COMPASS Production System Overview

Artem Petrosyan[1,*]

[1]Joint Institute for Nuclear Research, 141980 Joliot-Curie 6, Dubna, Russian Federation

Abstract. Migration of COMPASS data processing to Grid environment has started in 2015 from a small prototype, deployed on a single virtual machine. Since summer of 2017, the system works in production mode, distributing jobs to two traditional Grid sites: CERN and JINR. Now the infrastructure of COMPASS Grid Production System includes 6 virtual machines, each is reserved for one production service: database, PanDA, Auto Pilot Factory, Monitoring, CRIC information system and, finally, production system (ProdSys) management instance, which provides a user interface for production manager and hosts services of automatic processing. Support of COMPASS virtual organization is provided by CERN IT. CRIC is also deployed at CERN Cloud Service. Other ProdSys services are deployed at JINR Cloud Service. There are two storage elements at CERN: EOS for short-term storage and Castor for long-term storage. During last year, along with providing a 24/7 service, the system was instrumented by many features, which allow automating data processing as much as possible. Recently, Blue Waters HPC has become a member of the computing infrastructure of the experiment. Details of implementation, workflow management, and infrastructure overview are presented in this article.

1 Introduction

COMPASS [1] depends on CERN IT services to store and process data. During the experiment lifecycle (data taking of the experiment has begun in 2002), some of IT services have become obsolete and during the nearest years the following services are going to be replaced by more modern ones: Castor [2] by EOS and CTA [3], IBM LSF by HTCondor [4], AFS [5] by EOS Filesystem in Userspace (FUSE). Such process of gradual replacement of computing infrastructure components strongly influences data processing of the experiment and triggers changes in software components, which interact with computing site, data, conditions, and metadata storage. In order to ease the consequences of current and future infrastructure changes, the computing model of the experiment must be adapted accordingly: it must unify a set of independent services. Support of several computing sites and distributed jobs submission may be performed by adding a Workload Management System (WMS). Availability of experiment software releases on any remote computing site can be achieved by installing them to CVMFS [6]. Thus, a Grid infrastructure of the experiment must be created. Running jobs via these additional layers allows the production system administrator to add or remove computing sites online by

[*] Corresponding author: artem.petrosyan@jinr.ru

simply changing configuration without any changes in the computation process. If the site goes into downtime, jobs do not reach it, and processing concentrates on other sites of the infrastructure. Usage of WMS allows treating various site resource managers as one computing queue.

COMPASS collects 0.5-1.5PB of raw data per year (Figure 1). Files with raw data are stored on Castor at CERN. Amount of files in one task (a set of jobs with the same execution parameters) usually does not exceed 50 000. Size of the raw data file is ~1GB. So, the size of all raw data files in the task is ~50TB. Such relatively small data volume, in combination with one central storage and good network connectivity between storage and computing sites, allow organizing data processing on Grid resources using a model of diskless computing sites. In this case, all data are stored at one single storage element (Castor), and being distributed to the worker nodes on computing sites at the moment when the job starts. CERN HTCondor and JINR Cream CE computing sites are used for data processing at the moment. Job results are being sent to EOS when the job is finished. Then, after merging, job results are being moved to Castor for the long term storage. Data flow of the processing is presented in Figure 2.

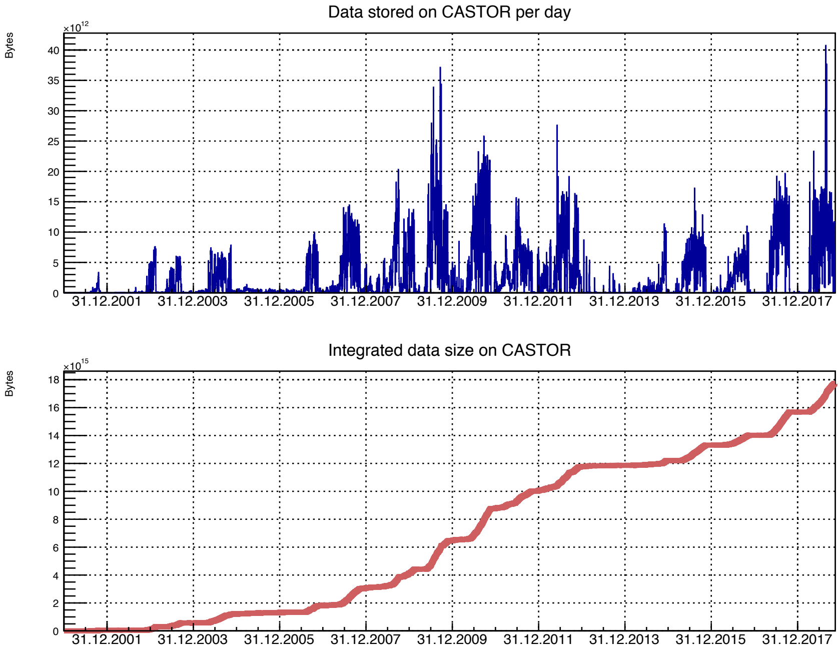


Fig. 1. COMPASS data on Castor: data stored per day (top) and integrated data size (bottom).

