# Achieving Resiliency in Production Worldwide Grid Services for the Large Hadron Collider at CERN

J.D. Shiers, CERN, Geneva, Switzerland

*Abstract*–The world's coolest machine – also the largest scientific instrument to date – will enter production in 2008. Operating at a temperature below $2^oK$, the Large Hadron Collider (LHC) at CERN will generate massive amounts of data – some 15PB per year – that will require significant computational and storage resources. A worldwide production Grid, the Worldwide LHC Computing Grid (WLCG) [1] has been setup, building on the infrastructures of two main Grids – the Open Science Grid (OSG) in the US [2] and the Enabling Grids for E-SciencE in Europe (EGEE) [3] and elsewhere. This is a highly complex system but which must provide a robust and resilient service. This paper describes the state of the Grid in terms of resiliency and is based on a workshop on WLCG Service Reliability held at CERN in November 2007. The goals of the workshop were to discuss and agree the primary techniques for designing, building, deploying and operating robust and resilient services. Concrete targets are to achieve a measurable improvement in service reliability by the time of a WLCG Collaboration workshop in April 2008, and to have fully met the established targets approximately one year later. In this context, reliability is defined as the "ability of a system/component to perform its required functions under the stated conditions."

## I. Introduction

Grids have proven to be an excellent way of federating resources across computer centres of varying sizes into much larger quasi-homogeneous infrastructures. However, as the ultimate distributed system to date, great care must be taken already at the design stage if overall reliability is to be achieved. As an example, the target service levels for compound, cross-site services in the WLCG environment are 99%. These services involve not only Grid components but also the services specific to the LHC experiments, required by key aspects of their computing models. From the viewpoint of the experiments, the most critical services cannot be down for more than 30 minutes without – in the worst case – preventing the experiment from taking data or severely impacting its production processing. This can only be achieved by a high degree of resilience and fault tolerance – it is clearly not possible for humans to intervene on such timescales, unless to perform a basic operation, such as restart of a daemon or system (candidates for automation). We describe below the target service levels presented by the experiments and the key techniques by which we are addressing them. Not all services are currently deployed so as to be able to meet these targets. However, a few simple techniques are believed to be sufficient to address almost all use cases.

Detailed service and availability targets have been drawn up by the WLCG collaboration and form part of the Memorandum of Understanding (MoU) that is signed by all participating sites. The level of service delivered is regularly reviewed by the management bodies of the project.
We summarize below the main lessons learnt from deploying WLCG production services. In particular, we discuss how the somewhat ambitious targets laid out in the MoU are addressed and the various deployment strategies that are required. A strong focus is placed on Reliability, Scalability and Accountability, which together lead to both manageability and usability. Techniques for zero user-visible downtime for the main types of service intervention are described, together with pathological cases that need special treatment. The requirements in terms of scalability are analyzed, calling for as much robustness and automation in the service as possible. The current status of the services is described, together with the outlook for the future, not only for the WLCG itself, but for large-scale grids in general, particularly in the context of moving to sustainable and scalable long-term infrastructures.

## II. The Requirements – "Critical Services"

Although the experiments presented their lists of critical services with differing classifications, the consequences of problems with their services can be grouped into the following categories:

1. Experiment down;
2. Production seriously perturbed;
3. Production perturbed;
4. Annoyance;
5. …

The deployment model of WLCG follows the familiar "tier" model, with the sum of resources at each tier being approximately constant. Tier0 is the accelerator centre – CERN – responsible for data taking and archiving, first pass processing and distribution of the output of this processing to some 10 Tier1 sites distributed around the globe. These sites are responsible for all subsequent reprocessing, distribution of analysis data to ~100 Tier2 sites, as well as provision of custodial data storage services to these sites for simulated data that is produced by them.

It should come as no surprise that, for an environment that is so strongly driven by bulk data, the majority of the services listed as most critical by the experiments are in the areas of database applications and / or data and storage management. Moreover, all of the most critical services are confined to the Tier0 site, with a drop in criticality for Tier1s and Tier2s.

## III. Deployment Techniques

A perhaps surprisingly short list of deployment techniques have been shown to be sufficient to offer robust and resilient services, including zero user-visible downtime for most of the common types of service intervention. These are not sufficient and must be coupled to an operations model that is discussed in details below, as well as ensuring that the corresponding applications and middleware are designed and implemented in such a fashion as to be able to exploit these techniques. There is insufficient room to describe these in detail here, but as they are rather widely deployed we simply list them together with references to further information.

