Journal of Physics: Conference Series

Using Xrootd to Federate Regional Storage

To cite this article: L Bauerdick *et al* 2012 *J. Phys.: Conf. Ser.* **396** 042009

Using Xrootd to Federate Regional Storage

L Bauerdick[1]; D Benjamin[2]; K Bloom[3]; B Bockelman[3]; D Bradley[4], S Dasu[4], M Ernst[5], R Gardner[6], A Hanushevsky[7], H Ito[5], D Lesny[8], P McGuigan[9], S McKee[10], O Rind[5], H Severini[11], I Sfiligoi[12], M Tadel[12], I Vukotic[6], S Williams[13], F Würthwein[12], A Yagil[12], W Yang[7]

[1]FNAL; [2]Duke U.; [3]U.Nebraska-Lincoln; [4]U.Wisconsin-Madison; [5]BNL; [6]U.Chicago; [7]SLAC; [8]U.Illinois at Urbana-Champaign; [9]U.Texas-Arlington; [10]U.Michigan; [11]U.Oklahoma; [12]UCSD; [13]Indiana U.

E-mail: bbockelm@cse.unl.edu, rwg@hep.uchicago.edu

Abstract. While the LHC data movement systems have demonstrated the ability to move data at the necessary throughput, we have identified two weaknesses: the latency for physicists to access data and the complexity of the tools involved. To address these, both ATLAS and CMS have begun to federate regional storage systems using Xrootd. Xrootd, referring to a protocol and implementation, allows us to provide data access to all disk-resident data from a single virtual endpoint. This "redirector" discovers the actual location of the data and redirects the client to the appropriate site. The approach is particularly advantageous since typically the redirection requires much less than 500 milliseconds and the Xrootd client is conveniently built into LHC physicists' analysis tools. Currently, there are three regional storage federations - a US ATLAS region, a European CMS region, and a US CMS region. The US ATLAS and US CMS regions include their respective Tier 1, Tier 2 and some Tier 3 facilities; a large percentage of experimental data is available via the federation. Additionally, US ATLAS has begun studying low-latency regional federations of close-by sites. From the base idea of federating storage behind an endpoint, the implementations and use cases diverge. The CMS software framework is capable of efficiently processing data over high-latency links, so using the remote site directly is comparable to accessing local data. The ATLAS processing model allows a broad spectrum of user applications with varying degrees of performance with regard to latency; a particular focus has been optimizing n-tuple analysis. Both VOs use GSI security. ATLAS has developed a mapping of VOMS roles to specific file system authorizations, while CMS has developed callouts to the site's mapping service. Each federation presents a global namespace to users. For ATLAS, the global-to-local mapping is based on a heuristic-based lookup from the site's local file catalog, while CMS does the mapping based on translations given in a configuration file. We will also cover the latest usage statistics and interesting use cases that have developed over the previous 18 months.

1. Introduction

Over the last 2 years of operations at the Large Hadron Collider (LHC), the LHC experiments' computing projects and the Worldwide LHC Computing Grid (WLCG) [1] have demonstrated they can satisfy the experiments' needs. A key part of the WLCG is the ability to move data between sites and deliver data to end-user applications. However, this short operational experience has also begun to demonstrate weaknesses in the computing models; we focus on two in particular:

- the latency for physicists to access data, and
- the complexity of the tools involved.

In the WLCG context, the mantra has always been "jobs go to data"; that is, data can only be accessed from a batch system job submitted through the grid to sites hosting the data. This rule *scales up* well to the number of concurrently running jobs throughout the global grid, but does not *scale down* in terms of usability for the physicists doing analysis. To access a single byte of data in this model involves writing a simple analysis, wrapping it into an analysis job (using experiment-provided tools), submitting it into a queue, and waiting for results. In the best case, a single byte takes about 15 minutes to access. Further, if there is a storage issue at the site resulting in the unavailability of a single byte needed for a job, it will crash and have to be rerun by the user.

If a user wants to copy the data to their local laptop, they must mentally leave their "analysis environment" and utilize a different set of tools for transferring data provided by either the experiment's computing collaboration or the grid middleware client providers. The quality of the data transfer tools varies by experiment, but often exposes users to unnecessary details such as file location or transient errors from the grid layer. It takes the user minutes, at best, to locate and download the file in question, even if they want only a single byte from it.

To solve these two issues, the CMS and ATLAS computing organizations have started building federated storage systems utilizing the Xrootd software suite [2]. We define a **federated storage system** to be a collection of disparate storage resources managed by cooperating but independent administrative domains transparently accessible via a common namespace. These are built using one or more dedicated Xrootd servers at each site and a virtually centralized *redirector*; the user contacts the central endpoint directly and is redirected to the site that can currently serve the data. This allows for data access regardless of the job's location - not only breaking the old mantra "jobs go to data," but also reducing the latency for data access. As Xrootd has a client included with almost all distributions of the ROOT framework, every single LHC analysis environment is already able to access the federations - removing the need for a user to "context switch" between local analysis and distributed/Grid analysis tools.

2. Federated Regional Storage Using Xrootd

Figure 1 outlines the basic unit functionality in the Xrootd system, the client redirection for reads (the federations described here are read-only). A client contacts a centralized endpoint with a request to open a file. This Xrootd server, called a *redirector*, has no storage, but is connected to a local daemon, the Cluster Management Service Daemon (*cmsd*), running in what is called "manager mode". The redirector's *xrootd* daemon queries the *cmsd* for a file's location; if the location is not known in the local memory cache, the *cmsd* will subsequently query all sites subscribed to the redirector. Using an algorithm based on weighted round robin, the redirector will send the client directly to another site [16]. In terms of metadata, the application requests access to the global file name and needs no knowledge of the file's location. Further, the data travels directly from source site to the application, and the application only needs to support the Xrootd protocol despite heterogeneous underlying storage.

