Network File System Version 4
Internet-Draft
Intended status: Informational
Expires: September 20, 2018

C. Lever
Oracle

March 19, 2018

RDMA Connection Manager Private Data For RPC-Over-RDMA Version 1
draft-cel-nfsv4-rpcrdma-cm-pvt-msg-03

Abstract

This document specifies the format of RDMA-CM Private Data exchanged between RPC-over-RDMA version 1 peers as a transport connection is established. Such messages indicate peer support for Remote Invalidation and larger-than-default inline thresholds. The message format is extensible.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 20, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. Requirements Language 3
3. Advertised Transport Capabilities 3
3.1. Inline Threshold Size 3
3.2. Remote Invalidation 4
4. Private Data Message Format 5
4.1. Fixed Mandatory Fields 5
4.2. Extending the Message Format 7
5. Interoperability Considerations 7
6. IANA Considerations 7
7. Security Considerations 7
8. References 7
8.1. Normative References 8
8.2. Informative References 8
Acknowledgments 8
Author's Address 9

1. Introduction

The RPC-over-RDMA version 1 transport protocol enables the use of RDMA data transfer for upper layer protocols based on RPC [RFC8166]. The terms "Remote Direct Memory Access" (RDMA) and "Direct Data Placement" (DDP) are introduced in [RFC5040].

The two most immediate shortcomings of RPC-over-RDMA version 1 are:

- Setting up an RDMA data transfer (via RDMA Read or Write) can be costly. The small maximum size of messages transmitted using RDMA Send forces the use of RDMA Read or Write operations even for relatively small messages and data payloads.

- Unlike most other contemporary RDMA-enabled storage protocols, there is no facility in RPC-over-RDMA version 1 that enables the use of Remote Invalidation [RFC5042].

The original specification of RPC-over-RDMA version 1 provided an out-of-band protocol for passing inline threshold values between connected peers [RFC5666]. However, [RFC8166] eliminated support for this protocol making it unavailable for this purpose.

RPC-over-RDMA version 1 has no means of extending its XDR definition such that interoperability with existing implementations is preserved. As a result, an out-of-band mechanism is needed to help relieve these limitations for existing RPC-over-RDMA version 1 implementations.

This document specifies a simple, non-XDR-based message format designed to pass between RPC-over-RDMA version 1 peers as each RDMA transport connection is first established. The purpose of this message format is two-fold:

- To provide immediate relief from certain performance constraints inherent in RPC-over-RDMA version 1

- To enable experimentation with parameters of the base RDMA transport over which RPC-over-RDMA runs

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Advertised Transport Capabilities

3.1. Inline Threshold Size

Section 3.3.2 of [RFC8166] defines the term "inline threshold." An inline threshold is the maximum number of bytes that can be transmitted using only one RDMA Send and one RDMA Receive. There are a pair of inline thresholds per transport connection, one for each direction of message flow.

If an incoming message exceeds the size of a receiver's inline threshold, the receive operation fails and the connection is typically terminated. To convey a message larger than a receiver's inline threshold, an NFS client uses explicit RDMA data transfer operations, which are more expensive to use than RDMA Send.

The default value of inline thresholds for RPC-over-RDMA version 1 connections is 1024 bytes in both directions (see Section 3.3.3 of [RFC8166]). This value is adequate for nearly all NFS version 3 procedures.

NFS version 4 COMPOUNDs are larger on average than NFSv3 procedures, forcing clients to use explicit RDMA operations for frequently-issued requests such as LOOKUP and GETATTR. The use of RPCSEC_GSS security also increases the average size of RPC messages, due to the larger size of credential material in RPC headers [RFC7861].

If a sender and receiver can somehow agree on larger inline thresholds, more RPC transactions avoid the cost of explicit RDMA operations.

3.2. Remote Invalidation

After an RDMA data transfer operation completes, an RDMA peer can use Remote Invalidation to request that the remote peer RNIC invalidate an STag associated with the data transfer [RFC5042].