- Understanding the impact of downtime or degradation to service. In some cases, it may even be acceptable for a problem only to be resolved the next working day whereas in others this would clearly be unacceptable: resources being limited, the effort (and money) needs to be focused in the right places;
- Application load balancing, based on dynamic DNS [4];
- Oracle Real Application Clusters, together with DataGuard and Streams for certain purposes;
- High/Availability Linux and hot spares as “catch-all” (and less favoured) solutions.

These techniques not only allow services to be resilient to single (or even multiple) component failure, but permit many of the common interventions to be performed with zero user-visible downtime. These include operating system, database or middleware upgrade or security patches as well as the addition of new hardware / replacement of old or failing nodes. In the case of the best behaving applications, these techniques have been fully supported for a number of years. Further work is required to make all of the main services sufficiently resilient – this is currently underway, being driven by the priorities of the experiments. We give two case studies that typify the use of these techniques below.

## IV. Routine Service Interventions – The LFC

The LCG File Catalog (LFC) – an important data management component for two of the four LHC experiments and used by many other Virtual Organizations outside High Energy Physics – is a good example of both of the favoured techniques, as it is implemented using load-balanced front-ends against an Oracle RAC backend. Only in the case of schema changes – typically scheduled no more than once per year – is there any user-visible downtime during standard service interventions. There is none for operating system patches and upgrades as well as upgrades to the middleware component itself. The technique is also capable of hiding more disruptive interventions, such as move to new hardware. It basically consists of taking one node out of service, performing the necessary intervention, re-adding it to the service and moving on to the next node.

## V. Migrating Oracle RACs

Many data centres nowadays run their production Oracle database systems on commodity hardware, mostly relying on software solutions for high availability and redundancy, software such as RAC and ASM. Such an approach, although very cost effective and increasingly popular, may lead to quite frequent hardware changes due to the relatively short lifecycle of commodity hardware, particularly when compared to certain branded enterprise products. Many commodity hardware vendors give non-extensible 3-years warranty only, which, after all, may impose database migrations every 3 years or even more often. Doing that without significantly compromising service availability becomes a challenge as database systems grow larger and larger. As described above, database services underlie many of the most critical services that form part of the overall WLCG environment and prolonged downtimes for service migration would not be acceptable.

The combination of Oracle RAC with DataGuard [5] – two well known and mature products – not only offers an elegant way of performing Oracle database migrations, but more importantly (and that was our initial goal) it enables such migrations with minimum impact on database users. Despite this unquestionable advantage, the approach is still simple and provides in addition a safety net of a kind, against unexpected problems. Finally, although the procedure presented here covers only the simplest possible case, it can be easily modified and extended to cover also some extra operations including: OS upgrade, cluster resize, storage management layer change and probably many others.

## VI. Building the Service – A Moving Target

Over the past three years, the computing service required by the LHC experiments at CERN has been deployed across some 200 sites in many countries around the world. The experience of this deployment and concurrent hardening is described in detail in [6]. The computing needs of the LHC experiments and the manner in which these are addressed by a worldwide Grid are explained in [7]. To summarize the main difficulties:

- During this period, the computing models of the experiments were undergoing constant development and validation. Experience with the service that was trying to address the requirements of these models often resulted in changes, which in turn meant revised requirements that had to be taken into account by the middleware and often in terms of service deployment. Some examples of major changes that had to be accounted for include the deployment model of the file catalog as well as the data models of all of the experiments, which underwent significant evolution during this deployment period and had a corresponding impact on the architecture of the reliable File Transfer Service;

- Simultaneously, the production Grid infrastructure was being built up – hardware was being deployed, procedures developed, communication channels setup and improved – in other words, the entire operations infrastructure was being designed and built.

Thus we were simultaneously trying to enter more than one "brave new world" at the same time. Whilst this has been an extremely hectic period, the startup schedule of the LHC gave a clear target by which production services needed to be ready (well in advance of the actual startup for essential preparation and validation of the experiments' production systems.) Although commissioning activities of the accelerator itself have uncovered a number of problems that are rapidly being addressed, the computing service needs to be fully established well in advance of first data – to permit the commissioning of the detector and offline systems, which clearly cannot be left to the last minute.

