FINAL REPORT

SCI: NMI DEVELOPMENT: GridNFS
National Science Foundation Grant SCI 0438298

Abstract

This report provides an overview of the GridNFS project, details project accomplishments, reports on current status, and discusses future challenges.

GridNFS is an information technology middleware component that extends distributed file system technology and flexible identity management techniques to meet the needs of grid-based virtual organizations. GridNFS fills the gap for two vital, missing capabilities:

- Transparent and secure data management integrated with existing authentication and authorization tools, and
- Scalable and agile name space management for establishing and controlling identity in virtual organizations and for specifying their data resources.

We addressed these problems in part by integrating the security and identity mechanisms of virtual organizations with those of NFSv4, the Internet standard for distributed filing. Authentication and authorization in GridNFS are based on X.509 credentials, which bridge NFSv4 and the Globus Security Infrastructure, allowing GSI identity to be used for access control lists of files managed by GridNFS servers. We also extended NFSv4 to support global naming, read-only replication, and mutable replication.

GridNFS overview

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.[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 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 new classes of problems unique to inter-institutional collaborations that must be solved. In this project, we addressed two of these problems. The dynamics of virtual organizations demand agile security mechanisms. These security mechanisms must have several properties:

- They must be strong enough to protect the integrity of data at all times.
- They should protect the confidentiality of data when necessary, yet be adaptable enough to accommodate the varying membership of virtual organizations.
- These security mechanisms must be able to delineate authorization limits precisely for users from outside the virtual organization.

Thus, the first problem to be addressed is the development of strong security mechanisms for virtual organizations. The second problem to be addressed is the need to develop a consistent, canonical way to name shared data (e.g.,

[1] DeSanctis, G. and Monge, P. 1998. Communication processes for virtual organizations. *J. of Computer-Mediated Communication* 3:4. Wiley InterScience, New York.

[2] Foster, I., Kesselman, C., and Tuecke, S. 2001. The anatomy of the grid: enabling scalable virtual organizations. *International J. of High Performance Computing Applications* 15:3. Sage Publications, Thousand Oaks, CA, pp. 200–222.

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 developed 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,[3] 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 bridge NFSv4 and the Globus Security Infrastructure,[4] allowing the same GSI identity used in Grid scheduling and access rights to be used to control access to storage managed by GridNFS.

Summary of GridNFS accomplishments

The GridNFS project focused on a number of critical areas required to enable the above vision. The major results of the project include:

- Modifications to the Linux-based, open source reference implementation of NFSv4 that enable the specific GridNFS capabilities detailed in this report. These changes to Linux are part of the mainline kernel, distributed worldwide.
- Close interaction with DESY developers to enable a "GridNFS" like interface to dCache[5] systems, a critical infrastructure component for LHC broadly deployed in Grid facilities worldwide.
- Close interaction with Condor developers to develop compatible security and identity mechanisms.
- Research and development of consistent mutable replication mechanisms that perform well even in the harshest Grid environments: high latency, write-dominated, and failure-prone.
- One journal paper, three refereed conference papers, four refereed workshop papers, four Internet drafts, seven technical reports, and one doctoral dissertation.

Detailed tasks and milestones

GridNFS name space for data

- 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.

CITI GridNFS developers implemented client and server support for the FS_LOCATIONS attribute. Client support was accepted into the Linux 2.6.17 kernel on June 9, 2006. Server support was accepted into the Linux 2.6.19 kernel on October 4, 2006.

Information about FS_LOCATIONS referrals is passed into the kernel through exportfs and mountd. These daemons were modified to process replicas and refer directives. The updated daemons have been part of the nfs-utils distribution since the February 27, 2007 nfs-utils-1.0.12 release.

GridNFS name space for users

- 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.

To accomplish user name integration, we extended NFSv4 to use VOMS,[6] the lingua franca of Grid identity mapping, and added GUMS[7] support to the NFSv4 ID map daemon (IDMAPD). This allows the following scenario for name translation:

[3] SHEPLER, S., CALLAGHAN, B., ROBINSON, D., THURLOW, R., BEAME, C., EISLER, M., AND NOVECK, D. 2003. Network file system (NFS) version 4 protocol. RFC 3530.