An RDMA consumer requests Remote Invalidation by posting an RDMA Send With Invalidate Work Request in place of an RDMA Send Work Request. The RDMA Send With Invalidate carries the R_key value of the STag to invalidate. Invalidation of that R_key is performed and then reported as part of the completion of a waiting Receive Work Request.

An RPC-over-RDMA responder might use Remote Invalidation when replying to an RPC request that provided Read or Write chunks. The requester avoids an extra Work Request, context switch, and interrupt to invalidate one chunk as part of completing an RPC transaction. The upshot is faster completion of RPC transactions that involve RDMA data transfer.

There are some important caveats which might contraindicate the use of Remote Invalidation:

- Remote Invalidation is not supported by all RNICs.

- Not all RPC-over-RDMA requester implementations can recognize when Remote Invalidate has occurred.

- Not all RPC-over-RDMA responder implementations can generate RDMA Send With Invalidate Work Requests.

- An RPC-over-RDMA requester that supports Remote Invalidation may choose to use R_keys that must not be invalidated remotely.

- On one connection, RPC-over-RDMA requesters can mix R_keys that may be invalidated remotely with some that must not.

- RPC-over-RDMA requesters often register more than one R_key per RPC. In one RPC, they can mix R_keys that may be invalidated remotely with some that must not.

Thus a responder must not employ Remote Invalidation unless it is aware of support for it in its own RDMA stack, and on the requester. And, without altering the XDR structure of RPC-over-RDMA version 1 messages, it is not possible to support Remote Invalidation with

requesters that mix R_keys that may and must not by invalidated remotely.

However, it is possible to provide a simple signaling mechanism for a requester to indicate it can deal with Remote Invalidation of any R_key it presents to a responder.

4. Private Data Message Format

With an InfiniBand lower layer, for example, RDMA connection setup uses the InfiniBand Connection Manager to establish a Reliable Connection [IBARCH]. When an RPC-over-RDMA version 1 transport connection is established, the client (which actively establishes connections) and the server (which passively accepts connections) MAY populate the CM Private Data field exchanged as part of CM connection establishment.

The transport properties exchanged via this mechanism are fixed for the life of the connection. Each new connection presents an opportunity for a fresh exchange.

For RPC-over-RDMA version 1, the CM Private Data field is formatted as described in this section. RPC clients and servers use the same format. If the capacity of the Private Data field is too small to contain this message format, or the underlying RDMA transport is not managed by a Connection Manager, CM Private Data cannot be used.

4.1. Fixed Mandatory Fields

The first 8 octets of the CM Private Data field is to be formatted as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Magic Number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |     Flags     |   Send Size   | Receive Size  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Magic Number
   This field contains a fixed 32-bit value that identifies the content of the Private Data field as an RPC-over-RDMA version 1 CM Private Data message. The value of this field MUST be 0xf6ab0e18, in big-endian order.

Version

This 8-bit field contains a message format version number. The value "1" in this field indicates that exactly eight octets are present, that they appear in the order described in this section, and that each has the meaning defined in this section.

Bit Flags

This 8-bit field contains eight bit flags that indicate the support status of optional features, such as Remote Invalidation. The meaning of these flags is defined in Section 4.1.1.

Send Size

This 8-bit field contains an encoded value corresponding to the maximum number of bytes this peer will transmit in a single RDMA Send. The value is encoded as described in Section 4.1.2.

Receive Size

This 8-bit field contains an encoded value corresponding to the maximum number of bytes this peer can receive with a single RDMA Receive. The value is encoded as described in Section 4.1.2.

The requester MUST use the smaller of its own send size and the responder's reported receive size as the requester-to-responder inline threshold. The responder MUST use the smaller of its own send size and the requester's reported receive size as the responder-to-requester inline threshold.

