OFA Interoperability Working Group

OFA-IWG Interoperability Test Plan Release 2.05-v2

March 22nd, 2017

Table of Contents

Revision History

Revision	Release Date	
0.50	Apr 4, 2006	• First FrameMaker Draft of the Interop Test Plan which was used in the March 2006 IBTA-OpenFabrics Plugfest.
0.51	Apr 25, 2006	• Added DAPL and updated MPI.
0.511	June 1, 2006	• Arkady Added iWARP.
0.52	May 30, 2006	• Added Intel MPI.
0.53	June 6, 2006	• Updated uDAPL section provided by Arkady.
0.54	June 13, 2006	• Updated entire Test Spec based on changes made by Arkady to incorporate iWARP into the Test Spec.
0.80	June 14, 2006	• Updated for the OFA conference in Paris and for BoD meeting. Added OFA logo and URL.
1.0	June 21, 2006	• Released after review and approval at the OFA conference in Paris.
1.01	Aug 17, 2006	• Updated the iWARP Equipment requirements in the General System Setup section.
1.02	Oct 31, 2006	• Updated Table 4 for iSER, Table 5 for SRP, Table 10 for uDAPL and corresponding info in Tables 17,18 and 22 as per request by Arkady. • Added new test section from Bob Jaworski for Fibre Channel Gateway.
1.03	Dec 10, 2006	• Updated test procedures based on the October 2006 OFA Interop Event. • Updated Fibre Channel Gateway test based on changes submitted by Karun Sharma (QLogic). • Added Ethernet Gateway test written by Karun Sharma (QLogic).
1.04	Mar 6, 2007	• Updated test procedures in preparation for the April 2007 OFA Interop Event

1.05	Mar 7, 2007	• Updated iWARP test procedures based on review by Mikkel Hagen of UNH-IOL. Added missing results tables.
1.06	April 3, 2007	• Updated for April 2007 Interop Event based on review from OFA IWG Meeting on 3/27/07.
1.07	April 3, 2007	• Updated for April 2007 Interop Event based on review from OFA IWG Meeting on 4/3/07
1.08	April 4, 2007	• Added list of Mandatory Tests for April 2007 Interop Event.
1.09	April 9, 2007	• Updated Intel MPI based on review by Arlin Davis.
1.10	April 10, 2007	• Updated after final review by Arlin Davis and after the OFA IWG meeting on 4/10/2007
1.11	Sep 7, 2007	• Updated with the latest scripts developed by UNH IOL and based on the results from the April 2007 Interop Event
1.12	Sep 12, 2007	• Updated the documents to embed the test scripts in the document.
1.13	Jan 22, 2008	• Updated the documents for the March 2008 OFA Interop event. IPoIB updated along with Cover Page and the Test Requirements section.
1.14	Feb 11, 2008	• Added the following tests: • Ethernet Switch Tests • IPoIB Connected Mode • RDMA Interop • 4. RDS
1.15	Feb 18, 2008	• Updates to the following tests: • Ethernet Switch Tests • IPoIB Connected Mode • 3. RDMA Interop
1.16	Feb 25, 2008	• Removed all reference to Low Latency Ethernet Switches. This is the version for the March 2008 Interop Event
1.17	March 3, 2008	• Added HP-MPI

1.18	July 22, 2008	• Updated HP-MPI based on results from the March 2008 Interop Event
1.19	July 28, 2008	• Updated HP-MPI URL for the tests. • Added section for Open MPI • Updated MPI based on feedback from UNH IOL
1.20	July 30, 2008	• Updated section for Open MPI and added tables • Updated IB SM Failover as per Nick Wood
1.21	Aug 1, 2008	• Updated SRP call srp_daemon -o -e -n • Updated IB SM Failover as Bob Jaworski • Updated HP-MPI • Updated Intel MPI • Updated Open MPI
1.22	Aug 29, 2008	• Added a section for MVAPICH 1 under OSU MPI
1.23	Feb 16, 2009	• Updated Link Init, Fabric Init, SRP, SDP, IPoIB CM, IPoIB DM based on updates received from UNH-IOL
1.24	Feb 23, 2009	• Updated Intel MPI and Open MPI to reflect the fact that they are not intended to work in a heterogeneous environment. • Updated the RDS test procedure • Updated the Test Glossary • Updated the Mandatory test table for April 2009
1.25	Feb 24, 2009	• Updated the RDS Test after review by the OFA IWG group.
1.26	Mar 13, 2009	• Restructured entire document to accommodate WinOF and OFED • Added NFS over RDMA to the test plan. • Added WinOF tests • Updated HP-MPI • Add List of Contributors
1.27	Mar 17, 2009	• Updates based on the review from the OFA IWG
1.28	Mar 27, 2009	• Added links in Chapter 10 to the InfiniBand Test Scripts • Added links to HP-MPI installation Packages

1.29	Aug 25, 2009	• Editorial & Technical updates based on April 2009 Interop Event. • Updated Mandatory tests for October 2009. • Added Topology Check • Added new Firmware Policy
1.30	Sep 4, 2009	• Updated Mandatory iWARP tests and several comments based on the review from Harry Cropper • Added changes suggested by Jess Robel from QLogic to IPoIB DM and CM and Fabric Init.
1.31	April 6, 2010	• Added definition of homogenous to Test Glossary • Added updates from the November 2009 Interop Event
1.32	April 20, 2010	• Updated after the OFA IWG meeting on 4/6/2010 • Updated MPI and MVAPICH based on changes requested by Jeff Laird and Intel
1.33	April 23, 2010	• Major changes to Section 8 which describes the Software and Firmware polices
1.34	July 20, 2010	• Changed uDAPL for iWARP to Beta for Aug 2010 GA Event • Removed HP MPI which is no longer supported • Added -mca mpi_leave_pinned 0 for OpenMPI • Add new parameters for MVAPICH2 for iWARP devices.
1.35	July 27, 2010	• Added new parameters for MVAPICH2 for iWARP devices. The parameter is: MV2_USE_RDMA_CM=1
1.36	Feb 22, 2011	• Added Link Init section as per changes provided by Chris Hutchins and approved by OFA IWG. • Updated Test Plan Status for April 2011 and October 2011 • Nick Wood from UNH-IOL updated NFSoRDMA • Marty requested that we update SRP Results Table 6 and remove the disconnect commands.
1.37	Oct 4, 2011	• Updated Test Plan Status for November 2011 • Added new Test Table for OS and OFED versions • Nick Wood updated Link Init for IB • Chris Hutchins updated RDMA Interop and RDMA Stress • Removed XANSation testing

1.38	Oct 11, 2011	• Changed Link Init Section from Recommendation to MOI • Updated Section 8 for Firmware, Software and Hardware Policies to bring in line with Logo Program Document • Updated InfiniBand Test Table 24
1.39	Oct 24, 2011	• Updated Open MPI as per changes submitted by Nick Wood • Updated RDMA Interop small test: drop iterations from 100000 to 25000 • Updated RDMA Interop large test, increase iterations from 100 to 300 • Updated IPoIB Part A:, drop iterations (number of pings) from 100 to 10.
1.41	Mar 20, 2012	• **General Instructions:** Added note that the OpenSM will be used to run all mandatory tests in the test plan and the Vendor SM testing will include testing IPoIB, RDMA Interop and Open MPI testing. • **General Instructions:** The OFILG decided as of April 2012 that the various ULPs contained in this test plan will only be tested if it is supported by the Operating System. • Logo Program Requirements: updated IB and iWARP. Made NFSoRDMA Mandatory and MVAPICH Optional. • **IPoIB:** Modified the way IPoIB is set to connected or datagram mode • **IPoIB:** Changed the ping interval in IPoIB tests from 0.01 to 0.2 • **IPoIB:** Reduced number of frame sizes tested in the Ping Test. • **MVAPICH:** Made testing of MVAPICH 1 & 2 Optional • **NFSoRDMA**: Eliminate the need to specify nfs-utils in the NFSoRDMA installation section • **NFSoRDMA:** Changed the way the servers are mounted in NFSoRDMA • **SDP:** Eliminated the need for vsftpd in SDP • **SDP:** Eliminated the environment variables section in SDP • **SDP:** Changed the way the netperf server is started in SDP • **SDP:** Made SDP mandatory only for those Operating Systems that support it. • **SRP**: Mandated that Targets only advertise two volumes in order to reduce the amount of time required to run the tests

1.42	Apr 3, 2012	• Updated Ethernet Test requirements to move NFSoRDMA to Beta for April 2012 • Changed the status of Intel MPI and OSU MVAPICH to deprecated meaning the tests are no longer being run or supported. • Updated SRP notes as per Marty Schlining
1.43	Aug 14, 2012	• Updated the definition for $NP in MVAPICH section 12.10.2, 2, ii • Updated Mandatory test tables for iWARP and IB • Cleared all change bars for October 2012 Interop event
1.44	Sep 18, 2012	• Removed Intel MPI because it is not Open Source • Removed SDP because no longer supported in OFED • Removed Ethernet Fabric Initialize, Failover and reconvergence. No longer applicable given DCB etc. • Removed TI RDS for iWARP because RDS does not support iWARP • Remove iWARP Connectivity - replaced by RDMA Interop test section • Added section 8 for OS Installation and OS Policy
1.45	Oct 9, 2012	• Add second test of SRP • Add RoCE test sections
1.46	Dec 17 2012	• Added note about NetApp Targets to SRP section • Added Ubuntu notes to section 9.2.2.1 • Added Ubuntu notes to section 11.2.2 Fabric Init • Added Ubuntu notes to section 11.6 SRP • Added Ubuntu notes to section 13.2.1 NFSoRDMA • Added Ubuntu notes to section 13.3.1 RDS • Added Ubuntu notes to section 13.5 RDMA Basic Interop • Added Ubuntu notes to section 13.5 RDMA Stress Test
1.47-v2	Mar 26,2013	• Updated the requirements for the OFILP for April 2013 • RoCE Updates suggested by IBM • Updated Section 9 regarding the OS and OFED • Added RoCE Stress Test to Table 16 and updated Section 13.6 RDMA Stress test • Added RSockets to the list of tests for future adoption • Added IPv6 to the list of tests for future adoption • Added Bonding for RoCE Ethernet interfaces to the list of tests for future adoption

1.47-v3	April 16, 2013	• Updated section 3.1 of the uDAPL test plan as per suggestion from Nate Landolt (UNH-IOL). Changed second RR to RW
1.48-v2	May 21, 2013	• Corrected typo in 13.5.12 - Nate Rubin • Added Open MPI Command line for RoCE as reported by Jeff Kopko from Emulex in 13.7.6 • Updated RDMA Basic Interop Tests for Emulex. Specified the max depth to be -t 127. See Sections 13.5.9, 13.5.11, 13.5.13 • Added section 13.7.2 on how to Install Open MPI for OFED 3.5 and later • Added section 13.7.3 on how to configure and build Open MPI 1.6.x for PowerLinux systems
1.48-v3	May 22, 2013	• Changed max depth to 126 for Emulex cards in section 13.5.xx • Added --mca btl openib,self to the PowerLinux command line in section 13.7.6 • Updated Topology Diagrams for IB, iWARP and RoCE
1.48-v4	May 23, 2013	• Updated the language in Sections 13.7.1-9 and 13.7.4-3 as per feedback from Brad Benton.
1.49-v1	Aug 11 2013	• Updated the mandatory requirements for IB, iWARP and ROCE.
1.49-v2	Aug 13 2013	• Updated the description of the Emulex ROCE hardware as per the request of Jeff Kopko.
1.49-v3	Sep 9, 2013	• Updated the NFSoRDMA Section as per the notes from Nate Rubin of UNH-IOL
1.49-v4	Sep 24, 2013	• Updated the RSockets Section as per the notes from Nate Rubin of UNH-IOL
1.49-v5	Oct 21, 2013	• Updated the RSockets Section to include the LD_PRELOAD command and also to change the -T option for the server to match with the Client -T command • Updated the RDMA Interop section to remove the deprecated rdma_bw commands from the iWARP section. As of OFED 3,5, iWARP must use the commands ib_write_bw -R -x 0, ib_send_bw -R -x 0 and ib_read_bw -R -x 0.

