NFS LOCK

Santhanakannan. Ramasamy

Creation Date: 22/12/2011

Version 1.0

1. Storage and Unix end

Network File System (NFS) Lock Recovery and Network Status Monitor

NFS lock recovery and Network Status Monitor

NFS versions 2 and 3 depend on the Network Lock Manager (NLM) protocol for file locking. Another RPC protocol called the Network Status Monitor (NSM), is used to notify clients of a loss of lock state because of a server reboot. When a NFS server grants a lock to a client it must maintain a record of the client that owns the lock. This information is maintained on disk. The individual lock state itself is non-persistent. If the server reboots the lock will be lost. The client needs to be notified so that it can reestablish the lock when the NFS server is again available. The filer NSM maintains its information as files in `/etc/sm`:

`state` state of the NSM
`monitor` list of hosts currently being monitored
`notify` list of hosts being notified after a reboot

Upon rebooting or cluster takeover, the filer reads the `/etc/sm/monitor` file to determine which clients held NLM file locks prior to reboot or cluster takeover. The clients to be notified are then copied into the `/etc/sm/notify` file and hence will be used for notifying clients. The filer notifies the clients via NSM that it has rebooted and lost all locks. Clients running a NSM daemon (`rpc.statd/statd`) will issue lock reclaim requests to rebuild the lock state that was lost during the filer reboot. When the filer reboots, there is an NLM grace period of 45 seconds during which the filer will not honor any new lock requests; it will only honor reclaim requests. The grace period gives all NFS clients that were holding locks the opportunity to reclaim their locks.

Client or network problems may prevent the filer Network Status Monitor from notifying all the monitored clients. Each client that cannot be contacted will delay the startup of NFS file services after the reboot. The filer will attempt to contact all the clients in the notify list before NFS services are completely started. The maximum timeout value for each client is 10 seconds. Issues that could prevent the filer from notifying clients:

1. The client is down or no longer available on the network.
2. The client is not running a NSM daemon; rpc.statd on linux, statd on Solaris,
3. There is a network connectivity or network equipment outage.
4. The filer cannot resolve client hostnames because it cannot contact the DNS or NIS services.

Error Messages

```
Error message: [sm_recover]: no address for host [nfs_client1]
Error message: [sm_recover]: get RPC port for
[host=unix1,prog=100024,ver=1,prot=17] failed
```

Checking for unavailable hosts in the /etc/sm/monitor file

1. Check the list of clients for the following:

   ```
   - ping the host from the filer
   - the client portmapper is functioning(rpcinfo -p hostname)
   - the client rpc.statd/statd is running (rpcinfo -p hostname)
   ```

   If a client fails the checks above, the problem should be corrected. If the client is permanently not available, then it can be removed from the monitor file using the `sm_mon` command.

2. The filer advanced mode command `sm_mon` can be used to remove a host from the monitor list:

   ```
   Enter  priv set advanced
   Enter  sm_mon -u  [client_name]
   Enter  priv set admin
   ```

Vfiler

The configuration of vfilers can impact the Network Status Monitor client notification process. Each vfiler maintains its own set of information files under the vfiler `/etc/sm` directory. After a reboot of the filer, NFS services are restarted on each vfiler. Each vfiler must notify the NFS clients in its `/etc/sm/notify` file so that the locks can be reclaimed. A non-responding client that is shared among vfilers would incur a 10 second timeout for each vfiler. A client that held locks on multiple vfilers, must be notified via NSM by each vfiler. This could impact the overall startup time before all NFS file services are completely operating.

Filer Cluster

In a filer cluster configuration, a partner takeover or giveback operation is the same as a reboot of the cluster partner. The nfs clients that held locks on the affected filer must be notified via the Network Status Monitor (NSM) as detailed in the sections above.

Because of the upper limit of 5 minutes on the startup of each service on a cluster failover/takeover, the delay in contacting the NLM clients could further compound the startup of the failed filer and could impact the overall availability of the cluster

2. Network File system (NFS) locks handled during a cluster failover

The NFS locks work the same way as that of any other cluster. But in a NetApp cluster it is only the data that is taken over and served to clients , unlike the application takeover that happens in an operating system cluster.

Here, the failover or takeover is the same as that of a reboot of a partner node.

