A tool for optimization of the production and user analysis on the Grid, C. Grigoras for the ALICE Collaboration

Costin Grigoras\textsuperscript{1}, Federico Carminati\textsuperscript{1}, Olga Vladimirovna Datkova\textsuperscript{1}, Steffen Schreiner\textsuperscript{1}, Sehoon Lee\textsuperscript{2}, Jianlin Zhu\textsuperscript{3}, Mihaela Gheata\textsuperscript{1}, Andrei Gheata\textsuperscript{1}, Pablo Saiz\textsuperscript{2}, Latchezar Betev\textsuperscript{1}, Fabrizio Furano\textsuperscript{1}, Patricia Mendez Lorenzo\textsuperscript{1}, Alina Gabriela Grigoras\textsuperscript{1}, Stefano Bagnasco\textsuperscript{4}, Andreas Joachim Peters\textsuperscript{1}, Maria Dolores Saiz Santos\textsuperscript{1}

\textsuperscript{1}CERN CH-1211, Geneva 23, Switzerland  
\textsuperscript{2}KISTI, Yusong, Republic of Korea  
\textsuperscript{3}Hua-Zhong Normal University, Wuhan, China  
\textsuperscript{4}INFN Torino, Italy

Costin.Grigoras@cern.ch

Abstract. With the LHC and ALICE entering a full operation and production modes, the amount of Simulation and RAW data processing and end user analysis computational tasks are increasing. The efficient management of all these tasks, all of which have large differences in lifecycle, amounts of processed data and methods to analyze the end result, required the development and deployment of new tools in addition to the already existing Grid infrastructure. To facilitate the management of the large scale simulation and raw data reconstruction tasks, ALICE has developed a production framework called a Lightweight Production Manager (LPM). The LPM is automatically submitting jobs to the Grid based on triggers and conditions, for example after a physics run completion. It follows the evolution of the job and publishes the results on the web for worldwide access by the ALICE physicists. This framework is tightly integrated with the ALICE Grid framework AliEn. In addition to the publication of the job status, LPM is also allowing a fully authenticated interface to the AliEn Grid catalogue, to browse and download files, and in the near future will provide simple types of data analysis through ROOT plugins. The framework is also being extended to allow management of end user jobs.

1. Preamble

In the period of steady LHC operation, the Grid usage is constant and high and, as foreseen, is used for massive raw and MonteCarlo (MC) production and also (quite successfully) for end user analysis. As a summary of last year’s -2010- activities, 5.8M MC production and analysis jobs, 1.8M raw data reconstruction and analysis jobs and 11.4M user analysis jobs were successfully executed. The 19M jobs used in total about 10000 CPU years, 10PB of storage space and 40PB of network traffic. Contributing to the Grid there are more than 90 centers on 5 continents while 410 people have analyzed data in this environment.

To help the Grid users and administrators, many applications have been developed in the early years of the Grid. ALICE has made an effort to consolidate all of these in a coherent set of monitoring and control tools. There are two main components we will be referring to in this article, AliEn (ALIce ENvironment)\textsuperscript{[1]} which provides a global file catalogue and means of running jobs and transferring data and MonALISA (MONitoring Agents using a Large Integrated Services Architecture)\textsuperscript{[2]} providing the monitoring and control part of the system.
2. Monitoring and control infrastructure

ALICE computing model splits the services required for Grid operations in two sets: central and per site services. The central services maintain the file catalog, jobs and transfer queues and offer authentication, authorization and accounting services. Together they provide a uniform Grid access to the users.

Each site runs a set of services on a dedicated local host (VoBox). These services are responsible for sending pilot jobs when jobs matching site resources are waiting in the queue, installing the required software packages, passing jobs’ messages to the central services while they are running and monitoring the usage of site resources.

The monitoring infrastructure is closely tied to the services distribution. On each site the VoBox runs both AliEn services and a MonALISA service, thus communication and monitoring stay within the site as much as possible. Together with AliEn central services a MonALISA Repository instance is collecting information from all sites all over the world, subscribing as much as possible to aggregated data series from the sites in order to offer an overall view of the system.

Data collection is largely based on the ApMon (Application Monitoring) library. Most of the AliEn services are instrumented to send periodic and event-based monitoring information to the closest MonALISA service. ApMon is also capable of sending host monitoring information from a parallel thread, thus providing computing fabric monitoring. Additionally a particular process ID - and its children - can be tracked for specific resource consumption. This feature is used by the AliEn Job Agents (job pilots) to account for the resources used by each job. At the site level filters aggregate the accounting information per site and per user, providing digested information to the repository. Figure 1 below shows how the AliEn services running on the VoBox send monitoring information to the local MonALISA services that in turn aggregate the values and send them to the central repository for archival, displaying and taking automatic actions.

