High-Energy Physics applications on the grid

Massimo Lamanna
CERN, European Organisation for Nuclear Research, Geneva, Switzerland

Introduction

In this article we will describe the usage of the Grid in the High-Energy Physics environment (HEP) at the beginning of 2008. We will almost exclusively leverage on the experience and plans of the four big experiments at the Large Hadron Collider (LHC) at CERN [1].

This choice has multiple motivations, the most important being the fact that 2008 is the turning point year for these experiments (ALICE, ATLAS, CMS and LHCb) which, after many years of preparations are basically ready to start (first LHC proton-proton collisions are expected to happen mid-2008). These experiments have played a crucial role in the evolution of grid technologies in the last several year and notably in connection with grid infrastructure projects. The most important projects are EGEE (Enabling Grid for E-sciencE) in Europe [2], OSG (Open Science Grid) in the US [3] and NDGF (Nordic Data Grid Facility) in the Nordic countries [4].

In the evolution of grid technology the HEP community and the HEP experiments have played a determinant role. The essential contribution was the enthusiastic promotion of the idea of grid computing formalised and popularised by I. Foster and K. Kesselmann in the late 1990's [5].
The importance of the HEP role can be judged by several facts:

1. The HEP community had already at that time the experience in creating long-lived collaborations across different and geographically distributed entities (Universities, Laboratories etc...) funded by the coherent effort of several funding agencies. The HEP experiments were already exceeding the several hundreds collaborators from several tens of universities in the early 1990's (e.g. CDF experiment at Fermilab US). At the same moment in time, thus still in the preparation phase, the LHC experiments were reaching an even larger scale (the largest LHC experiment, ATLAS, exceeds 2,100 physicists from 167 institutes in 37 countries). In a sense, the HEP world was proving that the collaboration scale the Grid was suggesting was attainable and even desirable when excellence and optimisation of resources requires to cross existing borders (national, institutional etc...).

2. The HEP community had already started a deep reflection about the way to provide the necessary computing power (and data handling capabilities) for the LHC research programme. The experience of the CERN LEP experiments (active between 1989 and 2000 at CERN) and of several other HEP experiments like CDF and D0 (Fermilab), BaBar (SLAC), NA48 and COMPASS (CERN) made very clear the importance of computing in terms of handling very large data

sample (1 PB range). This was not new: from the very beginning, nuclear and particle physics were early adopters of new computing technologies. The new point was the observation that the computing infrastructure (software and hardware) had to be planned well in advance both for cost reasons and to master the increasing complexity of the scientific data. Along these lines, CERN set up a review of the LHC computing (the so-called Hoffmann review [6]) in 1999 to prepare and formally secure the mechanisms to build and maintain the necessary computing infrastructure. Eventually, the LHC Computing Grid (LCG) project, led by Les Robertson (CERN), was started in 2001 [7]. Notably the LCG project was designed with the necessity to "cross" the experiment boundaries, fostering cross-experiment collaborations at the level of base tools (both in the application sector and in the infrastructure).

3. There were many examples of HEP experiments using distributed computing infrastructure well before LHC, notably on national centres like the IN2P3 Computing Centre in Lyon (France) or CINECA and CNAF in Bologna (Italy). The important point is that the Grid concept suggested a complete solution to concrete issues being experienced in the HEP domain (single sign-on, role-based access and global sharing of resources). When I. Foster delivered a very inspiring talk at the CHEP (Computing in High-Energy Physics conference, the lead event for computing in the HEP community) in March 2000, the HEP community was already designing (and validating with simulation studies and prototyping work) a hierarchical model which is still the foundation of the LCG infrastructure (MONARC project [8]).

4. The HEP community was at the heart of the European Data Grid project (EDG), led by F. Gagliardi (CERN) who then initiated the EGEE programme. The HEP experience together with innovative ideas and tools from the Grid community (most notably the Globus project led by I. Foster and the Condor project led by M. Livny) initiated a number of research and development studies on the middleware necessary to provide dependable services for user communities (HEP plus Biomedical and Earth Observation applications). The software stack adopted and evolved in EDG and then in use by EGEE is the underlying base to operate the grid. In parallel, several initiatives have been undertaken by the experiments to provide high-level services to serve specific needs. All the LHC experiments developed layers on top of the services provided by the different infrastructures. The reasons were multiple, but in general we recognise the following patterns:
    - Insulate the physicists community from an infrastructure that is in fast evolution (e.g. AliEN project developed in ALICE)
    - Provide a layer to optimise performances, in particular to increase efficiency, stability and minimise latency (e.g. DIRAC project developed in LHCb)
    - Federate different grids, providing an effective interoperability layer for data processing and data movement (e.g. DQ project in ATLAS)

    As a matter of fact all experiment-specific layers contain all the three patterns, with different level of emphasis depending on the needs of the experiments and the different phases of their evolution. All these projects were effectively a

continuous stimulus to progress (in the HEP community itself and in the grid communities at large). At the same time, they allowed the maximum usage of the resources available from the different infrastructures, overcoming interoperability and instability problems observed in the early stages.

5. The early feedback from the user community was a decisive factor to help the evolution of these complex technologies. The HEP community devoted significant resources (see for example the ARDA project described in this paper [9]) to work in close contact with the middleware communities.

6. The activities in (close connection with) the experiments, eventually matured in a coordinated process to fully close the feedback loop across the different partners. We are observing a sort of relay between the middleware community on one side, the infrastructure on the other and the applications, in particular HEP, on the third side. During the years three main phases have been observed. The first one had the main focus on the development of the middleware, especially prototyped in the pre-LCG phase. The second phase corresponded to the first years of LCG (and EGEE): the goal was essentially to demonstrate (by building it) a worldwide computing infrastructure. Progressively the focus went to a third phase where the feedback (and innovative ideas) are more and more coming from the user communities. I believe that either the role of the applications (HEP and others) will continue to be strengthened (via close collaboration) or the existing momentum will eventually be redistributed across national and application-specific solutions with possible loss of coherence. HEP, especially for the sociological strengths and its power of innovation mentioned at the beginning, is the best guarantee to keep the coherence achieved in the last few years.

In the recent years very interesting patterns of collaborations have been observed across different applications. In all major cases HEP played an important role. Initially the idea of several projects (notably EGEE) was to have the applications "validating" their services (the infrastructure, the middleware) by injecting user requirements and in using prototypes. In this perspective, a "generic" grid will be validated by exposing it to several (the more the better) user communities, effectively covering more and more use cases. It is one of the main successes of these projects to demonstrate grid usage from several applications (e.g. the spectacular usage rise observed in EGEE-2). The key point is anyway different: an infrastructure at the scale of the grid should not only demonstrate its value for a large number of users like a super computing centre but bring additional added value to its users.

On an infrastructure like the grid the applications sit side by side and benefit from each others experience. The fact that every activity had some specific (possibly non general) use case is largely counterbalanced by the fact to find (in a sister application) colleagues sharing solutions, advising etc... A team of scientists (or a company) should join the grid because the balance between the advantages of the new technology are largely exceeding the aggravation to change part of their working system (which is at the base of their activity or their business). Offering working

examples and new opportunities of collaboration should be one of the real methods to attract new application.

The convergence among applications is clearly not easy and cannot be established by decree . There are very positive examples, even between different communities as I mentioned, but should not be the only parameter for success. The convergence on common solutions, even in the HEP community, is not automatic and has not been achieved completely. There are good reasons for this: computing is not a generic tool (at least not yet) and it is on the critical path to get faster and better results. It is therefore understandable that (as we will describe in what follows) in few cases we can already observe full convergence. For some areas we hope more convergence will be achieved in the near future. Ultimately, some diversity will stay.

I think it is difficult to overestimate the importance of the visionary power to the irrevocable move to the grid as the solution for all the computing of all the leading-edge activities in LHC. This is something we have only observed in HEP so far, namely to commit the success of the most important scientific programme to the successful usage of promising new technology. HEP was the only science being at the same ready (technically and sociologically) to move to the grid and of course needing a computing infrastructure at an unprecedented level.

Often a parallel between the Web (invented at CERN during the LEP period) and the grid technologies has been done. The next years will tell if the parallel is appropriate.

The ARDA project in LCG/EGEE

In the case of HEP, a specific effort was set up in the years 2004-2008 to investigate the usage of the grid for the so-called end-user analysis: the ARDA project. In the following we will use some of the activities of this project to guide us in the HEP usage of grid technology and of the LCG infrastructure in particular.

ARDA stands for A Realisation of Distributed Analysis (http://cern.ch/arda) jointly funded by EGEE and by CERN and with substantial contributions of several institutes such as the Russian institutes in LCG and the Taipei Academia Sinica Grid Centre.

With the word *analysis* in HEP we mean all computing activities performed, almost independently, by individual physicists sometimes organised in small teams. In general they share a common software foundation but each individual/team has a set of different executables, in general tailored for a specific scientific task. All analyses share part of the input data (experimental data, both raw and reconstructed plus simulation data) but often rely on private copies of derived data . Frequent multiple passes on subsets of the data are the rule. The impact of this activity on the grid computing is relevant at least in three areas.