1.49-v6	Nov 5, 2013	• Updated the RDMA Interop section to remove the special Emulex option -t 126 which is no loner needed. • Updated the RDMA Interop section to include the "-s" option for bot the server and the client in all operations.
1.50-v1	Mar, 25 2014	• Updated document for April 2014 Interop Event • Added IPoCE to RoCE specific test pan
1.50-v2	May 6,2014	• Added the -n option to the server side of the RDMA Interop tests
1.51-v1	July 29, 2014	• Updated Test Status for October 2014 Events • Updated Open MPI test Section • Fixed a few cross references
1.51-v2	Aug 11, 2014	• Updated Topology diagrams for all transports
1.51-v3	Aug 18, 2014	• Updated Topology diagrams for iWARP and RoCE to eliminate arrows.
1.51-v4	Aug 25, 2014	• Updated Open MPI section after review by Dave Wyman from UNH-IOL
1.51-v5	Sep 23, 2014	• Updated Open MPI section with a few more small changes in PPC section and others. • Added a Test Results Key table to the end of the document
2.0	May 18, 2015	• Removed all WinOF references due to deprecation • Changed Editor to Dave Wyman • Updated Topology diagrams • Added/removed flags in RDMA IOP section • Converted to Word Docx, release 2.0
2.01	Jun 30, 2015	• Changed Editor to Keith M. Hoodlet • Added a “Table of Contents” section (hyperlinks and dynamic page numbering are planned for 2.02) • Major overhaul to document formatting as part of the migration from a previous documentation tool to Microsoft Word. • Command Line arguments are in the process of being clarified with the shell “$” pre-fix. • Minor edits to grammar have been performed in order to enhance readability.

2.01 (cont.)	Jun 30, 2015	• References to OSU-MPI and WinOF are in the process of being purged from the document (as these tests are no longer performed). • Clarifications of terms have been added (i.e. *Homogenous vs. Heterogeneous Clusters*)
2.02	July 7, 2015	• Added a “Table of Contents” with dynamic links to each respective section of the Test Plan document • Added section numbering to further distinguish portions of the Test Plan document.
2.03	July 16th, 2015	• Small corrections to iWARP “RDMA Basic Interop” commands added. • All references to WinOF have been purged from the document. • Topologies for the various technology configurations have been updated to reflect the July Logo event setup. • Connectathon section updated for NFSoRDMA. • Updated packet sizes in test procedures for IPoIB and IPoCE sections. • MPI implementation has been updated.
2.04	July 22nd, 2015	• Updated RoCE Topology to reflect current Logo testing.
2.05	June 16th, 2016	• Update syntax for certain commands throughout the document (updates are included in the Test Plan Structure) • Update IB SRP testing • Update IB iSER testing • Remove last modified date in footer because the header already contains the date. • Remove occurrences of arp with ip • Remove duplicate step in NFS over RDMA tests • Update grammar in certain places • TOC links should now export in PDFs properly • Change editor to Stefan Oesterreich & Jeremy Plsek
2.05-v2	March 22nd, 2017	• Remove RDMA Verify command from Table 15 • Fix section numbers for IPoCE. • Add Vinay Gupta to contributors.

List of Contributors

Name	Company
Mark Alan	HP
Brad Benton	IBM
Harry Cropper	Intel
Rupert Dance	Software Forge
Sujal Das	Mellanox
Arlin Davis	Intel
Johann George	QLogic
Mike Hagen	UNH-IOL
Mitko Haralanov	QLogic
Allen Hubbe	UNH-IOL
Christopher Hutchins	UNH-IOL
Bob Jaworski	QLogic
Arkady Kanevsky	NetApp
Llolsten Kaonga	Software Forge
Jeff Kopko	Emulex
Amit Krig	Mellanox
Jeff Laird	UNH-IOL
Jon Mason	Open Grid Computing

Edward Mossman	UNH-IOL
Bob Noseworthy	UNH-IOL
Yaroslav Pekelis	Mellanox
Jess Robel	Qlogic
Hal Rosenstock	HNR Consulting
Nate Rubin	UNH-IOL
Martin Schlining	DataDirect Networks
Karun Sharma	QLogic
Stan Smith	Intel
Dave Sommers	Intel (NetEffect)
Jeff Squyres	Cisco
Dennis Tolstenko	Lamprey Networks
Steve Wise	Open Grid Computing
Robert Woodruff	Intel
Nick Wood	UNH-IOL
Dave Wyman	UNH-IOL
Stefan Oesterreich	UNH-IOL
Keith M. Hoodlet	UNH-IOL
Jeremy Plsek	UNH-IOL
Vinay Gupta	Broadcom

Editor: Stefan Oesterreich & Jeremy Plsek

Legal Disclaimer

"This version of a proposed OpenFabrics Interop Test Plan is provided "AS IS" and without any warranty of any kind, including, without limitation, any express or implied warranty of non-infringement, merchantability or fitness for a particular purpose. In no event shall OpenFabrics, IBTA or any member of these groups be liable for any direct, indirect, special, exemplary, punitive, or consequential damages, including, without limitation, lost profits, even if advised of the possibility of such damages."

Conditional text tag *Explanation* is shown in green.
~~Conditional text tag *Deleted* is shown in red with strike through~~.
Conditional text tag *Proposal* is shown in turquoise (r0_g128_b128).
Conditional text tag *Author* is shown as is.
Conditional text tag Comment is shown in red with underline

1 Introduction

Server OEM customers have expressed the need for RDMA hardware and software to interoperate. Specifically, InfiniBand HCA and OpenFabric host software should interoperate with InfiniBand Switches, Gateways and Bridges loaded with management software provided by OEMs, as well as IB integrated server OEM vendors. Likewise, iWARP RNIC and OpenFabric host software should interoperate with Ethernet Switches and management software, as well as hardware, provided by Ethernet Switch OEMs and iWARP integrated server OEM vendors. As such, it is necessary that the interoperability test effort be industry-wide, where interoperability testing is conducted under the auspices of the appropriate networking organizations. For InfiniBand it is the IBTA, specifically within the charter of the CIWG. For iWARP it is the IETF.

1.1 Purpose

This document is intended to describe the production tests, going through each step and explaining each test along with the respective references. The purpose of this test plan is four fold:

1.) To define the scope, equipment and software needs - as well as test procedures - for verifying full interoperability of RDMA HW and SW. For Infiniband HW, it is InfiniBand HCAs using the latest OpenFabrics OFED software along with currently available OEM Switches and their management software. The target OEM IB Switch vendors are Intel and Mellanox. For iWARP HW, it is iWARP RNICs using the latest OpenFabrics OFED software with currently available OEM Ethernet Switches, Bridges, Gateways, Edge Devices etc. with their own respective management software.

2.) To serve as a basis for evaluating customer acceptance criteria for OFA host software interoperability and OFA Logo validation.

3.) To serve as a basis for the extension of InfiniBand IBTA CIWG test procedures related to interoperability and the use of these test procedures in upcoming PlugFest events as organized by the IBTA.

4.) To serve as a basis for the extension of iWARP test procedures for OpenFabrics software related to interoperability and the use of these test procedures in upcoming PlugFest events as organized by the UNH IOL OFILG testing service.

1.2 Intended Audience

- Project managers with OEM Switch, Router, Gateway, and Bridge Vendor companies who desire an understanding of the scope of testing, and look to participate in the extension of this test plan and procedures such that they meet their companies' respective requirements.
- IBTA, CIWG, iWARP and UNH IOL iWARP testing personnel and companies that seek to evaluate the scope of testing and to participate in the extension of this test plan and procedures as necessary to meet their requirements.
- Test engineering and project leads and managers who will conduct the testing based on this document.
- Customers and users of OFA host software that rely upon OFA Logo as an indication of interoperability.
- Integrators and OEM of RDMA products.

1.3 Test Plan Structure

This test plan is divided into two main sections:

1.) Interoperability testing using OFED for Linux

2.) An overview of the tests

Most tests contain example commands to be run through a command line interface. The syntax of the provided commands are intended for Bash. If using a different shell, syntax changes may be required.

Commands prefaced with a “$” will generally mean that the commands will not require super user privileges. Commands prefaced with a “#” will mean that super user privileges are required. This can be obtained by prepending “sudo” to the command, or by logging in to root before running the command with “su”.

1.4 InfiniBand Only - Test Overview

The tables below list all of the specific test procedures performed on InfiniBand Devices. Please see the *Transport Independent* section for tests that apply to all transports.

Table 1 – IB Link Initialize

Test #	Test	Description
1	Phy link up all ports	Check that all relevant LEDs are on for all HCAs and Switches.

Table 2 – IB Fabric Initialization

Test #	Test	Description
1	Fabric Initialization	Run SM from each node in the cluster and see that all ports are in an Armed or Active state.

Table 3 – IB IPoIB - Connect Mode (CM)

Test #	Test	Description
1	Ping all to all	Run SM from one of the nodes and check that all nodes responding. Repeat this step with all other SMs.
2	Connect disconnect host	Run SM from one of the nodes and check that all nodes responding.
3	FTP Procedure	Using a 4MB test file, “put” the file, then “get” the file, and finally - compare the file.

