Monte Carlo Production on the Grid by the H1 Collaboration

E.Bystritskaya¹, A.Fomenko², N.Gogitidze², B.Lobodzinski³

¹ Institute for Theoretical and Experimental Physics, ITEP, Moscow, Russia
² P.N.Lebedev Physical Institute, Russian Academy of Sciences, LPI, Moscow, Russia
³ Deutsches Elektronen-Synchrotron, DESY, Hamburg, Germany

E-mail: bogdan.lobodzinski@desy.de

Abstract. The High Energy Physics Experiment H1 [1] at Hadron-Electron Ring Accelerator (HERA) at DESY [2] is now in the era of high precision analyses based on the final and complete data sample. A natural consequence of this is the huge increase in the requirement for simulated Monte Carlo (MC) events. As a response to this increase, a framework for large scale MC production using the LCG Grid Infrastructure was developed. After 3 years of development the H1 MC Computing Framework has become a platform of high performance, reliability and robustness, operating on the top of the gLite infrastructure. The original framework has been expanded into a tool which can handle 600 million simulated MC events per month and 20,000 simultaneously supported jobs on the LHC Grid, at the same time decreasing operator effort to a minimum. An annual MC event production rate of over 2.5 billion events has been achieved, and the project is integral to the physics analysis performed by H1. Tools have also been developed, which allow modifications both of the H1 detector details, of the different levels of the MC production steps and which permit full monitoring of the jobs on the Grid sites. Based on the experience gained during the successful MC simulation, the H1 MC Framework is described. Also addressed are reasons for failures, deficiencies, bottlenecks and scaling boundaries as observed during this full scale physics analysis endeavor. The found solutions can easily be implemented by other experiments, and not necessarily only those devoted to HEP.

1. Introduction

After the end of the data taking at the HERA collider in June 2007 the H1 Collaboration entered the present period of high precision physics analyses based on the final data sample. These analyses require a huge number of simulated Monte Carlo (MC) events. The local computing infrastructure of the H1 collaboration at that time was by far not enough to give effective support for the rapidly growing demand of MC simulation. Thus there was a direct reason to start an effort for exploring the computing resources on the LHC Grid environment and to use them for the H1 MC production.

In this paper we first describe the architecture of the Monte Carlo production scheme, implemented for the LHC Grid type of resources, both in general terms and for the particular solution chosen for the H1 MC simulation, a solution which was dictated by and adopted to the existing H1 software infrastructure, designed and developed at a time when Grid computing standards were not yet established. Later we discuss error handling, bottlenecks and scaling boundaries which we found. The paper concludes with a summary of the experience gained,
from the point of view of efficiency of large scale calculation on the new paradigm of distributed computing and storage resources.

2. Overview of the Monte Carlo simulation scheme for the H1 experiment

The main steps in the chain of MC production is more or less the same for any high energy physics experiment. It consists of:

(i) MC event generation, resulting in 4-vector representations of outgoing particles.
(ii) Detector simulation, resulting in the electronic response of the detector to the outgoing particles.
(iii) Signal reconstruction, using the simulated detector response to reconstruct the 4-vectors of the involved event particles.

In case of the H1 experiment the steps in the general scheme of MC production is presented in figure 1, where VO abbreviation stands for Virtual Organization and HONE is the VO name of the H1 group on the LHC Grid.

![Figure 1](image)

Figure 1. Full scheme of the H1 MC simulation. Due to reasons explained in the text the preprocessing step (2) has to be performed locally on the User Interface (UI). The other steps, (1), (3) and (4), are performed on the Grid. Dark (blue) arrows denote the flow of jobs from UI to the Grid resources. Light shaded (green) arrows denote remote access to the H1 database (DB), which is realized during the Preprocessing and the Simulation and Reconstruction steps, during which the relevant run conditions of the simulation period are required. Dashed (orange) arrows describe the event data transfer from and to the Grid. The digits close to the data transfer arrows correspond to the MC steps associated with the transfer. The LHC Grid itself is denoted in form of the Google map, with sites available for HONE VO.

The existing development of the H1 reconstruction and simulation software forced us to perform the Preprocessing step, denoted in Fig. 1 as (2), locally on the UI. This is a problem, seen from the point of view of automatic MC simulation and being a relic from the time when the Grid paradigm was not considered at all during H1 software development. The software of
the more recent LHC experiments has been designed with the Grid as part of the architecture. As a result, in the H1 case the Preprocessing step is still performed locally on the UI, where the generator input files are merged and the job scripts are prepared for the next steps of Simulation and Reconstruction.

The first step of the MC production, the MC 4-vector event generation, is a quick process and can be performed alternatively on the Grid or on the local UI. The Grid application of step (1) was developed for the requirement of the the Data Preservation in High Energy Physics [3] project, and this tool is based on the same solution which is used for the simulation and reconstruction described in the next subsection.

The step “Simulation and Reconstruction” is built in the form of a single Grid job which starts the Simulation and Reconstruction process. A detailed description of this step is provided in [4]. The result of the Simulation and Reconstruction are DST (Data Summary Tape) files, which are copied to the final H1 storage tapes at DESY.

