J Grid Computing (2008) 6:3–13
DOI 10.1007/s10723-007-9087-3

High-Energy Physics on the Grid: the ATLAS and CMS Experience

Julia Andreeva · Simone Campana · Federica Fanzago · Juha Herrala

Received: 27 August 2006 / Accepted: 7 June 2007 / Published online: 6 September 2007

Abstract In this paper we present the experience of the ATLAS and CMS High-Energy Physics (HEP) experiments at the Large Hadron Collider (LHC) with the LCG/EGEE Grid infrastructure. The activity developed around the following two main lines: large-scale physics and detector simulations and end-user analysis. The LCG/EGEE Grid infrastructure offers a large amount of computing and storage resources and is growing very rapidly. It provides the natural environment for large-scale physics and detector simulations. Also, the analysis of these detector simulation data (and in the near future of the reconstructed data from physics collisions) requires efficient end-users access to Grid resources. In this paper, the main findings and lessons learned in terms of performance, robustness and scalability of the whole system are discussed in detail.

J. Andreeva · S. Campana · J. Herrala
CERN, European Organization for Nuclear Research,
1211 Geneva 23, Switzerland

S. Campana (✉)
Istituto Nazionale di Fisica Nucleare (INFN),
Sezione CNAF,
Bologna, Italy
e-mail: simone.campana@cern.ch

F. Fanzago
Istituto Nazionale di Fisica Nucleare (INFN),
Sezione di Padova,
Padova, Italy

Keywords Grid · HEP · LCG · EGEE · ATLAS · CMS

1 Introduction

The LHC experiments ALICE, ATLAS, CMS and LHCb are preparing for data acquisition starting in 2007 [1, 2]. This requires the use of major computing resources at many levels, particularly for the validation of the computing and data model, the testing of the complete software suite as well as tests of the analysis framework.

The complexity of the LHC detectors can be appreciated considering, as an example, the ATLAS experiment' s main parameters: 7,000 t of weight for 10,000 m^3 of volume, 150 millions of electronic channels to detect proton–proton collisions happening at a rate of 40 MHz. After on-line filtering, the detector alone is expected to deliver data at over 100 MB/s. More information can be found in [3].

The simulation and study of the detectors' response share most of the main features of the computational challenge which will be carried on the experimental data: continuous use of large distributed computing resources providing 50 MSI2k[1], large data transfers in the range 100 to 1,000 MB/s

[1] A modern Intel Pentium IV processor with a 2.8 GHz CPU corresponds to about 1 kSI2k; the current ATLAS share of CPU resources in the CERN batch facility corresponds to about 500 kSI2k.

over many months per year and distributed storage in the range of 10 PB/year. More details can be found in [4–7]. Because of these requirements, coupled with the fact that the LHC collaborations are large and geographically distributed efforts (ATLAS for example counts 1,600 physicists over more than 100 countries), the LHC experiments decided to adopt the innovative solution of Grid computing. This consists of a high level of decentralization and sharing of computing resources. Particularly for LHC computing, different facilities are organized in a hierarchical structure, with distinct roles at different levels.

The paper focuses on the ATLAS and CMS experience on the Grid and provides a description of the main activities of the LHC collaborations on the LCG/EGEE Grid infrastructure. We then describe the ATLAS simulation activities, the CMS distributed analysis and the common monitoring framework.

2 LHC Experiment Computing Activities

Currently the dominant activities are detector studies on simulated data, mainly divided in three steps:

1. Event generation: the final state configurations of particle collisions in the detectors are generated using programs relying on theoretical calculations, phenomenological models and experimental inputs;
2. Interaction between the collisions product and the detector: the interaction of the generated particles inside the detector is simulated (taking into account the geometry and detector materials);
3. Digitization: the detector response is simulated and it is written in a format equivalent to real output of the detector.

At nominal conditions, each event consists of several interactions, due to the high collision rate. To reproduce this effect, simulated interactions should be "piled-up" i.e. superimposed before digitization. The results are simulated events of the same complexity of the real events (~1,000 particle tracks).

The simulation process and the reconstruction of physical objects (such as tracks and electromagnetic clusters), both for simulated and real data, are normally referred to as "production" activities. Such activities, by their nature, are centrally-controlled and operated in a semi-automatic way by a small team of experts. The results are large datasets of events, so called "reconstructed events", to be used for calibration, detector studies and finally physics analysis.

Production and analysis are fundamentally different: while production is centrally controlled, data analysis performed by hundreds of independent scientists. This has implications in computational resource usage, data access patterns and resource access policies. In addition, the analysis is an iterative process limited by the latency of the system in delivering results, while in a large production the main point is to maximize the throughput and hence the number of simulated events in the unit of time. A typical analysis task consists of an application which is based on the experiment analysis framework and customized by each physicist, to process a given dataset. The ATLAS experiment expects to have more than 1,000 active users analyzing datasets consisting of several thousand files and up to a few terabytes of data. Since each recorded event is independent from the other, this processing can be trivially parallelized, splitting the user request into a number of independent units of computation, called "jobs", using the same executable. Every job would then be processing a given portion of the input dataset. The challenge consists not only in harnessing the power of this complex system but also providing an interface for the end user as simple as possible.

2.1 The LCG/EGEE Production Grid

The LHC Computing Project (LCG) [8] has the goal to provide a data storage and analysis infrastructure for the LHC experiments. The EGEE (Enabling Grid for E-sciencE) [9] project, started in April 2004, leads a worldwide effort to re-engineer existing Grid middleware (to ensure its robustness), and deploy it in a production infrastructure for science. The LCG/EGEE production Grid currently counts more than 200 sites spread around the globe, providing more than 20,000 CPUs and several PB of storage.

