Simulation and reconstruction of cosmic ray showers for the Pierre Auger Observatory on the EGEE grid

J. Chudoba$^{1,2}$, P. Nečesal$^1$, M. Nyklíček$^1$, J. Schovancová$^{1,2}$, P. Trávníček$^1$
for the Pierre Auger Collaboration

$^1$ Institute of Physics, ASCR, Na Slovance 2, Prague 8, CZ-18221, CZ
$^2$ CESNET, a.l.e., Zikova 4, Prague 6, CZ-16000, CZ
E-mail: schovan@fzu.cz

Abstract. We report on our experience with the Large scale Monte Carlo production for the Pierre Auger Observatory in past 3 years. We describe the Grid production framework of the Virtual Organisation auger, in particular the Grid production monitoring tools, custom database of simulated events and user’s access to the simulated data.

1. Introduction

1.1. Pierre Auger Cosmic Ray Observatory
The Pierre Auger Cosmic Ray Observatory [1] is studying the Universe’s highest energy particles ($E > 10^{18}$ eV) which impact into the Earth atmosphere and create air showers. The Auger Observatory is a hybrid detector using two independent detection methods. The first detecting method uses 1600 water tanks that cover enormous section of the Pampa Amarilla (yellow prairie) in western Argentina and serve as a particle detector measuring Cherenkov light. The second detecting method tracks the development of air showers by observing ultraviolet light emitted in the Earth’s atmosphere.

We describe the grid production framework of the Pierre Auger Observatory in this paper. Nature of performed simulations is described in Section 2. In Section 3 we give a brief overview of the production framework. More detailed description of the production monitoring tools is given in Section 4. In Section 5 we describe the Custom Database of simulated showers and data access for users.

2. Simulations of cosmic ray interactions
Interactions of cosmic rays particles with the nuclei of air gases at energies many orders of magnitude above the current accelerator capabilities induce unprecedented extensive air showers in the atmosphere. Different interaction models are used to describe the first interactions in such showers and their predictions are confronted with measured shower characteristics.
2.1. Auger Computing Model
The Auger Computing Model assumes that Simulation Committee collects simulation requests from the Physics Groups in the Collaboration, sets the priorities to the simulation tasks and communicates with the Production Manager what the definitions of the simulation chain will be. The usual simulation chain consists of CORSIKA \cite{2} simulations of showers, OFFLINE \cite{3} reconstruction of those showers and simulations of the detector response to those simulated showers, and the delivery of simulated data to the Users (Physics Group which has requested particular kind of simulations). The simulation part of the chain benefits from the Grid computing resources.

Libraries of cosmic ray showers with more than 47,000 simulated events using CORSIKA with EPOS \cite{4,5} or QGSjetII \cite{6,7} models were created. These showers are reused several times for simulation of detector response at different position within the detector array.

The CORSIKA simulations can run on every Auger grid site, while the simulation of the detector response and shower reconstruction run on 4 sites, where the corresponding SW (OFFLINE) is installed.

2.2. Computing resources
Virtual Organisation AUGER was created in 2006 by the Czech Auger group in cooperation with CESNET. The VO AUGER has almost 50 members. CESNET provides and maintains central resources such as LCG RB, gLite WMS, LB, UI, LFC, registration portal and the VOMS server. VO AUGER benefits from both dedicated and opportunistic computing resources available at the Grid provided by 16 different sites in 10 countries world-wide. AUGER sites offer ca 500 dedicated CPU cores and 100 TB of disk space at storage elements. At some sites we share available computing resources with local users. AUGER Grid resources are used mainly for the official CORSIKA and OFFLINE production.

3. Production framework
A Production framework for submission of many simulation jobs with different input parameters has been developed. The framework consists of three modules: the Job management, the Monitoring and the Custom Database modules. The Job management module utilises the gLite middleware \cite{8} for the job submission, while the Monitoring and Custom database modules benefit from the Django web framework \cite{9}.

