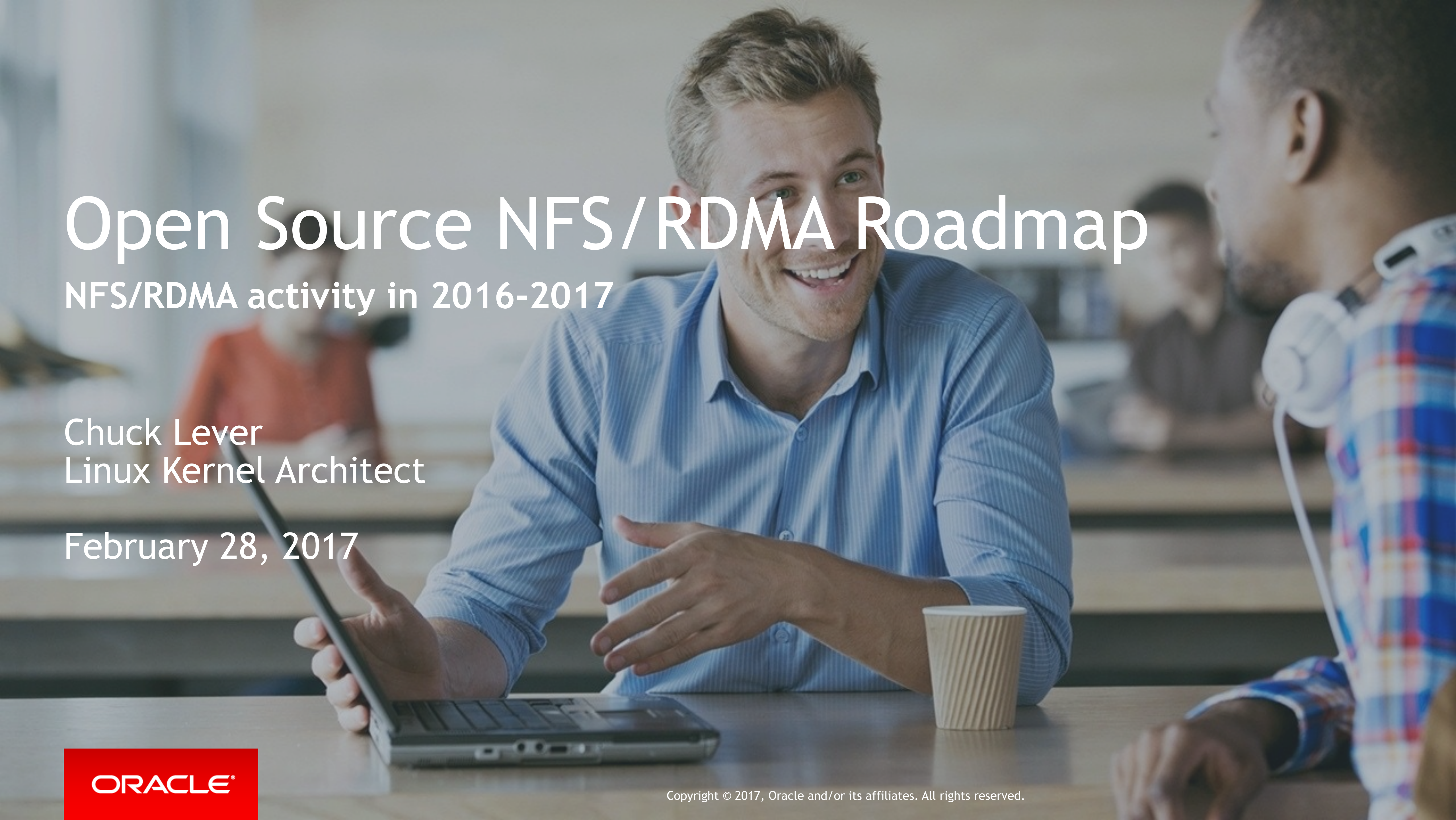


Open Source NFS/RDMA Roadmap

NFS/RDMA activity in 2016-2017

Chuck Lever
Linux Kernel Architect

February 28, 2017

ORACLE

Program Agenda

1. Upstream progress and forecast
2. RPC-over-RDMA Version One shortcomings
3. Addressing transport protocol issues

ORACLE

Upstream Activity In 2016

Linux NFS/RDMA Client

Major New Features

- NFSv4.1 support with backchannel
- NFS/RDMA with krb5, krb5i, krb5p
- Regular testing with IOMMU enabled
- Memory registration mode changes
  - Unsafe ALLPHYSICAL mode removed
  - Support for SG_GAP added to FRWR
  - Remaining modes are FRWR and FMR

ORACLE

Linux NFS/RDMA Client

Internal Improvements

- Transport resource requirements reduced
  - MRs allocated on demand (sets of 32)
  - Some large data structures moved off the stack
- Disconnection recovery made even more robust
- Converted to new rdma core APIs
  - ib_alloc_cq
  - ib_drain_qp
  - FMR scatterlist

ORACLE

Linux NFS/RDMA Client

Experimental Features

- Responder's choice Remote Invalidation
- Inline thresholds up to 64KB using gathered Send
- On connection, transport properties exchange via CM private data

Linux NFS/RDMA Server

Major New Features

- NFSv4.1 support with backchannel
- NFS/RDMA with krb5, krb5i, krb5p
- Regular testing with IOMMU enabled

Linux NFS/RDMA Server

Internal Improvements

- Complete support for RDMA_ERROR type messages
- IPv6 support completed
- Converted to new rdma core APIs
  - ib_alloc_cq

ORACLE

Linux NFS/RDMA Server

Experimental Features

- Responder's choice Remote Invalidation
- Inline thresholds up to 64KB
- On connection, transport properties exchange via CM private data

Wireshark

Fixes now in 2.3.0-rc and 2.2

- RPC-over-RDMA frame detection is now reliable
- RPC-over-RDMA Transport header parsing is working
  - Parse tree constructed and displayed properly
  - Display filters on header fields now functional
- RPC Call/Reply matching has been improved

ORACLE

Upstream Forecast

Linux NFS/RDMA Client

- Multi-path support
- Faster recovery from server reboot/failover
- Handle device driver unloading
- Move development platform from IB to RoCE
- Focus on full stack performance

