

Autosubmit: And end-to-end workflow manager

19th Workshop on High Performance Computing in Meteorology

Wilmer Uruchi 21/09/2021



Platforms

Different architectures.

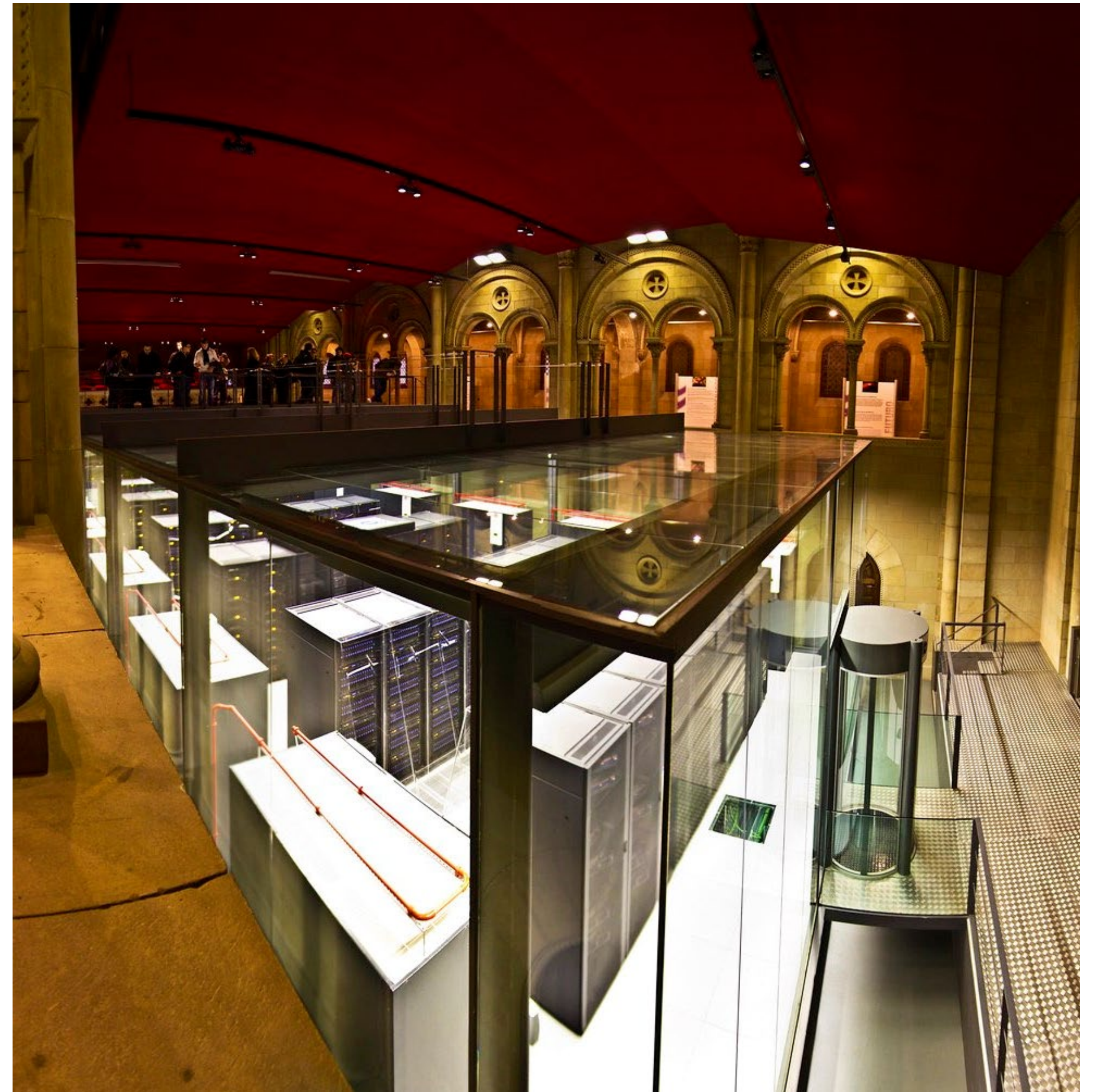
Different locations.

Different scale.

Different schedulers.

Different computational resources assignments for different users.

We can make it work under a single framework: **Autosubmit**.



