Storage and Data Management in EGEE

Graeme A Stewart[1], **David Cameron**[2], **Greig A Cowan**[3] **and Gavin McCance**[2]

[1]University of Glasgow
Glasgow G12 8QQ
UK,
Email: g.stewart@physics.gla.ac.uk

[2]European Organization for Nuclear Research
CERN CH-1211
Genève 23
Switzerland

[3]University of Edinburgh
Edinburgh EH9 3JZ
UK

Abstract

Distributed management of data is one of the most important problems facing grids. Within the Enabling Grids for Enabling eScience (EGEE) project, currently the world's largest production grid, a sophisticated hierarchy of data management and storage tools have been developed to help Virtual Organisations (VOs) with this task.

In this paper we review the technologies employed for storage and data management in EGEE, and the associated Worldwide LHC Computing Grid (WLCG). We describe from low level networking and site storage technologies, through data transfer and cataloging middleware components. A particular emphasis is placed on deployment of these services in a large scale production environment. We also examine the interface between generic and VO specific data management, taking the example of the ATLAS high energy physics experiment at CERN.

Keywords: Grid Computing, Data Management, Grid Storage

1 Introduction

The EGEE project, (EGEE Website 2006), is an EU funded project to develop a large scale eScience infrastructure for Europe. Now comprising over 200 distinct sites offering resources (see Figure 1), the project, now in its second phase, also has extensive links beyond Europe. One of its key goals is to develop a generic middleware stack, known as gLite (gLite Website 2006), to provide a high level of infrastructure for Virtual Organisations (VOs) using the grid to tackle their problems.

One of the key application areas for EGEE is high energy physics, with a particular orientation towards the Large Hadron Collider (LHC) experiment, currently under construction at CERN and due to begin data taking in 2007. The project to provide the computing infrastructure for the LHC is the Worldwide LHC Computing Grid (WLCG), which is a very important part of the EGEE project. The WLCG project also uses resources from the Open Science Grid (OSG Website 2006) in the United States and NorduGrid (NorduGrid Website 2006) in Northern Europe.

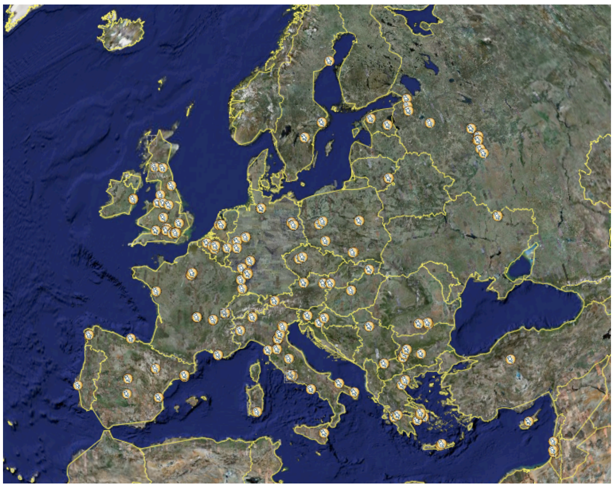

Figure 1: European EGEE/WLCG Sites

1.1 The Data Management Problem

Reliable movement and storage of data is a cornerstone of distributed systems. For data grids, where the volumes of data to be moved are huge, data management must offer a high degree of control, to ensure that data is placed correctly for processing and safe keeping; but also offer a view at a sufficiently high level to ensure that data placement can be managed efficaciously. Robustness of data storage and replication is also paramount – the data management system must be able to cope with as many errors itself as possible, saving valuable human time for those events where intervention really is necessary.

However, experience in the predecessor to EGEE, the European Data Grid (EDG) project, showed that these goals are difficult to achieve. Many problems were encountered with data management – particu-

Graeme Stewart and Greig Cowan wish to acknowledge the support of the GridPP project, funded through the UK's Particle Physics and Astronomy Research Council (PPARC).

larly that driven by grid jobs themselves (Baud and Casey 2004, Burke et al. 2004).

- When jobs triggered a data transfer from a remote site they would be unaware if the file requested was already in transit, usually requested by another job.
- File transfers were triggered without any controls on resources at the source and destination, often leading to overloading of storage elements at sites and consequent failure.

In addition performance problems with the file catalogs developed for EDG meant that catalog updates could fail, which led to retransmission of already transferred files, increasing load and further reducing efficiency.

Having seen how badly affected a data grid could be by poor data management middleware, concerted efforts were made in EGEE to improve data management software.

These efforts have improved data management considerably: from new storage solutions offering manageable SRM functionality to even the smaller sites (Section 3.2.3), new middleware components improving the robustness of file transfers (Section 4) and improvements in the file catalogs offered in the grid (Section 5.1). In addition dedicated high performance networking (Section 2.1) has been provisioned to carry the huge amounts of data that the LHC experiment will generate.

An overview showing how the different components described in this paper connect is shown in Figure 2.

