ND Reco/Sim: Metadata

Initial efforts to catalogue ND production

Feb. 03, 2024

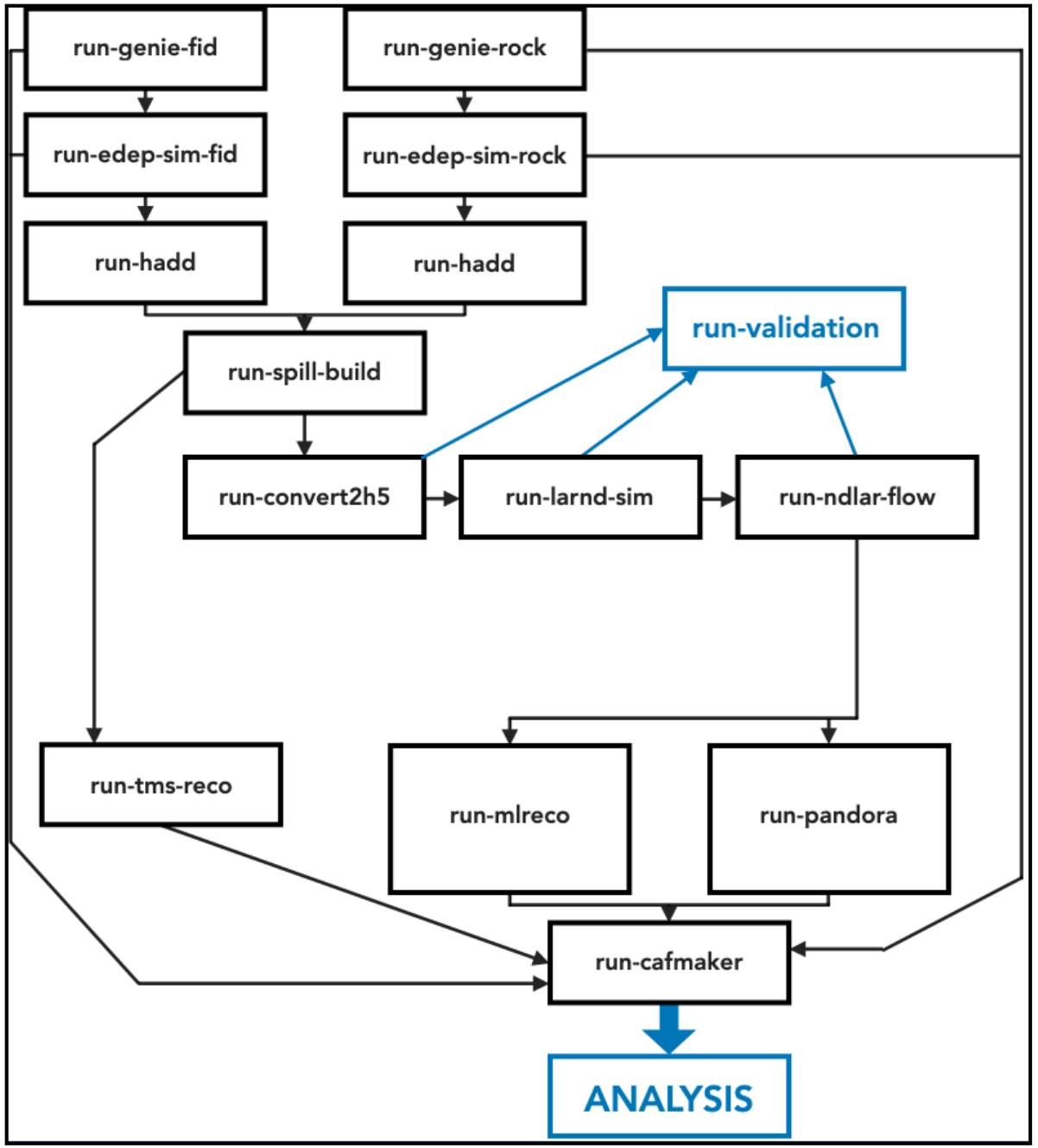
Core Software & Computing

Mike Dolce

DEEP UNDERGROUND NEUTRINO EXPERIMENT

Background

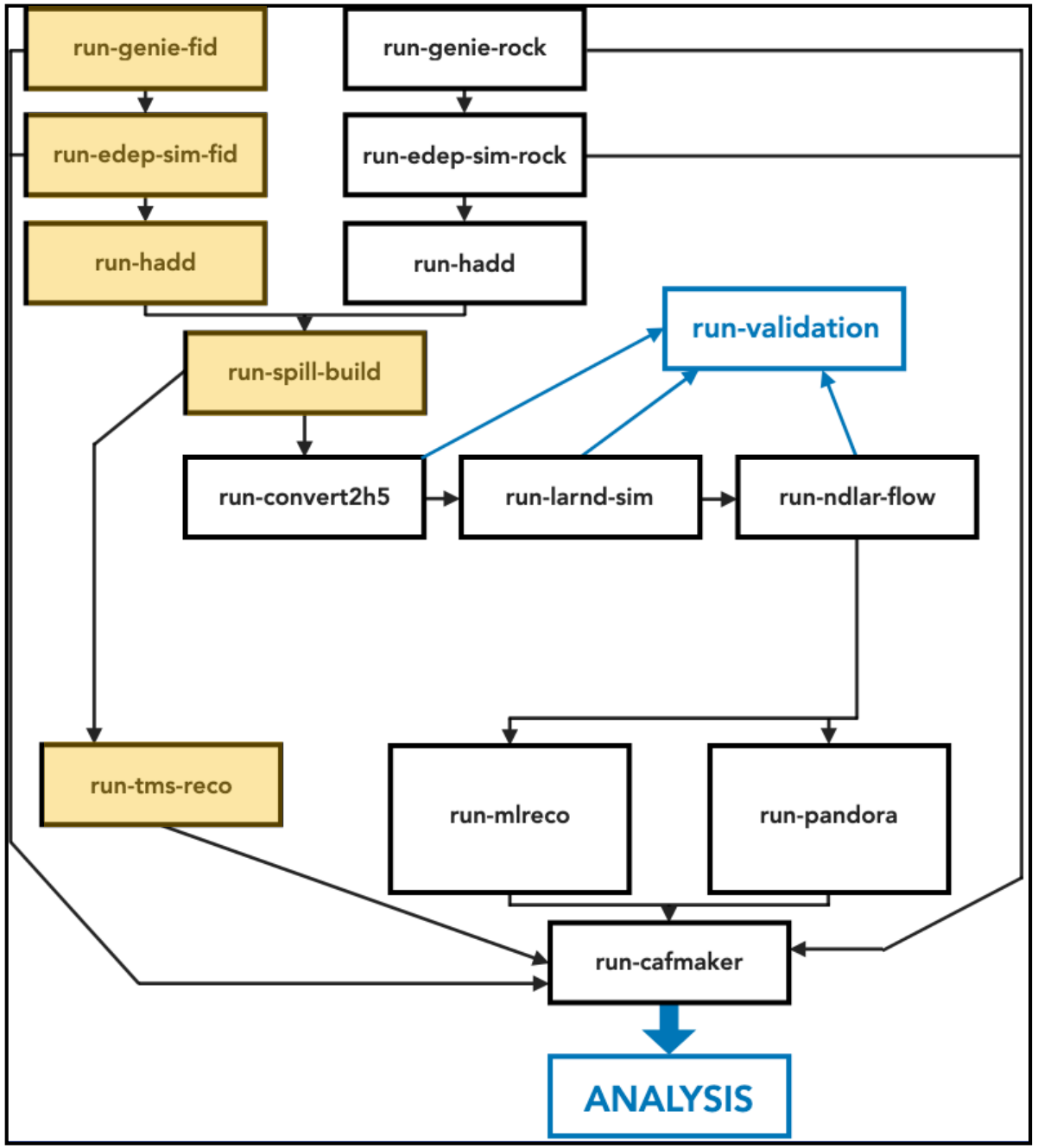
- Alex Booth and I are able to generate ND production somewhat cohesively.
  - Lot's of great progress presented at workshop.
- We have yet to produce a complete sample, from every stage of the workflow — we are all working on this.
  - Done entirely at NERSC.
- Currently, produce samples based on requests from analysis groups.
  - These are small sample requests.
  - We copy the files to GPVMs ourselves.



