Mass production of extensive air showers for the Pierre Auger Collaboration using Grid Technology

Julio Lozano Bahilo[1] for the Pierre Auger Collaboration[2]

[1] Universidad de Granada, Granada, Spain; [2] Pierre Auger Observatory, Malargüe, Mendoza, Argentina (Full author list: http://www.auger.org/archive/authors_2011_05.html)

e-mail address: julio@ugr.es

Abstract. When ultra-high energy cosmic rays enter the atmosphere they interact producing extensive air showers (EAS) which are the objects studied by the Pierre Auger Observatory. The number of particles involved in an EAS at these energies is of the order of billions and the generation of a single simulated EAS requires many hours of computing time with current processors. In addition, the storage space consumed by the output of one simulated EAS is very high. Therefore we have to make use of Grid resources to be able to generate sufficient quantities of showers for our physics studies in reasonable time periods. We have developed a set of highly automated scripts written in common software scripting languages in order to deal with the high number of jobs which we have to submit regularly to the Grid. In spite of the low number of sites supporting our Virtual Organization (VO) we have reached the top spot on CPU consumption among non LHC (Large Hadron Collider) VOs within EGI (European Grid Infrastructure).

1. Generation of extensive air showers (EAS) for the Pierre Auger Observatory

1.1. The Pierre Auger Observatory

The Pierre Auger Observatory [1] is a hybrid detector whose goal is to study ultra-high energy cosmic rays (UHECR). It makes use of fluorescence telescopes to collect fluorescence light produced as the cascade of secondary particles composing the EAS travels through the earth's atmosphere. It also uses water tanks which gather Cerenkov light emitted by the particles which remain at ground level after the partial absorption of the EAS in the air. Both detectors digitize light using photomultiplier tubes (PMT) and associated electronics.

The simulation of the events collected by the Pierre Auger detectors is done in two steps: first we generate the EAS with a given set of parameters and then we simulate the detector response and perform the reconstruction of the physics information of the shower. Those steps require different software packages with different requirements and thus it is more appropriate to write about both steps in separate sections. In this section we will describe briefly an EAS and discuss the software programs used in order to generate them.

1.2. EAS characteristics

The energies of the primary cosmic rays we are concerned with range roughly from 10^{16} to 10^{21} eV, thus being the most energetic particles studied by particle physics scientists. A single cosmic ray at those energies interacts soon after entering the earth's atmosphere producing an enormous cascade of secondary particles. It has a large electromagnetic component and a smaller hadronic component which have to be followed in order to characterize completely the development of the EAS. This means that the program to generate the shower must track billions of particles over tens of kilometers until the particles reach ground level. An example of such an EAS generated by a UHECR is depicted in figure 1 [2].

1.3. Simulation programs

There are two simulation packages widely used in our field in order to generate EAS: AIRES (AIRshower Extended Simulations) [3] and CORSIKA (Cosmic Ray SImulations for KAscade) [4] though we have only used CORSIKA in our Grid productions.

Both packages adopt high energy and low energy interaction models during the tracking of the secondary shower particles. Those models can be selected by the user and the executables are statically linked to the corresponding libraries when they are compiled. They are driven through input data files which contain parameters relevant to the set of EAS which we want to simulate: primary particle species, energy, zenith and azimuth ranges and other values changing the simulation conditions. We group these sets of events into *libraries*.

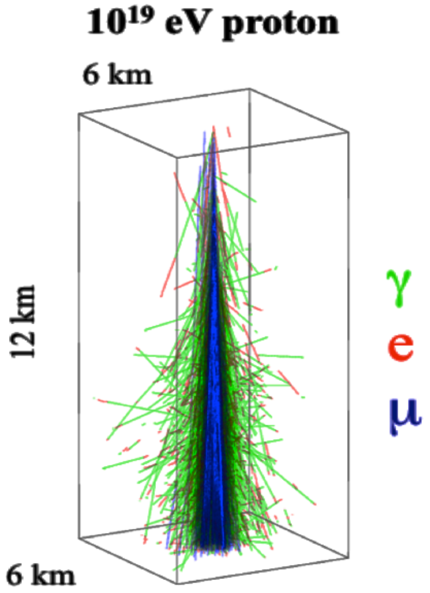

Figure 1. Image showing the development of a single EAS produced by a proton with an energy of 10^{19} eV. Most prominent particle types are shown in different colours [2].

The full simulation of a single EAS at the highest energies may take days, even months, of computing time and therefore we are forced to use statistical *thinning* techniques to avoid tracking individually each secondary particle while keeping an acceptable degree of precision. These techniques imply grouping a certain number of particles with similar characteristics when they fall below an energy value set by the user.