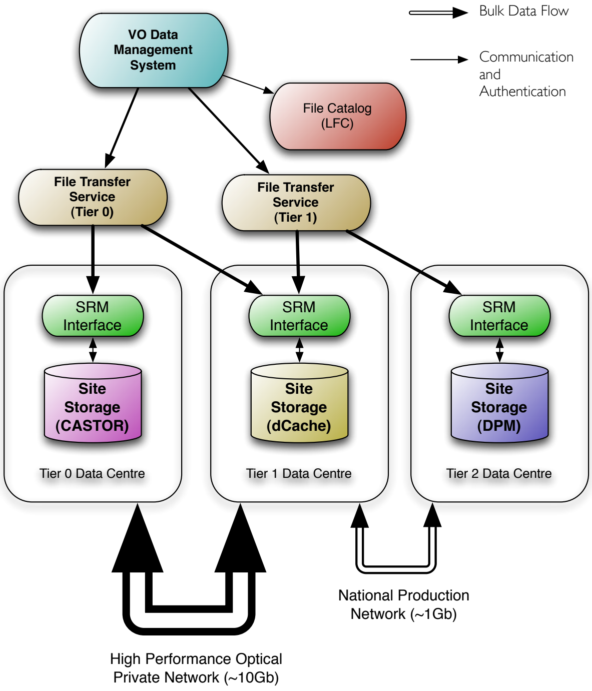
Bulk Data Flow
Communication and Authentication
VO Data Management System
File Catalog (LFC)
File Transfer Service (Tier 0)
File Transfer Service (Tier 1)
SRM Interface
Site Storage (CASTOR)
Tier 0 Data Centre
SRM Interface
Site Storage (dCache)
Tier 1 Data Centre
SRM Interface
Site Storage (DPM)
Tier 2 Data Centre
National Production Network (~1Gb)
High Performance Optical Private Network (~10Gb)

Figure 2: Overview of data management components in EGEE

2 Networking

2.1 The LHC Optical Private Network

The network is clearly a fundamental component for any grid – good networking is the zeroth law of grid computing. Without the ability to move data and results between sites the grid cannot exist.

This is particularly apparent in data grids, where such huge quantities of data need to be moved that this becomes one of the most expensive operations on the grid.

The Large Hadron Collider, currently under construction at the European Centre for Nuclear Research (CERN) in Geneva represents an extreme in data volumes, as it will generate upwards of 14PB of data a year, which requires to be distributed across the EGEE, OSG and NorduGrid grids. Such data volumes cannot be handled with current production networks, and so have required the provisioning of optical private network (OPN) links between CERN (Tier-0 centre) and the key computing centres around the world (Tier-1 centres). These links will provide dedicated 10Gb network pipes for LHC data transport.

2.2 Regional Networking

The requirements on networking from the Tier-1 centres to the smaller regional computing centres (Tier-2 centres) are much less arduous. In general, therfore, this networking is provided through the national academic networks in the region (NRENs).

Some Tier-2s, however, have a particularly large processing capacity for LHC work, and so in these cases sites may be connected to national high performance optical networks.

2.3 Data Transport Layers

As with all production network links data transport between sites is managed using TCP. TCP is the proven technology for providing reliable data flow between sites.

On top of TCP, a modified version of the standard FTP protocol (RFC959 1985) is used. Known as GridFTP, this protocol builds on standard FTP to incorporate x509 based security (RFC2228 1997), allowing seamless interaction with Grid Security Infrastructure (GSI). GridFTP also includes extensions (RFC2389 1998), such as multi-streaming, which can help to utilise the full bandwidth available on network links (Allcock 2003).

Currently it is GridFTP version 1 (from Globus Toolkit 2) which is supported in the EGEE storage solutions, however moves are underway to use the new GridFTP version 2 protocol (from Globus Toolkit 4) as this alleviates many problems experienced from the interaction of GridFTP with firewalls, and addresses some of the limitations of the version 1 protocol, (Mandrichenko 2003).

2.4 Network Optimisation

As part of the process of optimisation of data transport on the grid the network stacks on the source and destination hosts must be tuned. This must be done carefully, lest the stability of the machine be affected by overly aggressive kernel settings (Cowan et al. 2006), but on long links use of advanced TCP congestion control algorithms, such as TCP-BIC (Xu et al. 2004), can provide a noticeable performance benefit (Ferrari et al. 2006).

3 Storage Technology

As the storage requirements of some of the EGEE VOs (particularly the LHC experiments) are so considerable, computing centres are required to provide large amounts of storage. This ranges from custodial long term tape storage provided at the Tier-0 and

Tier-1 centres, to shorter term disk based storage at the smaller Tier-2 centres.

3.1 Storage Interface: SRM

A design goal of the grid is that this range of storage technologies should be, as far as possible, transparent to the VOs. In order for this to happen an abstract layer needs to be provided on top of the particular storage implementation used at each site.

The *Storage Resource Manager* (SRM) working group (SRM Working Group 2006) has been set up with just such a goal in mind. SRM provides a web service interface to storage, offering the basic functionality necessary to add files to a storage element, `srmPut()`; extract them, `srmGet()`, or delete them, `srmAdvisoryDelete()`.

Currently in EGEE it is SRM version 1 which is deployed and used (Bird et al. 2001). However, the functionality available through SRM v1 is not sufficient for the LHC VOs. In particular deficiencies were identified in regard to being able to control storage on tape, which is very secure, but slow, as opposed to data held on disk, which is a more fragile medium, but has a much faster access time. Indeed, use cases have been identified where LHC VOs require data to be held on tape and on disk simultaneously.

This functionality is being addressed in SRM v2.2, which is being developed rapidly for deployment later this year. This introduces reserved spaces, which can be tagged with the properties required by the VO (e.g., disk only, optimised for wide area access). In this way users can request the particular types of space necessary for particular sets of data.

3.2 Site Storage

As the scope and scale of storage services required from Tier-0 down to Tier-2 centres is so huge, it is no surprise that there is no single software product which can deliver the sophistication required by the Tier-0, yet be simple enough to be managed by a Tier-2.

3.2.1 CASTOR 2

To manage storage at CERN the CERN Advanced STORage manager has been developed (CASTOR Website 2006, Ponce 2006). This is a continuation of development which started with the SHIFT system (Scalable Heterogeneous Integrated FaciliTy), which then developed into CASTOR 1 in the late 1990s.

