WIKIPEDIA

Network File System

Network File System (**NFS**) is a distributed file system protocol originally developed by Sun Microsystems (Sun) in 1984,[1] allowing a user on a client computer to access files over a computer network much like local storage is accessed. NFS, like many other protocols, builds on the Open Network Computing Remote Procedure Call (ONC RPC) system. NFS is an open standard defined in a Request for Comments (RFC), allowing anyone to implement the protocol.

Contents

Versions and variations

Sun used version 1 only for in-house experimental purposes. When the development team added substantial changes to NFS version 1 and released it outside of Sun, they decided to release the new version as v2, so that version interoperation and RPC version fallback could be tested.[2][3]

NFSv2

Version 2 of the protocol (defined in RFC 1094, March 1989) originally operated only over User Datagram Protocol (UDP). Its designers meant to keep the server side stateless, with locking (for example) implemented outside of the core protocol. People involved in the creation of NFS version 2 include Russel Sandberg, Bob Lyon, Bill Joy, Steve Kleiman, and others.[1][4]

The Virtual File System interface allows a modular implementation, reflected in a simple protocol. By February 1986, implementations were demonstrated for operating systems such as System V release 2, DOS, and VAX/VMS using Eunice.[4] NFSv2 only allows the first 2 GB of a file to be read due to 32-bit limitations.

NFSv3

Version 3 (RFC 1813, June 1995) added:

- support for 64-bit file sizes and offsets, to handle files larger than 2 gigabytes (GB);
- support for asynchronous writes on the server, to improve write performance;
- additional file attributes in many replies, to avoid the need to re-fetch them;
- a READDIRPLUS operation, to get file handles[5] and attributes along with file names when scanning a directory;
- assorted other improvements.

The first NFS Version 3 proposal within Sun Microsystems was created not long after the release of NFS Version 2. The principal motivation was an attempt to mitigate the performance issue of the synchronous write operation in NFS Version 2.[6] By July 1992, implementation practice had solved many shortcomings of NFS Version 2, leaving only lack of large file support (64-bit file sizes and offsets) a pressing issue. This became an acute pain point for Digital Equipment Corporation with the introduction of a 64-bit version of Ultrix to support their newly released 64-bit RISC processor, the Alpha 21064. At the time of introduction of Version 3, vendor support for TCP as a transport-layer protocol began increasing. While several vendors had already added support for NFS Version 2 with TCP as a transport, Sun Microsystems added support for TCP as a transport for NFS at the same time it added support for Version 3. Using TCP as a transport made using NFS over a WAN more feasible, and allowed the use of larger read and write transfer sizes beyond the 8 KB limit imposed by User Datagram Protocol.

NFSv4

Version 4 (RFC 3010, December 2000; revised in RFC 3530, April 2003 and again in RFC 7530, March 2015), influenced by Andrew File System (AFS) and Server Message Block (SMB, also termed CIFS), includes performance improvements, mandates strong security, and introduces a stateful protocol.[7] Version 4 became the first version developed with the Internet Engineering Task Force (IETF) after Sun Microsystems handed over the development of the NFS protocols.

NFS version 4.1 (RFC 5661, January 2010; revised in RFC 8881, August 2020) aims to provide protocol support to take advantage of clustered server deployments including the ability to provide scalable parallel access to files distributed among multiple servers (pNFS extension). Version 4.1 includes Session trunking mechanism (Also known as NFS Multipathing) and available in some enterprise solutions as VMware ESXi.

NFS version 4.2 (RFC 7862) was published in November 2016[8] with new features including: server-side clone and copy, application I/O advise, sparse files, space reservation, application data block (ADB), labeled NFS with sec_label that accommodates any MAC security system, and two new operations for pNFS (LAYOUTERROR and LAYOUTSTATS).

One big advantage of NFSv4 over its predecessors is that only one UDP or TCP port, 2049, is used to run the service, which simplifies using the protocol across firewalls.

