Production Operations Management Service

Requirements Document

Authors: Paola Buitrago, with Robert Illingworth, Marc Mengel
Version: 1.0
Date: 5/10/2015

This document describes the requirements to implement the Production Operations Management Service to assist Intensity Frontier Experiments and the Offline Production Operations Service group to manage and monitor Production processing jobs. The tool will allow experiments to fill requests for data processing or Monte Carlo generation. Analysis tasks are explicitly out of the scope.

Revision History

Document History		
Version	Date	Comments
1.0	May 10, 2015	Initial version
1.1	June 5, 2015	First released version
1.2	June 30, 2015	Sign-off version

Table of Contents

Executive Summary

The Production Operations Management Service will allow the Offline Operation Service group to provide a more comprehensive and automated service for Intensity Frontier experiments. It will assist in the submission, monitoring and triaging of errors of production processing tasks. It will also work as an interface between the OPOS group and the experiment by providing the interface to place a new processing request and monitor the advance of a particular workflow, as opposed to submitting requests through SNOW.

Requirements Summary

The Production Operations Management Service is expected to assist in the different stages of processing a production request for an experimenter. Starting with the registration of a new request, approval, pre-launch checks, submission, monitoring and failure triage.

Assumptions, Risks, Dependencies

The system needs to inter-operate with SAM and jobsub; however it should interact with them through an abstraction layer so as to not be overly dependent on them. It should also use use existing experiment-specific job launch scripts to maintain experiment internal bookkeeping.

The system should not duplicate information or data available in already existing monitoring systems such as Gratia and FIFEmon. In this sense, the system would need to use the same sources of information of these systems or keep a summarized version from their data.

It is desirable to have the selected solution be accessible from multiple platforms including Mac, PC and Linux platforms.

Scope of the Project

The scope of this system is to allow the OPOS Group to manage and monitor their data-analysis and MC-processing jobs, and for the experiments to request processing tasks and MC event generation. It also will maintain information to assist in diagnosing/triage-ing job failures to route failed job information either to the experiment or appropriate service managers. Non-production jobs from experiment users are explicitly excluded from the current scope, however future requests may come to expand this to cover non-production jobs. Furthermore, this system should not duplicate the project tracking facilities in SAM, nor the service monitoring features in FIFEMon and Gratia. Rather, this system should refer to data in those other systems, or possibly keep summarized data from those systems.

Stakeholders

The following stakeholders have been identified:

- Lab management
  - Scientific Computing Services Head.
  - Scientific Distributed Computing Solutions Head.
- Experiment production groups
  - NOvA production group.
  - MINERvA production group.
  - Minos+ production group.
  - Microboone production group.
  - Mu2e production group.
  - G-2 production group.
  - Other experiments, as needed[1]
- Projects
  - Offline Production Operations Service group (OPOS).
- Support personnel
  - FIFEmon support team
  - REX-DH support team
- Oversight and Review
  - NOvA data quality group.
  - MINERvA data quality group.
  - Minos+ data quality group.

Roles

We define six roles for people who are actively involved in the Production Operations Management Service:

OPOS operator – A person who is responsible for fulfilling a particular production-processing request. This person would use the tool to review queued production requests, check request input information is consistent, initiate a request, monitor its advance, triage any problem and report back to the experiment.

OPOS supervisor – A person who leads the OPOS group and would monitor in a high level the progress of the experiment production processing. This person would require the generation of daily, weekly and overall reports.

Experimenter – A person from an experiment, typically, who is from the production group. This person can define a new request and register it with the Production Operations Management Service. He also can monitor the status of his/her experiment production.

---

1 New experiments will be added as they begin having the OPOS group do production processing for them.

Experiment approver – A designated person or group of people from an experiment who will have the ability to approve or reject an experiment request. This person can also prioritize the experiment's requests that are still in queue.

Reviewer – A person who may be internal or external to an experiment who is called upon to review aspects of the processing task executed with the aid of the Production Operations Management Service. It might include checking data quality or data reproducibility.

Guest – A person non-necessarily related to an experiment production group, the OPOS team or the supporting team. This person would want to get an overview of how production processing is doing. This can be a person from the general lab population. We may need to restrict viewing data by experiment membership.

Service provider – A person who provides support to the Production Operations Management Service. Service providers include:

- Data handling group.

Definitions, Acronyms and Abbreviations

Task – A task is a discrete processing unit of work, for example "process a single day's raw data", or "generate 10000 Monte-Carlo events for this decay process". A task may consist of one or more subtasks. Within a task there's a dependency between the subtasks – the next subtask should not be run until the previous dependencies have been completed. A task can be associated to a particular type of processing: Reco, MC, Calibration, etc.

Subtask - A single subtask may be split into multiple batch jobs.

