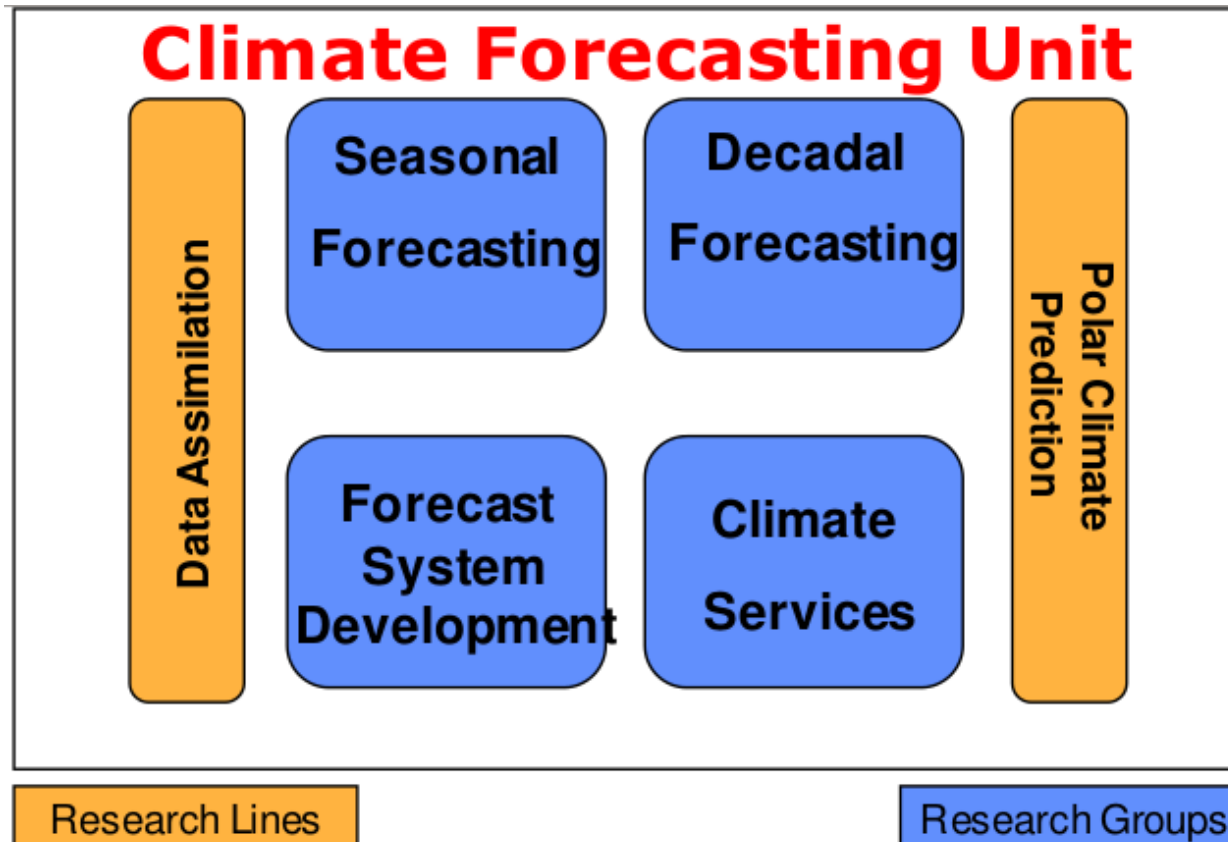


Autosubmit: A tool for managing climate prediction experiments

D. Manubens, M. Asif, O. Mula, F.J. Doblas-Reyes



Outline

- Motivation
- CFU workflow
- Autosubmit
- Future Work (IS-ENES2)
- Future Work (Autosubmit)