Table 4 – IB IPoIB - Datagram Mode (DM)

Test #	Test	Description
1	Ping all to all	Run SM from one of the nodes and check that all nodes are responding. Repeat this step with all other SMs.
2	Connect disconnect host	Run SM from one of the nodes and check that all nodes are responding.
3	FTP Procedure	Using a 4MB test file, “put” the file, then “get” the file, and finally - compare the file.

Table 5 – IB SM Tests

Test #	Test	Description
1	Basic sweep test	Verify that all SMs are NOT ACTIVE (after receiving the SMSet of SMInfo to DISABLE) and that the selected SM (SM1) is the master.
2	SM Priority test	Verify Subnet and SM's behavior according to the SM's priority.
3	Failover - Disable SM1	Disable the master SM and verify that the standby SM becomes the master and configures the cluster accordingly.
4	Failover - Disable SM2	Disable the master SM and verify that the standby SM becomes the master and configures the cluster accordingly.

Table 6 – IB SRP Tests

Test #	Test	Description
1	Basic dd application	Run a basic dd application from the SRP host connected to target.
2	IB SM kill	Kill the IB master SM while test is running and check that it completes properly.
3	Disconnect Host	Unload the SRP Host and check that the SRP connection properly disconnected.
4	Disconnect Target	Unload the SRP Target and check that the SRP connection properly disconnected.

Table 7 – IB Ethernet Gateway

Test #	Test	Description
1	Basic Setup	Connect the HCA of the IB host and Ethernet Gateway to the IB fabric. Connect the Ethernet gateway to the Ethernet network or Ethernet device. Start the SM to be used in this test.
2	Start ULP	Determine which ULP your ethernet gateway uses and be sure that ULP is running on the host.
3	Discover Gateway	Restart the ULP, or using the tool provided by the ULP – make sure that the host “discovers” the Ethernet Gateway.
4	SM Failover	While the ping is running, kill the master SM. Verify that the ping data transfer is unaffected.
5	Ethernet gateway reboot	Reboot the Ethernet Gateway. After the Ethernet Gateway comes up, verify that the host can discover the Ethernet Gateway (as it did before), and that we are able to configure the interfaces accordingly.
6	ULP restart	Restart the ULP used by the Ethernet Gateway and verify that after the ULP comes up, also verify that the host can discover the Ethernet Gateway and that we are able to configure the interfaces.

7	Unload/load ULP	Unload the ULP used by the Ethernet Gateway and check that the Ethernet Gateway shows it disconnected. Load the ULP and verify that the Ethernet gateway shows the connection.

Table 8 – IB Fibre Channel Gateway

Test #	Test	Description
1	Basic Setup	Connect the HCA of the IB host to the IB fabric. Connect the FC Gateway to the IB Fabric. Connect the FC Gateway to the FC network (or FC device). Start the SM to be used in this test.
2	Configure Gateway	Configure the FC Gateway appropriately (this is vendor specific).
3	Add Storage Device	Use the “ibsrpdm” tool in order to have the host "see" the FC storage device. Add the storage device as a target.
4	Basic dd application	Run a basic “dd” application from the SRP host connected to the target.
5	IB SM kill	Kill the IB master SM while the test is running, and check that it completes properly.
6	Disconnect Host/Target	Unload the SRP host / SRP Target (target first/host first) and check that the SRP connection is properly disconnected.
7	Load Host/Target	Load the SRP host / SRP Target. Using “ibsrpdm”, add the target.
8	dd after SRP Host and Target reloaded	Run a basic “dd” application from the SRP host to the FC storage device.
9	Reboot Gateway	Reboot the FC Gateway. After the FC Gateway comes up, verify using the “ibsrpdm” tool that the host can “see” the FC storage device. Add the storage device as a target.
10	dd after FC Gateway reboot	Verify that a basic “dd” works after rebooting the Gateway.

1.5 Ethernet Only - Test Overview

The tables below list all of the specific test procedures for iWARP and Ethernet Devices. Please see the *Transport Independent* section for tests that apply to all transports.

Table 9 – iWARP Link Initialize

Test #	Test	Description
1	Phy link up all ports	Check that all relevant green LEDs are on for all RN ICs and switches.
2	Verify basic IP connectivity	Verify IP and RDMA connectivity can occur by driving a minimally sized ICMP echo requests and replies across the link or equivalent traffic.

Table 10 – RoCE Link Initialize

Test #	Test	Description
1	Phy link up all ports	Check that all relevant green LEDs are on for all RCAs and switches.
2	Verify basic IP connectivity	Verify that IP and RDMA connectivity can occur by driving minimumally sized ICMP echo requests and replies across the link (or equivalent traffic).

1.6 Transport Independent - Test Overview

The tables below list the test procedures that apply to devices regardless of the transport.

Table 11 – TI iSER

Test #	Test	Description
1	Basic dd application	Run a basic “dd” application from the iSER host connected to the target.
2	IB SM kill	[IB Specific] Kill the IB master SM while the test is running and check that it completes properly.
3	Disconnect Initiator	Unload the iSER Host and check that the iSER connection properly disconnected.
4	Disconnect Target	Unload the iSER Target and check that the iSER connection properly disconnected.
5	Repeat with previous SM Slave	[IB Specific Test] Repeat steps 1-4 with the previous slave SM (we did not actually stop the target).

Table 12 – TI NFS over RDMA

Test #	Test	Description
1	File and directory creation	A total of six files and six directories are created
2	File and directory removal	Removes the directory tree that was just created by test1
3	Lookups across mount point	Changes directory to the test directory and gets the file status of the working directory
4	Setattr, getattr, and lookup	Permissions are changed (chmod) and the file status is retrieved (stat) for each file

5	Read and write	Creates a file (creat), Gets status of file (fstat) , Checks size of file, Writes 1048576 bytes into the file (write) in 8192 byte buffers, Closes file (close), Gets status of file (stat) , Checks the size of the file
6	Readdir	The program creates 200 files (creat). The current directory is opened (opendir), the beginning is found (rewinddir), and the directory is read (readdir) in a loop until the end is found
7	Link and rename	This program creates ten files. For each of these files, the file is renamed (rename) and file statistics are retrieved (stat) for both the new and old names
8	Symlink and readlink	This program makes 10 symlinks (symlink). It reads (readlink), and gets statistics for (lstat) each, and then removes them (unlink).
9	Statfs	This program changes directory to the test directory (chdir and/or mkdir) and gets the file system status on the current directory (statfs).

Table 13 – TI RDS

Test #	Test	Description
1	rds-ping procedure	Run “rds-ping” and verify that all hosts in the cluster can be reached
2	rds-stress procedure	Set up passive receiving instance as well as an active sender, and verify that data is exchanged without error

Table 14 – TI uDAPL

Test #	Test	Description
1	Point-to-Point Topology	Connection and simple send receive
2	Point-to-Point Topology	Verification, polling and scatter gather list

3	Switched Topology	Verification and private data
4	Switched Topology	Add multiple endpoints, polling, and scatter gather list
5	Switched Topology	Add RDMA Write
6	Switched Topology	Add RDMA Read
7	Multiple Switches	Multiple threads, RDMA Read, and RDMA Write
8	Multiple Switches	Pipeline test with RDMA Write and scatter gather list
9	Multiple Switches	Pipeline with RDMA Read

Table 15 – RDMA Basic InterOp

Test #	Test	Description
1	Small RDMA READ	Create an RDMA command sequence to send a READ operation of one byte.
2	Large RDMA READ	Create an RDMA command sequence to send a READ operation of 10,000,000 bytes
3	Small RDMA Write	Create an RDMA command sequence to send a Write operation of one byte
4	Large RDMA Write	Create an RDMA command sequence to send a Write operation of 10,000,000 bytes
5	Small RDMA SEND	Create an RDMA command sequence to send a SEND operation of one byte.
6	Large RDMA SEND	Create an RDMA command sequence to send a SEND operation of one million bytes

Table 16 – RDMA Stress Tests

Test #	Test	Description
1	Switch Load	For one pair of endpoints generate a stream of RDMA READ operations in one direction, and RDMA write operations in the opposite direction. For all remaining endpoint pairs configure an RDMA WRITE operation of 1 byte and have it sent 10,000 times on both streams of the endpoint pair.
2	Switch Fan In	Connect all possible endpoint pairs such that data exchanges between pairs must traverse the pair of ports interconnecting the switch
3	RoCE Stress Test	Stress the RoCE adapter by simultaneously transmitting both RoCE/IB traffic and IP level Ethernet traffic

Table 17 – RDMA Stress Tests

Test #	Test	Description
1	Socket calls	For each client, run socket tests for all size transfers (rstream -s <server-ip-address> -T s -S all)
2	Asynchronous calls	For each client run asynchronous tests for all size transfers (rstream -s <server-ip-address> -T a -S all)
3	Blocking calls	For each client run blocking tests for all size transfers (rstream -s <server-ip-address> -T b -S all)
4	Non-blocking calls	For each client run blocking tests for all size transfers (rstream -s <server-ip-address> -T n -S all)
5	Verified transfers	For each client run blocking tests for all size transfers (rstream -s <server-ip-address> -T v -S all)

1.7 Open MPI - Test Overview

Table 18 – TI – Open MPI Test Suite Description

Test #	Open MPI TESTs	Open MPI TESTs Suite Description
Phase 1: "Short" tests		
1	2	OMPI built with OpenFabrics support
2	3	OMPI basic functionality (hostname)
3	4.1	Simple MPI functionality (hello_c)
4	4.2	Simple MPI functionality (ring_c)
5	5	Point-to-point benchmark (NetPIPE)
6	6.1.1	Point-to-point benchmark (IMB PingPong multi)
7	6.1.2	Point-to-point benchmark (IMB PingPing multi)
Phase 2: "Long" tests		
8	6.2.1	Point-to-point benchmark (IMB PingPong)
9	6.2.2	Point-to-point benchmark (IMB PingPing)
10	6.2.3	Point-to-point benchmark (IMB Sendrecv)
11	6.2.4	Point-to-point benchmark (IMB Exchange)
12	6.2.5	Collective benchmark (IMB Bcast)
13	6.2.6	Collective benchmark (IMB Allgather)
14	6.2.7	Collective benchmark (IMB Allgatherv)
15	6.2.8	Collective benchmark (IMB Alltoall)

