# Production Management System for AMS Remote Computing Centers

V. Choutko[1], O. Demakov[1], A. Egorov[1], A. Eline[1], B. Shan[2], R. Shi[3]

[1]Massachusetts Institute of Technology [2]Beihang University [3]Southeast University

## Abstract

The Alpha Magnetic Spectrometer (AMS) on board of the International Space Station (ISS) requires a big amount of computing power in data production and data simulation. A large fraction of the computing resources has been contributed by the remote computing centers. AMS has 12 such centers, with different hardware and software configurations.

This paper presents a production management system for remote computing centers, including job acquiring, submitting, monitoring, transferring, accounting, etc. The design principle is modularized, light-weighted, and easy-to-be-deployed. The system is based on Deterministic Finite Automaton, and implemented by script languages, Python and Perl, and the built-in sqlite3 database on Linux operating systems. Different batch management systems, file system storage, and transferring protocols are supported. The details of the integration with Open Science Grid is presented in this paper as well.

## Objectives

To facilitate the production management in the remote computing centers, a production management system has been designed and implemented, which features:

- Light-weighted
  - Does not require communication between processes
  - Easy to deploy and configure
- Adaptable
  - Can use different batch systems, storage systems, transferring technologies, etc.

## Design highlights – deterministic finite automaton

The key goal is to have the highest possible efficiency for the production, i.e., to minimize the idle time of all working nodes.

To achieve this, we design the system using the Deterministic Finite Automaton model, which ensures the computing resources are always filled by production jobs.

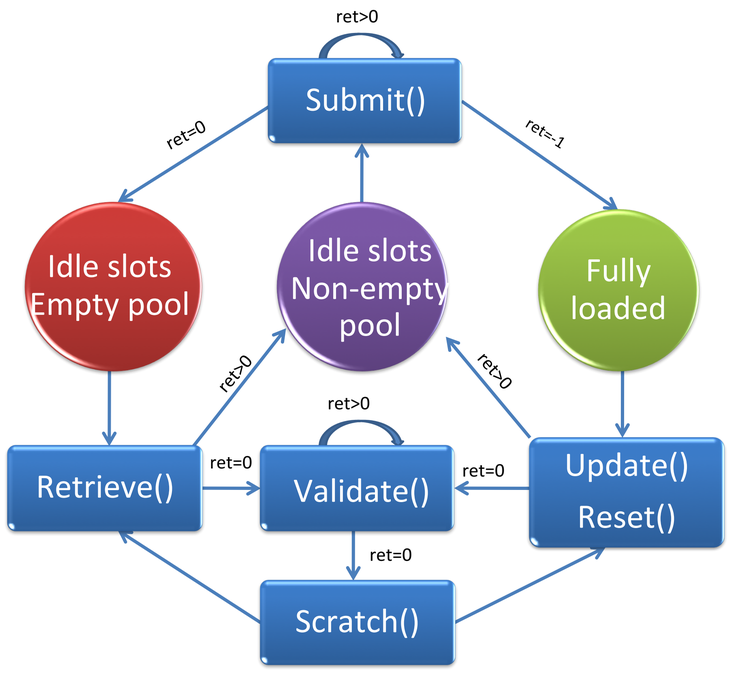


Figure 1: The production management system is designed based on the DFA model.

## Design highlights – Customization

Different computing centers usually have different configurations on hardware, software, storage, etc. To adapt the variations, we peel all the configuration related functions off the codes, and put them into a configuration file of environment variable definitions. This includes not only traditional "configurations", like the production home directory (AMSPROD_HOME), and the thread pool size (AMSPROD_QUEUE_SIZE), but also some production actions. For example, for job submission, we define an environment variable AMSPROD_CMD_SUBMIT, where we can define the real command (template) of job submission. In this case we don't have to redesign the code when changing batch system from LSF to condor.

## Design highlights – diagnostics on resources

Open Science Grid (OSG) provides facilitates access to distributed high throughput computing for research in the United States, and we started to run production jobs on OSG since February 2016. As OSG has over 100 individual sites and a variety of resources, it is important to identify specific failing/problematic nodes/sites and exclude them from job execution. This system provides automatic diagnostics on log files: it records the statistics of each job execution in the database, including the host and site, the status, etc. In case a job execution fails, it will analyze the log file and record the failing reason(s) as well, and:

1. Create a "blacklist" consisting of the recent "unreliable" sites;
2. Notify the administrator about the site-specific issues.

For example, in case all the jobs executed on site A in the past hour failed due to CVMFS issue, site A will be put into the blacklist in OSG submission script and a notification mail be be sent to the administrator.