Motivation



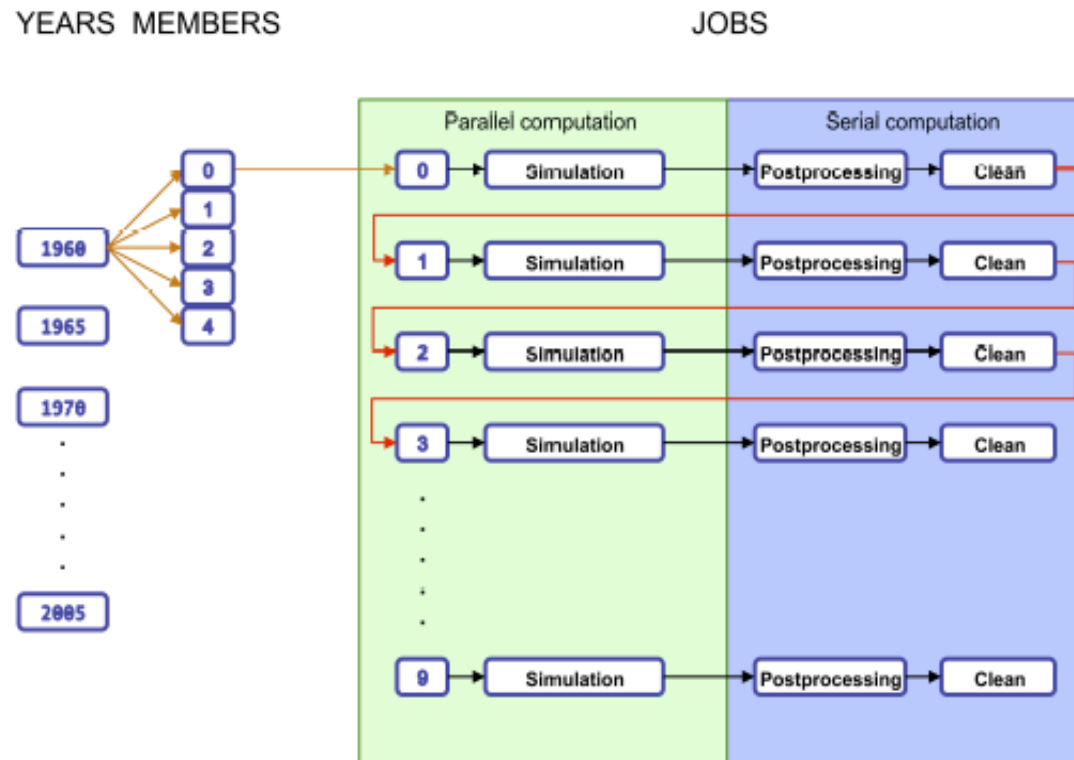
- IS-ENES2 work package 9 (WP9/JRA1) will develop and run a set of coordinated multi-model, multi-member (M4) coupled climate simulations, at high resolution (HR).
- The Task 5 of WP9 in particular aims to test and evaluate common HR simulations, and an assessment of the model deficiencies with regard to their computational performance will be made.
- Two tools to design workflows and monitor experiments are retained for assessment: Cylc and Autosubmit.
- Within WP9 task 5.1, IC3 will analyze the suitability of both options for running M4 HR climate ensembles.

CFU workflow

- Choose platform that is available for performing experiments, and check if enough computing time is available
- Check if the targeted model version has already been deployed
- Check if the SSH access to the platform has been set appropriately
- Autosubmit:
 - Multiple runs producing raw GRIB/NEMO output.
 - Postprocessing, converting raw output to standardized netCDF and computing some diagnostics
 - Transfer the post-processed data to local server
- Use the R package `s2dverification`, if necessary, updating the `Load.R` to reflect new data sets and new variables (diagnostics) available on the file system