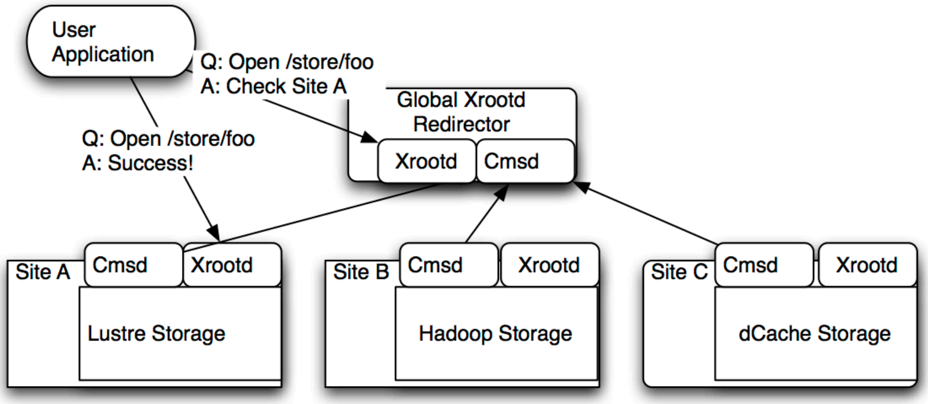


Figure 1. The basic redirection mechanism..

If the site is not a native Xrootd file system, the Xrootd server at the site will act as a proxy between the storage system and the client. The Xrootd server has a plugin architecture, allowing us to translate between the internal, site-specific protocol and the Xrootd protocol. For example, plugins exist for acting as a proxy for the dCap protocol [14], Hadoop Distributed File System [15], and Xrootd itself. We have developed a plugin for each storage technology accessible to us (in general, all that is needed is a thread-safe C-API to the site's storage); utilizing the existing site storage system is a requirement for a federation given the diversity of technologies used in the WLCG. As an optimization, if the source site natively implements the Xrootd protocol, the user application may be redirected directly to the server storing the data instead of going through a proxy.

Within the federation, a global namespace is used, with the global-to-local translation performed within the Xrootd server at the boundary between site and federation. Just as a plugin is used to proxy between the Xrootd and local data transfer protocols, there is a global-to-local namespace mapping plugin called the Name-to-Name (N2N). This allows the global namespace to be formed, essential for file location transparency to the user. Each experiment has implemented custom plugins; these are described in Section 3. The federations rely on any namespace organization and consistency issues to be handled within the experiment layer and not within Xrootd.

A federation provides a powerful mechanism for external users to interact with the site and accordingly needs management to prevent abuses from users. As a start toward proper resource management, we are interested in throttling the following quantities:

1. **Namespace queries**: If the location is not previously cached, a user's query at the redirector results in namespace activity at all the regional sites. Xrootd provides query throttling and queuing to prevent this.
2. **Bandwidth utilized**: This is not possible to throttle from within the Xrootd daemon, but sites are encouraged to add hardware- or network-based throttles, or QoS limits. Sites typically exercise the crude option of throttling rates based on the number of gigabit Ethernet interfaces deployed in the site's systems.
3. **Operations per second**: No limits currently exist for the number of I/O operations per second (IOPS).

Besides internal safeguards, abuse detection is available through cross-site monitoring and accounting. Xrootd has built-in monitoring which can send periodic UDP packets to a central host. The *summary monitoring* describes the overall activity of the host (number of redirections, login attempts, bytes transferred, CPU used, etc.) while the *detailed monitoring* packets report on perconnection and per-open file activity (i.e. each client/server interaction is logged). The monitoring work is the topic of a separate paper [6].

Authorization and authentication is handled within the Xrootd server, which has a modular plugin for both activities. Authentication is required to access data, although the redirectors to date allow unauthenticated access for file lookups. In the case of an authentication failure at a data server, the Xrootd client will return back to the original redirector and request a different source. The authorization for CMS and ATLAS is relatively simple - any experiment-owned data is readable by all members of the experiment's collaboration. Only personal data might have more restrictive read access. Authorization is assumed to be consistent; that is, a user should encounter the same permissions on file */store/foo* regardless of where it is located. It is currently up to the experiment to synchronize the authorization policy across all its sites; in [10], ALICE cleverly solves this issue by having the client carry along a centrally signed authorization for specific files. Neither CMS nor ATLAS have such a central signing authority; while we monitor for issues, we are at the mercy of the site's ability to implement correct policies.

Caching techniques are being explored utilizing Xrootd's File Residency Manager (FRM). This feature is particularly useful for Tier 3 centers to replace missing files or to automate downloads of data sets exported by sites in the federation. The figure below illustrates the workflow in the context of the Tier 3 site co-located with a Tier 1 center at Brookhaven National Lab (BNL). A local file request from a Tier 3 client is made to the Tier 3's redirector that queries the set of Tier 3 data servers for the file. If none answer one will invoke the FRM service that will query a global redirector for the file. If the file is available at some site within the federation, a transfer is initiated and the file is copied to the Tier 3 data server. This data server in turn serves the data to the client. This method has proved to be a powerful means of quickly transferring data sets to Tier 3 sites with a minimum of the usual data management required. This is illustrated in **Figure 2**.

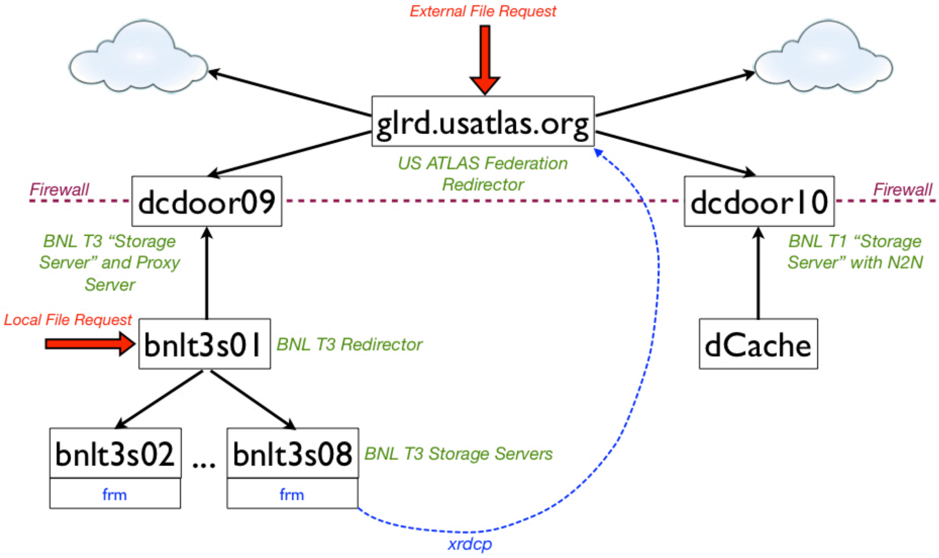


Figure 2. File caching implementation at BNL. In this case we use the Xrootd File Residency Manager (FRM) to manage the cache contents and the federation as a data source.