In the LCG model, the different computing facilities are hierarchically organized. The Tier-0 is hosted CERN, offers resources for the first event reconstruction (raw data collected by the experiments will be immediately processed, at the Tier-0, to obtain physical objects) and mass storage capabilities. Major computing centers (Tier-1s), geographically distributed in different regions, also offer mass storage capabilities. Moreover, they provide resources for data reprocessing (further reconstruction activities with better understanding of biases introduced by the experimental apparatus). Event simulation and most of the data analysis will take place at the Tier-2s. The experiments at the LHC have been relying on this infrastructure for several years already for large-scale simulations of physics processes and detector responses.

The LCG/EGEE middleware stack integrates contributions from various sources such as the gLite project [10], Virtual Data Toolkit [11], the European DataGrid Project [11], the DataTAG Project [11] and the LCG project. The LCG/EGEE middleware consists of two major components: a Workload Management System (WMS) and a Data Management System (DMS). The WMS handles job submission, job dispatching, and output retrieval. The DMS allows file replication between different storage locations and upload/download of files to/from the Grid. Grid resources and their description are published in a hierarchical information system. Various monitoring tools can be used to access status information of Workload Management and Data Management services. All relevant operations between parties are mutually authenticated, while the resources authorization service enforces the concepts of groups and roles inside the Virtual Organization. A more detailed description of the LCG Middleware can be found in [12].

3 The ATLAS Simulated Data Production Activity

The ATLAS collaboration started activities on Grid infrastructures in 2002 and has been relying completely on this approach since 2004. The ATLAS production system is based on a central database holding information about job states (available for submission, scheduled in a resource, running, being aborted and many others). In particular, the production system manages jobs definitions which are then dispatched to different processing back-ends. ATLAS uses resources distributed in three different Grid infrastructures: LCG/EGEE, OSG [13] and NDGF [14]. Since 2005, LCG/EGEE has delivered more than 60% of the total amount of CPU for the ATLAS official production activity; therefore, we will concentrate on this part of the infrastructure in the rest of this paper. Details on the interoperability are reported in [13].

The first fully distributed (on all three Grid backends) ATLAS Data Challenge[2] dates back to 2002 (DC2) and has been followed by a large-scale production, which took place before the Rome Physics workshop in June 2003. The second one consisted of a total of 380,000 jobs. This included physics event generation, simulation of the detector response and the electronic read-out, reconstruction of physics objects, both in ideal condition and with signal pile-up. About 45 TB of data, organized in 1.4 million files, have been stored in LCG Storage Elements and registered in the EDG Replica Location Service central catalogue. The Storage elements consisted of disk-only and tape-based Mass Storage Systems. During DC2, a total of 91,500 jobs ran on the LCG/EGEE Grid and no event reconstruction was performed. A more detailed description of DC2 and Rome workshop production activities can be found in [15]. After September 2005, ATLAS has been running continuous simulation on the LCG/EGEE infrastructure. These activities produced a considerable knowledge of the robustness, performance and scalability of the system.

3.1 The Data Management System

During production activities, ATLAS encountered most problems with the Data Management system. The largest amount of job failures (about 27% of submitted jobs) happened in uploading and downloading files (input and output) between Worker Nodes and Storage Elements[3]. File access at runtime consists of determining the physical location of input files via a File Catalogue and copying them locally to the Worker Node for POSIX access.

So far, ATLAS has been relying on a single central catalogue instance (deployed at CERN). At the time of DC2 and Rome Production this consisted in the EDG-Replica Location Service which has been showing inefficiencies and scalability issues at many levels. In particular, the average time for a file look-up of about 2 s under normal conditions was considered inadequate and could degrade considerably

[2] A Data Challenge consists of the full simulation and reprocessing of data coming from the detector, carried out with the same software and computing infrastructure expected to be employed during data taking.

[3] Nearly 70% during Rome production, 30–40% in more recent production activities.

under load, even during a relatively low-rate access scenario such as event simulation[4]. As a consequence, ATLAS suffered several occasions of catalogue downtime, lasting up to several days and causing massive job failures and waste of CPU resources. A new catalogue implementation, the LCG File Catalogue, has since been developed and deployed to overcome such problems. Used in production by ATLAS since October 2005, the LFC has shown considerable robustness (less than 1% of total failures due to catalogue unavailability), even under stress-exercises such as Service Challenge 3 (SC3)[5].

The ATLAS system evolved toward a distributed setup, where files are registered in local catalogues (LFCs at Tier-0 and every Tier-1) and organized in datasets defined and located in a central database instance. This setup, tested already during SC3 should reduce the single point of failure represented by a central catalogue and contribute to improve the overall robustness of the system. In addition, file movement has shown to be problematic as well. Storage Elements (consisting of simple GridFTP server controlling access to a back-end disk storage) can be easily overloaded in case of multiple simultaneous requests, resulting in excessive latencies for file transfers (up to several minutes just to start copying the first byte). In the ATLAS production scenario, where multiple (order of one hundred) jobs might need to access the same file, throttling of file access is not straightforward and long distance transfers between locations quite far apart (different countries or even continents) are not uncommon. The situation improved with the development and deployment of new storage elements back-ends such as CASTOR [16], dCache [17] and the LCG Disk Pool Manager [18], but failures due to Storage Elements downtimes still represent the largest fraction (16% of submitted jobs). Additionally, accessing data on Mass Storage System via the Storage Resource Manager [19] interface is still only partially supported[6]. To overcome some of these problems, ATLAS decided to move towards a more organized production model, where jobs requiring the same input data are forced to run in sites "close" to the data (ideally, in the same LAN, more realistically, in the same country) and frequently accessed files are guaranteed to have at least one permanent copy on disk. This has been achieved integrating the production system with the ATLAS Distributed Data Management service [20], which ensures reliable dataset replication based on a subscription system[7].

