

Data Challenge 4 Phase II Plans

Ken Herner, Elisabetta Pennacchio

DUNE Pre-Collaboration Meeting meeting

9 May 2022

DC4 Part II Plans

- Main Goals
 - Re-establish keep-up processing on raw data (or close to it)
 - Exercise as much new infrastructure as possible
 - Global GlideinWMS pool
 - MetaCat
 - Rucio
 - Data Dispatcher/Workflow Allocator
 - Take advantage of site storage for outputs as much as we can
- Other goals
 - Number of involved [sites](#): only those with storage, all?

Rough Status of New Pieces

- MetaCat
 - Need to start incorporating [API](#) into job scripts. Perhaps could be done in conjunction with ifdh and/or fife_wrap changes (i.e., how file declaration can be automatic with a json file), but could be done ourselves via some
- Rucio
 - Some progress on environment within jobs but need to ensure we can do a rucio upload within a job. Again, could be incorporated into FIFE tools but also possible to do by hand in our scripts as long as the environment works
- Global pool
 - Successfully tested, but not at scale. Hardware supposedly in place.
- Workflow Allocator
 - See Andrew's talk
- Data Dispatcher
- File delivery protocols to jobs
 - Steam, copy? Depends a bit on format

Possible Plans

- Plan A: all new
- Plan B: Mix of new and old
 - e.g. POMS with MetaCat/Rucio or Workflow Allocator with SAM (just examples)
 - Need to understand if there are component combinations that won't work
- Plan C: Existing setup
 - Would work fine for ProtoDUNE-II but we wouldn't learn much

Thoughts on Schedule

- Plan A is probably not realistic by June 13 (not much will get done next week)
- Running the processing part separately allows for more development time and more complete system sooner for beam datataking
 - Delay one month? Two? Should try to sort that out soon
- Question is how important it is to be processing data in near-real time while simultaneously doing large transfers
 - We did pull it off in 2018 after all, so again it wouldn't be anything new without new infrastructure

Making a Plan

- First, set out the choices
 - What component combinations are possible?
 - Does any component block any other?
- Times: by May 20, we should decide
 - What are the dates for Phase 2?
 - What is the component freeze date (work backwards N weeks)
 - Components should be verified to work within test jobs at least N weeks before starting the campaign to allow for full integration. Work can proceed in parallel for now.
 - How much data do we want to process?
- Can probably be fairly late-binding on the component choices but need to pick a freeze date
- If Phase 1 and 2 are not concurrent, we should set up some later period where we try to do both concurrently, even if only for a few days

Stuff we should start now (and can)

- Start testing MetaCat and Rucio individually in jobs
 - Suspect at least a month for each; can proceed in parallel
- Scale up Global Pool testing
- Understand state of Workflow Allocator and Data Dispatcher, especially wrt monitoring

POMS basic elements (quick reminder)

- Once a given processing workflow is defined (dunesw version, fcl files, input and output files handling), a campaigns are managed and monitored with the help of a WEB interface allowing to:

- Clone an already existing campaign
- Edit and modify a campaign with a GUI editor
- Check dependencies among campaign stages
- Launch submission (manually or by using POMS cron tab)

- Check submission history
- View detailed file report
- Generate a .ini file

Select	Campaign Name	id	Active	Submissions Running	Submission History	Files Report	Launch	Dependencies	GUI Editor	Clone Campaign	.ini File	Delete Campaign
<input type="checkbox"/>					5	6	4	3	2	1	7	
<input type="checkbox"/>	RITM1355274_ndk_BDM	6241	No	0								
<input type="checkbox"/>	RITM1355274_ndk_legacynbar	6225	No	0								
<input type="checkbox"/>	RITM1355274_ndk_protondecay	6222	No	0								
<input type="checkbox"/>	dune_nd_production_2022_v1	6199	No	0								
<input type="checkbox"/>	RITM1355274_ndk	6152	No	0								
<input type="checkbox"/>	RITM1331650_lightLE	6081	No	0								
<input type="checkbox"/>	VD_coldbox_TDE_2021	6076	Yes	0								
<input type="checkbox"/>	ProtoDUNE_SP_MC_1GeV_RITM1331140_lowG4	6067	No	0								
<input type="checkbox"/>	protoDUNE_SP_data_RITM1312299_cosmics	5997	No	0								
<input type="checkbox"/>	FDVDProd1	5816	No	0								

3 main input files needed to create a campaign

1. Login setup

- To specify login host, to start job submission, account, proxies,...
- Jobs are submitted to the shared pool; changing login host may allow to submit to the global pool

2. Job Type

- To specify launch scripts and recovery parameters (memory and CPU): **failed jobs are resubmitted automatically** → no manual intervention needed

The image shows two overlapping configuration windows. The 'login_setup' window on the left has the following fields: name (MC_Winter2022_v1_0_launch), host (dunegpvm15.fnal.gov), account (dunepro), and setup (export X509_USER_PROXY=/opt/dunepro/di). The 'job_type' window on the right has: name (mcc11_gen_g4), launch_script (fife_launch), parameters (["-c", "/dune/app/home/dunepro/poms_MCC]), output_file_patterns (%.root), and recoveries (["pending_files", ["-Osubmit.expected-lifetin]). Both windows have 'Reset', 'OK', and 'Cancel' buttons. Below the windows, two labels are circled: 'MC_Winter2022_v1_0_launch' in blue and 'mcc11_gen_g4' in yellow.

