Client Discovery of NFSv4.1 Server Multipath Addresses

Andy Adamson
andros@netapp.com

IETF 96, Berlin, 2016

Motivation

- Motivated by desire for session trunking
  - Linux prototype code
- pNFS data servers have the GETDEVICEINFO multipath4 to list potential session trunking addresses
- No such protocol feature for non-pNFS or MDS NFSv4.1+ servers
- How does the client discover a list of potential session trunking addresses?
- Meeting topic at June 2016 NFSv4 Bakeathon

Multipath Options for Session Trunking

- IP Layer
  - MPTCP RFC 6820
  - SCPT RFC 4960
- RPC Layer
  - Linux feature: multiple RPC transports per mount
- All take advantage of multiple network paths between the client and the server to:
  - Fully utilize network resource
  - Achieve better throughput
  - Failover for network failures (HA)

MPTCP

- Client and server RPC layer sees a single TCP connection which is multiplexed
  - NFS server can not assign resources per multiplex TCP, but is this really needed…
  - Automatic ‘session trunking‘
  - No support for other transports such as RDMA
- Detects existence of multiple network interfaces on the hosts and creates the multiple TCP flows
  - No need for additional client discovery of addresses
  - New features in discussion to respond to topology
    - Load balancing, path priority to name a few

SCTP

- SCTP association has multiple IP addresses
  - Originally for HA, new load sharing for performance.
- No specification for SCTP in ONC RPC
  - Will the NFS server see one SCTP association or the potentially multiple connections within an association?
  - No support for other transports such as RDMA
- Requires client discovery of server multipath
  - SCTP association is defined as [a set of IP addresses at A]+[Port-A]+[a set of IP addresses at Z]+[Port-Z]
- Still a new protocol that needs work
  - Performance only slightly better than TCP

Linux Multiple RPC transports

- New Linux RPC feature allows for multiple transports per mount
  - Supports a mix of transports (TCP, MPTCP, RDMA, SCTP, …)
- RPC layer controls use
  - Currently only round-robin algorithm implemented
  - Needs work: network topology, which path to prefer
- Server sees each connection as separate
- Requires client discovery of transport addresses

Client Discovery: Multipath Addresses

- Multiple host names on mount command
- Use a DNS A-record, or special trunk record
- Use fs_locations or fs_locations_info attribute on the pseudo file system

Multiple host names on mount

- Solaris supports this feature
  - Disliked due to the need to address all clients when when network conditions change. (as reported at June 2016 Bakeathon)
- Discussion on Linux kernel list
  - Linux RFC prototype received a mixed response
  - Support issues influenced the decision to not implement multiple host names mount feature
    - Response to change in network conditions
    - Dislike another mount option to support
    - Server should supply addresses

DNS for Multipath

- Use DNS A-records, or special trunk record
- Changing a DNS record due to an interface change on a server is problematic
  - Making a DNS change in many organizations is difficult and takes a lot of time as making a mistake that brings DNS down stops all computing services.
- DNS caching means that when a change occurs, need to wait the cache timeout, typically an hour.

fs_locations

- fs_locations/_info useful replica definition:
  - "When a set of servers have corresponding file systems at the same path within their namespaces, an array of server names may be provided."
  - Client could test each replica address for session trunking using the EXCHANGE_ID test.
    - Most addresses will be to different replica servers
- fs_locations/_info is per file system
  - session trunking is server wide.

fs_locations proposal

- Use an fs_locations or fs_locations_info replica attribute on the pseudo file system
  - pseudo file system does not migrate and is not replicated (I guess it could be - but what is the point?
- Define this to apply to the whole server
- Client test each address for session trunking with the EXCHANGE_ID test
  - All addresses will be from the same server

Summary

- Do we need to specify a means of client discovery of NFSv4.1 server multipath addresses?
- MPTCP gets us most of the way for TCP
  - Protocol development is active
- Is fs_locations/_info on pseudo file system a reasonable approach?

NetApp®

Thank you

Multipath Trunk Options

- Testing has shown that multiple connections
  - Improves performance
  - Improves availability
  - Testing:
  - MPTCP
    - http://multipath-tcp.org/pmwiki.php?n=Main.50Gbps
  - Prototype Linux NFS client with multiple RPC transports
  - SCTP
    - https://www.researchgate.net/publication/220776215_Performance_Comparison_of_SCTP_and_TCP_over_Linux_Platform
  - All trunk options require some sort of discovery of multipath addresses