16	6.2.9	Collective benchmark (IMB Reduce)
17	6.2.10	Collective benchmark (IMB Reduce_scatter)
18	6.2.11	Collective benchmark (IMB Allreduce)
19	6.2.12	Collective benchmark (IMB Barrier)
20	6.3.1	I/O benchmark (IMB S_Write_Indv)
21	6.3.2	I/O benchmark (IMB S_IWrite_Indv)
22	6.3.3	I/O benchmark (IMB S_Write_Expl)
23	6.3.4	I/O benchmark (IMB S_IWrite_Expl)
24	6.3.5	I/O benchmark (IMB P_Write_Indv)
25	6.3.6	I/O benchmark (IMB P_IWrite_Indv)
26	6.3.7	I/O benchmark (IMB P_Write_Shared)
27	6.3.8	I/O benchmark (IMB P_IWrite_Shared)
28	6.3.9	I/O benchmark (IMB P_Write_Priv)
29	6.3.10	I/O benchmark (IMB P_IWrite_Priv)
30	6.3.11	I/O benchmark (IMB P_Write_Expl)
31	6.3.12	I/O benchmark (IMB P_IWrite_Expl)
32	6.3.13	I/O benchmark (IMB C_Write_Indv)
33	6.3.14	I/O benchmark (IMB C_IWrite_Indv)
34	6.3.15	I/O benchmark (IMB C_Write_Shared)
35	6.3.16	I/O benchmark (IMB C_IWrite_Shared)
36	6.3.17	I/O benchmark (IMB C_Write_Expl)

37	6.3.18	I/O benchmark (IMB C_IWrite_Expl)
38	6.3.19	I/O benchmark (IMB S_Read_Indv)
39	6.3.20	I/O benchmark (IMB S_IRead_Indv)
40	6.3.21	I/O benchmark (IMB S_Read_Expl)
41	6.3.22	I/O benchmark (IMB S_IRead_Expl)
42	6.3.23	I/O benchmark (IMB P_Read_Indv)
43	6.3.24	I/O benchmark (IMB P_IRead_Indv)
44	6.3.25	I/O benchmark (IMB P_Read_Shared)
45	6.3.26	I/O benchmark (IMB P_IRead_Shared)
46	6.3.27	I/O benchmark (IMB P_Read_Priv)
47	6.3.28	I/O benchmark (IMB P_IRead_Priv)
48	6.3.29	I/O benchmark (IMB P_Read_Expl)
49	6.3.30	I/O benchmark (IMB P_IRead_Expl)
50	6.3.31	I/O benchmark (IMB C_Read_Indv)
51	6.3.32	I/O benchmark (IMB C_IRead_Indv)
52	6.3.33	I/O benchmark (IMB C_Read_Shared)
53	6.3.34	I/O benchmark (IMB C_IRead_Shared)
54	6.3.35	I/O benchmark (IMB C_Read_Expl)
55	6.3.36	I/O benchmark (IMB C_IRead_Expl)
56	6.3.37	I/O benchmark (IMB Open_Close)

1.8 Requirements for OFA Interoperability Logo Program

The following table indicates the mandatory tests which will be used for Interop Validation during the Interop Debug Event and the Interop GA Event using OFED 3.18. (Please note: “deprecated” means that the test is no longer being actively run during the OFA Interop Events.)

InfiniBand Transport Test Status for OFED 3.18

Test Procedure	Linux
IB Link Initialize	**Mandatory**
IB Fabric Initialization	**Mandatory**
IB IPoIB Connected Mode	**Mandatory**
IB IPoIB Datagram Mode	**Mandatory**
IB SM Failover/Handover - OpenSM	**Mandatory**
IB SM Failover/Handover - Vendor SM	**Optional**
IB SRP	**Mandatory**
IB Ethernet Gateway	**Beta**
IB Fibre Channel Gateway	**Beta**
TI iSER	**Beta**
TI NFS over RDMA	**Mandatory**
TI RDS	**Optional**
TI RSockets	**Mandatory**
TI uDAPL	**Mandatory**

TI Basic RDMA Interop	**Mandatory**
TI RDMA Stress	**Mandatory**
TI MPI Open MPI	**Mandatory**

(Please note: **Optional** means that this test will not be made mandatory because it depends on proprietary vendor capabilities. The test may be run during the OFA Interop Events and reported in the results but it will not affect eligibility for the OFA Logo List.)

iWARP Transport Test Status for OFED 3.18

Test Procedure	**Linux**
iWARP Link Initialize	**Mandatory**
TI iSER	**Beta**
TI NFS over RDMA	**Beta**
TI uDAPL	**Mandatory**
TI Basic RDMA Interop	**Mandatory**
TI RDMA Stress	**Mandatory**
TI MPI Open MPI	**Mandatory**

RoCE Transport Test Status for OFED 3.18

Test Procedure	*Linux*
RoCE Link Initialize	**Mandatory**
RoCE Fabric Init	**TBD**
RoCE IPoCE	**Mandatory**
RoCE InfiniBand Gateway	**TBD**
RoCE Fibre Channel Gateway	**TBD**
TI RSockets	**Mandatory**
TI iSER	**Beta**
TI NFS over RDMA	**Beta**
TI uDAPL	**Mandatory**
TI Basic RDMA Interop	**Mandatory**
TI RDMA Stress	**Beta**
TI MPI Open MPI (Homogeneous only because of x86 and Power PC)	**Mandatory**

Subjects not covered

Number	Subject/ Feature	Description	Due Date
1	iWARP peer to peer	Future Testing	TBD
2	IPv6 testing	Future Testing	TBD
3	RDMA_CM Tests	IBM wants to develop tests for processor-heterogeneous (x86_64/ppc64) setups.	TBD
4	Bonding over RoCE	IBM wants to make sure Link Aggregation works. If they have two devices, they would like to test fail over	TBD

Test Glossary

Technical Terms	
DCB	Data Center Bridging (used in RoCE)
HCA	IB Host Channel Adapter
IPoIB	IP over InfiniBand
iSER	iSCSI Extensions for RDMA
MPI	Message Passing Interface
RCA	RoCE Channel Adapter
RDF	Readme File
RDS	Reliable Datagram Sockets
RNIC	RDMA NIC (iWARP Network Interface Card)
RoCE	RDMA over Converged Ethernet

SA	IB Subnet Administration
SDN	Software Defined Network
SDP	Sockets Direct Protocol
SM	IB Subnet Manager
SPB	Shortest Path Bridging (used in RoCE)
SRP	SCSI RDMA Protocol
TD	Test Descriptions
TI	Transport Independent (tests)
TRILL	Transparent Interconnect of Lots of Links is a IETF Standard implemented by devices called RBridges (Routing Bridges) or TRILL Switches (used in RoCE)
uDAPL	User Direct Access Programming Library

1.9 Homogenous vs. Heterogeneous Clusters

Heterogeneous & homogeneous clusters are the same with one exception: *the end points must be from the **same vendor** in homogeneous clusters*. The table below defines the guidelines for building homogeneous and heterogeneous clusters

Description	*Homogenous*	*Heterogeneous*
Mixing switches (both models and vendor products)	**Encouraged**	**Encouraged**
The use of any InfiniBand subnet manager	**Encouraged**	**Encouraged**
All devices of the same model number shall use the same firmware.	**Mandatory**	**Mandatory**
Any mix of products from the same vendor is acceptable - e.g. different model HCAs	**Encouraged**	**Encouraged**
A mix of end points (HCA/RNIC) from different OFA vendors	**Prohibited**	**Mandatory**
Mixing x86-32 (ix86) and x86_64 Operating System - see notes	**Not-Tested**	**Not-Tested**
32 bit architecture and 32 bit OS - see notes	**Not-Tested**	**Not-Tested**
Mixing x86-32 and x86-64 user-level application	**Optional**	**Optional**
Mixed system architecture - x86 servers mixed with IA-64 (Itanium) servers	**Not-Tested**	**Not-Tested**
Mixed system architecture - x86_64 and ppc64 interoperability - this is only tested with IBM RoCE Adapters	**Optional**	**Optional**
Mixing endianness in system OS - this is only tested using ppc64 and IBM RoCE Adapters	**Optional**	**Optional**
Mixing the quantity of server RAM installed on the hosts	**Encouraged**	**Encouraged**

Mixing the server clock speeds	**Encouraged**	**Encouraged**
Mixing the number of server cores	**Encouraged**	**Encouraged**
Mixing PCIe generations	**Encouraged**	**Encouraged**
All servers shall run the same OFED version.	**Encouraged**	**Encouraged**
Mixing supported Operating Systems	**Encouraged**	**Encouraged**

(Please note: Intel drivers do not support 32 bit operating systems)

2 Use of OpenFabrics Software for Pre-Testing

Depending on the schedule of testing and bugs or issues encountered, different snapshots of latest OpenFabrics software will be used during pre-testing prior to the Interoperability Event. Any changes that result in the OpenFabrics software from interoperability testing per this test plan will be deposited back into the OpenFabrics repository so that the OpenFabrics development community will have full access to any bug fixes or feature additions that may result out of this testing effort. The frequency of such deposits will be determined based on completion of adequate testing of the said fixes or feature additions.

3 Use of OpenFabrics Software for IBTA/CIWG Compliance Plugfests

During the pre-testing phase, UNH-IOL will apply all reasonable effort to ensure that the OpenFabrics source and binary repositories are up-to-date with the latest OFED release. This will enable cable interoperability testing at plugfests to be conducted using software directly sourced from the OpenFabrics tree.Should there be any issues with the OpenFabrics community not accepting certain bug fixes or features with the time frames matching with Compliance Events, UNH-IOL will inform all participants about the same and offer those bug fixes or features in source code and binary formats directly to the participants and InfiniBand solution suppliers.

4 Use of OpenFabrics Software for OFA IWG Interoperability Events

During the pre-testing phase, UNH-IOL will apply all reasonable effort to ensure that the OpenFabrics source and binary repositories are up-to-date with the latest OFED releases chosen by the OFA IWG for use in the Interoperability Event. Should there be any issues with the OpenFabrics community not accepting certain bug fixes or features with the timeframes matching with Interoperability Events, UNH-IOL will inform all participants about the same and offer those bug fixes or features in source code and binary formats directly to the participants and InfiniBand solution suppliers.

