Internet Engineering Task Force (IETF)
Request for Comments: 9561
Category: Standards Track
ISSN: 2070-1721

C. Hellwig, Ed.

C. Lever
Oracle
S. Faibish
Opendrives.com
D. Black
Dell Technologies

April 2024

Using the Parallel NFS (pNFS) SCSI Layout to Access Non-Volatile Memory Express (NVMe) Storage Devices

Abstract

This document specifies how to use the Parallel Network File System (pNFS) Small Computer System Interface (SCSI) Layout Type to access storage devices using the Non-Volatile Memory Express (NVMe) protocol family.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9561.

Copyright Notice

Copyright (c) 2024 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

NFSv4.1 [RFC8881] includes a pNFS feature that allows reads and writes to be performed by means other than directing read and write operations to the server. Through use of this feature, the server, in the role of metadata server, is responsible for managing file and directory metadata while separate means are provided to execute reads and writes.

These other means of performing file reads and writes are defined by individual mapping types, which often have their own specifications.

The pNFS Small Computer System Interface (SCSI) layout [RFC8154] is a layout type that allows NFS clients to directly perform I/O to block storage devices while bypassing the Metadata Server (MDS). It is specified by using concepts from the SCSI protocol family for the data path to the storage devices.

NVM Express (NVMe), or the Non-Volatile Memory Host Controller Interface Specification, is a set of specifications to talk to storage devices over a number of protocols such as PCI Express (PCIe), Fibre Channel (FC), TCP/IP, or Remote Direct Memory Access (RDMA) networking. NVMe is currently the predominantly used protocol to access PCIe Solid State Disks (SSDs), and it is increasingly being adopted for remote storage access to replace SCSI-based protocols such as iSCSI.

This document defines how NVMe Namespaces using the NVM Command Set [NVME-NVM] exported by NVMe Controllers implementing the NVMe Base specification [NVME-BASE] are to be used as storage devices using the SCSI Layout Type. The definition is independent of the underlying transport used by the NVMe Controller and thus supports Controllers implementing a wide variety of transports, including PCIe, RDMA, TCP, and FC.

This document does not amend the existing SCSI layout document. Rather, it defines how NVMe Namespaces can be used within the SCSI Layout by establishing a mapping of the SCSI constructs used in the SCSI layout document to corresponding NVMe constructs.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "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.

1.2. General Definitions

The following definitions are included to provide context for the reader.

Client: The "client" is the entity that accesses the NFS server's resources. The client may be an application that contains the logic to access the NFS server directly, or it may be part of the

operating system that provides remote file system services for a set of applications.

Metadata Server (MDS): The Metadata Server (MDS) is the entity responsible for coordinating client access to a set of file systems and is identified by a server owner.

1.3. Numerical Conventions

Numerical values defined in the SCSI specifications (e.g., [SPC5]) and the NVMe specifications (e.g., [NVME-BASE]) are represented using the same conventions as those specifications wherein a 'b' suffix denotes a binary (base 2) number (e.g., 110b = 6 decimal) and an 'h' suffix denotes a hexadecimal (base 16) number (e.g., 1ch = 28 decimal).

2. SCSI Layout Mapping to NVMe

The SCSI layout definition [RFC8154] references only a few SCSI-specific concepts directly. This document provides a mapping from these SCSI concepts to NVM Express concepts that are used when using the pNFS SCSI layout with NVMe namespaces.

2.1. Volume Identification

The pNFS SCSI layout uses the Device Identification Vital Product Data (VPD) page (page code 83h) from [SPC5] to identify the devices used by a layout. Implementations that use NVMe namespaces as storage devices map NVMe namespace identifiers to a subset of the identifiers that the Device Identification VPD page supports for SCSI logical units.

To be used as storage devices for the pNFS SCSI layout, NVMe namespaces MUST support either the IEEE Extended Unique Identifier (EUI64) or Namespace Globally Unique Identifier (NGUID) value reported in a Namespace Identification Descriptor, the I/O Command Set Independent Identify Namespace data structure, and the Identify Namespace data structure, NVM Command Set [NVME-BASE]. If available, use of the NGUID value is preferred as it is the larger identifier.

> Note: The PS_DESIGNATOR_T10 and PS_DESIGNATOR_NAME have no equivalent in NVMe and cannot be used to identify NVMe storage devices.

The pnfs_scsi_base_volume_info4 structure for an NVMe namespace SHALL be constructed as follows:

1. The "sbv_code_set" field SHALL be set to PS_CODE_SET_BINARY.

2. The "pnfs_scsi_designator_type" field SHALL be set to PS_DESIGNATOR_EUI64.

3. The "sbv_designator" field SHALL contain either the NGUID or the EUI64 identifier for the namespace. If both NGUID and EUI64 identifiers are available, then the NGUID identifier SHOULD be used as it is the larger identifier.

