Linux NFS/IPv6 Status Report

Chuck Lever, Oracle

Acknowledgements

- Group Bull
- Jeff Layton
- Steve Dickson

Agenda

- Design and implementation
- Today’s feature set
- Missing features and next steps

Design & Implementation

Challenges: Requirements

- Marketing check box
  - No IPv6 applications or use cases provided
  - Make it so

Challenges: Integration

- Code base must continue to work on legacy installations
  - IPv4 features must change as little as possible; must never break
  - No existing regression test suites
  - Legacy code and requirements not documented
    - man pages only go so far

IPv6 Prerequisites

- Text-based mounts
- TI-RPC in user space
- TI-RPC-like support in kernel sunrpc

Text-based Mounts

- Difficult to add new features with legacy mount(2) API
  - Inflexible data structure
  - User space and kernel must be in lockstep
- Solution
  - Manage more of mount processing in kernel
  - Pass string of options, like most other Linux file systems

Text-based Mounts (kernel)

- Embrace and extend existing components in kernel
    - NFSROOT support
    - MNT client
    - String option parser, presentation address parser
- Must distinguish string from legacy nfs_mount_data blob
- Decide on return codes from mount(2)

Text-based Mounts (user)

- Mount.nfs decides at run-time to use text-based
- NFSv4 simply passes options string to the kernel
- NFSv2/v3 challenges:
  - How to convert string options to pmap parameters
  - When to retry and when to background
- What do we write into /etc/mtab for umount.nfs

IPv6 in sunrpc.ko (client)

- No TI-RPC library in the kernel
- rpcbind query support for protocol version 3 and 4
  - Netid support
- Mapped v4 or IPV6_ONLY?
- Separate transport capability for each address family?

IPv6 in sunrpc.ko (server)

- No TI-RPC library in the kernel
- rpcbind registration protocol version 3 and 4 support, with fallback
- Mapped v4 or IPv6_ONLY?
- One listener per address family, or multiple listeners?
- IPv6 support may be disabled dynamically

TI-RPC in User Space

- Linux community inertia
- Sockets v. streams
- Licensing
- How to supersede glibc’s RPC implementation
- Replace portmap with rpcbind
  - RPC over AF_UNIX sockets new to Linux

Down and Dirty

IPv6 Essentials

- Support mounting by (deprecated) site- and link-local addresses?
- Choosing between IPv4 and IPv6 with server that can support both
  - Mount-time choice
  - Allow dynamic switching between families?

Mount.nfs Command

- mount.nfs has it’s own portmap implementation
  - Better control of timeouts and version/protocol fallback
  - Must now support rpcbind v3 and v4
- Square brackets for escaping colons in raw IPv6 addresses

Mount.nfs Command

- mount.nfs determines the NFS version, transport protocol, and now the address family too
  - “proto=” and friends now take a netid
  - “udp” and “tcp” mount options retain their traditional meaning
  - Family negotiated when not specified

Umount.nfs Command

- Picks up mount options from /etc/mtab, but may have to renegotiate certain settings
- mount.nfs uses kernel’s MNT client, umount.nfs uses user space MNT client
- MNTPROC_UMNT is advisory
  - Short timeout
  - Does not affect umount.nfs command’s exit status

NSM

- Many legacy issues with statd and sm-notify already
  - NSM protocol is confusing and deprecated
  - 15-year old code base
- Monitor and notification lists stored in /var/lib/nfs/statd/
  - Directory structure and contents considered a formal API
  - How to store IPv6 addresses?

NSM, continued

- SM_MON upcall limited to either IP address or caller name, not both
  - Caller name: statd can recognize SM_NOTIFY from remote peer
  - IP address: statd can send SM_NOTIFY to correct peer via correct protocol family
- Sticking with IP address for now

NSM, continued

- Kernel depends on value of 16-byte “priv” cookie in NLM downcall
  - Was an IPv4 address padded to 16 bytes
  - Full IPv6 address with address family and other fields won’t fit
- Going with “random” cookie for now

netids and the Linux Kernel

- In user space, TI-RPC controls netid mapping
- Mount options are just a string, so netids are now passed to the kernel
- Kernel has its own rpcbind client
- Kernel must use heuristics and fixed netid mapping for now

Mountd and Exportfs

- Replace gethostby{name,addr} with get{name,addr}info
- The rest are just details
  - TI-RPC MNT service listener
  - ip_map upcall can send IPv6 presentation address
  - rmtab delimits fields with colons; must escape IPv6 addresses

Today’s Feature Set

Current Support, Client Side

- Can mount NFSv2, v3, and v4 servers over IPv6
- Use netids to force protocol family, but negotiate protocol family when netids are not specified
- Auxiliary protocols: NFSv4 callback, NLM locking (lockd, statd), TCP wrappers
- gssd

Current Support, Server Side

- User space rpcbind service already in place
- NLM server-side (lockd & statd) already done
- Kernel rpcbind client for registering kernel services with local rpcbind
- Few remaining kernel pieces targeted for 2.6.34
- User space: rpc.nfsd done, mountd & exportfs prototype in test

Distribution Plans I Know Of

- Client side
  - Fedora 13, RHEL 6 GA
- Server side
  - RHEL 6 update, potentially

Missing Features

Upstream: Untested

- krb5 (gssd), idmapd
- Mounting same export via IPv4 and IPv6
- FS_LOCATIONS
- Full support for (deprecated) site- and link-local
- NFSROOT
- NFSv4.1, especially pNFS

Upstream: Future Work

- Support for configurations with no IPv4 loopback
- Move more NFS mount processing into kernel
- Expose NLM's hosts cache to user space
- Full netid support in kernel

Upstream: Speculative

- Multi-homed NLM - multiple caller_names from same lockd
- AF_UNIX support in kernel
- Replace glibc TS-RPC with libtirpc

Distributor Challenges

- Ubuntu - still using portmap
- SuSE - unknown status
- Debian - unknown status

Next Steps

- Roll out client-side support
- Test, integrate, and roll out server-side support
- Help straggling distributors integrate NFS/IPv6
- Documentation

Questions or comments:

<linux-nfs@vger.kernel.org>