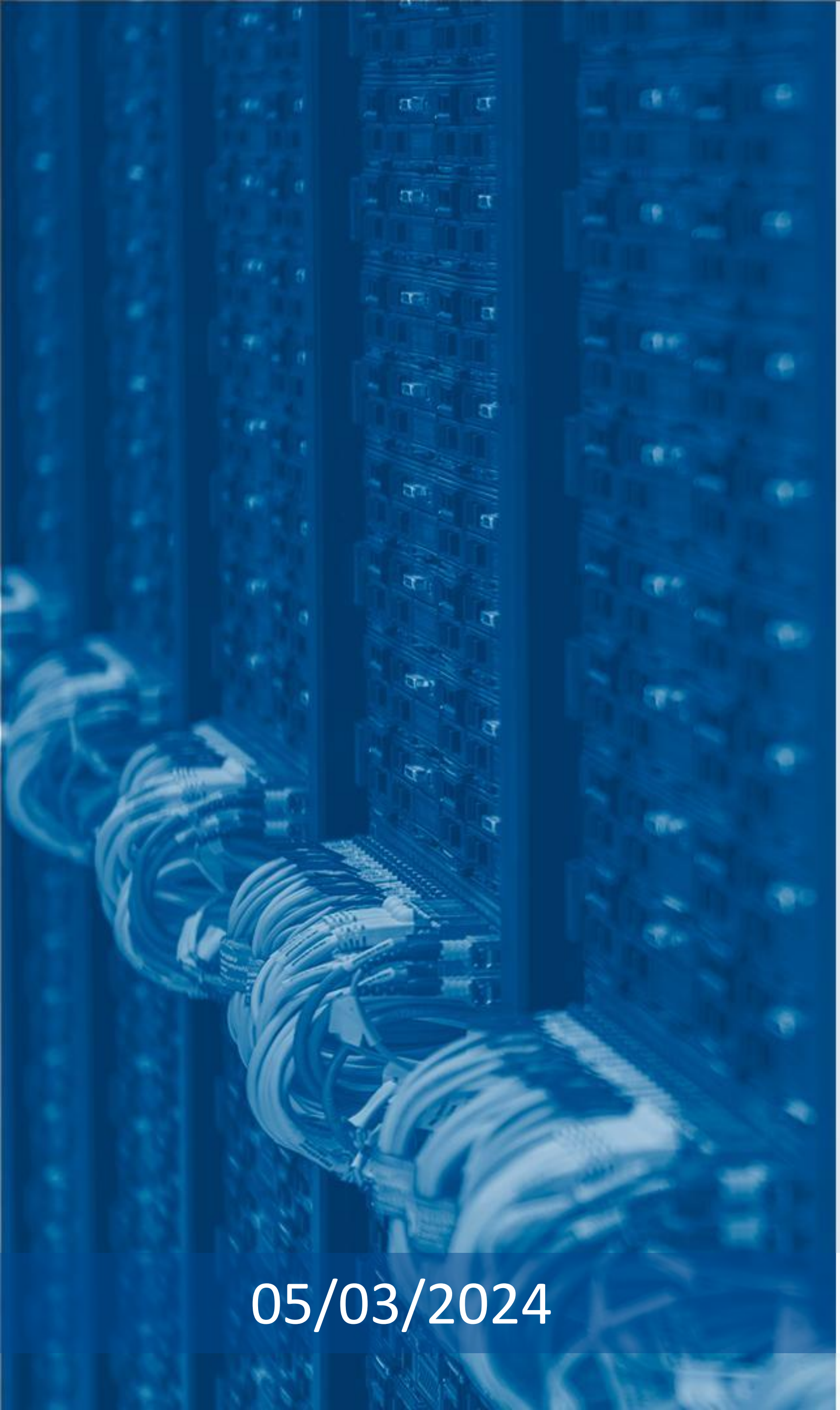


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

SUNSET + Autosubmit: Building multimodel skill assessment workflows

05/03/2024

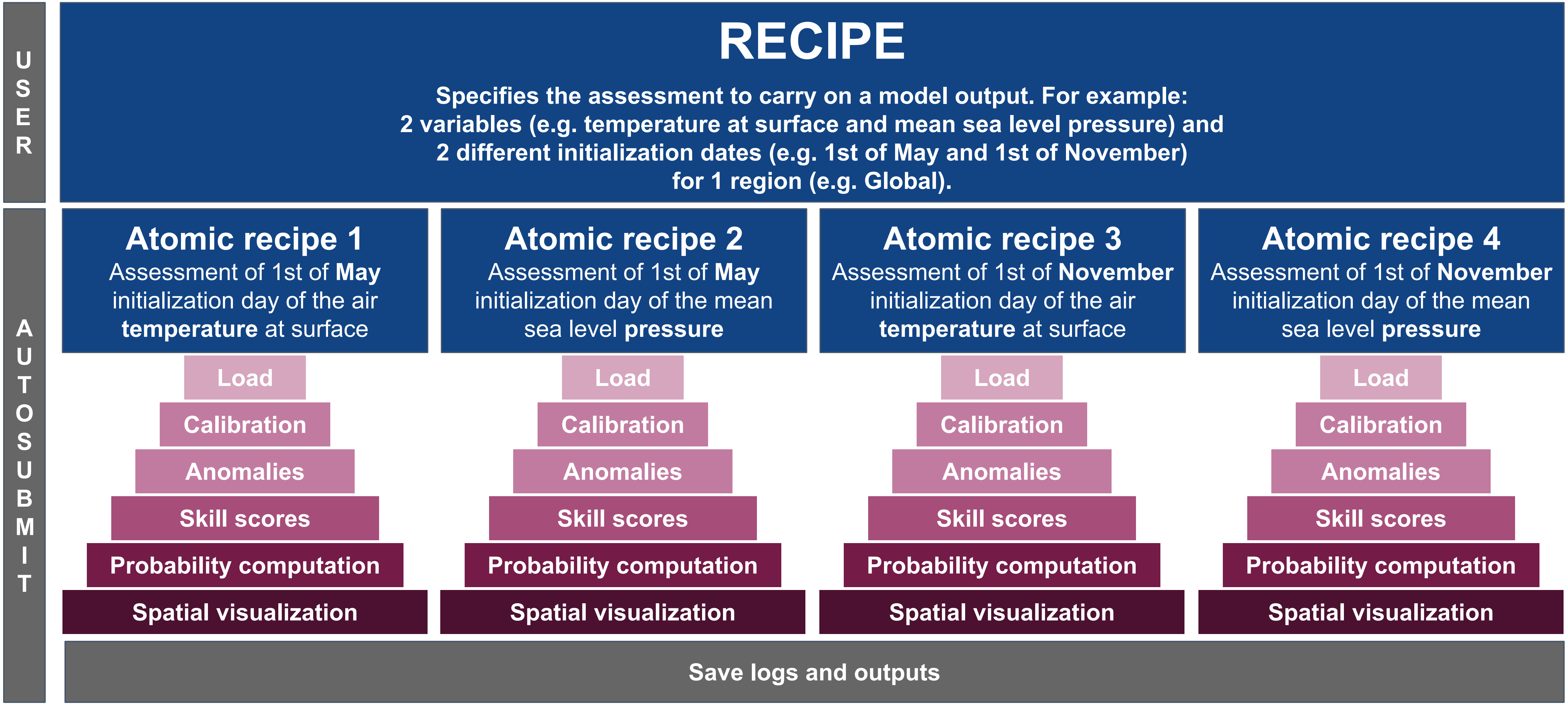
Victòria Agudetse Roures

Introduction

SUNSET is an **R-based tool for climate forecast verification** and post-processing workflows at subseasonal to decadal timescales, developed at BSC-ES taking advantage of the existing ecosystem of in-house R packages for climate data analysis. It is meant to be the go-to code for people in ESS doing this kind of work, and for projects that require it.

Autosubmit is “a **lightweight workflow manager** designed to meet climate research necessities. Unlike other workflow solutions in the domain, it integrates the capabilities of an experiment manager, workflow orchestrator and monitor in a self-contained application. The experiment manager allows for defining and configuring experiments, supported by a hierarchical database that ensures reproducibility and traceability. The orchestrator is designed to run complex workflows in research and operational mode by managing their dependencies and interfacing with local and remote hosts. These multi-scale workflows can involve from a few to thousands of steps and from one to multiple platforms”.

How SUNSET works: atomic recipes for independent verifications



USER

RECIPE

Specifies the assessment to carry on a model output. For example:
2 variables (e.g. temperature at surface and mean sea level pressure) and
2 different initialization dates (e.g. 1st of May and 1st of November)
for 1 region (e.g. Global).

