NFS

Enterprise applications

NetApp
April 25, 2024

Table of Contents

NFS

NFS configuration for Oracle databases

NetApp has been providing enterprise-grade NFS storage for over 30 years, and its use is growing with the push toward cloud-based infrastructures because of it's simplicity.

The NFS protocol includes multiple versions with varying requirements. For a complete description of NFS configuration with ONTAP, please see TR-4067 NFS on ONTAP Best Practices. The following sections cover some of the more critical requirements and common user errors.

NFS versions

The operating system NFS client must be supported by NetApp.

- NFSv3 is supported with OSs that follow the NFSv3 standard.
- NFSv3 is supported with the Oracle dNFS client.
- NFSv4 is supported with all OSs that follow the NFSv4 standard.
- NFSv4.1 and NFSv4.2 require specific OS support. Consult the NetApp IMT for supported OSs.
- Oracle dNFS support for NFSv4.1 requires Oracle 12.2.0.2 or higher.

> The NetApp support matrix for NFSv3 and NFSv4 does not include specific operating systems. All OSs that obey the RFC are generally supported. When searching the online IMT for NFSv3 or NFSv4 support, do not select a specific OS because there will be no matches displayed. All OSs are implicitly supported by the general policy.

Linux NFSv3 TCP slot tables

TCP slot tables are the NFSv3 equivalent of host bus adapter (HBA) queue depth. These tables control the number of NFS operations that can be outstanding at any one time. The default value is usually 16, which is far too low for optimum performance. The opposite problem occurs on newer Linux kernels, which can automatically increase the TCP slot table limit to a level that saturates the NFS server with requests.

For optimum performance and to prevent performance problems, adjust the kernel parameters that control the TCP slot tables.

Run the `sysctl -a | grep tcp.*.slot_table` command, and observe the following parameters:

```
# sysctl -a | grep tcp.*.slot_table
sunrpc.tcp_max_slot_table_entries = 128
sunrpc.tcp_slot_table_entries = 128
```

All Linux systems should include `sunrpc.tcp_slot_table_entries`, but only some include `sunrpc.tcp_max_slot_table_entries`. They should both be set to 128.

Caution
Failure to set these parameters may have significant effects on performance. In some cases, performance is limited because the linux OS is not issuing sufficient I/O. In other cases, I/O latencies increases as the linux OS attempts to issue more I/O than can be serviced.

ADR and NFS

Some customers have reported performance problems resulting from an excessive amount of I/O on data in the `ADR` location. The problem does not generally occur until a lot of performance data has accumulated. The reason for the excessive I/O is unknown, but this problem appears to be a result of Oracle processes repeatedly scanning the target directory for changes.

Removal of the `noac` and/or `actimeo=0` mount options allows host OS caching to occur and reduces storage I/O levels.

NetApp recommends to not place `ADR` data on a file system with `noac` or `actimeo=0` because performance problems are likely. Separate `ADR` data into a different mount point if necessary.

nfs-rootonly and mount-rootonly

ONTAP includes an NFS option called `nfs-rootonly` that controls whether the server accepts NFS traffic connections from high ports. As a security measure, only the root user is permitted to open TCP/IP connections using a source port below 1024 because such ports are normally reserved for OS use, not user processes. This restriction helps ensure that NFS traffic is from an actual operating system NFS client, and not a malicious process emulating an NFS client. The Oracle dNFS client is a userspace driver, but the process runs as root, so it is generally not required to change the value of `nfs-rootonly`. The connections is made from low ports.

The `mount-rootonly` option only applies to NFSv3. It controls whether the RPC MOUNT call be accepted from ports greater than 1024. When dNFS is used, the client is again running as root, so it able to open ports below 1024. This parameter has no effect.

Processes opening connections with dNFS over NFS versions 4.0 and higher do not run as root and therefore require ports over 1024. The `nfs-rootonly` parameter must be set to disabled for dNFS to complete the connection.

If `nfs-rootonly` is enabled, the result is a hang during the mount phase opening dNFS connections. The sqlplus output looks similar to this:

```
SQL>startup
ORACLE instance started.
Total System Global Area 4294963272 bytes
Fixed Size                  8904776 bytes
Variable Size             822083584 bytes
Database Buffers         3456106496 bytes
Redo Buffers                7868416 bytes
```

The parameter can be changed as follows:

```
Cluster01::> nfs server modify -nfs-rootonly disabled
```

In rare situations, you might need to change both nfs-rootonly and mount-rootonly to disabled. If a server is managing an extremely large number of TCP connections, it is possible that no ports below 1024 is available, and the OS is forced to use higher ports. These two ONTAP parameters would need to be changed to allow the connection to complete.