## Performance and applications

The following plots show the completion and the queue status as a function of time when we reconstructed our science data at the computing farm JUROPA (now renamed to JUAMS) in 2012.

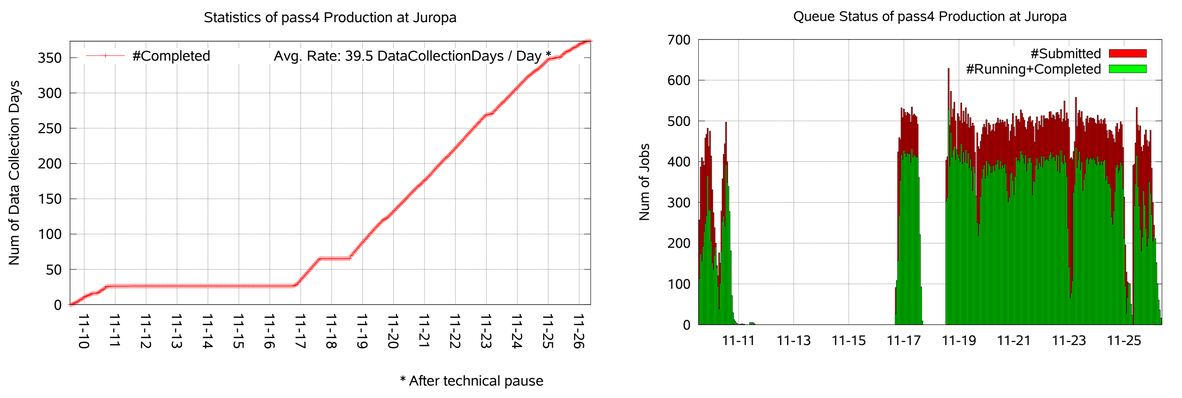


Figure 2: The completion and queue status of the pass4 data reconstruction in 2012.

The following plot shows the completion as a function of time when we reconstructed our science data in 2015 and a Monte-Carlo simulation campaign in 2016.

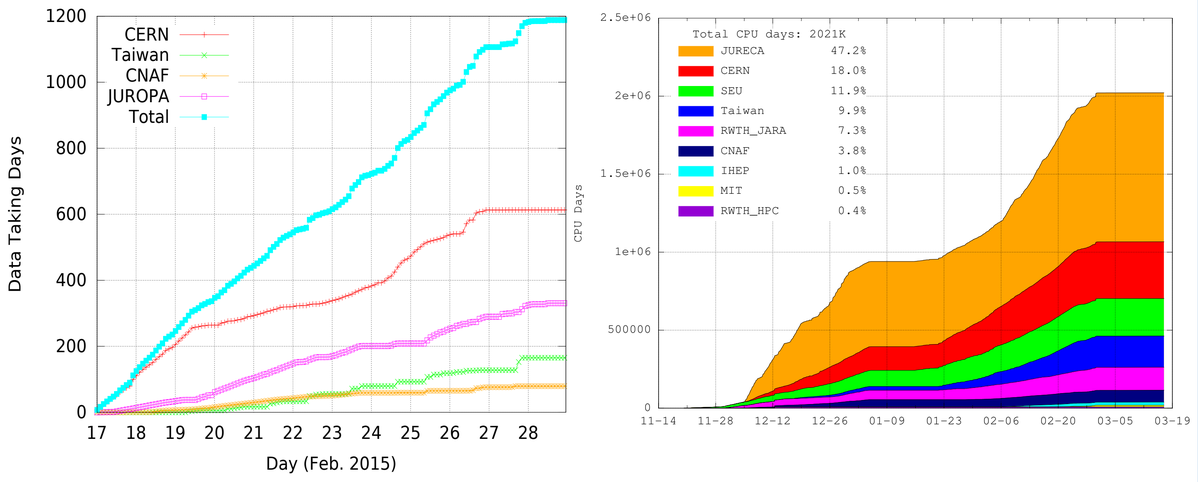


Figure 3: The completion status of pass6 data reconstruction in 2015 and MC simulation in 2016.

The described production system has been deployed and in JUAMS Germany, CNAF Italy, IN2P3 France, IHEP, NLAA China, CERN, and MIT (OSG) USA.

## Conclusion

The production management system for remote computing centers is designed using the deterministic finite automaton which ensures the computing resources are always filled by production jobs. It is implemented by script languages and Linux built-in database, which makes the deployment and customization fast and easy.

The recent development of the system includes the integration of OSG resources, and automatic diagnostics on OSG sites and hosts.

The system has been deployed in most of AMS remote computing centers and used for daily production management.