3. Implementation Details of CMS and ATLAS Federations

This section covers the implementation details for the CMS and ATLAS federations, focusing on how they can differ in computing models while utilizing the same Xrootd software through use of modular plugins.

3.1. Any Data, Any Time, Anywhere

The CMS infrastructure is named after the project funded to operate and develop it - "Any Data, Any Time, Anywhere" (AAA). This project started as an informal collaboration between multiple US Tier 2 Hadoop-based sites in 2010 as a mechanism to export data to users located outside their clusters. It grew quickly to four sites, and the idea of cross-site sharing was added. In late 2010 and early 2011, these sites outlined the basic technical mechanisms and worked to make sure requisite patches were added to the CMS application software to decrease sensitivity to latency. We estimate that the average CMSSW performance hit for accessing a file across the US is about 10% of wall time. This figure is based on spot-checks and informal performance tests, and needs rigorous study. Since late 2011, the project has been formally funded and adding the operational practices and monitoring necessary to classify it as a production service.

The AAA infrastructure currently consists of 5 Hadoop sites, 2 dCache sites, and a Lustre site; overall, it covers about 15PB of disk. The order of magnitude of daily usage is tens of TB, 5-15 different destination networks, and 10-20 unique users (the set of users changes daily).

CMS sites use two plugins in addition to the storage system plugin. The *XrdCmsTfc* module is a Name-to-Name (N2N) plugin that translates between the internal storage filename and the filename exported to the Xrootd server. This performs the translation using a list of regular-expression-based rules for mapping between local and global stored in an XML file. Maintained by each site for their own storage, CMS refers to this list as the "Trivial File Catalog". This provides a fast, deterministic mapping of all global file names to site file names relying on no external components. As an example, using this module, when a user requests Xrootd open a file named */store/foo*, the Xrootd server may request the file */mnt/hadoop/cms/store/foo* from an underlying HDFS storage. The TFC file is utilized throughout the CMS data management system, making this N2N module CMS-specific.

For authorization, CMS utilizes GSI/X509 for authentication and the LCMAPS framework for authorization. LCMAPS is a flexible, policy-based library with excellent C bindings used by both OSG and EGI for performing authorization decisions. We created the *XrdLcmaps* module for the GSI security plugin in Xrootd. The default configuration is to do a remote procedure call to the local site authorization service and map the X509 credentials to a Unix user account, but any configuration possible on the OSG is valid. The Unix user account is passed to the default Xrootd authorization layer that will do file and directory access checks based on a separate configuration file.

Monitoring and accounting is performed using a combination of:

- **MonALISA**: provides visualization of the summary monitoring data [4].
- **GLED**: provides analysis and visualization of the detailed monitoring data [3].
- **Nagios**: periodically runs basic functionality tests on each endpoint and verifies each site's data is available through the federation.
- **Gratia**: records basic information for each file transfer into a database for later accounting [8].

Figure 3 and **Figure 4** show sample graphs from CMS monitoring. **Table 1** shows a few basic statistics from the Gratia accounting database showing different ways to calculate the "average" job data rate - an important number to know when making capacity planning decisions.

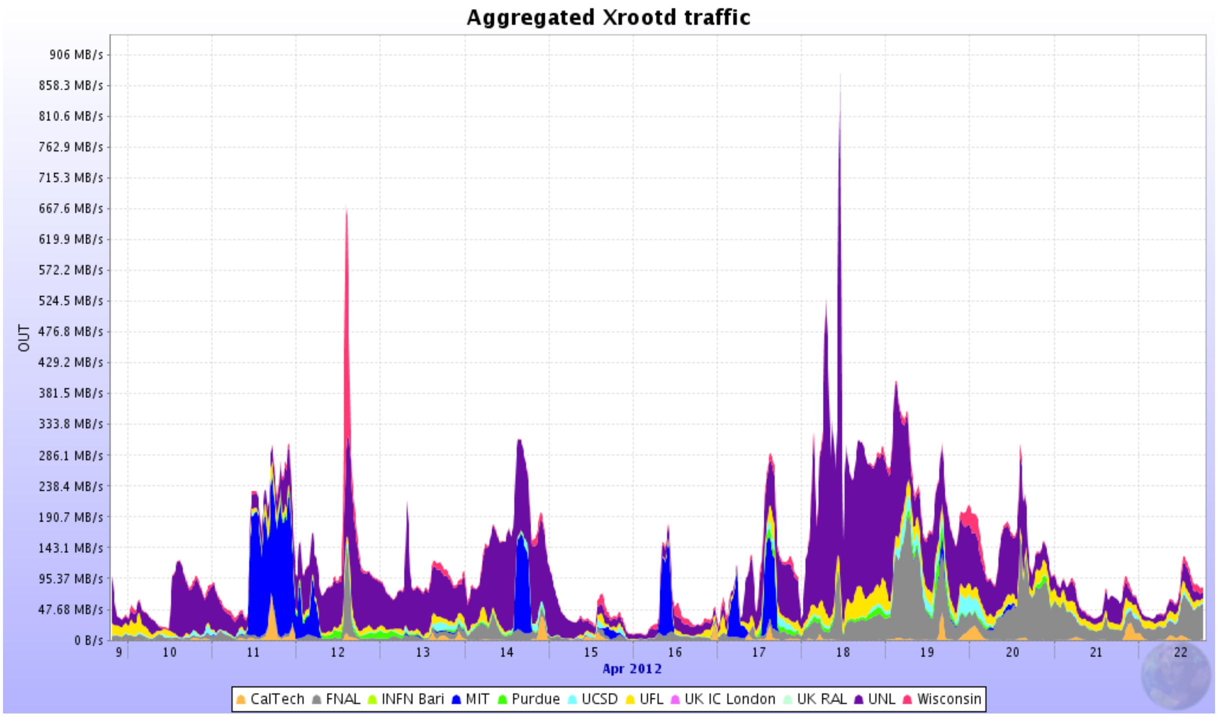


Figure 3. A monitoring graph generated by MonALISA showing aggregate client read rates (MB/sec) from AAA, broken down by source site. MonALISA also records other summary statistics, such as the number of I/O operations per second.

