PDVD Monte Carlo production

1. General introduction to MetaCat , Rucio and justIN

2. Status of Monte Carlo production

Elisabetta Pennacchio, IP2I
ProtoDUNE VD Sim/Reco Meeting
05/03/2024

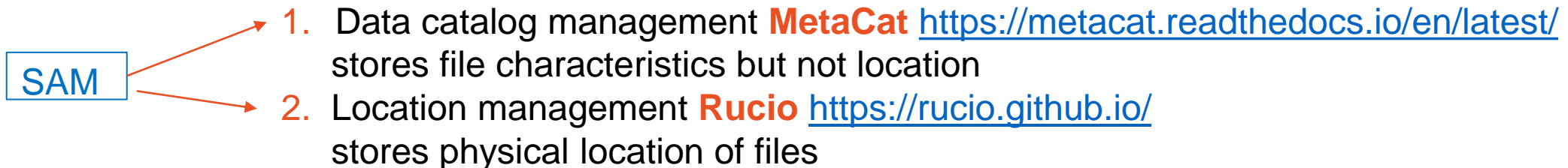# General introduction to MetaCat , Rucio and justIN

The Data Management subsystem brings data from the detectors to the archival storage facility (tapes) and then distributes data to the storage elements around the world.

SAM:

- designed for D0 and CDF
- Object cataloged: files and collection of files called SAM dataset.
- Data files are not stored in SAM but metadata are (metadata allows user to identify a file and locate data file)

SAM has served DUNE until now, a replacement is needed to have data movement capacity for a distributed worldwide storage system

DUNE is now transitioning to new systems (*simplified picture*)

SAM

1. Data catalog management **MetaCat** https://metacat.readthedocs.io/en/latest/
   stores file characteristics but not location
2. Location management **Rucio** https://rucio.github.io/
   stores physical location of files

SAM datasets and file locations are not supported anymore

➤ **Fermilab GPVMs moved to Alma 9 on April 30** (Heidi mail on April 28th)

➤ Presentation at the collaboration call on April 19th

➤ It is still possible to use SL7 in a container as explained in the wiki:
https://wiki.dunescience.org/wiki/SL7_to_Alma9_conversion

➤ This presentation assumes you are working with SL7

```
NAME="AlmaLinux"
VERSION="9.3 (Shamrock Pampas Cat)"
ID="almalinux"
ID_LIKE="rhel centos fedora"
VERSION_ID="9.3"
PLATFORM_ID="platform:el9"
PRETTY_NAME="AlmaLinux 9.3 (Shamrock Pampas Cat)"
ANSI_COLOR="0;34"
LOGO="fedora-logo-icon"
CPE_NAME="cpe:/o:almalinux:almalinux:9::baseos"
HOME_URL="https://almalinux.org/"
DOCUMENTATION_URL="https://wiki.almalinux.org/"
BUG_REPORT_URL="https://bugs.almalinux.org/"

ALMALINUX_MANTISBT_PROJECT="AlmaLinux-9"
ALMALINUX_MANTISBT_PROJECT_VERSION="9.3"
REDHAT_SUPPORT_PRODUCT="AlmaLinux"
REDHAT_SUPPORT_PRODUCT_VERSION="9.3"
```

How to find datasets and locate files in nutshell
(*a translation of samweb get-file-access-url*)

1) setup RUCIO environment (*on dunegpvm*)
   need to be DUNE member and have a FERMILAB valid account

```
>source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
>setup rucio
>setup python v3_9_15
>kx509
>export RUCIO_ACCOUNT= $USER
```

2) Datasets are listed here:
   https://wiki.dunescience.org/wiki/Data_Collections_Manager/data_sets

An example: dataset from latest reco2(tracks and shower) reconstruction campaign, nu sample, FD2-VD

fardet-vd:fardet-vd__fd_mc_2023a_reco2__full-reconstructed__v09_81_00d02__reco2_dunevd10kt_nu_1x8x6_3view_30deg_geov3__prodgenie_nu_numu2nue_nue2nutau_dunevd10kt_1x8x6_3view_30deg__out1__v1_official⧉	File count: 17140 Event count: 1714000	Reco2, Vertical drift