[4] FOSTER, I., KESSELMAN, C., TSUDIK, G., AND TUECKE, S. 1998. A security architecture for computational grids. In *Proceedings 5th ACM Conference on Computer and Communications Security* (San Francisco, November 1998). ACM, New York, pp. 83–92.

[5] FUHRMANN, P. 2004. dCache, the commodity cache. In *Proceedings 12th NASA Goddard and 21st IEEE Conference on Mass Storage Systems and Technologies* (Adelphi, MD, April 2004). B. KOBLER AND P.C. HARIHARAN, Eds. NTIS, Springfield, VA, pp. 171–176.

[6] ALFIERI, R., CECCHINI, R., CIASCHINI, V., DELL'AGNELLO, L., FROHNER, Á., GIANOLI, A. LÕRENTEY, K., AND SPATARO, F. 2004. VOMS, an authorization system for virtual organizations. In *Lecture Notes in Computer Science 2970: Grid Computing (Proceedings 1st European Across Grids Conference,* Santiago de Compostela, Spain, February 2003). F.F. RIVERA, M. BUBAK, A.G. TATO, AND R. DOALLO, Eds. Springer, Berlin/Heidelberg, pp. 33–40.

[7] BAKER, R., YU, D., AND WLODEK, T. 2003. A model for Grid user management. In *Proceedings of Conference on Computing in High Energy and Nuclear Physics* (La Jolla, March 2003).

- A process acquires a public key certificate (PKC), possibly a GSI proxy credential, or a KX.509[8] certificate.
- This PKC is then conditioned by the VOMS server to incorporate VOMS attributes.
- IDMAPD then passes the PKC to the GUMS server, which uses a Gridmap file or other mechanisms to translate the VOMS attributes into a UID.

Adding GUMS support to IDMAPD required a substantial redesign. IDMAPD has long been used for mapping user identity through the UNIX Name Service Switch, LDAP, etc, but accretion of mechanisms interferes with portability, so we provided IDMAPD with a plug-in architecture. This lets interested developers and distributions build and use the GUMS plug-in without requiring *all* NFSv4 developers and distributions to support GSI.

The plug-in version of IDMAPD, know as libnfsidmap, was released in December 2005. The current version, 0.21, was released in July 2008. The source code for libnfsidmap has been downloaded from CITI web space more than 7,000 times. The 1.0 release is planned for summer 2008.

Integrating GSI and GridNFS identity (I)

- 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.

We implemented and tested SPKM-3,[9] a GSI-compatible security mechanism for NFSv4. Our code is not fully integrated into the Linux kernel distribution but is available from CITI's website. Basic SPKM-3 functionality is integrated into the Linux kernel. In May 2007, we confirmed interoperability with Hummingbird's (still incomplete) SPKM-3 implementation.

We worked with the IETF community to update the SPKM-3 RFC and move it forward. We submitted Internet drafts to the IETF, received comments, addressed them, and resubmitted.[10] Substantive changes to SPKM-3 included the following:

- Clarifying naming and error handling in RFC 2847.
- We pruned cryptographic encryption and integrity algorithms to reflect current standards.
- We addressed a problem with error tokens, which did not carry the sender's certificate information, yet included an integrity field computed using the sender's private key. Consequently, the receiver lacked the sender's public key and was thus unable to verify the sender's signature. T solve this problem, we proposed a new error token that includes the sender's certificate.
- We proposed to apply integrity protection to request token authorization data.

Once public comments settled down, we scheduled an IETF BoF session to discuss our proposal. We presented our SPKM-3 draft on November 6, 2006 at an SPKM-3 BoF session at the 67th IETF meeting in San Diego. During the meeting several other alternative X.509-based mechanisms were proposed.