Task completion criteria – Examples of this include: "the task has completed the number of MC events requested", "all the input files have been consumed", etc.

Campaign – This is a type of task. Example of campaign: "Process all experiment data with version 9 of reconstruction", "run all 6 phases of MonteCarlo with specified versions of phases for a billion events"

Batch – This is a type of task. Small portion of a high level task to be submitted at once. Batches have a workflow matching campaign, but work on their subset of the data.

Batch task completion criteria – Some examples of batch-level task completion criteria would include: "output datasets reaching tape", "output datasets reaching some count as input dataset", "check script passing on output data and/or logs" or "experiment check-off saying it is finished".

Keep Up processing mode – Processing of production data that is done in a periodic fashion. Typically it would involve daily processing of new available data.

References

The content of this document is based on a series of meetings held between the developing group of the Production Operations Management Service and relevant stakeholders. The list of meetings held is:

Requirements gathering meeting	
Date	Meeting name
03/09/2015	OPOS group requirements gathering meeting
03/20/2015	CMS operations group requirements gathering
03/24/2015	Minerva production group requirements gathering
03/26/2015	Minos+ production group requirements gathering
03/27/2015	Microboone production group requirements gathering
03/31/2015	Nova production group requirements gathering
04/07/2015	Mu2e production group requirements gathering
04/20/2015	G-2 production group requirements gathering

Meeting notes and relevant information concerning this project can be found in the Production Operations Management Service Redmine project:
https://cdcvs.fnal.gov/redmine/projects/prod_mgmt_db

Some documents that also serve as source of information are listed as follows:

Referenced documents			
Document No.	Document Title	Date	Author
TBD	PUBS: Data Processing Software Framework	Dec 30th, 2014	Kazuhiro Terao
DocDB #5418	OPOS Service Level Agreement	Jan 28th, 2015	Andrew Norman
TBD	Requirements for Production Operations Management	Jan 5th, 2015	Michael Diesburg, Robert Illingworth, Andrew Norman

	Service		

Business Process Flow Diagrams

The workflow to process a production request is depicted in the diagram shown in Figure 1.

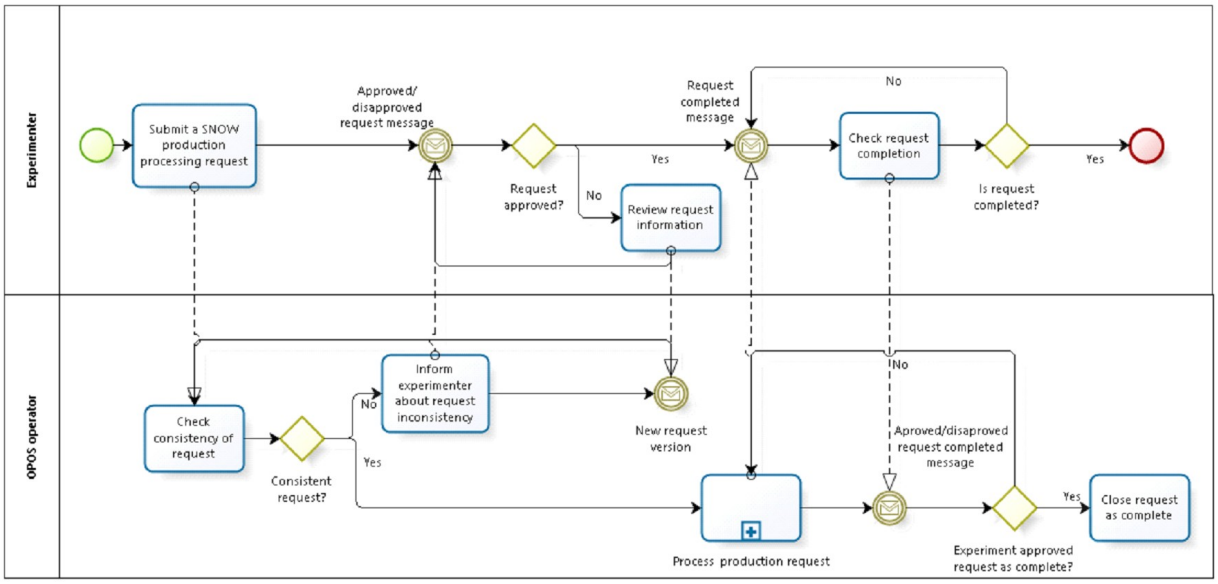

Figure 1. Process to fulfill a production request by the OPOS team

The process starts when an experimenter fills a new request using the Service Now system. The person that fills the request must be a member of the experiment that acts as an OPOS liaison. It is expected that the request contains at least the following information:

- Files to process - Depending on the tools used by the experiment, this can be defined with a text file, a sam definition, sam dimensions, among others.
- Scripts - Includes the setup and submission script, and the executable to run in each grid node. It can also include configuration files depending on the framework and experiment workflow.
- Expected processing time per file.
- Experiment software release.
- Priority.
- Special requirements: This might include: naming the output files with a special pattern, setting a special value in the metadata of output files, among others.