To list dataset content you can:

1) run the rucio command:
   rucio list-files *fardet-vd:fardet-vd__fd_mc_2023a_reco2__full-reconstructed__v09_81_00d02__reco2_dunevd10kt_nu_1x8x6_3view_30deg_geov3__prodgenie_nu_dunevd10kt_1x8x6_3view_30deg__out1__v1_official*

2) follow the arrow on the right

fardet-vd:fardet-vd__fd_mc_2023a_reco2__full-reconstructed__v09_81_00d02__reco2_dunevd10kt_nu_1x8x6_3view_30deg_geov3__prodgenie_nu_numu2nue_nue2nutau_dunevd10kt_1x8x6_3view_30deg__out1__v1_official⧉	File count: 17140 Event count: 1714000	Reco2, Vertical drift

to access MetaCat web interface

Dataset fardet-vd:fardet-vd__fd_mc_2023a_reco2__full-reconstructed__v09_81_00d02__reco2_dunevd10kt_nu_1x8x6_3view_30deg_geov3__prodgenie_nu_dunevd10kt_1x8x6_3view_30deg__out1__v1_official

100/21051 (approximate) files

fardet-vd:nu_dunevd10kt_1x8x6_3view_30deg_1002_134_20230809T200445Z_gen_g4_detsim_hitreco__20240220T211721Z_reco2.root ⧉
fardet-vd:nu_dunevd10kt_1x8x6_3view_30deg_1002_426_20230809T203543Z_gen_g4_detsim_hitreco__20240220T213156Z_reco2.root ⧉
fardet-vd:nu_dunevd10kt_1x8x6_3view_30deg_1237_482_20230802T013453Z_gen_g4_detsim_hitreco__20240219T192234Z_reco2.root ⧉
fardet-vd:nu_dunevd10kt_1x8x6_3view_30deg_1237_637_20230802T013511Z_gen_g4_detsim_hitreco__20240220T144713Z_reco2.root ⧉
fardet-vd:nu_dunevd10kt_1x8x6_3view_30deg_1237_739_20230802T013605Z_gen_g4_detsim_hitreco__20240220T142415Z_reco2.root ⧉
fardet-vd:nu_dunevd10kt_1x8x6_3view_30deg_1239_23_20230802T040422Z_gen_g4_detsim_hitreco__20240219T193540Z_reco2.root ⧉
fardet-vd:nu_dunevd10kt_1x8x6_3view_30deg_1239_325_20230802T040444Z_gen_g4_detsim_hitreco__20240219T194213Z_reco2.root ⧉
fardet-vd:nu_dunevd10kt_1x8x6_3view_30deg_1239_583_20230802T051827Z_gen_g4_detsim_hitreco__20240220T145932Z_reco2.root ⧉

6

To get file location for

fardet-vd:nu_dunevd10kt_1x8x6_3view_30deg_1002_134_20230809T200445Z_gen_g4_detsim_hitreco__20240220T211721Z_reco2.root

run:

rucio -a $USER list-file-replicas fardet-vd:nu_dunevd10kt_1x8x6_3view_30deg_1002_134_20230809T200445Z_gen_g4_detsim_hitreco__20240220T211721Z_reco2.root --pfns

root://dune.dcache.nikhef.nl:1094/pnfs/nikhef.nl/data/dune/generic/rucio/fardet-vd/a1/11/nu_dunevd10kt_1x8x6_3view_30deg_1002_134_20230809T200445Z_gen_g4_detsim_hitreco__20240220T211721Z_reco2.root

root://fndca1.fnal.gov:1094/pnfs/fnal.gov/usr/dune/tape_backed/dunepro//fardet-vd/full-reconstructed/2024/mc/out1/fd_mc_2023a_reco2/00/00/10/02/nu_dunevd10kt_1x8x6_3view_30deg_1002_134_20230809T200445Z_gen_g4_detsim_hitreco__20240220T211721Z_reco2.root_1709165820

root://fndca1.fnal.gov:1094/pnfs/fnal.gov/usr/dune/persistent/staging/fardet-vd/a1/11/nu_dunevd10kt_1x8x6_3view_30deg_1002_134_20230809T200445Z_gen_g4_detsim_hitreco__20240220T211721Z_reco2.root

