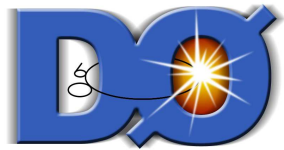


Operations: DØRepro Tools

Plans for the Future



Daniel Wicke
(Bergische Universität Wuppertal)



Outline

- Overview
- Feature
- Summary

Introduction

Support for additional production steps

- Production of RecoCert output (for data quality control)
- Skimming of Thumbnail (14 skims for different physics signatures)

Improvements to reduce operational work load

- Connection to Sam v7 Request System

RecoCert with Recovery

Conventional d0farm production runs recocert as part of processing raw-files.

- Large number of reco cert root-tuples produced
 - Huge number of small files (slow access in SAM)
 - need to be merged

Implementation plan

- Migrate current functionality to SamGrid (exists as part of TmbMerge).
- Add ability to recover missing recocert root-tuples
 - Need SamGrid jobtype to produce recocert only.
 - Need operational tools. (Extension of d0repro)

Operational tools

Should be simple extension of d0repro tools.

Help from DQ group asked.

Skimming

Skimming is a new class of jobs for our grid operations:

- One input file leads to many output skims.
- Each skim is merged with other skims of its kind.

⇒ New partial failure modes.

Requirements

Draft requirements documents summarise our understanding of the problem.

Error modes and special cases to treat

1. A grid job may fail in total: none of its batch jobs create output in SAM
2. A grid job may fail partially: 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 occasionally for skims with low selection rate. It may occur for extended periods of time when required triggers are turned off.

*All cases have to be handled and very well tested **before** we go into production!*

Failures may easily put ourselves in a situation where our only recourse is to redo the skims from scratch.

Suggested implementation

- Create two SamGrid job types
 - Skim production: Produces all skims for given input dataset.
Input dataset consists of merged thumbnails. Prototype exists
 - Skim merging: Merges given input dataset (separated by skims).
Input dataset consists of unmerged skimmed thumbnails.
Should be easy extension of existing merge jobtype
- Operational tools will have to deal with creating appropriate datasets.
This will be the big chunk

Current working model is

- Produce `sub_skimming` and `check_skimming` commands
will always reproduce all skims in case of failures
- Produce `sub_mergeskims` and `check_mergeskims` commands
Will have to deal with multiply produced skims (due to recovery)

How CD can help

Tools need to be extended to deal with multiply produced files (and merge only once).

New *_skimming commands need to be implemented.

Extensive testing required before it can be used for production.

Operational model layed-out above needs verification.

Full Request System

d0repro Documentation

```
set_status.py [merge|production] [new|approved|finished|...]
<dataset> <d0release> [--test]
```

Sets the status of a merge or production step to “new”, “approved”, ...

This emulates behaviour of the sam request-system

Features

d0repro tools keep track of more detailed request status:

- Request status and corresponding grid job status stored on local file system
- d0repro uses these status to prevent operator errors (esp. multiple submission)
- Can't prevent multiple submission from different operators
→ manual book-keeping

Utilising Sam Request System

To reduce manual book-keeping needed with multiple operators a central database for keeping the status of a production request required.

- Modify d0repro tools to store all status info in SAM request system.
- Thus allow any operator to work on any request without in advance manual assignment.
- Should significantly reduce overhead of running with multiple sites.

Required extensions

- Additional fields in `request_handler` to store *global_job_id* and *#input files*
- Elaborate request status to map existing implementation

Summary