Marenostrum4

Autosubmit

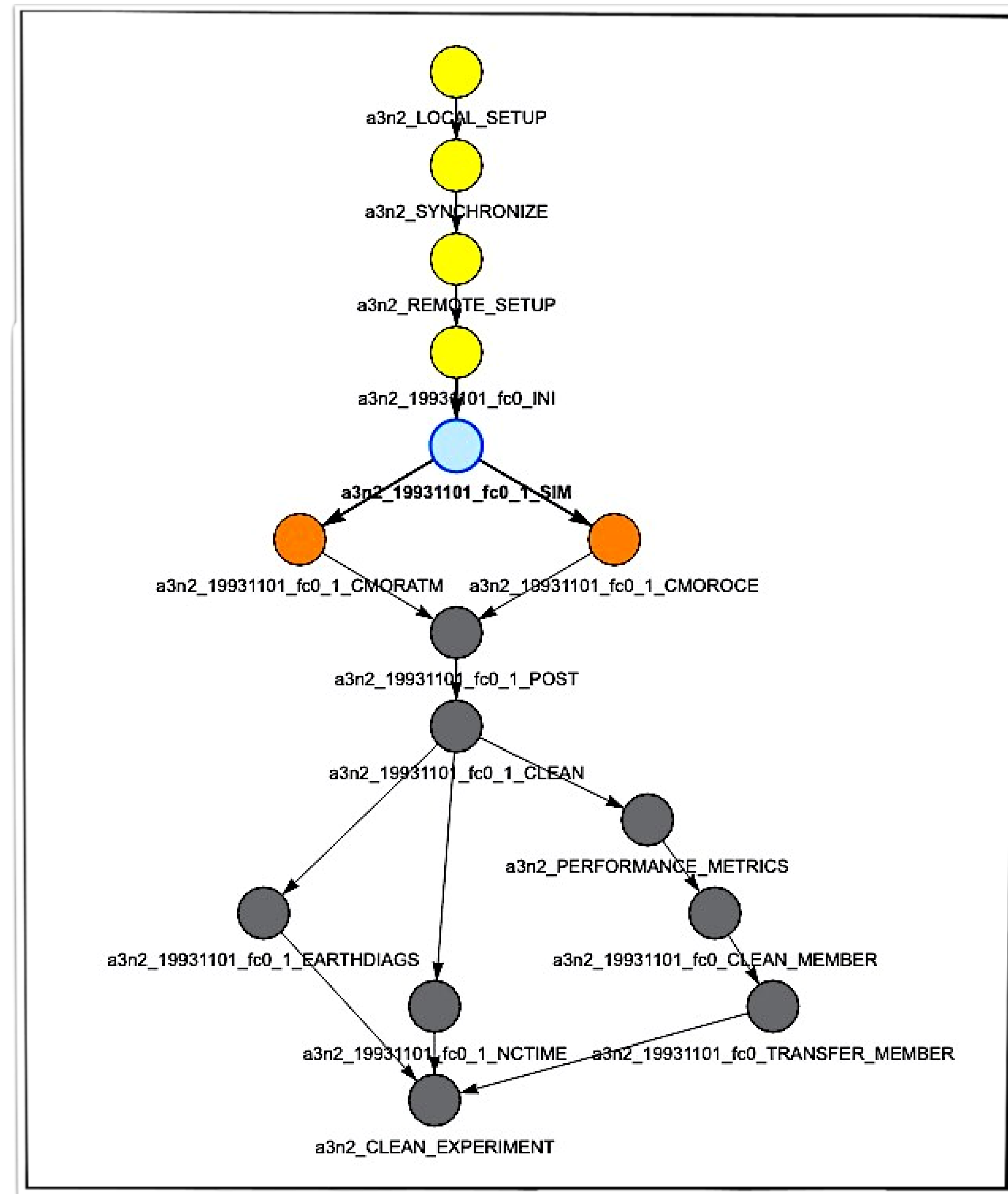
A Brief Description

- Autosubmit is a **workflow manager** that orchestrates complex tasks, mainly scientific. It manages experiments running on HPC clusters, other remote and local platforms. It is written in Python and works on the terminal.
- The user configures the experiment workflow by defining jobs and setting dependencies between these jobs. Then, **Autosubmit** takes control and executes the experiment **automatically**.
- **Autosubmit** connects to the remote platforms by ssh to run scripts, submit batch jobs, retrieve logs, and other tasks that are part of managing a workflow.
- It handles **errors** and can report them to the user. It provides a way to **stop, restart, monitor, modify** the experiment; among other features.

There is no typical experiment

- **Different** configurations for different models. Each has its own set of job types, and each job type allows further configuration. A **general approach** adaptable to **any workflow**.
- Simulations, data transfers, cleaning, preprocessing. The result of any of these jobs can be the input of the next job.
- Our users can take advantage of **clusters** that implement different **architectures** and use them in the **same experiment**.
- An experiment takes the form of an **acyclic directed graph**.

A small experiment



The typical constraints

Under the SLURM model

- A share of computational resources is assigned to a **group of users**. Some groups can have an assignment corresponding to a large project with a **large share**, while others be assigned limited resources for a **small project**.
- Our main platform, **Marenostrum4**, implements the **SLURM** scheduling system. The system tries to distribute computation resources **fairly**.
- Depending on the current load, jobs have variable queuing times.
- Small computational resources assignments result in long queue times.
- Resources are precious.