The design goals of the CASTOR 2 system are to provide reliable central data recording of the data from the LHC experiments, as well as transparent access to this data. CASTOR 2 is designed around a mass storage (tape) system, and is therefore not suitable for deployment at sites without this facility.

CASTOR 2 provides a single namespace for all the files it manages. Access over the LAN is provided via the `rfio` and `root` protocols (support for `xroot`, the eXtended root protocol, is under development), with WAN access being mediated through SRM and gridftp.

Architecture CASTOR 2 has been designed around a central RDBMS system, which handles, as much as possible, the state of the system. Development now focuses on Oracle (support for PostgreSQL is frozen), and the resilience of this database is key to CASTOR's reliability.

Around this database core all daemons are designed to be stateless, allowing for redundancy and daemon restarts without loss of service.

The key component of CASTOR 2 has been the new stager, the component which manages the disk pools in front of the tape system (see Figure 3).

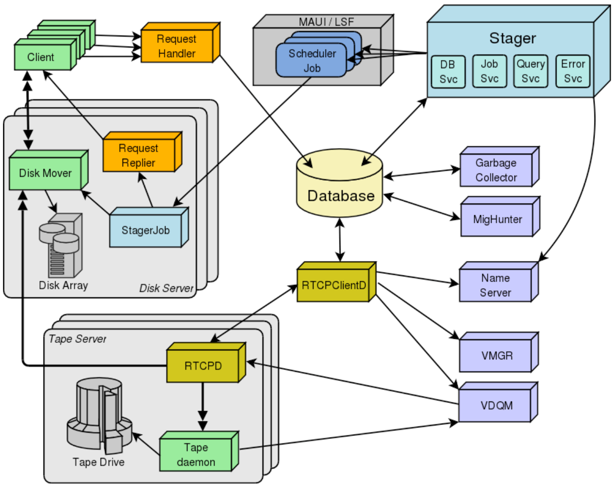

Figure 3: Architecture of CASTOR 2

In order for the stager to manage access to files on the disk pool it uses a scheduler plugin. This balances the load on each of the disk servers and can also provide policy and 'fair shares' considerations to file access. The scheduling problem for disk pools is in fact rather similar to that faced in computing farms (Lo Presti et al. 2006) and CASTOR currently uses the commercial LSF batch system scheduler. Support for the Maui scheduler is currently frozen.

CASTOR has the ability to dynamically replicate 'hot' files, and even to switch access on an open file to a less busy replica.

Logging and Monitoring A system as complex as CASTOR 2 needs constant monitoring to ensure the service is performing at an optimal level. Integration with the LEMON (LHC Era Monitoring) system at CERN is advanced, with alarms being issued when any abnormal conditions are detected. Logging for CASTOR 2 is done into the Oracle database, allowing the central gathering of logging information, plus the ability to cross query logs from different services.

Deployment and Status As CASTOR 2 is such a large installation it only exists at the largest sites in EGEE. These are CERN, plus 4 of the WLCG Tier-1 centres. The CASTOR 2 instance at CERN currently manages more than 50 million files and 5PB of data.

3.2.2 dCache

dCache (dCache Website 2006) is a system jointly developed by Deutsches Elektronen-Synchrotron (DESY) and Fermilab that aims to provide a mechanism for storing and retrieving huge amounts of data among a large number of heterogeneous server nodes, which can be of varying architectures (x86, ia32, ia64). It provides a single namespace view of all of the files that it manages and allows access to these files using a variety of a protocols, including SRM. By connecting dCache to a tape backend, it becomes a hierarchical storage manager (HSM).

Architecture An operational dCache system is composed of a number of *domains*, each running in its own Java virtual machine. Each domain (and its constituent *cells*) has a specific role to play, such as dealing with file access requests from clients, updating the filesystem namespace or facilitating communication between domains. The disk storage is partitioned into a set of disk *pools*, each of which can be assigned properties which control how files within the dCache behave, depending upon particular client requests. It is possible for all dCache services to exist on a single node, however, typical deployments (particularly at the Tier-2 level) have the SRM and namespace services separated from the data transport and disk pool services.

dCache Namespace The single dCache file namespace is provided by PNFS, the 'perfectly normal file system'. Written in C, PNFS provides a filesystem namespace view through an nfs2/3 interface. Chimera is the next generation replacement to PNFS and is currently undergoing evaluation. Like dCache, it is written in java. Chimera will provide filesystem emulation, based on a relational database management system which will allow for efficient lookup of file metadata.

File Access dCache provides local area access to its managed files though a LAN access protocol `dcap` (dCache access protocol) or `xroot`. WAN access is provided using dCache's own implementation of GridFTP, as well as the SRM protocol.

A recent enhancement to dCache is the ability to configure multiple I/O queues. Such a system allows for WAN and LAN file access to be separated, preventing the (slower) WAN I/O from blocking (fast) user access to files across the LAN.

Advanced Features Built into dCache is the ability to automatically load balance the system upon the detection of hot spots (i.e. files that generate a lot of read requests). In this case, the dCache will replicate the files in question to other disk pools in an attempt to smooth the overall load pattern. If all pools holding the particular dataset are busy then the dCache can stage the files in from a tertiary storage system (if present). Upon the arrival of a specific set of files, they can be automatically flushed to tape or moved to a set of disks that are configured specifically for clients to read files from them.

