NAME

sm-notify – send reboot notifications to NFS peers

SYNOPSIS

/usr/sbin/sm-notify [-dfn] [-m *minutes***] [-v** *name***] [-p** *notify-port***] [-P** *path***]**

DESCRIPTION

File locks are not part of persistent file system state. Lock state is thus lost when a host reboots.

Network file systems must also detect when lock state is lost because a remote host has rebooted. After an NFS client reboots, an NFS server must release all file locks held by applications that were running on that client. After a server reboots, a client must remind the server of file locks held by applications running on that client.

For NFS version 2 and version 3, the *Network Status Monitor* protocol (or NSM for short) is used to notify NFS peers of reboots. On Linux, two separate user-space components constitute the NSM service:

sm-notify
: A helper program that notifies NFS peers after the local system reboots

rpc.statd
: A daemon that listens for reboot notifications from other hosts, and manages the list of hosts to be notified when the local system reboots

The local NFS lock manager alerts its local **rpc.statd** of each remote peer that should be monitored. When the local system reboots, the **sm-notify** command notifies the NSM service on monitored peers of the reboot. When a remote reboots, that peer notifies the local **rpc.statd**, which in turn passes the reboot notification back to the local NFS lock manager.

NSM OPERATION IN DETAIL

The first file locking interaction between an NFS client and server causes the NFS lock managers on both peers to contact their local NSM service to store information about the opposite peer. On Linux, the local lock manager contacts **rpc.statd**.

rpc.statd records information about each monitored NFS peer on persistent storage. This information describes how to contact a remote peer in case the local system reboots, how to recognize which monitored peer is reporting a reboot, and how to notify the local lock manager when a monitored peer indicates it has rebooted.

An NFS client sends a hostname, known as the client's *caller_name*, in each file lock request. An NFS server can use this hostname to send asynchronous GRANT calls to a client, or to notify the client it has rebooted.

The Linux NFS server can provide the client's *caller_name* or the client's network address to **rpc.statd**. For the purposes of the NSM protocol, this name or address is known as the monitored peer's *mon_name*. In addition, the local lock manager tells **rpc.statd** what it thinks its own hostname is. For the purposes of the NSM protocol, this hostname is known as *my_name*.

There is no equivalent interaction between an NFS server and a client to inform the client of the server's *caller_name*. Therefore NFS clients do not actually know what *mon_name* an NFS server might use in an SM_NOTIFY request. The Linux NFS client records the server's hostname used on the mount command to identify rebooting NFS servers.

Reboot notification

When the local system reboots, the **sm-notify** command reads the list of monitored peers from persistent storage and sends an SM_NOTIFY request to the NSM service on each listed remote peer. It uses the *mon_name* string as the destination. To identify which host has rebooted, the **sm-notify** command normally sends *my_name* string recorded when that remote was monitored. The remote **rpc.statd** matches incoming SM_NOTIFY requests using this string, or the caller's network address, to one or more peers on its own monitor list.

If **rpc.statd** does not find a peer on its monitor list that matches an incoming SM_NOTIFY request, the notification is not forwarded to the local lock manager. In addition, each peer has its own *NSM state number*,

a 32-bit integer that is bumped after each reboot by the **sm-notify** command. **rpc.statd** uses this number to distinguish between actual reboots and replayed notifications.

Part of NFS lock recovery is rediscovering which peers need to be monitored again. The **sm-notify** command clears the monitor list on persistent storage after each reboot.

OPTIONS

-d Keeps **sm-notify** attached to its controlling terminal and running in the foreground so that notification progress may be monitored directly.

-f Send notifications even if **sm-notify** has already run since the last system reboot.

-m *retry-time*

Specifies the length of time, in minutes, to continue retrying notifications to unresponsive hosts. If this option is not specified, **sm-notify** attempts to send notifications for 15 minutes. Specifying a value of 0 causes **sm-notify** to continue sending notifications to unresponsive peers until it is manually killed.

Notifications are retried if sending fails, the remote does not respond, the remote's NSM service is not registered, or if there is a DNS failure which prevents the remote's *mon_name* from being resolved to an address.