3.2 The Workload Management System

The Workload Management relied on the LCG Resource Broker for the entire duration of the DC2 and part of the Rome Production. The main limitation observed is the insufficient submission and job-dispatching speed. The average time for job submission during DC2 and Rome Production has been observed to range between 6 and 13 s, depending on the load on the system. In case of heavy load, a single job submission could easily take more than 60 s to succeed. The new gLite Workload Management System, instrumented with bulk submission capabilities (submission of many jobs in a single operation) and multithreaded matchmaking of jobs to suitable resources (many jobs internally handled in parallel), has been shown to perform at a much higher rate and is being prepared to enter production. Tests on the LCG infrastructure measured a submission rate of about 2 Hz, while a bunch of 1,000 jobs could be delivered to the selected computing farms in less than 2,000 s, sufficient in this phase of the experiment activity.

3.3 The Information System

The Information System currently deployed on EGEE can be considered a fairly robust component (negligible contribution to the overall failure rate). During DC2, instead, the Information System was responsible for 40% of job failures and unavailability/degradation of other services. Nevertheless, excessive load of some services (especially Computing Elements)

[4] With 10,000 jobs per day and roughly 10 catalogue lookups per job, one expects 100,000 lookups per day.

[5] The Service Challenge exercise aims to stress different Grid services and provides an estimate of their readiness status by the start of data taking of the detectors.

[6] Some methods to facilitate data access in MSS, in principle present in the definition of the interface, are currently not implemented in every Storage Element type mentioned above. One example is the possibility to specify a minimum time of persistence of data on disk.

[7] When a dataset is subscribed to a particular storage location, the Distributed Data Management service interacts with the underlying Grid middleware to enforce such subscription. This includes, among others operations, catalogue lookups, scheduling of file transfers, validation of such transfers.

sometime can result in failures publishing the information or propagating them into the Information System tree. Many middleware clients are not robust against downtimes of the Information System and therefore ATLAS implemented a retry logic for several client tools. For quasi-static information concerning very critical services, ATLAS decided not to rely on the Information System but store them in static and ATLAS-specific local configuration files.

3.4 Monitoring

A global job monitoring has been developed gathering information from the ATLAS production database. Figure 1 shows the daily number of production jobs run successfully by ATLAS on different Grids (including EGEE) in a 2 months period in year 2006. The execution time of different job types can vary considerably, but the larger contribution in this plot comes from simulation jobs, which usually run for approximately one day of wall-clock time (elapsed time). In the last few months of continuous production, ATLAS ran successfully an average of about 2,500 jobs per day on the EGEE infrastructure, reaching a peak of 5,000 simultaneous jobs.

Figures 2 and 3 show respectively the number of ATLAS jobs and the amount of wall-clock time spent by ATLAS on the EGEE Grid on a daily basis. In particular, successful jobs and failed jobs are shown separately in green and red, respectively, to estimate the job efficiency. From Fig. 2 one can observe a quite high job failure rate, in average about 50%, reaching 70–80% in rare occasions. Most of those failures however impact only marginally on the production activity: for the large majority, they imply no waste of computing resources since jobs fail right after submission, usually because of a communication problem between the submission framework and computing services at the sites. In fact, despite some major problems (usually due to a failure of a central service) Fig. 3 shows that an average of more than 80% of the wall-clock time is spent for jobs finishing successfully.

4 The CMS Analysis Activity

In this section we review and discuss analysis activities of the CMS collaboration. CMS analysis jobs are created, submitted and monitored via CRAB

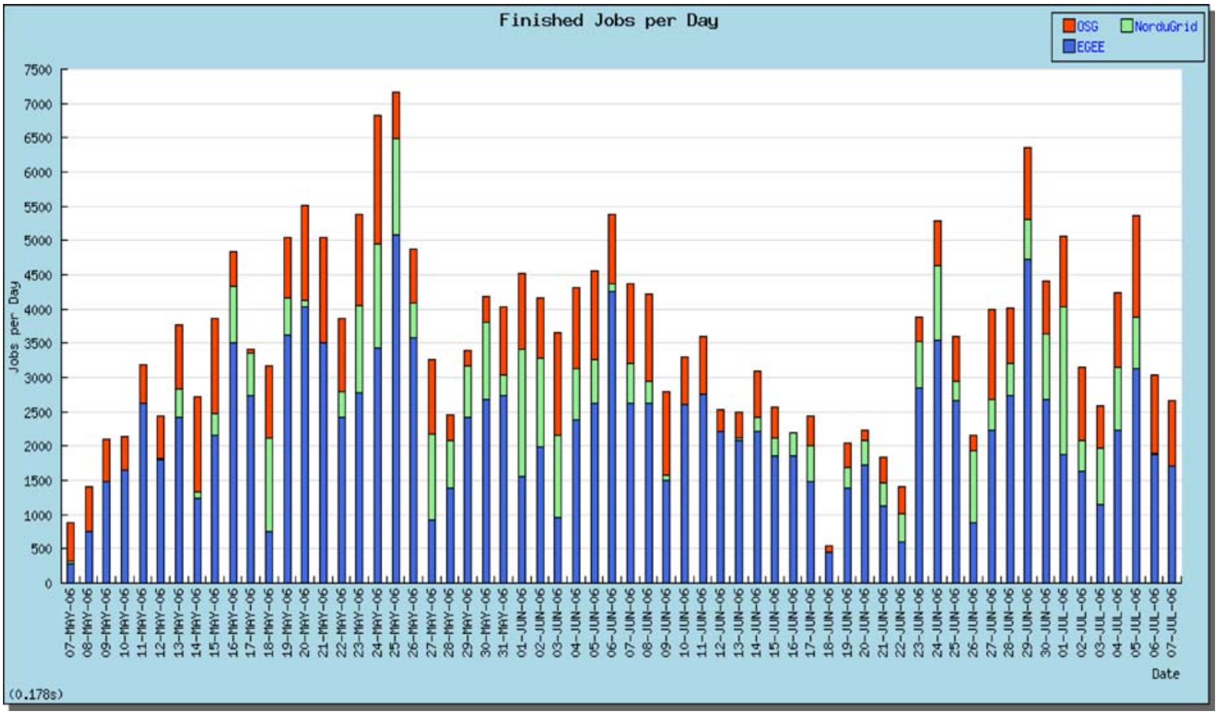


Fig. 1 Number of successful ATLAS production jobs run daily in various Grid infrastructures