On one side, the size of potential user community (in the case of the LHC experiments, several thousands physicists) is a call for a robust system which should be reasonably user friendly and transparent. Analysis is therefore very different from the organised activities (detector simulation, raw data reconstruction, etc...)

which are performed by a single expert team in a coordinated way.

Realistically if a large community has to use the grid this should not force unnecessary changes in the way of working (analysis is a day-to-day activity). With grid technologies being still in a fast-evolution phase the users should be shielded at least by non-essential changes in the internal components of the infrastructure.

The second area is again intimately connected to users expectations. Users are interested to perform analysis on the grid only if they can get a faster turn-around time or have access to larger or more complex data sets. The potential benefit larger resources could be reduced (or even disappear) if one needs continuous expert support as in troubleshooting activities. This observation translates into the requirement of a system which should not only provide sheer power but should be reliable and efficient. In this case the users can rely to have the results back within dependable time limits. High efficiency implies no need for too many time-consuming operations like resubmitting jobs due to failures of the system in accepting jobs, in accessing the data or in returning the results. Simple access to relevanet monitoring information is clearly the key.

The third area is data access. Data access on the grid is a field of research in itself. In the analysis use case users should be empowered with simple but powerful tools to access the data and perform data location functions. HEP is quite unique in the area of data management, as we will see in the following, due to the requirements coming from aggregated data sizes (over several PB per year over several years of functioning of the experiment and physics analysis), the need of replication and broad access (user communities of the order of several thousands scientists).

The LHC and the Grid Projects

The Large Hadron Collider (LHC) will start to operate in 2008. Four major LHC experiments (ALICE, ATLAS, CMS and LHCb) will collect roughly 15 PB of data per year which should be processed, calibrated and analyzed multiple times. Seamless access to the LHC data should be provided for 5,000 scientists in about 500 research institutions worldwide ***. The lifetime of the project is estimated to be around 20 years.

The goal of the LHC Computing Grid Project (LCG also called Worldwide LCG or WLCG) is to prepare and deploy the computing environment indispensable to perform the physics programme of the LHC project. This includes the detector simulation studies to push the detectors' performance to their limit, the calibration and the monitoring of the detectors during the data taking periods, the reconstruction of the raw data and other selection stages. All relevant data should be accessible to all the physicists worldwide participating in an experiment.

The LCG Technical Design Report [10] estimates the computing power required for the LHC data analysis to be of the order of 100,000 CPUs (CPU available in 2004). A globally distributed model for data storage and analysis was chosen. Originally the MONARC project (Models of Networked Analysis at Regional Centers for LHC

Experiments) suggested a hierarchical structure of distributed computing resources (partially modified due to the emerging grid technologies). CERN and multiple computing centers worldwide are providing resources for constructing the LCG infrastructure.

The infrastructure which has been built has a hierarchy of tiers of computing centres. CERN is the Tier0 centre of the infrastructure. Its main functions are the data recording and permanent storage capability (tape system). The system should be capable to sustain up to 1.25 GB/s of data recording rate (ALICE experiment during heavy-ion operations) and store several tens of PB per year. The Tier0 provides CPU power for data calibration and first-pass reconstruction. The Tier0 distributes data to the Tier1 according to policies agreed with each experiment.

The infrastructure has 11 Tier1s. Each Tier1 has custodial responsibility for the data received from the Tier0 and for data processed in the Tier1 layer. Tier1 CPU will be heavily used in data reprocessing and in preparing big data sample for analysis. The Tier1s are: ASGC Taipei, BNL US, CNAF-INFN Italy, FNAL US, GridKa Germany, IN2P3 France, NDGF in the nordic countries, NIKHEF/SARA in the Netherlands, PIC Spain, RAL UK and TRIUMF Canada. All Tier1s have support and data distribution responsibility to the next level in the hierarchy, the Tier2 centres.

So far, around 100 Tier2s are participating in LCG. At variance with the Tier0 and Tier1s, Tier2s have no long-term data storage responsibility. Ultimately they will provide the computing resources for most of the analysis activities (hence serve the majority of the users). Tier2s have also a very important role to provide the bulk of the computing power for simulation activities.

Smaller facilities (Tier3) do exist, essentially to perform analysis on distilled data samples (downloaded from LCG centres). They are outside the scope of the LCG project and they are not discussed here.

The data rates and sizes for the first two years of LHC running are summarised in Table 1 (source: LCG Technical Design Report). The luminosity is $L = 2 \times 10^{33}\ cm^{-2}s^{-1}$ in 2008 and 2009 and then it will reach $L = 10^{34}\ cm^{-2}s^{-1}$ in 2010 (event rate scales up with luminosity; event sizes can also grow due to interaction pile-up). The canonical beam time for proton-proton operations is assumed to be 10^7 seconds in 2008 and 2009. For heavy-ion running a beam time of 10^6 seconds is assumed with $L = 5 \times 10^{26}\ cm^{-2}s^{-1}$.

The column RAW corresponds to the so called raw data, the events that have been read from the experiment read out channels, assembled and passed through a series of on-line filters. These data are recorded (also on tape for long-term custodial storage) at CERN and at the Tier1 (normally guaranteeing at least two complete copies across the whole LCG). Raw data enter in a chain of processing steps generating reconstructed information and analysis objects (ESD and AOD) to allow different types of physics and detector studies. The MC columns correspond to the required simulation data (MonteCarlo). Before the LHC starts this is the dominating activity on the grid (both for the simulation and the corresponding analysis).

	Rate [Hz]	RAW [MB]	ESD [MB]	AOD [kB]	MC [MB/evt]	MC %of real
ALICE HI	100	12.5	2.5	250	300	100
ALICE pp	100	1	0.04	4	0.4	100
ATLAS	200	1.6	0.5	100	2	20
CMS	150	1.5	0.25	50	2	100
LHCb	2000	0.025	0.025		0.5	20

Table 1: Event rate and data sizes at LHC start up for the LHC experiments. ALICE HI refers to the heavy-ion operations. All other entries correspond to the proton-proton operations.

The requirements in terms of CPU[1], disk and mass storage system (MSS) are given in Table 2 (source: LCG Technical Design Report).

Requirements - all experiments				
CPU (MSI2K)	**2007**	**2008**	**2009**	**2010**
CERN Total	10.0	25.3	34.5	53.7
CERN Tier-0	6.9	17.5	22.4	32.8
CERN T1/T2	3.1	7.8	12.1	20.9
All external Tier-1s	19.2	55.9	85.2	142.0
All Tier-2 s	23.6	61.3	90.4	136.6
Total	**53**	**143**	**210**	**332**
Disk(TB)				
CERN Total	2,200	6,600	9,200	12,600
CERN Tier-0	400	1,300	1,400	1,800
CERN T1/T2	1,800	5,300	7,800	10,800
All external Tier-1s	9,300	31,200	45,400	72,100
All Tier-2 s	5,200	18,800	32,400	49,200
Total	**17,000**	**57,000**	**87,000**	**134,000**
MSS (TB)				
CERN Total	4,900	18,000	31,100	45,600
CERN Tier-0	3,400	13,600	23,600	34,500
CERN T1/T2	1,500	4,400	7,500	11,100
All external Tier-1s	9,300	34,700	60,800	92,200
Total	**14,000**	**53,000**	**92,000**	**138,000**

Table 2: The requirements in terms of CPU, disk and tape storage.

The LCG infrastructure is built out of a collaborative effort on top of other projects

1 CPU power is measured in SPECint2000, a benchmark suite maintained by the Standard Performance Evaluation Corporation (SPEC: http://www.spec.org) to measure and compare compute-intensive integer performances. The measure has been found to scale well with typical HEP applications. As an indication, a single-core Intel Pentium 4 processor can deliver about 1,700 SPECint2000. MSI2K corresponds to 10^6 SPECINT2000.

and organizations like EGEE, OSG and NDGF. All these projects have a multiscience character, particularly prominent in the case of EGEE. In all cases the HEP community is one of the major drivers.

It is important to note that 2008 is the start-up year for LHC but also a key year for EGEE. 2008 marks the end of the first part of the EU-funded project launched in 2004 as a 4-year programme (EGEE-1 April 2004-March 2006 and EGEE-2 April 2006-April 2008). A third 2-year phase (EGEE-3) starts in May 2008 but 2008 will be incontestably the year where the plans for a longer-term, sustainable infrastructure will have to be clarified and unfolded.

HEP Analysis

Each experiments has prepared specific mechanisms to ease the access to the grid for their physics community. As an example we will start from the case of ATLAS and LHCb and their system called Ganga.

Ganga is a job-management system developed as an ATLAS- LHCb common project. ARDA started to collaborate with the Ganga team already in 2004 and progressively increased its contribution due to the interest and the potential of this system [11].