The dCache replica manager module allows for a dCache instance to operate with additional resiliency by controlling the number of replicas of each file within the system. The administrator can set the policy such that there will always be between N and M files available, each on a separate pool. This implies that all files will remain available in the case of pool failure or pool downtime for maintenance. Operating dCache in resilient mode can allow for efficient utilisation of available disk space present on the worker nodes of a batch farm.

dCache has an advanced pool selection mechanism that allows the system to be configured such that data is written or read to specific pools depending upon the IP address of the client, the PNFS directory the client is trying to write to, the protocol being used and the direction of the data flow. This allows for a single dCache instance to be partitioned. For example, if an experiment uses `xrootd` to access its data, then dCache can ensure that all files owned by the experiment are held on a particular set of disk pools that have been paid for by the experiment.

There is a command line interface (accessed via ssh) and a GUI that can be used for controlling pool setup and the load balancing configuration.

Status of Deployment and Support As of the writing of this paper, dCache has been deployed in a production environment at almost 40 WLCG sites. The scalability of dCache can be demonstrated by studying the composition of these sites. They range from small single machine installations that are present at some Tier-2 sites, all the way up to large LHC experiment specific centres such as that at Fermilab. This dCache installation is composed of around 300 pools, serving on the order of 200TB of data a day and dealing with 50 file open requests per second. Work done by the UK's GridPP collaboration (GridPP Website 2006) has significantly improved the integration of dCache with the WLCG installation method, YAIM.

Support for dCache users and administrators comes from a variety of sources. The dCache "Book" (de Riese et al. 2006) contains information about the system architecture and instructions for deployment. In addition, there is an active community of dCache users who support each other via the user-forum mailing list and a direct dCache support list. Workshops are regularly organised to discuss the latest dCache deployments and configurations.

3.2.3 DPM

The gLite Disk Pool Manager (DPM) was originally developed by the LCG project at CERN, specifically to address the issue of Tier-2 storage. It has now been adopted as part of the EGEE gLite middleware stack. Its emphasis is on ease of configuration and maintenance – Tier-2 centres rarely have dedicated storage support staff and the main requirement is for a storage system which will work reliably and simply with minimal interventions.

DPM is also oriented towards the hardware setups found in a typical Tier-2 centre, in particular DPM only supporting disk based installations.

Architecture DPM's architecture is shown in Figure 4. Separate daemons control the namespace (DPNS); the status and configuration of the disk pools and filesystems (DPM); the SRM service (SRMv1 and SRMv2). For a typical Tier-2 all of these daemons coexist on the same machine, but if a larger deployment is undertaken, these daemons can be run on separate nodes.

DPM is written entirely in C, and much of the nameserver code is shared with CASTOR.

DPM disk servers, the machines on which the storage is actually hosted, run gridftp daemons for WAN transfers and and `rfio` daemons for LAN access.

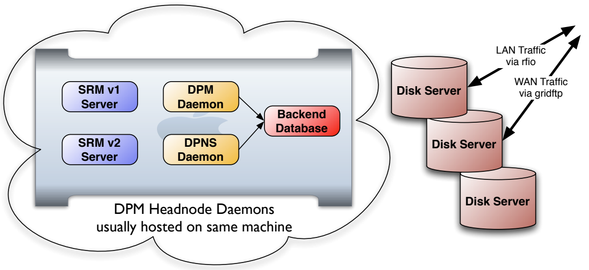

Figure 4: Architecture of DPM

DPM requires a backend database to operate. In most Tier-2s MySQL is used, although support for Oracle is also provided.

Configuration Almost all of the configuration for DPM is held in its database, which makes DPM very easy to configure via its command line utilities. Given that Tier-2 centres have limited time to spend on managing their storage resources good support for DPM is given through the YAIM configuration tools.

Load Balancing DPM applies a round robin approach to selecting filesystems for incoming data. While this is not very sophisticated, compared to dCache or CASTOR, it should suffice for EGEE Tier-2 centres. It does require Tier-2s to make important decisions about the system's layout at deployment time (Lemaitre and Baud 2006).

Support for replication of files within DPM, i.e., having two physical copies of 'hot' files, is available, but at the moment no automatic replication features are provided.

Maintenance In order to ease the life of Tier-2 system administrators utilities are provided in DPM to drain filesystems and migrate data to other disk servers, allowing maintenance work and scheduled interventions on the system.

Status of Deployment and Support DPM is deployed at more then 70 sites within EGEE. Support is provided primarily through the EGEE regional operations centres. Access to developers and other experts is provided through the GGUS ticket system.

3.3 Storage Optimisation

Part of the goal of the WLCG service challenges has been to ready the infrastructure of the grid for data taking at the LHC. Storage optimisation has played a large part in this.

Tier-1 centres have tuned their systems through the WLCG 'service challenge' transfer tests. This has achieved a peak rate of 1600MB/s exported from CERN around the world, as seen in Figure 5.

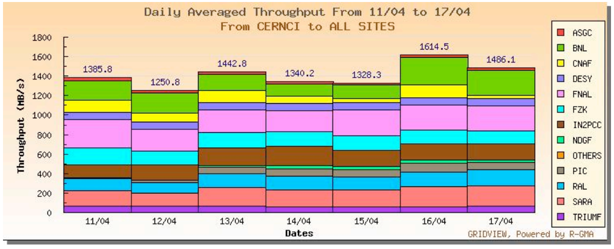

Figure 5: Disk to disk transfer tests as part of WLCG service challenge 4

For Tier-2 centres the EGEE Regional Operations Centres (ROCs) have taken the lead in testing. The UK-Ireland ROC undertook an extensive series of tests of its Tier-1 to Tier-2 data transfer rates. This revealed a considerable number of problems at Tier-2 centres whose storage, up to that point, had not been particularly stressed (Stewart 2006).

