COMPASS GRID PRODUCTION SYSTEM

A.Sh. Petrosyan

Joint Institute for Nuclear Research, 6 Joliot-Curie, Dubna, Moscow region, 141980, Russia

E-mail: artem.petrosyan@jinr.ru

LHC Computing Grid was a pioneer integration effort, managed to unite computing and storage resources all over the world, thus making them available to experiments on the Large Hadron Collider. During decade of LHC computing, Grid software has learned to effectively utilise different types of computing resources, such as classic computing clusters, clouds and hyper power computers. While the resources experiments use are the same, data flow differs from experiment to experiment. A crucial part of each experiment computing is a production system, which describes logic and controls data processing of the experiment. COMPASS always relied on CERN facilities, and, when CERN, during hardware and software upgrade, started migration to resources, available only via Grid, faced the problem of insufficiency of resources to process data on. To make COMPASS data processing able to work via Grid, the development of the new production system has started. Key features of modern production system for COMPASS are: distributed data processing, support of different type of computing resources, support of arbitrary amount of computing sites. Build blocks for the production system are taken from achievements of LHC experiments, but logic of data processing is COMPASS-specific.

Keywords: COMPASS, PanDA, workload management system, Grid, Condor, distributed data management, production system

1. Introduction

All physics experiments have same steps of data taking, processing, archiving, but details of these steps are absolutely different for each experiment. Implementation of data processing of the experiment comes from its computing model, which, in its turn, is described by physics processes, collected data type, volumes, chosen software technologies, data transformations, available computing resources and their types, type of storage, etc.

Data of COMPASS experiment [1], after being taken, are delivered to Castor storage for further processing. Metadata, which describes conditions, setups, year, period, run chunk number and various other parameters of data are stored in MySQL database (previously, Oracle used to be a primary storage for metadata). Starting from this point, offline data processing begins.

Being resident or CERN, COMPASS depends on CERN IT services to store and process data. During experiment's lifecycle (data taking of the experiment has begun in 2002), some of IT services became obsolete and during next 2-3 years the following services will be replaced by more modern ones: Castor by EOS (link), lxbatch LSF by Condor, AFS by EOS. This process of simultaneous replacement of computing infrastructure components strongly influences data processing of the experiment and triggered changes of software components which interact with computing site, data, conditions and metadata storage. In order to ease consequences of current and future infrastructure changes, the computing model of the experiment was adapted accordingly [2][3]. Support of several computing sites and distributed jobs submission was performed via adding Auto Pilot Factory and PanDA workload management system [4][5]. Availability of experiment software releases on any remote computing site was achieved by installing them on CVMFS. 3 computing sites were defined: CERN Condor, JINR Tier-2, Trieste Tier-2. Thus, Grid infrastructure of experiment was created. Running jobs via these additional layers allowed production system administrator to add or remove computing sites online by simply changing configuration without any changes in computation process. If the site goes to downtime, jobs simply are not receiving it and workflow concentrates on other sites of the infrastructure. Usage of PanDA us to treat various site resource managers as one, like if they were one large computing queue.

To manage tasks and jobs in such distributed infrastructure, special software must be created, because data processing of any experiment is unique. Usual name of such software is a production system. It covers all steps of data processing: from task definition and datasets selection, jobs submission and monitoring of their statuses, to decision making mechanisms, which control data processing through all the steps. In case of COMPASS such software was already presented, however, during the migration to Grid environment and distributed computing, it became clear that it has to be replaced by a brand new one. The reasons were the following:

- previous system was too strongly integrated with existing computing infrastructure and it was prepare to work in "local" environment of production manager's account;
- it was not designed to work with any other type of computing resource except LSF;
- it used commands, available only in interactive mode to submit and control jobs.

Adding one more computing site in the previously used production system was impossible.

Thus, the development of the new production system, which has to cover all the needs of data processing in the distributed heterogeneous computing environment and will be able to overcome the limitations of the previous implementation described above, has begun.