The basic idea is to offer a simple, efficient and consistent user interface in a variety of heterogeneous environments: from local clusters to global grid systems. It is natural that a user develops an application on a laptop, moves to a local batch system for optimising the analysis algorithm onto richer data sets and eventually performs full-statistics runs on the grid. Moving from one stage to another applies also in the reverse order (from the grid to the laptop) for example when a bug-fix or an algorithm improvement should be developed and tested.

This approach responds to the fact that the physics analysis (also on the grid) is an activity performed by a large community of physicists using a variety of applications. These applications are typically built on a simulation or event reconstruction framework (foundation framework) which is experiment specific and enriched with custom code provided by each physicist. Ganga supports users using the foundation libraries by appropriate plug-ins simplifying the configuration stage of the foundation environment and of the user-specific with their custom code. On the other hand, Ganga leaves the freedom to run completely independent custom applications (or to contribute new application plug-ins).

Ganga shields users completely from the job submission details (basically the execution back-end is selected by the users by a software switch and Ganga generates the appropriate stubs to execute user code on the available resources). This is essential to allow users to execute on different back-ends in a seamless way as mentioned before.

It is interesting to note that this approach also shields the users from the evolution of the middleware, hence it fully responds to the first area mentioned in the introduction.

Ganga is written in Python. Current versions are available under the GNU Public Licence. Ganga acts as a front-end to submission of computational intensive jobs to a variety of submission back-ends:

- Several batch system including LSF, PBS and Condor
- Grid middleware like different flavours of the LCG/EGEE middleware or NorduGrid (NDGF)
- Specialised workload management systems for the grid such as Dirac (LHCb experiment) and Panda (ATLAS experiment)

Since Ganga scripts are Python scripts, the entire power of Python is available for creating complex tasks yet the user is not obliged to be a Python expert. In tutorials new users typically learn the necessary syntax within the first 30 minutes. In Figure 1 we show a basic example which is used in most of our tutorial sessions.

```
#
# Ganga example
# submit 3 jobs, one local, one on batch, one to the grid
#

j=Job(backend=Interactive(),application=Executable())
j.application.exe="/bin/echo"
j.application.args=["Hello world"]
j.submit()

j2=j.copy() # make a copy of the last job
j2.backend=LSF(queue=?8nm?) # submit to LSF
j2.submit()

j3=j.copy()
j3.backend=LCG() # run on the Grid
j3.submit()
```

Figure 1: A simple example where the same job (Hello world) is submitted to the local machine, a batch system (LSF) and the LCG grid.

Finally, Ganga keeps track of the jobs created and submitted by the user as records in a job repository. This allows the user to manipulate Ganga jobs in between sessions. Manipulations include being able to submit, kill, resubmit, copy and delete jobs. The repository is updated by a monitoring loop which queries all used back-ends for the status of the jobs and updates the status or triggers actions based on the state transition. For example, a job that changes into a completed state triggers the retrieval of the registered outputs from the submission back-end.

Figure 2 illustrates for the very large user basis which has been built around Ganga. It is important to note that around 25% of the user community (over 50 regular users each month) comes from non-HEP communities.

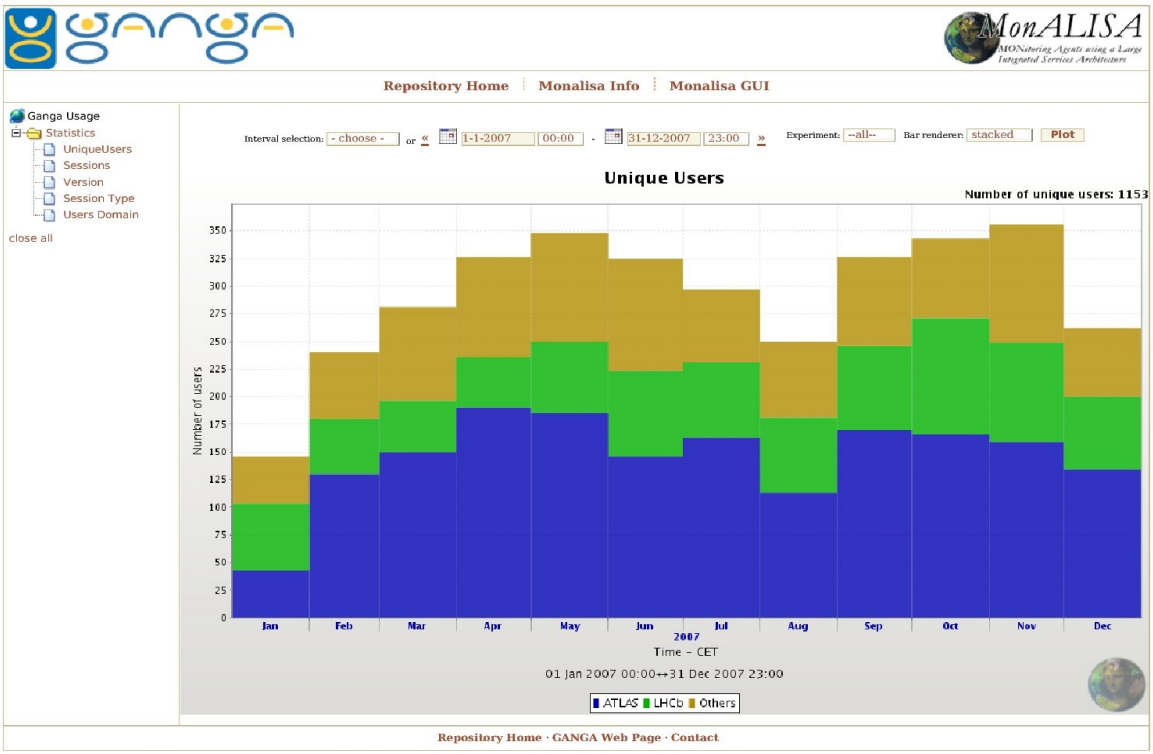


Figure 2: Ganga usage as reported by MonALISA [12]. In 2007 over 1,000 distinct users (unique users) tried out Ganga. Each month, over 100 ATLAS users and about 50 LHCb users use Ganga for their activities. An additional 50 users (25% of the total) are coming from other communities (mainly outside HEP).

As an example of the usage of Ganga outside ATLAS and LHCb I use an example from theoretical physics. QCD describes the interaction of the constituents of the hadronic matter (quark and gluons) and ultimately the structure of nuclei. When QCD is studied on discrete systems (Lattice QCD) it requires non-trivial computing resources. The application that we present here is a study of phase transitions in a quark-gluon plasma [13].

The interest of the example from the computing point of view sits mainly in the fact that Ganga allows a very fast porting of an application onto the grid. The clear scientific advantage is that, with an investment of about 1 week during summer 2007 for porting and running on the EGEE infrastructure, the available statistics has been multiplied by 4 compared to the one collected over several months on dedicated resources.

The application performs a series of iterations descriptive of the space-time lattice to be investigated. Of these lattices 21 different versions exist, all describing slightly different physical conditions. Independent (from a random number generation point of view) programs running on the different lattice configurations produce results that can be statistically added to study the behaviors of the quark-gluon plasma.

Since the result improves with the number of iterations performed and since the result is saved in the space-time lattice it makes sense to run the application for as long as possible (ideally until the batch queue time is reached). Therefore the decision was taken to run in an infinite loop and to regularly send back results to a simple server. This allows the script which runs on the worker-node to be very simple and to make sure that if a job crashes or gets killed the latest result is still available. Since results were sent back every hour on average a job would waste one hour at most (out of several days of running).

We have exploited the natural parallelism (the 21 space-time lattice files) together with the free parameters in the configuration file. With this strategy around 450 jobs were submitted using Ganga to both the EGEE Grid and to the CERN LSF batch system. This resulted in about 9,500 CPU cores to be used. The jobs ran for about one week after which they were terminated (via Ganga). Within this week the results from more than 30 CPU-years could be harvested. A subset of these results have been used for presentation in conferences as Lattice 2007. The jobs ran on more than 50 sites, with a majority of jobs running on fast Intel Xeon processors (see Figure 3).