- **Results, 3 different locations. How to choose?**

 tape_backed: it is the copy from tape, may require to be staged

Two other locations are shown: one in Europe and one in US. Select the one nearest to your location
(page 17: list of storage elements)

Once the file located, you can run *lar*:

lar -c anatree_dunevd10kt_1x8x6_3view_30deg_geov3.fcl
root://dune.dcache.nikhef.nl:1094/pnfs/nikhef.nl/data/dune/generic/rucio/fardet-vd/a1/11/nu_dunevd10kt_1x8x6_3view_30deg_1002_134_20230809T200445Z_gen_g4_detsim_hitreco__20240220T211721Z_reco2.root

This example shows how to run lar on a file.

To process several files by submitting batch jobs on the grid → justIN (*new workflow system replacing POMS)*

justIN is a workflow system that processes data by satisfying the requirements of data location/data catalog, rapid code distribution service and job submission to the grid.

justIN ties:
1. MetaCat search queries that obtain lists of files to process
2. Rucio knowledge of where replicas of files are
3. a table of site-to-storage "distances" to make best choices about where to run each type of job

To process data using justIN :

➢ **need to provide a jobscript (shell script) with some basic tasks:**

- ▪ Set software environment
- ▪ Find where the data is
- ▪ Process the data
- ▪ Save the output in a given location

➢ **run a justIN command line**

justin simple-workflow   args…..

once you ran the command, you get the workflow ID.
In case of any problem, you can stop your workflow by running
finish-workflow --workflow-id <ID>

- A complete tutorial on how to use justIN is available [here](#)

DUNE justIN tutorial

- This tutorial, with additional specific DUNE examples, has been presented at the FD simulation and reconstruction meeting on April 22 ([slides](#))

- These instructions refer to SL7, instructions on how to use it with Alma 9 are in preparation
  The basic concepts (jobscript, rapid code distribution to jobs via cvmfs) stay unchanged)

- justIN command [main page](#)

justin command man page

This man page is distributed along with the justin command itself.

justin command

The justin command allows the creation and monitoring of workflows, and various queries of the justIN database's knowledge of sites, storages etc.

See the snapshot of its man page for the full list of subcommands, options, and their syntax.
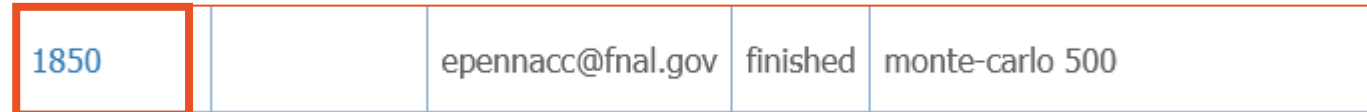
- By following the examples from https://justin.dune.hep.ac.uk/docs/tutorials.dune.md
  and in the presentation(slides 8 →18) you can :

  1. Understand how a jobscript is structured
  2. Process data using standard code
  3. Process data using customized fcl files and/or customized code
  4. Select the input dataset
  5. Specify where your output should go (jobs writing to scratch)

- Examples of jobscripts are provided as well (from the GitHub   production repository)

- The jobscripts checklist is available in the backup

- The following slides will not repeat these concepts, but provide some practical examples on how to "navigate" through justIN dashboard

Two general remarks:
- ALWAYS test code and jobscript before sending jobs to the grid
- For any large processing (MC or DATA) producing large output that has to be shared within the Collaboration, please contact the production group.