## VII. THE TARGETS FOR THE SERVICE

In terms of the WLCG, the services that a given site must offer, the resources that should be made available to given virtual organizations (VOs), as well as the service availability and maximum time for responding to problems are detailed in the WLCG Memorandum of Understanding (MoU) [8][8] that is signed by each institute that is a member of the collaboration. The status of the services is monitored on an hourly basis by an automated framework with trouble tickets raised in case of problems. The availability of the individual services at a site together with the average availability on a monthly basis is reviewed by the management board of the project, with sites that do not meet their targets required to provide a detailed explanation for the problems seen as well as the steps that are being taken to resolve them. These measures have resulted in a steady improvement in the basic services. The tests are complemented by VO-specific ones and a flexible monitoring framework has been established – designed to complement, rather than replace – existing VO and site monitoring. The target service levels and time to respond are rather aggressive – up to 99% availability is demanded of the accelerator centre at CERN with its complex, compound storage, computational and networking services. Notwithstanding the sometimes tumultuous changes that have taken place during the deployment and ramp-up phase of the project, we have nevertheless arrived at a few well-tested and straightforward strategies for delivering services of the needed reliability, although this has typically had to be retro-fitted to existing services. We suggest that this development, deployment and operational experience is sufficiently generic to be of value to future Grids and indeed similar or at least equivalent strategies will be required if the appropriate service level is to be offered with an affordable level of manpower.

## VIII. RUNNING ROBUST AND RELIABLE SERVICES

Starting in August 2005, and based on the service levels defined in the WLCG MoU, an *a priori* analysis of the Tier0 WLCG services was performed. This analysis was performed as part of the commissioning of the WLCG service, driven to a large degree by a series of four "service challenges", aimed at *"achieving the goal of a production quality world-wide Grid that meets the requirements of the LHC experiments in terms of functionality and scale."* The above-mentioned analysis targeted not only the hardware needs, but also the middleware requirements, operational procedures and all other service aspects involved in setting up robust and reliable services. In addition, the feedback and experience from the early months of Service Challenge 3 – the first that included an attempt to provide a full set of services required by the LHC experiments as well as explicit experiment testing – called for a significant number of service updates. In order to perform these, a "long shutdown" of several days was scheduled during October 2005. It was well understood that such an intervention could not normally be performed on a production service, but this was felt to be the least intrusive method available at that time to perform the numerous pending upgrades – including not only deployment of new middleware releases, but also network reconfiguration, hardware moves and reallocation. Unfortunately, sufficient hardware was still unavailable to redeploy the services in an optimal manner, and their redeployment continued over a period of many months. This was first done using a regular "intervention slot" – simplifying not only scheduling of such interventions with the experiments but also their production planning. However, it was soon realized that the coupling between the various services – not to mention their impact that in many cases extended way beyond the host site and was often Grid-wide – called for a less intrusive manner of performing such changes.

More recently, a re-analysis – an *"a posteriori analysis"* – was performed to take into account not only the prolonged experience with the production service, but also progress in making at least the most common interventions transparent to users. This analysis is still on-going and is likely to be extended to experiment-specific services that require the same degree of reliability as those built on Grid middleware components, but already consensus on the methodologies that have proved effective for delivering robust services has emerged. Before these are detailed, we contrast Grid to cluster / mainframe services.

## IX. THE WHOLE IS GREATER THAN THE SUM OF THE PARTS

The (W)LCG Technical Design Report (TDR) [9] lists two motivations for adopting a Grid solution. These are as follows:

1. *Significant costs of [ providing ] maintaining and upgrading the necessary resources … more easily handled in a distributed environment, where individual institutes and … organisations can fund local resources … whilst contributing to the global goal*

2. *… no single points of failure. Multiple copies of the data, automatic reassigning of tasks to resources… facilitates access to data for all scientists independent of location. … round the clock monitoring and support.*

For funding reasons, the first argument is clearly extremely important – for the reason stated in addition to the fact that many of the institutes involved are multi-disciplinary. Thus, not only for resource sharing within a site but also to bolster the scientific and intellectual environment in the collaborating countries, such a scenario is much healthier than one where all resources are concentrated at the host laboratory (and acquired locally).

