CITI Technical Report 05–3

GridNFS

Global Storage for Global Collaborations

Peter Honeyman
honey@citi.umich.edu

W. Andros (Andy) Adamson
andros@citi.umich.edu

Shawn McKee
smckee@umich.edu

ABSTRACT

GridNFS combines the NFSv4 protocol and a collection of supporting middleware services configured to run in a Globus environment. GridNFS provides a file system name space that spans a virtual organization, security that meshes with Globus, fine-grained access control lists to support virtual organization groups and users, and secure file system access for jobs scheduled in an indeterminate future.

By combining and integrating standard Internet protocols, GridNFS remains fully compatible with standards-compliant desktop and enterprise network services. Furthermore, GridNFS middleware enhances those environments with global naming and facile identity representation for agile access control across security domains.

May 17, 2005

GridNFS: Global Storage for Global Collaborations

Peter Honeyman
honey@citi.umich.edu

W. Andros (Andy) Adamson
andros@citi.umich.edu

Shawn McKee
smckee@umich.edu

1. Storage requirements for global collaborations

Grid technologies are defined and driven by the needs of science. Grid-based physics collaborations that span the globe allow specialized instruments to be shared by disparate teams that analyze data sets on large, parallel compute clusters. These clusters and the scale of data produce and consume would have been nearly unimaginable only a decade ago.

It is becoming common for teams of scientists to form *virtual organizations*: geographically distributed, functionally diverse groups linked by electronic communication, relying on lateral, dynamic relationships for coordination [1]. Within the Grid, the need for flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources presents unique authentication, authorization, resource access, resource discovery, and other challenges [2].

Collaborations on a global scale, such as the ATLAS project centered at CERN, generate massive amounts of data and share them across dynamically organized hierarchies made up of cohorts of collaborators. These large, distributed collaborations represent many overlapping virtual organizations that are frequently updated as users and resources join and leave the virtual organization.

Dynamic virtual organizations create several new classes of problems unique to inter-institutional collaborations. The GridNFS project addresses two of these problems:

The dynamics of virtual organizations demand agile security mechanisms. Security mechanisms for virtual organizations must be strong enough to protect the integrity of data at all times and to protect the confidentiality of data when necessary. They must be adaptable, to accommodate the varying membership of virtual organizations. They must be able to delineate precise authorization limits for users from outside the virtual organization.

Global collaboration requires a canonical way to name shared data (e.g., file names). This need is driven by the vast amounts of data generated by modern collaborative physics, which must be accessible to a widely dispersed collaborative community.

To address these problems, we are developing GridNFS, a middleware solution that melds distributed file system technology with flexible identity management techniques to meet the needs of Grid-based virtual organizations.

The foundation for data sharing in GridNFS is NFS version 4 [3], the IETF standard for distributed file systems that is designed for security, extensibility, and performance. GridNFS meets the challenges of authentication and authorization with X.509 credentials, which bridge NFSv4 and the Globus Security Infrastructure, allowing GSI identity to control access to files exported by GridNFS servers.

By tying together these middleware technologies, we fill the gap for two vital, missing capabilities: *transparent and secure data management* integrated with existing Grid authentication and authorization tools, and *scalable and agile name space management* for establishing and controlling identity in virtual organizations and for specifying VO data resources.

GridNFS is a new approach that extends "best of breed" Internet technologies with established Grid architectures and protocols to meet these immediate needs and is positioned to adapt to the future needs of Grid computing through the minor versioning provision of the NFSv4 standard.

2. The GridNFS approach

The challenge of building and maintaining a virtual organization can be viewed as one of effective management of users and resources. In isolation, users and resources each require powerful mechanisms for global naming, so that virtual organizations can extend and incorporate users and resources into a consistent framework. In combination, the coordinated

management of users and resources requires powerful means for authentication and authorization. Grid exigencies intensify the challenge, as they introduce the requirement for authorizations that become active dynamically, without user participation subsequent to the making of an authorized request.

GridNFS combines Grid and NMI [4] infrastructures with NFSv4 to solve these challenges. Here is a scenario that shows how these technologies interact.