NFS export polices: superuser and setuid

If Oracle binaries are located on an NFS share, the export policy must include superuser and setuid permissions.

Shared NFS exports used for generic file services such as user home directories usually squash the root user. This means a request from the root user on a host that has mounted a filesystem is remapped as a different user with lower privileges. This helps secure data by preventing a root user on a particular server from accessing data on the shared server. The setuid bit can also be a security risk on a shared environment. The setuid bit allows a process to be run as a different user than the user invoking the command. For example, a shell script that was owned by root with the setuid bit runs as root. If that shell script could be changed by other users, any non-root user could issue a command as root by updating the script.

The Oracle binaries include files owned by root and use the setuid bit. If Oracle binaries are installed on an NFS share, the export policy must include the appropriate superuser and setuid permissions. In the example below, the rule includes both `allow-suid` and permits `superuser` (root) access for NFS clients using system authentication.

```
Cluster01::> export-policy rule show -vserver vserver1 -policyname orabin
-fields allow-suid,superuser
vserver   policyname ruleindex superuser allow-suid
--------- ---------- --------- --------- ----------
vserver1  orabin     1         sys       true
```

NFSv4/4.1 configuration

For most applications, there is very little difference between NFSv3 and NFSv4. Application I/O is usually very simple I/O and does not benefit significantly from some of the advanced features available in NFSv4. Higher versions of NFS should not be viewed as an “upgrade” from a database storage perspective, but instead as versions of NFS that include additional features. For example, if the end-to-end security of kerberos privacy mode (krb5p) is required, then NFSv4 is required.

NetApp recommends using NFSv4.1 if NFSv4 capabilities are required. There are some functional enhancements to the NFSv4 protocol in NFSv4.1 that improve resiliency in certain edge cases.

Switching to NFSv4 is more complicated than simply changing the mount options from vers=3 to vers=4.1. A more complete explanation of NFSv4 configuration with ONTAP, including guidance on configuring the OS, see TR-4067 NFS on ONTAP best practices. The following sections of this TR explain some of the basic requirements for using NFSv4.

NFSv4 domain

A complete explanation of NFSv4/4.1 configuration is beyond the scope of this document, but one commonly encountered problem is a mismatch in domain mapping. From a sysadmin point of view, the NFS file systems appear to behave normally, but applications report errors about permissions and/or setuid on the certain files. In some cases, administrators have incorrectly concluded that the permissions of the application binaries have been damaged and have run chown or chmod commands when the actual problem was the domain name.

The NFSv4 domain name is set on the ONTAP SVM:

```
Cluster01::> nfs server show -fields v4-id-domain
vserver   v4-id-domain
--------- ------------
vserver1  my.lab
```

The NFSv4 domain name on the host is set in `/etc/idmap.cfg`

```
[root@host1 etc]# head /etc/idmapd.conf
[General]
#Verbosity = 0
# The following should be set to the local NFSv4 domain name
# The default is the host's DNS domain name.
Domain = my.lab
```

The domain names must match. If they do not, mapping errors similar to the following appear in `/var/log/messages`:

```
Apr 12 11:43:08 host1 nfsidmap[16298]: nss_getpwnam: name 'root@my.lab'
does not map into domain 'default.com'
```

Application binaries, such as Oracle database binaries, include files owned by root with the setuid bit, which means a mismatch in the NFSv4 domain names causes failures with Oracle startup and a warning about the ownership or permissions of a file called `oradism`, which is located in the `$ORACLE_HOME/bin` directory. It should appear as follows:

```
[root@host1 etc]# ls -l /orabin/product/19.3.0.0/dbhome_1/bin/oradism
-rwsr-x--- 1 root oinstall 147848 Apr 17  2019
/orabin/product/19.3.0.0/dbhome_1/bin/oradism
```

If this file appears with ownership of nobody, there may be an NFSv4 domain mapping problem.

```
[root@host1 bin]# ls -l oradism
-rwsr-x--- 1 nobody oinstall 147848 Apr 17  2019 oradism
```

To fix this, check the `/etc/idmap.cfg` file against the v4-id-domain setting on ONTAP and ensure they are consistent. If they are not, make the required changes, run `nfsidmap -c`, and wait a moment for the changes to propagate. The file ownership should then be properly recognized as root. If a user had attempted to run `chown root` on this file before the NFS domains configure was corrected, it might be necessary to run `chown root` again.