Other extensions

WebNFS, an extension to Version 2 and Version 3, allows NFS to integrate more easily into Web-browsers and to enable operation through firewalls. In 2007 Sun Microsystems open-sourced their client-side WebNFS implementation.[9]

Various side-band protocols have become associated with NFS. Note:

- the byte-range advisory Network Lock Manager (NLM) protocol (added to support UNIX System V file locking APIs)
- the remote quota-reporting (RQUOTAD) protocol, which allows NFS users to view their data-storage quotas on NFS servers
- NFS over RDMA, an adaptation of NFS that uses remote direct memory access (RDMA) as a transport[10][11]
- NFS-Ganesha, an NFS server, running in user-space and supporting various file systems like GPFS/Spectrum Scale, CephFS via respective FSAL (File System Abstraction Layer) modules. The CephFS FSAL supported using libcephfs[12]
- Trusted NFS (TNFS)[13]

Platforms

NFS is often used with Unix operating systems (such as Solaris, AIX, HP-UX), Apple's macOS, and Unix-like operating systems (such as Linux and FreeBSD). It is also available to operating systems such as Acorn RISC OS,[14] AmigaOS, the classic Mac OS, OpenVMS,[3] MS-DOS,[15] Microsoft Windows,[16] OS/2,[17] ArcaOS,[18] Novell NetWare,[19] and IBM i.[20] Alternative remote file access protocols include the Server Message Block (SMB, also termed CIFS), Apple Filing Protocol (AFP), NetWare Core Protocol (NCP), and OS/400 File Server file system (QFileSvr.400).

SMB and NetWare Core Protocol (NCP) occur more often than NFS on systems running Microsoft Windows; AFP occurs more often than NFS in Apple Macintosh systems; and QFileSvr.400 occurs more often in IBM i systems. Haiku in 2012 added NFSv4 support as part of a Google Summer of Code project.
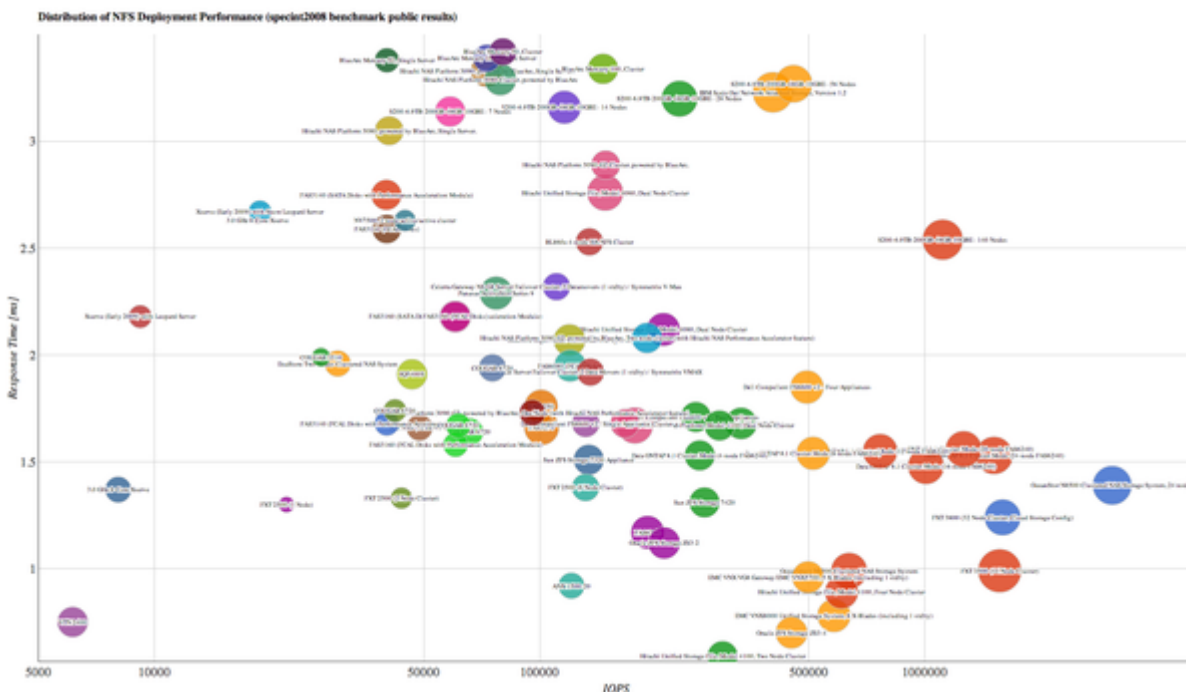
NFS SPECsfs2008 performance comparison, as of 22 November 2013