In particular the following lessons were learned:

- Separation of the SRM service node from the disk server pools improves reliability greatly.
- As the i/o from each disk server is limited, a number of smaller disk servers will perform better than a single large one.
- Network provisioning is time consuming and needs to be undertaken well in advance of need.

More detailed tests of filesystems and transfer parameters relevant to smaller sites have also been undertaken (Cowan et al. 2006), which has shown that modern filesystems, such as `xfs`, can improve transfer rates by a significant margin.

4 File Transfer Service

As noted in Section 1.1, the reliable transport of data is one of the cornerstones of distributed systems. Transport mechanisms have to be scalable and efficient, making optimal usage of the available network and storage bandwidth. In production grids the most important requirement is robustness, meaning that the current high performance transfer protocols need to be run over extended periods of time with little supervision. The transfer middleware has to be able to apply policies for failure, adapting parameters dynamically or raising alerts where necessary. In large Grids, we have the additional complication of having to support multiple administrative domains while enforcing local site policies.

The gLite File Transfer Service (FTS) is a Grid fabric infrastructure service designed to provide sites with a reliable and manageable way of serving the file movement requests of their VOs. Some of the key concepts of the FTS are outlined below.

4.1 Channels and Management

To ease management of the service, users' transfers are assigned to different channels upon submission. Each channel corresponds to a specific point-to-point link between two sites or between groups of sites. An FTS server serves a configurable set of channels. Every channel is unidirectional, i.e., it is intentional that different FTS instances may serve the two directions of a network link between two sites. Usually the receiving site configures the corresponding channel, as it is shown below.

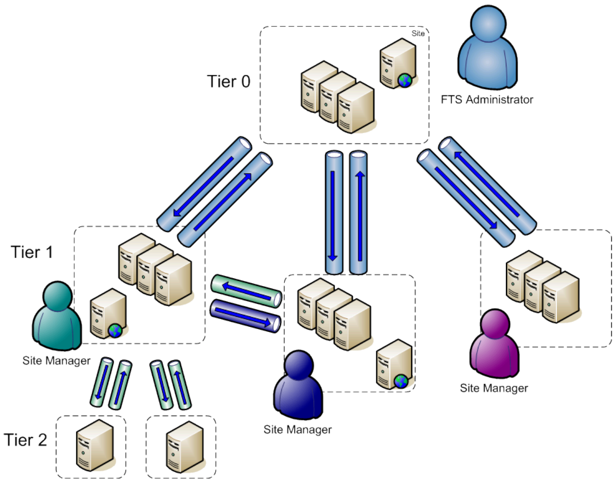

Figure 6: The dedicated channels layout in a multi-tier hierarchy

The channel is the logical unit of management of the service. All transfers on the same channel share the same properties (such as what transfer mode they are running, the default TCP buffer size, default number of parallel streams, etc.). The bandwidth used and the relative VO shares are also controllable separately for each channel. Individual channels may be turned off without affecting the others, for example if a site needs to go down for a manual intervention. User's jobs are queued while the channel is switched off. We distinguish between *Dedicated* and *Non-dedicated* channels.

Dedicated channels correspond to point-to-point connections between sites. In the optimal production case these are dedicated network links where the FTS is set up to have full control over the reserved bandwidth and the capacity is not shared with any other service. If some sharing exists, the parameters can still be tuned empirically to achieve an average high bandwidth but it is of course more difficult to maximise bandwidth usage at all times in this case. Usually the FTS runs two channels, one for each direction of a dedicated channel.

Non-dedicated or 'catch-all' channels correspond to groups of sites, for example, 'all other sites to me' or the full catch-all 'everywhere to everywhere'. These are ideal for small VOs, where the administrative burden of many channels is excessive. For larger scale production environments, they also serve to "sweep up" the transfers that are not otherwise assigned to the main dedicated channels.

4.1.1 Client Interaction

The users of the service interact with it via a webservice which exposes a SOAP based interface. The service uses a submit / poll pattern for job submission. Each transfer job may contain multiple file transfer requests. The job's state machine is exposed to the users following the basic Submited, Pending, Active, Done/Failed pattern. Full information on job or file failures is provided to the client upon terminal failure of a job.

The calls to the service are secured via X509 certificates, and full audit is available to the service administrator. The actual transfers are performed using the user's X509 credentials retrieved and renewed from a Myproxy server, so the source and destination SRM servers also have full audit of who has been using them.

Future work is planned to provide a notification mechanism for clients (to avoid them having to poll for the job status). It will make use of a suitable existing notification transport (e.g. basic XMPP or a more advanced and persistent MQ based messaging system).

4.1.2 Administrator Interaction

The service defines multiple classes of administrator: service administrator, VO administrator, and channel administrator.

- VO administrators have full access to the details of transfers belonging to users of their VO. They can view, cancel and reprioritise these as necessary. Typically a VO manager, with responsibility for bulk data movement of the VO, will have this role.

- Channel administrators have full access to details of transfers assigned to their channels. They can view, cancel, and hold these transfers. They are also able to change the operational parameters of the channels such as the current bandwidth utilisation, stopping a channel for a manual intervention, etc. Typically, the site administrators of the source and destination of a channel will have this role.

- Service administrator has the rights to create and drop new channels and assign new administrators. They also have fuller access to the logs and monitoring functions of the service.

The roles are fully definable on the server and are authorised by X509 certificate subject name or Virtual Organisation Membership Service (VOMS) role.

Further monitoring is available, via the webservice statistics interface. Web-based summaries describing service utilisation and current service problems are in preparation. These will allow the site administrators to know immediately if there are any problems with their storage services (from the client point of view) and will provide debugging information to them.