> *A scientist at an enterprise desktop or laptop uses a Grid client to schedule use of Grid resources. She identifies input and output data objects by global names; in fact, they are the same names that she uses on her desktop computer to identify the data resources, which lets her provide appropriate access controls over the resources.*
>
> *Once the reservation is completely described, a scheduler determines when and where the job will be run and stores appropriate proxy credentials with the scheduled job. Data sets pre-staged through a tiered system of data access are automatically replicated onto secure servers in the local neighborhood of the compute engines to be used.*
>
> *At last, the job is ready to be executed. Proxy credentials that reflect the authorizations of the requesting scientist are provided to data and computing resources and used directly to authorize access to the scientist's files.*
>
> *Finally, the task is complete. Replication nodes for input data are automatically removed from service, while output data is distributed through replication and through conventional tiered mechanisms. The results can be viewed directly by the scientist, whether she is sitting in front of a highly customized visualization workstation in her laboratory or running a conventional application on a commodity laptop while sipping coffee halfway across the world.*

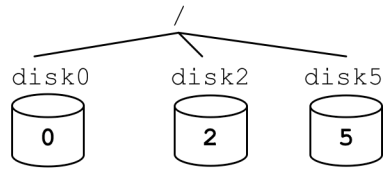
In the remainder of this section, we detail the middleware technologies that provide the transparency, security, and ease-of-use featured in the scenario.

2.1. GridNFS name space for data

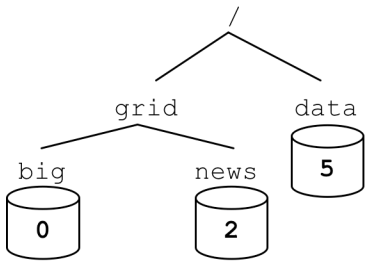
NFSv4 uses the familiar hierarchical style of naming common to modern file systems. However, NFSv4 has two useful features that assist in the engineering of name spaces for virtual organizations.

The first feature is the NFSv4 server pseudo-file system. The server presents clients with a *root file handle* that represents the logical root of the file system tree provided by the server. The server constructs a logical image of the file system it wants clients to see by gluing physical file systems under its control under this logical root. Any gaps between the logical root and the physical file systems are filled with pseudo-directories. Clients then mount the logical root constructed by the server.

Figure 1 shows a pseudo-directory `/grid` that is used to connect server physical volumes into an exported name space. Because the server constructs the view seen by NFSv4 clients, users and applications in a virtual organization have a common name space relative to that server. To provide a common, global name space, the problem remains to knit the server file systems into a common name space for a virtual organization.



The server's file system contains three volumes mounted as `/disk0`, `/disk2`, and `/disk5`.



Clients see `/grid/big`, `/grid/news`, and `/data`. `/grid` is a pseudo-directory that fills the gap between the exported root and the big and news volumes.

Figure 1. Name space construction.

The second NFSv4 feature that we use for constructing a global name space is the FS_LOCATIONS attribute. This lets a server redirect client access requests. When a client encounters a pseudo-node, it retrieves the attribute, whose value is a list of {server, path} pairs. The client picks one from the list and continues with its request at the selected server and path.

Figure 2 shows how redirection can knit server name spaces together. In this example, a traversal to `/c` is redirected to one of the copies maintained by Org 1 and Org 2. Redirection provides essential flexibility in name space construction, allowing the administrators of a virtual organization to dictate the form and shape of that space, especially in the part of the name space close to the root.

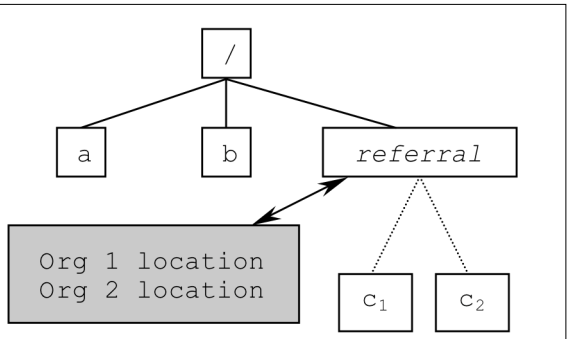


Figure 2. Redirection.

One more feature needed for GridNFS name space construction is consensus on the root of the name space. Related matters are under discussion by IETF working groups. We anticipate a solution that builds on the emerging standard that uses SRV records for locating servers [5]; initially, NFSv4 clients simply mount the pseudo-file system root of a virtual organization's GridNFS server under a directory named /GRIDNFS.

With these features in place, researchers in the VO can discuss data sets with names like /GRIDNFS/VO-PROJECT/HOTDATA/2006/11/24/FILE24 and, with appropriate data management policies, can expect that path name to yield identical results throughout the virtual organization. This simplifies administration of a GridNFS client—it is configured once (and only once) to mount the virtual organization's file hierarchy at /GRIDNFS.

