Network File System Version 4
Internet-Draft
Obsoletes: 5667 (if approved)
Intended status: Standards Track
Expires: January 1, 2017

C. Lever, Ed.
Oracle

June 30, 2016

Network File System (NFS) Upper Layer Binding To RPC-Over-RDMA draft-ietf-nfsv4-rfc5667bis-01

Abstract

This document specifies the Upper Layer Bindings of Network File System (NFS) protocol versions to RPC-over-RDMA transports. Such Upper Layer Bindings are required to enable RPC-based protocols to use direct data placement when conveying large data payloads on RPC-over-RDMA transports. This document obsoletes RFC 5667.

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 http://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 January 1, 2017.

Copyright Notice

Table of Contents

1. Introduction

Remote Direct Memory Access Transport for Remote Procedure Call, Version One [I-D.ietf-nfsv4-rfc5666bis] (RPC-over-RDMA) enables the use of direct data placement to accelerate the transmission of large data payloads associated with RPC transactions.

Each RPC-over-RDMA transport header can convey lists of memory locations involved in direct transfers of data payloads. These memory locations correspond to XDR data items defined in an Upper Layer Protocol (such as NFS).

To facilitate interoperation, RPC client and server implementations must agree on what XDR data items in which RPC procedures are eligible for direct data placement (DDP).

This document specifies the set of XDR data items in each of the following NFS protocol versions that are eligible for DDP. It also contains additional material required of Upper Layer Bindings as specified in [I-D.ietf-nfsv4-rfc5666bis].

o NFS Version 2 [RFC1094]

- NFS Version 3 [RFC1813]
- NFS Version 4.0 [RFC7530]
- NFS Version 4.1 [RFC5661]
- NFS Version 4.2 [I-D.ietf-nfsv4-minorversion2]

The Upper Layer Binding specified in this document can be extended to cover the addition of new DDP-eligible XDR data items defined by versions of the NFS version 4 protocol specified after this document has been ratified.

1.1. 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 [RFC2119].

1.2. Changes Since RFC 5667

Corrections and updates made necessary by new language in [I-D.ietf-nfsv4-rfc5666bis] has been introduced. For example, references to deprecated features of RPC-over-RDMA Version One, such as RDMA_MSGP, and the use of the Read list for handling RPC replies, has been removed. The term "mapping" has been replaced with the term "binding" or "Upper Layer Binding" throughout the document. Material that duplicates what is in [I-D.ietf-nfsv4-rfc5666bis] has been deleted.

Material required by [I-D.ietf-nfsv4-rfc5666bis] for Upper Layer Bindings that was not present in [RFC5667] has been added, including discussion of how each NFS version properly estimates the maximum size of RPC replies.

The following changes have been made, relative to [RFC5667]:

- Ambiguous or erroneous uses of RFC2119 terms have been corrected.
- References to specific data movement mechanisms have been made generic or removed.
- References to obsolete RFCs have been replaced.
- Technical corrections have been made. For example, the mention of 12KB and 36KB inline thresholds have been removed. The reference to a non-existant NFS version 4 SYMLINK operation has been replaced with NFS version 4 CREATE(NF4LNK).

- An IANA Considerations Section has replaced the "Port Usage Considerations" Section.

- Code excerpts have been removed, and figures have been modernized.

- Language inconsistent with or contradictory to [I-D.ietf-nfsv4-rfc5666bis] has been removed from Sections 2 and 3, and both Sections have been combined into Section 2 in the present document.

- An explicit discussion of NFSv4.0 and NFSv4.1 backchannel operation will replace the previous treatment of callback operations. No NFSv4.x callback operation is DDP-eligible.

- The binding for NFSv4.1 has been completed. No additional DDP-eligible operations exist in NFSv4.1.

- A binding for NFSv4.2 has been added that includes discussion of new data-bearing operations like READ_PLUS.

1.3. Planned Changes To This Document

The following changes are planned, relative to [RFC5667]:

- The discussion of NFS version 4 COMPOUND handling will be completed.

- Remarks about handling DDP-eligibility violations will be introduced.

- A discussion of how the NFS binding to RPC-over-RDMA is extended by standards action will be added.

2. Conveying NFS Operations On RPC-Over-RDMA Transports

Definitions of terminology and a general discussion of how RPC-over-RDMA is used to convey RPC transactions can be found in [I-D.ietf-nfsv4-rfc5666bis]. In this section, these general principals are applied to the specifics of the NFS protocol.

