NetApp™

Go further, faster™

spNFS

A Simple pNFS Server

Dan Muntz, Mike Sager, Ricardo Labiaga

Agenda

- What is spNFS?
- Why spNFS?
- How does spNFS work?
- Availability of spNFS?
- Where is spNFS?
- Pros and Anti-pros
- Q&A

What is spNFS?

- spNFS is a files-based pNFS server
  - Keep it simple. Implement basic server functionality
    - Stripe width, stripe type, data servers defined in /etc/spnfsd.conf
  - Design currently splits functionality between kernel and userspace. Implementation can be moved more in either direction as fashion (Linux community) dictates.
  - Uses a non-clustered backend for data servers (DS)
    - Each DS exports an arbitrary file system for stripe storage
- spNFS is an ongoing project of the NFS Client Team at NetApp
  - Dan Muntz, Mike Sager, Ricardo Labiaga

Why spNFS?

- Serves as a testbed for pNFS clients
- May grow into a pNFS server for mainline Linux
- Drive adoption of the pNFS client into mainline Linux (Why the “Client Team” is writing a server)
- Create visibility for pNFS—demos, provide something to give people hands-on experience with pNFS

How does spNFS work?

- Most work is performed by a userspace daemon, `spnfsd` (built as part of nfs-utils)
- Symmetric routines in the kernel and daemon communicate via rpc_pipefs interface

fs/nfsd/vfs.c:

```
nfsd_unlink()
{...
 spnfs_remove(inode)
 ...
}
```

fs/nfsd/spnfs_ops.c

```
spnfs_remove(inode)
{...
 spnfs_upcall(msg)
 ...
}
```

fs/nfsd/spnfs_com.c

```
spnfs_upcall(inode)
{...
 send msg to spnfsd
 via rpc_pipefs
}
```

Kernel

User

spnfsd.c:

```
spnfscb()
{
 receive pipefs msg
 spnfs_msg_handler(msg)
 send pipefs msg
}
```

spnfsd.c

```
spnfs_msg_handler(msg)
{...
 spnfsd_remove(msg)
 ...
}
```

spnfsd_ops.c

```
spnfsd_remove(msg)
{...
 unlink stripe files
 return msg with
 data and status
}
```

How does spNFS work?

- `spnfsd` operations

```
spnfsd_layoutget
spnfsd_layoutcommit
spnfsd_layoutreturn
spnfsd_getdeviceiter
spnfsd_getdeviceinfo
spnfsd_setattr
spnfsd_open
spnfsd_create
spnfsd_remove
spnfsd_read
spnfsd_write
```

How does spNFS work?

- DSs are mounted on the MDS
  - Each DS has a `/pnfs` directory for stripe storage
  - `/pnfs` is exported to the MDS
  - The directories are mounted on the MDS at:
    - `/spnfs/<DS_IP_ADDRESS>`
  - Names are defined in `/etc/spnfsd.conf`
- `spnfsd` performs operations via these mounts
  - E.g., `spnfsd_open()` creates stripes by creating a file in `/spnfs/<DS_IP_ADDRESS>` for the appropriate DSs
  - `spnfsd_remove()` similarly removes stripes
  - Some operations are satisfied by information from `/etc/spnfsd.conf`

How does spNFS work?

- `/etc/spnfsd.conf`

```
[General]
Verbosity = 1
Stripe-size = 8192
Dense-striping = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
DS-Mount-Directory = /spnfs

[DataServers]
NumDS = 2

DS1_IP = 172.16.28.134
DS1_PORT = 2049
DS1_ROOT = /pnfs
DS1_ID = 1

DS2_IP = 172.16.28.141
DS2_PORT = 2049
DS2_ROOT = /pnfs
DS2_ID = 2
```

Availability of spNFS?

- spNFS is available now
- read/write-through-MDS support recently added (and it works as of yesterday)

Where is spNFS?

- spNFS kernel
  - git://linux-nfs.org/~dmuntz/spnfs.git
  - See Documentation/spnfs.txt
- spNFS userspace (nfs-utils with spnfsd)
  - git://linux-nfs.org/~dmuntz/nfs-utils.git
- Bleeding edge kernel
  - git://linux-nfs.org/~bhalevy/linux-pnfs.git

Pros and Anti-pros

- Why spNFS (redux)
  - Linux community pushing for userspace implementation
    - spNFS can easily move more functionality to userspace, or…
    - Can move more functionality into the kernel as needed for performance
  - Simple deployment
    - DSs are NFS servers (or anything that can be mounted into the MDS namespace, modulo control protocol support)
    - Supports arbitrary DS file systems
- Alternatives/Issues
  - Other efforts to provide a free pNFS server for Linux (e.g., CITI/IBM's GFS2-based pNFS server—it's clusterific)
  - Some issues are non-trivial to solve without a clustered storage backend
    - df: how do you calculate available space?
    - User quotas: DSs unaware of total usage per user.
    - File system consistency (fsck): heterogeneous fs consistency
      - Bigger problem for 4.2 striped metadata
    - Efficient read/write-through MDS

NetApp™

Q & A