
ORACLE®

Upstream Linux NFS/RDMA

2015 progress, 2016 plans

Chuck Lever
Linux Kernel Architect
Corporate Architecture, Upstream Linux Engineering
March 3, 2016

ORACLE

Program Agenda

1. Client updates and fixes
2. Server updates and fixes
3. Standards work
4. What's ahead
5. Fifth Element

Client Highlights

NFSv4.1, performance

NFSv4.1 With RDMA

Backchannel and other operational changes

- Designed and published conventions for backchannel operation over RPC-over-RDMA
- Client side backchannel implemented
- Larger COMPOUNDs
  - OPEN, LOOKUP
  - GETACL / SETACL
  - NFSv4.2 security labels

Increase r/wsize And Credit Limit

Improved utilization of hardware resources

- Don’t pre-allocate worst-case number of MRs
- Send/Receive buffers now managed on a list instead of a stack
- Acquire and recover MRs during registration to simplify disconnect recovery
- New maximum r/wsize is 1MB, credit limit is 128

Process RPC Replies Via Work Queue

Use unbound work queue instead of tasklet

- Reply handling no longer single-threaded
- Disabling IRQs no longer necessary
- Can perform synchronous MR invalidation before RPC completion
- Implicit flow control of Send Queue

Prepare For Hardware Advances

Call-outs for registration, invalidation

- Use of ib_alloc_mr() allows registration of arbitrary memory regions
- Use of per-PD lkey
- PHYSICAL no longer an automatic fallback mode
- Separate source files for FRWR, FMR, and PHYSICAL modes

Miscellaneous Improvements

Reliability and observability

- Send Read chunks correctly (tail buffer fix)
- Report human-readable errors
- Support swap-on-NFS/RDMA
- Transport fault injection
- Pin device during NFS mounts

Server Highlights

NFSv4.1, stability

NFSv4.1 With RDMA

Backchannel and other operational changes

- Designed and published conventions for backchannel operation over RPC-over-RDMA
- Server side backchannel implemented
- CREATE_SESSION adjustments
- Pre-allocate more control structures

Increase r/wsize

Improved utilization of hardware resources

- Address several bugs hit only with large READ or WRITE requests
- Observe device limits more carefully
  - FRWR page depth
  - max_sge_rd
- Disconnect client and server r/wsize maxima
- Bump server side maximum

Protocol Support Enhancements

- Support RDMA_NOMSG Call messages
- Handle trailing inline content in Call messages

Prepare For Hardware Advances

Call-outs for registration, invalidation

- Use of ib_alloc_mr() allows registration of arbitrary memory regions
- Use of per-PD lkey
- Remove open-coded checks for iWARP v. IB

NFS/RDMA Standards

NFSv4.1, extensibility

RFC 5666 and RFC 5666bis

RPC-over-RDMA revamp underway

- RFC 5666 appears to be incomplete
- Implementation experience I-D documents many issues
- nfsv4 WG approved RFC 5666bis to replace RFC 5666
  - Mission: level set, document current implementations
- rfc5666bis-04 now available on datatracker.ietf.org
- Finishing touches in 2016

RFC 5667

How NFS operates on RPC-over-RDMA

- NFS/RDMA Binding also needs update
  - Discussion of NFS COMPOUND is incomplete
  - No remarks about backchannel
  - Some guidance made obsolete by rfc5666bis
- nfsv4 WG is aware of these issues
- Watch this space

Accessing Persistent Memory Via RDMA

Push model, RDMA Commit

- Fast networks and storage work better if clients to initiate RDMA operations
  - Servers expose PMEM
  - Clients drive RDMA Read and Write
  - Change from current RDMA-enabled storage protocols
- No durability guarantees after RDMA Write
  - Clients require a new RDMA operation (Commit) to remotely flush written data from caches onto durable storage
- Currently at the proposal stage

RPC-Over-RDMA Futures

Growth opportunities

- Desirable features
  - Remote invalidation
  - In-band negotiation of connection parameters
  - Push model
- RPC-over-RDMA Version One is creaky; enabling extensibility has been a challenge
- RPC-over-RDMA Version Two is also a possibility

Looking Ahead

Security, performance

ORACLE
CLOUD SOLUTIONS

Outlook: Cloudy.

Applications. Platform. Infrastructure.

Client/Server: RPCSEC_GSS On RPC-Over-RDMA

Kerberos with NFS/RDMA

- Larger authenticators and verifiers want larger inline thresholds
- Authentication-only a matter of shooting down bugs
- Integrity/privacy require bounce buffers
- Standards guidance needed

Server: NFS WRITE Performance Improvements

Several attack vectors

- Larger inline thresholds
- Drive RDMA Reads from a work queue
- Zero-copy NFS WRITE: splice
- Faster FH checking per operation
- NFS open caching?

Client/Server: NFSv4.1 Enhancements

NFSv4.1 set to dominate NFS deployments

- Larger inline thresholds
  - Larger NFSv4 COMPOUNDs
  - Larger backchannel operations
- More backchannel parallelism
- Session trunking may permit multiple QPs per mount point
- Experience with pNFS

Client/Server: Overhaul Of Linux Kernel IB Core API

Enable new hardware capabilities, reduce code duplication

- 2015
  - Page vectors replaced with s/g lists
  - ib_devattr merged into ib_device
- Now
  - New CQ API to enable common functions hidden in core
  - New ib_drain_qp API
- 1H 2016
  - New RDMA Read API to hide differences between iWARP and IB

Client: Full MR Fencing Before Completion

Close memory exposure windows

- MR invalidation is asynchronous when RPC terminates due to
  - POSIX signal
  - RPC soft timeout
  - Local I/O error during reply decode
- MR is still exposed briefly while memory is re-used
- Signal is worst: server can reply before invalidation completes, and update client memory
- Fixing may require changes to the RPC client finite state machine

Client: Device Detach With Active NFS Mounts

Device is pinned by active mounts

- MR invalidation runs in parallel with transport reconnect
- Detach would require replacing transport resources that could be in use by completions and other asynchronous events
- To address this very rare usage scenario might require aggressive resource management such as locking and refcounting (i.e., have undesirable performance impact)

Q & A

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Hardware and Software
Engineered to Work Together