A Typical Forecasting Experiment

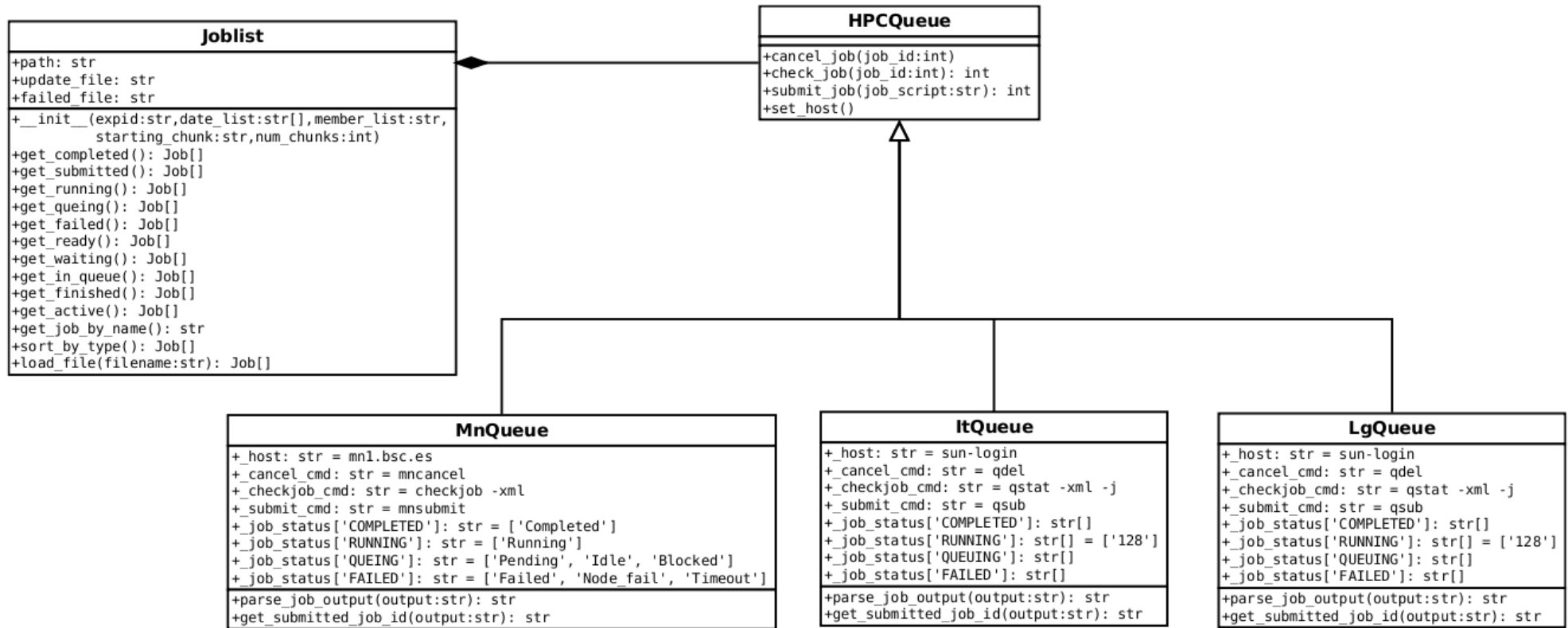
In a typical climate forecast experiment with five members and ten start dates, each start date and member is being run for ten years. Many EC-Earth partners run them using 10 chunks of one year forecast length, with accompanying post-processing and cleaning jobs. The experiment will be made of **50 independent simulations**, each submitting 30 jobs (10 simulations, 10 post-processing and 10 cleaning) with specific dependencies between them.



