

## Distributed Analysis Experiences within D-GRID

J. Elmsheuser, O. Biebel, G. Duckeck, J. Kennedy, G. Kroboth, B. Ruckert, D. Schaile, and C. Serfon

The distributed data analysis using Grid resources is one of the fundamental applications in high energy physics to be addressed and realized in the near future [1]. An efficient analysis environment and the know how to use and enhance it are key goals for the community to achieve, if we are to profit from the high investments made into the accelerator and detectors at the LHC.

The needs to manage the resources are very high. In every experiment up to a thousand physicist will be submitting analysis jobs into the Grid, namely LCG [2], the grid flavor developed especially for the large hadron collider LHC. Appropriate user interfaces and helper applications have to be made available to assure that all users can use the Grid without too much expertise in Grid technology. These tools enlarge the number of grid users from a few production administrators to potentially all participating physicists.

Within the D-Grid High Energy Physics Community Project [3] we are working on distributed and interactive data analysis on the Grid. In this context a gap analysis has been pursued to identify missing features and components of distributed analysis tools. All this was done with a closer look into the computing environment, Athena, of the ATLAS experiment [4]. An automatic job manager and scheduler should fulfill the following specifications or functionalities: interface for job configuration, job submission interface for Grid and batch systems, integration of data management, resource estimation, job monitoring, job error checking, collecting and merging of the results and job archive.

The job and scheduling manager GANGA [5] proved to be a very good candidate and was closely examined. GANGA (Gaudi / Athena and Grid Alliance) is an interface to the Grid that is being developed jointly by the ATLAS and LHCb experiments. GANGA is a front end for job definition and management of analysis jobs to run in a distributed environment. It helps in the creation and configuration of user analysis jobs, submission of the jobs, monitors job status and helps in saving any output. In particular GANGA aims to help with setting up jobs that run the standard ATLAS and LHCb applications. It can be run on the command line, with Python scripts or via a graphical user interface. Many of the required features are already included in GANGA, but several are missing or some need further refinement. GANGA has been used in tests and real user interaction. It performs well in configuring, submitting, monitoring and output retrieval of a few hundred to thousand jobs. Figure 1 illustrates the workflow a job scheduler is performing during job execution.

We have extended the functionality of GANGA in numerous areas. The most important are the extension of job splitting for ATLAS jobs and the integration of the ATLAS data management system DDM/DQ2 with direct access to the input data files via POSIX I/O. These items are corner stones for an efficient distributed analysis system.

The job parallelization functionality provides a splitting based on input files or input parameters. The number of

input files of a dataset are evenly distributed among the parallel jobs and executed independently at the same time. Similarly jobs can be started in parallel with different parameter sets. Since high energy physics applications in general are easily dividable into independent tasks, the performance increases linearly with the number of jobs. Due to the integration of the ATLAS data management system DDM/DQ2 users only need to provide a single input dataset name to process all corresponding data files on the Grid. The Grid jobs are sent to the site where the data is located. All additional input data staging and preparation is done automatically for the user. The necessary domain and host-name recognition of the site for the DDM/DQ2 integration is an important part of the procedure. A direct access via POSIX I/O to the input files stored on a gLite compatible storage element such as dCache or Castor is provided via ROOT [6] protocols. Thus large datasets can be processed without the need of downloading input files to a temporary location on the Grid site.

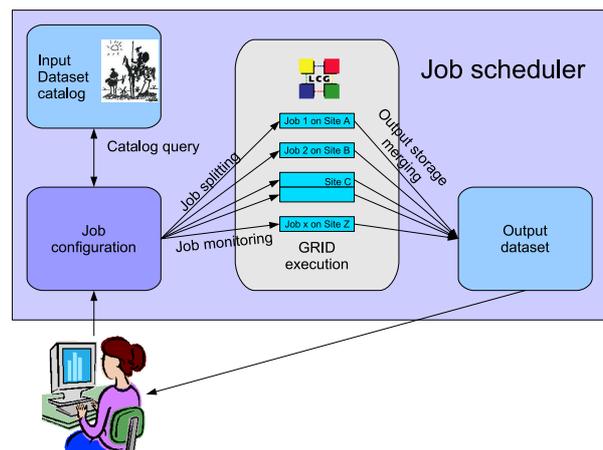


Fig. 1: Workflow of the job scheduler GANGA.

We have added several further functionalities to extend and ease the usage of the job scheduler GANGA. Grid jobs can be started using the Condor-G interface. This significantly enhances the speed of bulk job submission compared to the submission using the LCG resource broker. A plugin for merging of output data and output log-files has been implemented. Output files of Grid jobs can be downloaded from remote storage elements in background threads and merged into a single output file.

Numerous tests using GANGA have been carried out at different Grid sites like the Tier 1 computing centers at Karlsruhe and Lyon. Since the beginning of the year there are over 300 unique users at different sites.

### References

- [1] D. Baberis *et al.*, Common Use Cases for a HEP Common Application Layer for Analysis, LHC-SC2-2003-032
- [2] LCG project web page: <http://lcg.web.cern.ch/LCG/>
- [3] D-Grid HEPCG project web page: <https://www.d-grid.de/>
- [4] ATLAS Computing TDR, CERN-LHCC-2005-022
- [5] GANGA project web page: <http://ganga.web.cern.ch/ganga/>
- [6] ROOT project web page: <http://root.cern.ch/>