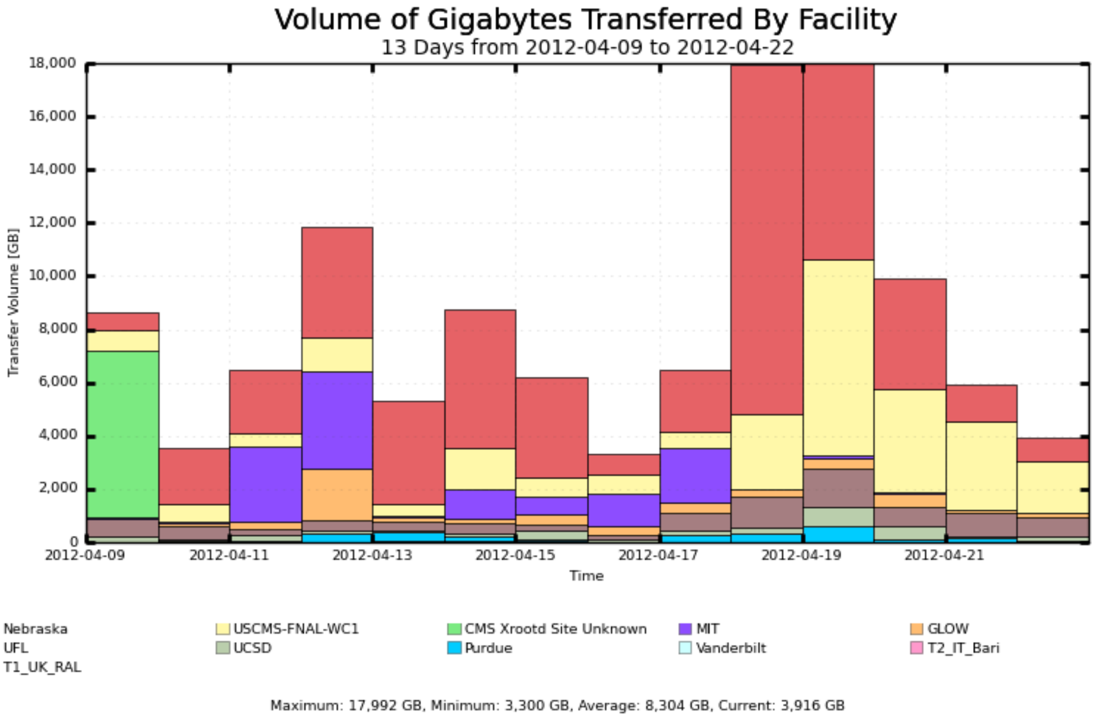


Figure 4. An accounting graph generated by Gratia showing aggregate client read volumes for AAA, broken down by source site. One can filter and display graphs based on user name or destination sites.

Method	Minimum Transfer Length (minutes)	Number of transfers	Rate (KB/s)
Average of rates	5	618,118	648
Average of rates	15	484,985	457
Rate of sums	5	618,631	223
Rate of sums	15	485,240	201

Table 1. Transfer averages from AAA accounting over a 30-day period in April 2012, filtered on a minimum transfer length to remove crashed jobs and monitoring jobs. The "rate of sums" is calculated by sum(transfer volume)/sum(transfer duration), where the sum is taken over all recorded transfers. The "average of rates" is the average of (transfer volume)/(transfer duration) for each transfer.

3.2. ATLAS Implementation: FAX

The ATLAS infrastructure, called "Federated ATLAS Xrootd" (FAX), began as an R&D project within the ATLAS Distributed Computing program. A prototype infrastructure has been deployed at the Tier 1 center at Brookhaven, at each US Tier 2 center, and a number of Tier 3 sites. The service implements an ATLAS global namespace and provides three redirectors: a "global" redirector hosted at Brookhaven Lab which provides a single point of contact to all the ATLAS experimental data on disk in the US using the Xrootd protocol; a Midwest U.S. redirector (hosted at the University of Chicago) which connects the storage facilities of the Midwest and Great Lakes regional Tier 2s (all within 5 milliseconds one-way latency) and the Oklahoma University Tier 2 site (10 milliseconds one-way latency); and a redirector dedicated to Tier 3 sites in which operational expectations are relaxed relative to the Tier 1 and Tier 2's which are bound by WLCG MOU requirements. Additionally western and eastern regional redirectors are planned to group nearby sites pending on-going I/O performance and redirection tests. There are also future plans to setup an ATLAS redirector at CERN and to extend the federation development effort to European sites. The current FAX infrastructure is illustrated in **Figure 5**.