Hosts are not removed from the notification list until a valid reply has been received. However, the SM_NOTIFY procedure has a void result. There is no way for **sm-notify** to tell if the remote recognized the sender and has started appropriate lock recovery.

-n Prevents **sm-notify** from updating the local system's NSM state number.

-p *port* Specifies the source port number **sm-notify** should use when sending reboot notifications. If this option is not specified, a randomly chosen ephemeral port is used.

This option can be used to traverse a firewall between client and server.

–P, ––state–directory–path *pathname*

Specifies the pathname of the parent directory where NSM state information resides. If this option is not specified, **sm-notify** uses */var/lib/nfs* by default.

After starting, **sm-notify** attempts to set its effective UID and GID to the owner and group of the subdirectory **sm** of this directory. After changing the effective ids, **sm-notify** only needs to access files in **sm** and **sm.bak** within the state-directory-path.

-v *ipaddr* | *hostname*

Specifies the network address from which to send reboot notifications, and the *mon_name* argument to use when sending SM_NOTIFY requests. If this option is not specified, **sm-notify** uses a wildcard address as the transport bind address, and uses the *my_name* recorded when the remote was monitored as the *mon_name* argument when sending SM_NOTIFY requests.

The *ipaddr* form can be expressed as either an IPv4 or an IPv6 presentation address. If the *ipaddr* form is used, the **sm-notify** command converts this address to a hostname for use as the *mon_name* argument when sending SM_NOTIFY requests.

This option can be useful in multi-homed configurations where the remote requires notification from a specific network address.

SECURITY

The **sm-notify** command must be started as root to acquire privileges needed to access the state information database. It drops root privileges as soon as it starts up to reduce the risk of a privilege escalation attack.

During normal operation, the effective user ID it chooses is the owner of the state directory. This allows it to continue to access files in that directory after it has dropped its root privileges. To control which user ID **rpc.statd** chooses, simply use chown(1) to set the owner of the state directory.

ADDITIONAL NOTES

Lock recovery after a reboot is critical to maintaining data integrity and preventing unnecessary application hangs.

To help **rpc.statd** match SM_NOTIFY requests to NLM requests, a number of best practices should be observed, including:

The UTS nodename of your systems should match the DNS names that NFS peers use to contact them

The UTS nodenames of your systems should always be fully qualified domain names

The forward and reverse DNS mapping of the UTS nodenames should be consistent

The hostname the client uses to mount the server should match the server's *mon_name* in SM_NOTIFY requests it sends

Unmounting an NFS file system does not necessarily stop either the NFS client or server from monitoring each other. Both may continue monitoring each other for a time in case subsequent NFS traffic between the two results in fresh mounts and additional file locking.

On Linux, if the **lockd** kernel module is unloaded during normal operation, all remote NFS peers are unmonitored. This can happen on an NFS client, for example, if an automounter removes all NFS mount points due to inactivity.

IPv6 and TI-RPC support

TI-RPC is a pre-requisite for supporting NFS on IPv6. If TI-RPC support is built into the **sm-notify** command ,it will choose an appropriate IPv4 or IPv6 transport based on the network address returned by DNS for each remote peer. It should be fully compatible with remote systems that do not support TI-RPC or IPv6.

Currently, the **sm-notify** command supports sending notification only via datagram transport protocols.

FILES

/var/lib/nfs/sm	directory containing monitor list
/var/lib/nfs/sm.bak	directory containing notify list
/var/lib/nfs/state	NSM state number for this host
/proc/sys/fs/nfs/nsm_local_state	kernel's copy of the NSM state number

SEE ALSO

rpc.statd(8), nfs(5), uname(2), hostname(7)

RFC 1094 - "NFS: Network File System Protocol Specification"
RFC 1813 - "NFS Version 3 Protocol Specification"
OpenGroup Protocols for Interworking: XNFS, Version 3W - Chapter 11

AUTHORS

Olaf Kirch <okir@suse.de>
Chuck Lever <chuck.lever@oracle.com>