NMI DEVELOPMENT: GridNFS

Application to Domain Science

Grid technologies have been 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 that are linked by electronic forms of communication and that rely on lateral, dynamic relationships for coordination [DeSanctis]. 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 [Foster].

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 collections of collaborators. These large distributed collaborations represent many overlapping virtual organizations that are frequently updated as users and resources enter and exit the virtual organization. Dynamic virtual organizations create several new classes of problems unique to inter-institutional collaborations that must be solved.

In this proposal, we aim to address two of these problems. The dynamics of virtual organizations demand agile security mechanisms. These security mechanisms must have several properties. First, they must be strong enough to protect the integrity of data at all times. Second, they should protect the confidentiality of data when necessary, yet be adaptable enough to accommodate the varying membership of virtual organizations. Third, these security mechanisms must be able to delineate authorization limits precisely for users from outside the virtual organization. Thus, the first problem addressed by this proposal is the development of strong security mechanisms for virtual organizations. The second problem addressed by this proposal is the need to develop a consistent canonical way to name shared data (e.g. filenames). 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 propose to develop GridNFS, a middleware solution that extends distributed file system technology and flexible identity management techniques to meet the needs of grid-based virtual organizations. The foundation for data sharing in GridNFS is NFS version 4 [Shepler], the IETF standard for distributed file systems that is designed for security, extensibility, and high performance. The challenges of authentication and authorization in GridNFS are met with X.509 credentials, which can bridge NFSv4 and the Globus Security Infrastructure, allowing GSI identity to be used in access control lists on 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.
- Scalable and agile name space management for establishing and controlling identity in virtual organizations and for specifying virtual organization 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 IETF minor versioning provision of the NFSv4 standard.

Proposed Approach

The challenge of building and maintaining a virtual organization can be viewed as one of effective management of users and resources. In isolation, both users and resources require powerful mechanisms for global naming, so that virtual organizations can extend to 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.

We propose to meld Grid and NMI infrastructures with NFSv4, producing GridNFS, to solve the challenges outlined above. Here is one scenario of how we see these technologies interacting.

> *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.

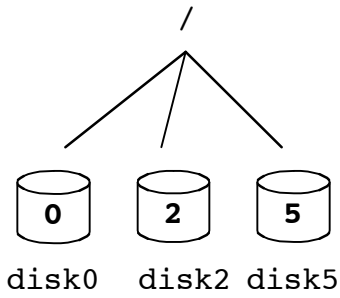
GridNFS name space for data

NFSv4 used the same familiar hierarchical style of naming as other modern file systems.

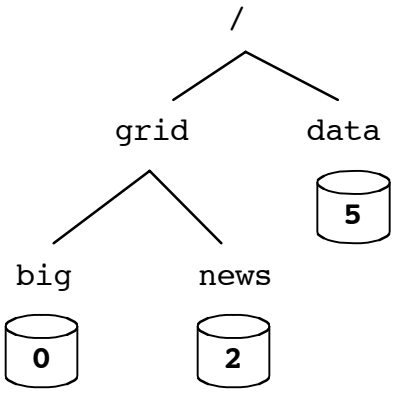
However, NFSv4 has two useful features that assist in the construction (and destruction!) of name spaces for virtual organizations.

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.

In the following example, a pseudo-directory is used to connect server physical volumes into a coherent name space.



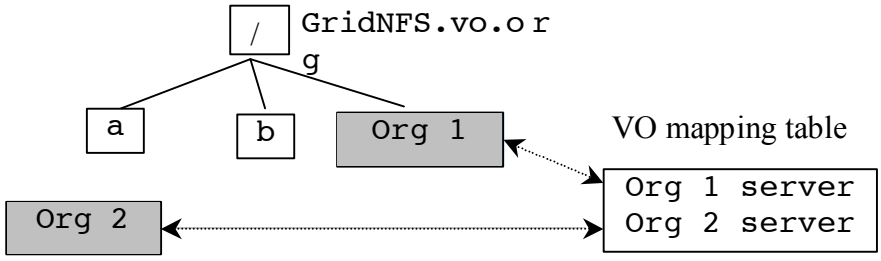
The server's file system contains three volumes mounted as `/disk0`, `/disk2`, and `/disk5`.



Clients see `/grid/big`, `/grid/news`, and `/data`. The `/grid` directory is a pseudo-directory invented by the server to fill the name gap between the exported root and the "big" and "news" volumes.

