Data Production for the Muon g-2 experiment

Renee Fatemi, Kim Siang Khaw, Liang Li, Adam Lyon

February 2, 2017

1 Production Workflow

This document outlines the workflow of the data production of the Muon g-2 experiment. There are two different production chain: simulation and DAQ.

1.1 Simulation production workflow

Simulation chain involves generating Geant4-based simulated data files, digitization of the truth information and reconstruction of the digitized information. Interaction between the gm2 instance and jobsub and SAM in the simulation chain is summarized in Fig. 1.

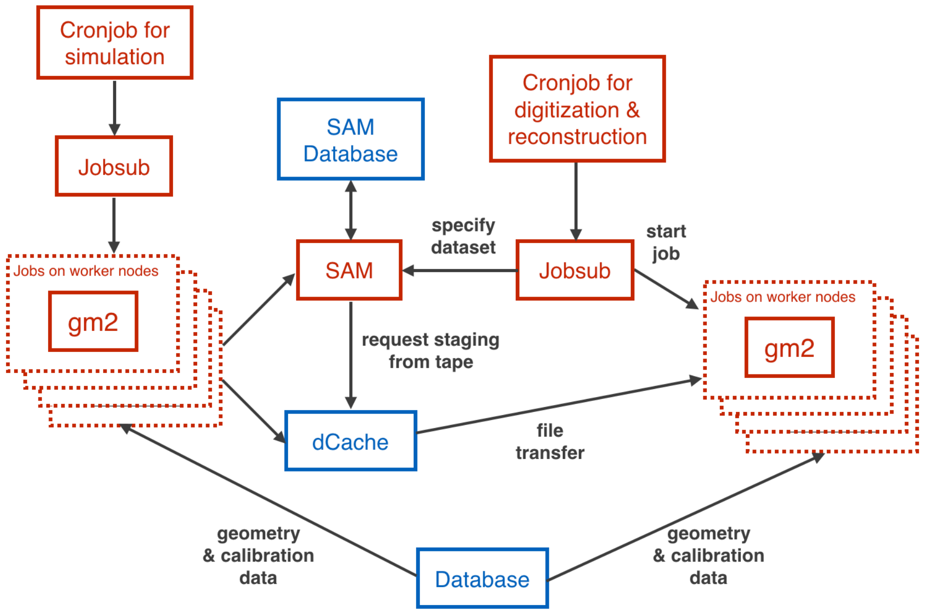

Figure 1: Workflow for the simulation production.

The following describes the basic steps in the simulation production:

1. A cronjob is setup to submit jobs to the FNAL grid at a specific time interval.
2. Worker nodes then execute the submitted scripts to generate simulated data. Database may or may not be used for the simulation.

3. Metadata of the generated data files are communicated to SAM data handling system and the files are transferred to FNAL permanent storage area.

4. Another cronjob independent of Cronjob1 is setup to submit jobs to the FNAL to digitize and reconstruct the simulated data.

5. Worker nodes then specify SAM dataset to be digitized and reconstructed.

6. The reconstructed data are then stored in the permanent storage area.

1.2 DAQ production workflow

Interaction between the gm2 instance and jobsub and SAM in the DAQ chain is summarized in Fig. 2.