Example: PDVD MC simulations

justIN · Dashboard **Workflows** Jobs AWT Sites Storages Docs Login

Workflows

1850		epennacc@fnal.gov	finished	monte-carlo 500

Workflow 1850

Description	
State	finished
MQL	monte-carlo 500
Scope	vd-protodune
HTCondor Group	group_dune.prod_mcsim
Finding	once
Submitted by	epennacc@fnal.gov
Created	2024-04-17 18:26:39
Submitted	2024-04-17 18:26:39
Started	2024-04-17 18:27:29
Finished	2024-04-18 11:45:59

File states per stage

Stage ID	Files	Finding	Unallocated	Allocated	Outputting	Processed	Not found	Failed
1	500	0	0	0	0	500	0	0

File states are described here:
https://justin.dune.hep.ac.uk/docs/files.md

12

Workflow 1850

Description	
State	finished
MQL	monte-carlo 500
Scope	vd-protodune
HTCondor Group	group_dune.prod_mcsim
Finding	once
Submitted by	epennacc@fnal.gov
Created	2024-04-17 18:26:39
Submitted	2024-04-17 18:26:39
Started	2024-04-17 18:27:29
Finished	2024-04-18 11:45:59

File states per stage

Stage ID	Files	Finding	Unallocated	Allocated	Outputting	Processed	Not found	Failed
1	500	0	0	0	0	500	0	0

To access full statistics:
-sites where jobs ran
-storage used for input/output

To access details of each job (see next page)

Files for Workflow 1850, Stage 1 (state = processed)

Page: 1 Next

DID	Attempts	State	Allocated				
			Time	RSE	Site	Jobsub ID	
monte-carlo-001850-000500	1	processed	2024-04-17 18:55:34	MONTECARLO	US_FNAL-T1	9981.9@justin-prod-sch	
monte-carlo-001850-000499	1	processed	2024-04-17 18:55:32	MONTECARLO	US_FNAL-T1	9981.18@justin-prod-sc	

File monte-carlo-001850-000500

Workflow ID	1850
Stage ID	1
Size	0 bytes (0.00GB)
State	processed
Processed	2024-04-17 21:29:50
Replicas	MONTECARLO: 000500
Allocations	9981.9@justin-prod-sched02.dune.hep.ac.uk 2024-04-17 18:55:34 (finished US_FNAL-T1 MONTECARLO)
List file events	

Jobsub ID	9981.9@justin-prod-sched02.dune.hep.ac.uk	
Workflow ID	1850	
Stage ID	1	
User name	epennacc@fnal.gov	
HTCondor Group	group_dune.prod_mcsim	
Requested	Processors	1
	RSS bytes	4193255424 (3999 MB)
	Wall seconds limit	80000 (22 hours)
Submitted time	2024-04-17 18:28:51	
Site	US_FNAL-T1	
Entry	CMSHTPC_T1_US_FNAL_condce_opp1_whole	
Last heartbeat	2024-04-17 21:29:50	
From worker node	Hostname	dunegli-12337-0-cmswn5009.fnal.gov
	cpuinfo	AMD EPYC 7543 32-Core Processor
	OS release	Scientific Linux release 7.9 (Nitrogen)
	Processors	1
	RSS bytes	4193255424 (3999 MB)
	Wall seconds limit	171000 (47 hours)
	Inner Apptainer?	True
Job state	finished	
Allocator name	justin-allocator-pro.dune.hep.ac.uk	
Started	2024-04-17 18:55:31	
Input files	monte-carlo-001850-000500	
Jobscript	Exit code	0
	Real time	2h (9114s)
	CPU time	2h (8944s = 98%)
Outputting started	2024-04-17 21:27:25	
Output files	vd-protodune:prod_cosmics_protodunevd_20240417T185534Z_000500_gen_g4_stage1.root vd-protodune:prod_cosmics_protodunevd_20240417T185534Z_000500_gen_g4_stage1_g4_stage2.root vd-protodune:prod_cosmics_protodunevd_20240417T185534Z_000500_gen_g4_stage1_g4_stage2_detsim.root vd-protodune:prod_cosmics_protodunevd_20240417T185534Z_000500_gen_g4_stage1_g4_stage2_detsim_reco.root	
Finished	2024-04-17 21:29:50	
Saved logs	justin-logs:9981.9-justin-prod-sched02.dune.hep.ac.uk.logs.tgz	
List job events Wrapper job log		

To access log files

14

For each file, you see where it was processed and which Rucio Storage Element it came from.