Figure 1: collecting monitoring information in the central MonALISA Repository for ALICE

Similarly other data sources are instrumented to send monitoring information to the central MonALISA Service instances. In particular the raw data registration service reports information about each raw data chunk that is collected by the experiment while Shuttle (ALICE’s service for extracting and publication of condition data) reports processing status for each detector-specific algorithm for each raw data run number.
3. Lightweight Production Manager (LPM)

MonALISA central Repository correlates many sources of information and runs this automatic production manager, LPM, triggering it whenever the number of queued jobs of particular activities falls below preset thresholds (A). When triggered LPM tries to fill the queue with more jobs so that at least another threshold (B) is reached (with B>=A).

To achieve this target LPM first looks at all jobs for the respective activity that it has submitted but have not completed yet. If there are subjobs that failed for any reason they are resubmitted until a preset completion percentage target is reached or until it has tried too many times. If after this step all subjobs are in final states (either success or failure) then LPM checks for dependencies in the processing chain and if the completion target is satisfied it starts the dependent task(s).

If after this check there are still too few jobs waiting in the queue then LPM looks for top tasks that can be started and submits them if possible.

3.1. Running MonteCarlo productions

LPM chains for MonteCarlo productions usually point to a particular job description file (JDL) in the AliEn catalogue and have a sequential run number as argument. Whenever a new chain is defined in the web interface it automatically suggests a unique run number range. LPM submits the jobs following the run numbers in the sequence until the end of the predefined range when it disables the entry.

Several productions can be run in parallel, and by a simple weighting system the relative priorities of them can be set.

Other operating modes are executing a given task with run numbers from a list instead of a range, for example when running MonteCarlo simulation jobs with real raw data runs as anchors or more generic calling a custom Java class that determines what needs to be executed next based on its internal algorithm.

3.2. Raw data automatic reconstruction and analysis

This activity got the highest priority in the past year. Following ALICE’s computing model, raw data is promptly reconstructed at Tier0 (CERN) and mirrored as soon as possible to one of the Tier1 ALICE centers.

LPM is in charge of keeping the Tier0 resources busy by running reconstruction and analysis jobs as soon as data is available. For this custom Java plug-ins are chained in LPM and they correlate the various data sources to see which new raw data runs are available, whether or not they are fully transferred to the main storage element, checking if the run has been marked as “good for reconstruction”, if the calibration data was successfully stored for it and so on. Only these runs are automatically reconstructed, though any runs can be manually added to the processing queue and LPM will run them too as soon as there are free resources.

If the decision is taken to reconstruct a particular run then it is also scheduled for mirroring to a Tier1. To select one the monitoring information is taken into account, making sure the target storage is functional and the allocated space matches the distribution ratio indicated by the MoU-s. A mechanism similar to the job management is taking care of the data transfers.

Subsequent reconstruction passes are similarly processed, the production manager marks a set of run numbers to be reprocessed and LPM handles the actual submission.

Analysis jobs can follow the reconstruction automatically or be executed at a later time over all the runs in a given period or range. Figure 2 below shows a part of the processing chain for raw data where the analysis jobs are automatically executed following a successful reconstruction.

To clarify this particular example, the top entry (ID 355) is an extra calibration task. The job splits in as many subjobs as files in the run. If at least 95% of the subjobs are successful LPM executes the next task in the chain (ID 356) that merges the output from all these subjobs in a single file and registers the new calibration object. Similarly ID 357 is the actual raw data reconstruction job, which if successful triggers three jobs in parallel (ROOT tags merging, central analysis train and Quality Assurance) and so on. There is no limit to either the depth or the width of the dependency tree.
allowing defining complex scenarios such as merging output in several stages to limit memory utilization in single jobs.

| ID  | JHL                              | Parameters | Parent-run completion | Target run completion | All run user weight | Options               | Override               |
|-----|----------------------------------|------------|-----------------------|-----------------------|---------------------|-----------------------|------------------------|
| 351 | #parent LHCF                  |            | 90%                   | 100%                  | 100%                | Add dependency       | Edit       | Delete | Disable | Execute |
| 352 | #alicesim Merge               | #OUTPUTDIR #1 aliic:/alice... | 95%                   | 100%                  | 100%                | Add dependency       | Edit       | Delete | Disable | Execute |
| 357 | #V0112                        |            | 100%                  | 100%                  | 100%                | Add dependency       | Edit       | Delete | Disable | Execute |
| 363 |Alicearming /63b/br (Binary)PD/19/TAGC/Pergeutsch/beaustageage2.pdf (Odd) |            | 100%                  | 100%                  | 100%                | Add dependency       | Edit       | Delete | Disable | Execute |
| 365 | #alicesim AIC 321              |            | 100%                  | 100%                  | 100%                | Add dependency       | Edit       | Delete | Disable | Execute |
| 366 | #alicesim Merge               |            | 10%                   | 10%                   | 10%                 | Add dependency       | Edit       | Delete | Disable | Execute |
| 367 | #alicesim CFA/daf/29         |            | 100%                  | 100%                  | 100%                | Add dependency       | Edit       | Delete | Disable | Execute |
| 368 | #alicesim Merge               |            | 10%                   | 10%                   | 10%                 | Add dependency       | Edit       | Delete | Disable | Execute |

Entire branches of the tree can be temporarily disabled if needed, and can be later manually executed on the missed jobs (for example when there is a bug in the processing software that is solved in a subsequent release).