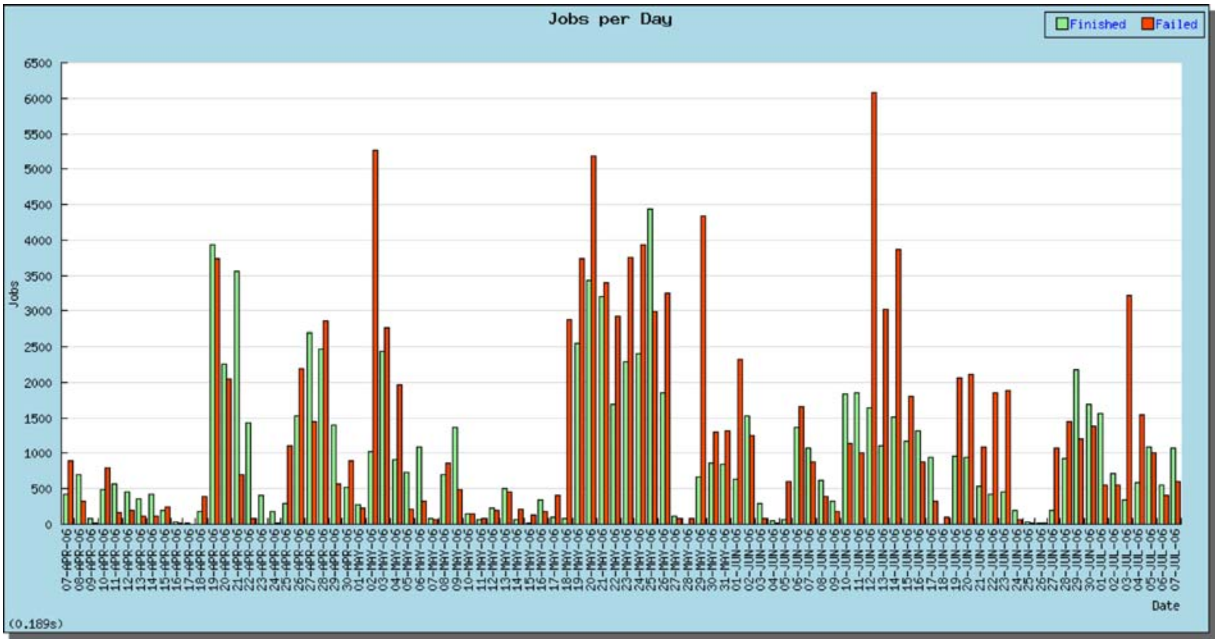


Fig. 2 Number of jobs submitted daily by ATLAS (successful and unsuccessful) on LCG/EGEE

(CMS Remote Analysis Builder) [21], the official analysis tool. Other systems have been used in the prototype stage (e.g. ASAP [22] in CMS) or are in use in the other collaborations (like Ganga for ATLAS and LHCb [23] and the ALICE system [24]).

The two main components of the analysis framework are the data location system and the CRAB job submission tool. The data location system is composed of a central database located at CERN containing information about the available data and several

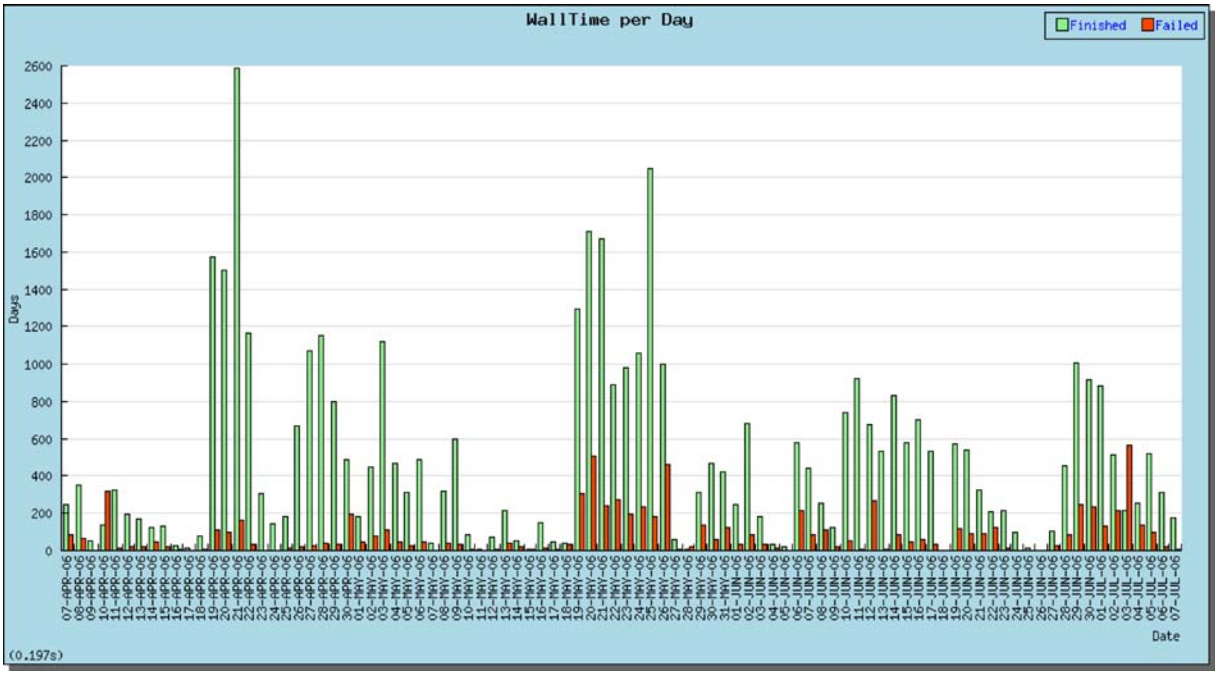


Fig. 3 WallClockTime spent daily by ATLAS jobs (successful and unsuccessful) on LCG/EGEE

