# DEMONSTRATION OF PRODUCTION OF MONTE CARLO SIMULATED DATA OF THE CMS EXPERIMENT USING THE EU DATAGRID TESTBED

Dr D. J. Colling (Imperial College London), on behalf of CMS and the GridPP Collaboration

**Key words to describe the work:** Applications, EU DataGrid, Particle Physics, CMS

**Key Objectives:** One of the key objectives of the EU DataGrid project is to provide the computing grid infrastructure required by the experiments at the Large Hadron Collider (LHC) at CERN. Here we demonstrate progress towards that objective.

**Motivation for the work (problems addressed):** To provide a grid solution to the task of producing Monte Carlo simulations of the data that will be produced by the CMS experiment at the LHC. The task of producing Monte Carlo simulated data is one common to all Particle Physics experiments.

## The Application

Particle Physicists in the UK are preparing for the turning on of the experiments at the LHC[1] at CERN[2] in 2007. These experiments will produce vast quantities of data (millions of GigaBytes per year). The processing of these data will be performed over large computational Grids. Currently large quantities of Monte Carlo simulated data are required in order to optimise the design of the detectors and to understand issues associated with the triggering and detector performance. The CMS experiment[3] is one of the two general-purpose experiments that will run at the LHC. Here we demonstrate how Monte Carlo simulated data of the CMS experiment can be produced using middleware developed by the EU DataGrid project. The Monte Carlo simulation that we are demonstrating is a two stage process. The first stage, CMKIN, simulates the initial interaction of the colliding protons and the hundreds of particles produced. All short-lived particles are decayed leaving only stable and long-lived products. This stage of the simulation is comparatively fast taking only a fraction of a second[1] to simulate each collision and write out approximately 120KB per event. The second stage, CMSIM, is the simulation of the particle interactions as they traverse the material of the CMS experiments and its sub-detectors. CMS is a very large detector and so the accurate simulation of the particles traversing its volume takes approximately 1 minute of CPU time and writes out approximately 1MB per event. Many millions of events are simulated during a typical production run. The steering cards for both the CMKIN and CMSIM programs are produced by a package called IMPALA[4], which then passes the job to a package called BOSS[5] which in turn wraps the job and submits it either to a local batch system or to the Grid. The BOSS wrapping enables the application output to be monitored. Often the simulation has additional stages where background events are added and the electronic response of the detectors is also simulated, however these have not yet been ported to the EU DataGrid testbed.

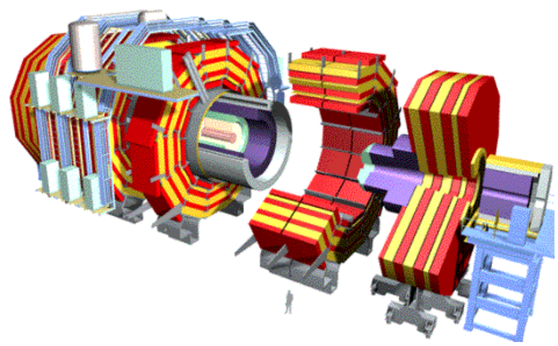

**Figure 1. Schematic diagram of the CMS detector showing its physical size and the quantity of material traversed by the particles.**

## The Middleware and Testbed

The EU DataGrid[6] middleware is built on top of standard Grid tools such as Globus[7] and Condor-G[8]. The testbed consists of Compute Elements (CE) and Storage Elements (SE) installed at sites around Europe. The CE is often the front-end node of a computing cluster consisting of many Worker Nodes (WN) to which the CE can submit jobs using the local batch system. SEs are Grid nodes that provide the storage required by the applications and can range from individual machines with large amounts of local disk space to Mass Storage Facilities. The CEs are configured to know about SEs that are close to them. All CEs and SEs publish information about themselves using MDS. All copies of files stored at the SEs are registered in a Replica Catalogue (RC).

[1] All estimates of CPU time refer to a 1GHz Intel Pentium III.