It works exactly the same if one of the filer head is rebooted. When the head that takes over comes back up(after the reboot that it goes through to do takeover), the filer will try to reclaim its locks via `statd`. When a lock is issued, the filer puts the client name in `/etc/sm`. Then when the failed head comes back up, statd contacts each client and asks the clients to tell the filer what locks they have. The filer will then re-establish those locks in memory based on the response of the clients.

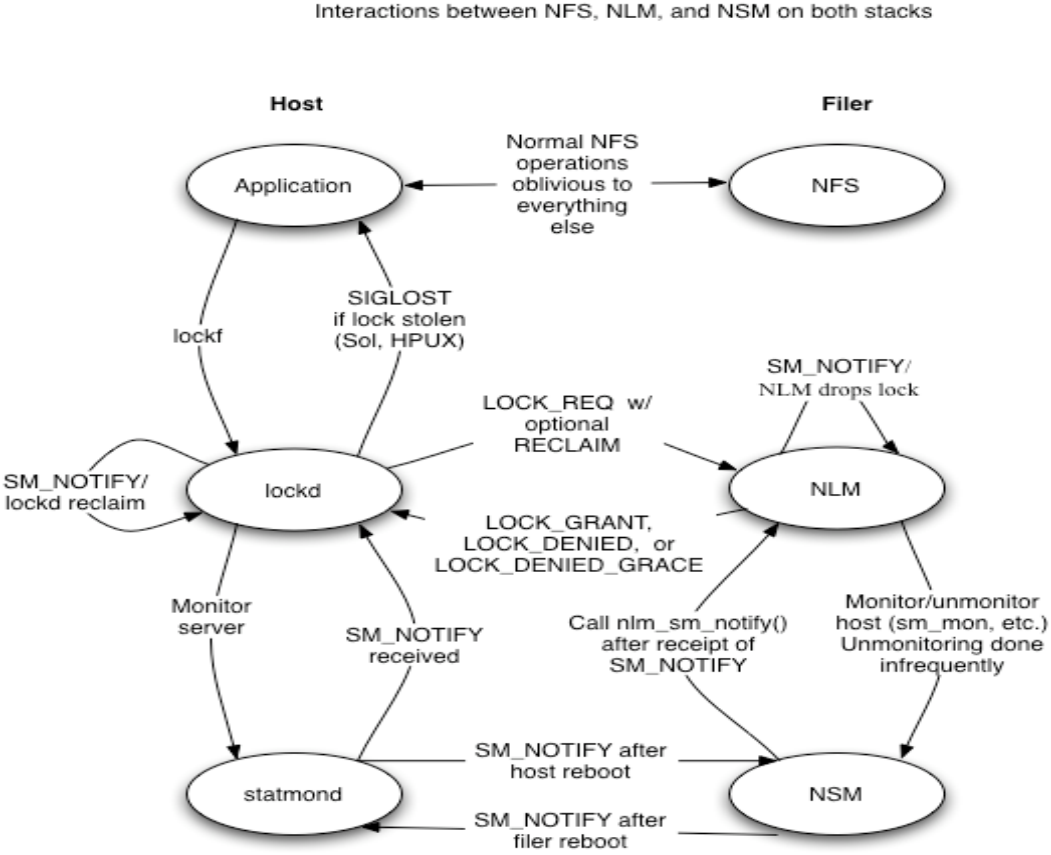
3. Hand shack process:

NSM: Network status Monitor

File locking over NFS requires a facility to notify peers in case of a reboot, so that clients can reclaim locks after a server crash, and/or servers can release locks held by the rebooted client.

This is a two-step process: during normal operations, a mechanism is required to keep track of which hosts need to be informed of a reboot. And of course, notifications need to be sent out during reboot. The protocol used for this is called NSM, for Network Status Monitor.
Commonly, these two features are provided by the rpc.statd daemon.