The H1 physics analysis software is, since about the year 2000, based on object oriented (OO) techniques and uses Root [5] based files as input. This leads to the last stage of the H1 MC chain, the H1OO post-processing, which produces the analysis input files in Root format. Details of this step is described in [6]. In this part the resulting DST files from the Simulation and Reconstruction stage are converted into Object Data Store (ODS) files, which are 1-1 copies of the DST files, in Root format. On the basis of the ODS files the H1OO algorithms define physics particles, the 4-vectors of which are stored in the µODS output files. Event summary information that is used for tagging of processes is stored in a separate H1 Analysis Tag (HAT) data layer. Thus, as a result of the H1OO post-processing stage two files are created in addition to the DST output files, µODS and HAT. Both files are copied to the final H1 tape storage and are part of the analysis level. The input to the H1OO post-processing, step (4), is the DST files produced by step (3) on the Grid sites.

2.1. The H1 Monte Carlo Framework
Basic details of the H1 Monte Carlo Framework are described in [4]. Here we present the management level of the framework which was previously not described. We stress that this framework solution is based on the gLite infrastructure at present and does not use the “pilot jobs” concept.

Every step of the H1 MC simulation chain can be started independently using input files available on the H1 tape repository. Due to the complicated structure of steering text files required by each step of the MC scheme, a command line interface can only be used by experts. In order to minimize this requirement and to avoid possible mistakes a graphical panel was developed, which greatly simplifies common control and monitoring of all steps in the MC production cycle and which allows management of each step of the MC. An example of the control panel is shown in Fig. 2.

A job exists in one of the following states:
- **new**: job is registered in the MC DB but is not yet submitted to the Grid,
- **running**: job is submitted to the Grid,
- **done**: job is done in the Grid,
- **received**: job output is downloaded from the Grid,
- **failed**: job has failed due to any reason,
- **broken**: job has failed permanently,
- **succeeded**: job has finished correctly,
- **finished**: the job output files are copied to the H1 tape storage.
Figure 2. View of the control panel used for the management of H1 MC production. Requests corresponding to MC Event Generation, Simulation and Reconstruction and H100 Post-processing are distinguished by different prefixes to the MC index. The prefix 100 is used for MC Event Generation and 200 for the H100 requests. In the example, only Simulation and Reconstruction requests are currently being processed.

Only the job state **broken** requires manual intervention. The Control Panel allows the full steering and monitoring of the MC processing, including the most important tasks:

- a detailed check of the job status and log entries related to a given job,
- movement of a given job from one state to another,
- full reset and restart of any given request,
- clean-up procedure: removal of all files related to the MC request on the Grid and the UI,
- selection of the available CEs (Computing Element),
- a check of the output files,
- registration of the request in the MC DB.

Detailed information about the production status and associated plots is provided by the MonaLISA H1 repository as described in [4].

2.2. H1 Monitoring Services

Error handling is one of the most important ingredients of the H1 MC Framework. Error handling was initially based on the official Service Availability Monitoring (SAM), later replaced by a successor based on Nagios [7]. It was soon clear that the SAM system was too unstable and too poorly equipped for the particular checks mechanism required in the H1 monitoring. Therefore a local monitoring service was developed in the H1 MC production architecture. The general scheme of this system is based on the regular submission of short test jobs to each component of the LHC Grid available for the HONE VO. As test jobs we use short versions of real production jobs. To locate services available for HONE VO a service
published by the Berkeley Database Information Index (BDII) is used. The monitoring system includes tests of the CE (Computing Element), Cream-CE (Computing Resource Execution And Management Computing Element), SE (Storage Element) and WMS (Workload Management System) services. As parameters of measurement the monitoring platform checks on the:

- exit status of the test job,
- time which the test job spends in each Grid state,
- RW (Read/Write) availability of the SE and the data transfer rates,
- usage of the WMS services for job submission.

Critical values for each check listed above are defined. If the measured parameter exceeds the critical value an entry is made into the “black list” of services, otherwise the component remains in the “white list” of services. Both lists are created automatically and are used during MC production by the H1 MC Framework.

3. Performance and efficiency of Monte Carlo simulations

The most important feature of a large scale production platform is the performance of the system. In the present case the performance is given by the amount of MC events that can be simulated on the LHC Grid within a given period of time. The productivity of the H1 Framework is presented in Fig. 3, where the yearly production rate is shown. The availability of the Grid can be seen as a huge increase of the simulation rate starting in 2006 when the H1 Collaboration began actively using the LHC Grid.

![Figure 3](image.png)

**Figure 3.** MC production by the H1 Collaboration per year on all available resources.

The second important parameter for assessing the simulation framework is the efficiency of mass production. The efficiency is defined as the ratio of successful tasks over all tasks within a given period of time. The tasks can be divided into those related to the data transfer or to the job execution. The efficiency is highly influenced by the environment into which the jobs are submitted. The measured efficiency of the data transfer, being the most basic estimate, is presented in left panel of Fig. 4.