ORACLE

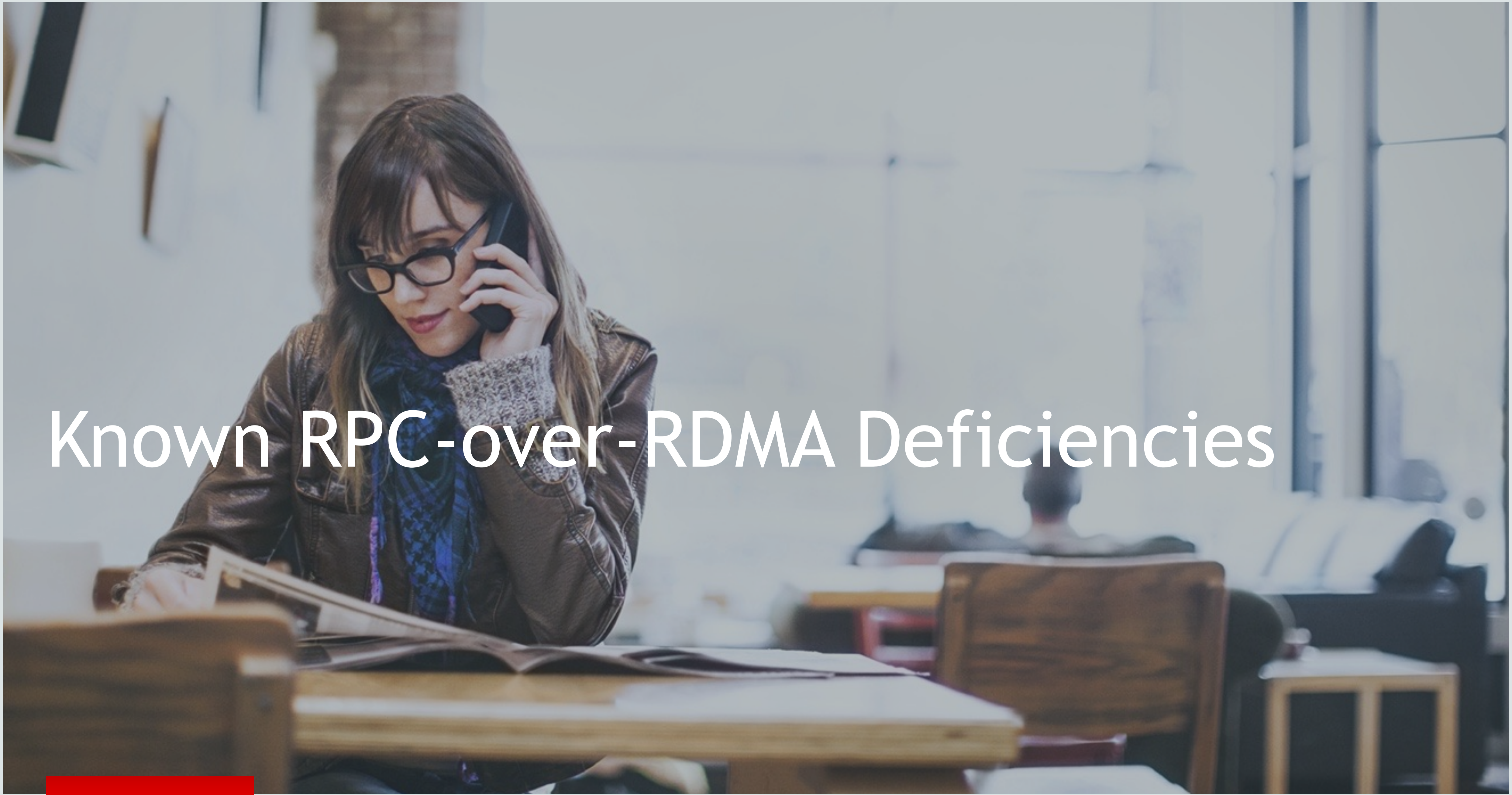
Linux NFS/RDMA Server

- Conversion to core rdma_rw API
- Multi-path support
- Improve Receive efficiency
  - Reduce load on memory allocator
  - Less DMA mapping
- Move development platform from IB to RoCE

Wireshark

Next Steps

- Re-assembly of RDMA_NOMSG messages
- Re-assembly of RDMA_MSG messages with chunks



Known RPC-over-RDMA Deficiencies

ORACLE

Reply Size Estimation

- Requesters must provide adequate resources to receive RPC replies
- Requesters struggle to determine the maximum size of certain variable-length elements in Upper Layer Protocols
  - Examples include ACLs, NFSv4.2 READ_PLUS results
  - Is this a ULP design problem, or a deficiency in the transport?
- Responders have no mechanism to report a catastrophic lack of reply resources

ORACLE

Cancelled RPCs

- An RPC transaction can be cancelled due to an asynchronous event on the requester (*e.g.* ^C)
- Nevertheless, responder still tries to write results into a Write or Reply chunk
  - If the MR is now invalid, connection is lost
  - If the MR was re-used, reply data might be corrupted
- Requesters must invalidate chunks immediately or set them aside until the responder has sent a matching reply

ORACLE

Multiple Write Chunks

- Read chunks have XDR position fields, Write chunks do not
  - In other words, Write chunks are not fully self-describing
- Requester can provide multiple Write chunks, but it may be ambiguous how the responder consumes them
- How does requester know which result went into which Write chunk?