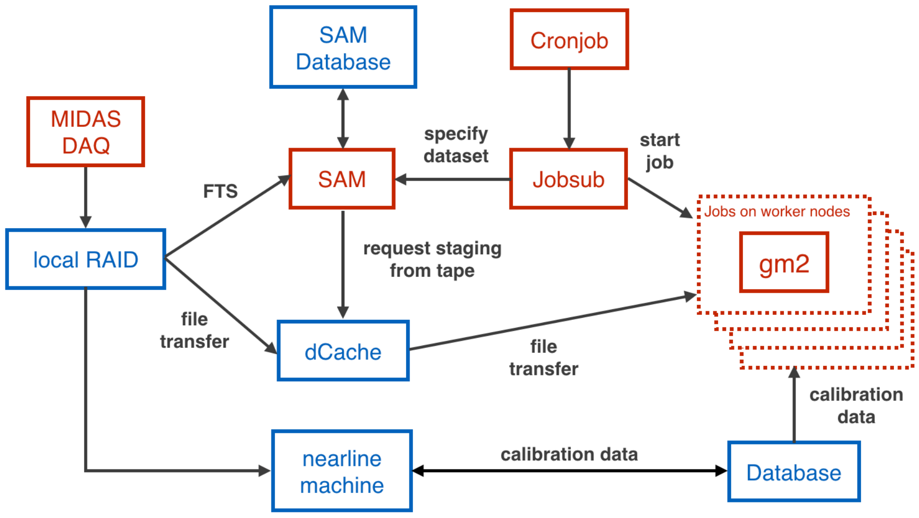

Figure 2: Workflow for the DAQ production.

The following describes the basic steps in the DAQ production:

1. MIDAS DAQ outputs raw data and stores them into local RAID storage.

2. A backend machine running FTS transfers raw files to the permanent storage area while communicates with SAM regarding the metadata of the files.

3. At the same time, a nearline machine analyzes specific calibration runs and extracts calibrations from these runs. All the constants are stored in the database.

4. A cronjob is setup to submit jobs to unpack and reconstruct the DAQ data.

5. Worker nodes then specify SAM dataset to be unpacked and reconstructed.

6. The reconstructed data are then stored in the permanent storage area.

2 SAM Metadata: Metadata definition and dataset definition

How to insert metadata using FTS, art, etc. What are the metadata we want to insert, etc.

3 Workflow of the data production: Implementation

Once the production team is identified and assembled, the various types of production should be consolidated. The productions fall naturally into categories of simulation and real data, with the later divided into commissioning and runtime productions.

I Simulations

  i MDC1 - uses muon gas gun to produce sample of 10^{10} muons (10% of muons be collected in the experiment).

  ii MDC2 - uses inflector beam gun to produce sufficient sample size to develop lost muon analyses and study optimal injection parameters for the kickers and quads. Detailed studies of beam dynamics and average spin effects will need much larger samples (TBD).

II Data

  i Commissioning

  - Mock Fast DAQ - data taken with no detectors attached. Useful for Q-method studies.
  - Laser - data taken in the months before beam arrives to test calorimeter, laser and DAQ system
  - Magnetic Field DAQ - data taken during commissioning period to exercise DAQ
  - Additional types to be identified ....

  ii Runtime

  - FAST DAQ
  - Magnetic Field DAQ
  - Fiber Harp
  - Additional types to be identified ....

Within these categories it will likely be necessary to make further distinctions, for example proton runs vs muon runs in the Runtime FAST DAQ category. This list will need to be worked out as the experiment progresses and updated here.

Of the productions listed above the MDC1 and Mock Fast DAQ are the most natural candidates to use for development and testing of the production workflow. The simulation framework in release *v*7_03 is production-ready and there are files on dCache from existing Mock Fast DAQ data. If production tools were in place then job submission could commence immediately. In fact small productions, on the order of $\sim$ million events, have already been produced for MDC1, and the entire SLAC test stand data has been produced.

In order to launch full scale productions the team needs to become proficient with the POMS tool, which will aid in tracking job submission. The complete set of metadata needs to be defined for SAM (see Section 2 above) and submission scripts need to be written (see Section 4 below). Eventually the team will also need to take advantage of the resources provided by the Open Science Grid, presumably this is the next step to take once production is running smoothly on FermiGrid.

Considering these tasks the production team has set the goal of launching, in parallel, MDC1 and Mock DAQ data productions by February 1st. The full production team is still being assembled and should be complete by the first week in December. The team will be composed of a Production Team Leader, Liang Li, and four additional members, one which of which will serve as an onsite deputy leader. The deputy leader, along with three other members will work with the leader to develop the POMS framework, SAM metadata and scripts. Half of the team will focus on the simulation productions (MDC1 to start) and the other half on the data productions (Mock DAQ).

4 Versioning of the scripts/codes (related to releases)

Production runs code from a release from CVMFS. The release identifier (e.g. `v7_03_00` must be stored in the SAM metadata for the produced files as application version). If a problem with a release is found during production, then release management is consulted and an emergency release may be made and uploaded to CVMFS. Out-of-band code should never be run in production.

