RPC-over-RDMA version two
Credit accounting revisited

Chuck Lever <chuck.lever@oracle.com>
Linux Prototype

- Based on the existing v1 implementation
- Client and server
- Limited: only v1 credit accounting; no transport properties, peer authentication, or new error codes
- Note that without full v2 credit accounting, the prototype can’t do important new pieces of rpcrdma-version-two such as message continuation or control plane messages
Challenges with Flow Control

- rpcrdma-version-two-04 Section 4.2.1.1 (Granting Credits) is not implementable:
  - The rdma_credits field adequately advertises the two credit windows
  - But an RPC Reply no longer carries an implicit single credit ACK, since there is no longer a strict one-to-one relationship between RPC message and RDMA message
  - Thus there’s no way for a sender to determine how many credits the receiver has already consumed
Challenges with Flow Control

- Proposal: Use a classic credit-based flow control protocol instead of what is described in S 4.2.1
- RDMA Send/Receive channel ops are reliable and in-order
  - Therefore the number of messages sent/received since the connection was established is an implicit sequence number
  - Each sender provides, via message header fields:
    - A credit grant (a.k.a a window size)
    - The number of messages received so far on this connection
Challenges with Flow Control

- Proposed wire changes (see -05)
  - Replace the single split 32-bit rdma_credits field
    - Re-use rdma_credits field as the sender’s receive credit window size
    - New 32-bit field (or some other protocol element) to convey the number of messages the sender has received on the connection
Challenges with Flow Control

• Understanding the boundary between protocol and algorithm
  • The spec specifies protocol elements and their semantics
  • It also specifies when senders must constrain their transmission based on the advertised window
  • No other discussion of algorithm is provided
A Modest Proposal
New IANA Registries

• QUIC RFCs define new IANA registries for error codes and transport properties. Should RPC/RDMA version 2?

• What about other aspects of the protocol, such as header types?
WG Bureaucratic Actions

- Extend the milestone date for delivery of rpcrdma-version-two
- Evaluate the priority of work on rpcrdma-version-two based on:
  - Current number of RPC/RDMA v2 prototypes
  - The expected benefits of the new protocol elements
  - Other projects in front of the WG (i.e., rfc5661bis, QUIC/TLS, etc)
  - Available prototyping, authorship, and review resources
Prototype Next Steps

• Near-term:
  • Implement proposed credit accounting protocol
  • Implement message continuation

• Later:
  • Transport properties

• Peer authentication is still under-specifed
Supplemental Material
Bibliography

• RFC 8166 - RPC over an RDMA Transport

• https://datatracker.ietf.org/doc/draft-ietf-nfsv4-rpcrdma-version-two