2 Production System components

The COMPASS Production System [7] is designed to provide automatic data processing on a set of heterogeneous computing resources. To ensure smooth processing on a variety of distributed resources, the production system is organized as a set of independent layers:

1. The task request layer is a web application, where production manager can define tasks, jobs and their parameters, and control their progress;
2. The job definition layer is an automatic process, once task is defined, jobs are being generated automatically;

3. The job execution layer is PanDA [8], responsible for delivery and manage jobs execution on the worker nodes;
4. The workflow management is a set of services, running on ProdSys machine, providing automatic data processing in accordance with task and job parameters;
5. The data management is a set of services, responsible for data pre-processing and post-processing;
6. The monitoring: system is instrumented with rich monitoring, with each component providing monitoring metrics, and allow to track tasks and jobs states at any time during the processing, and, also, health of the system in general.

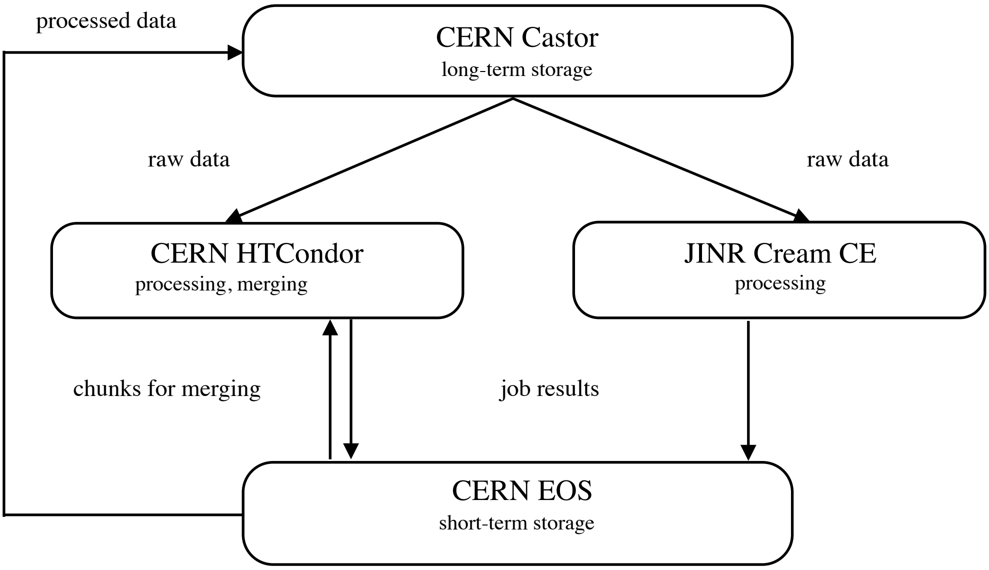


Fig. 2. COMPASS data processing work flow.

2.1 Task request layer

Task request layer is a web application, built on top of Django Admin Engine [9]. Tasks and jobs are declared as Django models. The application works on MySQL database and Apache web server. In the task request web interface production manager may define a task as a set of parameters, such as name, type, site, home, path to software installation, which will be used to process data, software release, name of production, year, period, parameters of software release, name of the configuration file of software release, files or runs list to be processed, number of attempts to be used. Once all necessary fields are filled with data, the production manager changes the status of the task from “draft” to “ready” and automatic processing of the task begins.

2.2 Job definition layer

The jobs are generated automatically, depending on a “files source” parameter, defined by the production manager. If files are defined as a files list, then jobs generator simply generates a job per each line. If files are defined as a runs list, then jobs generator first requests another script to prepare a list of files for each run in the list. Then it generates a job for each returned file of the run. The source of files list may be Oracle database for rarely processed years, for more often processed years list of files for runs are presented as a set of configuration text files, stored on CVMFS. In case of configuration files, a number of events in a raw file are also imported synchronously with the job generation. The number of events is used later to control that job was processed correctly. For old files, taken from the Oracle database, the number of events is filled by a separated process, after jobs generation.