SLURM

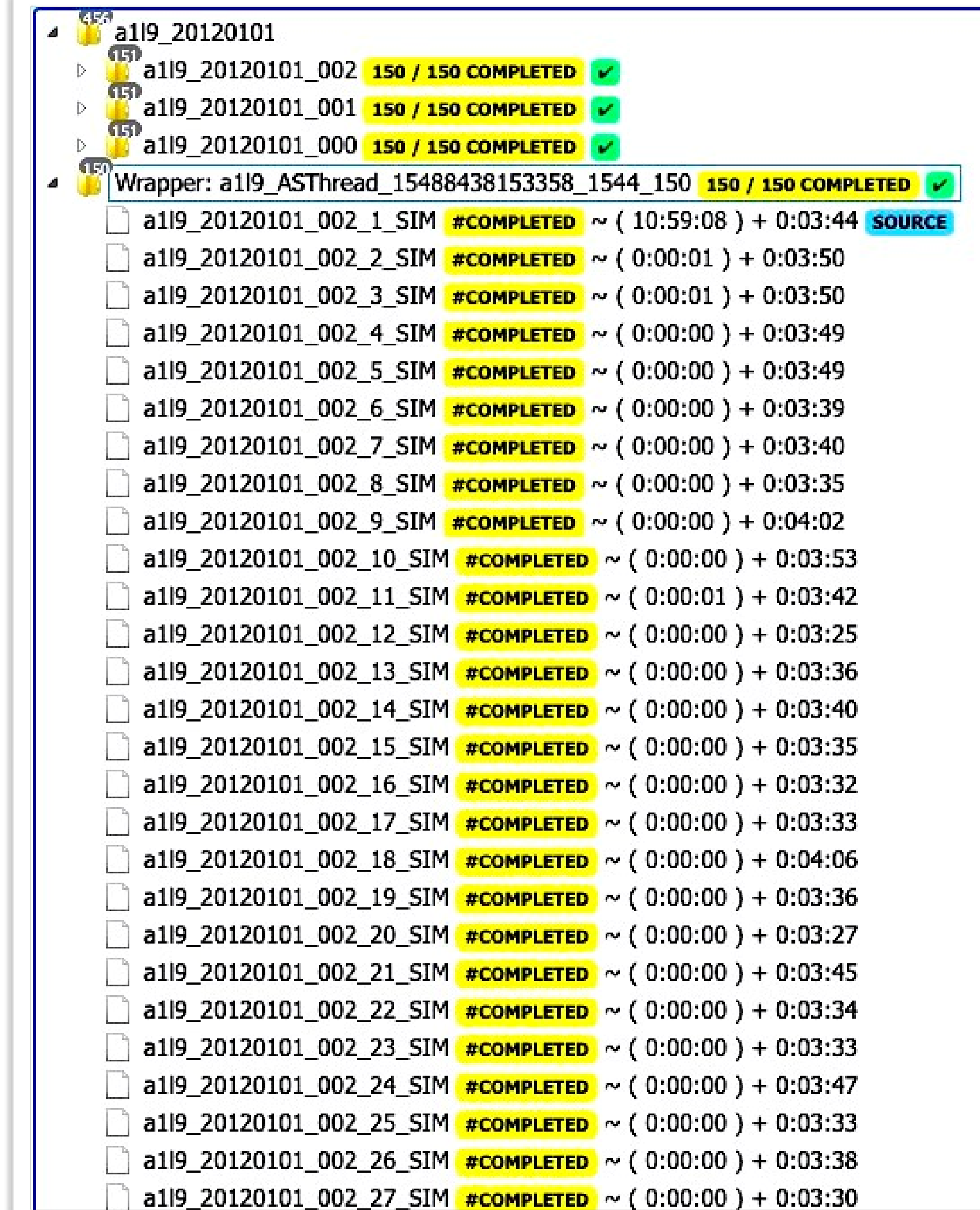
As implemented in Marenosturm4

- We can divide the **SLURM** scheduling mechanism in: **Priority**, **Scheduler**.
- **SLURM** uses some factors to determine the **Priority** value of a job. The higher its value the sooner it will be executed, or the less time it will spend in the **queue**. We focus on **two** important factors: **Age**, **Size**.
- A job that spends time in the queue gains **Age** factor.
- The higher the number of **CPUs** a job requests, the larger the value of the **Size** factor. Reward High Parallelism.
- The **Scheduler** implements a **backfill** mode. This algorithm tries to minimise idle time by starting lower **Priority** when that action would not affect higher **Priority** jobs. The **wallclock**, the estimated running time users declare for their jobs, and the number of requested **CPUs** are used to compute the result of **backfill**.

Wrappers

Using restrictions in our favor

- We can use the rules imposed by the **SLURM** scheduling mechanism in favor of our users.
- We can maximise priority by requesting many **CPUs**. How? **Autosubmit** can wrap similar jobs into a single package that will be sent to the scheduler as if they were a single large job. Thus, increasing its **Size** factor.
- There are different types of **wrappers**: horizontal, vertical, mixed. The user can fit the experiment to the wrapper or choose the proper wrapper for the experiment.
- We can maximise **Priority** by **holding** jobs in the **queue** in advance but preventing them from being executed.



▲	452	a1l9_20120101	
▶	157	a1l9_20120101_002	150 / 150 COMPLETED ✓
▶	157	a1l9_20120101_001	150 / 150 COMPLETED ✓
▶	157	a1l9_20120101_000	150 / 150 COMPLETED ✓
▲	157	Wrapper: a1l9_ASThread_15488438153358_1544_150	150 / 150 COMPLETED ✓
		a1l9_20120101_002_1_SIM	#COMPLETED ~ (10:59:08) + 0:03:44 SOURCE
		a1l9_20120101_002_2_SIM	#COMPLETED ~ (0:00:01) + 0:03:50
		a1l9_20120101_002_3_SIM	#COMPLETED ~ (0:00:01) + 0:03:50
		a1l9_20120101_002_4_SIM	#COMPLETED ~ (0:00:00) + 0:03:49
		a1l9_20120101_002_5_SIM	#COMPLETED ~ (0:00:00) + 0:03:49
		a1l9_20120101_002_6_SIM	#COMPLETED ~ (0:00:00) + 0:03:39
		a1l9_20120101_002_7_SIM	#COMPLETED ~ (0:00:00) + 0:03:40
		a1l9_20120101_002_8_SIM	#COMPLETED ~ (0:00:00) + 0:03:35
		a1l9_20120101_002_9_SIM	#COMPLETED ~ (0:00:00) + 0:04:02
		a1l9_20120101_002_10_SIM	#COMPLETED ~ (0:00:00) + 0:03:53
		a1l9_20120101_002_11_SIM	#COMPLETED ~ (0:00:01) + 0:03:42
		a1l9_20120101_002_12_SIM	#COMPLETED ~ (0:00:00) + 0:03:25
		a1l9_20120101_002_13_SIM	#COMPLETED ~ (0:00:00) + 0:03:36
		a1l9_20120101_002_14_SIM	#COMPLETED ~ (0:00:00) + 0:03:40
		a1l9_20120101_002_15_SIM	#COMPLETED ~ (0:00:00) + 0:03:35
		a1l9_20120101_002_16_SIM	#COMPLETED ~ (0:00:00) + 0:03:32
		a1l9_20120101_002_17_SIM	#COMPLETED ~ (0:00:00) + 0:03:33
		a1l9_20120101_002_18_SIM	#COMPLETED ~ (0:00:00) + 0:04:06
		a1l9_20120101_002_19_SIM	#COMPLETED ~ (0:00:00) + 0:03:36
		a1l9_20120101_002_20_SIM	#COMPLETED ~ (0:00:00) + 0:03:27
		a1l9_20120101_002_21_SIM	#COMPLETED ~ (0:00:00) + 0:03:45
		a1l9_20120101_002_22_SIM	#COMPLETED ~ (0:00:00) + 0:03:34
		a1l9_20120101_002_23_SIM	#COMPLETED ~ (0:00:00) + 0:03:33
		a1l9_20120101_002_24_SIM	#COMPLETED ~ (0:00:00) + 0:03:47
		a1l9_20120101_002_25_SIM	#COMPLETED ~ (0:00:00) + 0:03:33
		a1l9_20120101_002_26_SIM	#COMPLETED ~ (0:00:00) + 0:03:38
		a1l9_20120101_002_27_SIM	#COMPLETED ~ (0:00:00) + 0:03:30