2.1. Use Of The Read List

The Read list in each RPC-over-RDMA transport header represents a set of memory regions containing DDP-eligible NFS argument data. Large data items, such as the file data payload of an NFS WRITE request, are referenced by the Read list and placed directly into server memory.

XDR unmarshaling code on the NFS server identifies the correspondence between Read chunks and particular NFS arguments via the chunk Position value encoded in each Read chunk.

2.2. Use Of The Write List

The Write list in each RPC-over-RDMA transport header represents a set of memory regions that can receive DDP-eligible NFS result data. Large data items such as the payload of an NFS READ request are referenced by the Write list and placed directly into client memory.

Each Write chunk corresponds to a specific XDR data item in an NFS reply. This document specifies how NFS client and server implementations identify the correspondence between Write chunks and each XDR result.

2.3. Construction Of Individual Chunks

Each Read chunk is represented as a list of segments at the same XDR Position, and each Write chunk is represented as an array of segments. An NFS client thus has the flexibility to advertise a set of discontiguous memory regions in which to send or receive a single DDP-eligible data item.

2.4. Use Of Long Calls And Replies

Small RPC messages are conveyed using RDMA Send operations which are of limited size. If an NFS request is too large to be conveyed via an RDMA Send, and there are no DDP-eligible data items that can be removed, an NFS client must send the request using a Long Call. The entire NFS request is sent in a special Read chunk.

If a client expects that an NFS reply will be too large to be conveyed via an RDMA Send, it provides a Reply chunk in the RPC-over-RDMA transport header conveying the NFS request. The server can place the entire NFS reply in the Reply chunk.

These are described in more detail in [I-D.ietf-nfsv4-rfc5666bis].

3. NFS Versions 2 And 3 Upper Layer Binding

An NFS client MAY send a single Read chunk to supply opaque file data for an NFS WRITE procedure, or the pathname for an NFS SYMLINK procedure. For all other NFS procedures, the server MUST ignore Read chunks that have a non-zero value in their Position fields, and Read chunks beyond the first in the Read list.

Similarly, an NFS client MAY provide a single Write chunk to receive either opaque file data from an NFS READ procedure, or the pathname from an NFS READLINK procedure. The server MUST ignore the Write list for any other NFS procedure, and any Write chunks beyond the first in the Write list.

There are no NFS version 2 or 3 procedures that have DDP-eligible data items in both their Call and Reply. However, if an NFS client is sending a Long Call or Reply, it MAY provide a combination of Read list, Write list, and/or a Reply chunk in the same transaction.

NFS clients already successfully estimate the maximum reply size of each operation in order to provide an adequate set of buffers to receive each NFS reply. An NFS client provides a Reply chunk when the maximum possible reply size is larger than the client's responder inline threshold.

How does the server respond if the client has not provided enough Write list resources to handle an NFS WRITE or READLINK reply? How does the server respond if the client has not provided enough Reply chunk resources to handle an NFS reply?

4. NFS Version 4 Upper Layer Binding

This specification applies to NFS Version 4.0 [RFC7530], NFS Version 4.1 [RFC5661], and NFS Version 4.2 [I-D.ietf-nfsv4-minorversion2]. It also applies to the callback protocols associated with each of these minor versions.

An NFS client MAY send a Read chunk to supply opaque file data for a WRITE operation or the pathname for a CREATE(NF4LNK) operation in an NFS version 4 COMPOUND procedure. An NFS client MUST NOT send a Read chunk that corresponds with any other XDR data item in any other NFS version 4 operation.

Similarly, an NFS client MAY provide a Write chunk to receive either opaque file data from a READ operation, NFS4_CONTENT_DATA from a READ_PLUS operation, or the pathname from a READLINK operation in an NFS version 4 COMPOUND procedure. An NFS client MUST NOT provide a Write chunk that corresponds with any other XDR data item in any other NFS version 4 operation.

There is no prohibition against an NFS version 4 COMPOUND procedure constructed with both a READ and WRITE operation, say. Thus it is possible for NFS version 4 COMPOUND procedures to use both the Read list and Write list simultaneously. An NFS client MAY provide a Read list and a Write list in the same transaction if it is sending a Long Call or Reply.