2. Production system overview

The production system of the experiment must meet the following expectations:

- must support data processing from task definition till data archiving;
- must provide support of all types of data processing: Monte-Carlo simulation, reconstruction, user analysis;
- must require minimum software development and include as much as possible components of already developed systems;
- must support any type of computing and storage resources;
- must provide user-friendly and fully functional interface;
- must be secure, flexible, easy to extend and deploy;

- must provide monitoring of each steps of data processing and user actions.

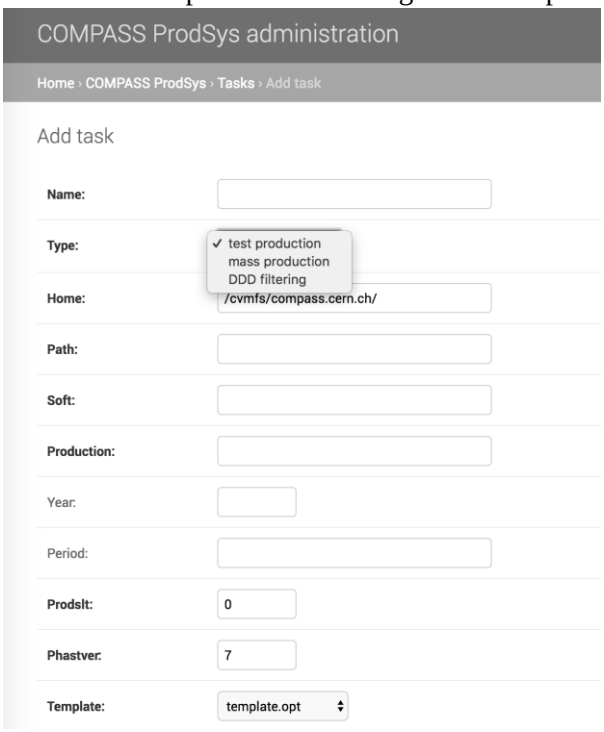

Figure 1. Task definition interface

Production system has the following list of components:

- tasks and jobs definition and management interface;
- status tracking and decision making mechanism;
- data management mechanism;
- jobs submission mechanism;
- jobs delivery to remote sites;
- jobs execution on remote sites;
- monitoring.

Since logic of processing of each experiment is unique, tasks and jobs definition, management interface and decision tracking mechanism had to be developed from scratch. All other components of the infrastructure initially were developed to cover needs of ATLAS distributed computing and were adapted for COMPASS: PanDA workload management system manages jobs delivery, execution on remote sites and monitoring via its components: PanDA server, AutoPyFactory, Pilot and Monitoring. Django framework was chosen to build web user interface. RDBMS is MySQL. Programming language is Python, it is the same for all components of the system, which makes it easy to integrate. Decision making mechanism organised as set of micro-services, each runs periodically and performs its small operation basing on conditions in system's core database. Data management mechanism prepares, delivers and archives files.

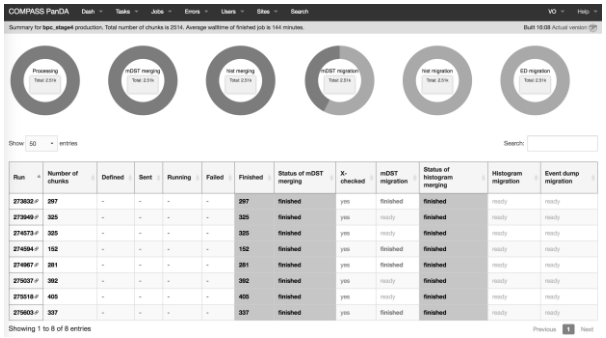

Figure 2. Production summary

While users analysis implies set of jobs, mass production is much more complicated process and requires data management and decision making mechanisms. It includes the following steps:

- task definition as a set of runs or a list of file with common list of parameters;
- jobs generation for task;
- warm-up files on Castor so that they move from long-term storage to short-term storage;
- jobs submission, where one initial file is one job and each job produces results of three types: mDST, histogram, and event dump;
- merging of results of jobs of each run to achieve a set of 4Gb files for most optimal storage on Castor;
- cross check to ensure that number of events in resulting jobs is the same as number of events in merged file;
- merging of histograms for each run;
- merging of event dump files;
- archiving of all result files to Castor.