4.1.3 Server Architecture

The FTS server consists of multiple distinct components, with minimal coupling to ensure maximum service availability. It was a design goal to be able to upgrade, fix and stop one part of the service without impacting upon the rest of the service. For example, the channel agents can be down for upgrade while the web-service stays up, ensuring that clients are still able to submit new jobs and query the status of existing ones while the intervention is ongoing. The agents daemons are distinct and are designed to be spread over multiple machines to cope with the load. The components are:

- the database. This is the only critical component of the service since all the state of the service is kept here. All components share access to the same schema. The FTS currently only supports an Oracle based schema, though work is ongoing to port a "light" version of the service to MySQL. The service will run on any current version of Oracle (9i or 10g). We rely on the availability features of Oracle 10g (Oracle Real Application Cluster) to provide maximum availability for this critical component.

- the web-service. This exposes the secure SOAP based interface allowing users to submit and query jobs. It also exposes a WSDL for channel management functions for authorised channel administrators. The server is stateless and is designed to be load-balanced (for example by DNS load-balancing) for maximum performance and availability.

- the VO agents. These daemons work on specific states of the job, and apply VO-specific policies to the job. For example, the VO agents are responsible for applying the retry policy in case of a transfer failure and this may be different for different VOs. The VO agent also has hooks for extra pre- and post-transfer plugins, such as VO-specific cataloguing operations.

- the channel agents. These are responsible for performing the actual transfers for a given channel, handling the interaction with the SRM and gridFTP servers. Both third-party gridFTP and SRM copy transfers are supported.

The VO agents plug-in via a Python based plug-in mechanism to allow VOs to replace the default retry policy and cataloguing policy with their own policies.

4.1.4 Deployment

The FTS is currently deployed on the EGEE preproduction testbed and is deployed in production for the WLCG. There are choices to be made in how to deploy the FTS servers so as to provide full transfer coverage with the minimum amount of resource utilisation. The deployment should:

- provide full coverage for all the use cases. i.e. there should be no use-cases whose transfers cannot be accomplished by some FTS server in the transfer network;
- it should prefer dedicated channels for the major production point-to-point links. This way sites have fuller control over these major production channels;
- each FTS server should have a manageable set of channels. For example, having a single server for the whole Grid defining a vast $n(n-1)$ mesh of channels is not manageable;
- the servers should go where the manpower resources are concentrated, as much as possible. This usually means preferring national computing centres, and making that centre responsible for all transfers belonging to its country.

For the LCG production grid, there is a single server at CERN which is responsible for all the data going into or out of CERN; the largest data flows are between CERN and the Tier-1 national labs. These are managed on dedicated FTS channels which run transfers over the dedicated network links of the OPN, described in Section 2.1.

Each national Tier-1 centre has an FTS server which is responsible for pulling data from other Tier-1 centres and is responsible for controlling the data transfer into its associated Tier-2 sites. Typically, the Tier-1 centre defines one channel per other Tier-1 centre, pulling data from it. It makes use of various catch-all channels to serve the lower bandwidth needs of the Tier-2 centres.

For reasons of manpower and to reduce the overall complexity of the transfer network, FTS servers are not usually deployed at Tier-2 centres.

5 File Catalogs

As noted in Baud and Casey (2004) poor file catalog performance in EDG severely impacted on the data management functionality available to VOs, and placed severe limits on the numbers of files which could be handled.

File catalog implementations typically assign a unique identifier to each catalog entry, usually called a Globally Unique Identifier (GUID). This GUID then maps to an entry in the catalog's namespace, known as a Logical File Name (LFN) and to the physical replicas of this file, known as the Site (or Storage) URLs, or SURLs.

The catalog design in EDG used a separate catalog to map GUIDs to SURLs (in the the Local Replica Catalog, LRC) and to map GUIDs to LFNs (in the Replica Metadata Catalog (RMC). This meant that queries which required access to information in both catalogs were particularly slow.

5.1 LFC

To address the problems with the EDG catalog tools, a new file catalog was designed, the *LCG File Catalog*, or LFC.

The schema architecture of the LFC, shown in Figure 7 unified the LFNs, GUIDs and SURLs in a single database, greatly speeding up queries.

The LFC is implemented in C and shares code with the CASTOR and DPM nameservers. Both MySQL and Oracle database backends are supported for the LFC.

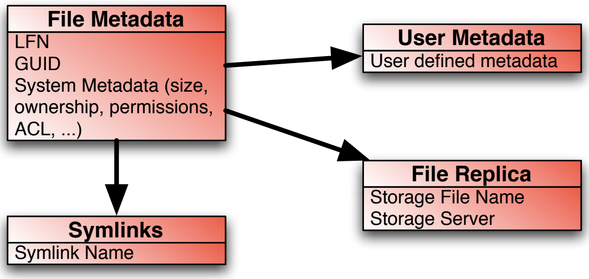

Figure 7: Architecture of LFC's schema

5.1.1 Performance

Baud et al. (2005) evaluated the performance of the LFC and found that, even on modest hardware, millions of entries could be sustained and that query rates of 100s per second could be sustained.

5.1.2 Deployment and Operations

Due to the critical importance of the central file catalog for the LHC VOs, the central catalogs at CERN are deployed using a redundant Oracle database backend.

In addition LFCs can be deployed as local catalogs and these are used by several of the LHC VOs, including ATLAS. Currently there are about 50 local catalog instances in EGEE.

6 VO Data Management

Despite the improvements in data management tools described above, the complete set of data management functionality offered can still be below that required by a particular VO. One of the principal limitations is that data management, for the middleware, happens at the file level – there is no general concept of filesets or data groupings.