The second argument needs further analysis and is indeed similar to the 3rd criterion in Ian Foster's checklist [10][10]:

"*… to deliver nontrivial qualities of service.* (A Grid allows its constituent resources to be used in a coordinated fashion to deliver various qualities of service, relating for example to response time, throughput, availability, and security, and/or co-allocation of multiple resource types to meet complex user demands, so that the utility of the combined system is significantly greater than that of the sum of its parts.)"

With the exception of services and processing that is performed at the Tier0 site, the fact that much of the data – e.g. with the exception of Monte Carlo data in a given Tier2's output buffer – is replicated at several or many sites, the partial or even total failure of a site should not stop the associated production or analysis. Similarly, some of the services – such as the FTS – are already designed to cater for service interruptions at source and/or sink site. For example, if the storage element (SE) at a given site is about to enter scheduled maintenance, the corresponding FTS *channels* that source or sink data in that SE can be paused. This still allows new transfer requests to be queued, but they will not be attempted until the channel is re-opened, avoiding wasting bandwidth on transfers that are bound to fail and reducing the background load on support staff (analysing "obvious" failures).

## X. Building Robust Services

Robust services can only be delivered through careful planning complemented by a combination of techniques, including the appropriate steps at application design and implementation level, as well as at the deployment and operational stage. We describe below very simple techniques that have proven extremely effective and widely applicable in designing and delivering reliable services with a reasonable level of effort and – importantly – largely avoiding fire-fighting and panic.

Two mindsets that are particularly important in this respect are:

- *Think service* – a service is far more than a middleware release in a 'production' repository;
- *Think Grid* – a Grid is the ultimate distributed computing system (so far). A change to a service deployed at a given site or site(s) may well have an impact far wider than the local community and must be planned and announced accordingly.

Before we list the techniques that are in daily use for deploying and operating the WLCG service, we consider some of the issues related to failures and support calls, together with their associated costs.

Consider, for example, the reliable File Transfer Service. Given the expected data volumes and rates, a large LHC experiment would transfer some $10^5 – 10^6$ 1GB files per day. The percentage of such transfers that fail in such a way that human intervention is required must be extremely low, particularly as the problems seen after automatic retries are often complex and time consuming to resolve. Other examples come from user support costs. A ticket that a ticket processing manager spends 1 hour on (and many take much more to solve) has a real and non-negligible cost associated with it. Not all such problems can be avoided purely through good documentation and robust services, but there is clearly very strong motivation to do so. Finally, any operational issues that require human follow up must be reduced to the absolute minimum – anything that can be documented in English (or indeed any other language) can also be programmed as a script or in a higher level language – computers are simply much better at doing repetitive tasks rapidly than humans, whose particular analytical skills are best used elsewhere.

## XI. Check-list for New Services

Before a new service is deployed – be it in a Grid or non-Grid environment – a straightforward checklist has proven invaluable in ensuring that the resultant services are of the required quality. Ideally, this work starts well prior to deployment – the middleware must be designed and written with reliability in mind. This includes details such as error messages and logging – this must be consistent and in an agreed place to which the necessary support teams have access if required (the latter is non-trivial in the case of cross-site services). The application must be designed to cope with "glitches" – e.g. short-lived problems with services on which they depend and which are simply unavoidable in a distributed environment. Where possible, the ability to share the load across multiple load balanced servers offers numerous advantages, including transparency to many common service interventions and even middleware upgrades. In the case of a database backend, the ability to re-establish a connection and – assuming a database cluster – failover transparently from one node to another are highly desirable, if not mandatory, features. The appropriate hardware must obviously be allocated – avoiding (except in cases such as batch worker nodes) single points of failure through power supplies or feeds, network connections and so forth. Finally, a minimum set of operational procedures – including contact names and addresses – together with a basic set of tests (no contact, high load etc.) is needed. The necessary workflows also need to be established in the support lines, together with diagnostic tests for the various levels of support / operations teams. Starting with these essentials, the service manager can readily add more tests and procedures as experience shows are required.