AUTOSUBMIT

Atomic recipe 1

Assessment of 1st of **May** initialization day of the air **temperature** at surface

- Load
- Calibration
- Anomalies
- Skill scores
- Probability computation
- Spatial visualization

Atomic recipe 2

Assessment of 1st of **May** initialization day of the mean sea level **pressure**

- Load
- Calibration
- Anomalies
- Skill scores
- Probability computation
- Spatial visualization

Atomic recipe 3

Assessment of 1st of **November** initialization day of the air **temperature** at surface

- Load
- Calibration
- Anomalies
- Skill scores
- Probability computation
- Spatial visualization

Atomic recipe 4

Assessment of 1st of **November** initialization day of the mean sea level **pressure**

- Load
- Calibration
- Anomalies
- Skill scores
- Probability computation
- Spatial visualization

Save logs and outputs

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Diagram by Núria Pérez-Zanón

How is this translated into Autosubmit?

USER

RECIPE

**Specifies the assessment to carry on a model output. For example:
2 variables (e.g. temperature at surface and mean sea level pressure) and
2 different regions (e.g. 1st of May and 1st of November)
for 1 region (e.g. Europe).**

SUNSET

**The SUNSET launcher splits the recipe into atomic recipes,
modifies the template for the Autosubmit configuration,
and saves the configuration files in the Autosubmit experiment folder.**

AUTOSUBMIT

Atomic recipe 1	Atomic recipe 2	Atomic recipe 3	Atomic recipe 4
Assessment of 1st of **May** initialization day of the air **temperature** at surface	Assessment of 1st of **May** initialization day of the mean sea level **pressure**	Assessment of 1st of **November** initialization day of the air **temperature** at surface	Assessment of 1st of **November** initialization day of the mean sea level **pressure**
Run script auto-verification.sh %CHUNK% = 1	**Run script auto-verification.sh %CHUNK% = 2**	**Run script auto-verification.sh %CHUNK% = 3**	**Run script auto-verification.sh %CHUNK% = 4**

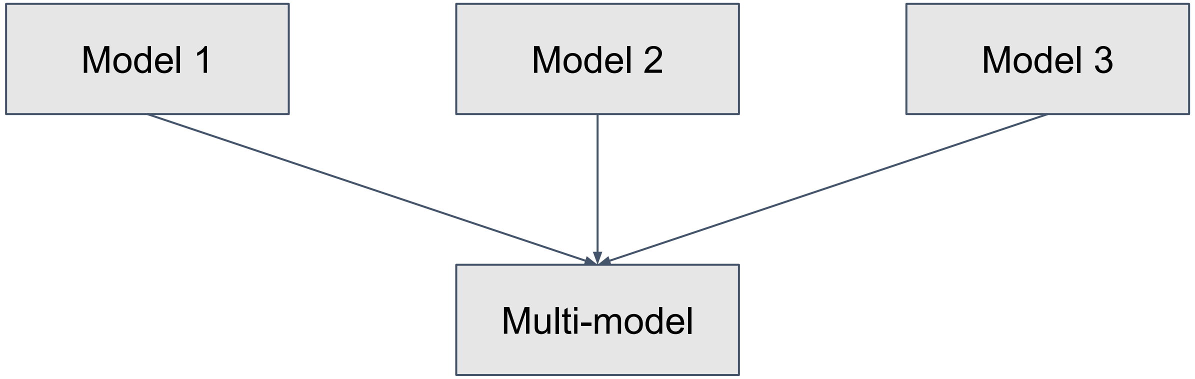
Building a multi-model forecast

There are different methods to build a multi-model ensemble forecast:

- The simplest method, ‘pooling’, consists of directly merging all the ensemble members of the individual models into the same array.
- More complex methods may include selecting certain ensemble members or assigning weights to each according to a certain metric.

In SUNSET, **the models are first processed separately**. After the post-processing, they are loaded together and merged according to the specifications of the user.

```
Datasets:
  System:
    - {name: ECMWF-SEAS5.1}
    - {name: CMCC-SPS3.5}
    - {name: DWD-GCFS2.1}
  Multimodel:
    execute: yes
    approach: pooled
    createFrom: Anomalies
```



Model 1

Model 2

Model 3

Multi-model

How do we make this work in SUNSET?

The concept is simple, but things get complicated when we have a recipe that **also needs to be split along multiple variables, regions, initialization dates…** because in this case there should be multiple multi-model jobs.

Let’s say we want to combine the previous two examples:

2 variables
2 init. dates -> 3*2*2 = 12 ‘chunks’ -> 2*2 = 4 multi-model assessments (3 models each)
3 models

Each multi-model assessment is independent from the others, so to maximize efficiency and convenience, it should only depend on **the necessary chunks**. This configuration will be different for each recipe, and needs to be created on-the-fly. How can we do this?