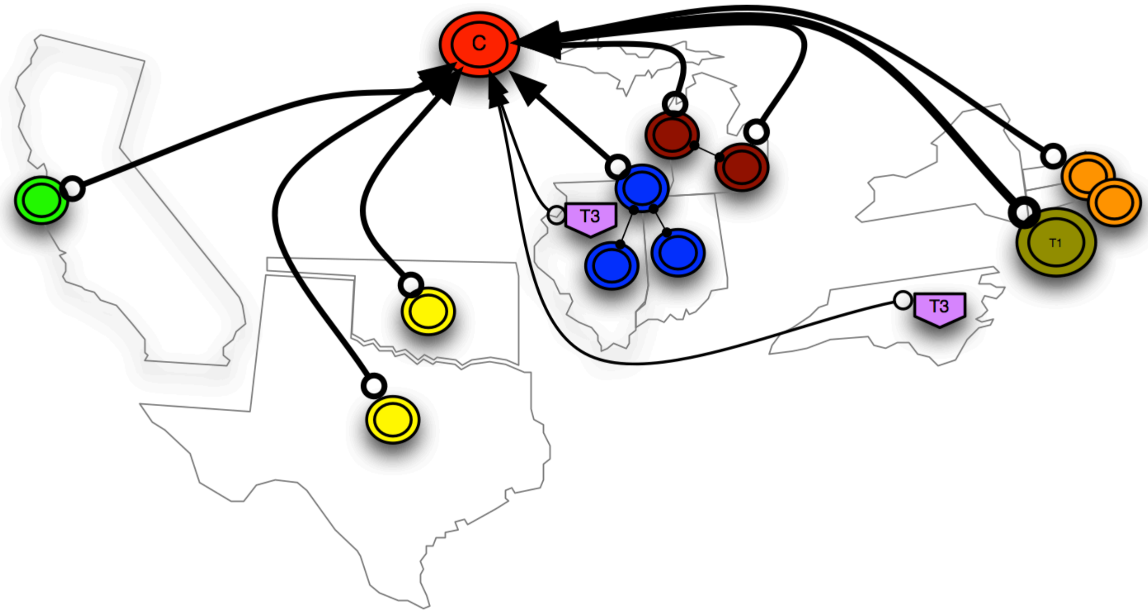


Figure 5. Schematic map of federated sites in US ATLAS as seen from an analysis client ("C") directly accessing datasets in the ATLAS global namespace. Shown are Tier 1, 2 and Tier 3 centers participating in the prototype deployment.

The Tier 1 center uses dCache for the backend storage, as do two of the Tier 2's. As the remaining three Tier 2's use Xrootd, GPFS or Lustre file systems, the utility of Xrootd as a federation technology was seen as particularly advantageous for federating heterogeneous storage services as this allowed sites to quickly federate their data without making any significant infrastructure changes. It also allows sites to leave a federation without impacting the local processing.

The Tier 2 sites at SLAC and the University of Texas at Arlington both use Xrootd as their backend storage system and have experienced trouble free operation since their initial deployments in 2006. Federating these sites is accomplished with an Xrootd proxy service that reports to the global redirector.

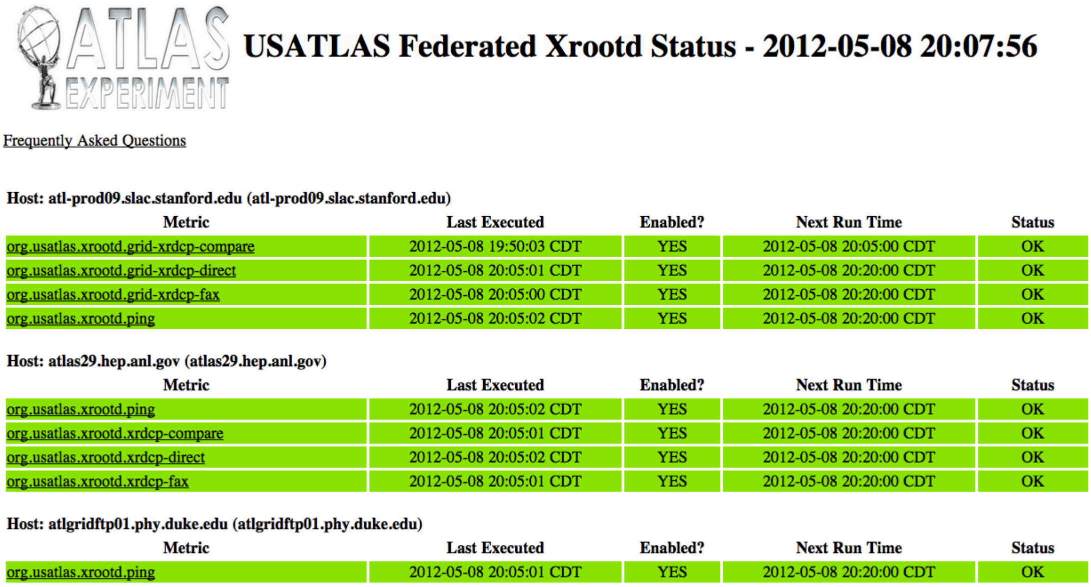
The ATLAS Great Lakes Tier-2 (AGLT2) uses Xrootd on top of dCache and provides 2.2PB between its two sites at Michigan State University and the University of Michigan. Currently, this system is federated using an Xrootd proxy service, talking to a dCap door on the backend. This provides a simple installation on a single server but requires all client-sever traffic to pass through this server. Work is on going to use the dCache native Xrootd door with a name-to-name mapping plugin developed for dCache. The Midwest Tier 2 center (MWT2, comprised of the University of Chicago, Indiana University and the University of Illinois at Urbana-Champaign) exports the dCache namespace as an overlay, discussed below. The local redirectors at AGLT2 and MWT2 are each configured to report to a regional redirector so that nearly 5 PB of ATLAS data on disk can be read from either site with good performance. Tier 3's in the region can access datasets using only a single URL, the global file name, and an appropriate GSI proxy. This regional director is linked to the FAX global director.

3.2.1. FAX monitoring

A status monitor has been built to continuously check and display the availability of the endpoint servers in the federation. The system leverages the Resource and Service Validation (RSV) framework [13], originally developed by OSG to probe the status of OSG services. A set of Xrootd-specific probes were developed to:

- **Ping** the availability of the redirector server at each site
- **Copy** a pre-placed test file from the site using *xrdcp*.
- **Check redirection** from the global redirector by requesting files unique to a site.
- **Perform a simple comparison check** of the files copied directly or via the global redirector

A web-display is used to report the status every 15 minutes and to provide detailed logging information when failures occur. **Figure 6** shows the status monitor view for a few sites in the FAX system.



USATLAS Federated Xrootd Status - 2012-05-08 20:07:56

Frequently Asked Questions

Host: atl-prod09.slac.stanford.edu (atl-prod09.slac.stanford.edu)

Metric	Last Executed	Enabled?	Next Run Time	Status
org.usatlas.xrootd.grid-xrdcp-compare	2012-05-08 19:50:03 CDT	YES	2012-05-08 20:05:00 CDT	OK
org.usatlas.xrootd.grid-xrdcp-direct	2012-05-08 20:05:01 CDT	YES	2012-05-08 20:20:00 CDT	OK
org.usatlas.xrootd.grid-xrdcp-fax	2012-05-08 20:05:00 CDT	YES	2012-05-08 20:20:00 CDT	OK
org.usatlas.xrootd.ping	2012-05-08 20:05:02 CDT	YES	2012-05-08 20:20:00 CDT	OK

Host: atlas29.hep.anl.gov (atlas29.hep.anl.gov)