Because the server is able to construct a view that is shared among clients, clients in a virtual organization can mount the server's exported root and be assured that they share a common name space relative to that server. It remains only to provide clients in a virtual organization a common name space for server file systems to provide a common, global name space.

The second feature is the FS_LOCATIONS attribute, which lets a server redirect client access requests. When a client see an FS_LOCATIONS attribute on a pseudo-node, it retrieves the value of the attribute, a list of {server, path} pairs. The client picks one from the list and retries its request at the selected server and path.



In our application, the redirection list usually consists of a single entry. 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 at in the part of the name space close to the root.

The third feature needed for GridNFS name space construction is consensus on the root of the GridNFS 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 [Gulbrandsen], but initially we intend for clients simply to mount the pseudo-filesystem 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 makes administration of a GridNFS client trivial – 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.

Task: Implement FS_LOCATIONS

The FS_LOCATIONS attribute is optional, and not yet implemented in the Linux NFSv4 client and server.

- Implement and test FS_LOCATIONS for the Linux NFSv4 client and server.
- Work with the IETF NFSv4 working groups to define extensions to the FS_LOCATIONS attribute useful for defining and controlling access to Grid data collections.

GridNFS name space for users

In Globus GSI as in NFSv4, users are identified by name, not by number. Globus GSI uses X.509 distinguished names. NFSv4 also uses names that correspond to RPCSEC_GSS mechanisms; X.509 DNs and Kerberos-style user@realm identifiers are required in any compliant implementation. Because virtual organizations span autonomous security and naming domains, GridNFS must translate identifying strings at domain borders.

GSI uses a proxy X.509 certificate for the user's DN that is mapped to a local name by the Globus Gatekeeper. In recent versions of Globus, this mapping is done either by consulting a local flat file called the *gridmap* file, or by an ad hoc callout interface. There are no restrictions on this GSI DN mapping, e.g., many DNs can be mapped to the same local name.

The NFSv4 protocol uses names for access control, e.g., in `getacl` and `setacl` calls. A single NFSv4 user can have multiple DNs and identities in any number of Kerberos v5 realms. Yet the desktop clients and file servers generally use numeric identifiers for access control, e.g., the Linux client and server use UNIX UIDs. Thus, the UID must be mapped into an RPCSEC_GSS principal by the client (resp., server) and back to a UID by the server (resp., client). The situation is messy – knowing which credential to use depends on the access control list attached to the resource – and is complicated by the weak support for credential management in the Linux kernel.

At the end of the day, though, local file systems on Linux represent ownership as numeric UIDs. While this mapping is operating system dependent, many systems support nsswitch, a means of identifying the choices of mapping techniques.

The NFSv4 working group is considering extensions to the standard that securely automates the mapping process, as this would help UID-based systems support the use of ACLs for foreign users. This effort is essential to GridNFS, so that VO identity can be used across all of the GridNFS servers in the virtual organization. Furthermore, a single name space for VO groups would also allow integrating GridNFS with group-based GSI resource control.

Task: User name integration

- Extend the NFSv4 name translation mechanisms to support the Gridmap approach.
- Implement a secure and automated mechanism for bidirectional translation of foreign users and groups with local UIDs and GIDs.

Integrating GSI and GridNFS Identity in a Virtual Organization

In establishing a virtual organization in the Grid, sub-organizations establish a certificate chain by exchanging self-signed X.509 certificates. Users in the sub-organization are issued signed certificates, so the requirements for establishing identity are satisfied and users can be granted access to resources across organizational boundaries, which is the virtual organization's *raison d etre*.

NFSv4 also mandates an X.509-based security mechanism. The Simple Public Key Mechanism version 3 (SPKM-3) [Eisler] is an X.509-based security mechanism for GSS

that is mandatory for NFSv4 implementations. SPKM-3 allows an anonymous user, i.e., one with no X.509 user credentials, to establish a secure channel with a server that does have a public key pair and an X.509 certificate. SPKM-3 is based on SPKM-1 [Adams], but SPKM-3 requires Diffie-Hellman key exchange and allows RSA, while SPKM-1 requires RSA but allows Diffie-Hellman.

LIPKEY, the Low Infrastructure Public Key GSSAPI security mechanism, also mandatory for NFSv4, uses SPKM-3 to establish a secure channel between the NFSv4 client and server. The client then sends a user name and password to the server over the secure channel.

SPKM-3 also provides for mutual authentication. A user with X.509 user credentials can establish a secure channel using Diffie-Hellman key exchange in conjunction with one of the SPKM-3 recommended integrity algorithms. This allows GSI credentials to be used with NFSv4.