ORACLE

Security Issues

- RPC-over-RDMA Transport header fields are not protected by RPCSEC GSS
- GSS integrity and privacy services make it difficult to offload data transfer
- RDMA consumers need to detect the presence of offloaded security (*e.g.* iWARP on IPsec)

ORACLE

Performance Issues

- No convenient way to support Remote Invalidation with clients who use persistent memory registration
- Receive still requires some data copying
- Default inline threshold (1KB) is inadequate for NFSv4
- Existing RPC stacks are too slow for low-latency transports and storage
- Single QP I/O throughput is hardware-limited

ORACLE

Performance Issues

RDMA Read

- RDMA Read requires an extra round trip
- Transport cannot do RDMA Read efficiently by itself
  - Current implementations Read into anonymous pages and copy or flip them into their page cache
  - Upper layer (NFS) knows the eventual pages where data needs to land
  - Upper layer must handle non-page-aligned NFS WRITE requests, in order to avoid copying some payload data

ORACLE

Structural Issues

- Computing precisely how much RPC message data can fit inline is difficult because chunk lists, which share the inline space with the RPC message, vary in length
- No clean way to extend the RPC-over-RDMA protocol without a new version
- No control plane to do things such as:
  - Connection keep-alive and testing
  - Exchange of transport properties
  - Post-retransmit credit resyncs

ORACLE

Structural Issues

Credit Accounting

- Credit accounting assumes every RPC-over-RDMA message today is associated with exactly one RPC message
  - Might want to send a keep-alive, or exchange transport properties
  - A control plane would have to share Receive resources with data plane
  - Might want to send a single large RPC using multiple RDMA Sends (credits)
  - Might want to send several RPCs in one RDMA Send
  - No call direction field in RPC-over-RDMA Transport header: direction of an RPC-over-RDMA message not carrying an RPC message is not known

ORACLE

Meeting These Challenges

ORACLE

Performance On RPC-over-RDMA Version One

Use NFSv4.1 and pNFS

- Multi-path enables the use of more than one QP per mount point
- pNFS block layouts with iSER or NVMe/F enables efficient RDMA-native I/O to multiple DSes
- Possible new “push mode” layout type enables client to drive RDMA Read and Write directly to remote persistent memory
  - No extra RDMA Read round-trip when client initiates RDMA
  - No interrupts on the server during read and write I/O
  - Direct access to storage

ORACLE

RPC-over-RDMA Version Two

New Transport Protocol Features

- Larger default inline threshold
- Flexible support for Remote Invalidation
- Message direction now part of Transport header
- Extensibility
  - Built-in exchange of transport properties
  - Ability to introduce new messsage types

ORACLE

RPC-over-RDMA Version Two

Extensions To Make More Use of RDMA Send

- Message Continuation
  - Use multiple RDMA Send operations to transmit a large RPC
  - No memory registration or invalidation is needed
  - Reduces the risk of requesters providing inadequate receive resources
  - Makes credit accounting more complex

ORACLE

RPC-over-RDMA Version Two

Extensions To Make More Use of RDMA Send

- Send-based direct data placement
  - Enable the receiver to catch aligned data in buffers that can be flipped into a file’s page cache
  - Used to transfer DDP eligible data items
  - Not effective for updating data in a page with other data, or data that starts in the middle of a page
  - Not effective for platforms with pages larger than the inline threshold

ORACLE

Unresolved Areas

- Handling cancelled RPCs
- Creating and managing a control plane
- Generic zero-copy receive
- Protecting the Transport header
- Reliably matching Write chunks to results
- Using chunks in the backchannel

ORACLE

Follow Along At Home

- draft-andros-nfsv4-client-multipath-discovery
- draft-cel-nfsv4-reminv-design
- draft-cel-nfsv4-rpcrdma-cm-pvt-msg
- draft-cel-nfsv4-rpcrdma-version-two
- draft-dnoveck-nfsv4-rpcrdma-rtissues
- draft-dnoveck-nfsv4-rpcrdma-rtrext

ORACLE


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.