Consensus was not reached at the BoF, so Sam Hartman, the IETF Security Area Director, appointed a small design group to review the proposed solutions and pick one to move forward. The design team was unable to make a firm decision about a final selection, but made it clear that SPKM-3 would not be considered. Informal consensus pointed to an alternative X.509-based GSS mechanism, called PKU2U.[11] With SPKM-3 apparently dead, we began to review PKU2U and submitted comments on PKU2U Internet drafts.

[8] KORNIEVSKAIA, O., HONEYMAN, P., DOSTER, B., AND COFFMAN, K. 2001. Kerberized credential translation: a solution to web access control. In *Proceedings 10th USENIX Security Symposium* (Washington, D.C., August 2001). USENIX Association, Berkeley, CA, pp. 235–250.

[9] EISLER, M. 2000. LIPKEY—A low infrastructure public key mechanism using SPKM. RFC 2847.

[10] ADAMSON, W. AND KORNIEVSKAIA, O. October 14, 2005. Low Infrastructure Mutual Authentication Using SPKM-3. Internet draft-adamson-nfsv4-spkm3-00. Expired April 17, 2006.

ADAMS, C. AND EISLER, M. May 31, 2006. Low Infrastructure Public Key Mechanisms: SPKM-3 and LIPKEY. Internet draft-adamson-rfc2847-bis-00. W. ADAMSON AND O. KORNIEVSKAIA, Eds. Expired December 2, 2006.

ADAMS, C. AND EISLER, M. August 18, 2006. Low Infrastructure Public Key Mechanisms: SPKM-3 and LIPKEY. Internet draft-adamson-rfc2847-bis-01. W. ADAMSON AND O. KORNIEVSKAIA, Eds. Expired February 19, 2007.

EISLER, M., ADAMSON, W., AND KORNIEVSKAIA, O. August 2006. Low Infrastructure Public Key Mechanisms: SPKM-3 and LIPKEY. Internet draft-adamson-rfc2847-bis-02. Expired February 2, 2007.

[11] ZHU, L., ALTMAN, J., AND WILLIAMS, N. July 14, 2008. Public key cryptography based user-to-user authentication. Internet draft-zhu-pku2u-07. Expires January 15, 2009.

Integrating GSI and GridNFS identity (II)

- Implement and test GSI-compatible security for the Linux NFSv4 client and server, integrate them into the Linux distribution.
- Work within the IETF to promote a standards-based extension to GSS and/or NFSv4 that is compatible with GSI.

With SPKM-3 seeming to be a dead end, we turned to Condor, a popular GSI-based Grid application, as a vehicle to advance identity integration within the Grid environment. We installed the OSG toolkit, which also uses GSI, into NFSv4. We then modified the Globus toolkit gatekeeper application in order to use NFSv4, as follows.

The globus-gatekeeper daemon, which receives GSI-secured requests for services like job management and data transfers, performs authorization checks—in the simplest case via a Gridmap file—then executes a request running as the requester. The request runs as a separate process using the identity returned by the Gridmap file (or other mapping mechanism, VOMS, Walden,[12] etc). To run a protected NFSv4-resident executable on behalf of the requester, the process needs access to the user's X.509 credentials. These credentials are stored in a place unknown to SPKM-3, so we modified the gatekeeper to store the requester's credentials in the default SPKM-3 location.

In the process of integrating NFSv4 with GSI, we discovered defects in the Linux NFSv4 implementation: the code cannot handle variable size RPC header verifiers, and the communication path between GSSD and the kernel cannot tolerate messages larger than a kernel page. We documented the former bug, but with SPKM-3 out of the picture, we did not submit a patch. The message size bug interferes with PK-based access control, in which credentials usually exceed 4 KB (due in part to sending them in text rather than binary form). We produced a patch for the problem, but it as not been pushed upstream to the mainline kernel by the kernel maintainers.

