Name-Prashant Walunj B350

Assignment 3 (Operating System)

* *Network File system(NFS)*

Network File System (**NFS**) is a distributed file system protocol originally developed by Sun Microsystems in 1984, 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. The NFS is an open standard defined in Request for Comments (RFC), allowing anyone to implement the protocol.

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

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.

Name-Prashant Walunj B350

Assignment 3 (Operating System)

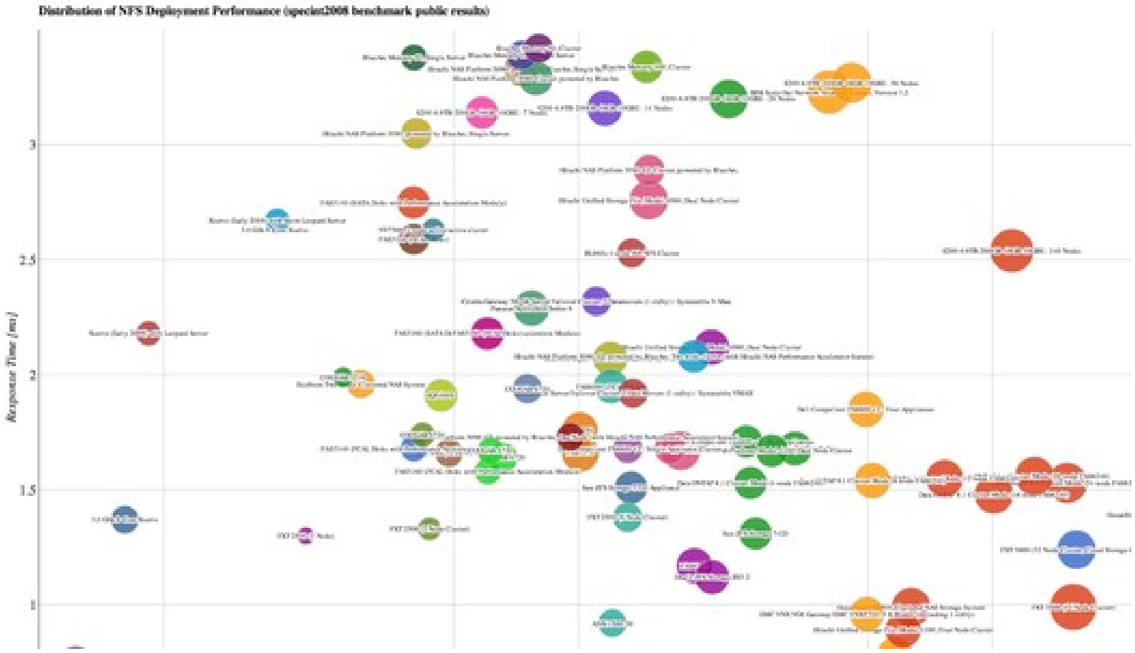
NFSv4

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.

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,[13] the classic Mac OS, OpenVMS,[3] MS-DOS,[14] Microsoft Windows,[15] Novell NetWare,[16] and IBM AS/400.[17] 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 AS/400 systems. Haiku recently[when?] added NFSv4 support as part of a Google Summer of Code project.



Name-Prashant Walunj B350

Assignment 3 (Operating System)

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.