local databases at each site. The local databases store site specific information to locate the data, for example, which local Storage Element holds the data at which physical address and which protocol can be used to access the data.

The CRAB system has been designed to simplify the operations of the end user creating and submitting analysis jobs to the Grid infrastructure (Fig. 4). Users develop their analysis code in an interactive environment and decide which data to analyze. CRAB allows them to run their application on the data available at remote sites in the same manner as on local environment (where the application is frequently developed and tuned). In order to prepare the analysis jobs for the Grid, the user has to provide:

- data parameters in order to select a given dataset, total number of events to be accessed and number of events for each job;
- analysis executable (typically user specific) and corresponding configuration parameters;
- output file names and their storage location

Data discovery, resource matching, job creation and submission, status monitoring and output retrieval are fully handled by CRAB. In order to discover data location on the Grid, CRAB communicates with the CMS-specific data location system and translates this information to directives for the Grid Workload Management System for resource selection. CRAB wraps the user analysis executable, which will be run on remote resources, with relevant information, including the CMS environment setup. CRAB splits one analysis task into a number of jobs according to user provided information and each job executes the same code on different sections of the data. The user code is submitted to the remote resource via the input sandbox, together with the job. Active jobs are monitored querying the Grid Logging and Bookkeeping System. The output management of the job is either handled through the output sandbox or the job wrapper may copy the output to a storage location if required.

4.1 Experience Running CRAB

The CRAB job submission tool was intensively used by CMS physicists to analyze data for the preparation of the Physics Technical Design Report [25] and during the last phase of the LCG Service Challenge 3. CRAB has been used since 2005 and the internal job monitoring statistics show that about 50,000 jobs have been submitted during the first months after the initial deployment (August–September, 2005). During this period, the weekly rate increased from 6,000 to 15,000 jobs as the tool was attracting more users. About 210 different datasets were accessed at least once, while single datasets have been accessed up to 15,000 times. Jobs were submitted from 25 User Interfaces and data was stored into 40 different remote sites. The overall job success rate has been about 75%, where success is defined as jobs that arrive at remote sites and produce outputs, while the remaining 25% of job aborts is due to site setup problem or Grid services failure. After the development of CRAB and its large scale employment, distributed analysis in CMS became the everyday practice of the physics community. In the past, in fact, only a limited number of experienced users submitted jobs to a small subset of Grid sites, while most users were still relying on local batch farms. During June–July 2006, CMS analysis jobs were submitted by more than 70 users to 85 different sites.

The LCG Service Challenge 3 involved all LHC experiments during the period of July–November, 2005. The main objective was a realistic test of data transfer and data access use cases. The challenge was

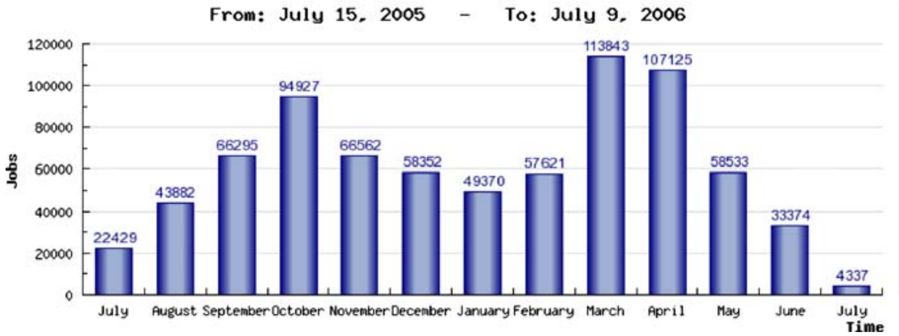


Fig. 4 CRAB activity as number of analysis jobs submitted per month

divided in two phases. The first phase was performed in July and CMS tested data transfers between Tier-0, Tier-1 and Tier-2 centers. The second phase started in September, and it included also analysis tests. The analysis jobs, prepared, submitted and handled by CRAB aimed to process the simulated data transferred to Tier-2s in the context of the throughput exercise (on the files published on the CMS specific central and local catalogues). The challenge involved a significant number of sites: 7 Tier-1 and 13 Tier-2. Generally, the Service Challenge (especially the "throughput" phase) demonstrated that many services were not sufficiently tested before the start of the exercise and that the distributed computing system still needs support and attention. However, specifically for the analysis phase with CRAB, the success rate of the Grid infrastructure was more than 90%. The most common failures resulted in data preparation and in the CMS application itself.

5 Job Monitoring using the Experiment Dashboard

As in the case of ATLAS and CMS, the LHC experiments are depending on the Grid infrastructure for their core activities. Since an overall picture of all the experiment activities on such a world-wide computing and storage infrastructure is difficult to achieve, advanced monitoring systems are vital. The aim of the Experiment Dashboard project [26] is to provide a single entry point to the monitoring data collected from various sources of the Grid-based distributed computing systems of the LHC experiments.