RFC 8154 [RFC8154] specifies the "sbv_designator" field as an XDR variable length opaque<> (refer to Section 4.10 of RFC 4506 [RFC4506]). The length of that XDR opaque<> value (part of its XDR representation) indicates which NVMe identifier is present. That length MUST be 16 octets for an NVMe NGUID identifier and MUST be 8 octets for an NVMe EUI64 identifier. All other lengths MUST NOT be used with an NVMe namespace.

2.2. Client Fencing

The SCSI layout uses Persistent Reservations (PRs) to provide client fencing. For this to be achieved, both the MDS and the Clients have to register a key with the storage device, and the MDS has to create a reservation on the storage device.

The following subsections provide a full mapping of the required PERSISTENT RESERVE IN and PERSISTENT RESERVE OUT SCSI commands [SPC5] to NVMe commands that MUST be used when using NVMe namespaces as storage devices for the pNFS SCSI layout.

2.2.1. PRs - Key Registration

On NVMe namespaces, reservation keys are registered using the Reservation Register command (refer to Section 7.3 of [NVME-BASE]) with the Reservation Register Action (RREGA) field set to 000b (i.e., Register Reservation Key) and supplying the reservation key in the New Reservation Key (NRKEY) field.

Reservation keys are unregistered using the Reservation Register command with the Reservation Register Action (RREGA) field set to 001b (i.e., Unregister Reservation Key) and supplying the reservation key in the Current Reservation Key (CRKEY) field.

One important difference between SCSI Persistent Reservations and NVMe Reservations is that NVMe reservation keys always apply to all controllers used by a host (as indicated by the NVMe Host Identifier). This behavior is analogous to setting the ALL_TG_PT bit when registering a SCSI Reservation Key, and it is always supported by NVMe Reservations, unlike the ALL_TG_PT for which SCSI support is inconsistent and cannot be relied upon. Registering a reservation key with a namespace creates an association between a host and a namespace. A host that is a registrant of a namespace may use any controller with which that host is associated (i.e., that has the same Host Identifier, refer to Section 5.27.1.25 of [NVME-BASE]) to access that namespace as a registrant.

2.2.2. PRs - MDS Registration and Reservation

Before returning a PNFS_SCSI_VOLUME_BASE volume to the client, the MDS needs to prepare the volume for fencing using PRs. This is done by registering the reservation generated for the MDS with the device (see Section 2.2.1) followed by a Reservation Acquire command (refer to Section 7.2 of [NVME-BASE]) with the Reservation Acquire Action (RACQA) field set to 000b (i.e., Acquire) and the Reservation Type (RTYPE) field set to 4h (i.e., Exclusive Access - Registrants Only Reservation).

2.2.3. Fencing Action

In case of a non-responding client, the MDS fences the client by executing a Reservation Acquire command (refer to Section 7.2 of [NVME-BASE]), with the Reservation Acquire Action (RACQA) field set to 001b (i.e., Preempt) or 010b (i.e., Preempt and Abort), the Current Reservation Key (CRKEY) field set to the server’s reservation key, the Preempt Reservation Key (PRKEY) field set to the reservation key associated with the non-responding client, and the Reservation Type (RTYPE) field set to 4h (i.e., Exclusive Access - Registrants Only Reservation). The client can distinguish I/O errors due to fencing from other errors based on the Reservation Conflict NVMe status code.

2.2.4. Client Recovery after a Fence Action

If an NVMe command issued by the client to the storage device returns a non-retryable error (refer to the DNR bit defined in Figure 92 in

[NVME-BASE]), the client MUST commit all layouts that use the storage device through the MDS, return all outstanding layouts for the device, forget the device ID, and unregister the reservation key.

2.3. Volatile Write Caches

For NVMe controllers, a volatile write cache is enabled if bit 0 of the Volatile Write Cache (VWC) field in the Identify Controller data structure, I/O Command Set Independent (refer to Figure 275 in [NVME-BASE]) is set and the Volatile Write Cache Enable (WCE) bit (i.e., bit 00) in the Volatile Write Cache Feature (Feature Identifier 06h) (refer to Section 5.27.1.4 of [NVME-BASE]) is set. If a volatile write cache is enabled on an NVMe namespace used as a storage device for the pNFS SCSI layout, the pNFS server (MDS) MUST use the NVMe Flush command to flush the volatile write cache to stable storage before the LAYOUTCOMMIT operation returns by using the Flush command (refer to Section 7.1 of [NVME-BASE]). The NVMe Flush command is the equivalent to the SCSI SYNCHRONIZE CACHE commands.

3. Security Considerations

NFSv4 clients access NFSv4 metadata servers using the NFSv4 protocol. The security considerations generally described in [RFC8881] apply to a client’s interactions with the metadata server. However, NFSv4 clients and servers access NVMe storage devices at a lower layer than NFSv4. NFSv4 and RPC security are not directly applicable to the I/Os to data servers using NVMe. Refer to Sections 2.4.6 (Extents Are Permissions) and 4 (Security Considerations) of [RFC8154] for the security considerations of direct access to block storage from NFS clients.

pNFS with an NVMe layout can be used with NVMe transports (e.g., NVMe over PCIe [NVME-PCIE]) that provide essentially no additional security functionality. Or, pNFS may be used with storage protocols such as NVMe over TCP [NVME-TCP] that can provide significant transport layer security.