Oracle directNFS

Oracle databases can use NFS in two ways.

First, it can use a filesystem mounted using the native NFS client that is part of the operating system. This is sometimes called kernel NFS, or kNFS. The NFS filesystem is mounted and used by the Oracle database exactly the same as any other application would use an NFS filesystem.

The second method is Oracle Direct NFS (dNFS). This is an implementation of the NFS standard within the Oracle database software. It does not change the way Oracle databases are configured or managed by the DBA. As long as the storage system itself has the correct settings, the use of dNFS should be transparent to the DBA team and end users.

A database with the dNFS feature enabled still has the usual NFS filesystems mounted. Once the database is open, the Oracle database opens a set of TCP/IP sessions and performs NFS operations directly.

Direct NFS

The primary value of Oracle's Direct NFS is to bypass the host NFS client and perform NFS file operations directly on an NFS server. Enabling it only requires changing the Oracle Disk Manager (ODM) library. Instructions for this process are provided in the Oracle documentation.

Using dNFS results in a significant improvement in I/O performance and decreases the load on the host and the storage system because I/O is performed in the most efficient way possible.

In addition, Oracle dNFS includes an **option** for network interface multipathing and fault-tolerance. For example, two 10Gb interfaces can be bound together to offer 20Gb of bandwidth. A failure of one interface results in I/O being retried on the other interface. The overall operation is very similar to FC multipathing. Multipathing was common years ago when 1Gb ethernet was the most common standard. A 10Gb NIC is sufficient for most Oracle workloads, but if more is required 10Gb NICs can be bonded.

When dNFS is used, it is critical that all patches described in Oracle Doc 1495104.1 are installed. If a patch cannot be installed, the environment must be evaluated to make sure that the bugs described in that document do not cause problems. In some cases, an inability to install the required patches prevents the use of dNFS.

Do not use dNFS with any type of round-robin name resolution, including DNS, DDNS, NIS or any other method. This includes the DNS load balancing feature available in ONTAP. When an Oracle database using dNFS resolves a host name to an IP address it must not change on subsequent lookups. This can result in Oracle database crashes and possible data corruption.

Direct NFS and host file system access

Using dNFS can occasionally cause problems for applications or user activities that rely on the visible file systems mounted on the host because the dNFS client accesses the file system out of band from the host OS. The dNFS client can create, delete, and modify files without the knowledge of the OS.

When the mount options for single-instance databases are used, they enable caching of file and directory attributes, which also means that the contents of a directory are cached. Therefore, dNFS can create a file, and

there is a short lag before the OS rereads the directory contents and the file becomes visible to the user. This is not generally a problem, but, on rare occasions, utilities such as SAP BR*Tools might have issues. If this happens, address the problem by changing the mount options to use the recommendations for Oracle RAC. This change results in the disabling of all host caching.

Only change mount options when (a) dNFS is used and (b) a problem results from a lag in file visibility. If dNFS is not in use, using Oracle RAC mount options on a single-instance database results in degraded performance.

See the note about `nosharecache` in Linux NFS mount options for a Linux-specific dNFS issue that can produce unusual results.

Oracle databases and NFS leases and locks

NFSv3 is stateless. That effectively means that the NFS server (ONTAP) doesn't keep track of which file systems are mounted, by whom, or which locks are truly in place.

ONTAP does have some features that will record mount attempts so you have an idea which clients may be accessing data, and there may be advisory locks present, but that information isn't guaranteed to be 100% complete. It can't be complete, because tracking NFS client state is not part of the NFSv3 standard.

NFSv4 statefulness

In contrast, NFSv4 is stateful. The NFSv4 server tracks which clients are using which file systems, which files exist, which files and/or regions of files are locked, etc. This means there needs to be regular communication between an NFSv4 server to keep the state data current.

The most important states being managed by the NFS server are NFSv4 Locks and NFSv4 Leases, and they are very much intertwined. You need to understand how each works by itself, and how they relate to one another.