Once the request is filled, The OPOS supervisor does an initial check on the request, to make sure it is in line with the agreed to work and that basic requirements are met, and then assigns the work to an OPOS operator, who checks if the request is consistent, assisted where possible by

the tool. This includes: checking that the required scripts are in place, the input file sets are available and correctly defined, the processing won't lead to undesired issues like duplicated files, etc. If necessary, the supervisor or operator will require the experimenter to complete the request information and/or check its consistency. Once the request is approved by the OPOS staff, what comes next is to actually proceed to process the request. This includes the activities depicted in Figure 2.

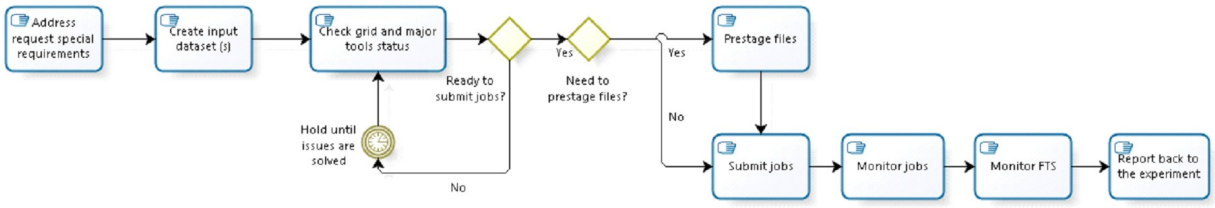

Figure 2. Activities followed by OPOS operator to fullfil a production request.

If the request contains any special requirements, that will be addressed in the first step. Provided the special requirements have been addressed, the OPOS operator manually creates the dataset or input datasets to be processed. In parallel, the operator would check, using tools as FIFEmon and Gratia, whether the grid and some major tools are in good condition to submit. If the grid is too busy or there is a known problem with one of the major tools, the operator will hold until a proper scenario for submitting is present. After this, the operator will pre-stage the input files if needed in order to avoid job inefficiency. Next, actual submission would be done using the submission script provided by the experiment. Monitoring tools like the condor queue or the sam station, the operator would monitor the state and progress of the jobs. Once jobs are complete, condor logs would be fetched and scanned in order to determine if the processing was successful or not. Next step in the process would be to monitor output files as they are registered with SAM and taken to tape by an FTS instance. During the whole process, several types of errors can occur which would require further manual inspection. The OPOS operator will inspect until finding whether the error was caused by some tool failure or due to some bug in the experiment provided scripts.

Currently, the process is highly manual and dependent on the OPOS operator which limits the extend to which the OPOS team can scale in term of the number of experiments on boarded and number of request handled. The goal of the Production Monitoring Database would be to provide a tool that will automate several activities in this process and will require low operator intervention. In the cases where the operator intervention is required, the Production Monitoring Database is expected to assist and by making the tasks less error prone and less time consuming.

Functional and System Requirements

Task Requirements

The following are the requirements involving the management and execution of production tasks of different granularity levels.

Number	Requirement	Source	Priority
T1	The solution shall support campaigns, which may involve multiple dependent tasks/workflows	POMS Team – CMS meeting	
T2	The solution shall support the following campaign states: new, pending approval, approved, running, and succeeded/failed.	POMS Team	
T3	The solution shall support tracking batches including information like sam project progress, job exit codes, and output files handled by the FTS.	POMS Team	
T4	The solution shall implement a mechanism to break down campaign datasets into batch-level datasets.	POMS Team	
T5	The solution shall provide an interface to specify the task completion criteria.	Req-PMD - 01/15	
T6	The solution shall support the following batch-level task states: pending, submitted, project running (if sam project involved), jobs completed, batch completed.	POMS Team	
T7	The solution shall allow the configuration of periodic batch-level tasks to satisfy keep up processing mode.	POMS Team	
T8	The solution shall execute tasks automatically by submitting the required jobs.	POMS Team	

Monitoring Requirements

The following are the requirements involving the monitoring of tasks.