Although there are some moves to develop generic higher level data management components, e.g., the gLite AMGA metadata catalog, frequently the problem is encountered that at the higher level each VO sees their data in quite a specific way. While efforts to find generic solutions will continue, it often falls to the VOs themselves to design and implement the higher levels of data management functionality.

As an example of this, within the WLCG context, we examine the ATLAS experiment's data management system, Don Quijote 2.

6.1 Don Quijote 2

The ATLAS experiment's distributed data management system, called Don Quijote (DQ2) (Branco et al. 2006), is designed to interact with the underlying Grid middleware services to provide a single entry point for ATLAS users requiring access to data and to implement the data flow as described in the ATLAS Computing Model (ATLAS TDR 2005).

The scope of the system encompasses the management of file-based data of all types (event data, conditions data, user-defined file sets containing files of any type). The requirements, design and development of the system draw heavily on the 2004 Data Challenge 2 experience gained with the ATLAS-developed Don Quijote distributed data manager (DQ) and the Grid middleware components underlying it. The other principal input is the experiment's Computing Model, which provides a broad view of the environment and parameters the DQ2 system must support.

6.1.1 Design

The system architecture is based on the grouping of file-based data into datasets, which are collections of files. A set of catalogs store information on the location of datasets, their constituent files and associated system metadata. Data movement is requested at the dataset-level and a distributed set of agents interact with the file-level Grid middleware services described above to control the data movement and cataloging.

6.1.2 Datasets

Datasets are the unit of data movement in ATLAS and typically contain event data sharing some association (temporal, physics selection, processing stage, etc.), but can contain any sort of file-based data. All files managed by DQ2 are constituents of datasets. A file can be a constituent of multiple datasets. Constituent files are identified by logical identifiers (LFN, GUID). Datasets hold a mutability state, which can be open or frozen (permanently locked). Datasets that are not frozen can have files added to them and can be versioned, to support discrete changes in their content tagged by version IDs.

6.1.3 Dataset Catalogs

A set of catalogs store information on the location of datasets, their constituent files and associated system metadata. These catalogs and their interactions are shown in Figure 8.

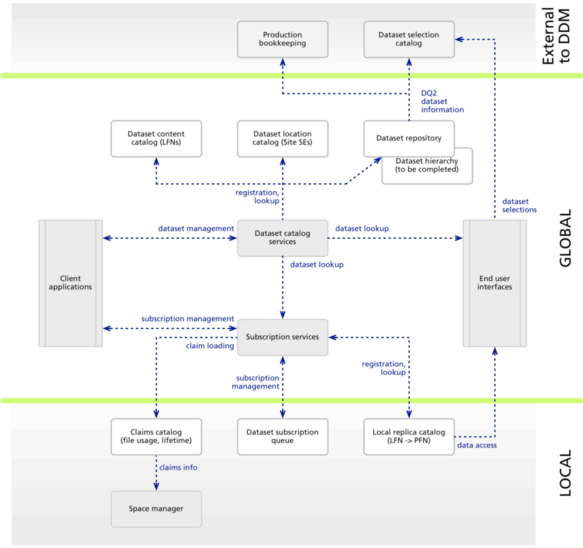

Figure 8: The DQ2 dataset catalogs and their interactions.

The dataset repository is a catalogue of datasets. Each dataset is represented by one entry in the catalogue, identified by a unique ID and a name string (also unique). The repository catalogues DQ2 system metadata for datasets, and all dataset versions. It serves as the principal catalogue and lookup source for datasets defined and made available by DQ2.

The dataset selection catalogue is the catalogue by which physics selections are made to identify datasets of interest. It is not a part of DQ2, but receives information from DQ2 (the dataset repository) on file-based datasets and associated metadata.

The dataset content catalogue records the logical file constituents of datasets. This catalog records all (logical) files managed by DQ2, with global scope, so scalability is a major issue. Organisation of files into datasets produces a manageable system. Physical file locations at a site are available only at the site (SE) level, which is the level at which this information is managed and relevant, via the local file catalogue.

The data location catalog provides lookup of the site locations at which copies of datasets can be found. Once the site on which a dataset resides is known, the local file catalog at the site is contacted to resolve the SURL corresponding to the physical file location.

The data subscription service enables users and sites to obtain data updates in an automated way via 'subscriptions' to mutable datasets, as described in the next section. The subscription catalog stores subscriptions of datasets to sites.

6.1.4 Data Movement

All managed data movement in the system is automated using the subscription system. The idea is that a site subscribes to a dataset, and DQ2 services resident at the site act to keep the site's copy of the dataset up to date with respect to any changes that might be made to the dataset over time. Data movement is triggered from the destination side, such that local uploading can be done using site-specific mechanisms if desired, with no requirement that other sites be aware of these specialised mechanisms. A number of independent entities per site are involved in the data movement process, interacting with Grid middleware services to perform the principal steps of the movement process: fetching of the set of logical files to be transferred on the basis of the dataset content minus any files already present locally; replica resolution to find the available file replicas via the dataset location catalogue, content catalogue and local replica catalogues; allocation of the transfers across available sources according to policy; bulk reliable file transfer; and file- and dataset-level verification of the transfer.

Deployment The data movement agents are deployed to the same level as LFC and FTS services, i.e., one set of agents at CERN (the Tier-0 centre) and one at each of 10 Tier-1 centres that handle ATLAS data. The ATLAS computing model assumes a hierarchical style of data movement, with direct data transfer from CERN only to the Tier-1 centres and from each Tier-1 centre to a 'cloud' of associated (usually geographically) Tier-2 centres. The agents running at each Tier-1 centre manage data flow within this cloud of associated Tier-2 centres and pulls data to the Tier-1 from CERN and other Tier-1 centres. Locating the agents within the same site as the Grid middleware services reduces greatly the penalty of communication with them, especially the LFCs, which are queried very frequently.