Administration of the root of the virtual organization is also easy: as data servers come and go, redirection points are added to or removed from the name hierarchy rooted at VO-PROJECT/. All clients immediately see the change.

Finally, because the redirection points refer to data servers in autonomous domains under the control of members of the virtual organization, delegation of policy and control to the members of the virtual organization is consistent with their responsibilities and investment.

2.2. GridNFS name space for users

Users in a virtual organization are identified by names bound to credentials drawn from multiple autonomous security and naming domains. To instantiate a virtual organization in the Grid, sub-organizations establish a certificate chain by exchanging X.509 certificates or by agreeing on a certification authority. Users in sub-organizations can then use their existing credentials for access to resources across organizational boundaries. Because GSI and NFSv4 both support X.509 distinguished names as a form of user credential, GridNFS is able to dovetail the two.

The NFSv4 protocol specifies names for access control, e.g., in **`getacl`** and **`setacl`** calls, but NFSv4 clients and server platforms represent users locally with numeric identifiers. To map names and numbers consistently, CITI's GridNFS prototype stores the maps in LDAP, but many systems use **`nsswitch`** to configure choices of mapping techniques. IETF's NFSv4 working group is considering extensions to the NSS name-to-ID mapping standard to support ACLs for foreign users.

GSI uses a proxy X.509 certificate for the user's DN that is mapped to a local name by the Globus Gatekeeper by consulting a local flat file called **`grid-map`**; recent versions also support an ad hoc callout interface, e.g., to CITI's LDAP maps.

The NFSv4 protocol requires support for the Simple Public Key Mechanism version 3 [6], an X.509-based security mechanism that establishes a secure channel between a client and a server. SPKM-3 requires servers to use X.509 credentials but allows users to be anonymous, i.e., to establish the secure channel without X.509 credentials, or to use an X.509 certificate, e.g., GSI credentials, for mutual authentication.

2.3. Automated replication

Availability and performance are vital for Grid middleware. Replication is central to any effective scheme for availability, and is beneficial for performance and scalability. Replication plays an important role in GridNFS by allowing data to be pre-staged to compute engines.

Grid applications often need access to enormous (read-only) data sets, so they use GridFTP to stage data on cluster computers in advance, which allows the data to be accessed at furious rates when it is needed. Often, much of the pre-staging can be automated, but conflicts arise with security and disk management facilities on the cluster node.

Automated data replication offers a tantalizing alternative to manual or semi-automated pre-staging. CITI's replication and migration prototype for NFSv4 [7] offers per-file granularity, read and write access, and optimal performance for read-only files. We are extending that work to mesh well with Grid security, to automate the creation and destruction of replication sites, and to pre-stage data securely on those sites.

With this development in hand, we can pre-stage data to GridNFS servers in a very tight neighborhood surrounding a compute server. From there, the option remains to use GridFTP to move the data the "last mile" or to use emerging parallel access mechanisms to provide direct access to Grid applications from the GridNFS name space.

2.4. Wide-area performance

Superior performance is critical for GridNFS to be accepted as a central middleware component. NFSv4 has the essential core to provide excellent performance, but the special requirements of Grid network—long fat pipes, parallel networks, and tiered distribution—demand close attention. The GridNFS project is pursing several avenues to meet performance requirements:

- Engineering Linux kernel data paths and RPC overheads;
- Examining opportunities for hardware-assists to data transfer, such as RDMA and TCP offload; and
- Exploiting "minor version" extensions to NFSv4 for integration with parallel storage.

3. The dawn of NFSv4

Several technologies have struggled to meet the performance and scaling requirements of Grid data access.

AFS [8], used extensively in the physics community, features a global name space, secure access control, a rich back-end management system, and an energetic open-source development community. Yet AFS has limitations that make it unsuitable for many Grid applications: AFS does not meet the performance requirements of cluster computers, especially with massive files, and relies on a UDP-based network library whose design target is nearly 20 years out of date. Furthermore, its coarse-grained security model interferes with access control across autonomous security domains and is not well suited for pre-scheduled access.

NFSv3 [9] is also used extensively on the Grid, e.g., for data sharing between cluster nodes. Like AFS, NFSv3 was built on UDP, but it is now based on TCP and offers superior performance across a wide range of network conditions. However, NFSv3 security deficiencies preclude its use in a WAN or any other untrusted environment. In many ways, NFSv3 suffers from the same problem as AFS: its design target is long obsolete, e.g., insistence on stateless servers, motivated by reliability and scaling considerations mooted a decade ago.