File states per stage

Stage ID	Files	Finding	Unallocated	Allocated	Outputting	Processed	Not found	Failed
1	500	0	0	0	0	500	0	0

Monte Carlo

DID	Attempts	State	Allocated			
			Time	RSE	Site	Jobsub ID
monte-carlo-001850-000500	1	processed	2024-04-17 18:55:34	MONTECARLO	US_FNAL-T1	9981.9@justin-prod-sched02.dune.hep.ac.uk
monte-carlo-001850-000499	1	processed	2024-04-17 18:55:32	MONTECARLO	US_FNAL-T1	9981.18@justin-prod-sched02.dune.hep.ac.uk
monte-carlo-001850-000498	1	processed	2024-04-17 18:55:31	MONTECARLO	US_FNAL-T1	9980.183@justin-prod-sched02.dune.hep.ac.uk
monte-carlo-001850-000497	1	processed	2024-04-17 18:55:31	MONTECARLO	US_FNAL-T1	9981.11@justin-prod-sched02.dune.hep.ac.uk

Data processing

DID	Attempts	State	Allocated			
			Time	RSE	Site	Jobsub ID
vd-coldbox-bottom:np02_bde_coldbox_run021445_0121_dataflow0_datawriter_0_20230501T171719.hdf5	1	allocated	2024-05-03 08:03:14	CERN_PDUNE_EOS	NL_NIKHEF	206634.25@justin-prod-sched01.dune.hep.ac.uk
vd-coldbox-bottom:np02_bde_coldbox_run021445_0110_dataflow0_datawriter_0_20230501T171322.hdf5	1	allocated	2024-05-03 08:03:14	CERN_PDUNE_EOS	NL_NIKHEF	206634.61@justin-prod-sched01.dune.hep.ac.uk
vd-coldbox-bottom:np02_bde_coldbox_run021445_0125_dataflow0_datawriter_0_20230501T171845.hdf5	1	allocated	2024-05-03 08:03:14	CERN_PDUNE_EOS	NL_NIKHEF	206634.37@justin-prod-sched01.dune.hep.ac.uk

15

How it looks like if there are failed jobs

Workflow 1872

Description	
State	finished
MQL	files from higuera:fardet-hd__fd_mc_he_2023a__mc__hit-reconstructed__prodgenie_atmnu_max_weighted_randompolicy_dune10kt_1x2x6.fcl__v09_79_00d02__preliminary skip 6000 limit 5000 ordere
Scope	usertests
HTCondor Group	group_dune
Finding	once
Submitted by	epennacc@fnal.gov
Created	2024-04-19 05:46:00
Submitted	2024-04-19 05:46:00
Started	2024-04-19 05:46:53
Finished	2024-04-21 07:59:02

File states per stage

Stage ID	Files	Finding	Unallocated	Allocated	Outputting	Processed	Not found	Failed
1	5000	0	0	0	0	4989	0	11

To list storage elements (where data can be)