Task: GSI-compatible security mechanism for NFSv4

The Linux open-source implementation of NFSv4 is architecturally compatible with GSI-based security.

- Implement and test SPKM-3 GSSAPI security mechanism with mutual authentication for the Linux NFSv4 client and server, integrate them into the Linux distribution.
- Work within the IETF to review RFC 2847 for completeness.
- Ensure GSI and SPKM-3 Linux NFSv4 implementations are compatible.

Automatic secure dynamic 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 can play an important role in GridNFS by allowing data to be staged "near" compute engines prior to execution.

Grid applications often need access to enormous read-only data, and use GridFTP to stage the data on cluster computers in advance. This allows the data to be accessed at furious rates when it is needed. Much of the pre-staging can be often 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, especially if replication sites can be constructed dynamically, securely, and automatically. Researchers at CITI have begun developing a replication and migration scheme for NFSv4 [Zhang] that offers per-file granularity and optimal performance for read-only files. We propose to complete that implementation, extend it to mesh well with Grid security, to develop tools 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.

Task: Automatic secure dynamic replication for GridNFS

The CITI research effort in replication for NFSv4 must be fleshed-out and tested in the context of the special requirements of domain scientists and Grid applications.

- Implement server-to-server read-only replication for GridNFS in Linux.
- Develop the means to create, populate, and destroy GridNFS replication sites securely and under the control of a Grid task scheduler.
- Test and measure the functionality and performance of GridNFS replication in Grid deployment scenarios
- Tune the design and implementation of replication for GridNFS based on the results of performance and functionality tests.

Metropolitan- and wide-area performance

Superior performance is critical for GridNFS to be accepted as a central middleware component, and will be a focus of research and development activities. NFSv4 has the essential core to provide excellent performance, but the special requirements of Grid network – long fat pipes, parallel networks, tiered distribution – will demand close attention in the development of GridNFS.

Aspects of this activity include performance issues in Linux kernel data paths, RPC overheads, opportunities for hardware-assists to data transfer, such as RDMA and TCP offload, and emerging IETF extensions to NFSv4 for parallel access and integration with parallel back-end data stores. CITI technologists and researchers are engaged in RDMA and parallel access developments, which offers considerable intellectual leverage and opportunities to develop from a running start.

Task: Tune GridNFS for metropolitan- and wide-area performance

Tiered distribution of Grid data sets offers opportunities and constraints in tuning GridNFS.

- Measure GridNFS in representative Grid computing contexts, identify bottlenecks and tuning opportunities.
- Compare with GridFTP, use GridFTP speed enhancements as a roadmap for GridNFS development.
- Investigate developments in hardware assists and software support for high-speed networking in Linux; integrate these developments with GridNFS where possible.

Comparison with other approaches

Data sharing is at the heart of grid computing. Several technologies have struggled to meet the performance and scaling requirements of grid data sharing, with varying degrees of success.

AFS [Satyanarayanan], used extensively in the physics community, offers the advantages of a global name space, secure access control, a mature 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, AFS' coarse-grained security model interferes with access control across autonomous security domains and is unsuited for "just in time" access control needed to enable cluster nodes to access resources in a scheduling interval that is determined long after a work unit is submitted to the scheduler.

NFSv3 [Callaghan] 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 has long suffered from well-known security deficiencies, which precludes its use in a WAN environment. In many ways, NFSv3 suffers from the same basic problem as AFS: its design target is long obsolete, e.g., the insistence on stateless servers, motivated by reliability and scaling considerations that were passed by over a decade ago.

At present, the method of choice for transporting data on the grid is **GridFTP** [Globus], 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 restarted. 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 [Pike].

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 security is mandatory in NFSv4; a compliant implementation is required to support diverse security models. NFSv4 extends the communication framework of NFSv3 to support chained requests, includes support for scalable and consistent client caching, and internationalization. Much attention has been given to making NFSv4 operate well in a WAN Internet environment.

The NFSv4 protocol has many configurable options. Some options – protocol choice, read and write sizes, security flavor – are determined by client mount and server export interfaces. Other options such as adherence to POSIX standards for byte-range locking or access control lists are dictated by capabilities of the exported file system and are communicated via attributes. Still other options, such as name-to-ID mapping and name space construction, are the determined by the administrators of the environment in which NFSv4 runs.