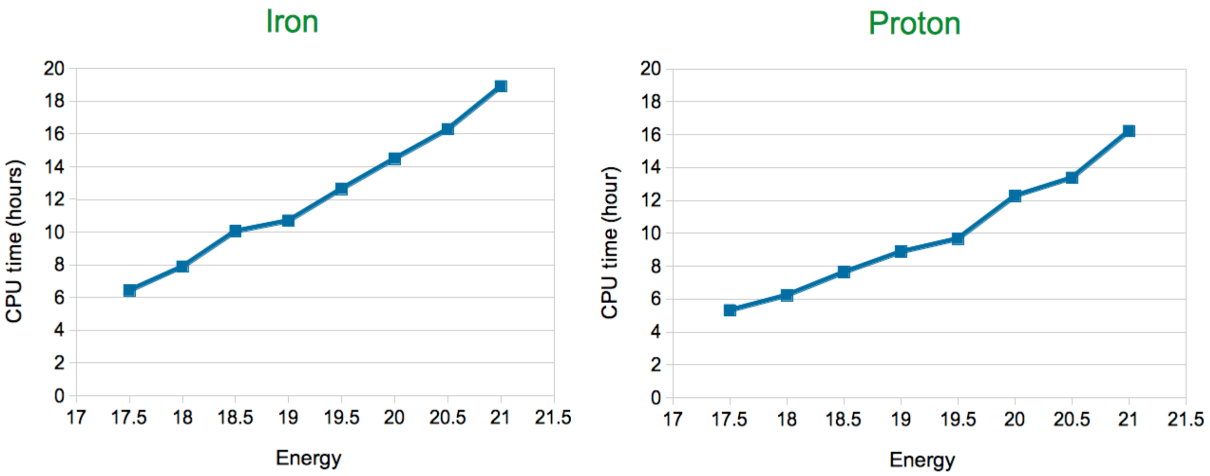

Figure 2. Left: CORSIKA average execution time as a function of the logarithm of the energy for iron primary cosmic rays. Right: same plot for proton cosmic rays

Usually, the applied *thinning* parameter depends directly on the energy of the cosmic ray. Even if the *thinning* is stronger for higher energies, the CPU consumption times, as can be seen in figure 2, increase almost linearly with the logarithm of the energy of the primary cosmic ray. There is also a dependence on the type of primary so that lighter nuclei primary consume lower amounts of time, and varying the models and parameters can make a difference. On average it takes several hours to complete a single simulation, although the spread is high.

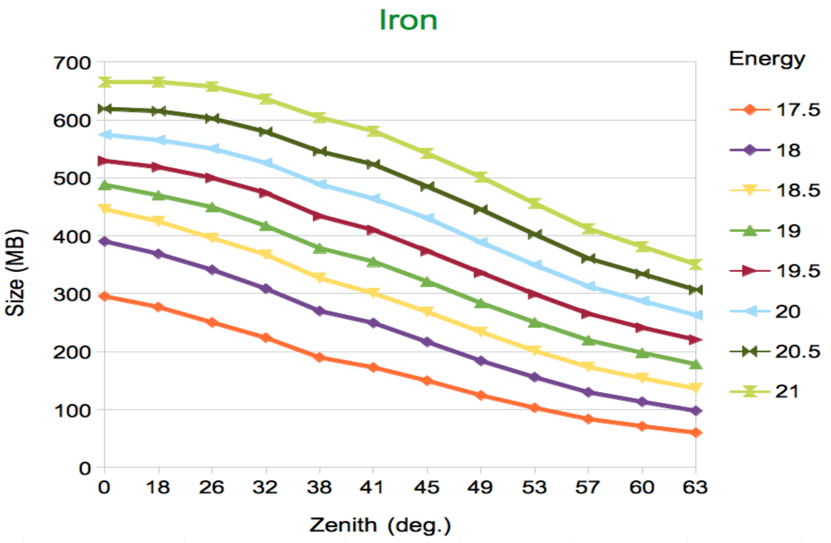

Figure 3. Size in MBs of the output files containing information on generated showers as a function of the zenith angle of the primary for iron primaries at diverse energies (expressed as log(E)).

The size of the output files change dramatically according to the primary energy, but they also have a strong dependence on the zenith angle as seen in figure 3. The values also depend on models and parameters so we present just the values obtained for our typical simulations which, on average, give sizes of tens to hundreds of MBs.