All input data used by the H1 MC jobs are replicated on the SE’s available for HONE VO sites. Therefore, the inefficiency of the data transfer is related mainly to the data transport from a WN (Work Node) to the local SE or from the local SE to a WN. Another factor of the inefficiency is the further copying of the resulting files on the local SE to the central SE for H1 MC production located at DESY-Zeuthen.

In general two main problems are observed, which influence the efficiency of the data transfer:
Figure 4. Left: The monthly efficiency of data transfer to and from the Grid sites, for the period 2007-2012. Right: The monthly efficiency of job execution of H1 jobs submitted the Grid for the same period.

(i) very low data transfer rate ($\sim$ kb/s),
(ii) the copy command *lcg-cp* gets stuck due to unknown reasons.

Both factors activate the timeout mechanism built into the MC job. However, as can be seen from the plot, the efficiency of the data transfer is growing; this is connected to several improvement cycles of the Grid environment during the time period in question.

The second kind of inefficiency is related to the job execution. Although it is clear that the job efficiency itself depends on the data transfer efficiency, there are in addition other, very frequent factors which deteriorate the execution efficiency. They can be listed as

- lack of free space in the home directory of the Working Node,
- job abortion on a CE or Cream-CE due to wrong authorization of a site used for job submission for HONE VO,
- disturbances in access to the LCG File Catalog (LFC).

The job execution efficiency is plotted in the right part of Fig. 4. Despite these limiting factors we observe an increasing efficiency with time.

3.1. Bottlenecks, scaling boundaries, limitations

In this section we summarize the most important of the observed factors, which limit the overall performance.

- Due to the strong dependency between the Grid Middleware and the H1 MC Framework every migration to a new release of gLite or to a new Grid Middleware requires intense participation in the modifications and testing of all changes which have to be introduced into the H1 MC Framework.
- Continuous monitoring of the availability of all the sites for HONE VO and changes in the environment of these sites becomes a very critical issue due to the pattern of H1 large scale MC production having peak fluctuations rather than being constant in time. Requesters expect MC output files to be ready as soon as possible. Without detailed knowledge of the Grid environment at any time, provided by the monitoring system, fast processing of the MC requests will not be possible.
• Low data transfer rate from the local UI storage to the H1 tape storage can become a serious problem, in particular due to the peaked structure of the H1 MC production profile, where the transfer of large output data volume is often required on a short timescale.

• In the case of simultaneous support of a large number of jobs, the need for manual intervention becomes more frequent. Usually the reasons of such job failures remain unknown. The operator needs to find the shortest way to get the output into the tape storage. Needless to say, such manual intervention introduces unwanted delay into the MC processing.

• The transfer rate of data between Tier-2 sites can be rather slow. It varies between 2 Mb/s and 14 Mb/s. Lower rates than 2 Mb/s causes failure of a job to complete.

4. Conclusions
Observations based on experience gathered during 6 years of usage of the LHC Grid environment for H1 MC production are presented.

• The H1 MC Framework is created in the form of a wrapper on top of the existing Grid Middleware. Continuous monitoring of the Grid Middleware is required, as well as intensive work during migration to new releases of the Grid Middleware. In our opinion, for an experiment smaller than the LHC collaborations it is much easier to use the existing core infrastructure than to write a new workload management system, in particular since it is very difficult to predict the future development of the Grid environment.

• Error handling is an important part of an effective large scale production. In order to do proper error handling a fully controllable monitoring system was developed. It is of critical importance to use the same methods for the continuous checks of the Grid components which are also used for the mass production. Another important aspect of the monitoring system is the possibility to add new specific tests as soon as the demand arises.

• The quality of information published by the BDII system varies from site to site. This is the direct reason that a monitoring system is needed at all.

• The development of the H1 MC Framework continues, based on the work with specific troubles which occur in the day-to-day real production.

• The H1 MC jobs do not need large input data. A memory working area less than 10 GB per job is sufficient to execute even the most size demanding part of the H1 MC production chain, namely Simulation and Reconstruction. Thus, it is preferred to use direct copying of the necessary input files from the local SE to a WN instead of using NFS mounted disks. This is a more general solution, which allows our jobs to be of a more uniform structure. From this point of view it is enough to simply base the Data Management System on the LFC solution.

After several years of continuous improvement, the H1 MC framework is now a highly efficient, reliable and robust platform. With only few modifications, the existing framework could be relatively easily adopted by other experiments, where a large production on the Grid is necessary, either for MC simulation or for processing of real data.

References
[1] http://h1.desy.de/
[2] Deutsches Elektronen-Synchrotron: http://www.desy.de/
[3] http://www.dphep.org/
[4] B.Lobodzinski et al. 2010, J. Phys.: Conf. Ser. 219 (2011) 072040
[5] http://root.cern.ch/
[6] M.Steder 2011, J. Phys.: Conf. Ser. 331 032051
[7] http://lcg.web.cern.ch/lcg/monitor.htm; https://twiki.cern.ch/twiki/bin/view/LCG/GridServiceMonitoringInfo