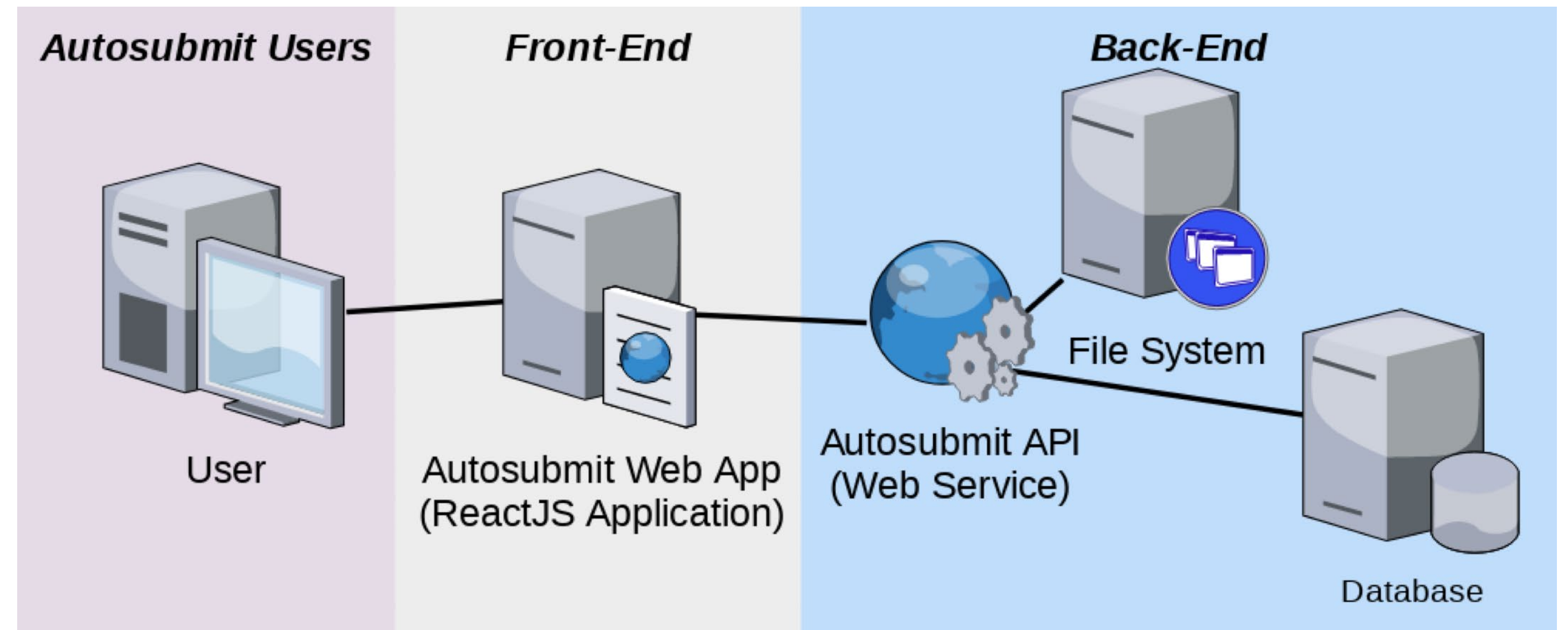
Autosubmit API & GUI

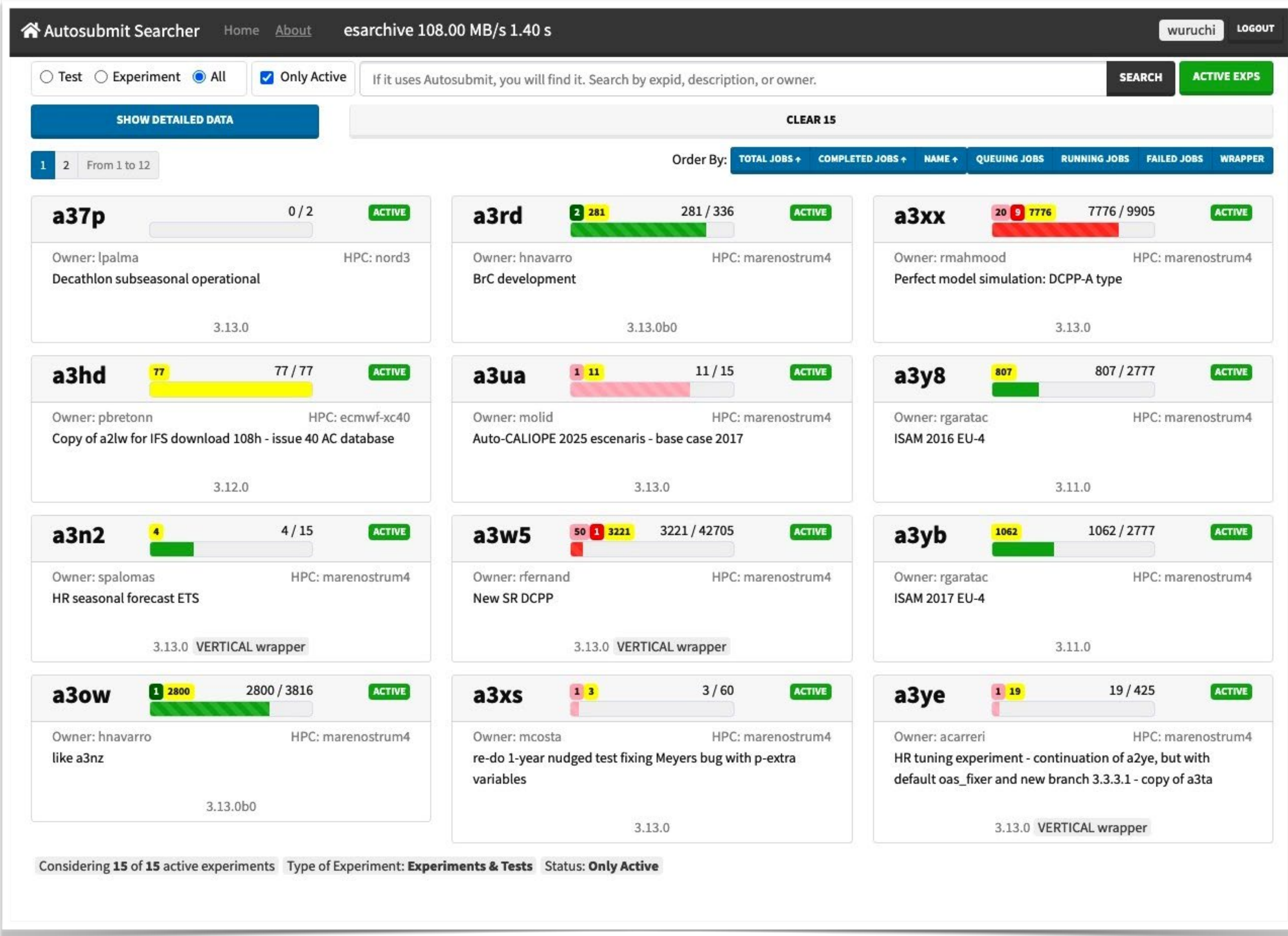


**Barcelona
Supercomputing
Center**
*Centro Nacional
de Supercomputación*

Autosubmit API

It seemed natural that the execution information generated by **Autosubmit** as items in the File System and Databases could also be served as an **API**.





Autosubmit GUI

Autosubmit API can be consumed by a Web front-end.

We apply web tools for data representation.

We can centralise monitoring access to all experiments under Autosubmit.

We handle experiments consisting of tens of jobs to the tens of thousands.