2. Detector simulation and reconstruction

The detector simulation package, referred to as Offline [5], is a software framework composed of independent modules which allows one to perform any step or set of steps within the simulation and reconstruction chain. XML files contain configurable parameters to modify the behavior of the modules. The tracking of the particles traversing the water tanks, which is done with the help of the GEANT4 [6] package, is the most time consuming process.

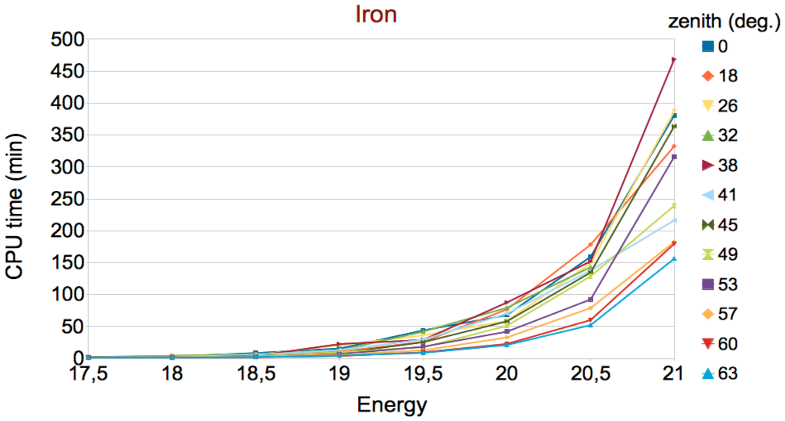

Figure 4. CPU time of an Offline execution as a function of the energy for various zenith angles.

The execution times of the Offline are smaller in general than those of the shower generation program. Whereas the generation of a shower requires several hours, the processing of one of those showers takes normally less than or about an hour except for the highest energies (see figure 4). The plot shows an approximately exponential rise.

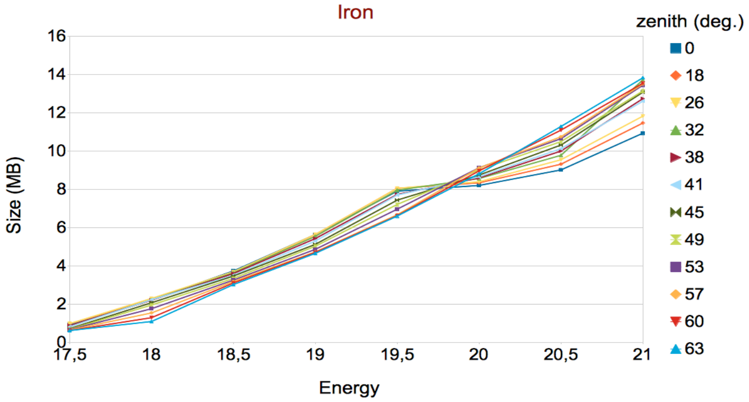

Figure 5. Total size of Offline output files as a function of the energy for various zenith angles.

The sizes of the resulting output files are much smaller (figure 5) since they just contain the information collected by the detectors, some processed information, and physical shower reconstructed parameters. They increase with the energy and do not depend very much on the zenith angle of the cosmic ray.

3. Grid computing model for official Auger simulations

The submission of jobs to Grid proceeds via a set of scripts written in Bash and Python which are running continuously at a User Interface (UI). They look for certain input files needed by our programs: CORSIKA needs input data files and the Offline requires a list of Logical File Names (LFN) associated to the input showers. The input files are represented by the squares within the light shaded areas in figure 7. In addition to the files mentioned previously, an execution script of the corresponding software package is also sent to the Grid within each individual job. This script is automatically generated from a common template. The Job Description Language (JDL) file required by the submission command is also created with the help of a template file. The presence of those files in certain directories is checked by the *Management scripts* (see also the same figure 6) and it triggers the submission of jobs to Grid Computing Elements (CE) as collections of jobs handled by the Workload Management System (WMS) services.