As soon as all jobs are generated, the system changes task status to “jobs ready”. At this point, the production manager may check and ensure that all the jobs were generated correctly and apply any necessary changes. If all jobs are OK, production manager changes task status to “send”.

2.3 Job execution layer

In order to hide diversity of computing resources involved into data processing, a special software product is used: PanDA (Production and Distributed Analysis) system, which is responsible for running jobs at the available computing sites. PanDA, via its components, such as Auto Pilot Factory and Pilot, provides jobs delivery to the sites, their execution, data stage-in, and stage-out and monitoring of job execution at any step. ProdSys only has to create a job definition and send a created job to PanDA and then check its status in order to re-send the job in case of failure or continue its movement through the statuses.

Once the first job of the task comes to the computing site, starts, and sends “running” message to the PanDA server, system changes status of the task to “running”.

2.4 Workflow management

There are several job statuses in ProdSys, at the same time, there is another list of job statuses in PanDA. But only statuses that are needed for further decision making are being taken into account by ProdSys: the corresponding type of operation is applied for each job status in the system.

When the job is finished, the process has failed and its error code is in the list of suspicious statuses, a “manual check is needed” status may be applied to the job. It is marked in monitoring and the production manager starts an investigation to understand, what is wrong with this particular job. If a job has status “failed”, it will simply be re-submitted to PanDA.

There are several processing types supported by ProdSys: processing, merging, cross-checking of merging, merging of histograms, merging of event dumps, cross-check of event dumps, status on Castor. Once either main or sub-status of the job changes, it triggers one of the modules to take the job into further processing. Thus, the job is being moved through the statuses.

Steps of the processing may vary in accordance with task type. There are the following task types defined in the system: test production, mass production, technical production, DDD (Data acquisition system Data Decoding) filtering. Initial jobs of test and mass production generate result mini Data Summary Tree (mDST), histogram, event dump file, and logs. Jobs of DDD filtering tasks do not generate mDST files and histograms.

When all jobs of a single run (usually up to 1100 files, depending on beam intensity) are finished, mDST files of this run are merged into resulting files with declared size. Such merging is done on the computing site and runs through PanDA. After all merging jobs are finished, cross-check starts. Cross-check process ensures that a number of events in result files before and after merging are the same. If these values are different, a run will be marked as suspicious for further investigation of the production manager. Merging of histograms and events dumps are done in the same manner through PanDA. A separate background process checks the results of event dumps merging.

When all jobs of the task are finished successfully, the production manager checks that the generated results are correct, and then changes the status of the task from “running” to “archive”. This change initiates the post-processing data management, described in the next section.

2.5 Data management

There are three steps of data management: pre-processing, data management during processing and post-processing. Data management before and after processing is done by a set of data management modules. Data management during data processing is done by PanDA.

COMPASS raw data are stored on Castor. Castor stores files on tapes, there is also a disk cache for files, recently accessed or waiting to be migrated to tapes. In order to get a file from Castor, it must be first moved from tape to disk. There are statuses “staging” and “staged” for jobs in ProdSys. They are filled by data management services, which prepare files on Castor. Services use native Castor commands, such as stager_get and stager_qry. A job is marked “staging” when a production manager changes the task status to “send” and the system starts data preparation. Before sending files to the computing sites, the system must request (“staging”) them from tapes to disks on Castor and ensure that files were migrated (“staged”). Only after that, a job, which uses a staged file can be sent for processing.

A component of PanDA, called Pilot, is responsible for job execution on the computing sites. It may also transfer files, needed for processing, to the site and, after the job is finished, from the site. COMPASS computing model with one single storage element allows delegating that to Pilot. Thus, Pilot performs all data transfers for jobs.

Post-processing services are responsible for data archiving and clean-up. They copy job results from EOS to Castor and remove PanDA Pilot logs from EOS. Also, they add logs of jobs into large archive files and copy them to Castor. Data delivery from EOS to Castor is done by CERN FTS service, services of ProdSys send transfer requests to FTS.
Once the post-processing is finished, task status changes to “archived”.