3. .cfg file, whose main blocks are:

- [global] : group, experiment, versions, variable definitions, fcls, output and log directories,...
 - [env_pass] : environment variables and values to pass with `-e` option to jobsub
 - [executable] : executable to run with options several executables (*executable_1*, *executable_2*,..) can be specified (sim, g4, reco,...)
 - [job_output] : specify output filenames and path, enable metadata creation (*declare_metadata*)
 - [job_setup] : setup commands to run, prescript: command to run before execution, postscript: command to run after execution
 - [sam_consumer] : to check all what is SAM related
 - [submit] : argument to jobsub_submit
- Values in the .cfg file can be overwritten/updated by using the GUI in POMS/, allowing to reuse the same configuration file in different campaigns

Monitoring tools

- POMS campaigns are monitored with **Landscape**
 - Summary of submitted, running and failed jobs
 - Estimation of CPU of memory efficiency of running jobs
 - Information on held jobs (jobid, hold_date, HoldReasonCode)
<https://fifemon.fnal.gov/monitor/d/000000146/why-are-my-jobs-held?orgId=1&var-user=dunepro>
- Shifter dashboard: <https://fifemon.fnal.gov/monitor/d/HMj5hqVik/production-shifter?orgId=1&var-vo=dune>
- Summary plots of memory and CPU can be obtained with KIBANA
- Output files are written to temporary directories. The File Transfer Service (FTS) watches these directories, and transfers files to their final destination
 - FTS monitoring page: <http://dunesamgpvm01.fnal.gov:8787/fts/status>
 - ssh tunnel or a VPN needed from outside FERMILAB
- Some examples in the following pages