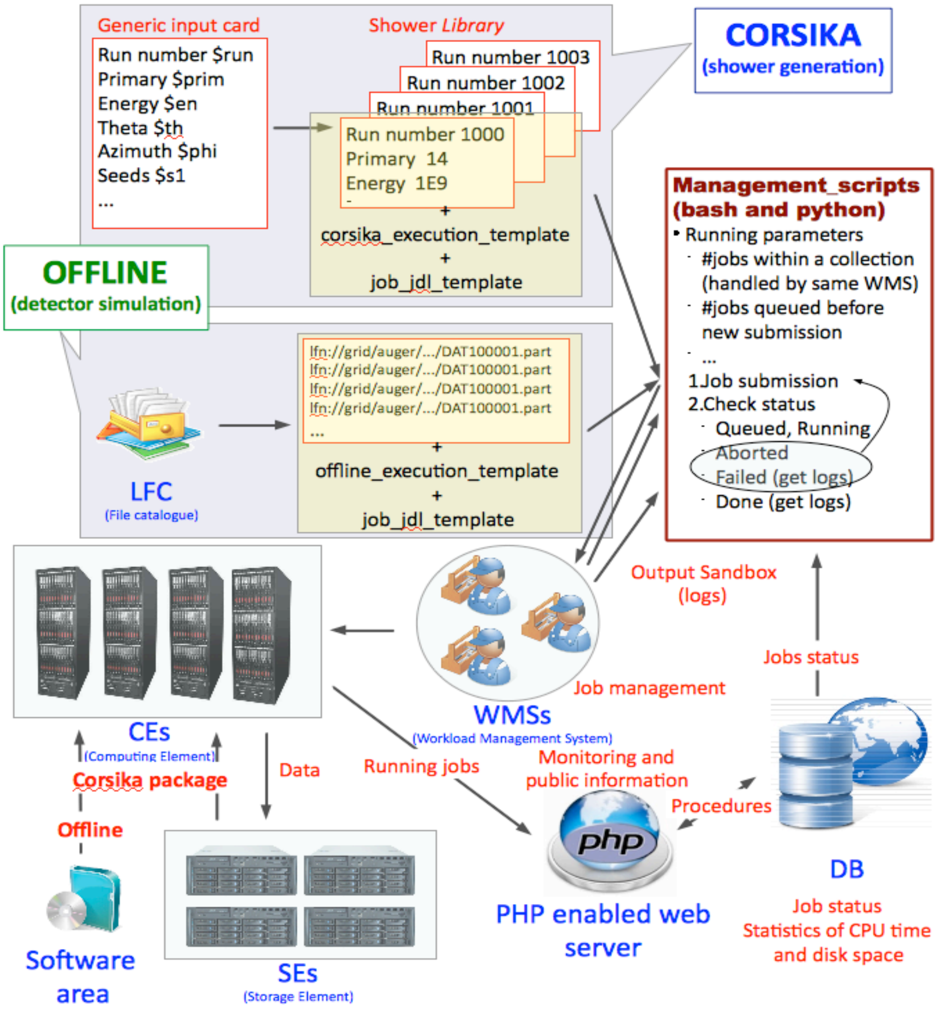

Figure 6. Workflow of the Auger simulation productions.

A set of parameters determines how the scripts behave:

- frequency of job submission depending on the ratio of running to queued jobs
- number of jobs submitted within a collection
- maximum number of jobs to be submitted in a submission cycle
- allowed number of aborted and failed jobs before interrupting the main script execution
- etc …

The individual jobs are submitted together as a collection of jobs until the requested number of total jobs has been submitted. At this time, aborted and/or failed jobs may be resubmitted.

The scripts react according to the status of the previously submitted jobs and determine whether more jobs must be sent. The status is flagged and transmitted, along with some additional information, to a database which is coupled to a PHP-enabled web server. Thus we have appropriate means to monitor constantly the progress of the production and we can also set-up public web pages with updated information.

Detailed information on successfully executed jobs is obtained from the output logs and stored in the database in order to have detailed information of consumed CPU times, output files sizes, sites involved and other aspects of the job execution.

To reduce the load of the jobs on the network we copy our software packages in the Sofware Areas (SA) of the sites, though we can also fetch the CORSIKA package from Storage Elements (SE) where it has been previously placed. CORSIKA can be compiled statically and be copied in a single operational package. On the other hand the Offline software has many dependencies and needs a complicated and time consuming installation procedure; it is also heavier. In both cases, installation jobs are submitted before any new *library* is started. By *library* we mean the set of showers simulated with the same CORSIKA compilation options, high energy and low energy models in particular, and the same kind of primary cosmic ray.

4. Performance of our Grid production model

The Auger Virtual Organization is a relatively young VO with access to a limited number of EGI (European Grid Infrastructure) [7] sites, including some in latin America, and a low number of *prioritized* CPUs: just about a few hundred. In spite of the continuous and diverse problems which we have faced, the average production rate has been very good. As seen in figure 7, the daily CPU consumption presents remarkable fluctuations, though the tendency as shown in the cumulative mean (light shaded area) indicates a rise of the CPU consumption since the beginning of the year 2011 until we reached a stable average at about 850 CPU*days.