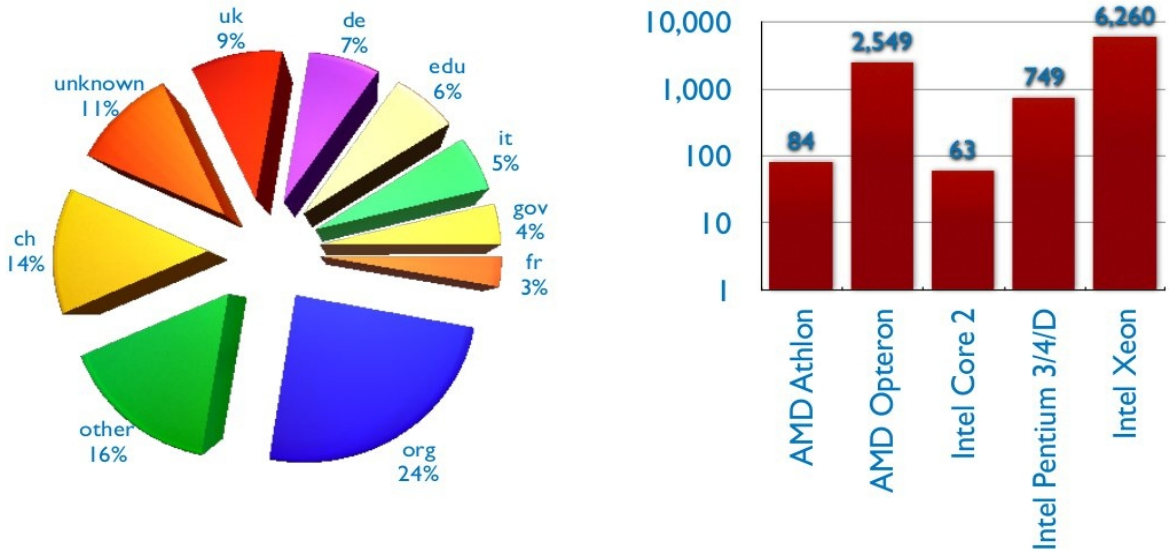


Figure 3: Distribution of top level domains of the sites and the distribution of processors used for the lattice QCD application. Note the log scale in the processor distribution plot.

This example is a neat demonstration of the power of Ganga as a tool to facilitate the usage of the grid. The original goal to isolate HEP users from the details of the execution back-end led to the development of Ganga which is attracting users from different activities. Often new users discover the tool by themselves and then start using it.

Within the EGEE context, we have observed the value of Ganga also as a tutorial tool. The choice of the Python language (its flexibility and the availability of powerful

extension modules) helps to guide the new users into realistic scenarios without unnecessary technicalities. The final result is that users end a 3-hour tutorial and are in a position to continue experimenting and preparing to use the EGEE production infrastructure without further dedicated support effort. Ganga is used in ATLAS and LHCb. ALICE and CMS designed their own strategies to support users on the Grid.

In the case of ALICE, the system conveniently couples their grid back end (AliEn) with the ROOT framework [14] (at the basis of their C++ framework ALIRoot). The key component is a very efficient gateway (a service used by multiple users) to deal with user commands. This service caches the authentication states of the clients in order to provide efficient access for interactive users. This pattern (described in the paper for the original ALICE implementation [15]) is actually more and more used in different areas of the grid middleware since it couples the strict security standards needed by the grid (basically the usage of X.509 security) with the responsiveness needed by any interactive application. As a side (but very important) effect this mechanism avoids excessive load generated by security at the server level since the server does not authenticate all the clients at each interaction but it basically delegates this to a (set of) trusted services. In particular, the searches in the ALICE (AliEn) file catalogue can be done in a transparent way from the user prompt and from ROOT with high efficiency (also implementing features like filename completion etc...). Again, the complexity of the sophisticated solution to provide simple and efficient access is hidden.

In the case of CMS they developed CRAB (CMS Remote Analysis Builder) [16] an application which is somewhat similar to Ganga. In the original form it was basically a client tool helping the user to submit and control jobs on the grid via a convenient set of commands and tools. More recently the usage of an optional server has been introduced allowing disconnected operations like, for example, automatic intelligent resubmission while the user is actually not connected.
CRAB is a very successful application also in terms of user response. In 2007, 20k jobs per day (with efficiency exceeding 90%) have been executed by CMS users making CRAB the most intensively used tool in the HEP grid environment. In the next chapter, we display a snapshot of usage of CRAB in figures 6 and 7.

The Dashboard and the Grid Reliability Tools

Monitoring is a vital component in a distributed system. Grid projects had to invest considerable effort in particular when entering a production phase. The HEP community contributed to this effort, building on previous experience and adding innovative contributions.

It is clear that a tool like Ganga does not prevent execution problems if these are connected, for example, to a misconfigured site or to a failure in the middleware stack. Such investigations need monitoring information. As a matter of fact, all the different actors in the grid world (operation support, middleware developers, individual users, application managers) need easy access and correlation tools on the

available information.

A special role is being played by the Service Availability Monitor (SAM) developed at CERN within the EGEE and LCG projects [17]. SAM is capable to schedule tests on the grid infrastructure (as grid jobs and as commands from grid user interfaces) in order to collect operational data. In Figure 4 the SAM status for a part of the EGEE infrastructure is shown. Computer-centres' statuses are indicated by a colour code. These data are essential to spot operational problems and also to calculate the availability of the different computer centres pledging resources to a given virtual organisation (this is the case of LCG, where monthly reports of the computer centres are published and compared with the expected resources).

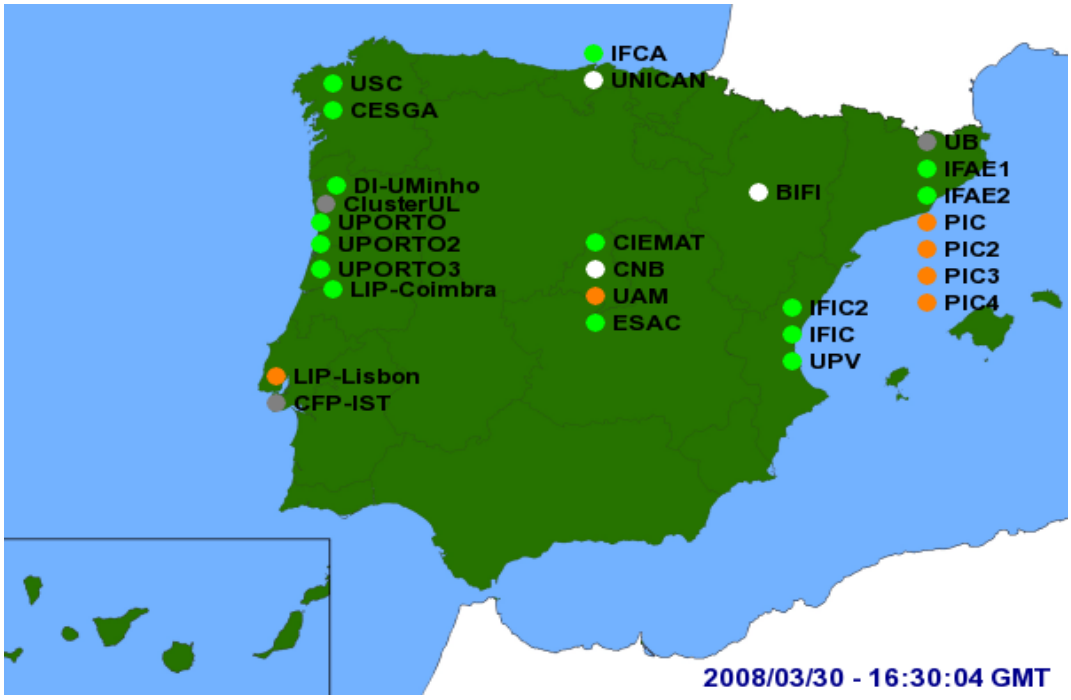


Figure 4: SAM status for a part of the EGEE infrastructure. Computer-centres' statuses are indicated by a colour code.

SAM is an essential tool to operate the grid. In addition it is important to correlate this data with the actual user activity (usage and efficiency seen by the different types of jobs). The correlation is not always very simple due to the different way different jobs (and different user communities) use the grid services offered by the computer centres. A complementary view is needed and the applications should be involved. In practice this generated a collaboration between the HEP user communities and the operation team (at the origin of SAM and other infrastructure-oriented monitoring systems).

The combination of the experience of the monitoring system of CDF (FNAL) and the user monitor of an early ARDA analysis prototype were used to start the CMS Dashboard project (later renamed *(Experiment) Dashboard* since the same foundation is used by all 4 LHC experiments [18]). The project thus started as a

collaboration between ARDA and the CMS experiment.

The strategy was to give to all grid actors the right tool to manipulate and display the available data. The grid operation support, for example, could use the Dashboard to isolate site-specific troubles and use the statistics of error message to fix the problem. Middleware development teams could collect large statistics of error conditions, concentrating on the most common (hence most annoying for the users) factoring out site or application problems. Users are clearly interested to follow the execution (including error conditions) of their own jobs while the activity managers are interested in global figures like resource usage.

In the development of the project, the emphasis was given to the aggregation of existing information and no special effort was devoted in the development of new sensors or protocols. The main components of the Dashboard are then information collectors, the data storage (an Oracle data base) and the services responsible for data retrieval and information presentation (command-line tools, web pages etc...).

The Dashboard is using multiple sources of information, for example SAM. In addition it collects informations from other grid monitoring systems like R-GMA (Relational Grid Monitoring Architecture) [19], GridIce (Monitoring tool for Grid Systems) [20] and IMRTM (Imperial College Real Time Monitoring of the Resource Brokers) [21].
Information from experiment-specific services (like the ATLAS Data Management), central databases (ATLAS Production database) and servers of the MonALISA [12] monitoring system are used. Information is transported to the Dashboard via various protocols (depending on the capability of the information providers).