Shifter dashboard

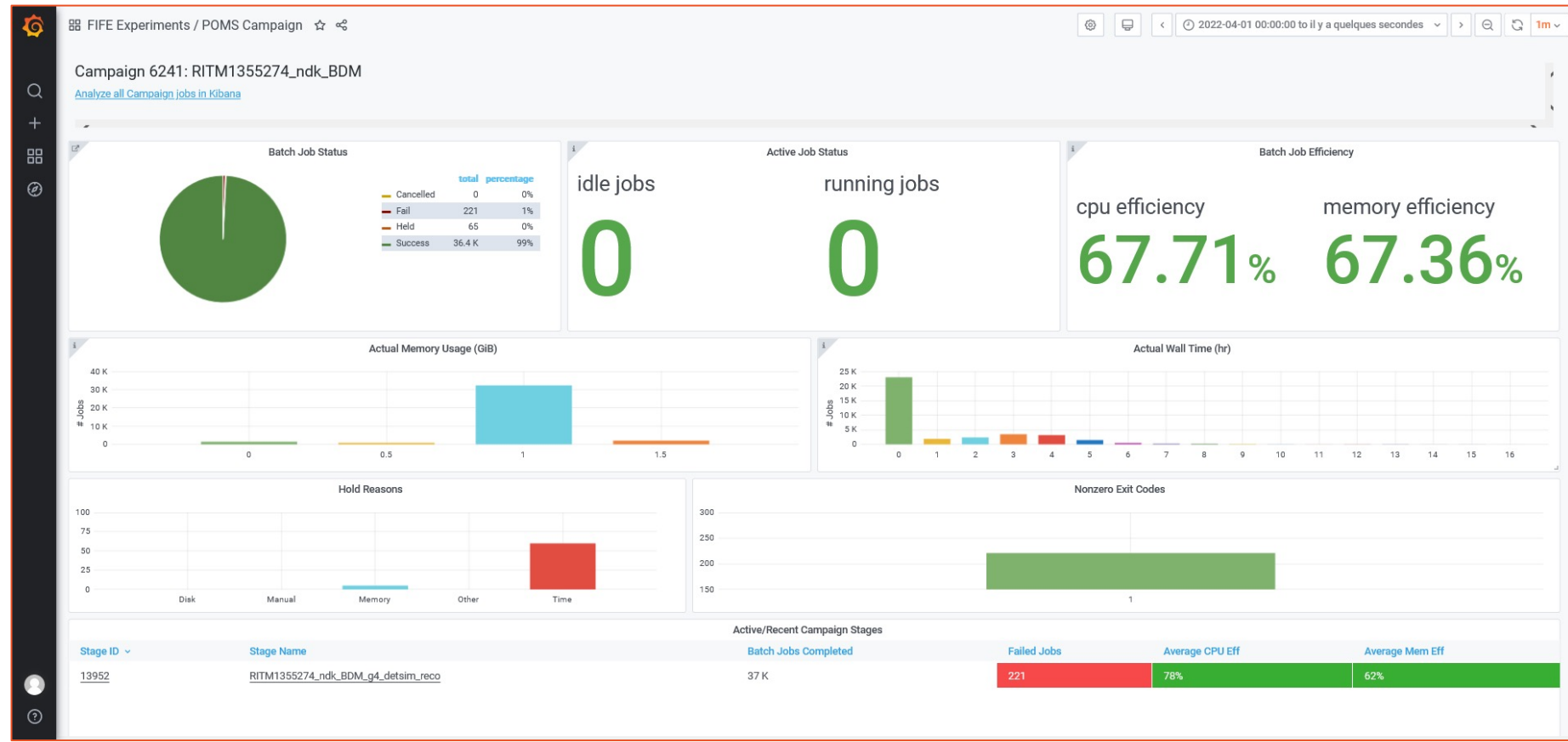
A specific time range can be selected



Campaign ID	Campaign Name (click for stages)	Active POMS Campaigns
6241	RITM1355274_ndk_BDM	Batch Jobs Completed
6225	RITM1355274_ndk_Jegacynnbar	37 K
6222	RITM1355274_ndk_protondecay	2 K
6199	dune_nd_production_2022_v1	2 K
6152	RITM1355274_ndk	55
6076	VD_coldbox_TDE_2021	71 K
		1 K

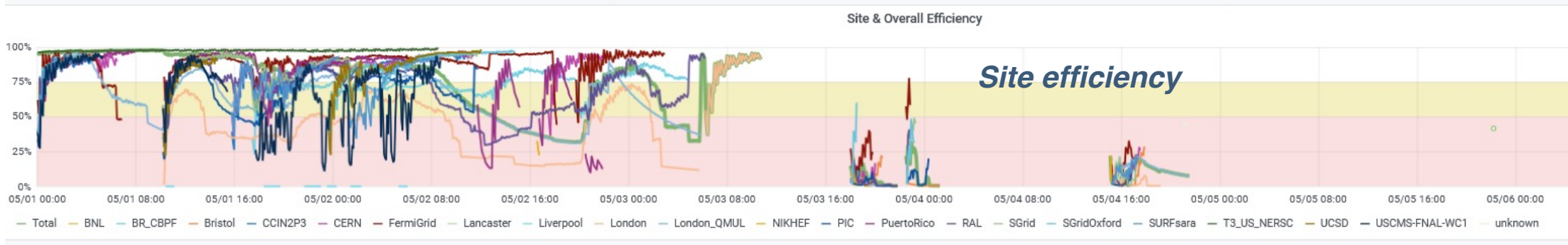
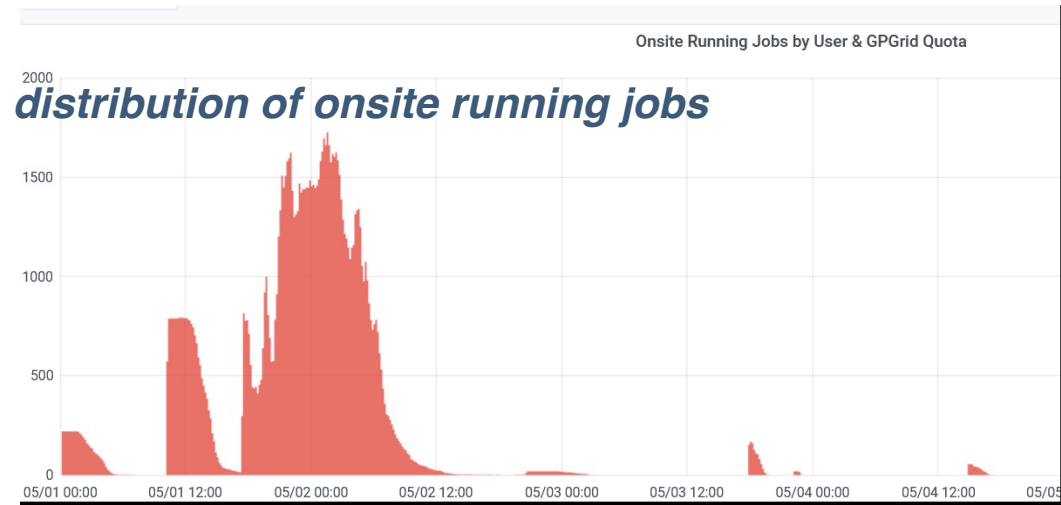
Links to grafana

Links to kibana



Other useful information :

For a select time range



FTS monitoring

- statistics of output files transfer to their final location

Summary

Summary table

FTS: OK	SAM: OK
Completed files:	236836
All error files:	88
Failed transfers:	1
In progress files:	0
Waiting on tape:	0
Transferring:	0
New files:	0

Details for each transferred file are available as well :

+Recent completed transfers (50 hidden)

+Failed transfers (1 hidden)

+All errors (88 hidden)

+In progress (0 hidden)

+New (0 hidden)

+Configuration → *To list directories monitored by FTS*

Summary distributions for last 30 days