5 General System Setup

Configuration

The test environment for the user interface contains:

5.1 IB HW Units

IB Equipment

Equipment	**Amount**	**Details**	**Check**
Servers with OS installed	12 or more	The OS should support OpenFabrics Software.	
4X IB Cables	30 or more	Between 1 meter => 10 meters.	
IB Switches	4	The number and types of switches needed from member companies or OEMs is dependent on variations in subnet management and other IBTA defined management software. For example if the software on Switch A is different from the software used in Switch B, both Switches will be needed. Note that it is not dependent on number of ports supported by a switch.	
IB HCAs	12 or more		

5.2 IB Software

Linux platforms
OFED - Most Current Tested Release
IB HCA FW – Version XXX - Vendor Specific
IB Switch FW candidate – Version XXX - Vendor Specific
IB Switch SW – Version XXX - Vendor Specific

5.3 iWARP HW Units

iWARP Equipment

Equipment	**Amount**	**Details**	**Check**
Servers with OS installed	5 or more	The OS should support OpenFabrics Software.	
4X CX4 or SFP Cables	10 or more	Between 1 meter => 10 meters.	
10 GbE Switches	1 or more	At least one 10 GbE switch must be made available to support the various RNICs in the Fabric. There is no need to have multiple switches if there are enough ports on the primary switches to support all the devices in the fabric.	
iWARP RNIC	5 or more	Each vendor must supply 5 or more RNICs in order to support MPI testing.	

5.4 iWARP Software

Linux platforms
OFED - Most Current Tested Release
iWARP RNIC FW – Version XXX - Vendor Specific
10GbE Switch FW candidate – Version XXX - Vendor Specific
10GbE Switch SW – Version XXX - Vendor Specific

Vendor Specific Notes

Currently there is no interoperability between cxgb4 and nes if peer2peer is enabled. Both nes and cxgb4 have their own proprietary ways of doing "client must send the first fpdu". The Chelsio parameter file /sys/module/iw_cxgb4/parameters/peer2peer should be modified on all hosts to contain the appropriate value for each test. For example: the value must be set to ’1’ for the uDAPL test.

Arlin Davis suggests the following given the current situation:

1.) The dapltest -T P (performance tests) will always send data from server side first. This test will NOT work reliably with iWARP vendors.
2.) The dapltest -T T (transaction tests) should work fine with both IB and iWARP vendors given that it always sends from client side first.
3.) I recommend using only dapltest transaction mode (-T T) in your test plan and removing -T P mode tests.

5.5 RoCE HW Units

RoCE Equipment

Equipment	**Amount**	**Details**	**Check**
Servers with OS installed	5 or more	The OS should support OpenFabrics Software.	
4X QSFP+ Cables	10 or more	Between 1 meter => 10 meters.	
GbE DCB Switches	1 or more	At least one 10 or 40 GbE DCB switch must be made available to support the various RCAs in the Fabric. There is no need to have multiple switches if there are enough ports on the primary switches to support all the devices in the fabric.	
RoCE RCA	5 or more	Each vendor must supply 5 or more RCAs in order to support MPI testing.	

5.6 RoCE Software

Linux platforms
OFED - Most Current Tested Release
RoCE FW – Version XXX - Vendor Specific
10/40 GbE DCB Switch FW candidate – Version XXX - Vendor Specific
10/40 GbE DCB Switch SW – Version XXX - Vendor Specific

5.7 MPI testing

HCA/RCA/RNIC vendors must provide a minimum of *five* adapters. The adapters need not be all the same model (but certainly can be).

6 IB HW description & connectivity

The test contains two major parts. This description is for each of those parts.

6.1 Basic connectivity (P1P1)

1.) HCA 1 should be connected from port 1 to the switch.
2.) HCA 2 should be connected from port 1 to the switch.
3.) Both should be connected with compliant InfiniBand cables

6.2 Switches and Software Needed

Switches provided by OEMs

It is necessary that Switches provided by OEMs cover the full breadth of software versions supported by the Switch OEMs. Port count is not critical for the tests. It is recommended that OEMs provide six switches covering all variations of software supported on the Switches.

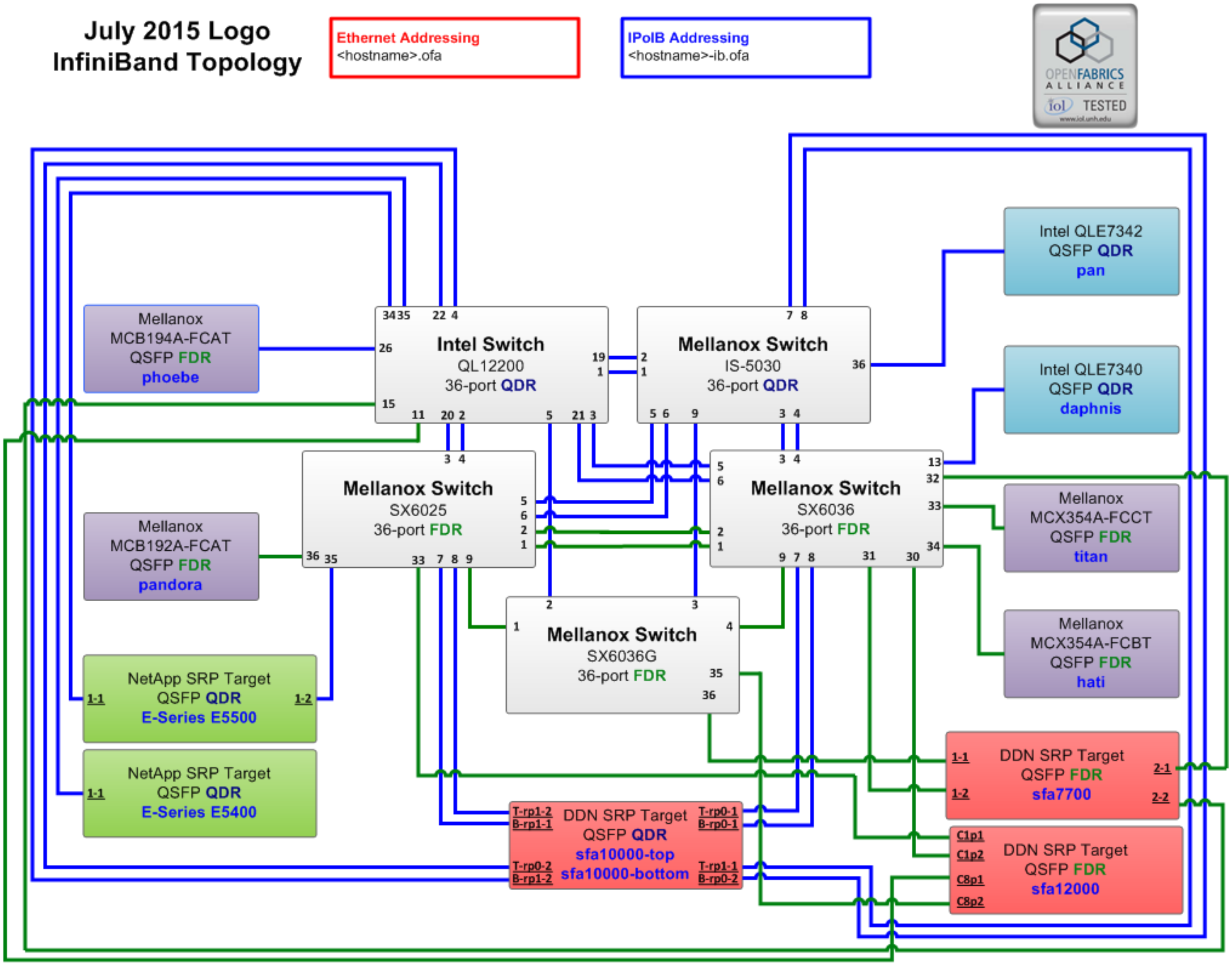
OpenFabrics software running on Hosts

Where there are dependencies of OEM provided and IBTA defined management software (such as subnet managers and agents, performance managers and agents etc.) with OpenFabrics software running on Hosts, such software should be provided to UNH-IOL for interoperability testing. Any known dependencies should be communicated to UNH-IOL.

6.3 Cluster Connectivity

Hosts and Targets 1-6 should be connected from port 1 or 2 to ports X in all switches using compliant InfiniBand cables.

Infiniband Interop Setup



7 iWARP HW description & connectivity

7.1 iWARP Basic connectivity (P1P1)

1.) When possible, RNIC 1 on one host should be directly connected to RNIC 2 on another host – otherwise they should be connected to a 10GbE switch with 10GbE cables.

7.2 Switches and Software Needed.

Switches provided by OEMs

It is necessary that Switches provided by OEMs cover the full breadth of software versions supported by the Switch OEMs. Port count is not critical for the tests. It is recommended that OEMs provide a switch per variations of software supported on the Switch.