Number	Requirement	Source	Priority
M1	The solution shall show the progress of a campaign in terms of percentage of event, files or jobs complete (whichever matches better the particular task)	POMS Team	
M2	The solution shall show the subtasks and specific jobs associated with a task	POMS Team	
M3	The solution shall let the user query the amount of time a job has been in it's current state	POMS Team	
M4	The solution shall show the last time an experiment job run in a particular site	POMS Team	
M5	The solution shall trace the workflow of processing tasks from creation to completion	POMS Team	
M6	The solution shall monitor the advance of processing subtasks	POMS Team	
M7	The tool shall record the current state of tasks and subtasks	Req-PMD - 01/15	
M8	The tool shall record the transition of state for every task and subtask including a timestamp and reason for the change	Req-PMD - 01/15	
M9	The tool shall record information about each particular job, including facility, jobID, and type of system it ran on.	Req-PMD - 01/15	
M10	The tool shall keep track on the ongoing state (staging data files, processing, transferring output, etc.) of the job and the state transition.	Req-PMD - 01/15	
M11	The tool shall keep information about jobs for at least 30 days after they complete to support post-mortem diagnosis.	POMS Team	

PreLaunch Requirements

The following are the requirements actions to take before launching grid jobs.

Number	Requirement	Source	Priority
P1	The solution shall check if the major services required for launching jobs are up/down before submitting (manually or automatically)	POMS Team	
P2	The solution shall check if the software required to execute a job is available before submitting (manually or automatically)	POMS Team	
P3	The solution should check job queues for an experiment or a particular role, as required.	POMS Team	
P4	The solution shall check the input information for a submission and make sure they are consistent and feasible before allowing it to be submitted to OPOS, and/or queued to run.	POMS Team	

Job Submission Requirements

The following are the requirements related to launching grid jobs.

Number	Requirement	Source	Priority
J1	Batch-level tasks shall be submitted automatically when dependent-upon tasks complete their running phase	Req-PMD - 01/15	
J2	The solution shall support using experiment's specific submission scripts to launch jobs.	POMS Team	
J3	The solution shall support grid authentication with suitable proxies	POMS Team	
J4	The solution shall store information about who or what created a task and the type of task	POMS Team	
J5	The tool shall store and be able to use information about how to execute a subtask (which script to execute and how)	POMS Team	
J6	The solution shall store all information necessary to reproduce the task at a later date (code release versions, input dataset, generator data, random seed (MC case), ...	Req-PMD - 01/15	
J7	The tool shall include free format text in requests to store and retrieve experiment specific task configuration information	Req-PMD - 01/15	
J8	The tool shall be able to manage DAG type dependencies between subtasks	Req-PMD - 01/15	
J9	The system shall support different	Req-PMD -	

	experiment-specific job submission scripts.	01/15	
J10	The tool shall be aware of service availability (i.e batch queue length, SAM service status, etc.) and hold job submissions if services are overloaded or unavailable.	Req-PMD - 01/15	
J11	The tool shall support job submission to all jobsub-available slot types (i.e offsite, onsite, opportunistic, etc.)	Req-PMD - 01/15	
J12	The tool shall offer and approbation mechanism for the tasks that require it.	Req-PMD - 01/15	
J13	The tool shall allow certain user roles to authorize/approve tasks.	Req-PMD - 01/15	
J14	The tool shall allow certain user roles to prioritize the tasks that are in the queue.	Req-PMD - 01/15	
J15	The tool shall allow individuals or groups within the experiment to create a new request.	Req-PMD - 01/15	
J16	The tool shall queue the incoming tasks before start processing it.	Req-PMD - 01/15	

Diagnostics Requirements

The following are the requirements related to triaging failing jobs and recovering from them.

Number	Requirement	Source	Priority
D1	The solution shall provide the following information for failed jobs: condor exit code, job log files, related SAM logs, related FTS logs, related ifdh logs and status of major services during the job execution.	POMS Team	
D2	The system shall implement a mechanism for black hole detection and remediation	POMS Team	
D3	The tool shall provide a mechanism to resubmit the failed parts of incomplete tasks.	Req-PMD - 01/15	

Request Workflow Requirements

The following are the requirements related to the registration, prioritization and approval of new requests.

Number	Requirement	Source	Priority
W1	The solution shall allow the registration of job approvers per production stage	POMS Team	
W2	The solution shall offer an interface for approver to accept, reject or, possibly, consolidate requests	POMS Team	
W3	The solution shall present to approvers a list of request pending to be approved	POMS Team	
W4	The solution shall allow the authorized experimenters to change priority of queuing requests	POMS Team	
W5	The solution shall keep history of already completed requests	OPOS Team	
W6	The solution shall allow an experimenter to clone an already existing/completed request	OPOS Team	

System Requirements

Number	Requirement	Source	Priority
S1	The solution shall offer a web interface, a command line interface and a scriptable interface to support automation.	Req-PMD - 01/15	
S2	The web interface of the solution shall provide monitoring and management functionality.	Req-PMD - 01/15	
S3	The solution needs to be available 9x5 for the OPOS group.	OPOS Team	

Computing Policy Requirements

Number	Requirement	Source	Priority
CP1	The solution shall comply with Fermilab Computing Policies.	General	