Typical implementation

Assuming a Unix-style scenario in which one machine (the client) needs access to data stored on another machine (the NFS server):

1. The server implements NFS daemon processes, running by default as `nfsd`, to make its data generically available to clients.
2. The server administrator determines what to make available, exporting the names and parameters of directories, typically using the `/etc/exports` configuration file and the `exportfs` command.
3. The server security-administration ensures that it can recognize and approve validated clients.
4. The server network configuration ensures that appropriate clients can negotiate with it through any firewall system.
5. The client machine requests access to exported data, typically by issuing a `mount` command. (The client asks the server (rpcbind) which port the NFS server is using, the client connects to the NFS server (nfsd), nfsd passes the request to mountd)
6. If all goes well, users on the client machine can then view and interact with mounted filesystems on the server within the parameters permitted.

Note that automation of the NFS mounting process may take place — perhaps using `/etc/fstab` and/or automounting facilities.

Protocol development

During the development of the ONC protocol (called SunRPC at the time), only Apollo's Network Computing System (NCS) offered comparable functionality. Two competing groups developed over fundamental differences in the two remote procedure call systems. Arguments focused on the method for data-encoding — ONC's External Data Representation (XDR) always rendered integers in big-endian order, even if both peers of the connection had little-endian machine-architectures, whereas NCS's method attempted to avoid byte-swap whenever two peers shared a common endianness in their machine-architectures. An industry-group called the Network Computing Forum formed (March 1987) in an (ultimately unsuccessful) attempt to reconcile the two network-computing environments.

In 1987, Sun and AT&T announced they would jointly develop AT&T's UNIX System V Release 4.[21] This caused many of AT&T's other licensees of UNIX System to become concerned that this would put Sun in an advantaged position, and ultimately led to Digital Equipment, HP, IBM, and others forming the Open Software Foundation (OSF) in 1988. Ironically, Sun and AT&T had formerly competed over Sun's NFS versus AT&T's Remote File System (RFS), and the quick adoption of NFS over RFS by Digital Equipment, HP, IBM, and many other computer vendors tipped the majority of users in favor of NFS. NFS interoperability was aided by events called "Connectathons" starting in 1986 that allowed vendor-neutral testing of implementations with each other.[22] OSF adopted the Distributed Computing Environment (DCE) and the DCE Distributed File System (DFS) over Sun/ONC RPC and NFS. DFS used DCE as the RPC, and DFS derived from the Andrew File System (AFS); DCE itself derived from a suite of technologies, including Apollo's NCS and Kerberos.

1990s

Sun Microsystems and the Internet Society (ISOC) reached an agreement to cede "change control" of ONC RPC so that the ISOC's engineering-standards body, the Internet Engineering Task Force (IETF), could publish standards documents (RFCs) related to ONC RPC protocols and could extend ONC RPC. OSF attempted to make DCE RPC an IETF standard, but ultimately proved unwilling to give up change control.

Later, the IETF chose to extend ONC RPC by adding a new authentication flavor based on Generic Security Services Application Program Interface (GSSAPI), RPCSEC GSS, to meet IETF requirements that protocol standards have adequate security.

Later, Sun and ISOC reached a similar agreement to give ISOC change control over NFS, although writing the contract carefully to exclude NFS version 2 and version 3. Instead, ISOC gained the right to add new versions to the NFS protocol, which resulted in IETF specifying NFS version 4 in 2003.

2000s

By the 21st century, neither DFS nor AFS had achieved any major commercial success as compared to SMB-CIFS or NFS. IBM, which had formerly acquired the primary commercial vendor of DFS and AFS, Transarc, donated most of the AFS source code to the free software community in 2000. The OpenAFS project lives on. In early 2005, IBM announced end of sales for AFS and DFS.

In January, 2010, Panasas proposed an NFSv4.1 based on their *Parallel NFS* (pNFS) technology claiming to improve data-access parallelism[23] capability. The NFSv4.1 protocol defines a method of separating the filesystem meta-data from file data location; it goes beyond the simple name/data separation by striping the data amongst a set of data servers. This differs from the traditional NFS server which holds the names of files and their data under the single umbrella of the server. Some products are multi-node NFS servers, but the participation of the client in separation of meta-data and data is limited.