COMPLETED
RUNNING
QUEUING
FAILED

From one terminal to another and back



Generate file text: **READY** **WAITING** **COMPLETED** **SUSPENDED** **FAILED**

CLOSE



The command has been copied to the clipboard. Paste it in your terminal.

Generate file text: **READY** **WAITING** **COMPLETED** **SUSPENDED** **FAILED**

CLOSE



Wrappers

a3pd_20210917_000_1_HERMES 

Start: 2021-09-17 **End:** 2021-09-18
Section: HERMES
Member: 000 **Chunk:** 1
Platform: marenostrium4 **QoS:** bsc_es **Id:** 17529785
Processors: 256 **Wallclock:** 00:30
Queue: 00:00:00 **Run:** 00:01:12
Status: COMPLETED **OUT:** 3 **IN:** 1
/esarchive/autosubmit/a3pd/tmp/LOG_a3pd/a. **COPY OUT** ☐
/esarchive/autosubmit/a3pd/tmp/LOG_a3pd/a: **COPY ERR** ☐
Submit: 2021-09-17 07:26:24 **SYPD:** 3.29
Start: 2021-09-17 07:26:24
Finish: 2021-09-17 07:27:36

auto-CALIOPE-urban: operational | Branch: master | Hpc:



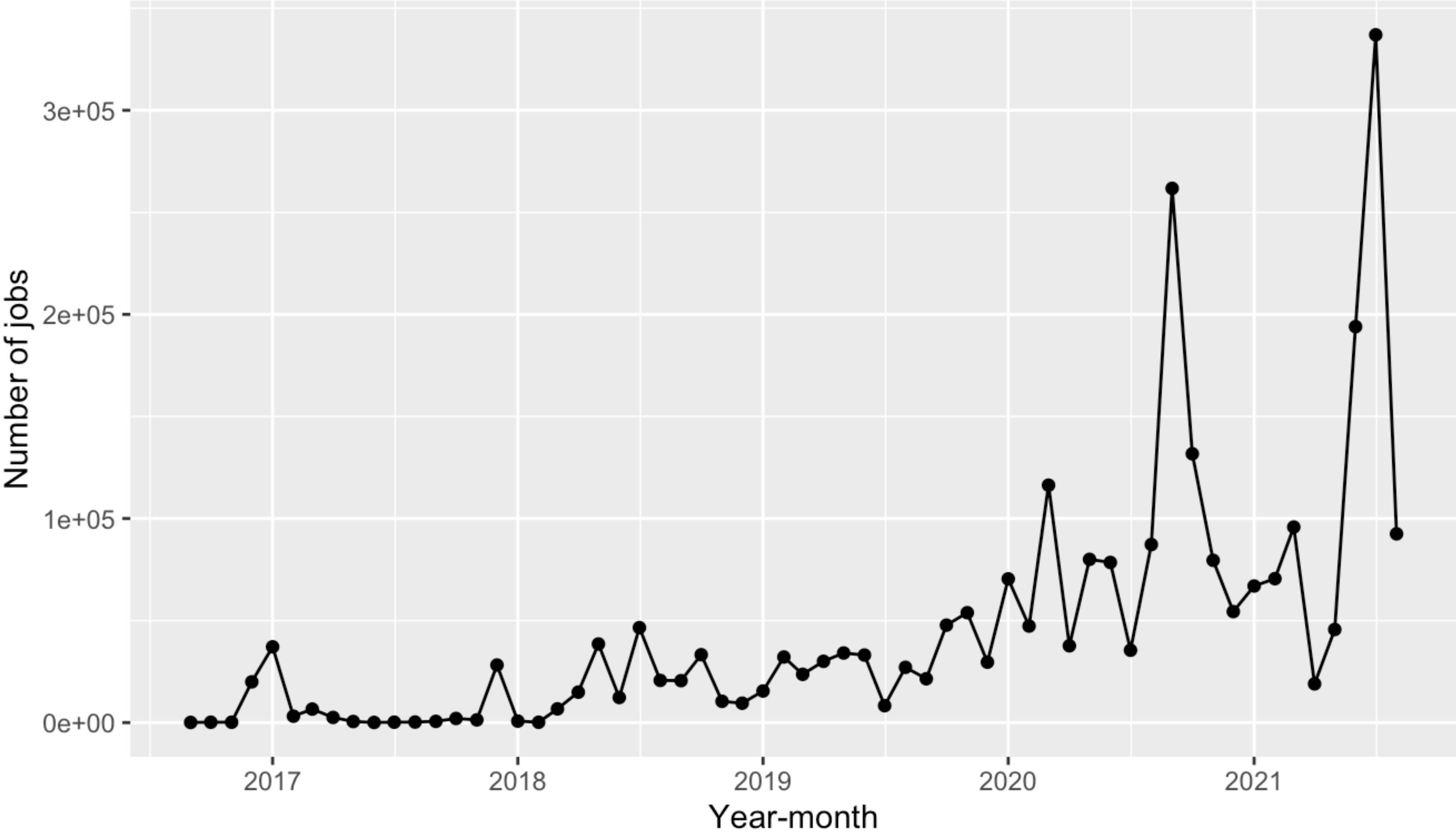
Autosubmit, Autosubmit API & Autosubmit GUI
work together as a single distributed system:
An end-to-end workflow manager.

Data



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

Approximated number of COMPLETED jobs accumulated by year-month



Usage

yearmonth <date>	n_jobs <int>
2021-07-01	336980
2020-09-01	261752
2021-06-01	193997
2020-10-01	131703
2020-03-01	116327
2021-03-01	95818
2021-08-01	92489
2020-08-01	87244
2020-05-01	79994
2020-11-01	79518

Performance Metrics

We gather information about the execution of jobs in the experiment.