## XII. Daily and Weekly Operations Meetings

One of the key secrets to running smooth services is a regular operations meeting. This has been in place at CERN since decades before the Grid and used to be performed by vendor (CERN having a number of large mainframes / clusters at that time). In recent years, these meetings have been extended to cover the Grid world, with a clear impact on the state of the Grid services. On occasion, people have expressed ‘disappointment’ that there is not an atmosphere of mad panic / firefighting at these meetings – but this is precisely the point – these meetings are to ensure a smooth service, exactly the opposite of firefighting. Instead of being an overhead, these meetings act as an excellent point of information exchange, and in fact significantly reduce the amount of time spent on identifying and debugging problems. The meetings typically take around 10 minutes – slightly longer on Mondays – and quickly run through alarms and problems seen since the last meeting. If not immediately solved, the problems are assigned to a system administrator or technical expert as appropriate. More importantly, they expose weaknesses in the services – such as lack of adequate monitoring or alarms – and peer pressure proves a very effective mechanism for ensuring that these holes are rapidly plugged. Once a week, any outstanding tickets against the CERN Regional Operations Centre are reviewed, again ensuring that problems are not left unaddressed for prolonged periods. Another important topic that is reviewed daily is any intervention scheduled for that day, or any foreseen in the coming days. It cannot be stressed too highly how important adequate preparation for interventions has repeatedly been proved to be – it is not just a question of informing fellow service providers and users, but also ensuring that the intervention proceeds smoothly. All too often, a well debugged procedure runs into problems (often because it is not strictly followed, or as the availability of needed colleagues for a given step has not been checked), turning a smooth or even “transparent” intervention into a prolonged downtime that may even need a further intervention to adequately complete. In the worst cases, unannounced “transparent” interventions have resulted in severe service degradation that have led to extreme user dissatisfaction and have been extremely costly in terms of manpower to resolve. We have therefore agreed simple procedures for announcing scheduled interventions of various lengths, as well as unscheduled interventions. Equally importantly, an announcement through the agreed channels is required when the service is fully restored (or periodic announcements in case of prolonged problems), as well as an open post-mortem, recording any unforeseen problems, their resolutions and lessons for the future.

These daily – primarily site-oriented (see caveat above) meetings are complemented by weekly joint operations meetings with all the main sites that have a similar agenda but also include VO-specific issues. In addition, a service coordination / planning meeting, that addresses changes coming up in the next one to two weeks, ensures good information flow between the various groups involved in the service. Finally, less frequent meetings are held to ensure that the operations tools adequately address the needs of the community. These meetings are typically held bi-annually.

## XIII. Mind The Gap

During the conference mentioned earlier, a number of service problems came up that highlighted the need for well documented – and followed – procedures, as well as excellent communication. To be explicit, three significant service problems came up in a single week – all of which were easily avoidable. These were as follows:

- A bug in Oracle client libraries – both documented and already fixed in production releases – caused a number of daemons to go into an infinite loop (after 248 days of uptime – the maximum number of clock-ticks (1/100 s) that can be represented in a 32bit signed integer). An analysis of the problem revealed that not only are there numerous methods for deploying Oracle client releases but also there was no consistent agreement for which of these to use, nor for moving to new versions;
- A change to the service availability algorithm – approved in principle at the various management boards – was released in production without being scheduled or even announced via the regular operations meetings. This caused significant knock-on effects in other service monitoring tools;
- A database house-keeping exercise resulted in an index being de-selected, with following service overload and meltdown.

Whilst it is unlikely that all such problems can be avoided in the future, we cannot afford to tolerate such a high rate of completely avoidable issues. Hopefully, the experience from these events will reinforce the widespread adoption of the simple and lightweight procedures that have been shown to work in exactly these situations.

## XIV. *Caveat Emptor*

A final piece of cautionary advice concerns coupling between services and the sometimes unexpected consequences. Two concrete examples in this area relate to the choice of database synchronization technology that we have deployed. This is used for the file catalog middleware component and for detector calibration and alignment information – in both cases the corresponding information is kept in sync between the main sites with minimal delays. However, this has meant on one occasion that a key database feature had to be disabled – with significant consequences on the ability to recover the service in case of accidental loss of data – and on another led to silent data corruption. On balance, the benefits certainly outweigh the drawbacks, but underline the need for openness and transparency – the reasons for choosing a specific release and the consequences must be clear to all, particularly in the case of complex, layered services.