In the process of integrating Condor and NFSv4, we discovered a defect in Condor involving the transfer of user credentials. Condor provides a transfer mechanism for a job’s input and output files if a distributed file system is not available on the submit and execute nodes; when a distributed file system is available, Condor lets the user simply specify their locations. However, to use Condor and NFSv4, the user’s credential (i.e., the user’s X.509 proxy) is needed to access the files. Condor provides a way for a user to propagate credentials to the execute node, however, that forces all input and output files to be transferred to and from the execute node as well.

We also uncovered a problem with the Condor daemon. Before executing a user's job, the Condor daemon establishes standard output (UNIX stdout) and standard error output (UNIX stderr) paths for the job. However, the Condor daemon lacks appropriate credentials to establish stdout and stderr paths if they are directed to files in NFSv4. (We informed the Condor team about these limitations and they are working on a new design.)

We found other problems with Condor/NFSv4 integration. On an NFSv4 client, SPKM-3 looks for user credentials in a default location based on the user's UID. However, when the client is a Condor execute node, the UID is not meaningful, and may be one that is shared across all Condor jobs. We addressed this by using new functionality in Linux—the key ring—which allows processes to control a diverse collection of credentials. We added SPKM-3 key ring support to GSSD and tested successfully with Condor. (We also uncovered a number of bugs in the nascent Linux key ring implementation.)

In addition to integrating NFSv4 with Condor, we also worked on the problems that arise when execution lifetime is longer than the lifetime of credentials used to submit the job. For some jobs, Condor can renew a job's credentials using MyProxy.[13] We proposed an alternative that extends this functionality to any job type by using Condor's job wrapper mechanism: instead of executing a user-specified program directly, the Condor daemon executes a wrapper application that monitors the job's credentials in addition to running the user-specified program. When the monitor determines that credentials are about to expire, it uses MyProxy to refresh them. As with Condor's own renewal mechanism, this requires the user to place a long-lived proxy in MyProxy prior to submitting her Condor job.

Our solution does not handle the case in which the user's job is started after the submitting credentials expire. We have proposed to Condor team that they provide credential renewal to all job types. However, in the non-NFSv4 environment, only Grid jobs use X.509-based credentials, so the Condor team deferred making this change.

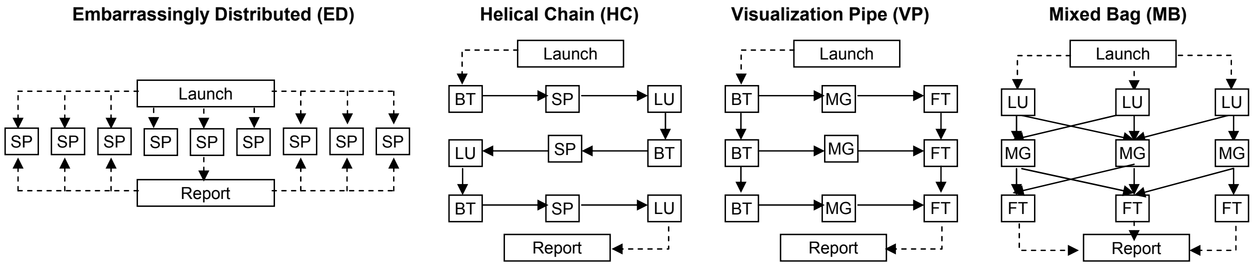
[12] Kirschner, B.A., Hacker, T.J., Adamson, W.A., and Athey, B.D. 2004. Walden: a scalable solution for grid account management. In *Proceedings 5th IEEE/ACM International Workshop on Grid Computing* (Pittsburgh, November 2004). R. Buyya, Ed. IEEE Press, Los Alamitos, CA, pp. 102–109.

[13] Novotny, J., Tuecke, S., and Welch, V. 2001. An online credential repository for the Grid: MyProxy. In *Proceedings 10th International Symposium on High Performance Distributed Computing* (San Francisco, August 2001). IEEE Press, Los Alamitos, CA, pp. 104–114.