NFSv4 locks

With NFSv3, locks are advisory. An NFS client can still modify or delete a "locked" file. An NFSv3 lock doesn't expire by itself, it must be removed. This creates problems. For example, if you have a clustered application that creates NFSv3 locks, and one of the nodes fails, what do you do? You can code the application on the surviving nodes to remove the locks, but how do you know that's safe? Maybe the "failed" node is operational, but isn't communicating with the rest of the cluster?

With NFSv4, locks have a limited duration. As long as the client holding the locks continues to check in with the NFSv4 server, no other client is permitted to acquire those locks. If a client fails to check in with the NFSv4, the locks eventually get revoked by the server and other clients will be able to request and obtain locks.

NFsv4 leases

NFSv4 locks are associated with an NFSv4 lease. When an NFSv4 client establishes a connection with an NFSv4 server, it gets a lease. If the client obtains a lock (there are many types of locks) then the lock is associated with the lease.

This lease has a defined timeout. By default, ONTAP will set the timeout value to 30 seconds:

```
Cluster01::*> nfs server show -vserver vserver1 -fields v4-lease-seconds

vserver   v4-lease-seconds
--------- ----------------
vserver1  30
```

This means that an NFSv4 client needs to check in with the NFSv4 server every 30 seconds to renew its leases.

The lease is automatically renewed by any activity, so if the client is doing work there's no need to perform addition operations. If an application becomes quiet and is not doing real work, it's going to need to perform a sort of keep-alive operation (called a SEQUENCE) instead. It's essentially just saying "I'm still here, please refresh my leases."

```
*Question:* What happens if you lose network connectivity for 31 seconds?
```

NFSv3 is stateless. It's not expecting communication from the clients. NFSv4 is stateful, and once that lease period elapses, the lease expires, and locks are revoked and the locked files are made available to other clients.

With NFSv3, you could move network cables around, reboot network switches, make configuration changes, and be fairly sure that nothing bad would happen. Applications would normally just wait patiently for the network connection to work again.

With NFSv4, you have 30 seconds (unless you've increased the value of that parameter within ONTAP) to complete your work. If you exceed that, your leases time out. Normally this results in application crashes.

As an example, if you have an Oracle database, and you experience a loss of network connectivity (sometimes called a "network partition") that exceeds the lease timeout, you will crash the database.

Here's an example of what happens in the Oracle alert log if this happens:

```
2022-10-11T15:52:55.206231-04:00
Errors in file /orabin/diag/rdbms/ntap/NTAP/trace/NTAP_ckpt_25444.trc:
ORA-00202: control file: '/redo0/NTAP/ctrl/control01.ctl'
ORA-27072: File I/O error
Linux-x86_64 Error: 5: Input/output error
Additional information: 4
Additional information: 1
Additional information: 4294967295
2022-10-11T15:52:59.842508-04:00
Errors in file /orabin/diag/rdbms/ntap/NTAP/trace/NTAP_ckpt_25444.trc:
ORA-00206: error in writing (block 3, # blocks 1) of control file
ORA-00202: control file: '/redo1/NTAP/ctrl/control02.ctl'
ORA-27061: waiting for async I/Os failed
```

If you look at the syslogs, you should see several of these errors:

```
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
```

The log messages are usually the first sign of a problem, other than the application freeze. Typically, you see nothing at all during the network outage because processes and the OS itself are blocked attempting to access the NFS file system.

The errors appear after the network is operational again. In the example above, once connectivity was reestablished, the OS attempted to reacquire the locks, but it was too late. The lease had expired and the locks were removed. That results in an error that propagates up to the Oracle layer and causes the message in the alert log. You might see variations on these patterns depending on the version and configuration of the database.

In summary, NFSv3 tolerates network interruption, but NFSv4 is more sensitive and imposes a defined lease period.

What if a 30 second timeout isn't acceptable? What if you manage a dynamically changing network where switches are rebooted or cables are relocated and the result is the occasional network interruption? You could choose to extend the lease period, but whether you want to do that requires an explanation of NFSv4 grace periods.

NFSv4 grace periods

If an NFSv3 server is rebooted, it's ready to serve IO almost instantly. It was not maintaining any sort of state about clients. The result is that an ONTAP takeover operation often appears to be close to instantaneous. The moment a controller is ready to start serving data it will send an ARP to the network that signals the change in topology. Clients normally detect this almost instantly and data resumes flowing.