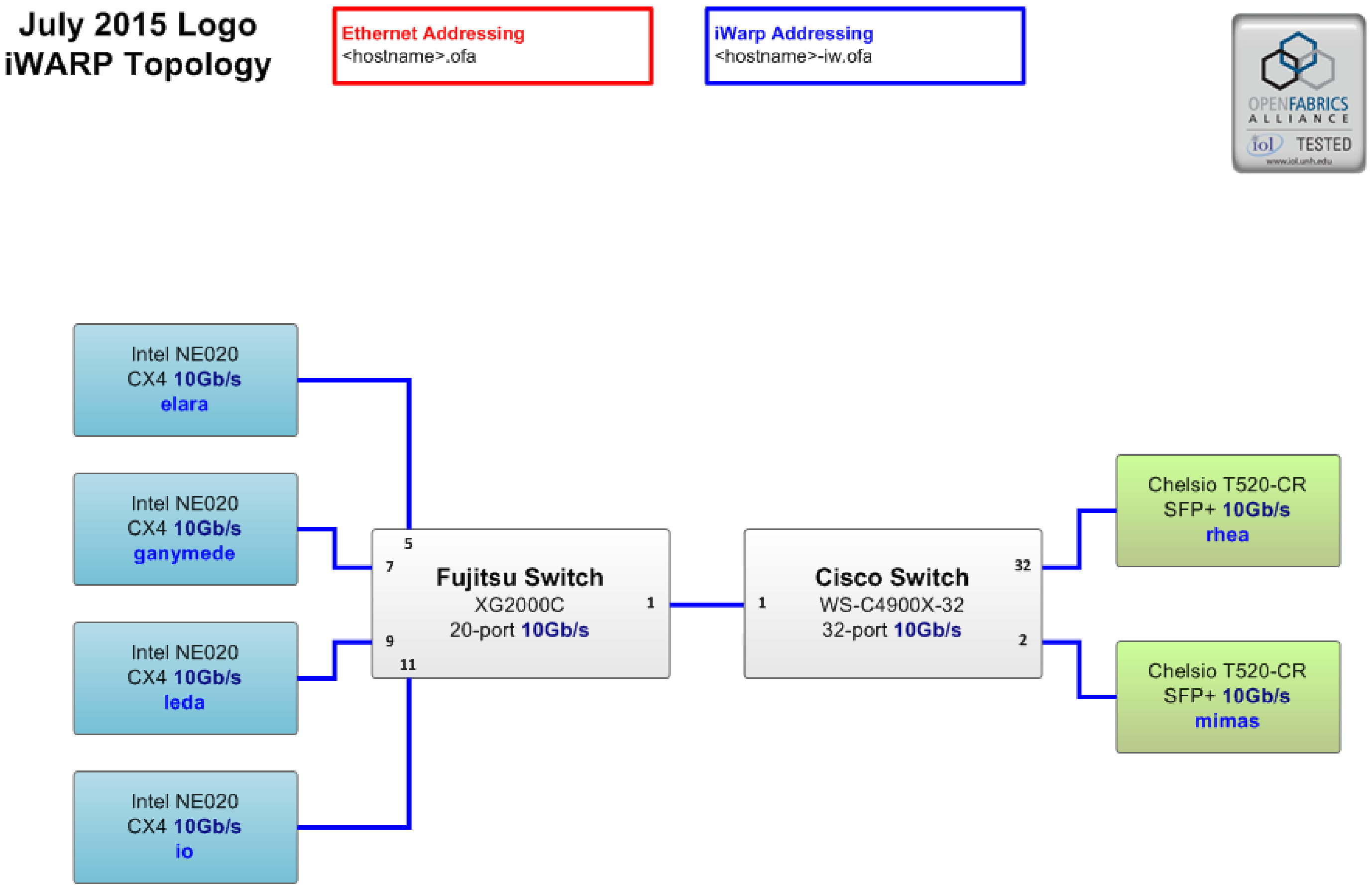
OpenFabrics software running on RNICs

Where there are dependencies of OEM provided with OpenFabrics software running on RNICs, such software should be provided to UNH-IOL for interoperability testing, and any known dependencies should be communicated with the UNH-IOL.

7.3 iWARP Fabric Connectivity

(Please note: Hosts should be connected to switches using 10GbE cables)

iWARP Interop Setup



7.4 Gateway, Bridges, Routers Connectivity (TBD)

8 RoCE HW description & connectivity

8.1 RoCE Basic connectivity (P1P1)

1.) RCA 1 on one host should be directly connected to RCA 2 on another host or to a 10/40 GbE Switch DCB enabled.
2.) Connected with 10/40 GbE cables

8.2 Switches and Software Needed

Switches provided by OEMs

RoCE testing was introduced as of October 2012 and the choice of Ethernet Fabrics such as Fabric Path, QFabric, MLAG, SPB, TRILL will not initially be addressed. This allows us to start Beta Testing RoCE with just one 10/40 GbE Ethernet Switch which is DCB enabled. In future Interop events, we will consider using multiple switches from vendors such as Brocade, Cisco, Extreme, HP, Mellanox and others, which will allow us to test various Ethernet Fabric solutions.

OpenFabrics software running on RCAs

Where there are dependencies OEM provided with OpenFabrics software running on RCAs, such software should be provided to UNH-IOL for interoperability testing, and any known dependencies should be communicated to UNH-IOL.

RoCE Priority Levels

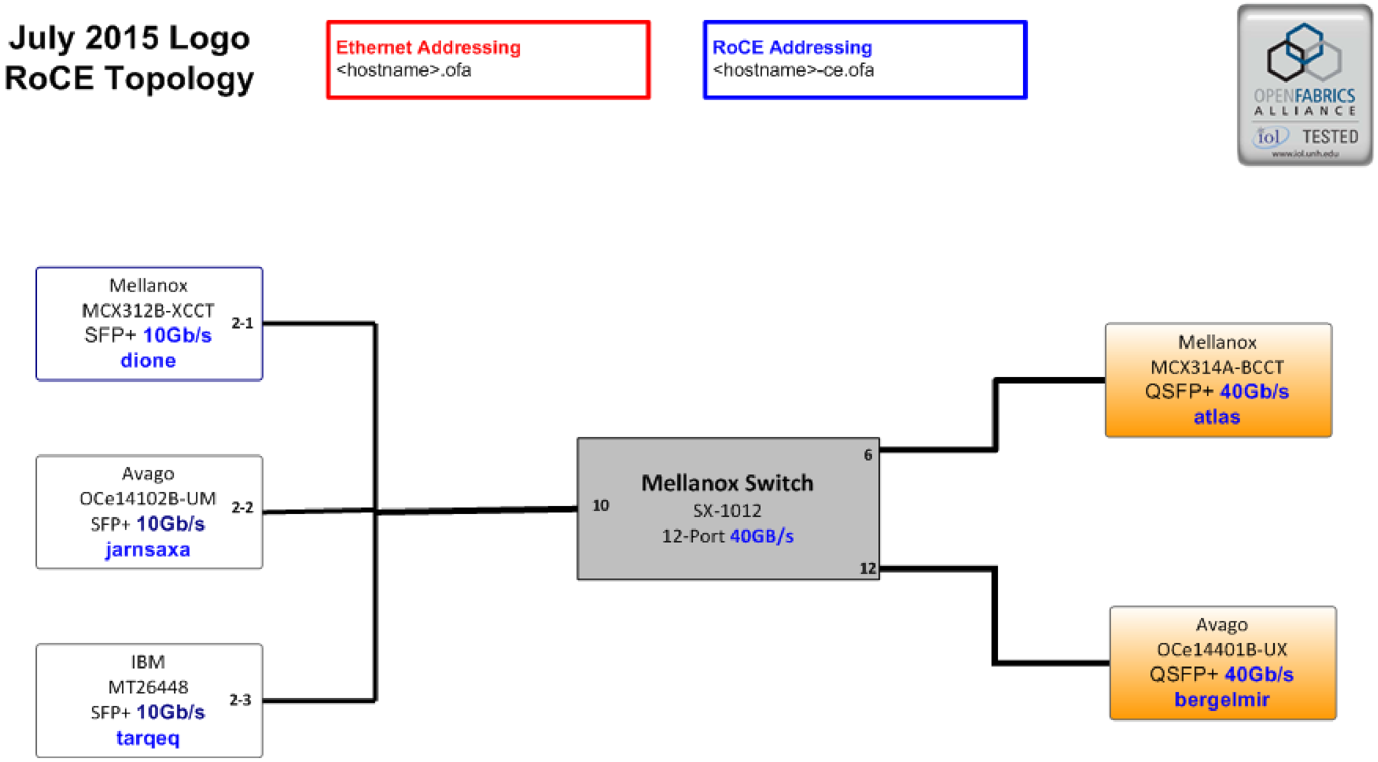
Ethernet provides a construct, called a Priority Level which corresponds conceptually to InfiniBand's SLs. Eight priorities, numbered zero through seven are supported. As in InfiniBand, a verbs consumer accessing a RoCE port specifies its desired service level, which is then mapped to a given Ethernet Priority. The default mapping is as follows:

1.) SL 0-7 are mapped directly to Priorities 0-7 respectively
2.) SL 8-15 are reserved.

8.3 RoCE Fabric Connectivity

(Please note: Hosts should be connected to switches using 10/40 GbE cables.)

RoCE Interop Setup



9 Firmware & Software installation

9.1 Firmware Policy

Firmware Policy during the Interop Debug Event

The firmware used during the Interop Debug Event is at the discretion of the device vendor. Vendors will be allowed to make changes to the firmware during the Interop Debug Event. However changes should be made as early in the event period as possible to reduce the amount of retesting which will result from these changes.

Firmware Policy during the Interop GA Event

The firmware image used during the Interop GA Event must be provided to the UNH-IOL at least one week prior to the event. No firmware changes of any kind are allowed during the Interop GA Event. If the vendor does not provide updated firmware by the deadline, then the UNH-IOL will use the firmware from the Interop Debug Event or from the vendor's website, whichever is more current.

Firmware Policy after the Interop GA Event

The firmware used to obtain the OFA Logo (or a child of this firmware with the same base functionality) must be the default publicly available firmware on the vendor's website and must be the default firmware that is shipped with the product. This must be completed within six months of the Interop GA Event. *Please refer to firmware burning tools and procedures documentation from HCA IB vendor!*

9.2 Operating System Policy

The OS used during an Interop Debug Event will be determined by the OFA IWG and will be known as the primary OS. All available updates will be installed prior to the start of the Interop Debug Event and frozen in place for the duration of the Interop Debug Event. In the event that some hardware is not supported by the primary OS, an alternate OS may be approved by the OFA IWG. As of the October 2014 Interop Debug Event, RHEL 7.x will be used for IBM RoCE Adapters since there are no CentOS, Scientific Linux or Ubuntu distributions for PowerPC platforms.

The OS used during an Interop GA Event will be the same agreed-upon versions of the OS tested during the Interop Debug Event. The updates applied at the start of the Interop Debug Event will remain frozen in place for the duration of the Interop GA Event.

9.2 Operating System Policy (continued)

In addition to the mandatory testing performed using the primary OS, beta testing using the secondary operating systems may be performed after completion of mandatory testing. The secondary operating systems are deployed in a similar manner to the primary OS, in that updates are applied at the beginning of the Interop Debug Event and frozen in place for the duration of the Interop GA Event.

9.3 Operating System Installation

Install the primary OS on all hosts in the cluster. Use a package manager to update all installed packages to their latest versions available as of the start of the Interop Debug Event.
Install the secondary operating systems on all hosts in the cluster. Use a package manager to update all installed packages to their latest versions available as of the start of the Interop Debug Event. Install and test as many secondary operating systems as time permits.

Ubuntu
For Ubuntu 12.04 and 12.10 Server edition, run the following commands to enable the IB interface and then assign the IP address in /etc/network/interfaces

- # apt-get install ibutils infiniband-diags srptools mpitests
- # modprobe mlx4_ib #Mellanox ConnectX cards
- # modprobe rdma_ucm
- # modprobe ib_umad
- # modprobe ib_ipoib

Note:
Most of the commands used here and in the following tests require root-level privileges. Either use 'sudo -i' to simulate a Root login shell or prepend 'sudo' to all the commands.
The OFED version included in packages and modules available in Ubuntu 12.04 and 12.10 is OFED 1.4.2.

9.4 Software Policy

Software Policy during an Interop Debug Event

The software used during an Interop Debug Event will be an agreed-upon RC release of the subsequent OFED version. During the Interop Debug Event, vendors will be allowed to make changes to the software - provided that the changes are based on the same RC release. Vendors are not allowed to extensively modify the software or completely replace it. A vendor supplied version of OFED may be used during the event if the current version of OFED does not include drivers required for a new product; However, the vendor must follow the guidelines described in the OFA Logo Program and make the drivers available within 6 months. Furthermore, the vendor must also include them in the next GA version of the OFED.