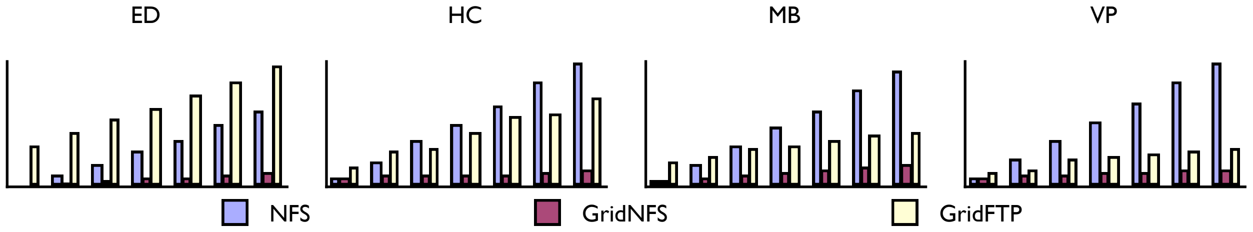
Automatic secure dynamic replication for GridNFS

- 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.

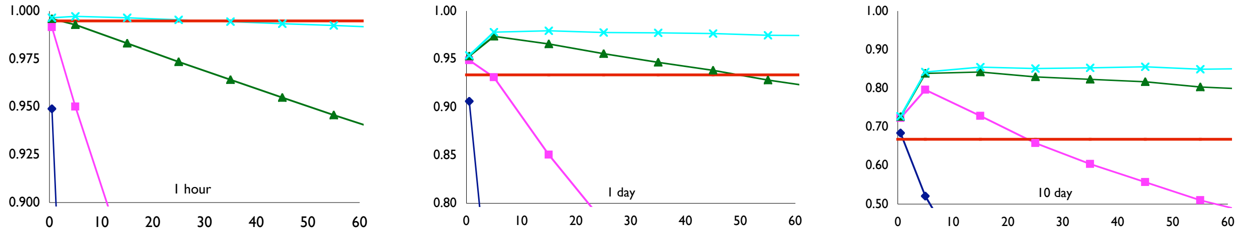
CITI provided support for read-only replication by implementing the FS_LOCATIONS attribute and ancillary utilities. We also developed an experimental server-to-server protocol for mutable replication that supports a range of consistency semantics: strict consistency, close-to-open semantics, or sequential consistency.[14,15] We evaluated the effect of different consistency models using the NAS Grid Benchmarks[16] to evaluate performance in a number of I/O distribution scenarios.



We identified tradeoffs between performance and availability that follow from the choice of the consistency model and compared performance with native NFSv4 and with GridFTP. The following graphs illustrate the performance results we found; details are in the published papers.



We also evaluated performance and availability tradeoffs in replicated file systems.[17] We examined long running tasks (1 hour, 1 day, 10 day) with varying amounts of I/O and used models of failure and correlated failure to estimate the utilization of a compute plant as a function of the distance of storage replication units.



[14] ZHANG, J., AND HONEYMAN, P. 2006. Naming, migration, and replication for NFSv4. In *Proceedings 5th International System Administration and Network Engineering Conference* (Delft, May 2006).

[15] Zhang, J., and Honeyman, P. 2008. A replicated file system for grid computing. *Concurrency and Computation: Practice and Experience* 20:9, Wiley Interscience, New York, pp. 1113–1130.

[16] FRUMKIN, M. AND VAN DER WIJNGAART, R.F. 2002. NAS Grid benchmarks: a tool for Grid space exploration. *Cluster Computing* 5:3, Kluwer Academic Publishers, The Netherlands, pp. 247–255.

[17] ZHANG, J., AND HONEYMAN, P. 2008. Performance and availability tradeoffs in replicated file systems. In *Proceedings Workshop on Resiliency in High Performance Computing* (Lyon, May 2008).

We found that for very short tasks (1 hour), replication offered no utilization advantage over simply restarting a job (the **red line** in the charts above), but as running times increased, storage replication does improve utilization for jobs with modest—and even intensive—write loads. Furthermore, we found that for long-running jobs (10 days), correlated failures are high enough to prescribe distant replication servers.