NFSv4, however, will produce a brief pause. It's just part of how NFSv4 works.

NFSv4 servers need to track the leases, locks, and who's using what data. If an NFS server panics and reboots, or loses power for a moment, or is restarted during maintenance activity, the result is the lease/lock and other client information is lost. The server needs to figure out which client is using what data before resuming operations. This is where the grace period comes in.

If you suddenly power cycle your NFSv4 server. When it comes back up, clients that attempt to resume IO will get a response that essentially says, "I have lost lease/lock information. Would you like to re-register your locks?" That's the start of the grace period. It defaults to 45 seconds on ONTAP:

```
Cluster01::> nfs server show -vserver vserver1 -fields v4-grace-seconds

vserver   v4-grace-seconds
--------- ----------------
vserver1  45
```

The result is that, after a restart, a controller will pause IO while all the clients reclaim their leases and locks. Once the grace period ends, the server will resume IO operations.

Lease timeouts vs grace periods

The grace period and the lease period are connected. As mentioned above, the default lease timeout is 30 seconds, which means NFSv4 clients must check in with the server at least every 30 seconds or they lose their leases and, in turn, their locks. The grace period exists to allow an NFS server to rebuild lease/lock data, and it defaults to 45 seconds. ONTAP requires the grace period to be 15 seconds longer than the lease period. This ensures that an NFS client environment that is designed to renew leases at least every 30 seconds will have the ability to check in with the server after a restart. A grace period of 45 seconds ensures that all those clients that expect to renew their leases at least every 30 seconds definitely have the opportunity to do so.

If a 30 second timeout isn't acceptable, you could choose to extend the lease period. If you want to increase the lease timeout to 60 seconds in order to withstand a 60 second network outage, you're going to have to increase the grace period to at least 75 seconds. ONTAP requires it to be 15 seconds higher than the lease period. That means you're going to experience longer IO pauses during controller failover.

This shouldn't normally be a problem. Typical users only update ONTAP controllers once or twice per year, and unplanned failover due to hardware failures are extremely rare. In addition, if you had a network where a 60-second network outage was a concerning possibility, and you needed to the lease timeout to 60 seconds, then you probably wouldn't object to rare storage system failover resulting in a 75 second pause either. You've already acknowledged you have a network that's pausing for 60+ seconds rather frequently.

NFS caching with Oracle databases

The presence of any of the following mount options causes host caching to be disabled:

```
cio, actimeo=0, noac, forcedirectio
```

These settings can have a severe negative effect on the speed of software installation, patching, and backup/restore operations. In some cases, especially with clustered applications, these options are required as an inevitable result of the need to deliver cache-coherency across all nodes in the cluster. In other cases, customers mistakenly use these parameters and the result is unnecessary performance damage.

Many customers temporarily remove these mount options during installation or patching of the application binaries. This removal can be performed safely if the user verifies that no other processes are actively using the target directory during the installation or patching process.

NFS transfer sizes with Oracle databases

By default, ONTAP limits NFS I/O sizes to 64K.

Random I/O with an most applications and databases uses a much smaller block size which is well below the 64K maximum. Large-block I/O is usually parallelized, so the 64K maximum is also not a limitation to obtaining maximum bandwidth.

There are some workloads where the 64K maximum does create a limitation. In particular, single-threaded operations such as backup or recovery operation or a database full table scan run faster and more efficiently if the database can perform fewer but larger I/Os. The optimum I/O handling size for ONTAP is 256K.

The maximum transfer size for a given ONTAP SVM can be changed as follows:

```
Cluster01::> set advanced
Warning: These advanced commands are potentially dangerous; use them only
when directed to do so by NetApp personnel.
Do you want to continue? {y|n}: y
Cluster01::*> nfs server modify -vserver vserver1 -tcp-max-xfer-size
262144
Cluster01::*>
```

Caution
Never decrease the maximum allowable transfer size on ONTAP below the value of rsize/wsize of currently mounted NFS file systems. This can create hangs or even data corruption with some operating systems. For example, if NFS clients are currently set at an rsize/wsize of 65536, then the ONTAP maximum transfer size could be adjusted between 65536 and 1048576 with no effect because the clients themselves are limited. Reducing the maximum transfer size below 65536 can damage availability or data.

Copyright information

Trademark information