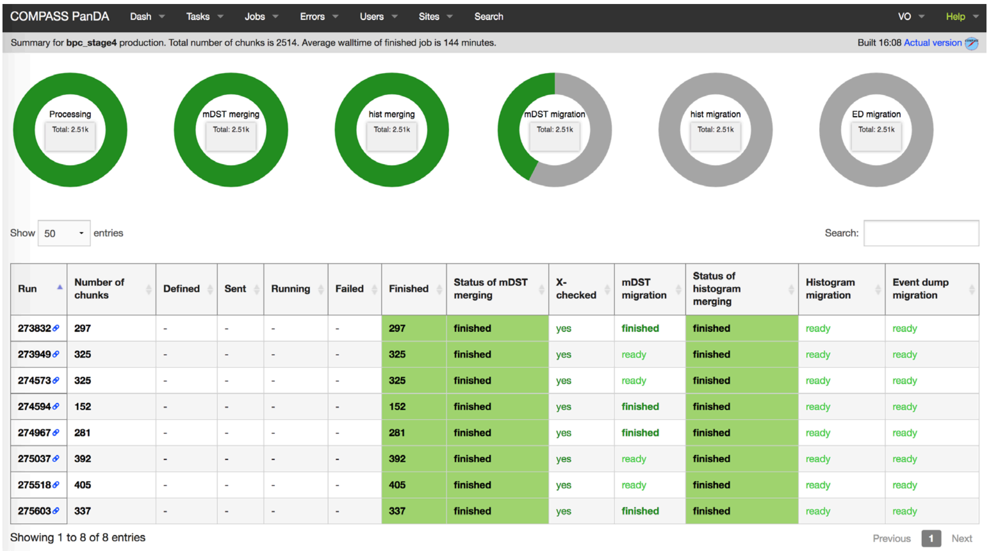
2.6 Monitoring

As any automatic system, ProdSys provides as much monitoring as possible.

Each system component sends heartbeats and reports:

- ProdSys management interface, where production manager can view all running tasks at one page;
- Each background workflow and data flow management process of ProdSys provides log, which can be accessed via web interface by the production manager;
- PanDA monitoring [10] presents experiment-related monitoring of task processing (Figure 3). This component was adapted for COMPASS data processing in order to present experiment-oriented features, such as number of events, runs info, merging of mDST, histograms and event dumps, archiving process, etc.;
- PanDA Auto Pilot Factory monitoring page presents sent, running and finished Pilots and their logs.

CERN IT and JINR Cloud service provide monitoring pages for HTCondor and FTS at CERN and virtual machines infrastructure at JINR.



Summary for **bpc_stage4** production. Total number of chunks is 2514. Average walltime of finished job is 144 minutes.

Built 16:08 Actual version

Run	Number of chunks	Defined	Sent	Running	Failed	Finished	Status of mDST merging	X-checked	mDST migration	Status of histogram merging	Histogram migration	Event dump migration
273832	297	-	-	-	-	297	finished	yes	finished	finished	ready	ready
273949	325	-	-	-	-	325	finished	yes	ready	finished	ready	ready
274573	325	-	-	-	-	325	finished	yes	ready	finished	ready	ready
274594	152	-	-	-	-	152	finished	yes	finished	finished	ready	ready
274967	281	-	-	-	-	281	finished	yes	finished	finished	ready	ready
275037	392	-	-	-	-	392	finished	yes	ready	finished	ready	ready
275518	405	-	-	-	-	405	finished	yes	ready	finished	ready	ready
275603	337	-	-	-	-	337	finished	yes	finished	finished	ready	ready

Showing 1 to 8 of 8 entries

Fig. 3. PanDA monitoring.

3 Processing on Blue Waters HPC

In 2016 COMPASS received an allocation on Blue Waters HPC [11], located in Urbana Champaign, University of Illinois. In late 2017, the project of adaptation of COMPASS ProdSys to work with Blue Waters has started.

In case of Blue Waters raw data is being delivered manually via Globus Online endpoint by the project manager, thus, steps which cover stage-in and stage-out on Castor may be turned off. And, in order to enable processing on Blue Waters, task definition and job execution components of the production system had to be changed. A full list of changes in the production system is presented below:

- Tasks definition: site selection and raw data location on Blue Waters were added to the user interface. Since automatic data delivery to Blue Waters is not yet available, a manual task assignment used in ProdSys;
- Data management components: stage-in and stage-out are turned off for Blue Waters tasks. This step is turned off, together with the stage-out step which moves data from EOS to Castor. Both these steps are done at the moment by the production manager. All other steps of automation for task definition and jobs generation works in the same manner as for the regular tasks;
- Jobs execution on site: PanDA Multi-Job Pilot [12] was used with several extensions.

Jobs execution on HPC machines makes much higher demands on the infrastructure because a number of running jobs can reach 100 000 or even 150 000, while on the existing Grid infrastructure number of simultaneously running jobs rarely reaches 20 000. Processing on Blue Waters was the main reason which triggered the upgrade of ProdSys infrastructure.