All user interaction with EU DataGrid is through the User Interface (UI). The UI is typically installed on desktop machines. The user describes the requirements of their job using a Job Description Language (JDL) based on the Condor ClassAds. In the JDL file the user can specify (amongst other attributes) the files to be sent with the job in the Input Sandbox, the name of the executable, the machine requirements (CPU speed, Memory, OS etc), the name of the files to which standard output and error should be written, the input data files required by the executable, the RC that has knowledge of the location of these files and finally which small files they would like to be returned in the Output Sandbox. The job is then submitted to the Resource Broker (RB). The RB is at the centre of the testbed and controls which jobs are run at which locations. Typically there is one RB per Virtual Organisation (VO) although a UI can submit to a list of RBs to avoid there being a single point of failure. The RB compares the requirements of the job as described in the JDL and matches the job to the best resource. In performing this matching the RB will contact the RC to obtain knowledge of the location of all copies of the data files required by the job. The RB will then use the Job Submission Service (JSS) to submit the job to the best matched location. After the job has completed the Output Sandbox is stored on the RB until retrieved by the user. In this way no assumptions are made about the network connectivity of the UI machine.

A user can interrogate the Logging and Bookkeeping Service (LB) to obtain information about the progress of their job through this process. However application level monitoring is performed using the R-GMA[9].

Network monitoring of packet round trip time and throughput is performed between the different sites of the testbed.

## The Demonstration

A web portal has been developed which allows the configuration of IMPALA jobs and their submission via BOSS to the Grid. R-GMA has been interfaced to BOSS allowing the retrieval of the application monitoring information. The portal allows the user to retrieve both this information and information from the LB about the state of their jobs. The steering cards produced by IMPALA are sent in the Input Sandbox along with the executable. The CMKIN part of the production writes its output to an SE close to the CE on which it is running and the location of the output is registered in an RC. The CMSIM job which requires these data are automatically sent to a CE that is close to the CMKIN output. The portal is shown on one of the display screens (see Figure 2). The second screen shows monitoring information about the sites used in this demonstration and the network connectivity between them. Only a few of the testbed sites are used in this demonstration.

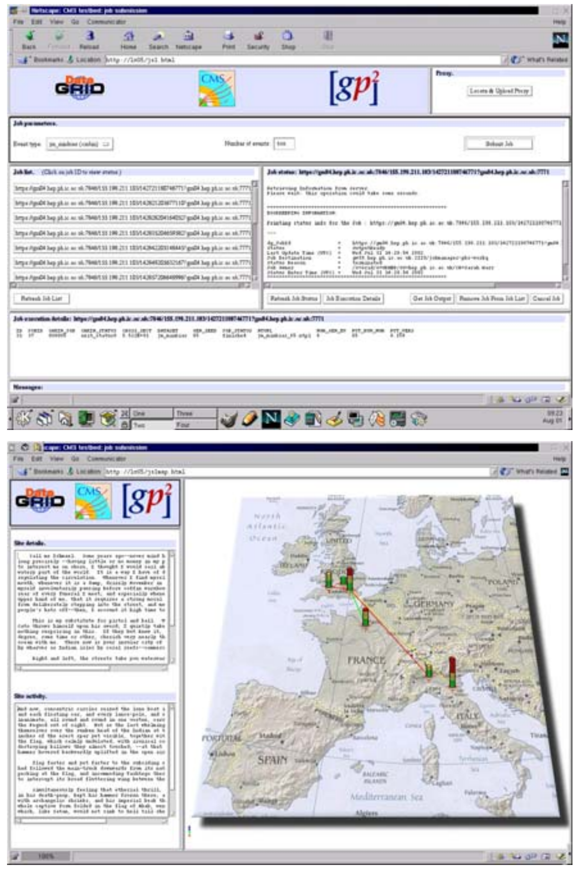

**Figure 2 Screen shots of the IMPALA job submission portal and the monitoring pages**


1. The Large Hadron Collider: http://user.web.cern.ch/user/Index/LHC.html
2. CERN Laboratory: http://www.cern.ch
3. The CMS Experiment: http://cmsinfo.cern.ch/
4. The IMPALA Package: http://computing.fnal.gov/cms/Monitor/cms_production.html
5. The BOSS Package: http://www.bo.infn.it/cms/computing/BOSS/
6. DataGrid: http://eu-datagrid.web.cern.ch/eu-datagrid
7. Globus: http://www.globus.org
8. Condor: http://www.cs.wisc.edu/condor/
9. R-GMA: http://www.gridpp.ac.uk/abstracts/AllHands_RGMA.doc