NAME

pNFS - NFS Version 4.1 and 4.2 Parallel NFS Protocol

DESCRIPTION

The NFSv4.1 and NFSv4.2 client and server provides support for the pNFS specification; see *Network File System (NFS) Version 4 Minor Version 1 Protocol RFC 5661, Network File System (NFS) Version 4 Minor Version 2 Protocol RFC 7862 and Parallel NFS (pNFS) Flexible File Layout RFC 8435.* A pNFS service separates Read/Write operations from all other NFSv4.1 and NFSv4.2 operations, which are referred to as Metadata operations. The Read/Write operations are performed directly on the Data Server (DS) where the file's data resides, bypassing the NFS server. All other file operations are performed on the NFS server, which is referred to as a Metadata Server (MDS). NFS clients that do not support pNFS perform Read/Write operations on the MDS, which acts as a proxy for the appropriate DS(s).

The NFSv4.1 and NFSv4.2 protocols provide two pieces of information to pNFS aware clients that allow them to perform Read/Write operations directly on the DS.

The first is DeviceInfo, which is static information defining the DS server. The critical piece of information in DeviceInfo for the layout types supported by FreeBSD is the IP address that is used to perform RPCs on the DS. It also indicates which version of NFS the DS supports, I/O size and other layout specific information. In the DeviceInfo, there is a DeviceID which, for the FreeBSD server is unique to the DS configuration and changes whenever the nfsd daemon is restarted or the server is rebooted.

The second is the layout, which is per file and references the DeviceInfo to use via the DeviceID. It is for a byte range of a file and is either Read or Read/Write. For the FreeBSD server, a layout covers all bytes of a file. A layout may be recalled by the MDS using a LayoutRecall callback. When a client returns a layout via the LayoutReturn operation it can indicate that error(s) were encountered while doing I/O on the DS, at least for certain layout types such as the Flexible File Layout.

The FreeBSD client and server supports two layout types.

The File Layout is described in RFC5661 and uses the NFSv4.1 or NFSv4.2 protocol to perform I/O on the DS. It does not support client aware DS mirroring and, as such, the FreeBSD server only provides File Layout support for non-mirrored configurations.

The Flexible File Layout allows the use of the NFSv3, NFSv4.0, NFSv4.1 or NFSv4.2 protocol to perform I/O on the DS and does support client aware mirroring. As such, the FreeBSD server uses Flexible File Layout layouts for the mirrored DS configurations. The FreeBSD server supports the "tightly coupled" variant and all DSs allow use of the NFSv4.2 or NFSv4.1 protocol for I/O operations.

Clients that support the Flexible File Layout will do writes and commits to all DS mirrors in the mirror set.

A FreeBSD pNFS service consists of a single MDS server plus one or more DS servers, all of which are FreeBSD systems. For a non-mirrored configuration, the FreeBSD server will issue File Layout layouts by default. However that default can be set to the Flexible File Layout by setting the sysctl(8) sysctl "vfs.nfsd.default_flexfile" to one. Mirrored server configurations will only issue Flexible File Layouts. pNFS clients mount the MDS as they would a single NFS server.

A FreeBSD pNFS client must be running the nfscbd(8) daemon and use the mount options "nfsv4,minorversion=2,pnfs or" "nfsv4,minorversion=1,pnfs".

When files are created, the MDS creates a file tree identical to what a single NFS server creates, except that all the regular (VREG) files will be empty. As such, if you look at the exported tree on the MDS directly on the MDS server (not via an NFS mount), the files will all be of size zero. Each of these files will also have two extended attributes in the system attribute name space:

pnfsd.dsfile - This extended attribute stores the information that the
MDS needs to find the data file on a DS(s) for this file.
pnfsd.dsattr - This extended attribute stores the Size, AccessTime,
ModifyTime, Change and SpaceUsed attributes for the file.

For each regular (VREG) file, the MDS creates a data file on one (or on N of them for the mirrored case, where N is the mirror_level) of the DS(s) where the file’s data will be stored. The name of this file is the file handle of the file on the MDS in hexadecimal at time of file creation. The data file will have the same file ownership, mode and NFSv4 ACL (if ACLs are enabled for the file system) as the file on the MDS, so that permission checking can be done on the DS. This is referred to as "tightly coupled" for the Flexible File Layout.

For pNFS aware clients, the service generates File Layout or Flexible File Layout layouts and associated DeviceInfo. For non-pNFS aware NFS clients, the pNFS service appears just like a normal NFS service. For the non-pNFS aware client, the MDS will perform I/O operations on the appropriate DS(s), acting as a proxy for the non-pNFS aware client. This is also true for NFSv3 and NFSv4.0 mounts, since these are always non-pNFS aware.

It is possible to assign a DS to an MDS exported file system so that it will store data for files on the MDS exported file system. If a DS is not assigned to an MDS exported file system, it will store data for files on all exported file systems on the MDS.

If mirroring is enabled, the pNFS service will continue to function when DS(s) have failed, so long is

there is at least one DS still operational that stores data for files on all of the MDS exported file systems. After a disabled mirrored DS is repaired, it is possible to recover the DS as a mirror while the pNFS service continues to function.

See pnfsserver(4) for information on how to set up a FreeBSD pNFS service.

SEE ALSO

nfsv4(4), pnfsserver(4), exports(5), fstab(5), rc.conf(5), nfscbd(8), nfsd(8), nfsuserd(8), pnfsdscopymr(8), pnfsdsfile(8), pnfsdskill(8)

BUGS

Linux kernel versions prior to 4.12 only supports NFSv3 DSs in its client and will do all I/O through the MDS. For Linux 4.12 kernels, support for NFSv4.1 DSs was added, but I have seen Linux client crashes when testing this client. For Linux 4.17-rc2 kernels, I have not seen client crashes during testing, but it only supports the "loosely coupled" variant. To make it work correctly when mounting the FreeBSD server, you must set the sysctl "vfs.nfsd.flexlinuxhack" to one so that it works around the Linux client driver’s limitations. Wihout this sysctl being set, there will be access errors, since the Linux client will use the authenticator in the layout (uid=999, gid=999) and not the authenticator specified in the RPC header.

Linux 5.n kernels appear to be patched so that it uses the authenticator in the RPC header and, as such, the above sysctl should not need to be set.

Since the MDS cannot be mirrored, it is a single point of failure just as a non pNFS server is.