4 Infrastructure

COMPASS Production System has started from two virtual machines: PanDA server with PanDA, Auto Pilot Factory and Monitoring on one of them, and ProdSys management on the other. Later, PanDA monitoring machine was added. The information system was integrated with ProdSys management interface.

Data processing during July and August of 2018 has shown that setup with one single PanDA server and MultiJob Pilot on Blue Waters and standard Pilot on CERN HTCondor may handle 50 000 of simultaneously running jobs under the management of ProdSys both on Grid sites and HPC. Such load was achieved on a setup with PanDA database, server and Auto Pilot Factory, running together at the same physical machine, deployed at JINR Cloud Service [13]. However, to reach the target of up to 150 000 jobs on HPC and 20 000 on Grid sites reliably, computing infrastructure had to be changed: PanDA database, Server and Auto Pilot Factory, have to run on separate nodes. Such setup was prepared in late summer 2018. Migration to the new setup was performed in September 2018. At the moment, each key service of the system runs on a separate node: ProdSys management, PanDA server, Auto Pilot Factory, PanDA monitoring. CRIC information system [14], deployed at CERN, has become a member of COMPASS ProdSys infrastructure recently. A previous simple information system, integrated with ProdSys interface, is replaced by CRIC.

5 Summary

During the last decade, software services developed to process data of experiments on Large Hadron Collider have turned to software products. Projects such as PanDA, Rucio, AGIS, which were initially developed in the interest of one collaboration, are now used in various areas. Any project may use any combination of these products and the main effort lies in Grid environment preparation and logic of data processing definition, which are unique for each project or experiment.

COMPASS ProdSys is running in production mode since summer 2017. During this period more than 3 000 000 raw files containing 80 000 000 events were processed, 35 000 000 files were generated, 500TB of merged data were migrated to Castor.

COMPASS computing model with one storage element and disk-less computing sites allows running jobs without any complicated distributed data management tools, using only built-in data movers of PanDA Pilot and a set of background services of ProdSys. However, since Blue Waters has become a member of the computing infrastructure of the experiment, the role of data management is increasing. Manual data transfers between sites are the point of possible data inconsistency. Thus, adding distributed data management component is under consideration. In addition, processing on HPC has triggered a redesign of the infrastructure. Migration to the new component of PanDA: Harvester [15] which was designed to work both on Grid and HPC resources, is under consideration as well.

References

1. P. Abbon et al, *The COMPASS experiment at CERN*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **577**, 455-518 (2007)
2. Castor, CERN Advanced STORage manager, http://castor.web.cern.ch/
3. EOS, https://eos.web.cern.ch/
4. HTCondor, https://research.cs.wisc.edu/htcondor/
5. CERN AFS, Andrew File System, http://information-technology.web.cern.ch/services/afs-service
6. CVMFS, CernVM File System, https://cernvm.cern.ch/portal/filesystem/
7. A. Petrosyan, *COMPASS Grid Production System*, CEUR Workshop Proceedings **2023**, 234-238 (2017)

8. A. Klimentov et al., *Next generation workload management system for big data on heterogeneous distributed computing,* Journal of Physics Conference Series **608** 012040 (2015)
9. Django framework, https://www.djangoproject.com/
10. S. Padolski, T. Korchuganova, T. Wenaus, M. Grigorieva, A. Alexeev, M. Titov, A. Klimentov, *Data visualization and representation in ATLAS BigPanDA monitoring*, Scientific Visualization, **10**, 69-76 (2018)
11. A. Petrosyan, *COMPASS Production System: processing on HPC*, CEUR Workshop Proceedings **2267**, 139-144 (2018)
12. K. De, A. Klimentov, D. Oleynik, S. Panitkin, A. Petrosyan, J. Schovancova, A. Vaniachine, T. Wenaus on behalf of the ATLAS Collaboration, *Integration of PanDA workload management system with Titan supercomputer at OLCF*, Journal of Physics Conference Series **664** 092020 (2015)
13. A.V. Baranov, N.A. Balashov, N.A. Kutovskiy, R.N. Semenov, *JINR cloud infrastructure evolution*, Physics of Particles and Nuclei Letters **13**, 672-675 (2016)
14. A. Anisenkov, *Computing Resource Information Catalog: the ATLAS Grid Information system evolution for other communities*, CEUR Workshop Proceedings **2023**, 1-5 (2017)
15. A. Anisenkov, D. Drizhuk, W. Guan, M. Lassnig, P. Nilsson, D. Oleynik on behalf of the ATLAS Collaboration, *Global heterogeneous resource harvesting: the next-generation PanDA Pilot for ATLAS*, Journal of Physics Conference Series **1085** 032031 (2018)