Production scripts and the top level FCL files may be submitted with the job instead of pulled from CVMFS. This mechanism may be advantageous as these files may need to be updated at a faster rate than uploading to CVMFS can provide, especially in early phases of production. To accommodate this situation, a `gm2production` git repository may be set up to manage the scripts and top level FCL files. Scripts and FCL files that are actually run for production must be committed to git and pushed to Redmine. The git hash of the commit should be recorded in the SAM metadata for files produced. Eventually, when things are stable, `gm2production` should be released to CVMFS and necessary files pulled from there.

5 POMS: What do we know so far

The production team plans to use POMS to submit and track jobs and production campaigns. In order to use POMS you must be added as a g-2 user. Please contact the current production leaders if you are part of the production team and need to become a user. You must also have a .k5login script in your home directory on the FNAL virtual machines. If you don't already have a .k5login file create one and include both your (*user.principal*@FNAL.GOV) and POMS' kerberos principal (poms/cd/pomsgpvm01.fnal.gov@FNAL.GOV) on separate lines. A good introduction to the POMS system can be found at this link:

`https://indico.fnal.gov/getFile.py/access?contribId=8&resId=0&materialId=slides&confId=12120`

A snapshot of the POMS web interface is shown in Fig 3. Three levels of settings must be configured in order to launch a production campaign: *launch template*, *job type*, and *campaign layer*. They can be accessed from the left panel under the **Campaign Layers** as shown in Fig 3.

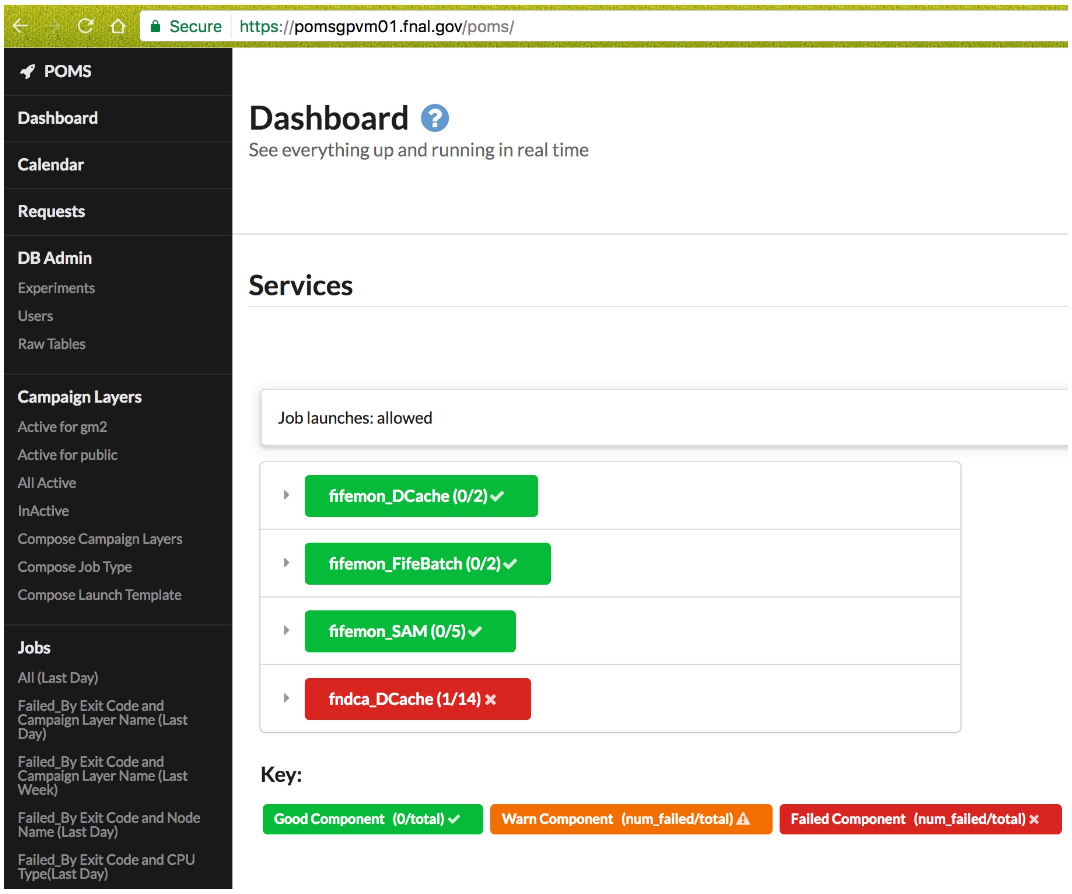
Secure https://pomsgpvm01.fnal.gov/poms/
POMS
Dashboard
Calendar
Requests
DB Admin
Experiments
Users
Raw Tables
Campaign Layers
Active for gm2
Active for public
All Active
InActive
Compose Campaign Layers
Compose Job Type
Compose Launch Template
Jobs
All (Last Day)
Failed_By Exit Code and Campaign Layer Name (Last Day)
Failed_By Exit Code and Campaign Layer Name (Last Week)
Failed_By Exit Code and Node Name (Last Day)
Failed_By Exit Code and CPU Type(Last Day)
Dashboard
See everything up and running in real time
Services
Job launches: allowed
fifemon_DCache (0/2) ✓
fifemon_FifeBatch (0/2) ✓
fifemon_SAM (0/5) ✓
fndca_DCache (1/14) ✗
Key:
Good Component (0/total) ✓
Warn Component (num_failed/total) ⚠
Failed Component (num_failed/total) ✗