Tune GridNFS for metropolitan- and wide-area performance

- 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.

We built a performance test rig for measuring and optimizing GridNFS performance in high-speed, high-latency environments from the following parts:

- Rackmount Pro RM5024 5U SAS chassis with 1350 W (3 + 1 redundant, hot-swap) power supplies
- Supermicro X7DBE motherboard
- Two 64-bit Xeon E5345 quad core processors @ 2.33 GHz, 4x2 MB cache, 1.333 GHz FSB
- Eight Kingston 2 GB 240-pin FB-DIMM ECC DDR2 667 MHz DRAM (PC2 5300) dual channel
- Two Areca ARC-1231ML 12-port PCI-Express (x8) to SATA II RAID adapter
- 24 Western Digital Caviar SE16 WD3200AAKS (320 GB, 7200 RPM, SATA, 3 Gbps, 16 MB cache)
- Western Digital Caviar WD RE WD1600YS system disk (160 GB, 7200 RPM, SATA, 3 Gbps, 16 MB cache)
- Myricom 10G-PCIE-8A-R PCI-Express (x8) 10-G Ethernet NIC, 10G-XFP-SR optical transceiver

From our experiments—preliminary and unpublished—so far, we have identified a number of challenges in maximizing disk-to-disk transfer rates. However, after a fair amount of TCP/IP tuning, we are able to saturate a 10 GbE network on memory-to-memory transfers. Disk array and LVM tuning are ongoing: we can achieve 800 MBps local NFS I/O when writing and 1.2 GBps when reading. We have more work to do in tuning GridNFS over long-haul networks: our best read performance is 590 MBps and our best write performance is 380 MBps.

GridNFS project publications

1. HONEYMAN, P., ADAMSON, W.A., AND MCKEE, S. 2005. GridNFS: global storage for global collaborations. CITI Technical Report 05-3. Center for Information Technology Integration, University of Michigan, Ann Arbor, May 2005.

2. HONEYMAN, P., ADAMSON, W.A., AND MCKEE, S. 2005. GridNFS: global storage for global collaborations. In *Proceedings IEEE-CS International Symposium on Local to Global Data Interoperability—Challenges and Technologies* (Cagliari, June 2005). IEEE Press, Los Alamitos, CA.

3. ADAMSON, W. AND KORNIEVSKAIA, O. 2005. Low Infrastructure Mutual Authentication Using SPKM-3. Internet draft-adamson-nfsv4-spkm3-00. October 14, 2005, expired April 17, 2006.

4. ZHANG, J., AND HONEYMAN, P. 2006. Naming, migration, and replication for NFSv4. CITI Technical Report 06-1. Center for Information Technology Integration, University of Michigan, Ann Arbor, January 2006.

5. ZHANG, J., AND HONEYMAN, P. 2006. Reliable replication at low cost. CITI Technical Report 06-2. Center for Information Technology Integration, University of Michigan, Ann Arbor, January 2006.

6. ADAMS, C. AND EISLER, M. May 31, 2006. Low Infrastructure Public Key Mechanisms: SPKM-3 and LIPKEY. Internet draft-adamson-rfc2847-bis-00. W. ADAMSON AND O. KORNIEVSKAIA, Eds. Expired December 2, 2006.

7. ZHANG, J., AND HONEYMAN, P. 2006. Naming, migration, and replication for NFSv4. In *Proceedings 5th International System Administration and Network Engineering Conference*, Delft, May 2006.

8. ZHANG, J., AND HONEYMAN, P. 2006. Hierarchical replication control. CITI Technical Report 06-3. Center for Information Technology Integration, University of Michigan, Ann Arbor, May 2006.

9. ADAMSON, W.A., HILDEBRAND, D., HONEYMAN, P., MCKEE, S., AND ZHANG, J. 2006. Extending NFSv4 for petascale data management. CITI Technical Report 06-5. Center for Information Technology Integration, University of Michigan, Ann Arbor, May 2006.