Credit A. Booth

TMS Study

- Have been working with TMS group to produce MC for different geometry studies.
  - genie → edep-sim → hadd → spill-build.
- TMS group performed the **`run-tms-reco`** stage themselves (Asa N.'s talks: ND Reco/Sim & TMS Meeting).
- Completed, but doing so raised DUNE Computing attention:
  - **We should have metadata for these files!**
- Declare as a **`metacat`** dataset.



Credit A. Booth

Current Workflow

- Have spoke with Steve Timm about metadata and **`metacat`**.
  - We use a script living in 2x2_sim repo:
    - 2x2_sim/admin/dump_metadata.py.
- The script makes a json file for each ROOT/h5 file in a directory on DUNE GPVMs.
  - Loops over all files in directory, extracts the file extension and applies the metadata provided.
- Advised to give the json and ROOT files to Steve Timm, he would declare the files to **`metacat`** and create the dataset — success?
  - These files will live in the **`ndprod`** namespace.
  - I think they would automatically be declared to **`rucio`** too?

Questions about Metadata

- From DUNE metacat glossary, **`core.application.version`** expects a DUNESW version.
  - What is the right choice for ND production? Omit for now?
- There are fields relating to "cluster" that we will omit.
  - **`justIN`** should provide these for free — we can wait until then for those fields.
  - Is **`core.application.family`** OK?
- Anything else here amiss?


```
"file_name": "TMSGeometryStudy_1E18_FHC.spill.nu.0000122.EDEPSIM_SPILLS.root",
"namespace": "ndprod",
"file_size": 2160742841,
"checksum": "1c2a9a9b",
"dune.campaign": "TMSGeometry_hybrid_study",
"metadata": {
    "core.application.family": "ND_Production",
    "core.application.name": "run-spill-build",
    "core.application.version": "xx",
    "core.data_stream": "physics",
    "core.data_tier": "simulated",
    "core.event_count": 132,
    "core.first_event": 122000,
    "core.last_event": 122131,
    "core.file_type": "mc",
    "core.file_format": "root",
    "core.group": "dune",
    "core.run_type": "neardet-2x2",
    "core.runs": [
        122
    ],
    "core.file_content_status": "good",
    "retention.class": "physics",
    "retention.status": "active",
    "dune_mc.name": "TMSGeometry_hybrid_study",
    "dune_mc.generators": "genie",
    "dune_mc.genie_tune": "AR23_20i_00_000",
    "dune_mc.top_volume": "volArgonCubeDetector75",
    "dune_mc.geometry_version": "nd_hall_with_lar_tms-hybrid_sand.gdml",
    "dune_mc.2x2_sim.tag": "nd-production-v02.01",
    "dune_mc.fireworks4dune.tag": "main",
    "dune_mc.ND_Production.tag": "nd-production-v02.01",
    "dune_mc.nu": true,
    "dune_mc.rock": false,
    "cluster.gen_site": "nersc",
    "cluster.hostname": "xx",
    "cluster.os": "xx",
    "cluster.os_version": "xx",
    "cluster.compiler": "xx"
```

Summary & Questions

- ND Production is fairly cohesive, moving forward to metadata for legitimate analysis requests now, with **`metacat`**.
  - We (Alex & I) plan to have several batches of files coming in shortly — continue this workflow for the near future?
  - TMS group alone has multiple productions coming up, to **`run-tms-reco`**.
- We use **`dump_metadata.sh`** script from 2x2_sim repo to produce our metadata for **`metacat`**.
  - We know we are missing some fields, and have added others (not listed on Glossary).
  - Json files provided to Steve Timm, who will declare the files to **`metacat`**, and then **`rucio`** — waiting to hear back.
  - These datasets will be owned by **`dunepro`**, and live in the **`ndprod`** namespace.
- Once these datasets exist, where will they be documented?
  - ND_Production repo wiki? 2x2_sim? Somewhere else?
- How might our workflow evolve as **`dunepro`** when time comes?
  - At some point, we would like to have this run *during* the job, not after.
  - Can we write files to dropbox (and read the metadata) to declare automatically.