The NFSv4.1 pNFS server is a set of server resources or components; these are assumed to be controlled by the meta-data server.

The pNFS client still accesses one meta-data server for traversal or interaction with the namespace; when the client moves data to and from the server it may directly interact with the set of data servers belonging to the pNFS server collection. The NFSv4.1 client can be enabled to be a direct participant in the exact location of file data and to avoid solitary interaction with one NFS server when moving data.

In addition to pNFS, NFSv4.1 provides:

- Sessions
- Directory Delegation and Notifications
- Multi-server Namespace
- access control lists and discretionary access control
- Retention Attributions
- SECINFO_NO_NAME

See also

- 9P (protocol) - Plan 9 Filesystem Protocol
- Alluxio
- Andrew File System
- BeeGFS, the parallel file system
- CacheFS - a caching mechanism for Linux NFS clients
- Hadoop Distributed File System (HDFS)
- Kerberos (protocol)
- Network Information Service
- Remote File System

- Root squash
- Samba (software)
- Secure Shell Filesystem - mount a remote directory using only a ssh login on the remote computer
- Server Message Block
- Shared resource
- TCP Wrapper

References

1. Russel Sandberg; David Goldberg; Steve Kleiman; Dan Walsh; Bob Lyon (1985). "Design and Implementation of the Sun Network Filesystem". USENIX. CiteSeerX 10.1.1.14.473 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.473).
2. *NFS Illustrated* (2000) by Brent Callaghan - ISBN 0-201-32570-5
3. "HP TCP/IP Services for OpenVMS Management" (http://h41379.www4.hpe.com/doc/83final/6526/6526pro_052.html). *h41379.www4.hpe.com*. HP. Retrieved 24 September 2016.
4. Russel Sandberg. "The Sun Network Filesystem: Design, Implementation and Experience" (http://www.cse.buffalo.edu/faculty/tkosar/cse710_spring13/papers/nfs.pdf) (PDF). *Technical Report*. Sun Microsystems.
5. Arpaci-Dusseau, Remzi; Arpaci-Dusseau, Andrea (March 2015). *Operating Systems: Three Easy Pieces* (http://pages.cs.wisc.edu/~remzi/OSTEP/dist-nfs.pdf) (PDF) (.9 ed.). Arpaci-Dusseau Books. p. 5. Retrieved 8 November 2017.
6. Brian Pawlowski; Chet Juszczak; Peter Staubach; Carl Smith; Diane Lebel; David Hitz (1994). "NFS Version 3 Design and Implementation" (https://www.usenix.org/legacy/publications/library/proceedings/bos94/full_papers/pawlowski.ps). USENIX.
7. "NFS Version 4" (http://www.usenix.org/events/usenix05/tech/italks.html#nFSv4). USENIX. 2005-04-14.
8. Haynes, Thomas (2016-11-01). "NFS Version 4 Minor Version 2" (https://tools.ietf.org/html/rfc7862).
9. yanfs.dev.java.net (https://yanfs.dev.java.net/)
10. Tom Talpey (February 28, 2006). "NFS/RDMA Implementation(s) Update" (https://web.archive.org/web/20110512180716/http://www.connectathon.org/talks06/talpey-cthon06-nfs-rdma.pdf) (PDF). Network Appliance, Inc. Archived from the original (http://www.connectathon.org/talks06/talpey-cthon06-nfs-rdma.pdf) (PDF) on May 12, 2011.
11. Brent Callaghan (January 28, 2002). "NFS over RDMA" (http://www.usenix.org/events/fast02/wips/callaghan.pdf) (PDF). Sun Microsystems.
12. Singh, Karan (2016). "4: Working with the Ceph Filesystem". *Ceph Cookbook* (https://books.google.com/books?id=C2xLDAAAQBAJ). Birmingham: Packt Publishing Ltd. p. 110. ISBN 9781784397364. Retrieved 2017-03-21. "NFS-Ganesha is an NFS server that runs in user space and supports the CephFS FSAL (File System Abstraction Layer) using libcephfs."
13. Glover <fglover@zk3.dec.com>, Fred. "A Specification of Trusted NFS (TNFS) Protocol Extensions" (https://tools.ietf.org/html/draft-ietf-tnfs-spec-01.html). *tools.ietf.org*.
14. [1] (http://www.cp15.org/networking/)
15. "Other Software by SUN Microsystems" (http://www.computinghistory.org.uk/det/12549/PC-NFS%20version%205.0/). *www.computinghistory.org.uk*. The Centre for Computing History. Retrieved 24 September 2016.
16. "Introduction to Microsoft Windows Services for UNIX 3.5" (https://technet.microsoft.com/en-us/library/bb463212.aspx). *technet.microsoft.com*. Microsoft. Retrieved 24 September 2016.