The Production framework was designed with needs of the Virtual Organisation AUGER users in mind, but it can be used by a generic VO or grid user willing to simulate CORSIKA showers on the Grid, since it can be easily adapted.

There are three user roles to which the Production framework aims: the Users, i.e. the physicists, the Shifters, i.e. people who are submitting the jobs, and the Production Manager, who is responsible for the production flow.

3.1. Job management module
The Job Management Module is a set of BASH scripts facilitate easy job submission, OutputSandbox retrieval and job resubmission from a gLite middleware User Interface. It can be used without superuser priviledges to the User Interface machine.

The production framework has been used with two applications: the CORSIKA and the OFFLINE. CORSIKA is delivered to a Worker Node via a tarball stored on several Storage Elements, thus accessible from every Computing Element supporting VO AUGER. The OFFLINE requires interactive installation, thus the most recent version is installed by local administrators of the corresponding Computing Element. The submission scripts are a bit different for each application, since the nature of simulation is different.
4. Production monitoring
The Monitoring module serves as a visualisation tool for the production monitoring. It helps us to distinguish between jobs which produced simulated data for further processing and physics analysis and of those whose output is useless for physicists.

The entire lack of production-ready monitoring tools with user-defined parameters in the gLite middleware results in development of many different monitoring tools, each of those meets only its developer's VO requirements.

In 2007 and 2008 we have tested deployment of the Job Provenance [10], a generic gLite service designed for long-term archiving of information on executed jobs focusing on scalability, extensibility, uniform data view, and configurability, that allows more specialized catalogues to be easily built, in the Auger grid production [11]. When the Job Provenance becomes gLite-certified and production-ready it will be a powerful tool for the production monitoring and possibly for development of Custom Databases based on user-predefined tags.

In the meantime, we felt an intensive need for a monitoring tool which helps us distinguish successful jobs with useful data for physics analysis and those jobs which failed to produce data worth analysis, to recognize the failure cause so we could react to it, and to deliver useful data to the Users.

Production monitoring with the Monitoring Module is based both on gLite middleware logging and bookkeeping information and user-predefined information acquired from the job logs. Visualisation of the production current status is achieved with a web application. An update of the visualised information can be performed either via a command line tools or via a web application.

4.1. Job log analysis
Usefulness of a job outputs is determined by its terminal state published by the gLite middleware and user-predefined tags in the job logs. The possible terminal states exposed by the gLite middleware are “Done (Success)”, “Done (Failed)” and “Aborted”. The job in “Aborted” state has no OutputSandbox, thus the only information about the failed simulation we have are the inputs we provided, and the logging and bookkeeping logs. On the other hand, jobs in one of the “Done” terminal states do have OutputSandbox, therefore we can analyse their stdout and stderr logs in addition to the available logging and bookkeeping logs.

We have a tool to analyse these stdout, stderr and logging and bookkeeping logs which produces a XML file with information about jobs, and a tarball with analysed job logs. The Production Manager then triggers (either manually or in a cron job) update of the monitoring database via the Dashboard API.

4.2. Production Dashboard
The Production Dashboard is a web application written in Python and based on the Django web framework [9]. We use the MySQL database engine. There is an authorisation and permission tools implemented in the dashboard application, thus all the dashboard views are password-protected and user-account based. The Production Dashboard for the Pierre Auger Observatory Grid production resides at a webservice hosted at the Institute of Physics. The Production Dashboard database is daily backed up. A new instance of the Production Dashboard can be set up in matter of tens of minutes at any other institute of the Pierre Auger Observatory Collaboration.

Not only the Production Dashboard application represents a web frontend to the production monitoring, but it also serves as a web frontend for the Custom database and user data access.

The Production Dashboard Homepage, shown in Fig. 1, represents a guidepost to all the features it offers. Every Dashboard page consists of five areas: the header and footer, left and right column, and the central content column. The left column with links is devoted to the
Figure 1. Homepage of the Production Dashboard webserver.

Users, while the the links in the header area are dedicated to the Shifters or the Production Manager. The right column usually shows explanatory notes to the objects shown in the central content column.