Currently, the Experiment Dashboard serves the ATLAS and CMS experiments and the main development has focused around job monitoring. The objective of the job monitoring service is to provide a complete view of the experiments activities (such as production and analysis) in terms of jobs submitted to the Grid infrastructure and, on the other hand, to pin down error conditions and bottlenecks. This is achieved by storing and displaying various quantitative and qualitative characteristics of the Grid usage and by combining Grid-related information with experiment-specific information in the Dashboard database.

The Experiment Dashboard relies on the Oracle database infrastructure at CERN. The data collectors-gather both Grid-related information and application-specific information in order to compile a comprehensive picture of the overall job success rate. The Grid-related information is obtained from the Information System (active sites and queues) and from the Logging and Bookkeeping system of the Resource Brokers (job status changes and destination queues for individual jobs). The application-specific information is gathered throughout the job lifetime – submission, runtime and output retrieval – via the MonAlisa monitoring system [27] developed at Caltech University. MonAlisa is widely used by the LHC experiments for monitoring of the local sites, network traffic and for the application level monitoring.

A web interface on top of the Dashboard database provides interactive access to the monitoring information. The database schema has been designed to store the main monitoring indicators, such as resource usage and sharing, Grid behavior, application robustness, and data distribution. All this information can be aggregated and presented per user, per site, per application or per dataset. This information allows to present relevant quantities of the current state of the experiment activities on the Grid, for example, how many jobs are running, pending and accomplished, and which fraction of the accomplished jobs were successful. Main quantities in resource utilization are CPU, memory consumption and input–output rates. Distribution of the quantities over time is supported as well, which allows, for example, examining Grid behavior in terms of success rate or reasons of failures as a function of time.

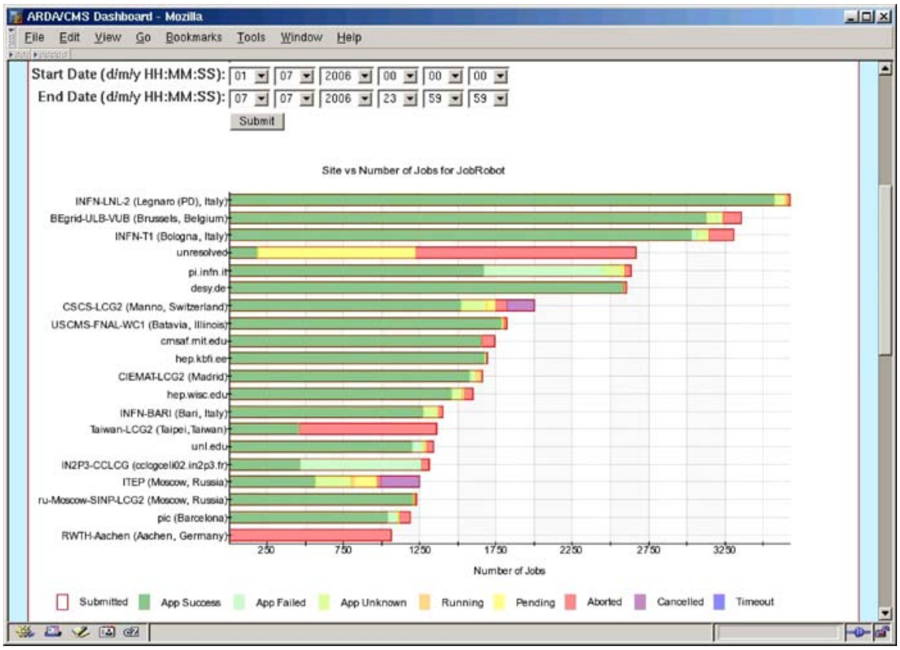
However, the Experiment Dashboard should not only show the state of the activities on the Grid, but also assist experiments in improving their overall distributed computing systems. Thus, one of the main requirements for job monitoring is the ability to indicate job execution problems of different origin. Possible reasons of the problems may be related directly to the Grid infrastructure or to the activities of the experiments. The problems may also originate from a mixture of Grid-related and experiment-specific aspects, which often makes tracking of such problems difficult and time-consuming. Some of the so-called Grid-related problems are frequently connected to incorrect site configuration. Frequently, application problems are connected to data publishing errors and run-time application errors. Tracking of various problems with the Experiment Dashboard is straightforward due to detailed information provided by the user interface for any single job.

Detection of sites configuration problems is an important goal of the Experiment Dashboard. The system provides three interfaces to follow job processing at the given site: interactive, historical and site reliability view. Figure 5 shows the example of the interactive page where CMS jobs are sorted by site. Red color corresponds to aborted jobs, light-green corresponds to the jobs which were successfully accomlished from the Grid point of view but failed to run the application properly due to other problems as user code or VO specific services. Problems related to the data access or corrupted distribution of the experiment software at a given site will cause the application failures reported to the dashboard from the job wrapper and therefore can be detected through the Dashboard interactive interface. Exit status reason for the aborted jobs is also shown at the interface page. However, in many cases exit status reason which is returned by the Logging and Bookkeeping System does not explain the reason of the Grid failure. The Dashboard site reliability view provides very detailed record of the status changes for the accomplished jobs and allows a better understanding of the problems at the sites.

Problems related to a certain Grid service can be also detected via the Dashboard interactive interface. Figure 6 shows an example of the efficiency of different Resource Brokers. All the jobs shown at Fig. 6 are submitted to a single site but via the variety of Resource Brokers. Therefore the job failures are related to the Resource Broker rather than the problems at the site of destination. As one can see at Fig 6, the Resource Broker showing higher failure can be pinned down.

CMS jobs are instrumented to report to the MonAlisa server input/output rates between the worker nodes and data storage system. Due to it the CMS Dashboard is providing an estimation of the aggregated input/output rates between storage system and active worker nodes. The Dashboard system also helps to keep record of resource sharing between physics activities within an experiment, such as production and analysis, between different analysis groups, and between individual users.