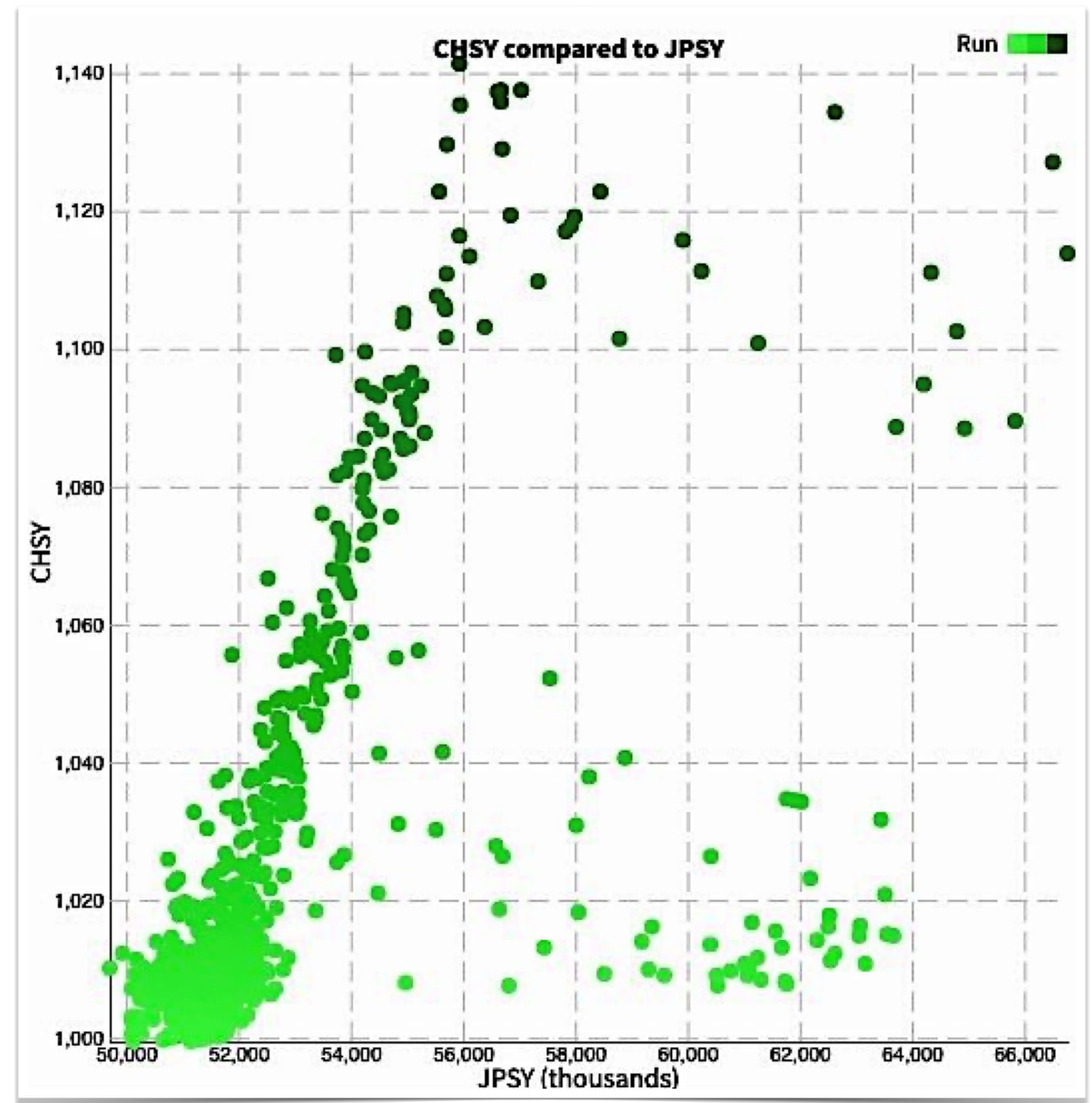
The metrics are currently defined based on “**computational resources spent per simulated time**”.

Not all jobs perform simulation.

Some projects might **not include simulation jobs** in their execution.

Main metrics: Simulated Years per Day, Joules per Simulated Year, Core Hours per Simulated Year.

Autosubmit GUI implements a module that displays the metrics using flexible tools (D3js).



Statistics from the time frame: Start of experiment to 2021-09-16 18:35:00

CPU Consumption 89.68 % Total Queue Time 38.48 hours

Description	Count
Jobs Submitted	31
Jobs Run	31
Jobs Completed	19
Jobs Failed	12

Considers number of jobs and retries.

Description	Hours
Expected Consumption	49.00
Real Consumption	38.48
Failed Real Consumption	10.28

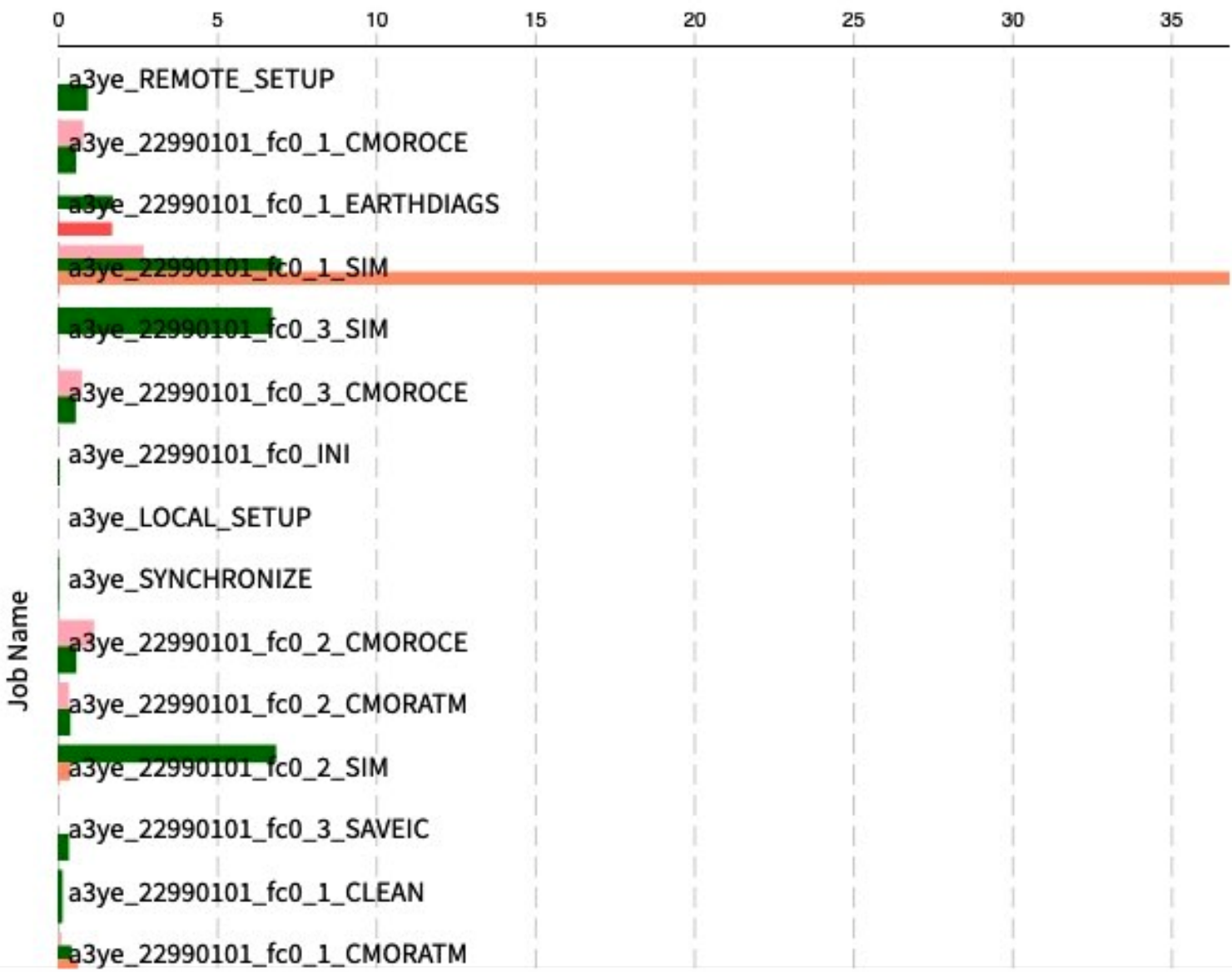
Considers the running time of the jobs and retries.

