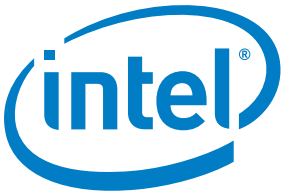

Configuring Non-Volatile Memory Express* (NVMe*) over Fabrics on Intel® Omni-Path Architecture

Application Note

October 2017

Document Number: J78967-1.0

Legal Disclaimer

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting: http://www.intel.com/design/literature.htm

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at http://www.intel.com/ or from the OEM or retailer.

No computer system can be absolutely secure.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Contents

1 Introduction 5

2 Overview 5

3 Installation and Configuration 6

3.1 Hardware and Operating System Requirements 6
3.2 Install the Intel® Omni-Path IFS Software Stack 6
3.3 Configure the Fabric 7
3.4 Upgrade the OS Kernel 8
3.5 Configure the Intel® Omni-Path Stack 9
3.6 Configure NVMe* over Fabrics Target and Host 9
3.6.1 Start the target system 9
3.6.2 Start the host system 10
3.6.3 Stop the host system 11
3.6.4 Stop the target system 11

Revision History

Date	Revision	Description
October 2017	1.0	Initial release of document.

1 Introduction

The NVM Express* (NVMe*) specification has been extended to provide NVMe* over Fabrics such as Ethernet*, InfiniBand*, and Intel® Omni-Path. NVMe* over Fabrics offers a way to export NVMe SSD performance outside of a single server. To make this happen, a high-speed interface between client and server is required, and a special protocol is used between them. The advantage of the NVMe* over Fabrics specification is that it does not rely on a particular hardware interface. It works with most RDMA-enabled fabrics such as Ethernet* (iWarp*, RoCe*), InfiniBand*, or Intel® Omni-Path fabric.

Using NVMe* over Fabrics, you can attach and detach NVMe SSDs to the clients (Hosts) based on the requirement of a given workload or application. You can also partition the SSD and slice it into pieces or aggregate multiple SSDs if the workload requires it. Alternatively, you can design multi-path configurations where a storage device (target) is attached to multiple hosts at the same time. This configuration flexibility has advantages for many uses in HPC architecture design.

This application note focuses on a simple Intel® Omni-Path implementation and describes how to implement a point-to-point configuration with one target and one host server. The concepts in this document can be expanded to more complex topologies, however, it is outside the scope of this document.

For an overview of NVMe over Fabrics, refer to:
http://www.nvmexpress.org/wp-content/uploads/NVMe_Over_Fabrics.pdf

2 Overview

This guide contains the following sections:

- Section 3.1, Hardware and Operating System Requirements
- Section 3.2, Install the Intel® Omni-Path IFS Software Stack
- Section 3.3, Configure the Fabric
- Section 3.4, Upgrade the OS Kernel
- Section 3.5, Configure the Intel® Omni-Path Stack
- Section 3.6, Configure NVMe* over Fabrics Target and Host

3 Installation and Configuration

This section describes how to install and configure NVMe* over Fabrics using CentOS* 7.3 (1611). Using this OS requires a kernel upgrade, which is also described. These steps can be applied to other distributions with minor changes in the installation process, however, the details are outside the scope of this application note.

Note: The following OS distributions include native support for NVMe* over Fabrics: RHEL* 7.4, CentOS* 7.4 (1708), and SLES* 12 SP3. If you install Intel® Omni-Path Fabric Software Release 10.6 (or later) on one of these distributions, you do not need to upgrade the kernel as described in Section 3.4.

3.1 Hardware and Operating System Requirements

The steps below assume that CentOS* 7.3 (1611) has been pre-installed.

1. Install Development Tools:
```
# yum groupinstall 'Development Tools'
# yum update
```
2. Install required software components for Intel® Omni-Path software stack.

   Be aware that the required components are dependent on the version of Intel® Omni-Path software that is installed and may be OS-dependent. For details, see the *Intel® Omni-Path Software Installation Guide, OS RPMs Installation Prerequisites* section.

   In this example:
```
# yum install libibmad libibverbs librdmacm libibcm qperf perftest
rdma infinipath-psm expat elfutils-libelf-devel libstdc++-devel
gcc-gfortran atlas tcl expect tcsh sysfsutils pciutils bc libibumad
libibumad-devel libibumad libibumad-devel libibverbs-devel libibmad-
devel librdmacm-devel ibacm-devel openssl-devel libuuid-devel expat-
devel infinipath-psm-devel valgrind-devel libgnome libibverbs*
opensm-libs libhfi1 papi ncurses-devel hwloc hwloc-gui
```
3. Identify hardware topology by running `lstopo` utilities:
```
# lstopo -v --of png > ./numa_layout.png
```

Intel recommends that the fabric interface and corresponding NVMe* SSDs are located on the same socket. This provides optimal performance and minimizes cross-socket traffic.

