-rw-r--r-- 1 dbeltran Earth 2.3K Nov  6 18:08 cp_compiled.sh
-rw-r--r-- 1 dbeltran Earth 6.2K Nov  6 18:08 da.sh
-rw-r--r-- 1 dbeltran Earth 1.2K Nov  6 18:08 get_out_files.sh
-rw-r--r-- 1 dbeltran Earth 1.5K Nov  6 18:08 local_send_initial_DA.sh
-rw-r--r-- 1 dbeltran Earth 2.9K Nov  6 18:08 local_send_source.sh
-rw-r--r-- 1 dbeltran Earth 4.6K Nov  6 18:08 local_send_static.sh
-rw-r--r-- 1 dbeltran Earth 5.5K Nov  6 18:08 reduce.sh
drwxr-xr-x 2 dbeltran Earth 4.0K Nov 13 16:15 remotesetup
-rw-r--r-- 1 dbeltran Earth 41K Nov  6 18:08 sim.sh
```

open file **conf/proj\_XXXX.conf**. You will find there several MONARCH configuration parameters and values.

# Monarch - Monitor

autosubmit monitor <expid>





# Monarch - Expand Workflow

Edit `expdef_<expid>.conf` and add an higher Number of chunks

**NUMCHUNKS = 4**

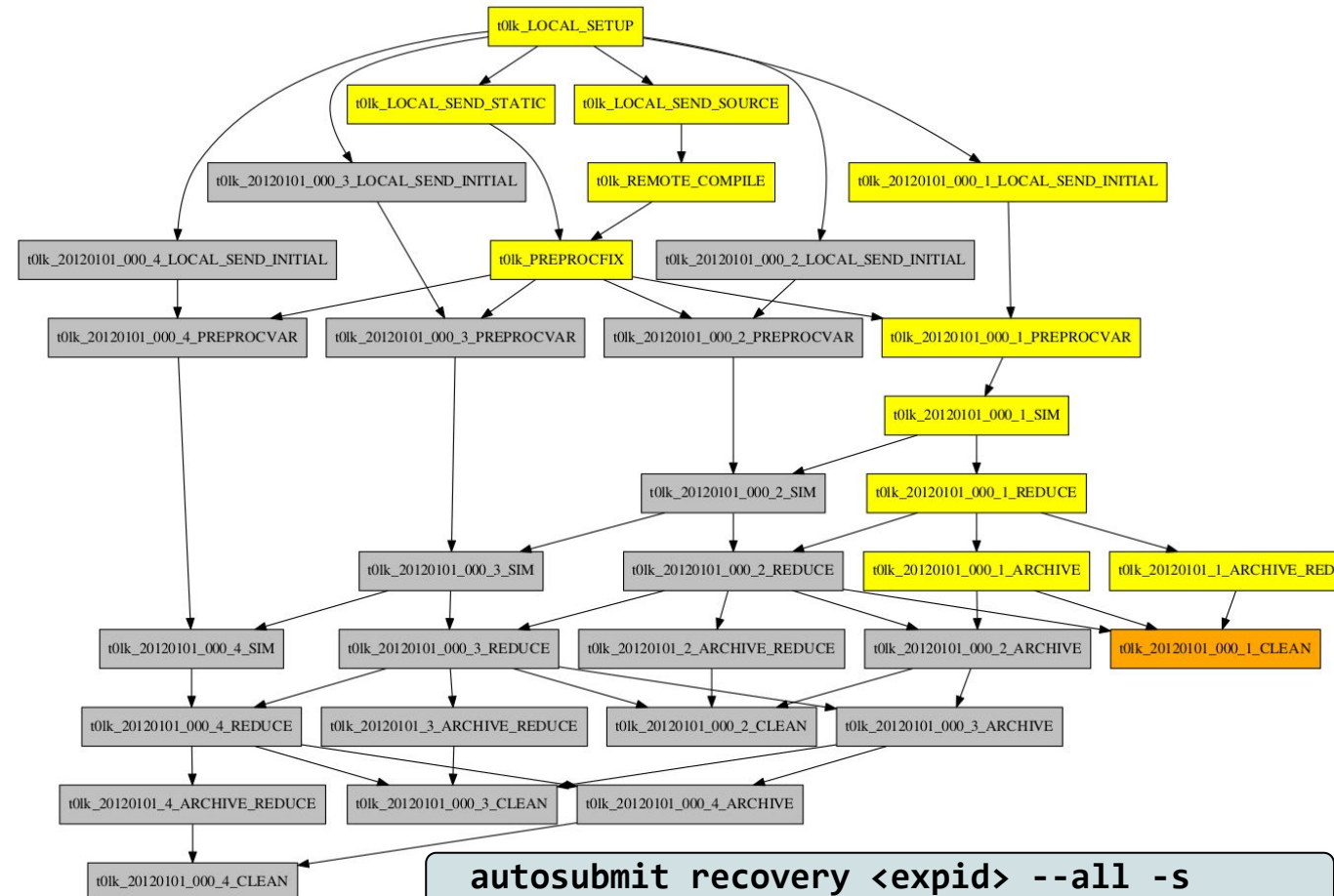
Jobs affected:

[LOCAL\_SEND\_INITIAL]  
[PREPROCVAR]  
[SIM]  
[ARCHIVE]  
[REDUCE]  
[ARCHIVE\_REDUCE]  
[CLEAN]

Resume run with  
**autosubmit run <expid>**

**autosubmit monitor <expid>**

**autosubmit setstatus <expid> -t WAITING -ft CLEAN -s**



# Monarch - Integrate a wrapper

The idea is that any use case has a wrapper to speed up the workflow

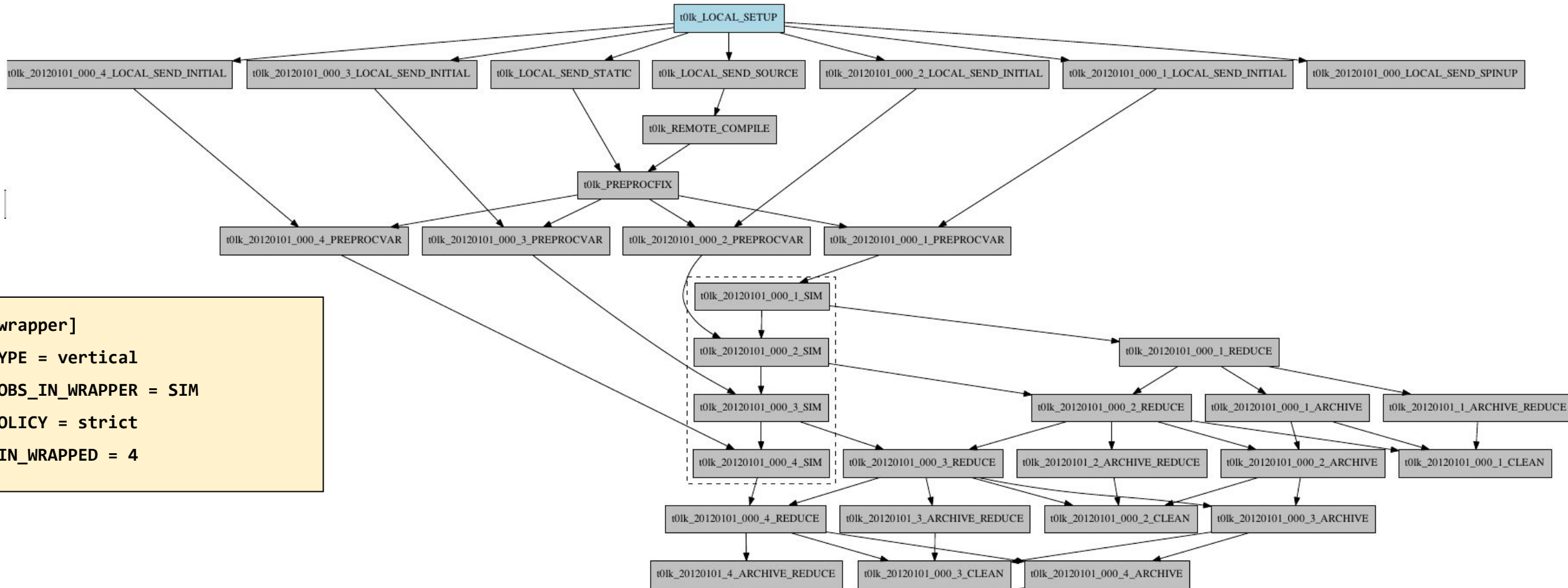
Dimension	Vertical 1D	Vertical > 1D	
Horizontal 1D	-----	<b>Vertical wrapper</b> Speed up sequential chunks	