The collection of input information implies regular access to the information sources. They are retrieved and stored in the Dashboard database. To provide a reliable monitoring system, data collectors should run permanently to recover any missing data in case of failures (and restart the necessary components). The Dashboard framework provides all the necessary tools to manage and monitor these agents, each focusing on a specific subset of the required tasks.

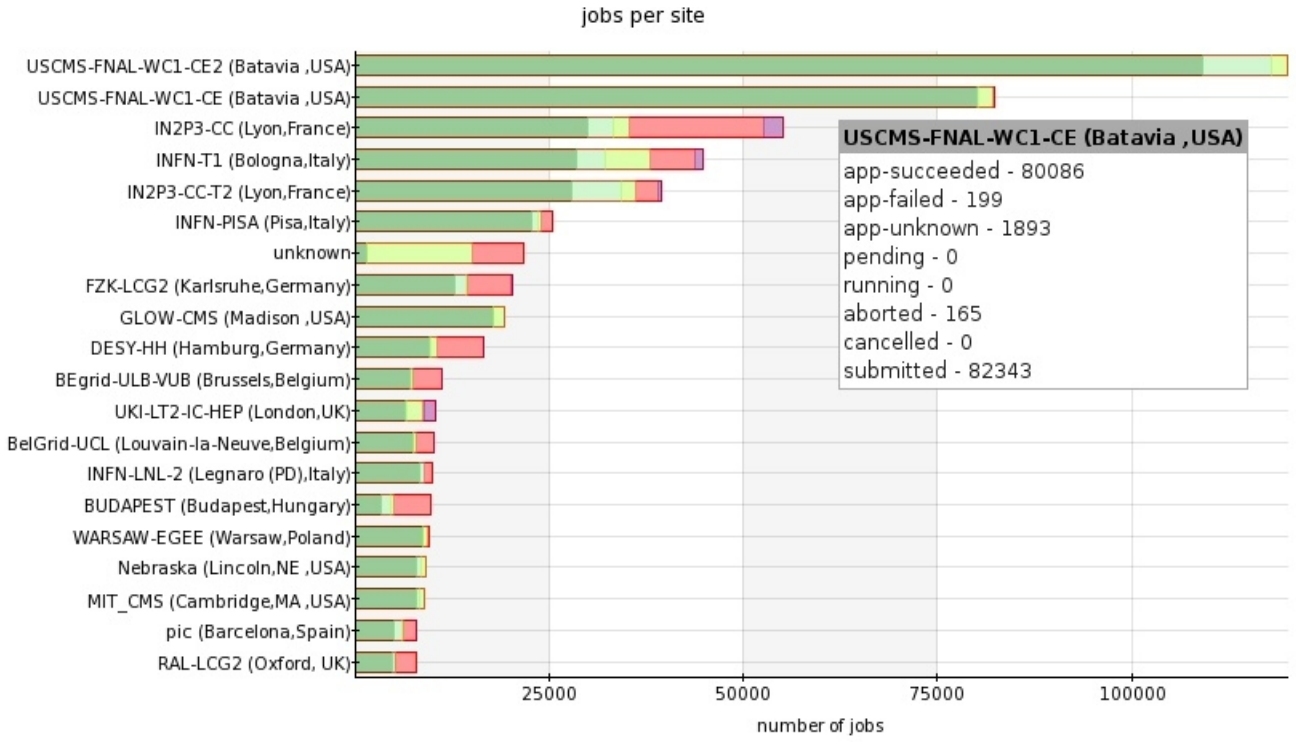


Figure 5: Dashboard Job Monitor. Summary of CMS production jobs (October 2007). The Experiment Dashboard accounts for all CMS jobs on both the infrastructures used by the experiment (EGEE and OSG).

In Figure 5 we present one of the main views of the Dashboard, namely the Job Monitor. We display as an example the summary of CMS production jobs (1 week at the beginning of 2008). It is worth noting that, since the LHC experiments use as a rule more than one grid infrastructure, the Dashboard has been designed in order to collect information from all used resources. The centres listed in the display belong to EGEE with the exception of the US sites (belonging to OSG).

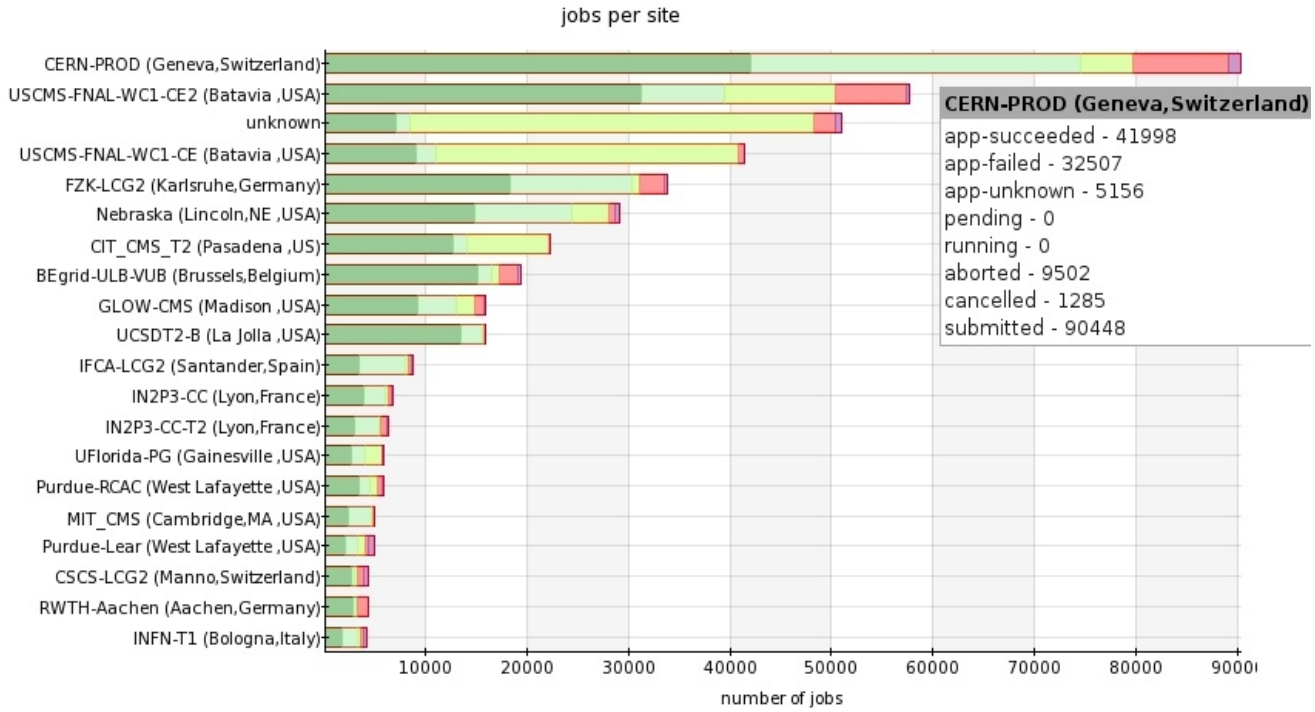


Figure 6: Dashboard Job Monitor. Summary of CMS analysis jobs (October 2007). As in Figure 5 the Experiment Dashboard accounts for all CMS jobs (submitted with the CRAB system) on both the EGEE and OSG infrastructures used by the experiment.

In Figure 6 we present also an alternative view from the Job Monitor. The dashboard database provides here the view of the analysis jobs (submitted by the CMS tools CRAB). These summary views are interesting for both the resource managers both at the participating sites and the ones responsible for the computing of the experiment as a whole.

Users are clearly more interested to concentrate on their own work,in particular to pin down problems in their activity.