Software Policy during the Interop GA event

The software used during an Interop GA Event will be the GA release of the same OFED version as was used during the Interop Debug Event. No software changes of any kind are allowed during the Interop GA Event. It is the vendor's responsibility to ensure that any changes made during the Interop Debug Event are present in the OFED GA release. Vendors whose products do not use firmware may request that patches be applied to an OFED GA release if that release has known defects which prevents the vendor product from being interoperable. The Arbitration Committee will be responsible for approving the requested patches.

Software Policy after the Interop GA event

All products that are granted the OFA Logo must be distributed by default with the OFED GA version (or a later revision of OFED with the same base functionality).
(Note: Please refer to software installation manual from HCA IB vendors, and to software installation manual from RNIC vendors)

9.5 Summary

For the Interop GA Event the vendor cannot update or change any part of the device under test - this includes hardware, firmware and software. The only exception is for an outright hardware failure, in which case the hardware may be replaced with an identical piece of hardware with the same SW and FW.
If an end user requests customized firmware or a modified version of the OFED, then the vendor must disclose that the modified OFED is not an OFA certified configuration.
The OFA reserves the right to revoke the OFA Logo for products that do not follow these policies.
These policies will be in effect for the April 2011 Interop Events and all events thereafter.

9.6 Hardware Policy

For MPI testing, HCA/RNIC vendors must provide at least five adapters. The adapters need not be all the same model, but they are allowed to be.

9.7 OFED Usage

1.) OFED Release Candidates (RC) should be used during the Interop Debug Event. This allows vendors to resolve bugs and issues and commit them to the OFED tree before the OFED General Availability (GA) is released.

2.) OFED GA versions shall be used for the Interop GA Events.

10 General Instructions

10.1 First step Instructions

1.) Burn the FW release XXX on all HCAs and RNICs using the above procedure as required by vendor.

2.) Establish the Host and Target Configuration

   a. Install OFED software on host systems (using a 64 bit OS) configured to run OFED.
   b. Configure non-OFED systems for use in the cluster as per the vendors instructions.
   c. Configure iSER/SRP targets for use in the cluster as per the vendors instructions.
   d. Install the switch or gateway with the candidate SW stack as required by vendor.
   e. Burn the switch or gateway with the released FW as required by vendor.
   f. Connect the Hosts and Targets to an appropriate switch following the basic connectivity.

10.2 InfiniBand Subnet Managers

1.) The OpenSM will be used to run all mandatory tests in the test plan

2.) Vendor SM testing will include testing IPoIB, RDMA Interop and Open MPI testing. In order to reduce the scope of testing, iSER, NFS over RDMA, RDS, SDP, SM Failover and SRP will not be performed using vendor SMs.

10.3 Operating System Considerations

1.) The OFILG decided as of April 2012 that the various ULPs contained in this test plan will only be tested if it is supported by the Operating System.

2.) As a requirement for the OFILG Logo, a vendor's DUT must pass all mandatory testing using an agreed upon primary OS and OpenSM. Additional beta testing is performed using secondary Operating Systems. This beta testing has no bearing on whether the OFILG Logo is granted to a device It is purely informative.

11 InfiniBand Specific Interop Procedures using OFED

(Note: UNH-IOL has created automated scripts to run many of the OFED based tests. Please contact them at ofalab@iol.unh.edu if you wish to obtain copies of the latest scripts.

11.1 IB Link Initialization using OFED for Linux

Procedure

1.) Select a pair of devices to test from the created topology
2.) Determine the maximum port width and lane speed supported by both devices
3.) Select a cable to use which has been certified for the link parameters determined by step 2 of section 11.1.1 during an IBTA Plugfest held within the last 6 months
4.) Disconnect all IB cables from the selected devices
5.) Shutdown all SMs running on the selected devices
6.) Connect the selected devices back to back using the cable selected during step 3 of section 11.1.1
7.) Wait for a physical indication that a link has been established
8.) Verify that the link created in step 6 of section 11.1.1 has come up with the parameters determined in step 2 of section 11.1.1
9.) Repeat steps 1-8 with a different device pairing
    a. All unique device pairs present in the created topology must be tested; excluding SRP target to SRP target, and gateway to SRP target
    b. Each device must link at the maximum port width and lane speed supported by both devices in all pairings for said device to pass link initialization testing

Method of Implementation for all Linux OSs

1.) To perform step 7 of section 11.1.1:
    a. Look for link LEDs on the ports that are being used
2.) To perform step 8 of section 11.1.1:
    a. ssh into a device supporting such remote connections and is running the OFED stack; usually a compute node with an HCA
    b. Run "# ibdiagnet -wt <desired-topology-file-name>"
    c. Check the topology file created by the previous command:
        i. Match the GUIDs to the devices in the selected pair
        ii. Verify link width is the highest common denominator of pair capabilities (1x, 4x, 12x)
        iii. Verify link speed is the highest common denominator of pair capabilities (2.5G, 5G, 10G, 14G)
3.) To determine switch to SRP target and switch to switch link parameters:
    a. Run the commands outlined by step 2 of section 11.1.2 from a third device
        i. Should be a compute node with an HCA that is linked to a switch that is part of the desired pairing
        ii. Carefully match the GUIDS as there are more than just two in the topology file

11.2 IB Fabric Initialization using OFED

Architect the Network we want to build.

1.) Develop a cluster diagram based on the devices that have been submitted for Interop Testing and assign IP addresses to the IPoIB interfaces and the ethernet management interfaces (please see below).

Procedure

1.) Connect the HCAs and switches as per the Architected Network and make sure that no SM/SA is running on the Fabric.
2.) Start an SM on a device and let it initialize (all SM's will need to be tested)
3.) Visually verify that all devices are in the active state. Verify that the LED is on when the port is active.
4.) Run "# ibdiagnet -wt <file>" to generate a topology file
5.) Run "# ibdiagnet -pc" to clear all port counters
6.) Wait 17 seconds as per the specifications requirements.
7.) Run "# ibdiagnet -c 1000" to send 1000 node descriptions.
8.) Run "# ibdiagnet" to generate fabric report.
    a. Use /tmp/ibdiagnet.sm file to determine running sm.
    b. sminfo can also be used to determine the master SM or saquery -s to find all SMs.
9.) Run "# ibchecknet" to build guid list.
10.) Run "# ibdiagnet -t <file>" to compare current topology to the previously generated topology file
    a. **(Note**: For Ubuntu, "# ibdiagnet -t <tp>" requires local system name specified. Use ibstat to find a match and do "# ibdiagnet -s <sys name> -t <tp>".)

Verification Procedures

1.) Review "PM Counters" section of the fabric report. There should be no illegal PM counters. The Specification says there should be no errors in 17 seconds.
2.) Review "Subnet Manager" section of the fabric report. Verify that the running SM is the intended one to be started and verify number of nodes and switches in the fabric.
3.) Review the "ibchecknet" report and verify that there are no duplicate GUIDs in the fabric
4.) Verify that step 10 above indicates that the topology before the test and the topology after the test are the same.

Restart all devices in the fabric and follow Sections 11.2.2 and 11.2.3. Run the SM from a different device in the fabric until all SMs present have been used. All SMs on managed switches (including those switches running **opensm**) should be tested and at least one instance of **opensm** on an HCA must be tested. If there are HCAs from more than one vendor, then **opensm** should be run from each vendor's HCA.

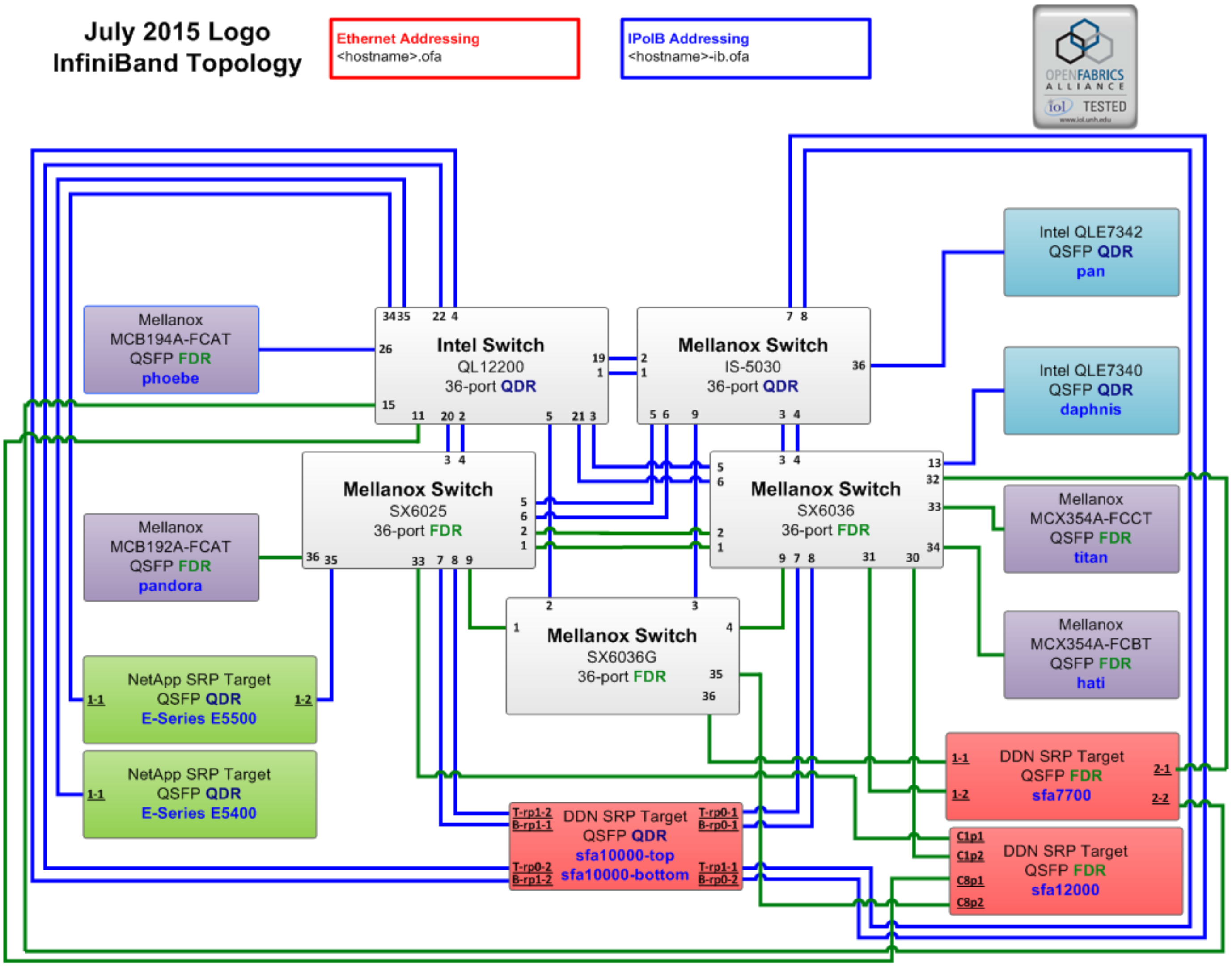
(**Note:** Each device must pass all verification procedures with every SM to pass Fabric Initialization test.)

ibdiagnet commands

Commands	**Description**
# ibdiagnet -c 1000	Send 1000 node descriptions
# ibdiagnet -h	Help
# ibdiagnet -lw 4x - ls 2.5	Specify link width and speed
# ibdiagnet - pc	Clear counters
# ibdiagnet -t <file>	Compare current topology to saved topology
# ibdiagnet -wt	Writes the topology to a file

Note: The topology file is being generated after the SM starts but before any testing has started. The topology comparison is being performed after testing has been completed but before the systems get rebooted. A topology check is performed during every part of every test section that does not specifically state "change the topology". For example Fabric Init only has 1 part so there is only 1 check but RDS has 2 parts so 2 checks are performed. However, IPoIB has 3 parts for each of 2 modes but 1 of those parts specifically says to change the topology so only 4 checks occur.

