Distributed Analysis Experiences within D-GRID

J. Elmsheuser, G. Duckeck, J. Kennedy, and D. Schaile

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 physicists will be submitting analysis jobs into the Grid. 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.

Since September 2005 five community projects (astro physics, climate research, particle physics, engineering science, medicine, arts) and the D-Grid [2] integration project (DGI) started within the D-Grid consortium to build a sustainable Grid infrastructure in Germany. This infrastructure will help to establish methods of e-science in the German scientific community. The community projects will develop together with the integration project a general and sustainable Grid-infrastructure, that will be available for all German scientists.

In the context of the German D-Grid project, many aspects of Grid tools have been assessed. Our group focusses on the review and further development of existing Grid middleware and tools for distributed and interactive analysis. A gap analysis is pursued to identify missing features and components. All this is done with a closer look into the computing environment, Athena, of the ATLAS experiment [3].

An analysis job at the ATLAS experiment will typically consist of a Python or shell script that configures and runs a user algorithm in the Athena framework, reading and writing event files and/or filling histograms/n-tuples. More interactive analysis may be performed on large datasets stored as n-tuples. The distributed analysis system must be flexible enough to support all work models depending on the needs of a single user or an analysis team. A distributed analysis system should be robust and easy to use by all collaboration members. The look and feel of the system should be the same whether one sends a job to one's own machine, a local interactive cluster, the local batch system, or the Grid.

There are several scenarios relevant for a user analysis: (1) analysis with fast response time and a high level of user interaction, (2) analysis with intermediate response time and interaction, (3) analysis with long response times and a low level of user interaction.

The first point is well matched by the parallel ROOT facility PROOF [5] for interactive usage and fast turn around times on a local computing cluster. Similarly, DIANE [6] can be used on a local computing cluster or Grid environment for fast response and parallel processing but with lower user level interaction. For the second and third point an automatic job manager and scheduler in a distributed analysis environment is the key feature for a robust system. Here the distributed analysis environment GANGA [4] is planned to serve the need for a common user interface for user analysis job configuration, scheduling and submission to different Grid flavors.

GANGA has been successfully extended to incorporate different functionalities: a plugin was written to execute generic and experiment specific user programs on PBS batch systems. This extension was developed and intensively tested at the German Tier1 computing site Gridka in Karlsruhe. There also exists a prototype for the parallel execution of several ATLAS framework jobs. This plugin is used to split Athena jobs into several sub jobs based on the data set. The sub jobs can be executed on local or remote computing systems.

A typical example for a user task is a small scale Monte Carlo production of a few ten-thousand events. This was exercised using GANGA and the ATLAS experiment components for generation, simulation and reconstruction. This scenario mimics the analysis patterns mentioned before of intermediate to low user interaction and intermediate to long response times.

The Monte Carlo event production was done in several steps: event generation, simulation, digitization and reconstruction. For each step jobs were sent to 3 different German Tier1/2 sites in separate grid jobs with input and output files and results stored on the grid storage element at Gridka in Karlsruhe. Every job consisted of a start and wrapper script that configured the different Athena settings and input and output datasets. The processing of the datasets, that consisted of a few thousand events, had all been parallelized into sub jobs of 50 events each.

GANGA performs well in this test of configuring, submitting, monitoring and output retrieving at a scale of a few hundred jobs. The submission time of a single job to the LCG is about 10-20 seconds, ie. submission of a few hundred jobs need a bulk submission feature as in gLite [7]. Furthermore, the error handling and recovery of failed jobs in the user analysis code needs to be improved by an automatic resubmission or error parsing. This could be assisted by a bookkeeping mechanism of the processed data sets.

To conclude, the distributed data analysis using Grid resources is one of the fundamental applications in high energy physics that is being used in the upcoming phase of LHC data taking. Several different user analysis scenarios require different response times and levels of user influence.

References

[1] D.Baberis *et al.*, Common Use Cases for a HEP Common Application Layer for Analysis, LHC-SC2-2003-032
[2] D-Grid project webpage: `http://www.d-grid.de/`
[3] ATLAS Computing TDR, CERN-LHCC-2005-022
[4] GANGA project webpage: `http://ganga.web.cern.ch/ganga/`
[5] ROOT and PROOF webpage: `http://root.cern.ch/`
[6] DIANE project webpage: `http://cern.ch/diane/`
[7] gLite project webpage: `http://glite.web.cern.ch/glite/`