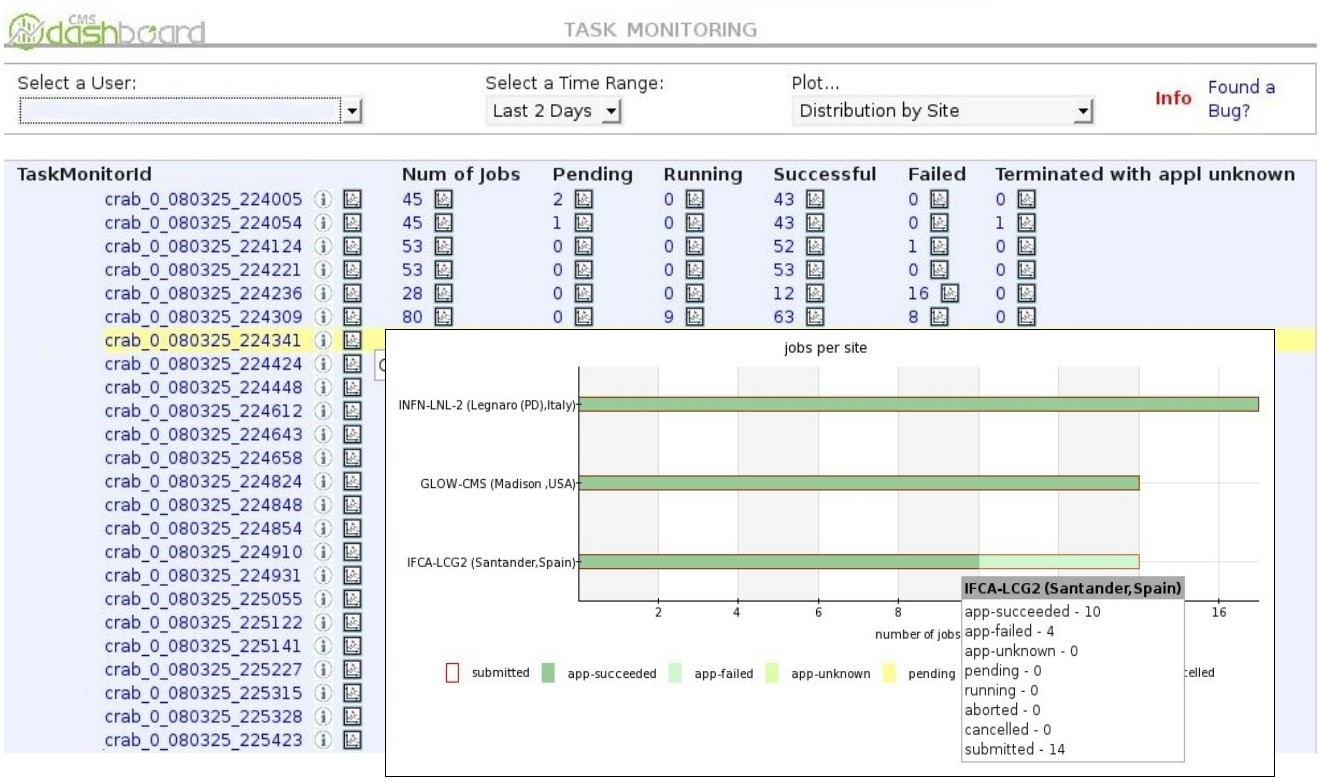


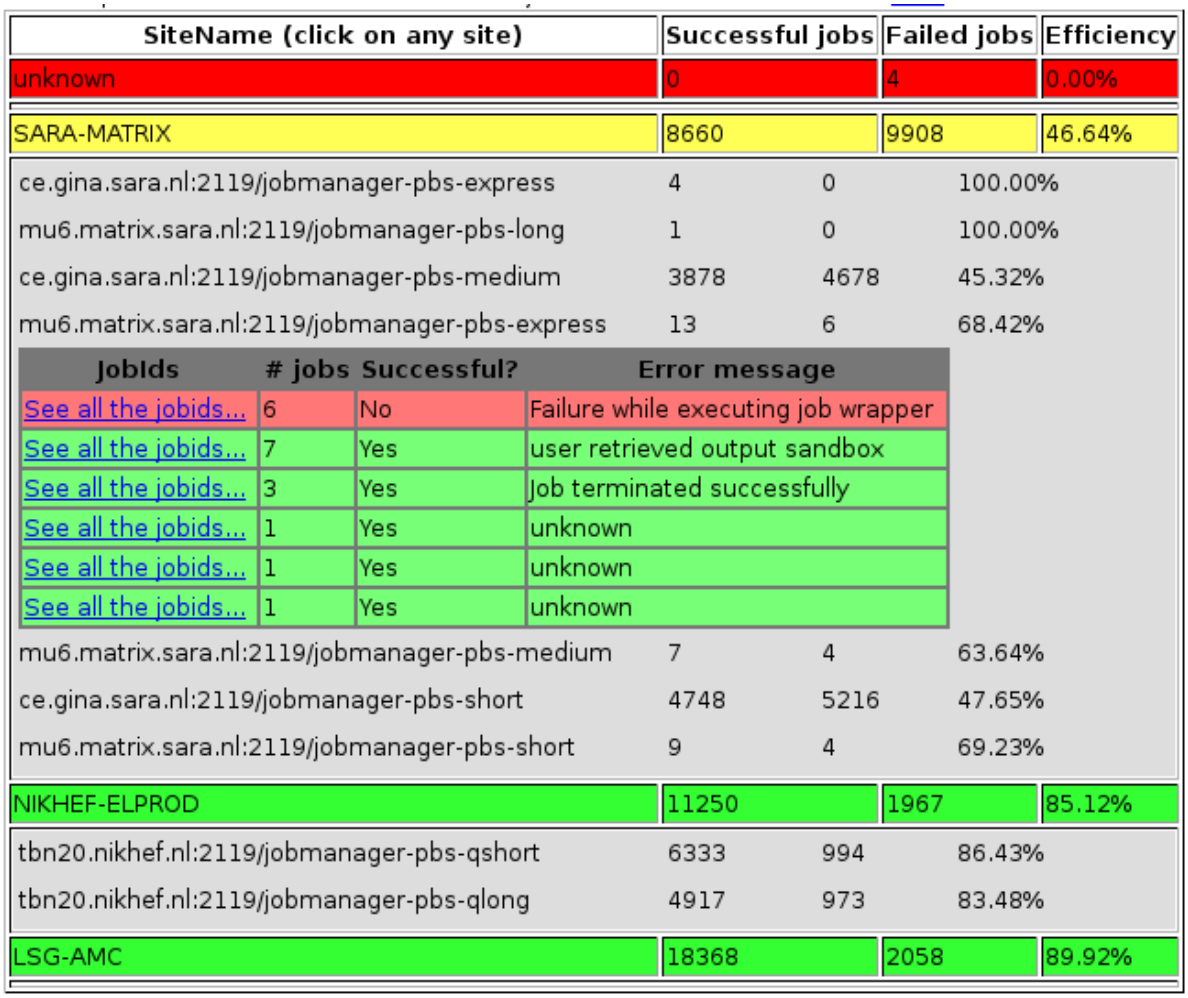
Figure 7: Dashboard Task Monitor. A snapshot of a user page is shown. There is the possibility to have a breakdown of each task (normally a set of jobs sharing the same executable running independently on a coherent dataset, i.e. a set of files).

In Figure 7 we drilled down to the view provided for a given user. It is important to know that a user rarely submits single jobs. Due to the data quantities to be analysed, data are often organised in *datasets*, in general collections of files containing a coherent collection of data. In this case the action to analyse a single dataset generates (in this case within CRAB) a set of jobs (for example one job per data file). Jobs are executed on different sites since data are replicated across the LCG infrastructure.

The importance of an activity like the Dashboard is clear and documented by the interest in the HEP community (usage by the 4 LHC experiments): the Dashboard provides unbiased views of the delivered performances to specific user communities by measuring the efficiency of the users application by monitoring directly the activity of all the users. All of the project (and the Job Monitor in particular) has generated interest in several applications in EGEE. Biomedical applications (VL-eMed) have adopted it and Diligent (Digital Libraries) are considering to evaluate it on their infrastructure.

In Figure 8 we show another Dashboard application: the Site Efficiency. In this case, the Dashboard shows the installation in use for VL-eMed (the same application runs for the HEP communities as well). In this application job attempts are identified and the grid failures are categorized and associated to a given grid resource in a site. In case a job is resubmitted multiple times due to failures each job attempt is taken

into account to test all available grid sites. The main difference with the Job Monitor application (Figures 5, 6 and 7) is that in that case only the final execution of a job is considered. Site Efficiency permits to very quickly identify error patterns, typically connected to a site misconfiguration. In the case of common errors the tool points to a list of explanations/solutions which are accessible via the drill-down functionality of the tool.



SiteName (click on any site)	Successful jobs	Failed jobs	Efficiency
unknown	0	4	0.00%
SARA-MATRIX	8660	9908	46.64%
ce.gina.sara.nl:2119/jobmanager-pbs-express	4	0	100.00%
mu6.matrix.sara.nl:2119/jobmanager-pbs-long	1	0	100.00%
ce.gina.sara.nl:2119/jobmanager-pbs-medium	3878	4678	45.32%
mu6.matrix.sara.nl:2119/jobmanager-pbs-express	13	6	68.42%

JobIds	# jobs	Successful?	Error message
See all the jobids...	6	No	Failure while executing job wrapper
See all the jobids...	7	Yes	user retrieved output sandbox
See all the jobids...	3	Yes	Job terminated successfully
See all the jobids...	1	Yes	unknown
See all the jobids...	1	Yes	unknown
See all the jobids...	1	Yes	unknown

mu6.matrix.sara.nl:2119/jobmanager-pbs-medium	7	4	63.64%
ce.gina.sara.nl:2119/jobmanager-pbs-short	4748	5216	47.65%
mu6.matrix.sara.nl:2119/jobmanager-pbs-short	9	4	69.23%
NIKHEF-ELPROD	11250	1967	85.12%
tbn20.nikhef.nl:2119/jobmanager-pbs-qshort	6333	994	86.43%
tbn20.nikhef.nl:2119/jobmanager-pbs-qlong	4917	973	83.48%
LSG-AMC	18368	2058	89.92%

Figure 8: The Site Efficiency Dashboard application at work for VL-eMed . Job attempts are identified and the grid failures are categorized and associated to computing resources of the sites. The application permits to very quickly identify a specific error pattern.