Data replication and publishing is one of the most important factors for the success of user analysis on the Grid. The Experiment Dashboard provides monitoring of the various data management tasks. Atlas Data Management monitoring includes site service monitoring (by collecting callbacks sent by the agents handling the dataset transfer), dataset location information and generation of the transfer statistics. The CMS Dashboard provides monitoring of the Transfer Load Tests using data collected in the Phedex [28]

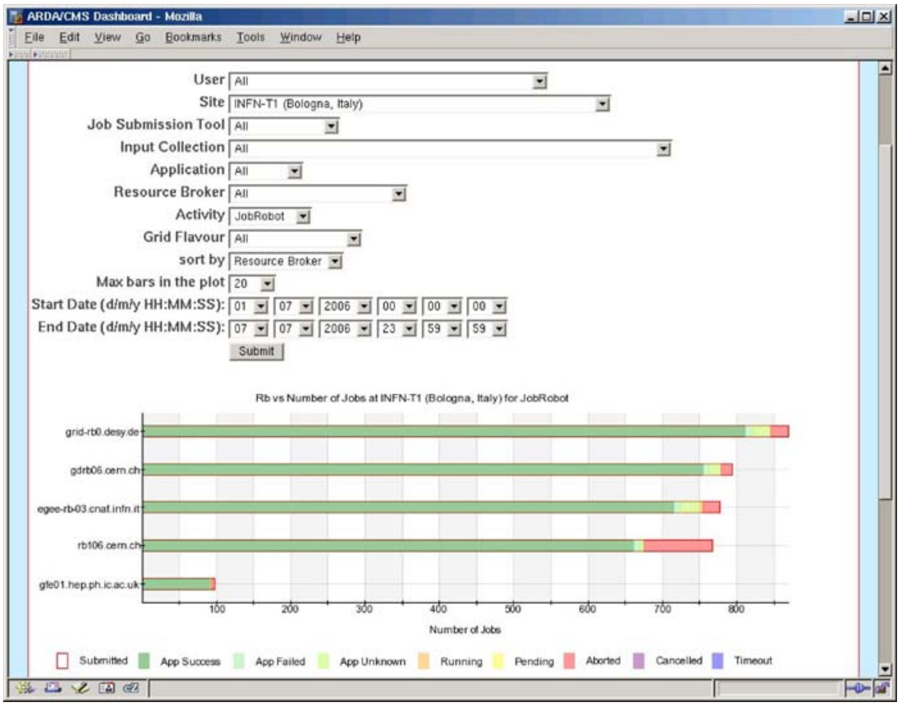


Start Date (d/m/y HH:MM:SS): 01 07 2006 00 00 00

End Date (d/m/y HH:MM:SS): 07 07 2006 23 59 59

ARDA/CMS Dashboard - Mozilla
File Edit View Go Bookmarks Tools Window Help
Submit
Site vs Number of Jobs for JobRobot
INFN-LNL-2 (Legnaro (PD), Italy)
BEgrid-ULB-VUB (Brussels, Belgium)
INFN-T1 (Bologna, Italy)
unresolved
pi.infn.it
desy.de
CSCS-LCG2 (Manno, Switzerland)
USCMS-FNAL-WC1 (Batavia, Illinois)
cmsaf.mit.edu
hep.kbfi.ee
CIEMAT-LCG2 (Madrid)
hep.wisc.edu
INFN-BARI (Bari, Italy)
Taiwan-LCG2 (Taipei,Taiwan)
unl.edu
IN2P3-CCLCG (cclcgceli02.in2p3.fr)
ITEP (Moscow, Russia)
ru-Moscow-SINP-LCG2 (Moscow, Russia)
pic (Barcelona)
RWTH-Aachen (Aachen, Germany)
250
750
1250
1750
2250
2750
3250
Number of Jobs
Submitted
App Success
App Failed
App Unknown
Running
Pending
Aborted
Cancelled
Timeout

Fig. 5 Activity of the Service Challenge 4 during the first seven days of July 2006 on the CMS Dashboard. The magnitude of the activity is presented as number of submitted jobs sorted for 20 most active sites. During the challenge, jobs are submitted by an automated tool based on CRAB. *Green color* on the plot indicates success, *red color* indicates failure, and *other colors* indicate jobs still active on the system



User All
Site INFN-T1 (Bologna, Italy)
Job Submission Tool All
Input Collection All
Application All
Resource Broker All
Activity JobRobot
Grid Flavour All
sort by Resource Broker
Max bars in the plot 20
Start Date (d/m/y HH:MM:SS): 01 07 2006 00 00 00
End Date (d/m/y HH:MM:SS): 07 07 2006 23 59 59
Submit

Fig. 6 This figure presents SC4 jobs submitted to the INFN Tier-1 site sorted by the Resource Broker, which was used in the submission phase. One can immediately notice of the *red color* that the Resource Broker at rb106.cern.ch has had more problems than the other ones. On the *top* of the figure, the complete set of the attributes of the interactive Dashboard web interface is visible

(Physics Experiment Data Export – data placement and file transfer system for CMS experiment) database. One of the final goals of the data management monitoring is to define rules to discover inefficiencies in data distribution and resource utilization and problems in data transfer and publishing.

In the future, the Experiment Dashboard will have a more active role, not only collecting data and displaying information, but also periodically analyzing the information and sending alarms to relevant persons in case of evident problems.