Description	Hours
Expected CPU Consumption	76,826.00
CPU Consumption	68,897.00
Failed CPU Consumption	19,324.22

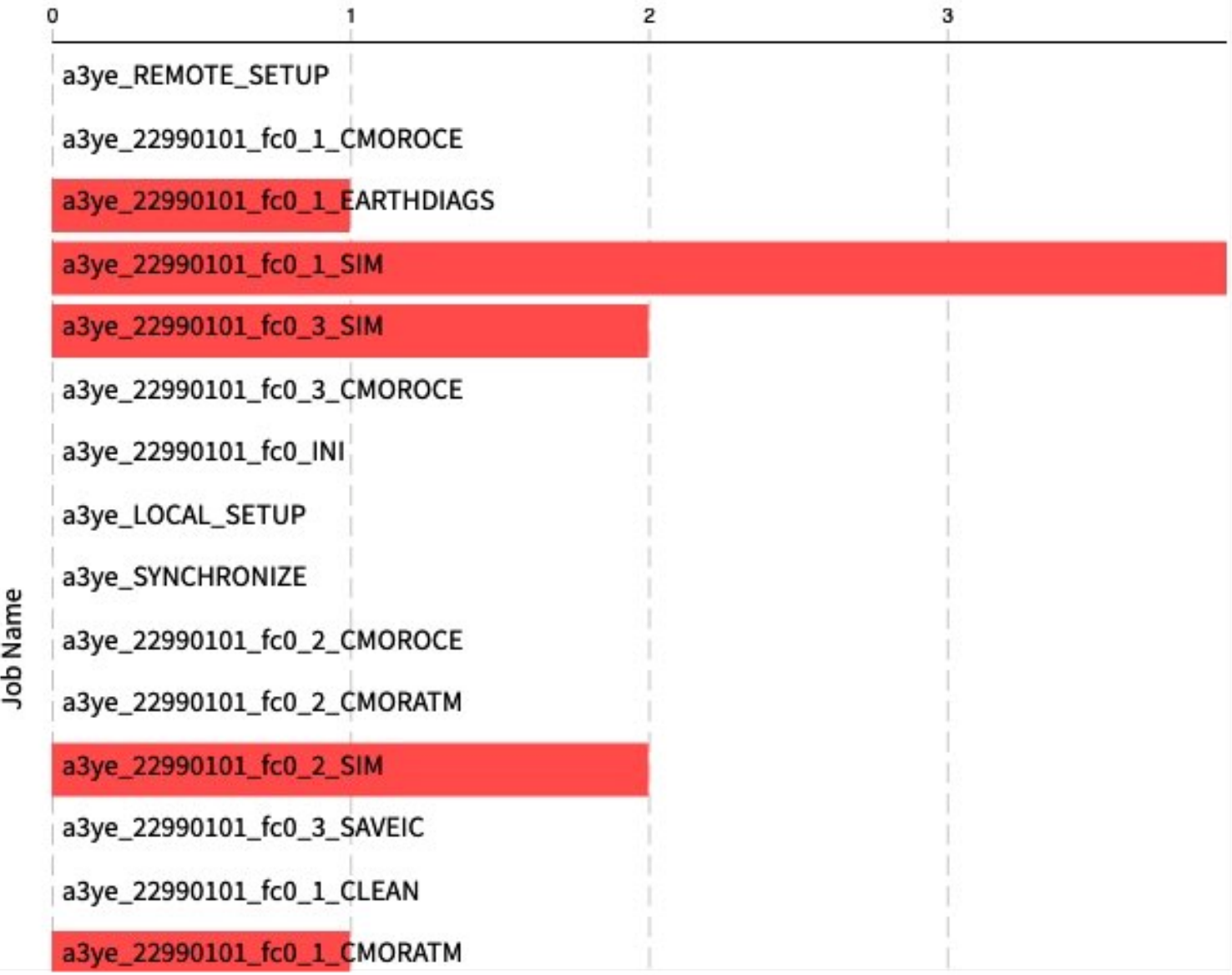
Considers the number of processors requested by the job (and retries) multiplied by the corresponding running time.

☒ Queue ☒ Run ☒ Failed Queue ☒ Failed Run ☐ Failed Attempts

Statistics



Failed Attempts per Job



Statistics

Users can monitor the time consumption of their experiments in a selected interval at a job level.

Historical Information

Autosubmit stores information from current and past executions of **experiments** and their **jobs**.

This information is provided to the user.

