ATLAS Data Reprocessing Workflow

Input to the ATLAS Sites Jamboree

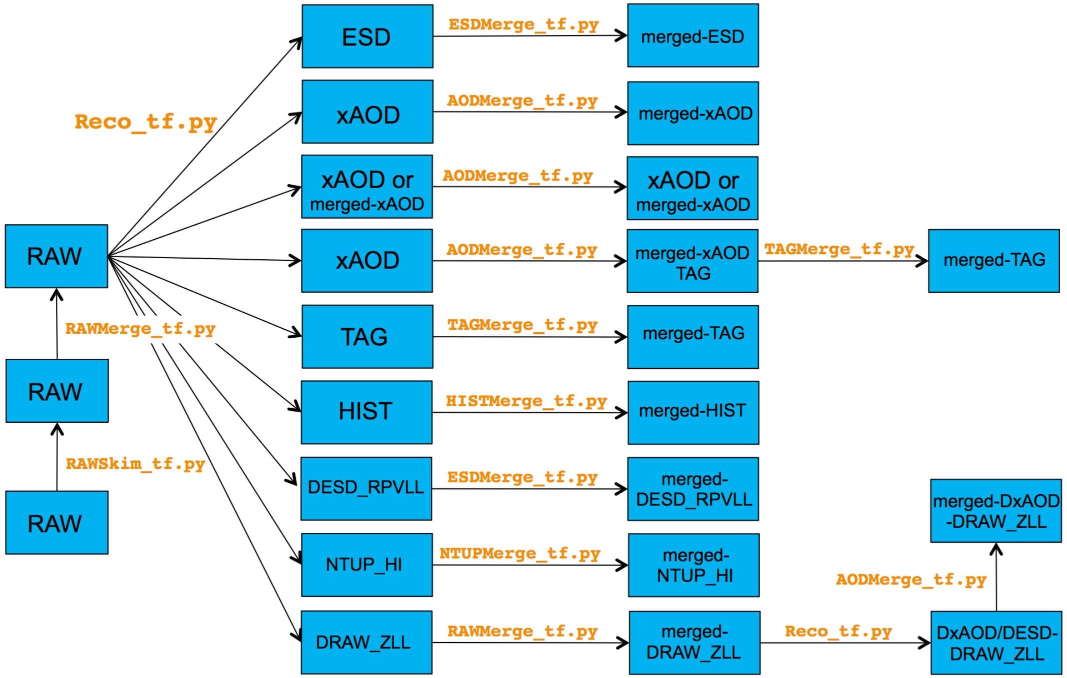
https://twiki.cern.ch/twiki/bin/view/Atlas/DataPreparationReprocessing

David South (DESY)

ATLAS Sites Jamboree
January 27th, 2016

ATLAS data reprocessing workflow

- Generally, a data reprocessing comprises a reco step, followed by merges
- Request is not single one-to-one steps: we have multiple outputs, varying levels of merging
  - Reco produces multiple outputs using the RAW inputs: AOD, HIST, DRAW*, DESD(M)*, DAOD*, and (very rarely) ESD
  - In the on-going 2015 data reprocessing campaign: 14 formats, including 4 DRAW, which subsequently run reco again to produce further DESD(M)* and DAOD* formats
  - That makes total of 3 different DAOD types in ATLAS: from RAW, from DRAW and later from AOD (derivations)



Reco_tf.py

RAW

RAWMerge_tf.py

RAW

RAWSkim_tf.py

RAW

ESD → ESDMerge_tf.py → merged-ESD

xAOD → AODMerge_tf.py → merged-xAOD

xAOD or merged-xAOD → AODMerge_tf.py → xAOD or merged-xAOD

xAOD → AODMerge_tf.py → merged-xAOD TAG → TAGMerge_tf.py → merged-TAG

TAG → TAGMerge_tf.py → merged-TAG

HIST → HISTMerge_tf.py → merged-HIST

DESD_RPVLL → ESDMerge_tf.py → merged-DESD_RPVLL

NTUP_HI → NTUPMerge_tf.py → merged-NTUP_HI

DRAW_ZLL → RAWMerge_tf.py → merged-DRAW_ZLL → Reco_tf.py → DxAOD/DESD-DRAW_ZLL → AODMerge_tf.py → merged-DxAOD-DRAW_ZLL

- All these additional formats requires quite some gymnastics in ProdSys2 using “changeType”, but it all works out in the end

ATLAS data reprocessing workflow (2)

- These happen 2-3 times year, either a “fast reprocessing” (conditions updates only) or “full reprocessing” change of software: for ADC and site support however they are the same
- Campaign comes in two parts:
- Express stream 1: (smaller) express + CosmicCalo streams used for Data Quality checks and sign off
  - Often submitted with high priority, mostly run on T1s, sometimes 1001
  - May also not be the full statistics, aim is for quick turnaround, a few days
- Full reprocessing: all relevant streams, multiple formats, usual priority is 890
  - A full reprocessing requires the support of the T2s, cannot run on just T1s
  - Better situation than in run 1, where we had three physics streams: now only physics_Main
  - Have tried to use T1s for merge steps to avoid file transfers, but not yet successful
- In the background we always have smaller and usually less complicated campaigns running, mostly producing AOD and HIST only

Some more details on reco and merges

- Reco run on m-core (8) with starting value of 2000MB/core, often takes more
- Reco jobs typically run for up to 3 hours, but the tails can go up to 24h
- By default we use 15 attempts before a job fails.. and this is often needed
  - Lost heartbeats, pilot failure, SLURM(?) errors, memory kick-ups by JEDI, ..
- Usually do 1:1 production: one RAW input file to one output file per format
  - However, the large size of the ESD in rel. 20 (which normally only exists temporarily, but still needs to be made in the job) means sometimes we split the input into chunks of 5000 events
- Merges are quicker, and in principle "easy" and "just run": don't really expect problems in the merge step, unless there are problems with missing inputs
- Merges are currently always run on s-core
- We don't use internal JEDI merging: cannot use this currently in ProdSys2 as we want to merge some outputs (to different levels, e.g AOD 10:1, HIST: 50:1) and not others, so merges are separate jobs, with multiple steps for HIST to produce a single file in the end

Some more data reprocessing specifics

- Unlike MC, we require 100% job completion
- Rely on the “exhausted” task state to prevent jobs finishing and merges then also finishing on incomplete datasets
  - More often than not the task can be completed via additional retries, tweaking parameters reassigning to different sites
  - This works very well, but requires manual intervention and babysitting of tasks, especially for larger reprocessing campaigns
- Merged AODs from the data processing are used as input to the derivations run by Nurcan et. al
  - Successfully implemented that the derivation tasks can already start before the reprocessing task is completed, picking up the AODs as they come in