Metric	Last Executed	Enabled?	Next Run Time	Status
org.usatlas.xrootd.ping	2012-05-08 20:05:02 CDT	YES	2012-05-08 20:20:00 CDT	OK
org.usatlas.xrootd.xrdcp-compare	2012-05-08 20:05:01 CDT	YES	2012-05-08 20:20:00 CDT	OK
org.usatlas.xrootd.xrdcp-direct	2012-05-08 20:05:02 CDT	YES	2012-05-08 20:20:00 CDT	OK
org.usatlas.xrootd.xrdcp-fax	2012-05-08 20:05:01 CDT	YES	2012-05-08 20:20:00 CDT	OK

Host: atlgridftp01.phy.duke.edu (atlgridftp01.phy.duke.edu)

Metric	Last Executed	Enabled?	Next Run Time	Status
org.usatlas.xrootd.ping	2012-05-08 20:05:01 CDT	YES	2012-05-08 20:20:00 CDT	OK

Figure 6. FAX status-monitoring page reporting status of participating sites during 15-minute intervals (a subset of sites are shown).

FAX additionally has leveraged monitoring developed for AAA. A MonALISA-based collection repository and web server have been deployed at SLAC to give a visual rendering of FAX-related traffic across sites, as shown in **Figure 7**.

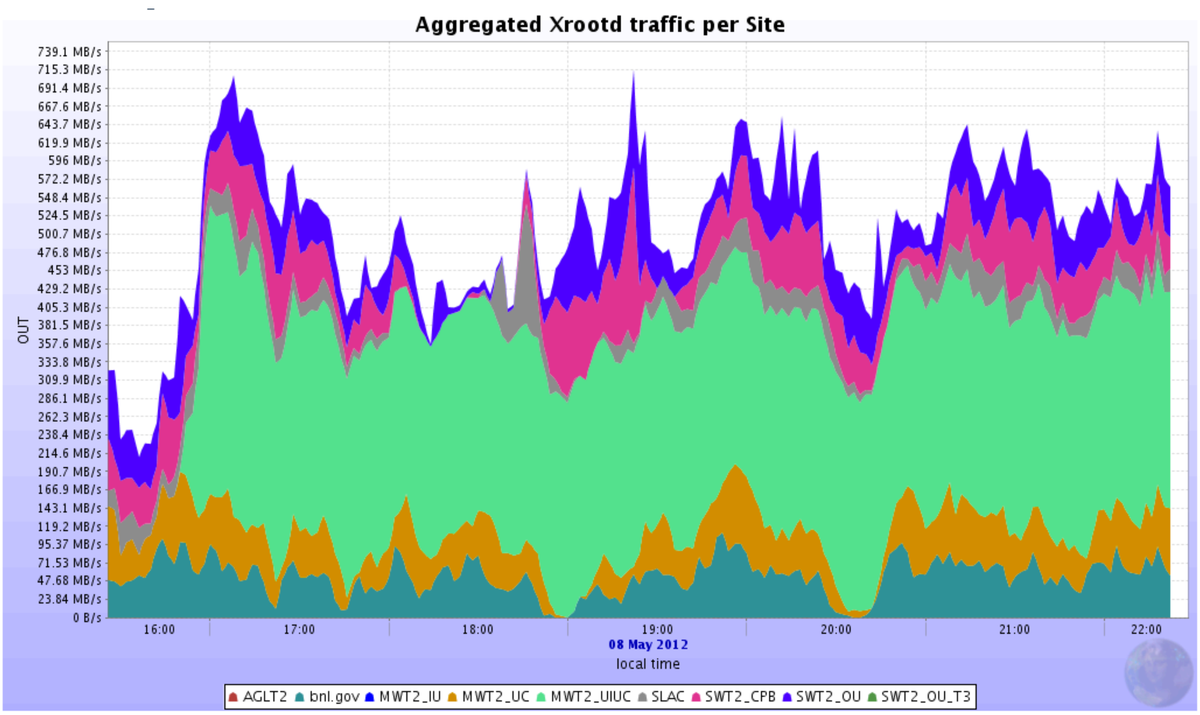


Figure 7. Traffic through the FAX system during commissioning tests of the infrastructure. Eight client sites were running up to 50 parallel jobs each copying the file using *xrdcp* and also reading it directly using a simple ROOT script.

3.2.2. Translating Global Namespace to Local Paths

As discussed in Section 2, a global namespace convention is necessary for a federation. This had to be established within ATLAS before federation work could begin. The convention developed is based on the path names recorded in the LCG File Catalog (LFC) database [7]. The LFC translates ATLAS logical file names to the physical path names at the site. For Tier 3 sites, the physical path name is usually formed by adding a common prefix to the logical file name in the global namespace. This allows the Tier 3 sites to be easily federated. Tier 1 and Tier 2 sites store files with more complicated, non-deterministic conventions, so the LFC must be consulted to convert the global name into a physical path name on a file-by-file basis. FAX has implemented an Xrootd plugin, *xrd-lfc*, to handle this translation. As Xrootd requires these translations to be fast, and the LFC database lookup is potentially expensive, lookup results are aggressively cached (caching also reduces the load on the LFC). By default, a cache entry lives for 2 hours and the cache can have up to 500,000 entries.

While the dCache storage system natively implements the Xrootd protocol, a special bypass exists. As a dCache pool stores complete files on a well-known directory on disk, we can run the SLAC Xrootd server on the same host and use the *xrd-lfc* plugin to translate from global file name to the name of the file stored on disk. This allows the SLAC Xrootd server to serve the dCache files directly from disk. Here, *xrd-lfc* is translating the global namespace to the local host namespace; the default mode translates the global namespace to the dCache namespace. We refer to this mode of operation as “Xrootd overlay”. It is in production at MWT2 for FAX, and a similar approach is taken to integrate FNAL dCache with the AAA infrastructure. Compared to the proxy-based approach, this direct approach has improved scalability. Compared to using the dCache Xrootd native implementation, the overlay avoids having the federation interact with the dCache namespace and integrates with the deployed monitoring.

3.2.3. Wide area performance and I/O tuning

We have investigated the anticipated performance of ATLAS analysis codes using FAX by measuring file copy and read times for various client-server combinations, percent of file read, and for various I/O options [11]. As the CMS codebase is significantly different, the results in this section are not comparable across experiments.