Our proposed 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 GSI, fine-grained access control lists to support virtual organization groups and users, and secure file system access for jobs scheduled in an indeterminate future.

Because GridNFS combines and integrates standard Internet protocols, it remains fully compatible with standards-compliant desktop and enterprise network services. Furthermore, the middleware developed for GridNFS enhances those environments as well by offering such desirable features as global naming and facile identity representation for agile access control across security domains.

Our proposal to develop GridNFS is not intended to replace GridFTP, but to work alongside it. For example, 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. For example, with GridNFS, domain scientists can work with files directly using conventional names, which promote effective data management. GridNFS also offers seamless support for operating system extensions such as RDMA or file replication and migration.

Integration with existing grid technologies

Integrated software builds on existing platforms; GridNFS is no exception. In particular, GridNFS relies on numerous technologies that have been incorporated into NMI and into widely deployed grid infrastructures.

- GSI, the grid security infrastructure, uses X.509 certificates and an SSL- and GSS-based mechanism to identify Grid users. GridNFS also uses X.509 and GSS, so GSI identities can be used directly to identify local and foreign users.
- NFSv4 and GSI both support X.509 distinguished names "on the wire", and share the requirement that DNs be mapped to and from local identities at desktops and compute engines. By reusing the GSI mappings, GridNFS provides a seamlessly integrated view of the name space for users in the virtual organization.
- Emerging Globus functionality includes authorization for access to resources. Currently, Globus uses groups for resource authorization. GridNFS can use the same groups, extending their use to file system authorization under the NFSv4 fine-grained access control model.
- Because GridNFS is a drop-in replacement for the local file system, it can be utilized easily by Grid data managers such as the SDSC Storage Resource Broker.

Testing and Deployment

Initial testing and deployment will take place on a dedicated Linux cluster at CITI, but as soon as a viable package is ready we intend to deploy it more broadly to test:

1) Performance of GridNFS in the wide-area network
2) Ease of use and installation at other sites
3) Manageability and functionality
4) Integration with existing grid systems and software

To this end we plan to work closely with the International Virtual Data Grid Laboratory [IVDGL] to deploy GridNFS on some of their systems and conduct interoperability tests between sites. In addition we plan to interface GridNFS into existing grid software (such

as Chimera, dCache, DIAL, Magda, Pegasus, SRM, etc.) in use on iVDGL to compare and contrast its capabilities with GridFTP (or other data transport applications).

Results from Prior Support

CITI is the central developer of the Linux-based, open source reference implementation of NFSv4. From July 1999 to June 2004, Sun Microsystems engaged CITI as a research and development partner, funding over $1.5M in development activities. That partnership produced a protocol compliant, high-performance NFSv4 client and server integrated into the Linux 2.6 kernel, available by default in every modern Linux distribution.

CITI has also worked with the Dept. of Energy Tri-Labs to extend NFSv4 to meet the special data handling needs of cluster computers. CITI is at the forefront of the development of NFSv4 extensions to access parallel back ends simultaneously through multiple NFSv4 servers.

CITI and Network Appliance are research partners, exploring and advancing NFS performance on Linux clients. CITI's migration and replication research is a recent outcome of the partnership with Network Appliance, as was some fundamental work to enhance the performance and scalability of Linux NFS clients [Honeyman].

Other partners, including IBM, Dell, and Netscape, have contributed to Linux and NFSv4 research and development at CITI, which has also led to a number of external publications [Provos1, Provos2, Molloy, Lever].

Shawn McKee: NSF 03-530, NSDL (10/01/2003-9/30/2004; $249,998) "Web Lecture Archiving System for Professional Society Meetings". This project is developing a system to record speakers (audio, video, presentation) and make them available in near real-time, via a web-lecture archive object, achieving expert results without requiring an expert operator.

Education and Outreach

CITI's partnerships have provided undergraduate students with the opportunity to intern at CITI and its partners and to make significant contributions to the state of the art as well as to emerging products. Students that cut their teeth on NFSv4 at CITI have gone on to become graduate student fellows at top universities, highly prized employees at Sun, IBM and Google, and summer interns Sun, Microsoft, Network Appliance, and Intel. CITI's undergraduate interns, who are among the department's top students academically, seek the opportunity to learn teamwork and technical leadership from the University of Michigan's most highly regarded technologists.

CITI also works side-by-side with graduate student research fellows engaged in advanced research while in pursuit of the PhD. Although CITI produces relatively few PhDs, they go on to distinguished careers in academia, such as Prof. Avi Rubin at Johns Hopkins University, or industry, such as LDAP inventor Dr. Tim Howes, who founded LoudCloud, Inc.

