Automating of Job Submission with Athena/AthenaMC in GANGA

T. Mahmoud, J. Ebke, O. Biebel, G. Duckeck, J. Elmsheuser, J. Kennedy, G. Krobath, B. Ruckert, D. Schaile, M. Schott, and C. Serfon

Because of the huge amount of data expected at LHC the usage of Grid resources for distributed data analysis is indispensable. The complexity of the Grid makes this usage difficult for unexperienced users. Therefore the four LHC experiments developed interfaces to their software with the Grid which allow users to exploit the Grid resources keeping the complexity of these resources hidden from them.
GANGA (Gaudi / Athena and Grid Alliance) [1,2,3] is the Grid interface of the ATLAS and LHCb experiments. As a front end, GANGA manages, schedules, and sets up jobs which run the standard ATLAS and LHCb applications, Athena and Gaudi respectively. This job management implies the job definition, creation, submission, status monitoring, and output saving.

Various reasons lead to job failures on the Grid. In most cases neither the user nor the software (GANGA, Athena, or Gaudi) has any bearing on these failures. The only thing a user can do in such cases is the resubmission of the failed job(s). This demands ample time, attention and efforts especially if the user is running some hundreds or even thousands of jobs simultaneously which can lead to unclear and confusing situations. The complexity increases if a user runs Monte-Carlo (MC) simulations where the process goes through three different correlated and chained stages. That is the output of certain jobs is required as input for others.

GangaTasks [4] is a job management tool which was developed within GANGA to facilitate job management. In particular it automates job submission, resubmission of failing jobs, and splitting and chaining of jobs with the Athena and AthenaMC [5], the ATLAS packages for analysis and MC simulations respectively.

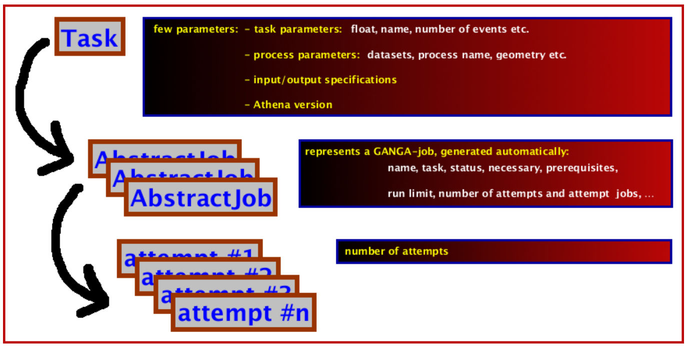
Task
few parameters: – task parameters: float, name, number of events etc.
– process parameters: datasets, process name, geometry etc.
– input/output specifications
– Athena version
AbstractJob
AbstractJob
AbstractJob
represents a GANGA-job, generated automatically:
name, task, status, necessary, prerequisites,
run limit, number of attempts and attempt jobs, ...
attempt #1
attempt #2
attempt #3
attempt #n
number of attempts

Fig. 1: Concept of tasks.

The concept of GangaTasks is the creation of job collections called *tasks*. Users create such tasks aiming to achieve MC simulations or data analysis. The job collection in a task, called *abstract jobs*, is created automatically according to few settings made by the user which tell the task what to do. An abstract job represents a GANGA-job which is submitted to a specified back-end (local batch system or Grid). Ideally the submitted GANGA-job completes successfully, returning the required output. If this is not the case, i.e the job fails, a new attempt of the GANGA-job starts and the job is resubmitted. This procedure of resubmission goes up to a given number of attempts. If exactly this number is reached, the job is declared as *ignored* and the user has to investigate manually why it failed.
In addition to automate jobs (re-) submission the splitting and chaining of MC-jobs run automatically. A task recognizes which job needs which input and waits with submission until that input is ready. Figure 2 illustrates the structure of jobs chaining created by GangaTasks for a small MC-job, see caption.

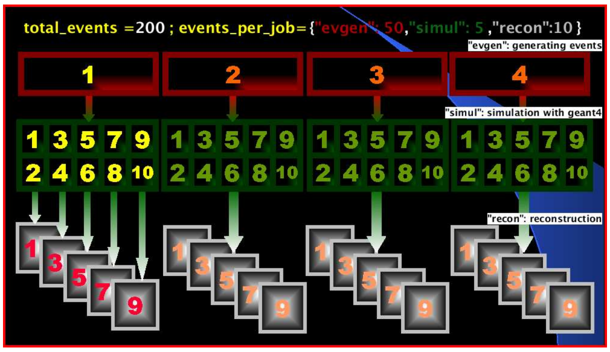
total_events =200 ; events_per_job={"evgen": 50,"simul": 5 ,"recon":10 }
"evgen": generating events
1
2
3
4
"simul": simulation with geant4
1 3 5 7 9
2 4 6 8 10
"recon": reconstruction
1 3 5 7 9

Fig. 2: A test task simulating 200 events of a physical process. The user can tell the task how many events should be handeled in a job of a stage. GangaTasks creates the jobs and arranges the correlation between them.

Another important feature of GangaTasks is *throttling*, where the user can specify how many jobs are allowed to run simultaneously on the Grid. This is important to find out if the settings, the applications, and the running environment of the task's jobs are proper. The user can run a small number of jobs, if they run successfully, he can increase the number of simultaneously running jobs.
Furthermore, GangaTasks insures secure results. It protects sensitive parameters (those which can influence the output) from being changed as soon as a task is submitted. Nevertheless the package offers a series of possibilities to achieve controlled changes, if required.

The package was successfully tested at CERN (Tier-0), GridKa (Tier-1) and LMU (Tier-2) within the ATLAS software. In the coming future it will be fully integrated into GANGA, i.e. LHCb users will be able to use it.

References

[1] GANGA project web page, http://ganga.web.cern.ch/ganga/
[2] Annual report 2007
[3] Annual report 2008
[4] Wiki page, https://twiki.cern.ch/twiki/bin/view/Atlas/GangaTasks
[5] ATLAS Computing TDR, CERN-LHCC-2005-022