Up to now, 755 production cycles have been handled by LPM, representing all central MonteCarlo and raw data-related activities, so 7.6M jobs in 2010.

4. Catalogue and jobs management

Central MonALISA Repository has a direct connection to AliEn via the command line interface of it. This connection is used for all management operations (LPM and scheduled transfers) and also for all user commands, presenting the results in a user-friendly web interface. All operations require certificate-based authentication, using the same authentication and authorization mechanisms as AliEn itself.

Integrated in the same web interface users query for monitoring and accounting values are a catalogue browser featuring file viewer with syntax highlighting and catalogue links, a text file editor, upload and download of any file, physical file location checker (including storage element current status reported by the monitoring part), file quota display and so on. From the catalogue browser users can also directly submit jobs when interacting with JDL files.

Another view allows users to interact with their jobs, from simply listing the active jobs to full tracking of the subjobs, accessing log files and resubmitting failed ones.

![Figure 2. LPM chain for raw data automatic reconstruction and analysis](image)

![Figure 3. Detailed view of a job and all its subjobs](image)
Figure 3 above is a snapshot of the job detailed view. On top the meta information of the job is displayed (comment, output directory, referenced software packages and versions) while most of the view is dedicated to the individual subjobs. The information provided by AliEn is combined with the quasi-real time monitoring of the jobs from MonALISA to generate live charts of the memory utilization for example (this being a hot topic for users) and to show where the job ran, for how long, how long did it take to save the output and so on.

The summary at the bottom of the page is very useful in spotting sites where particular problem show up, such as too long saving times that indicate storage problems or excessive failed jobs numbers usually pointing to inaccessible shared software area on the site or corrupted software package installation.

By collecting memory usage information from all active jobs the central repository also alerts the users (and the Grid managers as well) when the memory utilization grows above preset thresholds. The email notification includes full details of the offending process and the memory profile until then giving the user as much information as possible to be able to reproduce the problem (input data set, software versions, exact host where the job ran and so on). Since the problem is usually related to a particular configuration, this notification is only send every few hours if the problem persists. If however the memory profile shows a leak the user is notified more frequently (not more often than half an hour though) so that the worst cases are indicated for later analysis.

5. Conclusions
The Grid has entered a period of heavy and diverse use. To improve the efficiency of Grid operations, ALICE has developed and deployed a coherent set of tools for monitoring and steering the central and user-oriented computational tasks. These tools have been used in production for the entire 2010 to operate the MC and raw data production as well as to help the end users follow on their individual jobs and they have proved to be very useful in both cases. The general development direction is to provide the LPM functionality to end users as well to allow even simpler management of their tasks and continue extending the web interface to provide the full set of operations that are available via command line then start implementing interactions with ROOT files and objects.

References
[1] Bagnasco S, Betev L, Buncic P, Carminati F, Cirstoiu C, Grigoras C, Hayrapetyan A, Harutyunyan A, Peters A J and Saiz P 2007 AliEn : ALICE environment on the GRID, J. Phys.: Conf. Ser. 119
[2] I. Legrand, H. Newman, R. Voicu, C. Cirstoiu, C. Grigoras, C. Dobre, A. Muraru, A. Costan, M. Dediu and C. Stratan 2009 MonALISA: An agent based, dynamic service system to monitor, control and optimize distributed systems, Computer Physics Communications, Volume 180, Issue 12, December 2009, Pages 2472-2498
[3] Cirstoiu C, Legrand I, Voicu R and Stratan C 2006 ApMon http://monalisa.caltech.edu/monalisa__Download__ApMon.html
[4] The Scalla/XrootD Software Suite. July 2009 http://savannah.cern.ch/projects/xrootd/, http://xrootd.slac.standford.edu/
[5] RFC 1930, http://tools.ietf.org/html/rfc1930
[6] MonALISA Repository for ALICE, 2010 http://alimonitor.cern.ch/
[7] C. Zampolli, F. Carminati and A. Colla 2010 The SHUTTLE: the ALICE Framework for the extraction of the conditions Data , ACAT2010, Proceedings of Science, http://pos.sissa.it/archive/conferences/093/066/ACAT2010_066.pdf