It is the responsibility of those administering and deploying pNFS with an NVMe layout to ensure that appropriate protection is deployed to that protocol based on the deployment environment as well as the nature and sensitivity of the data and storage devices involved. When using IP-based storage protocols such as NVMe over TCP, data confidentiality and integrity SHOULD be provided for traffic between pNFS clients and NVMe storage devices by using a secure communication protocol such as Transport Layer Security (TLS) [RFC8446]. For NVMe over TCP, TLS SHOULD be used as described in [NVME-TCP] to protect traffic between pNFS clients and NVMe namespaces used as storage devices.

A secure communication protocol might not be needed for pNFS with NVMe layouts in environments where physical and/or logical security measures (e.g., air gaps, isolated VLANs) provide effective access control commensurate with the sensitivity and value of the storage devices and data involved (e.g., public website contents may be significantly less sensitive than a database containing personal identifying information, passwords, and other authentication credentials).

Physical security is a common means for protocols not based on IP. In environments where the security requirements for the storage protocol cannot be met, pNFS with an NVMe layout SHOULD NOT be deployed.

When security is available for the data server storage protocol, it is generally at a different granularity and with a different notion of identity than NFSv4 (e.g., NFSv4 controls user access to files,

and NVMe controls initiator access to volumes). As with pNFS with the block layout type [RFC5663], the pNFS client is responsible for enforcing appropriate correspondences between these security layers. In environments where the security requirements are such that client-side protection from access to storage outside of the layout is not sufficient, pNFS with a SCSI layout on a NVMe namespace SHOULD NOT be deployed.

As with other block-oriented pNFS layout types, the metadata server is able to fence off a client's access to the data on an NVMe namespace used as a storage device. If a metadata server revokes a layout, the client's access MUST be terminated at the storage devices via fencing as specified in Section 2.2. The client has a subsequent opportunity to acquire a new layout.

4. IANA Considerations

This document has no IANA actions.

5. References

5.1. Normative References

[NVME-BASE] NVM Express, Inc., "NVM Express Base Specification", Revision 2.0d, January 2024, <https://nvmexpress.org/wp-content/uploads/NVM-Express-Base-Specification-2.0d-2024.01.11-Ratified.pdf>.

[NVME-NVM] NVM Express, Inc., "NVM Express NVM Command Set Specification", Revision 1.0d, December 2023, <https://nvmexpress.org/wp-content/uploads/NVM-Express-NVM-Command-Set-Specification-1.0d-2023.12.28-Ratified.pdf>.

[NVME-TCP] NVM Express, Inc., "NVM Express TCP Transport Specification", Revision 1.0d, December 2023, <https://nvmexpress.org/wp-content/uploads/NVM-Express-TCP-Transport-Specification-1.0d-2023.12.27-Ratified.pdf>.

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

[RFC4506] Eisler, M., Ed., "XDR: External Data Representation Standard", STD 67, RFC 4506, DOI 10.17487/RFC4506, May 2006, <https://www.rfc-editor.org/info/rfc4506>.

[RFC5663] Black, D., Fridella, S., and J. Glasgow, "Parallel NFS (pNFS) Block/Volume Layout", RFC 5663, DOI 10.17487/RFC5663, January 2010, <https://www.rfc-editor.org/info/rfc5663>.

[RFC8154] Hellwig, C., "Parallel NFS (pNFS) Small Computer System Interface (SCSI) Layout", RFC 8154, DOI 10.17487/RFC8154, May 2017, <https://www.rfc-editor.org/info/rfc8154>.

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

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8881] Noveck, D., Ed. and C. Lever, "Network File System (NFS) Version 4 Minor Version 1 Protocol", RFC 8881, DOI 10.17487/RFC8881, August 2020, <https://www.rfc-editor.org/info/rfc8881>.

[SPC5] INCITS Technical Committee T10, "SCSI Primary Commands - 5 (SPC-5)", INCITS 502-2019, 2019.

5.2. Informative References

[NVME-PCIE] NVM Express, Inc., "NVMe over PCIe Transport Specification", Revision 1.0d, December 2023, <https://nvmexpress.org/wp-content/uploads/NVM-Express-PCIe-Transport-Specification-1.0d-2023.12.27-Ratified.pdf>.

Acknowledgements

Carsten Bormann converted an earlier RFCXML v2 source for this document to a markdown source format.

David Noveck provided ample feedback to various drafts of this document.

Authors' Addresses

Christoph Hellwig (editor)
Email: hch@lst.de

Charles Lever
Oracle Corporation
United States of America
Email: chuck.lever@oracle.com

Sorin Faibish
Opendrives.com
11 Selwyn Road
Newton, MA 02461
United States of America
Phone: +1 617-510-0422
Email: s.faibish@opendrives.com

David L. Black
Dell Technologies
176 South Street
Hopkinton, MA 01748
United States of America
Email: david.black@dell.com