Figure 8 illustrates one such study of data transfer performance from the Midwest Tier 2 Center (MWT2) to clients in the region with the following latencies:

1. **MWT2**: 0-5 msec. Data access via dCap protocol, dCache native Xrootd, and the Xrootd overlay were studied.
2. **AGLT2**: 3-4 msec.
3. **OU_OCHEP_SWT2**: 10 msec.

The results for MWT2 and AGLT2 reflect averages over multiple client locations, as these "sites" are federations unto themselves. The results suggest the following:

- Clients at AGLT2 can expect to take 25% longer if the data were at MWT2 rather than AGLT2. This is quite reasonable for the fallback use-case.
- Native dCap, dCache native Xrootd, and Xrootd overlay yield similar performance local or over the WAN for a given client location.
- Overall dCap averaged slightly faster for local transfers.
- dCap is more sensitive to wide area latency than dCache-Xrootd or the Xrootd overlay.

Note the metric here is the time required to read a 750 MB ATLAS file, in the format commonly used for analysis (D3PD). Results are obtained using ROOT 5.30 with a 30MB pre-fetch cache (see [12] for a discussion of ROOT I/O tuning). Overall the test indicates reasonable performance for regional wide area reads.

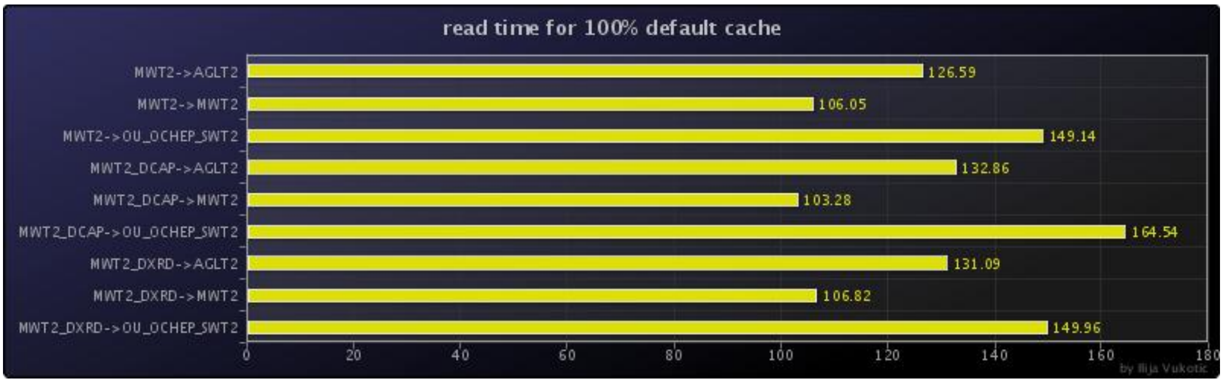


Figure 8. Comparison of average read times for (server → client) tests where the access method is varied. MWT2 is direct access through the SLAC Xrootd server. MWT2_DCAP is access through native dCache's dCap server. MWT2_DXRD is access through the Xrootd overlay discussed in Section 3.2.2.

We additionally studied the feasibility of using very long latency connections. **Figure 9** compares local client performance to two wide area server locations that have significant network latency.

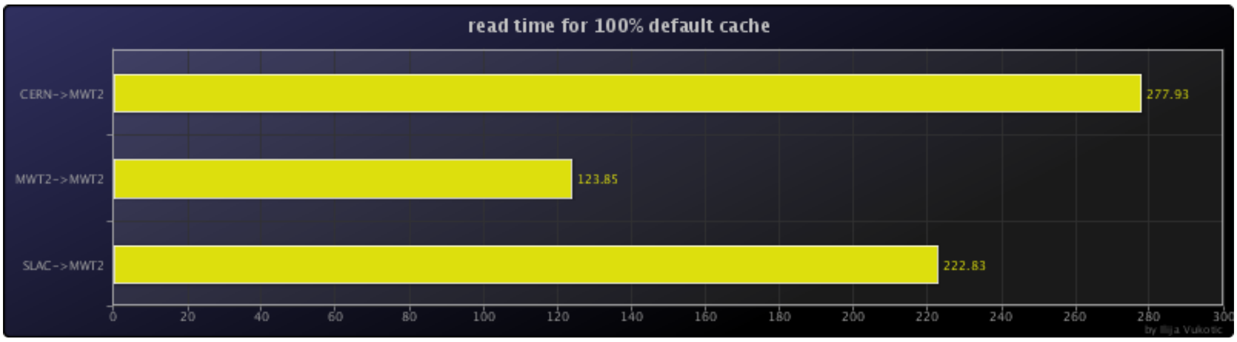


Figure 9. Latency effects for (Server → Client) read times for the combinations CERN → MWT2-Chicago (65 msec), Intra-federation Tier 2 MWT2 → MWT2 (50% local latency, 50% 5 msec), and SLAC → MWT2-Chicago (25 msec).

4. Use Cases for Federations

While we introduce federations as a tool for improving direct interactive access for users, several other use cases have since become apparent. In either AAA or FAX, any respective VO user can directly read files stored on disk in the federation. For CMS, this is currently any file on disk in the US - a powerful draw. Client tools can include *xrdcp* (for downloading an entire file), ROOT (for interactive analysis), or event viewers.

Another common use case is fallback - when a job running at a site is unable to open a file locally, instead of failing the job, simply open the file using the federation instead. In CMS, about 2% of jobs fail due to inaccessible files and could benefit from fallback. In fact, this was the first production use case as we had a good upper limit on the number of failed files, and there was little risk to trying fallback, as the job would have failed regardless.

4.1. Experiment-specific Use Cases

CMS grid jobs have not demonstrated a need for high rates of I/O (averages well less than 1MB/s) and have been carefully architected to not be sensitive to higher latencies. Thus, it is of reasonably low cost to end-users and the infrastructure to stream a job's data over the WAN. CMS has developed the capability to "overflow" user jobs (i.e. sending them to sites where data access is necessarily remote). The overflow is only performed if the following conditions have been met:

1. User job has been in queue for more than 6 hours.
2. The overflow destination has available slots and fallback enabled.
3. The source site is part of the Xrootd federation.
4. The total number of jobs is below a preset limit.