The future of this activity is that it will continue to grow. The availability of more data allows more sophisticated studies. Very important development are going on to propose a unified mechanism to exchange data (for example using ActiveMQ http://activemq.apache.org/) and to better interface with the different systems used in the grid computer centre (for example using Nagios http://www.nagios.org/). Here the idea is to feedback monitoring data (like grid efficiency at a site) into the monitoring system of the site itself, allowing seamless integration between local established operational procedures and the newly available information.

Data Management

Data management is particularly interesting in the case of HEP. In this case the quantity of data (every year several PB of data have to be *added* to the data store), the replication strategies (multiple complete copies should coexist over the LCG infrastructure to provide redundant storage) and the complex access patterns (especially at the level of end-user analysis) make data management a very interesting problem. ARDA invested a lot in this field, starting from middleware tests to monitor activities. For example a very important part of the Dashboard monitors data transfers at the level of the infrastructure services and at the level of experiment-specific steering systems.

Storage Resource Manager

Due to HEP specific requirements (actually much older than the grid idea) the definition of a standard to interface to mass storage has a long history. In recent years this problem has been discussed in the context of the Open Grid Forum (OGF) which led to the definition of SRM (Storage Resource Manager). The adoption of SRM within LCG considerably accelerated the convergence on a workable standard implementation. The deployment of a non-trivial infrastructure of SRM and the operational experience will in turn be essential in the further evolution of the SRM concept.

The complexity does not only depend on the difficulty of the performance required (data size, number of files, etc...) but also because SRM is effectively an interface to be implemented by the different mass storage systems supported and in use in the grid computer centres. LCG sites use 4 systems, namely CASTOR (notably working at the Tier0 and in 3 Tier1s), dCache (in use on most Tier1s), StoRM (at the Italian Tier1 and under consideration in other centres) and DPM (essentially deployed at Tier2s). Detail of the different implementations can be found under [22].

The experiments' requirements are satisfied with the SRM version 2.2 which is being deployed and now (beginning of 2008) over 160 endpoints are becoming available for the last round of readiness tests before the data taking (CCRC'08). Very much like the operations of the first services in LCG back in 2003, this is a proof-of-existence of the viability of the SRM solution to build such a complex infrastructure. It is clearly a start, since all this area is in constant evolution, but the fact that this infrastructure can be actually operated by shift crews and a good service is delivered to users is clearly very encouraging.

File Transfer Service

As an example of a high-level service built on the existing data infrastructure (and developed in close connection with the HEP community within the EGEE project) there is the File Transfer Service (FTS) [23]. FTS is a layer on top of storage (essentially SRM) and transfer protocols (globusFTP). Its main goal is to provide a dependable service namely a layer hiding short interruptions of the underlying services (essentially by retrying) and avoiding congestions by scheduling data transfer taking into account of the network capacity and shares across users and virtual organisations.

The experiments typically contact this service to schedule a transfer and poll it to see the status. By its nature the service collects bookkeeping information which are also essential for the operation teams maintaining it. In 2007, over 10 PB have been transferred.

Although at these moment this massive data movements are at the heart of the HEP applications only, I believe that in the near future more applications will depend on it to distribute files across vast infrastructures of storage elements.

In Figure 9 we show the data transfer of one of the first tests of the full chain of data acquisition in late 2007. During a week, the ATLAS detector collected cosmic-rays events following the schema expected in normal LHC operations. In this test, ATLAS distributed the raw data and of the centrally reconstructed data onto the full infrastructure (down to Tier2s); end-users performed data analysis at the remote sites.

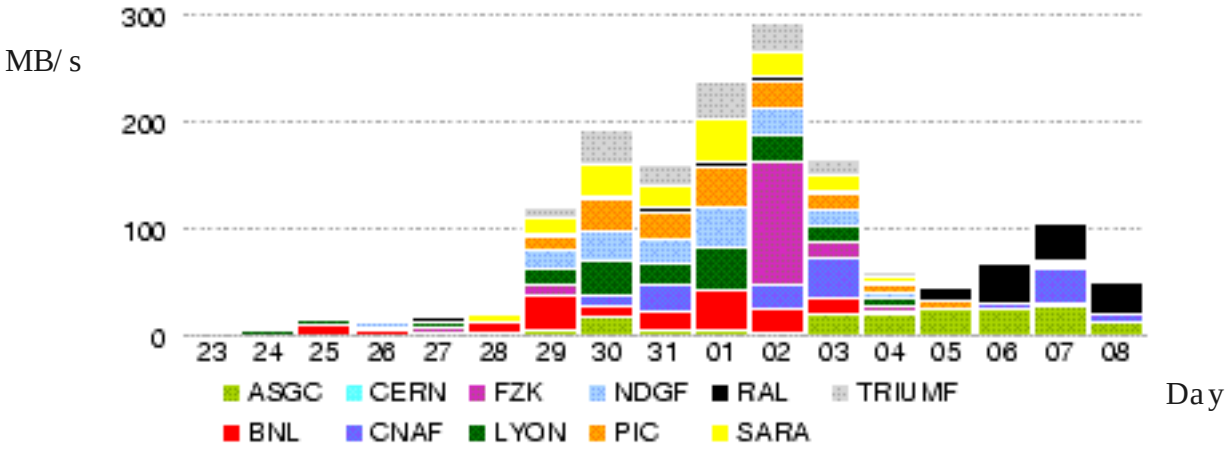


Figure 9: ATLAS cosmics data acquisition (August 23-September 8, 2007). The snapshot of the Dashboard shows the data distribution from CERN to the main regional centres supporting the ATLAS experiments.

Grid catalogues

The EGEE/LCG project has developed a very successful product called LFC (LCG File Catalogue). The LFC is a secure, lightweight and highly scalable POSIX-like file catalogue serving a variety of communities. LFC stores catalogue entries on a data base back-end: supported back-ends are Oracle and MySQL.

In HEP, ATLAS uses the LFC for the local file catalogues located at theTier0 and Tier1: these LFCs control the location of files at each Tier1 (and related Tier2s), while the ATLAS-specific catalogues orchestrate the overall data distribution and bookkeeping. LHCb uses LFC as a global file catalogue. In this case several Tier1s have a full read-only replica, synchronised using Oracle data streaming functionality (Oracle Streams: the replication is performed at the back-end level).

Globally (including non HEP applications) over 100 LFC instances are in use on the

EGEE infrastructure. The largest installations have more than 10 million entries. The evolution of this successful product had always the HEP use cases in mind although inputs from other user communities have been taken into account. During this evolution the product included more and more sophisticated features both to boost performance (like bulk operations for inserting and deleting entries) and to cover security needs (integration with the EGEE security infrastructure, data encryption etc...).

Other catalogues exist developed by the different experiments. One example is the AliEn catalogue which is at the centre of the AliEn system (the ALICE distributed system) [24]. In this case the catalogue keeps not only location information for data files (actually with metadata attributes) but is used by several components of the system. The catalogue contains also the information of software installations available at the different sites and the output of all the jobs.

As the final example of the fruitful collaboration between HEP and other sciences on catalogues, I choose the AMGA metadata catalogue (AMGA stands for ARDA Metadata Grid Access [25]). This system, originally developed by ARDA as a tool to validate the metadata interface in the EGEE middleware, was used as a laboratory to investigate efficient techniques to provide robust and efficient access to database in a grid context. AMGA is the basis of a few systems in the HEP world (most notably the LHCb bookkeeping catalogue).

The AMGA system has been adopted by several applications in completely different domains (see for example the Book of Abstract of the 2nd User Forum organised by EGEE in 2007 [26]). Applications range from Climatology to Multimedia. The application we use here as an example is High-Throughput Screening in Drug Discovery. The first application in this field is WISDOM [27] active on the EGEE infrastructure since 2005. In 2006, a new phase was started with the arrival of new collaborators (most notably by Academia Sinica Taipei [28]) and with the start of a set of campaigns against the H5N1 virus (Bird Flu).

The basic idea is to use the grid to perform collaborative screening of potentially active chemical compounds (called ligands). This activity, called docking, can be executed on the grid by assigning single combinations of proteins and ligands to independent execution units. In order to scale up this activity a central repository is needed (to assign the protein-ligand pairs, to store and display the results and to implement more complex workflows). The choice for this system has been AMGA (Figure 10). The decisive arguments in the choice were the performance and robustness in supporting multiple concurrent clients and its support for grid security.

Especially in the case of H5N1, one of the leading ideas is to prepare for a fast-response system in case of the appearance of dangerous mutation for humans. In 2007, the system has been demonstrated to perform as expected (delivering interesting candidates to be validated in the laboratory). A typical challenge scans several millions ligands using hundreds of CPU-years in a months real time. The result is an handful of promising preselected candidates for validation in the

laboratory.