Each process is controlled by a separate micro-service. For each step status check status mechanisms are implemented, for steps which require submission to remote resources retry mechanisms are implemented.

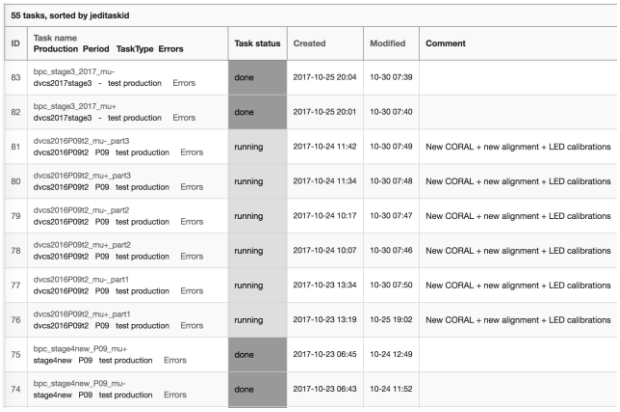

Figure 3. Tasks summary

Once components of the system work as independently, time-out, retry, load balancing mechanisms are implemented in order to achieve maximum use of available resources. Jobs submission performed automatically, amount of jobs to be submitted at each submission cycle is calculated basing on amount of running jobs, results of previous submissions, etc. Jobs submission and status checking in the system are divided in order to reduce load on services and machines where services are running.

Screenshots of production system user interface and services and jobs monitoring pages are shown on the figures 1-3.

3. Future plans

Future plans include enabling new computing resources, and, also, new types of computing resources, such as HPC facilities, in particular Blue Waters HPC of University of Illinois at Urbana-Champaign, where collaboration has active allocation.

Monte-Carlo processing will be covered by the production system, at the moment MC production is done by Analysis coordinators and users groups in their home institutes as users analysis, separately and not organised into one managed effort.

Users analysis will also be moved under the production management system.

Another direction of development is integrating Rucio [6] distributed data management system to build a new central data catalog, which will cover also data delivery to and from any remote site, involved into processing. This will help to organise data and to organise data accounting, transfers, archiving, etc.

4. Conclusion

Serious integration and software development efforts have been performed during 2016-2017 in order to organise a distributed processing of data gathered by experiment. New system has been working in a production mode since August 2017. During this period, via new Grid Production System almost 2 millions chunks, collected by COMPASS during 2015-2017, were processed and 0.5PB of resulting data was generated. Processing rate handled by the system is ~12000 of simultaneously running jobs.

All steps of processing are covered with rich monitoring services, allowing to get full picture of data processing at any particular period of time.

COMPASS collaboration received a brand new production system, based on widely used and actively supported software components and secured itself against decommissioning of LSF and phase-out of AFS and Castor.

All management components of the system are deployed on the JINR cloud service [7].

References

[1] Abbon P. et al. The COMPASS experiment at CERN // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. — 2007. — Vol. 577, Issue 3. — P. 455-518.

[2] Petrosyan A.Sh. PanDA for COMPASS at JINR // Physics of Particles and Nuclei Letters. — 2016. — Vol. 13, Issue 5. — P. 708-710.

[3] Petrosyan A.Sh., Zemlyanichkina E.V. PanDA for COMPASS: processing data via Grid // CEUR Workshop Proceedings, Vol. 1787. — P. 385-388.

[4] Maeno T. et al. Evolution of the ATLAS PanDA workload management system for exascale computational science // Journal of Physics Conference Series. — 2014. — Vol. 513. — http://inspirehep.net/record/1302031/

[5] Klimentov A. et al. Next generation workload management system for big data on heterogeneous distributed computing // Journal of Physics Conference Series. — 2015. — Vol. 608. — http://inspirehep.net/record/1372988/

[6] Rucio homepage, http://rucio.cern.ch/
[7] Baranov A.V., Balashov N.A., Kutovskiy N.A., Semenov R.N. JINR cloud infrastructure evolution // Physics of Particles and Nuclei Letters. — 2016. — Vol. 13, Issue 5. — P. 672-675.