3.2 Install the Intel® Omni-Path IFS Software Stack

The following steps are a summary of the required steps. For detailed information, refer to the *Intel® Omni-Path Software Installation Guide*.

1. Download the latest IFS stack for the supported Linux* distribution. This example assumes CentOS* 7.3 (1611):

   https://downloadcenter.intel.com/search?keyword=omni-path+host+fabric+interface

2. Unpack and run the installer:

```
# tar -xzf IntelOPA-IFS.RHEL73-x86_64.x.x.x.x.x.tgz
# cd IntelOPA-IFS.RHEL73-x86_64.x.x.x.x.x/
# ./INSTALL

Intel OPA x.x.x.x.x Software

  1) Install/Uninstall Software
  2) Reconfigure OFA IP over IB
  3) Reconfigure Driver Autostart
  4) Generate Supporting Information for Problem Report
  5) FastFabric (Host/Chassis/Switch Setup/Admin)

  X) Exit
```

3. Press **1** to Install software.
4. Press **I** to select all and **P** to perform the action.
5. Once installed, return to the main menu and select **2** to configure IPOIB. Follow the default recommendations with the following changes:

   On the target:
   - IP over IB address: 192.168.2.1
   - Auto-start all services including OPAFM.

   On the host:
   - IP over IB address: 192.168.2.2
   - Auto-start all services but OPAFM.
6. Press **P** to apply the changes and reboot both servers.

3.3 Configure the Fabric

Configure the fabric for best performance and check RDMA functionality.

1. Verify the Intel® Omni-Path link is up and modules are loaded after system restart. Check the opainfo output to ensure the OPA port state is Active. Check the ifconfig output for the IPoIB link status.

```
# modprobe hfi1
# modprobe ib_ipoib
# opainfo
# ifconfig
```

2. Configure optimal performance by enabling turbo states:

   Note: In this particular configuration, the base CPU frequency is 2.3 GHz.

```
# echo 100 > /sys/devices/system/cpu/intel_pstate/min_perf_pct
# echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo
# cpupower -c all frequency-set --min 2301000
# cpupower frequency-set -g performance
```

3. Verify CPU frequency:

```
# cpupower frequency-info
```

   For other tuning options that may be required, refer to the *Intel® Omni-Path Fabric Performance Tuning User Guide*.

4. Execute RDMA benchmark. See example output on a test configuration:

   On the target:

```
ib_write_bw -F -R -s 1048576
```

   On the host:

```
ib_write_bw -F -R -s 1048576 192.168.2.1
---------------------------------------------------------------------------------------
#bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]
1048576    5000             11778.20            11705.75                 0.011706
---------------------------------------------------------------------------------------
```

3.4 Upgrade the OS Kernel

Note: The following OS distributions include native support for NVMe* over Fabrics: RHEL* 7.4, CentOS* 7.4 (1708), and SLES* 12 SP3. If you install Intel® Omni-Path Fabric Software Release 10.6 (or later) on one of these distributions, you do not need to upgrade the kernel as described here.

Upgrade the OS kernel to one with integrated NVMe* over Fabrics software stack. Initial commits are included into kernel 4.5, while the latest kernel typically includes the most recent updates.

1. Extract new kernel and run a configuration tool:

```
# tar -xvf ./linux-4.11.5.tar.xz
# cd ./linux-4.11.5
# make menuconfig
```

2. In the Device Drivers category, select the following: NVMe, NVMe over Fabrics RDMA, InfiniBand support - Intel Omni-Path. Verify the following kernel modules are enabled in the configuration and save it. Press “/” for search:
   - CONFIG_NVME_CORE
   - CONFIG_BLK_DEV_NVME
   - CONFIG_NVME_FABRICS
   - CONFIG_NVME_RDMA
   - CONFIG_NVME_TARGET
   - CONFIG_NVME_TARGET_LOOP
   - CONFIG_NVME_TARGET_RDMA
   - INFINIBAND
   - INFINIBAND_HFI1
   - INFINIBAND_RDMAVT
3. Compile a kernel:

```
# make -j24
# make modules_install
# make install
```

4. Select the default boot kernel, copy a kernel name first:

```
# grep ^menuentry /boot/grub2/grub.cfg | cut -d "'" -f2

CentOS Linux (4.11.5) 7 (Core)
CentOS Linux (3.10.0-514.21.1.el7.x86_64) 7 (Core)
CentOS Linux (3.10.0-514.el7.x86_64) 7 (Core)
CentOS Linux (0-rescue-3a3ab26661a34fde8f2baee5c8489111) 7 (Core)
```

5. Set the new kernel for the default boot configuration:
```
# grub2-set-default 'CentOS Linux (4.11.5) 7 (Core)'
```
6. Reboot both systems, make sure they start with the kernel you need:
```
# uname -r
```

3.5 Configure the Intel® Omni-Path Stack

Configure the Intel® Omni-Path stack for best NVMe* over Fabrics performance.