Getting the dependencies right

The atomic recipes are named according to the parameters that can be split:

```
  atomic_recipe_sys-CMCC-SPS35_ref-ERA5_var-tas_reg-Global_sdate-0101.yml
  atomic_recipe_sys-DWD-GCFS21_ref-ERA5_var-tas_reg-Global_sdate-0101.yml
  atomic_recipe_sys-ECMWF-SEAS5_ref-ERA5_var-tas_reg-Global_sdate-0101.yml
```

The multi-model recipes are named similarly:

```
  atomic_recipe_sys-Multimodel_ref-ERA5_var-tas_reg-Global_sdate-0101.yml
```

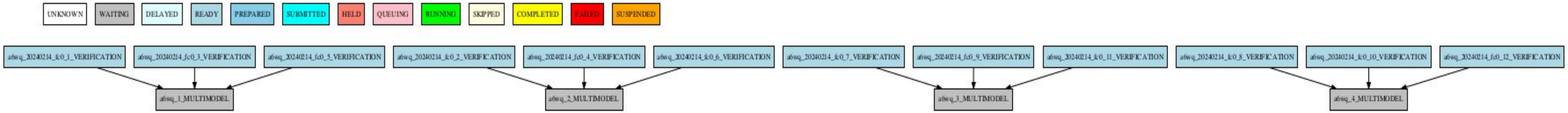
Using the R functions `lapply()` and `grep()`, we cross-reference the names of the recipes and their assigned chunk/split number, and we obtain a list of dependencies.

How is this translated into Autosubmit?

We can use the ‘splits’ feature in Autosubmit to create another independent level of job division. ‘Splits’ can be linked to ‘chunks’.

By writing this association in the ‘jobs’ section of the Autosubmit configuration, each ‘Multimodel’ job will have a dependency on the corresponding single-model jobs.

```
multimodel:
  FILE: autosubmit/auto-multimodel.sh
  (...)
  DEPENDENCIES:
    verification:
      SPLITS_FROM:
        '1':
          CHUNKS_TO: '1,3,5'
        '2':
          CHUNKS_TO: '2,4,6'
        '3':
          CHUNKS_TO: '7,9,11'
        '4':
          CHUNKS_TO: '8,10,12'
  SPLITS: 4
```


UNKNOWN
WAITING
DELAYED
READY
PREPARED
SUBMITTED
HELD
QUEUING
RUNNING
SKIPPED
COMPLETED
FAILED
SUSPENDED
a6wq_20240214_fc0_1_VERIFICATION
a6wq_20240214_fc0_3_VERIFICATION
a6wq_20240214_fc0_5_VERIFICATION
a6wq_20240214_fc0_2_VERIFICATION
a6wq_20240214_fc0_4_VERIFICATION
a6wq_20240214_fc0_6_VERIFICATION
a6wq_20240214_fc0_7_VERIFICATION
a6wq_20240214_fc0_9_VERIFICATION
a6wq_20240214_fc0_11_VERIFICATION
a6wq_20240214_fc0_8_VERIFICATION
a6wq_20240214_fc0_10_VERIFICATION
a6wq_20240214_fc0_12_VERIFICATION
a6wq_1_MULTIMODEL
a6wq_2_MULTIMODEL
a6wq_3_MULTIMODEL
a6wq_4_MULTIMODEL

… But how are the ‘chunks’ and ‘splits’ linked to their corresponding atomic recipe?

Linking it all together

Autosubmit allows us to define access configuration variables from inside the script.

This way, we can tell which ‘SPLIT’ or ‘CHUNK’ is running in any particular instance. The SUNSET launcher creates **two files** with shell commands to establish the correspondence:

- **chunk_to_recipe** for the single-model jobs
- **split_to_recipe** for the multi-model jobs

Example:

```
case $CHUNK in
  1) recipe='sys-ECMWF-SEAS51_ref-ERA5_var-tas_reg-Spain_sdate-0101' ;;
  2) recipe='sys-ECMWF-SEAS51_ref-ERA5_var-prlr_reg-Spain_sdate-0101' ;;
  3) recipe='sys-CMCC-SPS35_ref-ERA5_var-tas_reg-Spain_sdate-0101' ;;
  4) recipe='sys-CMCC-SPS35_ref-ERA5_var-prlr_reg-Spain_sdate-0101' ;;
  (...)
esac
```

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Thanks for your attention!

05/03/2024

Victòria Agudetse Roures