17. "NTFS plugin for NetDrive" (https://ecsoft2.org/network-file-system-nfs-plugin-netdrive). *ecsoft2.org*. Retrieved 2020-09-22.
18. "NetDrive for OS/2" (https://www.arcanoae.com/shop/netdrive-for-os2/). *arcanoae.com*. Retrieved 2020-09-22.
19. "NFS Gateway for NetWare 6.5" (https://www.novell.com/products/openenterpriseserver/features/nfs-gateway-for-netware65.html). *www.novell.com*. Novell. Retrieved 24 September 2016.
20. "OS/400 Network File System Support" (http://publib.boulder.ibm.com/html/as400/v5r1/ic2911/books/c4157141.pdf) (PDF). *publib.boulder.ibm.com*. IBM. Retrieved 24 September 2016.
21. Carole Patton (1987-10-26). "AT&T to License Sun Microsystems' SPARC Chip" (https://books.google.com/books?id=_z4EAAAAMBAJ&pg=PA37). *InfoWorld*. p. 37. Retrieved 2019-07-16.
22. "What is Connectathon?" (https://web.archive.org/web/19990128152940/http://www.connectathon.org/#whatis). *Original Connectathon.Org web site*. Archived from the original (http://www.connectathon.org/) on January 28, 1999.
23. "pNFS" (http://www.pnfs.com). Panasas. Retrieved August 4, 2013.

External links

- RFCs:
    - RFC 5661 (https://datatracker.ietf.org/doc/html/rfc5661) - Network File System (NFS) Version 4 Minor Version 1 Protocol
    - RFC 5403 (https://datatracker.ietf.org/doc/html/rfc5403) - RPCSEC_GSS Version 2
    - RFC 3530 (https://datatracker.ietf.org/doc/html/rfc3530) - NFS Version 4 Protocol Specification
    - RFC 2054 (https://datatracker.ietf.org/doc/html/rfc2054) - WebNFS Specification
    - RFC 2339 (https://datatracker.ietf.org/doc/html/rfc2339) - Sun/ISOC NFS Change Control Agreement
    - RFC 2203 (https://datatracker.ietf.org/doc/html/rfc2203) - RPCSEC_GSS Specification
    - RFC 1813 (https://datatracker.ietf.org/doc/html/rfc1813) - NFS Version 3 Protocol Specification
    - RFC 1790 (https://datatracker.ietf.org/doc/html/rfc1790) - Sun/ISOC ONC RPC Change Control Agreement
    - RFC 1094 (https://datatracker.ietf.org/doc/html/rfc1094) - NFS Version 2 Protocol Specification
- Various resources:
    - IETF: Network File System Version 4 (nfsv4) Charter (https://web.archive.org/web/20080216083308/http://www.ietf.org/html.charters/nfsv4-charter.html)
    - Linux NFS Overview, FAQ and HOWTO Documents (http://nfs.sourceforge.net/)
    - NFS operation explained with sequence diagrams (http://www.eventhelix.com/RealtimeMantra/Networking/NFS_Protocol_Sequence_Diagram.pdf) Archived (https://web.archive.org/web/20200922193017/http://www.eventhelix.com/RealtimeMantra/Networking/NFS_Protocol_Sequence_Diagram.pdf) 2020-09-22 at the Wayback Machine
- Review of "Why NFS Sucks" Paper from the 2006 Linux Symposium (http://nfsworld.blogspot.com/2006/10/review-of-why-nfs-sucks-paper-from.html) by Mike Eisler, October 27, 2006