NFSv4 was designed with the lessons of AFS and NFSv3 in mind. NFSv4 provides transparent, high-performance access to files and directories, but supplants NFSv3's troublesome lock and mount protocols. Strong and diverse security mechanisms are mandatory in NFSv4. NFSv4 also supports scalable and consistent client caching and internationalization. Much attention has been given to making NFSv4 operate well in a WAN or Internet environment.

At present, the method of choice for transporting data on the Grid is **GridFTP** [10], which is used directly and under the covers in many physics Grid applications. Because it was engineered with Grid applications in mind, GridFTP has many advantages: automatic negotiation of TCP options to fill the pipe, parallel data transfer, integrated Grid security, and partial transfers that can be resumed. In addition, as an application, GridFTP is easy to install and support across a broad range of platforms. On the other hand, because it is not integrated into the kernel, GridFTP cannot take advantage of kernel features like zero-copy access, range locks, integration into the operating system name space, and fine-grained sharing. Furthermore, we would argue that the URL-like name space is a bit of a mess, although this is mostly a matter of taste [11].

SRB [12] is mature Grid middleware that solves many data access and management problems, especially life cycle and metadata management for heterogeneous data sets. Like GridFTP, SRB lives outside the kernel and shares the concomitant advantages and disadvantages discussed above.

GridNFS works alongside GridFTP or SRB. In tiered projects such as ATLAS, GridFTP remains a natural choice for long-haul scheduled transfers among the upper tiers, while the file system semantics of GridNFS offers advantages in the lower tiers. Domain scientists can work directly with GridNFS files using conventional names. This promotes effective data management without obviating SRB's life-cycle strengths. GridNFS also offers transparent support for operating system extensions such as RDMA, file replication and migration, extended attributes, and parallel access.

3.1. Broader impacts

Concurrent with our development of GridNFS, NFSv4 is being deployed by many vendors: Sun, Network Appliance, IBM, HP, Hummingbird, and EMC actively participate in development and testing. We expect that NFSv4 will quickly and silently dis-

place NFSv3, just as NFSv3 displaced NFSv2. We also anticipate that its advanced features and vendor support will lead to embrace by remaining AFS installations. NFSv4 stands ready to realize the unkept promises of DCE/DFS for enterprise computing.

NFSv4 and the Grid are simultaneously poised for exponential growth in influence. The increasing commercial influence of NFSv4 dovetails with the GridNFS project over its initial three-year span. Cluster computing in problem domains ranging from high-energy physics to entertainment will rely increasingly on NFSv4 to tie massively parallel data engines to massively parallel compute servers. Parallel file systems such as Lustre, GPFS, GFS, PolyServe Matrix Server, and Panasas will use extensions of NFSv4 to position themselves as conventional, standards-compliant components, able to meet spectacular opportunities offered by TeraGrid and StarLight while continuing to satisfy the needs of enterprise desktops.

Acknowledgement

This research is supported in part by the NSF Middleware Initiative.

Bibliography

[1] G. DeSanctis and P. Monge, "Communication Processes for Virtual Organizations," *J. Computer-Mediated Comm.*, 1998.

[2] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int. J. of HPC Apps.*, 2001.

[3] S. Shepler, B. Callaghan, D. Robinson, R. Thurlow, C. Beame, M. Eisler, and D. Noveck, "Network File System (NFS) version 4 Protocol," *RFC 3530*, 2003.

[4] NSF Middleware Initiative, www.nsf-middleware.org/.

[5] A. Gulbrandsen, P. Vixie, and L. Esibov, "A DNS RR for specifying the location of services," *RFC 2782*, 2000.

[6] M. Eisler, "LIPKEY—A Low Infrastructure Public Key Mechanism Using SPKM," *RFC 2847*, 2000.

[7] J. Zhang and P. Honeyman, "Naming, Migration, and Replication in NFSv4," *CITI Tech. Rep. 03-2*, 2003.

[8] M. Satyanarayanan, J.H. Howard, D.A. Nichols, R.N. Sidebotham, A.Z. Spector, and M.J. West, "The ITC Distributed File System: Principles and Design," *SOSP*, 1985.

[9] B. Callaghan, B. Pawlowski, and P. Staubach, "NFS Version 3 Protocol Specification," *RFC 1813*, 1995.

[10] Globus Project, "GridFTP: Universal Data Transfer for the Grid," White Paper, 2000.

[11] R. Pike and P.J. Weinberger, "The Hideous Name," *USENIX*, 1985.

[12] C. Baru, R. Moore, A. Rajasekar, and M. Wan, "The SDSC storage resource broker," CASCON, 1998.