10. ADAMSON, W.A., HILDEBRAND, D., HONEYMAN, P., MCKEE, S., AND ZHANG, J. 2006. Extending NSFv4 for petascale data management. In *Proceedings HPDC Workshop on Next-Generation Distributed Data Management* (Paris, June 2006).

11. Adams, C. and Eisler, M. 2006. Low Infrastructure Public Key Mechanisms: SPKM-3 and LIPKEY. Internet draft-adamson-rfc2847-bis-01. W. Adamson and O. Kornievskaia, Eds. August 18, 2006, expired February 19, 2007.

12. Eisler, M., Adamson, W., and Kornievskaia, O. 2006. Low Infrastructure Public Key Mechanisms: SPKM-3 and LIPKEY. Internet draft-adamson-rfc2847-bis-02. August 2006, expired February 2, 2007.

13. Zhang, J., and Honeyman, P. 2006. Hierarchical replication control in a global file system. CITI Technical Report 06-7. Center for Information Technology Integration, University of Michigan, Ann Arbor, September 2006.

14. Zhang, J., and Honeyman, P. 2006. NFSv4 replication for grid storage middleware. In *Proceedings 4th International Workshop on Middleware for Grid Computing* (Melbourne, VIC, November 2006).

15. Zhang, J., and Honeyman, P. 2007. Hierarchical replication control in a global file system. In *Proceedings 7th IEEE International Symposium on Cluster Computing and the Grid* (Rio de Janeiro, May 2007). IEEE Press, Los Alamitos.

16. Zhang, J. 2007. *Network transparency in wide area collaborations*. Doctoral dissertation, University of Michigan, Ann Arbor, May 2007.

17. Adamson, W.A. 2007. NFSv4 and petascale data management. In *Proceedings HPDC Joint EGEE and OSG Workshop on Data Handling in Production Grids*, Monterey Bay, June 2007.

18. Zhang, J., and Honeyman, P. 2007. Performance and availability tradeoffs in replicated file systems. CITI Technical Report 07-3. Center for Information Technology Integration, University of Michigan, Ann Arbor, October 2007.

19. Zhang, J., and Honeyman, P. 2008. Performance and availability tradeoffs in replicated file systems. In *Proceedings Workshop on Resiliency in High Performance Computing* (Lyon, May 2008).

20. Zhang, J., and Honeyman, P. 2008. A replicated file system for grid computing. *Concurrency and Computation: Practice and Experience* 20:9, Wiley InterScience, New York, pp. 1113–1130.

Conclusion

The GridNFS project has been very successful in leveraging work on NFSv4 by researchers and developers at CITI and in other laboratories (academic, national, international, and industrial). Much work is ongoing; much remains:

- Notwithstanding CITI's success in moving GridNFS research and development into the mainline Linux kernel, broad-scale adoption of GridNFS capabilities awaits the embrace by vendors of NFSv4 as the default version of NFS in their distributions.
- GridNFS/dCache testing and WAN benchmarking is under way by University of Michigan researchers working in close cooperation with colleagues at DESY. The new Chimera name service provides a GridNFS interface to dCache storage.
- Concurrent with CITI's GridNFS research and development, new architectural features for NFSv4 have emerged: pNFS[18] provides parallel striped files across file, object, or block storage servers. Integrating GridNFS with pNFS poses real challenges but promises enormous rewards.
- The IETF position on public key mechanisms for NFSv4 is muddled. Symmetric key support (through Kerberos) may provide the needed leverage through protocols such as PKINIT[19] and PKU2U.

[18] Shepler, S., Eisler, M., and Noveck, D., Eds. May 12, 2008. NFS Version 4 Minor Version 1. Internet draft-ietf-nfsv4-minorversion1-23. Expires: November 13, 2008.

[19] Zhu, L. and Tung, B. 2006. Public key cryptography for initial authentication in Kerberos (PKINIT). RFC 4556.