Backup

Added Motivation for metadata

- Again, we are in a state where we can ~easily produce ND production samples for specific outputs.
- TMS group anticipates **further** geometry studies in near future.
  - Metadata implementation would help with this study alone (as many as 5 different TMS Reco samples).
  - Would be great to automate the metadata generation for this near-term samples.
- Also happens to be a todo item for ND production (GitHub todo).

MetaCat Glossary

https://dune.github.io/DataCatalogDocs/glossary.html

Additional terms used for reconstruction and simulation

dune.campaign:	A big scale activity used for production - examples are PDSPProd4a and fd_mc_2023a_reco2
dune.requestid:	The formal request id for the campaign in the system
dune.config_file:	The top level configuration used to produce this file
dune.workflow:	a description of the workflow that produced this file - produced by the JustIn system
dune.output_status:	the value should be "confirmed" - this tells you that the output exists
core.application.family:	broad description of the application (art/edepsim)
core.application.name:	the specific application, reco1/reco2/detsim…
core.application.version:	the DUNESW version

Minimal Monte Carlo terms ¶

core.group:	Should be *dune* or a physics group
dune_mc.gen_fcl_filename:	tells you the generator fcl file so you know what kind of mc it is.
dune_mc.geometry_version:	the geometry version used
dune_mc.generators:	

MC terms specific to particular detectors

dune_mc.electron_lifetime:	PD/FD - electron lifetime
dune_mc.space_charge:	PD/FD - space charge
dune_mc.with_cosmics:	PD/FD - cosmics as well as beam
Beam.momentum:	PD/FD - beam momentum in GeV/c
Beam.polarity:	PD/FD - polarity of beam

there may be others in future.

Minimal terms for raw data

[] denotes fields automatically filled in by the system

name:	File name
namespace:	Metacat namespace for file
checksums:	dictionary of checksums - adler32 is the default
***[created_timestamp]*:**	Unix timestamp for when file was cataloged
creator:	account that created the file
size:	size in bytes
***[fid]*:**	hash-name for the file - equivalent to namespace:name
***[retired]*:**	has this file been retired?
***[retired_by]*:**	who did it?
***[retired_timestamp]*:**	when was it retired
***[updated_by]*:**	who has updated this catalog entry?
***[updated_timestamp]*:**	when did they do it?
parents:	[The files that this file was produced from, you need to declare parents when making child files]
***[children]*:**	[list of files that are derived from this file - autogenerated when you declare the child as having parents]
metadata:	description of file contents with this content
core.data_stream:	type of data taking (commissioning, calibration, test, physics, cosmics)
core.data_tier:	type of data (raw, g4,)
core.end_time:	unix UTC time at which the process that created the file ended -
core.event_count:	number of events in the file
core.events:	[list of events in the file],
core.file_content_status:	status of the file - default is "good"
core.file_format:	format of the data (hdf5, root)
core.file_type:	flag to tell mc from data, (detector or mc)
core.first_event_number:	first event number
core.last_event_number:	last event number
core.run_type:	which detector took the data "protodune-sp, hd-fardet …"
core.runs:	[list of runs]
core.runs_subruns:	[list of subruns in run*100000+subrun format]
core.start_time:	unix UTC time at which the process that created the file started
retention.status:	should be "active" flag to tell if the file is being used and should be retained
retention.class:	flag used to determine retention status (physics, test, …)

Path to TMS hybrid and stereo files

- These are the files made from the original TMS geometry study request.
- These files would be added to **`metacat`** first.

`/pnfs/dune/persistent/users/mdolce/ND_Production`