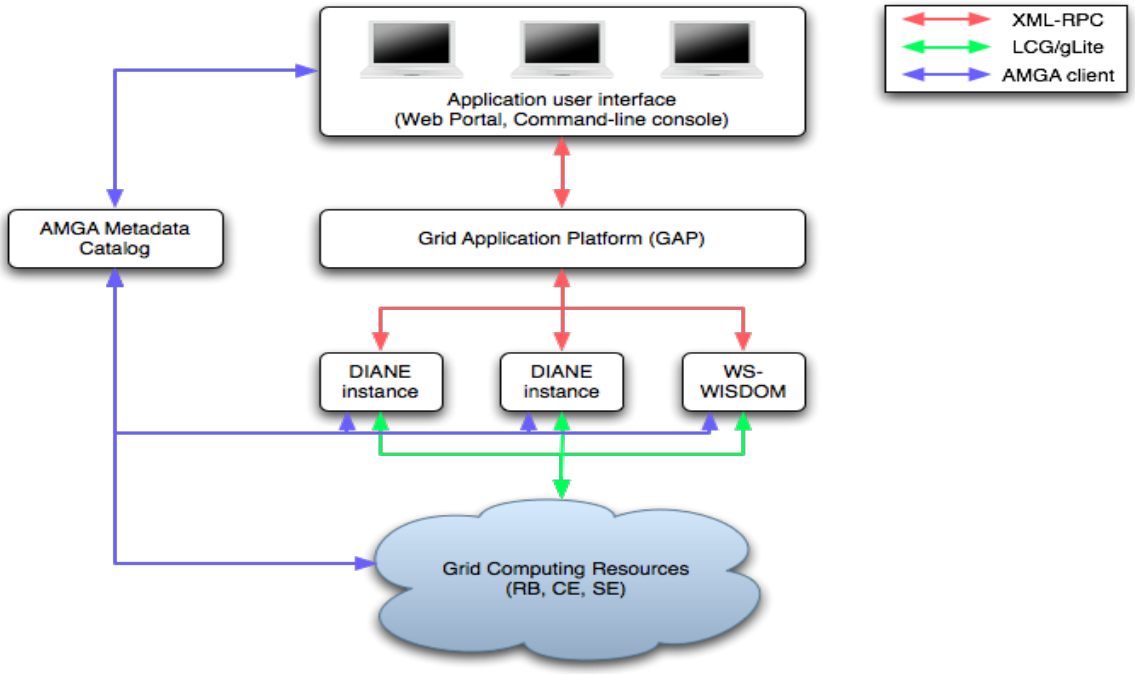


Figure 10: The system in use in the most recent challenges against H5N1 (Bird Flu) showing the integration of the AMGA system. The layer with the DIANE and WS-WISDOM is the component which controls the execution of the jobs on the grid. DIANE is actually a component of the Ganga system.

Conclusions

As mentioned in the introduction, the choice of grid technologies fro the computing in the LHC programme is a major milestone. The actual implementation of a production grid made possible the spectacular growth in usage also outside the HEP communities, in particular within the EGEE project. Close and successful collaboration of the high-energy physics community with other sciences in grid computing (in particular the adoption of solutions in new areas) is a promising sign of the level of maturity these technologies have reached.

Acknowledgments

I would like to thank all the team I coordinated, the so called ARDA team (2004-2008), which was built around the initial core team started in the framework of the LCG and EGEE projects. The team and its activities grew constantly, due to the continuous support of LCG and EGEE plus fruitful collaborations with other institutes most notably ASGC and the Russian LCG collaborators. I would like to thank especially Simon Lin and Eric Yang (ASGC); Slava Ilyin (SINP Moscow) and Vladimir Korenkov (JINR Dubna) for their support and excellent collaboration. A special thank goes to Iosif Legrand (Caltech) for the fruitful collaboration and support especially on the monitoring (MonALISA project). I would also like to thank

Harry Renshall for interesting discussions during the preparation of this manuscript. This work was partially funded by EGEE. EGEE is a project funded by the European Union under contract INFSO-RI-031688.

References

1. General updated information on the LHC programme can be found on the CERN web site (http://www.cern.ch). A recent review article on the first 2 years of LHC is: Fabiola Gianotti Physics during the first two years of the LHC *New J. Phys.* 9 (2007) 332. DOI: 10.1088/1367-2630/9/9/332.
2. Enabling Grid for E-sciencE (EGEE) home page: http://www.eu-egee.org
3. Open Science Grid (OSG) Web Page, http://www.opensciencegrid.org
4. Nordic Data Grid Facility (NDGF) Web Page, http://www.ndgf.org
5. Ian Foster and Carl Kesselman, The GRID: Blueprint for a New Computing Infrastructure , Morgan Kaufmann, 1998.
6. S. Bethke et al., Report of the Steering Group of the LHC Computing Review , CERN/LHC/2001-004, CERN/RRB-D 2001-3, 22 February 2001.
7. LHC Computing Grid (LCG) home page: http://cern.ch/lcg
8. Models of Networked Analysis at Regional Centers for LHC Experiments (MONARC) project home page, http://cern.ch/monarc
9. Massimo Lamanna, ARDA Experience in Collaborating with the LHC Experiments , Proceedings of the Computing in High Energy and Nuclear Physics CHEP06 Conference, editor S. Banerjee, Mumbai (India), February 2006, vol. I, p.1081.
10. The LCG Editorial Board, LHC Computing Grid Technical Design Report , LCG-TDR-001, CERN-LHCC-2005-024, June 2005.
11. Andrew Maier et al., Ganga: a job management and optimisation tool , Proceedings of the Computing in High Energy and Nuclear Physics CHEP07 Conference, Victoria (Canada), September 2007. The Ganga project home page is http://cern.ch/ganga
12. Monitoring Agents Using a Large Integrated Services (MonALISA) project home page: http://monalisa.cern.ch/monalisa.html
13. Philippe de Forcrand, Seyong Kim and Owe Philipsen , A QCD critical point at small chemical potential: is it there or not? , Proceedings of the Lattice 2007 Conference, August 2007, p.178.
14. ROOT is an object-oriented data analysis framework (http://root.cern.ch/).
15. Derek Feichtinger and Andreas J. Peters, Authorization of Data Access in Distributed Storage Systems , 6th IEEE/ACM International Workshop on Grid Computing 2005, 13-14 Nov. 2005; DOI: 10.1109/GRID.2005.1542739
16. Daniele Spiga et al., CRAB (CMS Remote Analysis Builder) , Proceedings of the Computing in High Energy and Nuclear Physics CHEP07 Conference, Victoria (Canada), September 2007.
17. Alexandre Duarte et al., Monitoring the EGEE/WLCG Grid Services , Proceedings of the Computing in High Energy and Nuclear Physics CHEP07 Conference, Victoria (Canada), September 2007. The project web page is http://sam-docs.web.cern.ch/sam-docs
18. A nice review of the Dashboard functionality can be extracted by the following contributions at the Computing in High Energy and Nuclear Physics CHEP07

Conference, Victoria (Canada), September 2007, : Julia Andreeva et al., Grid Monitoring from the VO/User perspective. Dashboard for the LHC experiments ; Ricardo Rocha et al., Monitoring the Atlas Distributed data Management System ; Pablo Saiz et al., Grid reliability .
19. R-GMA home page, http://www.r-gma.org
20. GridIce home page, http://gridice.forge.cnaf.infn.it
21. Imperial College Real Time Monitor: http://gridportal.hep.ph.ic.uk/rtm
22. Scientific Data Management by CRC Press/Taylor and Francis Books, Chapter 3: Dynamic storage management by F. Donno and M. Litmaath.
23. M. Schulz et al., Tools for the management of stored data and transfer of data: DPM and FTS , Proceeding of the Computing in High Energy and Nuclear Physics CHEP07 Conference, Victoria (Canada), September 2007.
24. Stefano Bagnasco et al., AliEn: ALICE environment of the grid , Proceeding of the Computing in High Energy and Nuclear Physics CHEP07 Conference, Victoria (Canada), September 2007.
25. Birger Koblitz et al., The AMGA Metadata Service , Journal of Grid Computing, 6,(1) March 2008, DOI 10.1007/s10723-007-9084-6. The AMGA web site is http://cern.ch/amga
26. EGEE User Forum Book of Abstract. EGEE User Forum, Manchester, May 9-11, 2007 EGEE-TR-2007-002
27. Nicolas Jacq et al., Grid-enabled Virtual Screening Against Malaria , Journal of Grid Computing 6,(1) March 2008, DOI 10.1007/s10723-007-9085-5
28. Hurng-Chun Lee et al., Grid-enabled high-throughput in silico screening against influenza A neuraminidase , IEEE Trans Nanobioscience (2006), 5(4), 288. See also the ASGC Taipei web site: http://www.twgrid.org/Application/Bioinformatics/AvainFlu-GAP