Project Plan

Months 1-12

- Implement and test `FS_LOCATIONS` for the Linux NFSv4 client and server.
- Extend the NFSv4 name translation mechanisms to support the Gridmap approach.
- Implement a secure and automated mechanism for bidirectional translation of foreign users and groups with local UIDs and GIDs.
- Implement and test SPKM-3 GSSAPI security mechanism with mutual authentication for the Linux NFSv4 client and server, integrate them into the Linux distribution.
- Ensure GSI and SPKM-3 Linux NFSv4 implementations are compatible.

Months 13-24

- Begin initial testing and deployment with iVDGL to address planning and design issues for support of e-Science activities
- Implement server-to-server read-only replication for GridNFS in Linux.
- Develop means to create, populate, and destroy GridNFS replication sites securely and under the control of a Grid task scheduler.
- Investigate developments in hardware assists and software support for high-speed networking in Linux; integrate these developments with GridNFS where possible.

Month 24: proof of concept demonstration

Initial testing and deployment will take place on a dedicated Linux cluster at CITI. We will utilize ATLAS grid software to test GridNFS deployment and interaction with an existing grid infrastructure:

- Simple modifications to ATLAS grid software to allow it to utilize GridNFS
- Contrast performance and design of grid software using GridNFS vs current methods
- Document steps needed to enable use of GridNFS with existing software
- Identify implications for ease of use and grid software design in the context of a virtual organization.

In Month 24, we will schedule a proof-of concept demonstration of the following software components of GridNFS:

- Hierarchical name space management for virtual organizations
- Virtual organization user name space translation and interoperability
- GSI interoperability with NFSv4 security
- Use of server replication for pre-staging data

Months 25-36

- Continue to investigate developments in hardware assists and software support for

high-speed networking in Linux; integrate these developments with GridNFS where possible

- Tune the design and implementation of replication for GridNFS based on the results of performance and functionality tests.
- Implement production-quality replication tools for GridNFS.
- The Month 24 demonstration positions us to deploy software components to iVDGL and begin to test:
  - Performance of the package in the wide-area network
  - Ease of use and installation at other sites
  - Manageability and functionality
  - Integration with existing grid systems and software

Metrics of success

We will use standard file system benchmarks such as Iozone and Bonnie to measure the performance of GridNFS, including:

- Compare with GridFTP, use GridFTP speed enhancements as a roadmap for GridNFS development.
- Measure GridNFS in representative Grid computing contexts, identify bottlenecks and tuning opportunities.
- Measure the efficacy of GridNFS replication by comparing throughput and latency between nearby replicated and remote data sets.
- Document steps for integration with existing grid software and infrastructure and implications for virtual organizations and grid software design and use.

Summary and Broader Impacts

Application to Bioinformatics

Administratively attached to the University of Michigan, the Michigan Center for Biological Information is one of five centers that make up the Michigan Life Sciences Corridor Core Technology Alliance. The MLSC-CTA's mission is to develop a collaborative network of technologically sophisticated core facilities to enhance life sciences research and product development throughout the State of Michigan. MLSC-CTA acts as a catalyst for development of life sciences and biotechnology research and development by providing access to advanced technologies to Michigan researchers affiliated with universities, private research institutes and biotechnology or pharmaceutical firms. The other MLSC-CTA centers include the Michigan Center for Genomics Technology at Wayne State University, the Michigan Proteome Consortium at the UM, the Michigan Center for Structural Biology at Michigan State University, and the Michigan Animal Model Consortium at the Van Andel Research Institute.

MCBI provides advanced bioinformatics and computational resources for investigators in

the academic and industrial sectors of Michigan. Researchers have access to bioinformatics tools, genomics and proteomics databases, supercomputing resources, bioinformatics training, and bioinformatics consulting.

MCBI has an ongoing research and development program focused on the integration of biological database information in a biologically meaningful way to permit new queries across databases and other biological resources. MCBI currently maintains local mirrors of a number of biological databases for Michigan users. These include the Ensembl databases, several oligonucleotide probe databases for optimal gene expression array and dot blot construction, a proteome standards database, and a human gene microarray standards database.