The Users should be interested in two dashboard features: the Shower browser (described in Section 5). The Shifters’ or Production Manager’s interest should be attracted to the “Shifter zone” where a list of all the dashboard features for the production monitoring resides. The Production Manager acquires the summary information about the CORSIKA and OFFLINE production from summary tables. The summary contains a table with number of successfully finished shower simulations, number of jobs which hasn’t been submitted yet or which has to be resubmitted, and the total number of jobs in every Library. The Dashboard also provides a tool which helps with production overview plots.

4.2.1. Recent jobs One of the Dashboard features is a list of the Recent jobs. This view queries the job database and prints total number of jobs whose simulation part began in past $N$ hours (where $N \in \{24, 48, 168, 336\}$) and such a job has already finished and the Production Manager has already updated the Production Dashboard database. The Recent jobs view offers either a list of jobs which has not passed the output validation process (i.e. their outputs cannot be used for further User’s analysis), or the list of all jobs submitted in the $N$ hours timespan.

The Shifter selects the timespan and validity (only failed jobs or all the jobs) view and a list of jobs fulfilling these criteria appears (Fig. 2). Every item on the list offers basic information about the job and a clickable link to the shower simulation details (Fig. 3) with job logs available. The Shifter can further study the logs and chase the production problems.

4.2.2. Dashboard view The Dashboard shows a color-coded matrix of showers of one Library. Showers are organised in [energy; zenith angle] bins in Libraries and ordered by their ID string. A job represents one attempt of simulation/reconstruction of an event (shower). Each job in the Dashboard is in a defined state and is shown with color-coded background corresponding to its state in the Dashboard.

Only a job which has not had problems with the Grid infrastructure (job exits with status 0, no problems with upload of simulated data, no middleware problems) and has not had problems...
Figure 2. The Recent jobs view offers count of jobs submitted and finished in the past 24, 48, 168 and 336 hours, both for CORSIKA and OFFLINE jobs. Shifter can navigate to pages with list of failed jobs and list of all jobs finished within the corresponding timeframe. Such an example of failed CORSIKA jobs list in a past 48 hours timeframe is shown. Failure legend is shown in the right column. Dots in the “More info” column (at the right side of the job list) represent a hyperlink to a page with corresponding job details, see Fig. 3.

The simulation SW can pass validation, thus its outputs can be analysed by the Users. Such validated jobs are emphasized with the blue background color. Jobs with other than blue background suffered from some problems, thus their outputs cannot be passed further to the User’s analysis.

The job status determination is based on information digged from the job stdout and stderr logs. The list of known failures evolves in time as we enlarge our statistics of simulated jobs and improve our understanding of the simulation pains.

The Shifter or the Production Manager selects Library (or OFFLINE version and Library) from the list, confirms the selection, and a table with color-coded jobs appears.

The CORSIKA Dashboard (Fig. 4) shows a list of showers in every [energy; zenith angle] bin of the selected Library. Every shower is represented with a rectangle with colored background. There is an ID string (prefix “DAT” and 6-digit order number of a shower in the overall production) and number of simulation jobs of this shower which has been submitted so far in every rectangle. The rectangles are clickable. If such a shower has already been simulated, click on particular rectangle redirects the browser view to a page with simulation jobs details (Fig. 3) and a link to the shower’s input. If a particular shower has not been simulated yet, click on corresponding rectangle redirects the browser view directly to a page with the particular shower simulation input.
4.2.3. Job resubmission  The Job resubmission feature is one of the Production Manager’s tools. The Production Manager selects a Library (or Library and OFFLINE version), confirms selection and the Frontend returns a text file (line break denoted by ‘\n’) with a list of inputs which should be (re)submitted. The corresponding selection forms look very similar to those used for the Dashboard view of a Library (Fig. 4).

4.2.4. Production Dashboard Admin  One of the benefits of the Django-based application is the automatic Django Admin Interface. It is a powerful and production-ready interface that enables immediate editing of the production objects, e.g. various definitions of a new Library, addition of a new application or introduction of a new version of an application, Shifter personnel, site definition, etc.