Some remarks need to be made about how NFS version 4 clients estimate reply size, and how DDP-eligibility violations are reported.

4.1. NFS Version 4 COMPOUND Considerations

An NFS version 4 COMPOUND procedure supplies arguments for a sequence of operations, and returns results from that sequence. A client MAY construct an NFS version 4 COMPOUND procedure that uses more than one chunk in either the Read list or Write list. The NFS client provides XDR Position values in each Read chunk to disambiguate which chunk is associated with which XDR data item.

However NFS server and client implementations must agree in advance on how to pair Write chunks with returned result data items. The mechanism specified in [I-D.ietf-nfsv4-rfc5666bis]) is applied here:

- The first chunk in the Write list MUST be used by the first READ or READLINK operation in an NFS version 4 COMPOUND procedure. The next Write chunk is used by the next READ or READLINK, and so on.
- If there are more READ or READLINK operations than Write chunks, then any remaining operations MUST return their results inline.
- If an NFS client presents a Write chunk, then the corresponding READ or READLINK operation MUST return its data by placing data into that chunk.
- If the Write chunk has zero RDMA segments, or if the total size of the segments is zero, then the corresponding READ or READLINK operation MUST return its result inline.

The following example shows a Write list with three Write chunks, A, B, and C. The server consumes the provided Write chunks by writing the results of the designated operations in the compound request, READ and READLINK, back to each chunk.

Write list:

```
A --> B --> C
```

NFS version 4 COMPOUND request:

```
PUTFH LOOKUP READ PUTFH LOOKUP READLINK PUTFH LOOKUP READ
              |                 |                 |
              v                 v                 v
              A                 B                 C
```

If the client does not want to have the READLINK result returned directly, it provides a zero-length array of segment triplets for buffer B or sets the values in the segment triplet for buffer B to zeros to indicate that the READLINK result must be returned inline.

Unlike NFS versions 2 and 3, the maximum size of an NFS version 4 COMPOUND is not bounded. However, typical NFS version 4 clients rarely issue such problematic requests. In practice, NFS version 4 clients behave in much more predictable ways. Rsize and wsize apply to COMPOUND operations by capping the total amount of data payload allowed in each COMPOUND. An extension to NFS version 4 supporting a comprehensive exchange of upper-layer message size parameters is part of [RFC5661].

4.2. NFS Version 4 Callbacks

The NFS version 4 protocols support server-initiated callbacks to notify clients of events such as recalled delegations. There are no DDP-eligible data items in callback protocols associated with NFSv4.0, NFSv4.1, or NFSv4.2.

In NFS version 4.1 and 4.2, callback operations may appear on the same connection as one used for NFS version 4 client requests. To operate on RPC-over-RDMA transports, NFS version 4 clients and servers MUST use the mechanism described in [I-D.ietf-nfsv4-rpcrdma-bidirection].

5. IANA Considerations

NFS use of direct data placement introduces a need for an additional NFS port number assignment for networks that share traditional UDP and TCP port spaces with RDMA services. The iWARP [RFC5041] [RFC5040] protocol is such an example (InfiniBand is not).

NFS servers for versions 2 and 3 [RFC1094] [RFC1813] traditionally listen for clients on UDP and TCP port 2049, and additionally, they register these with the portmapper and/or rpcbind [RFC1833] service. However, [RFC7530] requires NFS servers for version 4 to listen on TCP port 2049, and they are not required to register.

An NFS version 2 or version 3 server supporting RPC-over-RDMA on such a network and registering itself with the RPC portmapper MAY choose an arbitrary port, or MAY use the alternative well-known port number for its RPC-over-RDMA service. The chosen port MAY be registered with the RPC portmapper under the netid assigned by the requirement in [I-D.ietf-nfsv4-rfc5666bis].

An NFS version 4 server supporting RPC-over-RDMA on such a network MUST use the alternative well-known port number for its RPC-over-RDMA service. Clients SHOULD connect to this well-known port without consulting the RPC portmapper (as for NFSv4/TCP).

The port number assigned to an NFS service over an RPC-over-RDMA transport is available from the IANA port registry [RFC3232].

