

Moving the LHCb Monte Carlo production system to the GRID

Authors

E. van Herwijnen, P. Mato (CERN), F. Harris (Oxford), G.N. Patrick, R.A. Sansum (RAL), N. Brook (Bristol), M. McCubbin, G.D.Patel (Liverpool), A. Khan (Edinburgh), A. Tsaregorodtsev (Marseille), D. Galli, U. Marconi, V.Vagnoni (Bologna), S. Klous, H. Bulten (Nikhef)

Abstract

The fundamental elements of the LHCb Monte Carlo production system are described, covering security, job submission, execution, data handling and bookkeeping. An analysis is given of the main requirements for GRID facilities, together with some discussion as to how the GRID can enhance this system. A summary is given of the first experiences in moving the system to a GRID environment. The first planning for interfacing the LHCb OO framework to GRID services is outlined.

Keywords *Simulation , Framework, GRID*

1) Introduction

The LHCb Monte Carlo production system is being gradually moved to a GRID environment as part of the HEP Applications(WP8) activities within the DataGrid project. It is being used as a short/medium term use-case to define requirements for the DataGrid middleware, and also to test the first software releases from the project. We expect from this work the immediate benefit of obtaining homogeneous interfaces to the heterogeneous systems comprising the distributed facilities available to LHCb, and later advantages derived from the availability of flexible, fault-tolerant middleware.

2) LHCb Monte Carlo Production Environment

The current LHCb Monte Carlo code ("SICBMC") is Fortran-based, and uses Pythia 6.134 and QQ to generate events, and Geant 3.1 to track them through the detector. Events are written out in the form of Zebra banks. The reconstruction program ("SICBDST") is also Fortran based. A transition to OO software is underway, but this should be transparent to our usage of the GRID for the same mode of operation.

Our Monte Carlo code changes frequently so it is more convenient to do the steering of the MC production through scripts rather than code. However for the interfacing of the analysis software we certainly need the API interfaces, since data accesses will depend on selections made dynamically during code execution.

Table 1 shows the current LHCb MC production uses facilities at CERN, RAL, IN2P3 (Lyon) and Liverpool. The indicated size is the maximum number of CPUs ("computing elements") that we have been able to use simultaneously in production.

Centre	OS	Max. # of CEs used simultaneously	Batch system	Typical weekly production (#k of events)	Percentage submitted through the Grid
CERN	Linux	315	LSF	85	10%
RAL	Linux	30	PBS	35	100%
IN2P3	Linux	225	BQS	35	100%
Liverpool	Linux	300	Custom	150	0%

Table 1 LHCb distributed MC production

It is planned to extend this to Nikhef, INFN/Bologna and Glasgow/Edinburgh in 2001. Current event sizes are of the order of ~1 MB, and the current maximum rate for event production is at Liverpool, which can produce ~ 1 event/second.

The overall system environment is heterogeneous with respect to the batch control systems. However, as of early 2001, all farm production is Linux based. It should be noted that NT is still used for the Web servers for job submission, and remains a possibility as a second production platform for LHCb. Before entering into discussion of the system components we can state 2 major requirements which apply to all of the initial work with GRID software.

Requirement 1: The DATAGRID middleware should be callable from (in order of priority): The Linux command prompt, Shell scripts, Java programs(i.e. as a Java class library), and C++ (i.e. as a C++ class library).

Requirement 2: A working, standard, supported Globus installation kit should be available to all of our centres as part of the Testbed environment.

3) Job submission system

The LHCb MC jobs are submitted by the production manager using a Web page. A Web server may serve several farms, provided that AFS, or an equivalent service from GRID middleware, is available. We hope that GRID developments will allow us to run the whole system with just one server.

The production manager fills out a form specifying the number of jobs to be run, the event -type, number of events, kinematical parameters, batch queue selection and version numbers for the executable and detector database. A Java "servlet" generates the required job scripts, and also creates files specifying random number seeds and job options. It then issues a job submit command to run the script in batch.

We can currently also submit jobs to Lyon CCIN2P3 and RAL from the web server from CERN. Looking to integration of all our facilities in the GRID we have the following requirements :

Requirement 3: The testbeds participating in DATAGRID should have a "public" webserver available, from which facilities are reachable (e.g. LSF, PCSF, HPSS, AFS, CASTOR, SHIFT, ORACLE etc.) through the GRID, preferably without additional authentication.

Requirement 4 (security): The Datagrid should set up a mechanism whereby the necessary accounts for a user can be set up on all of the testbed sites via a single request. Similarly, a single certificate should be necessary to utilise the entire testbed, and sites should be configured to accept the certificates of all the participating certificate authorities so that access to the grid is transparent to the user.

Requirement 5 (security): As our jobs can run for a great length of time (several weeks) we need lasting security tokens for all resources that the job may access during this time.