6.1.5 Experience

DQ2 has been heavily tested in the framework of the LCG Service Challenges, which has provided a means of testing both the scalability of DQ2, and also testing of the integration with Grid middleware services. For these challenges a 'Tier-0 exercise' was run – a scaled down practice run of the data movement that will flow from CERN when the detector starts operation. The exercise consisted of generating fake raw data at CERN, processing the data at CERN and shipping out the raw and reconstructed data to Tier-1 centres and then to Tier-2 centres using DQ2. At operational level, the data throughput from CERN to all Tier-1 centres is 780MB/s and each Tier-1 centre ships 20MB/s of data shared among its associated Tier-2 centres. These exercises have proved very useful and have led to large improvements in DQ2, especially in the areas of monitoring of data transfer and the stability of the agents. At the time of writing the services run themselves with very little manual intervention and are moving data at an average rate of 1 PetaByte per month.

7 Conclusions

Data management has evolved rapidly within EGEE in the last few years. New services have been introduced and improvements in existing services have been made.

As the time when the LHC gathers data approaches, emphasis is now on service stability rather than adding new features. Frequently problems and weaknesses which do not arise in testing are exposed when services are used across the 200 EGEE sites. At the time of writing each of the LHC experiments are undertaking major Physics Data Challenges – these are difficult, but essential, tests of the end to end functionality of the grid systems involved.

However, the very difficulties involved in providing data management at such a scale have also ensured that the development of genuinely robust and performant grid level middleware is well on its way to being completed.

References

Allcock (Ed.), (2003), GridFTP: Protocol Extensions to FTP for the Grid. Global Grid Forum, GFD.20.

ATLAS Collaboration (2005), ATLAS Computing Technical Design Report, CERN.

Baud, J-P. & Casey, J. (2004), Evolution of LCG-2 Data Management. In *Computing In High Energy Physics Proceedings* (CHEP 04), Interlaken, Switzerland.

Baud, J-P., Casey, J., Lemaitre, S., Nicholson, C., Smith, D. & Stewart, G. (2005), LCG Data Management: From EDG to EGEE. In *UK eScience All Hands Meeting Proceedings*, Nottingham, UK.

Bird, I., Hess, B., Kowalski, A., Pertavik, D., Wellner, R., Sim, A., Shoshani, A. (2001), Common Storage Resource Manager Operations. Available from SRM website, http://sdm.lbl.gov/srm-wg/doc/srm.v1.0.pdf.

Burke, Stephen et al. (2004), HEP Applications Experience With The European Datagrid Middleware And Testbed, In Computing In High Energy Physics Proceedings (CHEP 04), Interlaken, Switzerland.

Branco, M., Cameron, D. & Wenaus, T. (2006), A Scalable Distributed Data Management System for ATLAS. In *Computing In High Energy Physics Proceedings* (CHEP 06), Mumbai, India.

CASTOR Web Site, http://castor.web.cern.ch/.

Cowan, G. A., Ferguson, J. K. & Stewart, G. A. (2006), Optimisation of Grid Enabled Storage at Small Sites, In *UK eScience All Hands Meeting Proceedings*, Nottingham, UK.

de Riese, M., Fuhrmann, P., Mkrtchyan, T., Ernst, M., Kulyavtsev, A., Podstavkov, V., Radicke, M. & Sharma, N. (2006)dCache, the Book. Available from dCache website, http://www.dcache.org/manuals/Book/.

dCache Web Site, http://www.dcache.org/.

EU EGEE Web Site, http://www.eu-egee.org/.

Ferrari, T., Bencivenni, M., De Girolamo, D., Zani, S. & Hirstius, A. (2006), TCP performance optimization for 10 Gb/s LHCOPN connections. Presented at HEPiX Spring Meeting, CASPUR, Rome.

gLite Middleware Web Site, http://www.glite.org/.

GridPP Project Web Site, http://www.gridpp.ac.uk/.

Lemaitre, S. & Baud, J-P. (2006), DPM Administration for Tier2s in CERN WLCG Tier-2 Workshop, CERN, Geneve. Available from http://indico.cern.ch/conferenceTimeTable.py ?confId=a058483.

Lo Presti, G., Cancio, G. & Ponce, S., (2006), Architecture Overview. Presentation during CASTOR 2 review at CERN.

Mandrichenko, I. (2003), GridFTP Protocol Improvements. Global Grid Forum, GWD-E-21.

NorduGrid Web Site, http://www.nordugrid.org/.

Open Science Grid Web Site, http://www.opensciencegrid.org/.

Ponce, S. (2006), CASTOR status and Overview. Presented at HEPiX Spring Meeting, CASPUR, Rome.

RFC 959, (1985), File Transfer Protocol. The Internet Society.

RFC 2228, (1997), FTP Security Extensions. The Internet Society.

RFC 2389, (1998), Feature negotiation mechanism for the File Transfer Protocol. The Internet Society.

Storage Resource Manager Working Group Web Site, http://sdm.lbl.gov/srm-wg/.

Stewart, G. (2006), The High Energy Data Pump: Software. Presented at HEPiX Spring Meeting, CASPUR, Rome.

Xu, L., Harfoush, K. & Rhee, I. (2004), Binary Increase Congestion Control for Fast, Long Distance Networks. IEEE INFOCOM, Hong Kong.