Access to the Admin Interface is user-account based. It enables three user modes: the superuser mode with access to all the Objects, common user mode with access to the Administration Interface and access rights to the Objects given by the Access Control List (ACL) settings, and restricted user mode without any access to the Administration Interface, but with access to password-protected areas of the website. Any user account can be added or disabled in matter of seconds.

The ACLs to an Object can be given either to the whole group of users or can be user-specific, both at the same time. Available set of ACLs to any Object is “Add”, “Change”, “View” and “Delete”.

Django Admin assigns a particular type of an HTML object (edit field, text box, selection list, etc.) to any field of an Object, type of the HTML object assigned depends on the design of such an Object. Appearance (presence of a particular HTML field, filter choice) and layout (order of HTML objects in the page) of an Administration page of any of the Objects depends on the developer’s choice.
5. Custom database and user data access

The Pierre Auger Collaboration associates about 400 scientists world-wide. However, only about 50 of the Collaborators are members of the VO AUGER. Nevertheless, every analysis task has delegated at least one Collaborator to become a member of the VO AUGER. All the CORSIKA simulated showers data is stored on the Storage Elements so they can be processed with the OFFLINE software on the Grid.

We have created a Custom Database of simulated and reconstructed events so the simulated events reach physicists reasonably soon. The Custom Database offers a password protected webpage (Fig. 5) for every griddified application as an user interface. Every such webpage enables Users to select set of events with properties of his/her interest. The selection criteria are type of the primary particle, its energy, and zenith angle at which it has arrived to the Earth’s surface,
hadronic model used for the simulation, and the version of the simulation software.

The User sets the selection criteria and queries the custom database and gets a list of the LFNs of the available events. Since all the events are on the Grid storage, we offer a tool which downloads these events to the User’s UI.

Figure 5. CORSIKA showers selection form. At this page user obtains a list of logical file names in the catalogue.

6. Conclusions
We have developed a production framework for the large scale Monte Carlo production performed on the Grid. The production framework consists of three main modules, the Job management, the Monitoring, and the Custom Database modules. We have created a library with more than 47,000 cosmic ray atmospheric interaction events and provided tools so the physicists reach these simulated events in a reasonably short time. The production framework can be modified and used by any VO.

Acknowledgements
Support from the INGO LA08016 and LC527 grants of the Czech Ministry of Education, and from the research grant AV0Z10100502 of the Academy of Sciences of the Czech Republic is gratefully acknowledged by the authors.

References
[1] Pierre Auger Cosmic Ray Observatory webpage http://www.auger.org/
[2] Heck D, Schatz G, Thouw T, Knapp J and Capdevielle J N FZKA-6019
[3] Argiro S et al. The Offline Software Framework of the Pierre Auger Observatory 2007 Nucl. Instrum. Meth. A580 1485-1496 (Preprint arXiv:astro-ph/0611311)
[4] Pierog T and Werner K 2008 Phys. Rev. Lett 101 907–914 (Preprint arXiv:astro-ph/0611311)
[5] Werner K, Liu F M and Pierog T 2006 Physical Review C 74 044902 (Preprint arXiv:hep-ph/0506232)
[6] Ostapchenko S 2008 Physical Review D (Particles and Fields) 77 034009 (pages 17)
[7] Ostapchenko S 2006 Physical Review D (Particles and Fields) 74 014026 (pages 17)
[8] gLite middleware documentation http://glite.web.cern.ch/glite/
[9] Django web framework http://www.djangoproject.com/
[10] Matyska L et al. Job tracking on a grid – the Logging and Bookkeeping and Job Provenance services 2007 Tech. rep. CESNET http://www.cesnet.cz/doc/techzpravy/
[11] Schovancová J et al. Experience with Large Scale Simulations on the EGEE Grid for the AUGER collaboration CIEMAT 2007 ed Gavela R, Marechal B, Barbera R et al.