Overflowing jobs helps CMS fully utilize its available computational resources, smoothing out the inefficiencies in data distribution. During the month of April, up to 10% of glideinWMS-based [9] analysis jobs utilized overflow, with occasional peaks over 20% [5].

ATLAS envisions a number of use cases for FAX, among other possibilities:

1. **Missing file replacement.** A job landing at a site may find that one or more of its input files are missing or unreadable. Rather than failing the job and re-brokering to a site with the input data set completely intact, a failover mechanism can be implemented where a FAX redirector can locate the missing data from another site and stage it to disk at the compute site.
2. **Read don't copy.** Many user analysis jobs, such as "skimming" or "slimming" jobs, or those which read group-level or user-produced n-tuple data, read only a small fraction of a data file. A cost function could be developed to select an appropriate redirector for optimal performance

based on server-client locations. The additional access options provided by FAX therefore results in more flexible job scheduling strategies, for example using opportunistic resources which have no locally managed ATLAS storage.

3. **Triggered file caching.** Tier 3 sites can utilize a small disk-based cache similar to the top tier disk storage at SLAC. Input files can be fetched automatically from FAX and be cached at local disk. Along with automatic purging, this provides a management-free (i.e. no catalogs to manage, consistency checking, data deletion) storage system.

5. Conclusions and Future Work

Several difficult problems have been carefully avoided by adding additional requirements into the described infrastructures. We define regions by the maximum network round-trip-time between any two sites in the region because redirection is not network-aware and selecting the "faraway" site has a high performance penalty for the user application. We hope to see the base Xrootd software add hooks for smarter redirection algorithms and clients that can read from multiple sources, and to test these on our data access infrastructures in the future.

From the detailed monitoring stream and the job accounting databases, CMS is starting to better quantify the loss of efficiency expected from doing remote I/O over the WAN. The current expected loss of 10% of CPU efficiency is based on experience and spot-checks, but this has not been thoroughly verified.

Currently, remote diagnostics are lacking in Xrootd. Work has been started with the summary monitoring for authentication failures, but there is no comprehensive framework for monitoring the errors clients receive. Near-failure cases (such as network or disk bottlenecks) are currently only detectable via manual investigation. Synchronizing authorization policies across sites is also a manual exercise, making it difficult to roll out and verify policy.

As Xrootd allows external users to interact with, and hence abuse, local resources, many site administrators are initially hesitant to run Xrootd. Namespace throttling was added to directly express these concerns, and it is possible that IOPS and bandwidth throttling will be necessary in the future.

The sustained transaction rates of the Xrootd redirector need to be carefully studied. When FAX is fully deployed in US, Europe, and perhaps Asia, redirectors at various levels will likely have to handle data discovery requests that are at least an order of magnitude higher than we have ever seen in a local site environment. ATLAS could also benefit from more sophisticated redirection techniques, taking into account time zone, TTL, bandwidth, or downtime. Further, a multi-source client implementation based on active probing would lessen the cost of a sub-optimal redirection decision.

We expect further use cases to evolve in the future; e.g., it is likely CMS will be interested in the namespace healing use case currently envisioned for the FAX infrastructure.

Overall, both US CMS and ATLAS have quickly rolled out a new data access infrastructure over the last 18 months. This infrastructure was started as a means to provide interactive users with improved data access with low latency and integrated clients, but has evolved to include overflow and local site caching. Along with increased usage, the necessary monitoring, accounting, and operations have been put into place to call these federations "production-quality".

6. Acknowledgements

The AAA and FAX systems span many people, locations, and projects. We gratefully acknowledge the support of the Department of Energy (grant DE-FG02-95ER40899) and the National Science

Foundation (grant PHY-1119200, PHY-0612811, PHY-1104549, PHY-1104447, PHY-1104664, PHY-0612805), which partially supported the work described in this paper.

References

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

[2] Dorigo A, Elmer P, Furano F and Hanushevsky A 2005 Xrootd - A highly scalable architecture for data access *WSEAS Transactions on Computers* (2005)

[3] Tadel M 2004 Gled - an Implementation of a hierarchic Server-client Model *Applied Parallel and Distributed Computing (Advances in Computation: Theory and Practice* vol 16) ed Pan Y and Yang L T (Nova Science Publishers) ISBN 1-59454-174-4

[4] Newman H, Legrand I, Galvez P, Voicu R, Cirstoiu C 2004 MonALISA: A Distributed Monitoring Service Architecture *Proceedings of CHEP 2003*

[5] Sfiligoi I, Würthwein F, Bockelman B, Bradley D 2012 Controlling overflowing of data-intensive jobs from oversubscribed sites *Preprint* (submitted to CHEP 2012 proceedings)

[6] Tadel M, et al 2012 Xrootd Monitoring for the CMS experiment *Preprint* (submitted to CHEP 2012 proceedings)

[7] Stewart G, Cameron D, Cowan G, McCance G 2007 Storage and Data Management in EGEE *Proceedings of the fifth Australasian symposium on ACSW frontiers* Volume 68

[8] Canal P 2011 Gratia: New Challenges in Grid Accounting *J. Phys.: Conf. Ser.* **331** 062028

[9] Sfiligoi I, Bradley D C, Holzman B, Mhashilkar P, Padhi S and Würthwein F 2009 The Pilot Way to Grid Resources Using glideinWMS *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol.2, pp.428-432. doi:10.1109/CSIE.2009.950

[10] Feichtinger D, Peters A 2005 Authorization of Data Access in Distributed Storage Systems *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing* (10.1109/GRID.2005.1542739)

[11] Vukotic I 2012 ATLAS WAN IO tests *Available at:* http://ivukotic.web.cern.ch/ivukotic/WAN/index.asp

[12] Canal P, Bockelman B, Brun R 2011 ROOT I/O: The Fast and Furious *J. Phys.: Conf. Ser.* **331** 042005

[13] Quick R. et al. 2009 RSV: OSG Fabric Monitoring and Interoperation with WLCG Monitoring Systems *Presented at CHEP 2009 (Computing in High Energy and Nuclear Physics)*

[14] dCache documentation, http://www.dcache.org/manuals/publications.shtml

[15] Shvachko K, et al 2010 The Hadoop Distributed File System *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies* 0(5), 0-10

[16] Xrootd documentation, http://xrootd.org/docs.html