Figure 3: A snapshot of the POMS web interface.

The *launch template* provides all the commands that POMS will need to setup the correct environment in your directory. Click on the *launch template* button and then select *gm2* under the *choose experiment* pull down menu. You should be able to see (and edit - but please don't) existing templates made by other users. Refer to the MD0_template as an example. To create your own template click the green **Add** button at the bottom of the page. In the pop-up window that appears fill in the following fields:

Name This is a descriptive name for your template. Launch templates can be used with several different job templates so make these names as general as possible. Including the release version in the title is a good idea.

Host This is the virtual machine you will launch your jobs from, for example *gm2gpvm02*.

Account This is the name of your account on the virtual machines where you will launch jobs.

Setup This field contains every single command you need to setup your environment and maneuver to the correct directory from which you will launch your jobs. Commands should not be separated by newlines (returns) but instead by semi-colons.

Once you have filled in the template press Submit to exit. You may return to edit the template by clicking not the leftmost icon next to your template name in the list.

The *job type* template holds the job submission command. Refer to MDC0_template as an example. To create your own template click the green **Add** button at the bottom of the page. In the pop-up window that appears fill in the following fields:

Name This is a descriptive name for your template. Job templates can be used with several different launch templates so make these names as general as possible. Including the job fcl in the title is a good idea.

Input Files Per Job It is not clear how important this field is or if POMS even uses the information. The fcl is considered an input file so it is always appropriate enter one in this field.

Output Files Per Job The number of output files from this job. It is not clear how important this field is or if POMS uses the information.

Output File Patterns This field denotes the output file format. It is not clear how important this field is or if POMS uses the information.

Launch Script Input the full path to the submission script.

Definition Parameters Enter inputs to the script.

Recovery Launches Leave blank. g-2 is not currently using the recovery functionality.

Once you have filled in the template press Submit to exit. You may return to edit the template by clicking not the leftmost icon next to your template name in the list.

The *compose campaign layers template* associates a specific *launch template* and a *job template* and provides the hook you will use to launch jobs. To create your own template click the green **Add** button at the bottom of the page. In the pop-up window that appears fill in the following fields:

Name This is a descriptive name for your template. Including a short descriptor of the ultimate physics goal in the title is a good idea.

VO Role Enter “production”

State Pull down “Active” if this production is ongoing. If the campaign is over switch to “Inactive”.

Software version Enter the release version for this production

Dataset Enter a description of the dataset you are running

Dataset Split Type Enter “mod(25)”

Completion Type This field is not required. It is not clear at this time the difference between “Complete” and “Located”.

Completion Pct This field is not required. It is not clear at this time how POMS uses the completion %.