1. Apply the following parameters to allocate threads for IB Verbs support with Intel® Omni-Path:
```
# systemctl stop opafm
# rmmod hfi1
# echo "options hfi1 krcvqs=8 sge_copy_mode=2 wss_threshold=70" >
 /etc/modprobe.d/hfi1.conf
# dracut -f
```
2. Restart the system. Make sure the target system is also running opafm:
```
# systemctl status opafm
```
3. Check if parameters are applied:
```
# grep . /sys/module/hfi1/parameters/*
```
4. Repeat the RDMA test to make sure the system runs at the same or better performance:

   On the target:
```
ib_write_bw -F -R -s 1048576
```

   On the host:
```
ib_write_bw -F -R -s 1048576 192.168.2.1
---------------------------------------------------------------------------------------
#bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]
1048576    5000             11774.87            11608.62                  0.011609
---------------------------------------------------------------------------------------
```

3.6 Configure NVMe* over Fabrics Target and Host

3.6.1 Start the target system

This example assumes that /dev/nvme0n1 NVMe SSD is used for NVMe* over Fabrics. If you do not have a physical NVMe* SSD, use a `null_blk` module instead:

```
# export TARGET=192.168.2.1
# export DEV=/dev/nvme0n1
# export PORT=4420
# export NVME_PORT=1
# export SUBSYSTEM=testsubsys
# export NQN=${SUBSYSTEM}
# export NSID=1
# modprobe configfs
# modprobe nvme
# modprobe nvmet
# modprobe nvmet_rdma
```

```
# cd /sys/kernel/config/nvmet/subsystems/
# echo creating ${SUBSYSTEM} on device ${DEV}
# mkdir ${SUBSYSTEM}
# sleep 1
# mkdir ${SUBSYSTEM}/namespaces/${NSID}
# echo -n ${DEV} > ${SUBSYSTEM}/namespaces/${NSID}/device_path
# echo -n 1 > ${SUBSYSTEM}/attr_allow_any_host
# echo -n 1 > ${SUBSYSTEM}/namespaces/${NSID}/enable
# cd /sys/kernel/config/nvmet/ports
# mkdir ${NVME_PORT}
# echo -n ipv4 > ${NVME_PORT}/addr_adrfam
# echo -n rdma > ${NVME_PORT}/addr_trtype
# echo -n not required > ${NVME_PORT}/addr_treq
# echo -n ${TARGET} > ${NVME_PORT}/addr_traddr
# echo -n ${PORT} > ${NVME_PORT}/addr_trsvcid
# ln -s ../subsystems/${SUBSYSTEM} ${NVME_PORT}/subsystems/${SUBSYSTEM}
```

3.6.2 Start the host system

1. Load required kernel modules:

```
# modprobe nvme_rdma
# modprobe nvme_fabrics
# modprobe nvme_core
# modprobe rdma_cm
# modprobe ib_core
```

2. Install the NVMe-CLI utility to simplify the discovery and connection process. Download the utility from: https://github.com/linux-nvme/nvme-cli
   Install the utility using: `make && make install`

3. Use the target system IP address and NVMe* over Fabrics port assigned in the previous section:

```
# nvme discover -t rdma -a 192.168.2.1 -s 4420

Discovery Log Number of Records 1, Generation counter 1
=====Discovery Log Entry 0======
trtype:  rdma
adrfam:  ipv4
subtype: nvme subsystem
treq:    not required
portid:  1
trsvcid: 4420
subnqn:  testsubsys
traddr:  192.168.2.1
rdma_prtype: unrecognized
rdma_qptype: unrecognized
rdma_cms:    unrecognized
rdma_pkey: 0x0000
```

4. Connect to the target system and specify a number of queue pairs. Use the same number as the SSD supports, in this case 32:

```
# nvme connect -t rdma -n testsubsys -a 192.168.2.1 -s 4420 -i 32
```

5. Verify that the NVMe device is created on the host system:

```
# ls /dev/nvm*
/dev/nvme0  /dev/nvme0n1  /dev/nvme-fabrics
```

3.6.3 Stop the host system

Stop the NVMe* over Fabrics connection with the NVMe-CLI utility:

```
# nvme disconnect –d /dev/nvme0n1
```

3.6.4 Stop the target system

Delete the NVMe* over Fabrics configuration and unload modules:

```
# export TARGET=192.168.2.1
# export PORT=4420
# export NVME_PORT=1
# export SUBSYSTEM=testsubsys
# export NQN=${SUBSYSTEM}
# cd /sys/kernel/config/nvmet/ports
# rm ${NVME_PORT}/subsystems/*
# rmdir ${NVME_PORT}
# cd /sys/kernel/config/nvmet/subsystems/
# echo -n 0 > ${SUBSYSTEM}/namespaces/${NSID}/enable
# rmdir ${SUBSYSTEM}/namespaces/${NSID}
# rmdir ${SUBSYSTEM}
# modprobe -r nvmet_rdma
# modprobe -r nvmet
# modprobe -r nvme
```