MCBI hopes to use GridNFS to provide access to its database archive. This access will supplement and perhaps eventually replace the more traditional access to such archives provided by the File Transfer Protocol (FTP), rsync, and various Web based access systems. MCBI sees the advantages of using GridNFS in this environment as:

- allowing access to remote files and databases using the same programs and procedures that researchers use to access such files on their own local systems
- avoiding the need for each researchers to make and maintain their own local copies of each database and to monitor for updates to the databases
- allowing controlled access to databases and files that cannot be made available to everyone due to license restrictions or a desire to limit access until results have been finalized or published
- allowing controlled access to the databases and files using credentials issues by a users home organization rather than having to create new credentials (Ids and passwords) specifically for access to the archive
- allowing us to experiment with caching databases and files using the facilities that are present in NFSv4 as a way to provide high performance access to remote files that today often requires that the databases and files to be stored on local disks

A wide range of applications can use and would benefit from GridNFS access to the MCBI biological database archive. An example of one such program that would benefit from GridNFS access is a modified version of BLAST being developed as part of some as yet unpublished work being done at the University of Michigan. In part this work separates the BLAST database into indexes that are separate from the complete database. The indexes can be copied or cached to local disks to improve performance and the complete database can be access using seek calls via GridNFS, so that only the portions of the database that are of interest rather than the entire database has to be transferred.

Most non-profit bio-database groups provide a means for mirroring their databases, and a large number of these databases can be found in one place at Bio-mirror.net, in a compressed format. While Bio-Mirror.net has many important databases such as Genbank, Swissprot, DDBJ, Prosite, Pfam, and Enzyme, it is missing important biological databases such as Kyoto Encyclopedia of Genes and Genomes (KEGG), BIND, and DIP. MCBI is currently exploring the feasibility of adding these additional databases to our local collection. Decompressing and installing these databases locally

can take a substantial amount of time, (e.g., four hours on a workstation in the case of Ensembl), and so is often a bottleneck for high-performance bioinformatics users. Allowing remote file system access to these files and databases using GridNFS could substantially reduce the human as well as computer overhead needed to access this data by more traditional file transfer means today.

The dawn of NFSv4

Concurrent with our development of GridNFS, NFSv4 will be deployed by many vendors: Sun, Network Appliance, IBM, HP, Hummingbird, and EMC have actively participated in development and testing. We expect that NFSv4 will quickly and silently displace NFSv3, just as NFSv3 rapidly 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.

The increasing influence of NFSv4 in the commercial space will also dovetail with the GridNFS project over its three-year span. Cluster computing in problem domains that range 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, and Panasas will use extensions of NFSv4 to position themselves as conventional, standards-compliant components, able to meet spectacular high-performance demands at the same time that they satisfy the mundane needs of enterprise desktops.

NFSv4 and the Grid are simultaneously poised for exponential growth in influence. The scientists and engineers of the University of Michigan are the leaders and best choice to combine that potential for the benefit of science and society.

REFERENCES

[DeSanctis] G. DeSanctis and P. Monge, "Communication Processes for Virtual Organizations," J. of Computer-Mediated Communication 3(4), June 1998.

[Foster] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," International J. of High Performance Computing Applications, August 2001.

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

[Gulbrandsen] A. Gulbrandsen, P. Vixie, and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)," RFC 2782, February 2000.

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

[Satyanarayanan] M. Satyanarayanan, John H. Howard, David A. Nichols, Robert N. Sidebotham, Alfred Z. Spector, Michael J. West, "The ITC Distributed File System: Principles and Design," Symposium on Operating Systems Principles, December 1985.

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

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

[Pike] Rob Pike and Peter Weinberger, "The Hideous Name," USENIX Conf., June 1985.

[IVDGL] The International Virtual Data Grid Laboratory, http://www.ivdgl.org.

[Honeyman] Peter Honeyman, Chuck Lever, Stephen P. Molloy, and Niels Provos, "The Linux Scalability Project," NLUUG Najaarsconferentie, November 1999.

[Provos1] Niels Provos and Chuck Lever, "Scalable Network I/O in Linux," USENIX Technical Conference FREENIX track, June 2000.

[Provos2] Niels Provos, Chuck Lever, and Stephen Tweedie, "Analyzing the Overload Behavior of a Simple Web Server," Linux Showcase and Conference, October 2000.

[Molloy] Stephen P. Molloy and Peter Honeyman, "Scalable Linux Scheduling," USENIX Technical Conference FREENIX track, June 2001.

[Lever] Chuck Lever and Peter Honeyman, "Linux NFS Client Write Performance," USENIX Technical Conference FREENIX Track, June 2001.