NFS/RDMA
Next Steps

Chuck Lever *Oracle*

What Is NFS/RDMA?

- Direct Memory Access (DMA) – a device transfers data directly to or from host memory
- Remote Direct Memory Access (RDMA) – a device transfers data directly between host memory and memory on other hosts on a network
- NFS/RDMA enables the data payloads of NFS READ and WRITE operations to be transferred between file server and client memory via RDMA

Observed Benefits

Linux v4.12 with Mellanox CX3 Pro at 56Gbps

- With large I/O payloads, 2-4x greater throughput than NFS/TCP
- NFS READ at line rate
- Double the 8KB IOPS rate of NFS/TCP
- Close to bare-metal performance in VM guests

Adoption

- Until recently, specialized hardware was required
- Performance benefits are not apparent with traditional storage technologies on slower networks
- Fallow code remains in the Linux distribution most commonly deployed in HPC and enterprise environments, which is RHEL 6
- There is a long pipeline to market for filesystems

Competition

- SMB Direct in Windows Server [MS-SMB]
- iSER – iSCSI Extensions for RDMA (RFC 7145)
- SRP – SCSI RDMA Protocol (ANSI INCITS 365-2007)
- NVMe/F – NVM Express over Fabrics (revision 1.0)

Accomplishments

IESG Pipeline

- Minty fresh
  - RFC 8166 – Remote Direct Memory Access Transport for Remote Procedure Call Version 1
  - RFC 8167 – Bidirectional Remote Procedure Call on RPC-over-RDMA Transports
- Up next
  - RFC 5667bis – Network File System (NFS) Upper Layer Binding To RPC-Over-RDMA Version One

Personal I-Ds

- andros – client-multipath-discovery
- cel – reminv-design, rpcrdma-cm-pvt-msg, rpcrdma-reliable-reply, rpcrdma-version-two
- dnoveck – nfsulb, rpcrdma-rtrext, rpcrdma-rtissues
- hellwig – rdma-layout, scsi-layout-nvme

Implementation Update

- Open source
  - Upstream Linux – NFSv4.1 on RDMA, RPCSEC with RDMA; experimental support for remote invalidation, large inline threshold
  - Wireshark – improvements to the RPC-over-RDMA dissector
- In prototype
  - Solaris – NFSv4.1 on RDMA

Challenges

Storage Advances

- *Latency*
    - Traditional NFS servers manage storage devices whose persistence latencies are measured in milliseconds
    - Storage Class Memory persistence latencies are measured in microseconds (or less)
- SCM persistence latency is ***smaller*** than the latency added by typical NFS and RPC client stacks

Storage Advances

- *Memory semantics*
  - Synchronous – No need for server threads to context-switch to guarantee data persistence
  - Cache-less – NFS data payloads can be placed directly into filesystems-in-memory
- Client could DMA data directly to server’s non-volatile memory to avoid I/O and data copying

NFS Server Operation

- Extra RDMA Read round-trip per RPC impacts operations with moderately sized RPC Call messages

- Protocol was architected to drive RDMA Read from XDR layer, but servers need to interpret NFS file handle to fully implement direct data placement during NFS WRITE operations

NFS Client Operation

- Handle POSIX signals non-destructively
  - ^C invalidates Write chunks, server tries to write result data into them: Remote Access Error
- Credit management corner cases
  - Detecting connection loss and server crashes
  - RPC retransmission
  - Unidirectional messages

Security

- Network and host multi-tenancy
- Integrity and confidentiality of Transport Headers
- Confidentiality with good performance
- Avoiding server DoS on ultra-fast networks

Interoperability

- Handle a broader variety of transport errors
  - From v1: VERS, BAD_XDR
  - Extensibility: INVAL_PROC, INVAL_OPTION
  - Expose chunk handling limits: READ_CHUNKS, WRITE_CHUNKS, SEGMENTS
  - Handle incorrect reply size estimation: WRITE_RESOURCE, REPLY_RESOURCE
  - All other: SYSTEM

Interoperability

- Eliminate reply size estimation
  - Currently requester estimates reply size and registers a large Write sink buffer
  - But many replies fit inline
  - For large replies, responder could instead expose a Read chunk, sends RDMA_NOMSG reply; requester pulls reply via RDMA Read

Interoperability

- Exchange transport properties
- Introduce transport protocol extensibility
- Handle multiple Read or Write chunks per RPC
- Enable Send- and Receive-in-place

Campaign Priorities

Grouping One

- Enable transport parallelism at the NFS layer
  - NFSv4 multi-path
  - pNFS SCSI layout type support for NVMe/F
  - pNFS RDMA layout type

Grouping Two

- Incrementally improve RPC-over-RDMA version 1
  - Replace CCP with per-connection property exchange to enable remote invalidation, large inline thresholds, and a few other features
  - rpcrdma-cm-pvt-msg is one way to do this, but there was an objection to using RDMA CM private data as a standard property exchange mechanism

Grouping Three

- Pursue RPC-over-RDMA version 2
  - Improvements in error recovery, reply size estimation, security
  - In-band transport property exchange to enable remote invalidation and larger inline thresholds
  - Transport extensibility

Discussion

- One – Enable transport parallelism at the NFS layer
- Two – Improve RPC-over-RDMA version 1
- Three – Pursue RPC-over-RDMA version 2
- Others?