4.1.1. Feature Support Flags

The bits in the Flags field are labeled from bit 8 to bit 15, as shown in the diagram above. When the Version field contains the value "1", the bits in the Flags field have the following meaning:

Bit 15

When a requester sets this flag, it sends only R_keys that can tolerate Remote Invalidation. When a responder sets this flag, it can generate RDMA Send With Invalidate Work Requests. When both peers on a connection set this flag, the responder MAY use RDMA Send With Invalidate when transmitting RPC Replies. When either peer on a connection clear this flag, the responder MUST use RDMA Send when transmitting RPC Replies.

Bits 14 - 8

These bits are reserved and must be zero.

4.1.2. Encoding the Inline Threshold Value

Inline threshold sizes from 1KB to 256KB can be represented in the Send Size and Receive Size fields. A sender computes the encoded value by dividing the actual value by 1024 and subtracting one from the result. A receiver decodes this value by performing a complementary set of operations.

4.2. Extending the Message Format

The Private Data format described above can be extended by adding additional optional fields which follow the first eight octets, or by making use of one of the reserved bits in the Flags fields. To introduce such changes while preserving interoperability, a new Version number is to be allocate, and new fields and bit flags are to be defined. A description of how receivers should behave if they do not recognize the new format is to be provided as well. Such situations may be addressed by specifying the new format in a document updating this one.

5. Interoperability Considerations

This extension is intended to interoperate with RPC-over-RDMA version 1 implementations that do not support the exchange of CM Private Data. When a peer does not receive a CM Private Data message which conforms to Section 4, it MUST act as if the remote peer supports only the default RPC-over-RDMA version 1 settings as defined in [RFC8166]. In other words, the peer is to behave as if a Private Data message was received in which bit 8 of the Flags field is zero. and both Size fields contain the value zero.

6. IANA Considerations

This document does not require actions by IANA.

7. Security Considerations

RDMA-CM Private Data typically traverses the link layer in the clear. A man-in-the-middle attack could alter the settings exchanged at connect time such that one or both peers might perform operations that result in premature termination of the connection.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5040] Recio, R., Metzler, B., Culley, P., Hilland, J., and D. Garcia, "A Remote Direct Memory Access Protocol Specification", RFC 5040, DOI 10.17487/RFC5040, October 2007, <http://www.rfc-editor.org/info/rfc5040>.

[RFC5042] Pinkerton, J. and E. Deleganes, "Direct Data Placement Protocol (DDP) / Remote Direct Memory Access Protocol (RDMAP) Security", RFC 5042, DOI 10.17487/RFC5042, October 2007, <http://www.rfc-editor.org/info/rfc5042>.

[RFC8166] Lever, C., Ed., Simpson, W., and T. Talpey, "Remote Direct Memory Access Transport for Remote Procedure Call Version 1", RFC 8166, DOI 10.17487/RFC8166, June 2017, <http://www.rfc-editor.org/info/rfc8166>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <http://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[IBARCH] InfiniBand Trade Association, "InfiniBand Architecture Specification Volume 1", Release 1.3, March 2015, <http://www.infinibandta.org/content/pages.php?pg=technology_download>.

[RFC5666] Talpey, T. and B. Callaghan, "Remote Direct Memory Access Transport for Remote Procedure Call", RFC 5666, DOI 10.17487/RFC5666, January 2010, <http://www.rfc-editor.org/info/rfc5666>.

[RFC7861] Adamson, A. and N. Williams, "Remote Procedure Call (RPC) Security Version 3", RFC 7861, DOI 10.17487/RFC7861, November 2016, <http://www.rfc-editor.org/info/rfc7861>.

Acknowledgments

Thanks to Christoph Hellwig and Devesh Sharma for suggesting this approach. The author also wishes to thank Bill Baker and Greg Marsden for their support of this work.

Special thanks go to Transport Area Director Spencer Dawkins, NFSV4 Working Group Chair Spencer Shepler, and NFSV4 Working Group Secretary Thomas Haynes.

Author's Address

Charles Lever
Oracle Corporation
1015 Granger Avenue
Ann Arbor, MI 48104
United States of America

Phone: +1 248 816 6463
Email: chuck.lever@oracle.com