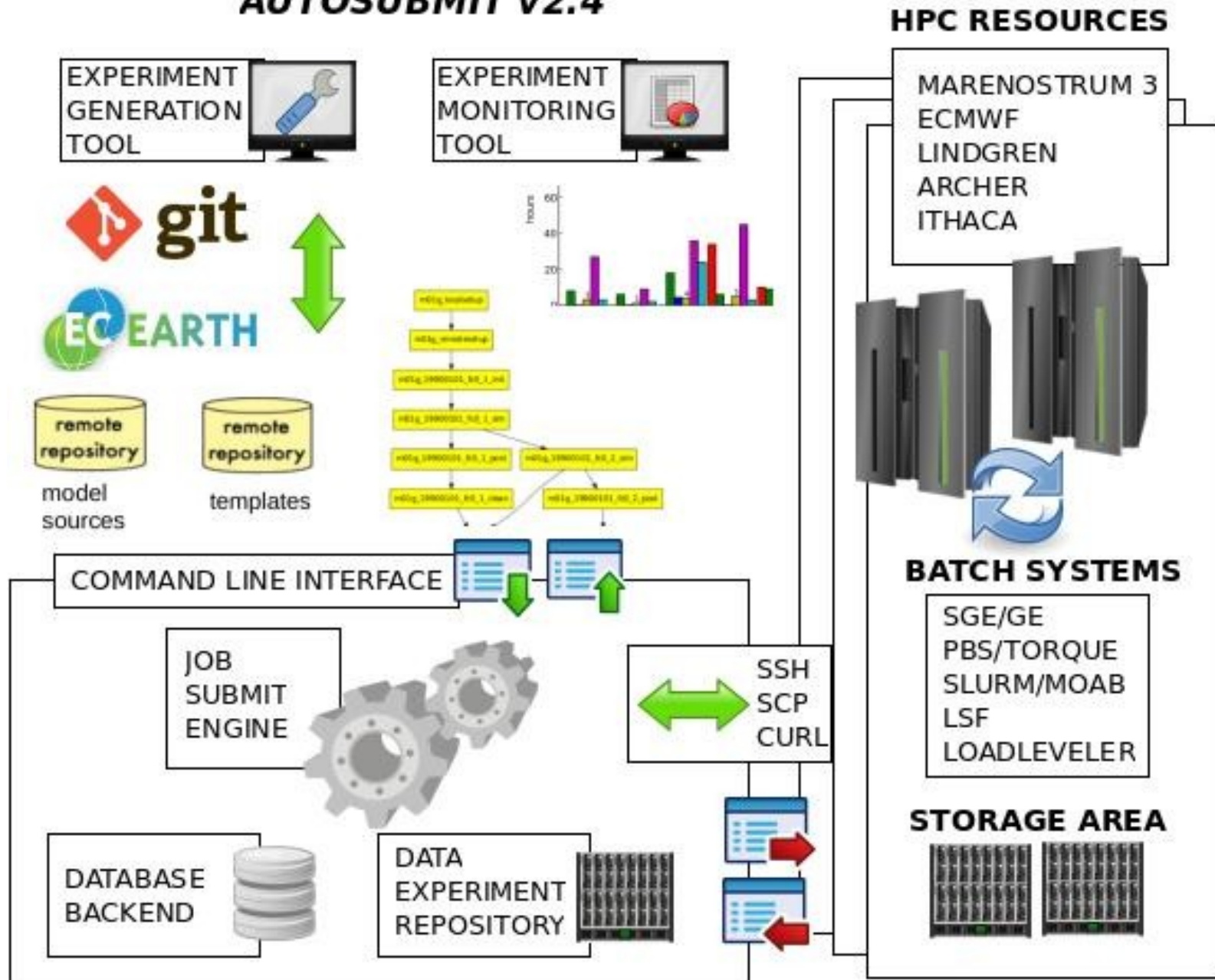
Autosubmit

- Autosubmit is a tool developed at CFU using Python (with object-oriented concepts) to create, manage and monitor experiments.
- Divided in 3 phases: ExpID assign, experiment creation (including access to a GIT repository), run.
- Separation experiment/autosubmit codes.
- Config files for autosubmit and experiment.
- SQLite database for experiment information.
- Common templates for all platforms.
- Fault tolerance, recovery after crashes.
- Automatic run statistics.
- Easily installable on any desktop/server machine with GNU/Linux
- Dealing with a list of schedulers and communication protocols:
 - SGE (Ithaca – IC3)
 - SLURM (MareNostrum - BSC)
 - PBS (HECToR and ARCHER – EPCC, Lindgren – PDC and Jaguar - OLCF)
 - LoadLeveler via ECMWF gateway (c1a and c2a – ECMWF)
 - LSF (MareNostrum 3 - BSC)

Autosubmit (contd.)