4) Job scripts running the SICB executables

The job script performs the following functions :

- (a) Copies the executable, detector database and job files to an area accessible to the job.
- (b) Executes the job.
- (c) If necessary copies data output to mass storage from local job storage.
- (d) Copies the job log file to a web browsable area.
- (e) A Java program is called to transfer data to CERN, if the job ran remotely, and update the Oracle database at CERN.

Requirement 6 (job scripting language): A common scripting language should be supported across the Datagrid, and Perl or Python are the LHCb preferred choice.

5) Storage of output datasets

The size of our datasets is between 0.5 and 1.5 Gbytes (500 events). For jobs that do generation, reconstruction, and pileup, there can be three datasets requiring of the order of 3 Gbytes. In the short term, we foresee the need to access specific mass storage systems at CERN, RAL and IN2P3:

- At CERN data are written directly onto CASTOR.
- At RAL data are copied via the 'tape' command to the RAL DataStore.
- At IN2P3 we use HPSS.

Requirement 7 (access to mass storage): The middleware giving us access to mass storage should be robust enough to handle many tens of simultaneous requests (today, in future growing to thousands of simultaneous requests). In case of error the system should interact with the batch facility and automatically put running jobs on hold and prevent new jobs from starting on a faulty node until the problem is fixed.

Requirement 8 (access to mass storage): The identification of files on the mass store should be possible via a hierarchical path and filenames of arbitrary length.

In the longer term, we expect to be able to use generic GRID data access tools.

6) Read/Write Access to User Data

This should be transparent all over the GRID.

Requirement 9 (access to user data): There should be a common API used to read and write data on the GRID.

Data generated outside CERN are replicated at CERN. A servlet at CERN is notified of the existence of some data and proprietary utilities are used to transfer the data to CASTOR at CERN.

Requirement 10 (Fast Data Transfer/Replication): A standard mechanism is required for transferring/replicating datasets over the GRID employing the fastest possible techniques (e.g. parallel transfers). Any data transfer mechanism should be robust and have the ability to recover from errors and transmission breaks. This must support a variety of formats (e.g. Zebra, ROOT).

7) Bookkeeping data base

Our bookkeeping database is an Oracle database on the central CERN Oracle service. Our Monte Carlo program queries the database which only knows about data on tape at CERN. Our requirements for the development of the system are :

Requirement 11 (meta data base-data cataloguing etc): The meta data base should be distributed (i.e. as soon as it is updated somewhere, the whole GRID knows about it). It should be integrated with the job submission system so that when an input dataset is required, the job submission system sends the script to where the data resides. It should have similar performance and accessibility as Oracle.

8) First experience of using Globus on Testbed0

LHCb has successfully adapted its MC production tools to automate production on the CERN Testbed0. Both RAL and IN2P3 have interfaced their batch farms to Globus gateways and we are now submitting all our production through these gateways. Using Globus we are able to remotely submit

jobs from the machine at CERN that runs the java servlets, whereas before a manual step was required (logging on to the remote system). Some "teething" problems with Globus had to be solved and some points are being discussed with the Globus team.

The percentage of production using the Grid is shown in Table 1.

9) Plans for testing first releases of Datagrid middleware

The first releases of DataGrid middleware will be available from Oct 1 (the so-called M9 release). We foresee to test the software provided by the following DataGrid work packages (WP): WP1 (basic job submission), WP2 (file copying), WP5 (mass storage handling). We will also use the software installation framework provided by WP4.

10) Interfacing GAUDI to GRID services

A project is commencing to interface the services of GAUDI, the LHCb software framework, to GRID services. For example, the 'Job Options' service of GAUDI will be interfaced by an applications user interface to the work scheduling services offered by Datagrid WP1. The application software will convert from high level requests such as 'the set of B->pi,pi events taken in runs MMM->NNN' to requests for files using experiment dependent metadata information. It should be noted that analysis jobs will often not know 'a priori' the set of files necessary for the job since this will be dependent on physics selection criteria information applied to 'event tag' data. Thus the analysis environment will necessitate statistically based evaluators on the 'cost' of jobs. This is quite different to the MC production environment which is part of the production environment of the experiment.

This work is proceeding in conjunction with the development of the LHCb analysis model for which prototypes are being developed. As such the LHCb model for using the GRID will evolve in an ongoing cycle of requirements, analysis, design, implementation and prototyping.

References

- [1] G. Barrand et al., *GAUDI: A Software Architecture and Framework for building HEP Data Processing Applications*, CHEP 2000 proceedings (see also <http://cern.ch/Gaudi>)
- [2] S. Amato et al., *LHCb Technical proposal*, CERN/LHCC 98-4.
- [3] SICB Users Guide, <http://lhcb-comp.web.cern.ch/lhcb-comp/SICB/html/sicbug.html>
- [4] EU DataGrid Project Home Page, <http://www.eu-datagrid.org/>