Historical data for a3rz_19900201_fc00_1_SIM

RunId	Counter	JobId	Submit	Start	Finish	Queue	Run	Status	Energy
2109171128	62	17534351	2021-09-17-11:28:45	2021-09-17-11:29:14		0:00:29	0:16:20	RUNNING	NA
2109161240	61	17517305	2021-09-16-12:41:02	2021-09-16-12:41:04	2021-09-16-12:43:47	0:00:02	0:02:43	FAILED	124K
2109161014	60	17516667	2021-09-16-10:14:59	2021-09-16-10:15:16	2021-09-16-10:18:30	0:00:17	0:03:14	FAILED	149K
2109161001	59	17516638	2021-09-16-10:03:38	2021-09-16-10:08:28	2021-09-16-10:11:45	0:04:50	0:03:17	FAILED	152K
2109151804	58	17507897	2021-09-15-18:04:45	2021-09-15-18:05:09	2021-09-15-19:33:16	0:00:24	1:28:07	COMPLETED	6550K
2109151756	57	17507852	2021-09-15-17:56:49	2021-09-15-17:56:49	2021-09-15-17:59:47	0:00:00	0:02:58	FAILED	125K
2109151752	56	17507817	2021-09-15-17:53:10	2021-09-15-17:53:26	2021-09-15-17:56:23	0:00:16	0:02:57	FAILED	121K
2109151725	55	17506521	2021-09-15-17:25:51	2021-09-15-17:26:25	2021-09-15-17:48:46	0:00:34	0:22:21	FAILED	NA
2109151406	54	17503379	2021-09-15-14:08:13	2021-09-15-14:08:38	2021-09-15-14:13:13	0:00:25	0:04:35	FAILED	123K
2109151358	53	17503320	2021-09-15-13:58:48	2021-09-15-14:00:25	2021-09-15-14:03:28	0:01:37	0:03:03	FAILED	135K
2109151341	52	17503237	2021-09-15-13:45:31	2021-09-15-13:46:13	2021-09-15-13:49:20	0:00:42	0:03:07	FAILED	124K
2109151332	51	17502895	2021-09-15-13:32:55	2021-09-15-13:34:12	2021-09-15-13:37:11	0:01:17	0:02:59	FAILED	124K
2109151258	50	17502658	2021-09-15-13:24:59	2021-09-15-13:26:11	2021-09-15-13:27:01	0:01:12	0:00:50	FAILED	1K
2109151241	49	17502160	2021-09-15-12:41:44	2021-09-15-12:43:28	2021-09-15-12:46:35	0:01:44	0:03:07	FAILED	132K
2109151217	48	17501563	2021-09-15-12:17:55	2021-09-15-12:18:31	2021-09-15-12:19:38	0:00:36	0:01:07	FAILED	6K
2109151013	47	17499877	2021-09-15-10:13:39	2021-09-15-10:13:43	2021-09-15-10:16:35	0:00:04	0:02:52	FAILED	128K
2109150936	46	17499233	2021-09-15-09:36:33	2021-09-15-09:37:18	2021-09-15-09:40:10	0:00:45	0:02:52	FAILED	124K
2109150915	45	17499006	2021-09-15-09:15:40	2021-09-15-09:15:40	2021-09-15-09:16:55	0:00:00	0:01:15	FAILED	9K
2109150913	44	17498975	2021-09-15-09:13:47	2021-09-15-09:14:03	2021-09-15-09:14:52	0:00:16	0:00:49	FAILED	1K
2109141729	43	17479946	2021-09-14-17:30:10	2021-09-14-17:31:12	2021-09-14-17:34:13	0:01:02	0:03:01	FAILED	125K
2107281320	42	16703413	2021-07-28-13:20:14	2021-07-28-13:20:45	2021-07-28-14:50:04	0:00:31	1:29:19	COMPLETED	6720K
2107281146	41	16703125	2021-07-28-12:37:06	2021-07-28-12:37:34	2021-07-28-12:38:26	0:00:28	0:00:52	FAILED	NA
2107161713	40	16667730	2021-07-16-17:13:33	2021-07-16-17:13:33	2021-07-16-18:42:28	0:00:00	1:28:55	COMPLETED	6470K
2107161609	39	16667299	2021-07-16-16:09:32	2021-07-16-16:09:44	2021-07-16-16:12:07	0:00:12	0:02:23	FAILED	NA
2107161538	38	16667118	2021-07-16-15:39:33	2021-07-16-15:39:45	2021-07-16-15:42:23	0:00:12	0:02:38	FAILED	NA
2107161526	37	16667044	2021-07-16-15:26:38	2021-07-16-15:26:49	2021-07-16-15:29:36	0:00:11	0:02:47	FAILED	NA
2107161510	36	16666972	2021-07-16-15:10:41	2021-07-16-15:10:53	2021-07-16-15:16:53	0:00:12	0:06:00	FAILED	NA
2107161504	35	16666947	2021-07-16-15:05:01	2021-07-16-15:05:25	2021-07-16-15:08:11	0:00:24	0:02:46	FAILED	NA
2107161414	34	16666716	2021-07-16-14:15:59	2021-07-16-14:16:10	2021-07-16-14:21:34	0:00:11	0:05:24	FAILED	NA
2107161224	33	16666638	2021-07-16-13:58:49	2021-07-16-14:04:34	2021-07-16-14:07:43	0:05:45	0:03:09	FAILED	NA
2107161105	32	16665053	2021-07-16-11:05:35	2021-07-16-11:05:47	2021-07-16-11:08:21	0:00:12	0:02:34	FAILED	NA
2107161059	31	16665033	2021-07-16-11:00:46	2021-07-16-11:01:28	2021-07-16-11:01:28	0:00:42	0:00:00	FAILED	NA
2107151743	30	16654314	2021-07-15-18:36:38	2021-07-15-18:36:49	2021-07-15-18:39:35	0:00:11	0:02:46	FAILED	NA
2107151739	29	16654110	2021-07-15-17:39:20	2021-07-15-17:39:36	2021-07-15-17:42:22	0:00:16	0:02:46	FAILED	NA

Next Steps




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

Use the data

- We have information from the execution of different experiments and jobs. We can use this data to train algorithms that could provide the user with useful predictions about the performance of the experiment, or suggest optimal configuration settings (e.g. **wallclock**, wrapper).

More interaction

- **Autosubmit GUI** implements monitoring and command generation tools. This year, it will also implement direct operations on the experiment through the interface, meaning that our users will be able to manage an experiment running on a **High Performance Computing** platform from their browsers, or even from their smartphones.

A photograph of the Chapel Torre Girona, a modern building with a traditional facade. The building features a central gabled section with a pediment and a circular window. Two tall, narrow bell towers with red-tiled roofs flank the central section. The facade is primarily light-colored stone or concrete, with a central section of reddish-brown panels. A large, modern glass entrance is visible in the foreground. The building is surrounded by lush green trees and a clear blue sky with some clouds. The text "Questions & Answers" is overlaid in the center of the image.

Questions & Answers