RSE name	Occupancy	Read (R,j)	Write (R,j)	LAN write	WAN write	Region / Country / Site		
CERN_PDUNE_CASTOR	Decommissioned							
CERN_PDUNE_EOS	0.75	True (T,T)	False (F,F)	davs	davs	Europe	CERN	CERN
DUNE_CERN_EOS	0.67	True (T,T)	True (T,T)	davs	davs	Europe	CERN	CERN
DUNE_ES_PIC	0.23	True (T,T)	True (T,T)	davs	davs	Europe	ES	ES_PIC
DUNE_FR_CCIN2P3	Decommissioned							
DUNE_FR_CCIN2P3_DISK	0.76	True (T,T)	True (T,T)	davs	davs	Europe	FR	FR_CCIN2P3
DUNE_FR_CCIN2P3_TAPE	Decommissioned							
DUNE_FR_CCIN2P3_XROOTD	Decommissioned							
DUNE_IN_TIFR	3.99	True (T,T)	False (T,F)	root	root	South_Asia	IN	IN_TIFR
DUNE_IT_INFN_CNAF	0.03	True (T,T)	False (T,F)	davs	davs	Europe	IT	IT_CNAF
DUNE_UK_LANCASTER_CEPH	1.00	True (T,T)	True (T,T)	davs	davs	Europe	UK	UK_Lancaster
DUNE_US_BNL_SDCC	0.99	True (T,T)	True (T,T)	davs	davs	North_America	US	US_BNL
DUNE_US_FNAL_DISK_STAGE	0.93	True (T,T)	True (T,T)	davs	davs	North_America	US	US_FNAL-FermiGrid
NIKHEF	0.63	True (T,T)	True (T,T)	davs	davs	Europe	NL	NL_NIKHEF

Details for each site are also availabale

RSE name	Occupancy	Read (R,j)	Write (R,j)	LAN write	WAN write	Region / Country / Site		
CERN_PDUNE_CASTOR	Decommissioned							
CERN_PDUNE_EOS	0.75	True (T,T)	False (F,F)	davs	davs	Europe	CERN	CERN
DUNE_CERN_EOS	0.67	True (T,T)	True (T,T)	davs	davs	Europe	CERN	CERN
DUNE_ES_PIC	0.23	True (T,T)	True (T,T)	davs	davs	Europe	ES	ES_PIC
DUNE_FR_CCIN2P3	Decommissioned							
DUNE_FR_CCIN2P3_DISK	0.76	True (T,T)	True (T,T)	davs	davs	Europe	FR	FR_CCIN2P3
DUNE_FR_CCIN2P3_TAPE	Decommissioned							
DUNE_FR_CCIN2P3_XROOTD	Decommissioned							
DUNE_IN_TIFR	3.99	True (T,T)	False (T,F)	root	root	South_Asia	IN	IN_TIFR
DUNE_IT_INFN_CNAF	0.03	True (T,T)	False (T,F)	davs	davs	Europe	IT	IT_CNAF
DUNE_UK_LANCASTER_CEPH	1.00	True (T,T)	True (T,T)	davs	davs	Europe	UK	UK_Lancaster
DUNE_US_BNL_SDCC	0.99	True (T,T)	True (T,T)	davs	davs	North_America	US	US_BNL
DUNE_US_FNAL_DISK_STAGE	0.93	True (T,T)	True (T,T)	davs	davs	North_America	US	US_FNAL-FermiGrid
NIKHEF	0.63	True (T,T)	True (T,T)	davs	davs	Europe	NL	NL_NIKHEF

Storage DUNE_FR_CCIN2P3_DISK

RSE name	DUNE_FR_CCIN2P3_DISK
Country	FR
Region	Europe
Site name	FR_CCIN2P3
Occupancy	0.77
Read	True (Rucio=True, justIN=True)
Write	True (Rucio=True, justIN=True)
Write schemes	LAN davs, WAN davs
All events AWT events GitHub issues	

Sites by distance

Only sites seen in the OSG configuration in the last 7 days are shown.

Test values are xrdcp read and rucio upload exit codes for jobs at each site contacting this RSE. 0 = success.