## XV. Conclusions on Robust Services

Taken together, these techniques and procedures have been demonstrated to be sufficient to offer robust and resilient services, but are unfortunately often overlooked. We know how to run reliable services – this is not to say that no user support issues remain! The issue of support for large and diverse user communities of a system with the complexity of the Grid is certainly one of the challenges that will need to be addressed by future e-infrastructures. In particular, it is essential that we neither design nor use Grids in such a way that the unavailability of a single service renders a site – or worse, the entire Grid – down. Such problems should, in the worst case, result in a small inefficiency of the overall Grid resources, rather than a downtime.

## XVI. remaining challenges for grids

Grids have proven extremely effective for aggregating significant resources across many different management domains, offering unparalleled computing power in a relatively consistent manner. There is no doubt some entry threshold below which it does not reasonably make sense for an individual site to join a Grid and correspondingly economies of scale for larger sites. Thus, rather than fund a larger number of uni-disciplinary sites, there is motivation for a smaller number of centers of excellence – provided that the challenges of providing robust and usable services, together with an acceptably light-weight migration process to the Grid can be solved. Finally, the need for both generic middleware as well as application specific support needs to be emphasized. The former will hopefully reduce to something of the order of operating system support that is required today. On the other hand, application support can be expected to have strong benefits for many years to come and should be fully supported in future projects.

## XVII. conclusions

After many years of research and development followed by production deployment and usage by many VOs, worldwide Grids are a reality. There is significant interest in longer-term sustainable infrastructures that are possible with the current funding models and work on the definition of the functions of and funding for such systems is now underway. Using a very simple classification of Grid applications, we have briefly explored how the corresponding communities could share common infrastructures to their mutual benefit. Finally, the operational and support costs of Grids need to be contained and the difficulties in supporting new communities and their applications minimized. These and other issues are being considered by a design study for a long term e-infrastructure [11][11].

## Acknowledgments

Many people have contributed to and continue to contribute to the ramp-up and delivery of WLCG Services. It is impossible – for simple space reasons – to mention every individual involved in this work. With apologies to those not explicitly mentioned, which includes regular participants in the LCG Service Coordination Meetings and the joint operations meetings – whilst gratefully acknowledging their work – the following warrant an explicit mention. The WLCG Service Coordination team consists of (in addition to the author): James Casey, Flavia Donno, Maarten Litmaath and Harry Renshall. Patricia Méndez Lorenzo coordinated and presented the *"a posteriori"* analysis of WLCG Tier0 services and Tim Bell coordinated the original *"a priori"* work. Ulrich Schwickerath, Miguel Coelho Dos Santos, Jean-Philippe Baud, Harry Renshall, Gavin McCance, Maria Girone and Dirk Düllmann all provided input to this recent analysis.

## References


[1] The Worldwide LHC Computing Grid (WLCG), http://lcg.web.cern.ch/LCG/.

[2] The Open Science Grid, http://www.opensciencegrid.org/.

[3] The Enabling Grids for E-sciencE (EGEE) project, http://public.euegee.org/.

[4] DNS load balancing and failover mechanism at CERN, V. Bahyl, N. Garfield, in the proceedings of the International Conference on Computing in High Energy Physics, Mumbai, India, February 2006.

[5] Cross-Platform Database Migration with Oracle Data Guard, CERN openlab technical note, Maria Girone, Jacek Wojcieszuk, Dawid Wojcik, Graeme Kerr

[6] J. D. Shiers, *"Lessons Learnt from Production WLCG Deployment"*, to appear in the proceedings of the International Conference on Computing in High Energy Physics, Victoria, BC, September 2007.

[7] The Worldwide LHC Computing Grid (worldwide LCG), Computer Physics Communications 177 (2007) 219–223, Jamie Shiers, CERN, 1211 Geneva 23, Switzerland

[8] Memorandum of Understanding for Collaboration in the Deployment and Exploitation of the Worldwide LHC Computing Grid, available at http://lcg.web.cern.ch/LCG/C-RRB/MoU/WLCGMoU.pdf.

[9] LCG Technical Design Report, CERN-LHCC-2005-024, available at http://lcg.web.cern.ch/LCG/tdr/.

[10] I. Foster, Argonne National Laboratory and University of Chicago, What is the Grid? A Three Point Checklist, 2002.

[11] The European Grid Initiative Design Study – website at http://www.eu-egi.org/.