Hand shack process:



There are two types of locks involved:

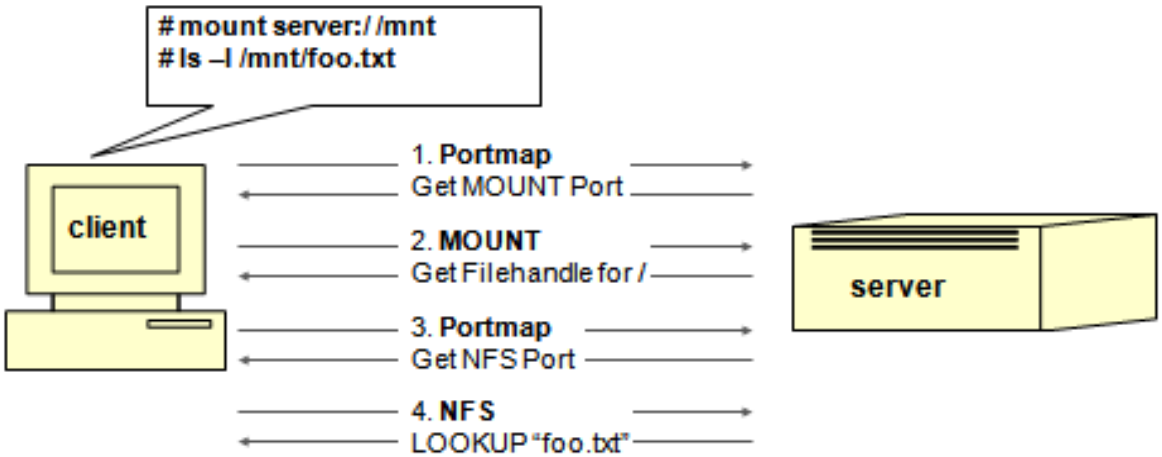
a. **OS level locks**

Oracle places OS level locks when a database is started and releases these OS level locks when a database is stopped. Depending on the exact version of the database software, it also places small files. For Oracle 7 and 8, this is the `$ORACLE_HOME/dbs/sgadef.dbf` file in .For later versions this is the`$ORACLE_HOME/dbs/lk` file.

b. **NFS locking daemons**

In order to be able to placed and release these OS locks on an NFS mounted file system you need the NFS locking daemons. These locking daemons are the rpc.lockd and the rpc.statd on most UNIXes. With these locking schemes we may encounter some problems where the Oracle or the OS errors are not much self explanatory.

Client operation:



4. Common Oracle DB Lock issues and solutions:

Problem 1:

Symptoms:

- Database startup fails
- Errors in `alert.log`:

ALTER DATABASE MOUNT
scumnt: failed to lock .../dbs/lk exclusive
<os error="">
Compaq Tru64 UNIX Error: 77: No locks available
HP-UX Error: 13: Permission denied
ORA-1102 signalled during: ALTER DATABASE MOUNT.

Cause:

- One or both of the locking daemons crashed for any reason
- Instance was down when the locking daemon(s) crashed
- When starting the instance locks cannot be set

Solution: Restart locking daemons and retry startup.

Problem 2:

Symptoms:

- (8i and before) Database startup fails with:
ORA-00600: internal error code, arguments:[2806], [60]

- OR -

- (After 9i) Database startup hangs

Cause:

- One or both of the locking daemons crashed for any reason
- Instance was running when the locking daemon(s) crashed
- Locks could not be released on files when the database was being shutdown since at least one of the daemons were not running.
- When trying to startup the database you had encountered the `Problem 1` and fixed that but restarting the locking daemons.
- In this case we have locking daemons running but there are locks on the files remaining although the database is not up.
- Locks remained are generally on the:

```
. sgadef<sid>.dbf
. lck<sid>
. controlfiles
.  redo logs
. datafiles
```

Solution:

- Verify that no oracle database processes are running.
- Umount all NFS file systems from the host in question -
- Kill statd and lockd processes on the host (take down NFS) -
- Remove files found in the `/etc/sm` directory on the filer. It is ok to remove, as this is secondary cache, and will reestablish from the memory (`sm_mon`)
- Remove locks on the filer (priv set advanced; `sm_mon -l`; priv set)
- Restart the NFS services
- Mount the file systems
- Copy lk file in `/$ORACLE_HOME/dbs` directory to a temporary location (in case a restore is needed).
- Remove lk file from `$ORACLE_HOME/dbs` directory .
- Start the database

How to clear NFS locks during network crash or outage for Oracle datafiles

Procedure / Workaround

Most of the time simply breaking the locks on the filer will allow the database to start back up. This should be tried before running through the detailed procedure,

If the filer is running Data ONTAP 7.3 or higher

lock status –h <hostname> → fully qualified or shortname(status check)
lock status -p protocol [-n] → protocol can be nlm,nfsv4,cifs and flexcache (case-insensitive)

Lock break commands:

1. Execute 'lock break -h [server]' to release any locks that still exist.
   Note: If the 'lock break -h [server]' does not work, ensure that the server name you are entering is not the same as the one that the filer has.
2. If the locks are not cleared, run `lock break -p nlm`. This will clear all the NFS locks on the filer. This is safe to run even if there are other databases being served from the filer

`lock break -h host` → To break the lock for specific hosts

`lock break -p protocol` → To break the lock for specific protocol (nlm,nfsv4,cifs and flexcache)

`lock break -net network` → To break the lock for specific network (subnet)

Summary of Corrective steps:

- Shutdown Oracle databases
- Unmount database volumes
- Kill lockd/statd processes on UNIX host
- Clear locks on filer
- Restart lockd/statd processes on UNIX host
- Remount the database volumes on the UNIX host
- Restart databases

Detailed Procedure:

1. Shutdown all Oracle databases being run by the affected server.
    1. Issue the Oracle shutdown immediate command and verify that no database processes are still running by issuing the UNIX command ps -ef |grep -i ora on the UNIX database host.

    2. If database processes are still running issue the Oracle shutdown abort command and use the UNIX command ps -ef |grep -i ora to verify that no database processes are still running.

    3. If database processes are still running do the following from the UNIX command line:
        - `ps -ef |grep ora` to get process id's (pid's) of remaining Oracle processes

        - `kill -9 pid` for each remaining Oracle process.

2. Unmount all database volumes using the UNIX umount command.

3. Kill statd and lockd processes on the UNIX host in the order specified below:
    1. Determine the process id's (pid's) of statd and lockd from the UNIX command line:

        ```
        a.ps -ef |grep lockd

        b. ps -ef |grep statd

        c.kill [lockd_process_id]

        d.kill [statd_process_id]
        ```

4. Remove locks from filer **(please refer above Lock brake for more details)**

    1. If the filer is running Data ONTAP 7.1 or higher run 'lock break -h [server]' to release any locks that still exist.
    Note: If the 'lock break -h [server]' doesn't work, ensure that the server name that you are entering is not the same as the one that the filer has.

    2. If the locks are not cleared, run 'lock break -p nlm' (This also requires Data ONTAP 7.1 or higher). This will clear all the NFS locks on the filer. This will not sever any NFS

connections, it will simply force the processes to re-request the locks for the files they are writing to.

5. Remove the NFS lock files on the host.

   rpc.statd uses gethostbyname() to determine the client's name, but lockd (in the Linux kernel) uses uname -n.

   By changing the HOSTNAME= fully qualified domain name, lockd will use an FQDN when contacting the storage. If there is a lnx_node1.iop.eng.netapp.com and also a lnx_node5.ppe.iop.eng.netapp.com contacting the same NetApp storage, the storage will be able to correctly distinguish the locks owned by each client. Therefore, we recommend using the fully qualified name in /etc/sysconfig/network. In addition to this, sm_mon -l or lock break on the storage will also clear the locks on the storage which will fix the lock recovery problem.

   Additionally, if the client's nodename is fully qualified (that is, it contains the hostname and the domain name spelled out), then rpc.statd must also use a fully qualified name. Likewise, if the nodename is unqualified, then rpc.statd must use an unqualified name. If the two values do not match, lock recovery will not work. Be sure the result of gethostbyname(3) matches the output of uname -n by adjusting your client's nodename in /etc/hosts, DNS, or your NIS databases.

6. Start the UNIX statd and lockd processes from the UNIX host command line in the order specified below:
   1. /usr/lib/nfs/statd
   2. /usr/lib/nfs/lockd

7. Mount the database volumes on the UNIX host.

8. Start the database(s) and test for availability.

Client timeout options/limt :

VM ESX host	180 sec
Oracle RAC nodes Voting disk timeout	200 sec
UNIX machines (NFS Timeo)	600 sec

Side note:

- *Network Lock Manager (NLM)*
  - *Enables file locking*
  - *Synchronous and asynchronous locking*
- *Network Status Monitor (NSM)*
  - *Facilitates lock recovery*
  - *The target of notifications are clients that requested NLM locks*
  - *Client list is maintained in persistent storage*
  - *Client and server maintain a state number*

- *NFSv2/v3*
  - *Collection of protocols (Portmap, MOUNT, NLM, NSM)*
  - *Works over UDP and TCP*