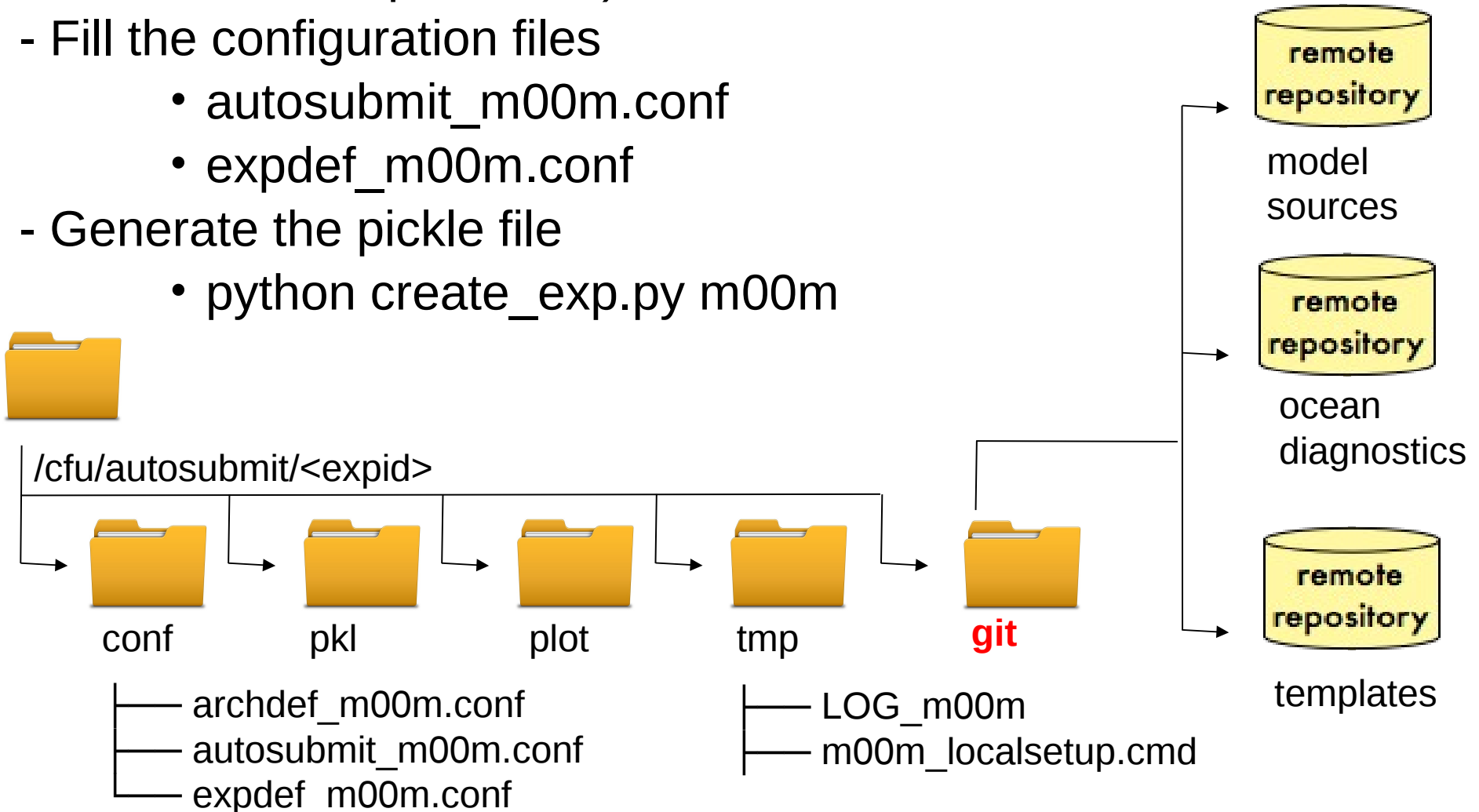


AUTOSUBMIT v2.4

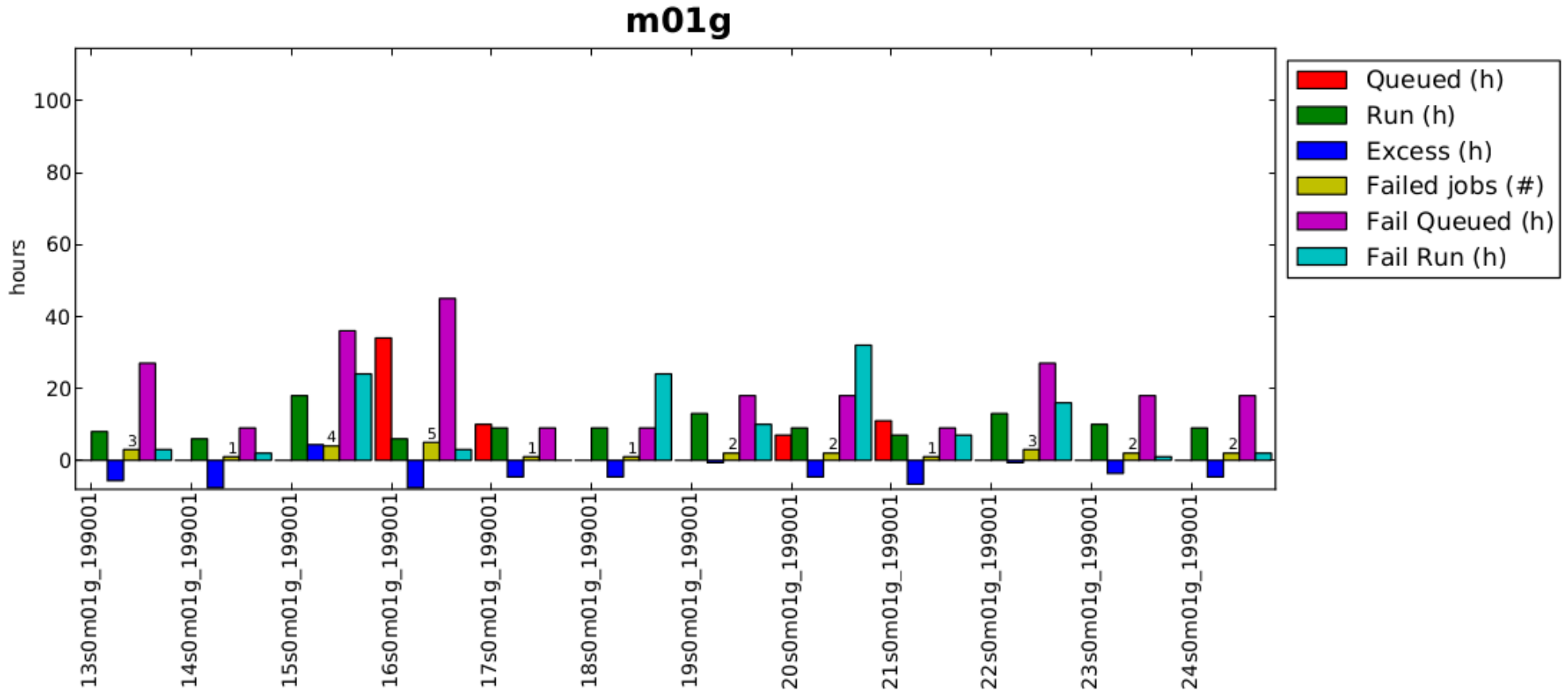


Creating a New Experiment

- Get an ID and create the needed directory structure (including access to GIT repositories)
- Fill the configuration files
 - autosubmit_m00m.conf
 - expdef_m00m.conf
- Generate the pickle file
 - python create_exp.py m00m



Automatic Run Statistics



Future Work (IS-ENES2)

- Autosubmit will be further developed into a general-purpose submission and monitoring tool optimized for M4 HR ensembles.
- Collaboration with CERFACS as part of milestone 9.2 to implement the ARPEGE/NEMO model under Autosubmit.
- A general-purpose wrapper that packs multi-member simulations into a single executable with suitable job control will be tested in WP9 task 5.2.
- Installation package and open source license

Future Work (Autosubmit)

- Integration of HPC's using SAGA (Simple API for Grid Applications)
 - SAGA python comply with OGF (Open Grid Forum) standards (how to interact with the middleware)
 - A number of adaptors are already implemented, to support different grid and cloud computing backends
 - SAGA provides units to compose high-level functionality across distinct distributed systems (e.g. submit jobs from same experiment to different platforms)
- Documenting experiments on simplified METAFOR standards by using relational databases (MySQL)
- Designing a web front-end for experiment creation and monitoring