6. Security Considerations

The RDMA transport for RPC [I-D.ietf-nfsv4-rfc5666bis] supports all RPC [RFC5531] security models, including RPCSEC_GSS [RFC2203] security and transport-level security. The choice of RDMA Read and RDMA Write to convey RPC argument and results does not affect this, since it only changes the method of data transfer. Specifically, the requirements of [I-D.ietf-nfsv4-rfc5666bis] ensure that this choice does not introduce new vulnerabilities.

Because this document defines only the binding of the NFS protocols atop [I-D.ietf-nfsv4-rfc5666bis], all relevant security considerations are therefore to be described at that layer.

7. Acknowledgments

The author gratefully acknowledges the work of Brent Callaghan and Tom Talpey on the original NFS Direct Data Placement specification [RFC5667]. The author also wishes to thank Bill Baker and Greg Marsden for their support of this work.

Dave Noveck provided excellent review, constructive suggestions, and consistent navigational guidance throughout the process of drafting this document.

Special thanks go to nfsv4 Working Group Chair Spencer Shepler and nfsv4 Working Group Secretary Thomas Haynes for their support.

8. References

8.1. Normative References

[I-D.ietf-nfsv4-minorversion2]
Haynes, T., "NFS Version 4 Minor Version 2", draft-ietf-nfsv4-minorversion2-41 (work in progress), January 2016.

[I-D.ietf-nfsv4-rfc5666bis]
Lever, C., Simpson, W., and T. Talpey, "Remote Direct Memory Access Transport for Remote Procedure Call, Version One", draft-ietf-nfsv4-rfc5666bis-07 (work in progress), May 2016.

[I-D.ietf-nfsv4-rpcrdma-bidirection]
Lever, C., "Bi-directional Remote Procedure Call On RPC-over-RDMA Transports", draft-ietf-nfsv4-rpcrdma-bidirection-05 (work in progress), June 2016.

[RFC1833] Srinivasan, R., "Binding Protocols for ONC RPC Version 2", RFC 1833, DOI 10.17487/RFC1833, August 1995, <http://www.rfc-editor.org/info/rfc1833>.

[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>.

[RFC2203] Eisler, M., Chiu, A., and L. Ling, "RPCSEC_GSS Protocol Specification", RFC 2203, DOI 10.17487/RFC2203, September 1997, <http://www.rfc-editor.org/info/rfc2203>.

[RFC5531] Thurlow, R., "RPC: Remote Procedure Call Protocol Specification Version 2", RFC 5531, DOI 10.17487/RFC5531, May 2009, <http://www.rfc-editor.org/info/rfc5531>.

[RFC5661] Shepler, S., Ed., Eisler, M., Ed., and D. Noveck, Ed., "Network File System (NFS) Version 4 Minor Version 1 Protocol", RFC 5661, DOI 10.17487/RFC5661, January 2010, <http://www.rfc-editor.org/info/rfc5661>.

[RFC7530] Haynes, T., Ed. and D. Noveck, Ed., "Network File System (NFS) Version 4 Protocol", RFC 7530, DOI 10.17487/RFC7530, March 2015, <http://www.rfc-editor.org/info/rfc7530>.

8.2. Informative References

[RFC1094] Nowicki, B., "NFS: Network File System Protocol specification", RFC 1094, DOI 10.17487/RFC1094, March 1989, <http://www.rfc-editor.org/info/rfc1094>.

[RFC1813] Callaghan, B., Pawlowski, B., and P. Staubach, "NFS Version 3 Protocol Specification", RFC 1813, DOI 10.17487/RFC1813, June 1995, <http://www.rfc-editor.org/info/rfc1813>.

[RFC3232] Reynolds, J., Ed., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, DOI 10.17487/RFC3232, January 2002, <http://www.rfc-editor.org/info/rfc3232>.

[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>.

[RFC5041] Shah, H., Pinkerton, J., Recio, R., and P. Culley, "Direct Data Placement over Reliable Transports", RFC 5041, DOI 10.17487/RFC5041, October 2007, <http://www.rfc-editor.org/info/rfc5041>.

[RFC5667] Talpey, T. and B. Callaghan, "Network File System (NFS) Direct Data Placement", RFC 5667, DOI 10.17487/RFC5667, January 2010, <http://www.rfc-editor.org/info/rfc5667>.

Author’s Address

Charles Lever (editor)
Oracle Corporation
1015 Granger Avenue
Ann Arbor, MI 48104
USA

Phone: +1 734 274 2396
Email: chuck.lever@oracle.com