Parameter Overrides Leave empty

Depends On Leave empty

Launch Template Choose from available g-2 *launch templates*

Job Type Choose from available g-2 *job type templates*

Once you have filled in the template press Submit to exit. You may return to edit the template by clicking not the leftmost icon next to your template name in the list.

Once you have filled out the *launch template*, *job type*, and *campaign layer* click on the *Active for gm2* tab and click on the Name of the *Campaign Layer* you would like to launch. When you click on the *Launch Campaign Jobs Now* tab POMS will attempt to submit the job through your account. The pop-up screen will show you the submission response so you can evaluate of the submission was successful. If the job were not successfully launched then you need to debug your templates. A good way to do this is to copy and paste the commands from the template in your home directory, modifying and updating your templates as you debug the problems.
Additional documentation by the POMS group is being developed here :

https://cdcvs.fnal.gov/redmine/projects/prod_mgmt_db/wiki/How_to_use_POMS

A Introduction to FIFE tools

This section serves as a summary of the detailed explanation given in https://cdcvs.fnal.gov/redmine/projects/fife/wiki/Introduction_to_FIFE_and_Component_Services regarding the tools provided by Fermilab for offline computing.

A.1 What is FIFE?

FabrIc for Frontier Experiments (FIFE) is the central tools and services provided by Fermilab to address common challenges in offline computing. FIFE is developed based on the collective experience from current and past experiments to provide options for designing offline computing for experiments like Muon g-2. FIFE is modular so experiments can take what they need, and new tools from outside communities can be incorporated as they develop.

A.2 Jobsub

Jobsub (Job Submission) is a suite of tools to manage batch/grid submission. These tools are designed to simplify the job submission process by:

- Defining common interfaces for experiments
- Integrating complex grid tools in a sensible manner
- Protecting shared resources from overload

A commonly used command for submitting a job is as the following:

```
jobsub_submit -G gm2 -N ${NJOBS} -M --OS=SL6 \
--resource-provides=usage_model=DEDICATED,OPPORTUNISTIC \
--role=Analysis --dataset_definition=${SAM_DATASET_NAME} \
file://$PWD/submit-localrelease.sh ${MAINFCLNAME} ${USER} ${SCRATCH_DIR}
```

Notice that the experiment name (gm2), role (Analysis or Production) and so has to be specified to successfully submit a job. To check the progress of your jobs,

```
jobsub_q -G gm2 --user=${USERNAME}
```

A.3 IF Data Handling Client Tools (ifdhc)

IFDH (Intensity Frontier Data Handling), is a suite of tools for data movement tasks for Fermilab experiments. IFDH encompasses moving input data from caches or storage elements to compute nodes (the ”last mile” of data movement) and moving output data potentially to those caches as part of the journey back to the user.

IFDH is:

- easy to use.
- does ’The Right Things’ for data transfer, and avoids things that cause phone calls stating,”Your jobs are hanging the BlueArc.”
- changes with the updated environment so users don’t have to keep changing scripts. (For example, when the new UberCache replaces dCache, it will be supported automatically in ifdh, and related command.)
- can be called from python scripts, plain C++ code, and art framework code.

To copy a data file stored in the pnfs area to a local area, simply type

```
ifdh cp /pnfs/GM2/mdc/mdc0/gm2ringsim.root /gm2/app/users/XXX/mydata/
```

Notice that the typical UNIX command for copy (cp) will work but it is not safe and your file might be corrupted.

A.4 SAM

Sequential Access via Metadata (SAM) is a data handling system organized as a set of servers which work together to store and retrieve files and associated metadata, including a complete record of the processing which has used the files.

As much as possible, SAM frees experiments from dealing with the specific details of their data so they can focus on the content rather than the technicalities. SAM provides:

- metadata catalogue — What's in the data files?
- location catalogue - Where are my files?
- data delivery service - Give me my files.

There are several summary talks regarding SAM given in our simulation meeting and collaboration meeting.

- NOvA SAM submission scripts review by L. Li (`http://gm2-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=3550`)
- SAM Review by L. Welty-Rieger (`http://gm2-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=3415`)
- Simulation Status and Data Handling with SAM by L. Welty-Rieger (`http://gm2-docdb.fnal.gov:8080/cgi-bin/ShowDocument?docid=3142)`