Horizontal > 1D	<b>Horizontal wrapper</b> Run independent jobs in parallel	<b>Horizontal-vertical</b> Speed up sequential series of parallel independent jobs	<b>Vertical-horizontal</b> Speed up parallel series of sequential chunks

# Monarch - Wrapper

Edit `Autosubmit_<expid>.conf` and add an vertical wrapper ( or just uncomment )

To previsualizate the wrappers without complete them:

```
autosubmit create <expid> -cw  
autosubmit monitor <expid> -cw
```



```
[wrapper]  
TYPE = vertical  
JOBS_IN_WRAPPER = SIM  
POLICY = strict  
MIN_WRAPPED = 4
```

# Monarch - Common commands

## **autosubmit refresh**

Update project experiment folder.

## **autosubmit inspect**

Generates a preview of the job scripts combined with AS settings (templates).

## **autosubmit recovery**

Synchronizes the status of the remote platform with the local platform.

## **autosubmit setstatus**

Allows to change the status of the workflow.

# Contact

<b>Get involved or contact us:</b>	
Autosubmit GitLab:	<a href="https://earth.bsc.es/gitlab/es/autosubmit">https://earth.bsc.es/gitlab/es/autosubmit</a>
Autosubmit Mailing List:	<a href="mailto:autosubmit@bsc.es">autosubmit@bsc.es</a>



**Barcelona  
Supercomputing  
Center**  
Centro Nacional de Supercomputación



# Thank you for your time

[daniel.beltran@bsc.es](mailto:daniel.beltran@bsc.es), [miguel.castrillo@bsc.es](mailto:miguel.castrillo@bsc.es), [wilmer.uruchi@bsc.es](mailto:wilmer.uruchi@bsc.es)

# ISSUE: Clean 1 was ran

Edit proj.conf

COUPLE\_DUST\_INIT=1

SPINUP\_FOLDER= /gpfs/archive/bsc32/esarchive/exp/monarch/<expid>/original\_files