6 Conclusions

The physics program at LHC is a challenge also for the computing systems which will require an enormous amount of computing and storage resources to be delivered by Grid infrastructures. Many improvements have been achieved in terms of integration of experiments computing systems with the Grid middleware and in setting up production quality Grid systems. In this contribution, two of the main current activities on the LCG/EGEE Grid have been presented: distributed production of simulated events and distributed analysis. Such activities are quite different in nature (data access patterns, scale of the exercise, granularity), however many common points have been identified.

Taking as an example ATLAS and CMS experiments, this paper demonstrated that during 2004–2006 the performance, scalability and robustness of the LCG/EGEE Grid significantly improved becoming the main production infrastructure for the LHC experiments. This is a big achievement compared with the prototype situation prior to 2004 [29]: usage of Grid resources was limited to well defined and focused computing exercises; most of simulated event production was still run on batch farms at individual computing centers and no automatic distributed analysis was in place. The existence of high level tools like CRAB and the Experiment Dashboard hides the complexity of the underlying Grid infrastructure and allows non-experienced users to effectively run analysis on a distributed environment.

In conclusion, several LHC experiments now rely on the LCG/EGEE Grid resources for many activities of different kind. The amount of effort required to run such activities is still considerable: several computing exercises on the Grid infrastructure still require a non negligible amount of expertise and must be carried on in a controlled environment. However, the robustness of the middleware and the overall reliability of the infrastructure are continuously improving.

Acknowledgements The authors would like to thank the LCG/EGEE operation teams (and in particular the CERN IT/GD group) for the fruitful collaboration in setting up and operating the experiments systems. We acknowledge fruitful discussion with the CRAB development team and the ATLAS production teams. Cross-experiment (and cross-discipline) contacts proved to be very useful, in particular in the EGEE NA4 (and in particular in the CERN IT/PSS group). EGEE is a project funded by the European Union under contract INFSO-RI-508833. Probably the most effective contribution to the evolution of analysis and monitor systems has been provided by the users themselves, with their experience and suggestions and we look forward for further collaboration. This work received support also from the Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung), Berlin, Germany.

References

1. Virdee, T.S.: Detectors at LHC. Phys. Rep. **403–404**, 401–434 (2004)
2. Gianotti, F.: Physics at the LHC. Phys. Rep. **403**, 379–399 (2004)
3. ATLAS Collaboration: ATLAS Technical Proposal. CERN/LHCC/94-43 (1994)
4. The ALICE Computing Group: ALICE Technical Design Report of the Computing. ALICE-TDR-012, CERN-LHCC-2005-018, June (2005)
5. The ATLAS Computing Group: ATLAS Computing Technical Design Report. ATLAS-TDR-017, CERN-LHCC-2005-022, June 2005
6. The CMS Computing Group: CMS Computing Technical Design Report. CMS-TDR-007, CERN-LHCC-2005-0223, June (2005)
7. The LHCb Computing Group: LHCb Computing. LHCb-TDR-11, CERN-LHCC-2005-019, June (2005)
8. The LCG Editorial Board: LHC Computing Grid Technical Design Report. LCG-TDR-001, CERN-LHCC-2005-024, June (2005)
9. Jones, B.: An overview of the EGEE project. Peer-to-Peer, Grid, and Service-Orientation in Digital Library Architectures, Lecture Notes in Computer Science, Volume 3664/2005, Springer, (2005)
10. Laure, E. et al.: Programming the Grid with gLite. Comput. Methods Sci. Technol. **12**(1), 33–45 (2006)
11. Laure, E.: The EU DataGrid Setting the Basis for Production Grids. Journal of Grid Computing **2**(4):299–400 (2004)
12. S. Campana, Maarten Litmaath, Andrea Sciaba: "LCG-2 Middleware Overview", CERN-LCG-GDEIS-498079, http://edms.cern.ch/file/498079/0.1/LCG-mw.pdf.
13. Field, L.: Grid deployment experiences: the interoperations activity between OSG and LCG. Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India)
14. Gronager, M.: LCG and ARC middleware interoperability. Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India)
15. Goossens, L.: Production system in ATLAS DC2. Conference on Computing in High Energy and Nuclear Physics (CHEP04), September 2004, Interlaken (Switzerland)
16. Campana, S. et al.: Analysis of the ATLAS Rome production experience on the LHC computing Grid. Proceedings of the First International Conference on e-Science and Grid Computing (e-Science '05), December 2005, Melbourne (Australia)
17. Barring, O. et al.: Storage resource sharing with CASTOR. 21st IEEE Conference on Mass Storage Systems and Technologies, April 2004, Adelphi MD (USA)
18. Perelmutov, T.: Enabling Grid features in dCache. Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India)
19. Burke, S. et al.: gLite3 user guide. CERN-LCG-GDEIS-722398, http://edms.cern.ch/document/722398/1.1
20. Shoshani, A.: Storage resource management. GGF4 – The Forth Global Grid Forum, February 2002, Toronto (Canada)
21. Branco, M., Cameron, D., Wenaus, T.: A scalable distributed data management system for ATLAS. Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India)
22. Fanzago, F. et al.: CRAB: a tool to enable CMS distributed analysis. Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India)
23. The CMS Collaboration: CMS Physics: Technical Design Report. Volume I: Detector Performance and Software. CERN-LHCC-2006-001, June 2005
24. Andreeva, J. et al.: CMS/ARDA activity within the CMS distributed computing system. Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India)
25. Egede, U. et al.: GANGA – A Grid user interface. Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India)
26. Ganis, G. et al.: PROOF – The Parallel ROOT Facility. Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India)
27. Legrand, I.: MonALISA : A Distributed service for monitoring, control and global optimization. Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India)
28. Rehn, J.: PhEDEx high-throughput data transfer management system. Conference on Computing in High Energy and Nuclear Physics (CHEP06), February 2006, Mumbai (India)
29. Burke, S. et al.: HEP applications and their experience with the use of dataGrid middleware. Journal of Grid Computing **2**, 369–386 (2004)