

Requirements document:

Inclusion of skimming in data processing with SAMGrid

Version 0.7

Preamble: As part of the ongoing ‘gridification’ of production activities at DØ primary processing is to be carried out with SAMGrid. To minimise the overall manpower effort this move has been made part of the SAM v7 transition. After reconstruction the data processing chain continues with separation into several ‘skims’ according to various physics signatures, so-called skimming. Currently this task is performed by custom scripts on CAB. To further reduce the person power needed to operate the reconstruction chain, skimming is the next task for ‘gridification’. This document details the requirements for including skimming into the primary processing production chain via SAMGrid. It follows informal discussions within DØ.

Requirements: Skimming uses a single (merged) thumbnail as input and creates multiple output files (called skims) all of which need to be kept. As skimming results in a significant reduction in the size of the output files, the many results need to be merged before being stored to tape. Only files of the same skim should be merged with each other.

It is of special importance to be able to recover from failures, that affect all or only a part of the output files. This recovery must be possible using information from SAM only, as logfiles in SAMGrid might fail independent of the actual status of the success in storing files. This also means that all output files need to be distinguishable by their metadata only and that the parentage information must be accurate.

Expected failure modes and special cases to treat:

Assumptions: Each grid jobs creates multiple batch jobs each of them will run on a single input file and produce all the corresponding skims in one go.

1. A grid jobs may fail in total, i.e. none of its batch jobs manage to create any output in SAM
2. A grid jobs may fail partialy, i.e. only some of its batch jobs create output in SAM.
3. Batch jobs may fail in total. None of the produced skims appears in SAM.
4. Batch jobs may fail partially. Some of the produced skims are stored to SAM others aren't.
5. Batch jobs may terminate early without notice. I.e. not all input events were considered for writing out the result.
6. Skims produced in a successful job may contain no events. This may happen occasional for skims with low selection rate. It may occur for extended periods of time when required triggers are turned off.

Possible implementation: Implementing skimming as two new job types (a skimming production step and a merging step) seems the only option. The skimming production job shall produce skims

for single inputfiles. These skims shall be stored to the sites durable location for later merging. Parantage information must be acurate to allows recovery. The number of events read by the executable shall be stored to the output skims metadata in addition to the events writen. This will allow to detect early termination of jobs.

Recovery is the big challenge as data duplication needs to be avoided. Probably the production of individual skims can't be turned off. Thus the recovery process must be able to drop files that already have merged siblings. It seems a good choice to let this filtering happen just before the merging step. D0repro tools should be extended to deal with this multiple output situation.

A natural separation between SamGrid functionality and d0repro functionality has so far been to let SamGrid deal with running the requested production, including at most cross checks of the validity of the request, while the d0repro tools are responsible to create the requests necessary to deal with a given production and the recovery from failed jobs. This separation should be kept for skimming.

Thus the bulk of development will be on the d0repro side.

File name conventions:

1. Names for intermediate files should be derived from input filename by prepending "Csshim-SKIM-" and postpending the release version used for skimmng, the global job id and the site name.

e.g.

CSskim-SKIM-recoT_all_..._p20.04.00_wicke_sammy.fnal.gov_062545_22621_FNAL

2. So far final (merged) output file names have the following scheme:

CSskim-SKIM-`date +%Y%m%d-%H%M%S`_p20.04.00

The last part is the version in which we did the skimming, not necessarily the reco version.

Having the skim name and the version in the filename are very useful, especially if you pull files out of SAM for testing locally...Otherwise you have no idea what they are.

The date doesn't seem to be a secure way of assuring uniqueness. Suggestions are welcome. Should we keep the version of reco and the version of the skimming in the filename?

Priorities: It makes most sense to put this system into production once the whole procedure, including recovery, is stable. Running recocert (without recovery) is of higher importance to DØ, but skimming takes priority over full error recovery in recocert.