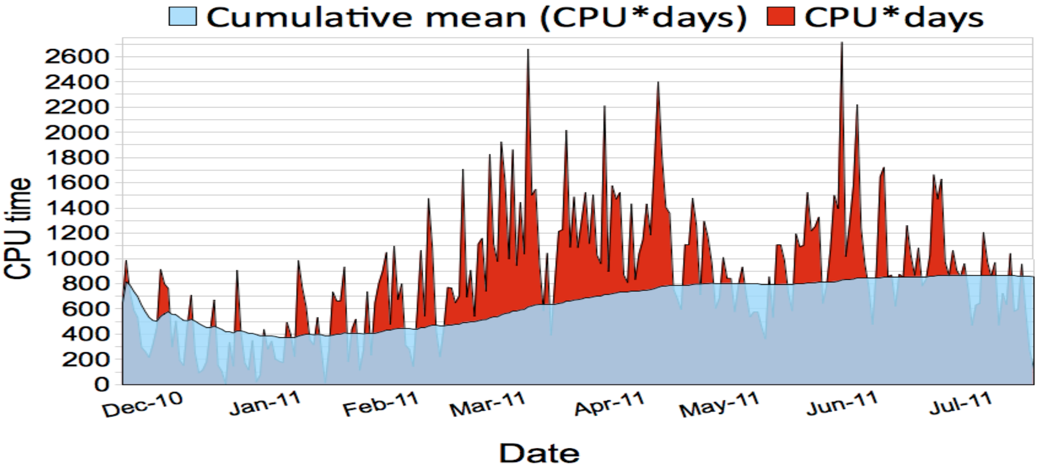

Figure 7. Daily CPU and cumulative mean CPU consumption of the Auger VO since end of 2010.

The continuous efforts to improve our set of automated scripts have paid off and we are being ranked consistently among the top 10 CPU consumers of the EGI Grid, only surpassed by the Large Hadron Collider (LHC) collaborations which dominate this statistic as shown in figure 8.

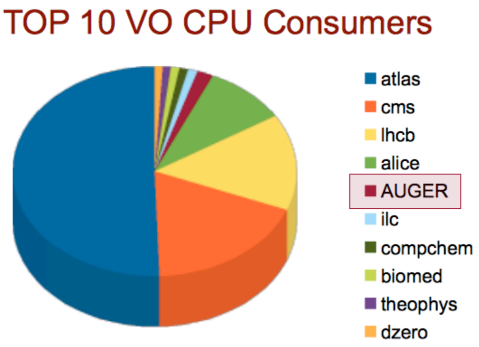

Figure 8. Percentage as pie chart of the top CPU consumers at EGI Grid. Auger ranks 5th.

Given the numbers presented for our consumed mean cumulative CPU time (figure 7) and the mean CPU time required for a shower with energy 10^{19} eV and zenith angle 38^0 (figure 2) we can calculate that the average daily number of iron showers of these characteristics we can produce is close to 2000.

5. Summary and conclusions

The Pierre Auger Observatory requires high numbers of simulated EAS which demand a very high level of computing time and storage space. Grid is an ideal technology for accessing the needed computing resources, although it suffers from the lack of stability. This can be specially the case for small Virtual Organizations where resources are concentrated on few sites.

We have developed a simple computing model based on scripts coupled to a web server and a database which make use of Grid middleware. The automation of all job related tasks and the development of monitoring tools have allowed us to produce hundreds of thousands of EAS to fulfill the needs of the Auger scientists. In spite of being a small team we have reached the 5th place in the CPU consumption ranking within the EGI Grid.

Acknowledgements

We gratefully acknowledge the contribution of the sites supporting the Auger Virtual Organization and the efforts of their administrators to maintain stable operation of their centers. The author also wants to thank the Auger Collaboration for its support to this project. He also wants to commend the remarkable effort put into this initiative by the members of the production teams at the University of Granada and at the FZU Institute in Prague.

References

[1] Abrahams J et al (Pierre Auger Collanoration) 2004 *Nucl. Instr. and Meth.* A **523**, pp 50-95.

[2] Pryke C 1998 *Phys. Dpt. Colloquium, Univ. of Wisconsin (Madison)* http://find.spa.umn.edu/~pryke/publ.html

[3] Sciutto SJ 2001 *astro-ph/0106044*, 27th ICRC Hamburg, August pp 7-15.

[4] Heck D Knapp J Capdevielle JN Schatz G Thouw T 1998 *Forschungszentrum Karlsruhe Report FZKA* 6019.

[5] Argiro S et al 2007 *Nucl. Instr. Meth.* A **580**, pp 1485–1496.

[6] Agostinelli S et al 2003 *Nucl. Instr. Meth.* A **506** pp 250-303.

[7] EGI, European Grid Infrastructure. http://www.egi.eu/