Site name	Distance	Enabled	Read test	Write test	Events	
FR_CCIN2P3	0	True	0	0	All	AWT
CERN	30	True	0	0	All	AWT
CH_UNIBE-LHEP	30	True	0	0	All	AWT
CZ_FZU	30	True	0	0	All	AWT
ES_CIEMAT	30	True	0	0	All	AWT
IN_TIFR	100	True			All	AW
US_BNL	100	True	0	0	All	AW
US_Caltech	100	True			All	AW
US_Colorado	100	True	0	0	All	AW

Status of production

Updated: 15 April 2024

Please find below a summary of the first three requested pre-productions:

1. protodune-vd_cosmics_pureAr
2. protodune-vd_cosmics_Xe10ppm
3. protodune-vd_electron_1GeV_pureAr

dunesw version: **v09_88_00d00 (v09_82_02d01)**

Dataset name: protodune-vd_cosmics_pureAr

Sample 1:

10K events already produced, some cleaning up required to remove files not correctly registered in MetaCat/Rucio,
Once done, final datasets can be prepared

Sample 2:

once datasets prepared, the workflow for this new sample can be developed and tested

backup

Jobscripts checklist

If you are writing your own jobscript or modifying one from someone else, please check the following are true:

- You check that the application you are running has "worked" somehow. Usually this will involve checking the lar executable has returned zero. This is available as the shell variable `$?` immediately after the command you are checking. If you put other commands between your application's command and checking `$?`, you will be checking if those other commands succeeded instead.
- When your jobscript has "worked", return 0. You can do this with `exit 0`
- When your jobscript has failed, return a non-zero value. This is logged by justIN so you can quickly see what is happening, and also stops justIN from uploading any of the output files (which are presumably wrong in some way?) to storage. Exit codes can be between 1 and 127. For example `exit 57`
- Whenever you run `justin-get-file` in your jobscript, check if the output is empty. In that case, there are no more input files to process and your jobscript should stop immediately with `exit 0` as it's ok, there's just nothing more to do.
- Once an input file has been processed successfully, add its PFN (like an xroot URL) to justin-processed-pfns.txt or its DID (Rucio scope:name) to justin-processed-dids.txt This tells justIN that the file can be marked as processed in its database, and does not need to be given to another job to try again.
- If your jobscript processes multiple input files, do not leave output files resulting from input files you fail to process successfully: those output files will be uploaded to storage too. A good pattern is to check the processing worked, and then rename the output file with mv to its final name if and only if the processing worked.
- If your jobscript produces metadata files for your output files, they must have exactly the same name as the output file they are about plus `.json` This is how justIN finds the metadata, and if none is found, then only very basic metadata is generated by justIN itself and stored in MetaCat.

How to setup MetaCat, Rucio and justIN (on dunegpvm)

first run:

/cvmfs/oasis.opensciencegrid.org/mis/apptainer/current/bin/apptainer shell --shell=/bin/bash -B /cvmfs,/exp,/nashome,/pnfs/dune,/opt,/run/user,/etc/hostname,/etc/hosts,/etc/krb5.conf --ipc --pid /cvmfs/singularity.opensciencegrid.org/fermilab/fnal-dev-sl7:latest

Then:

source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
setup python v3_9_15
setup rucio
kx509
export RUCIO_ACCOUNT= $USER
export METACAT_SERVER_URL=https://metacat.fnal.gov:9443/dune_meta_prod/app
export METACAT_AUTH_SERVER_URL=https://metacat.fnal.gov:8143/auth/dune
setup metacat
setup justin
justin version
rm -f /var/tmp/justin.session.`id -u`
justin time

Examples of metacat queries:

- To list files from run 21445 (as shown by Yoann)

Apptainer> metacat query   "files where 21445 in core.runs"

- To list first 100 files:

 Apptainer> metacat query   "files where 21445 in core.runs and core.data_tier=raw limit 100"

To check how many files belong to run 21445:

Apptainer> metacat query   -s "files where 21445 in core.runs"
Files: 304
Total size: 1296713825568 (1.297 TB)

====================
List files from a given workflow:

metacat query "files where dune.workflow['workflow_id'] in (1630)"

Links

Metacat WEB interface: https://metacat.fnal.gov:9443/dune_meta_prod/app/auth/login

justIN: https://justin.dune.hep.ac.uk/docs/

Slack channels: #workflow