Sample Network Configuration



11.3 IB IPoIB Connect Mode (CM) using OFED

Setup

Connect the HCAs and switches as per the Architected Network and make sure that no SM is running on the Fabric.

This procedure, as the previous ones, will be based on the cluster connectivity. An SM/SA which supports IPoIB (sufficient IB multicast support) will be running on the HCAs, or on a switch with an embedded SM/SA or a third HCA which would only run SM/SA for the partner pair (with a switch in the middle). This procedure has been developed for Linux and may be ported to Windows if there is sufficient vendor support.

Optional: In the procedures below, an IB analyzer can be inserted in the appropriate link to obtain traces and validate the aspects of the procedures specifically detailed below in subsequent sections.

IPoIB Interface Creation and IPoIB Subnet Creation

1.) Configure IPoIB address. All addresses must reside on the same subnet.
2.) Set interfaces to 10.0.0.x/24 (10.0.0.x/netmask 255.255.255.0) using the command:
   - $ ifconfig ib0 10.0.0.x netmask 255.255.255.0

Bringing the IPoIB in Connected Mode

1.) “# echo 'connected' > /sys/class/net/ib0/mode”
2.) Validate CM mode by checking that "/sys/class/net/<I/F name>/mode" equal to '**connected**’
3.) Repeat steps 1 and 2 on all nodes being tested.

11.3.1 Ping Procedures

Step A

1.) Stop all SM's and verify that none are running
2.) Power cycle all switches in the fabric (this insures that the new SM will configure all the links and create the multi-cast join)
3.) Start an SM (All SM's will need to be tested) and let it initialize
4.) Visually verify that all devices are in the active state. Verify that the LED is on when the port is active.
5.) Run "# ibdiagnet -r" to verify that the SM that was intended to start is the one that is running and that it is the master. You will need to know the GUID of the device since the SM will be reassigned on each reboot.
6.) Verify that all nodes and switches were discovered.
7.) **Note**: Ibdiagnet may show more switches than indicated by the physical number of switch platforms present. This is because some switches have multiple switch chips.

8.) Examine the arp table (via $ ip neigh show) and remove the destination node’s ib0 address from the sending node’s arp table (via # ip neigh flush <destination>).
9.) Ping every HCA except localhost with packet sizes of 511, 1025, 2044, 8192, 32768 and 65492.
10.) “$ ping -i 0.2 -t 3 -c 10 -s <ping size> <destination>”
    a. "-i" - interval 0.2 seconds
    b. "-t" - IP Time to Live equals 3 seconds
    c. "-c" - count equals 100
    d. "-s" - size of the ping
    e. "destination" - the IP address of the IPoIB interface being pinged.
11.) Repeat step #8 before issuing each ping command. Every packet size is a new ping command.

Note: In order to pass Step A, a reply must be received for every ping sent (without losing a single packet) while using each one of the SMs available in the cluster.

Informative test: ping every HCA except localhost with packet size 65507

Step B

1.) Bring up all HCAs but one.
2.) Start an SM (all SMs will need to be tested).
3.) Check for ping response between all node (All to All).
4.) A response from the disconnected HCA should not be returned.
5.) Disconnect one more HCA from the cluster.
6.) Ping to the newly disconnected HCA from all nodes (No response should be returned).
7.) Connect the first machine (the one that was not connected) and check for ping response from all nodes that are still connected.
8.) Connect the disconnected HCA to a different switch on the subnet which will change the topology.
9.) Ping again from all nodes (this time, there should be a response).
10.) Follow Step B, this time bring the interface down and then back up using ifconfig ibX down and ifconfig ibX up commands instead of physically disconnecting the HCAs.

Note: Each step must exhibit the expected behavior while using each SM in order for the device to pass Step B overall.

Step C

1.) Follow Step A and B using a different SM until all SM's have been used. Only one instance of each available SM is required. Steps A, B, and C must pass in order for the device to pass 10.3.4 overall.

11.3.2 SFTP Procedure

SFTP procedures require an SFTP server to be configured on each machine in the partner pair. An SFTP client needs to be available on each machine as well. The default RHEL install includes both.
A 4 MB file will be SFTP'd to the partner and then SFTP'd back and binary compared to the original file, this will be done in each direction and then bidirectional using every SM available.

Setup

1.) Make sure vsftpd is installed on each node for SFTP application.
2.) A special account for this should be created as follows:
    a. Username: Interop
    b. Password: openfabrics

Procedure

1.) Run SFTP server on all nodes.
2.) Start an SM (all SM's will need to be tested) and let it initialize
3.) Verify that the running SM is the intended one to be started.
4.) SFTP:
    a. Connect an HCA pair via SFTP on IPoIB using the specified user name and password.
    b. Put the 4MB file to the /tmp dir on the remote host.
    c. Get the same file to your local dir again.
    d. Compare the file using the command *cmp tfile tfile.orig.*
        i. The two must be identical
5.) Repeat the procedure with a different SM.

Note: Every node must SFTP the 4MB file to all others using all SM's and the files must be identical as determined by the binary compare in order for the device to pass 10.3.5 overall.

Note: Sections 10.3.4 and 10.3.5 must pass using the configuration determined by sections 10.3.1, 10.3.2, and 10.3.3 for the device to pass IPoIB Connected mode overall.

11.4 IB IPoIB Datagram Mode (DM) using OFED

Setup
Connect the HCAs and switches as per the Architected Network and make sure that no SM is running on the Fabric.

This procedure, as the previous ones, will be based on the cluster connectivity. An SM/SA which supports IPoIB (sufficient IB multicast support) will be running on the HCAs, or on a switch with an embedded SM/SA or a third HCA which would only run SM/SA for the partner pair (with a switch in the middle). This procedure has been developed for Linux and may be ported to Windows if there is sufficient vendor support.

Optional: In the procedures below, an IB analyzer can be inserted in the appropriate link to obtain traces and validate the aspects of the procedures specifically detailed below in subsequent sections.

IPoIB Interface Creation and IPoIB Subnet Creation

1.) Configure IPoIB address. All addresses must reside on the same subnet.
    a. Set interfaces to 10.0.0.x/24 (10.0.0.x/netmask 255.255.255.0) using the command *ifconfig ib0 10.0.0.x netmask 255.255.255.0*

Bringing the IPoIB in Datagram Mode

1.) “# echo 'datagram' > /sys/class/net/ib0/mode”
2.) Validate DM mode by checking that "/sys/class/net/<I/F name>/mode" equal to '**datagram**’
3.) Repeat steps 1-2 on all nodes being tested.

11.4.1 Ping Procedures

Step A

1.) Stop all SM's and verify that none are running
2.) Power cycle all switches in the fabric (this insures that the new SM will configure all the links and create the multi-cast join).
3.) Start an SM (All SM's will need to be tested) and let it initialize
    a. Visually verify that all devices are in the active state. Verify that the LED is on when the port is active.
    b. Run "# ibdiagnet -r" to verify that the SM that was intended to start is the one that is running and that it is the master. You will need to know the GUID of the device since the SM will be reassigned on each reboot.
    c. Verify that all nodes and switches were discovered.

Note: Ibdiagnet may show more switches than indicated by the physical number of switch platforms present. This is because some switches have multiple switch chips.

4.) Examine the arp table (via $ ip neigh show) and remove the destination node's ib0 address from the sending node's arp table (via # ip neigh flush <destination>).
5.) Issue the command: "# sysctl net.ipv4.neigh.ib0.unres_qlen=33"
    a. This sets the qlen variable to 33 which increases the buffer size so that the initial packet does not get when using ping sizes 8192 and greater.
6.) Ping every HCA except localhost with packet sizes of 511, 1025, 2044, 8192, 32768, 48568
7.) "$ ping -i 0.2 -t 3 -c 10 -s <ping size> <destination>"
    a. "-i" - interval 0.2 seconds
    b. "-t" - IP Time to Live equals 3 seconds
    c. "-c" - count equals 100
    d. "-s" - size of the ping
    e. "destination" - the IP address of the IPoIB interface being pinged.
8.) Repeat step #4 before issuing each ping command. Every packet size is a new ping command.

Note: In order to pass Step A, a reply must be received for every ping sent (without losing a single packet) while using each one of the SMs available in the cluster.

Informative test: ping every HCA except localhost with packet size of 65507

Step B

1.) Bring up all HCAs but one.
2.) Start an SM (all SMs will need to be tested).
3.) Check for ping response between all node (All to All).
    a. A response from the disconnected HCA should not be returned.
4.) Disconnect one more HCA from the cluster.
5.) Ping to the newly disconnected HCA from all nodes (No response should be returned).
6.) Connect the first machine (the one that was not connected) and check for ping response from all nodes that are still connected.
7.) Connect the disconnected HCA to a different switch on the subnet which will change the topology.
8.) Ping again from all nodes (this time, there should be a response).
9.) Follow Step B, this time bring the interface down and then back up using "$ ifconfig ibX down" and "$ ifconfig ibX up" commands instead of physically disconnecting the HCAs.

Note: Each step must exhibit the expected behavior while using each SM in order for the device to pass Step B overall.

Step C

1.) Follow Step A and B using a different SM until all SM's have been used. Only one instance of each available SM is required. Steps A, B, and C must pass in order for the device to pass 10.4.4 overall.
2.) Issue the command: "# sysctl net.ipv4.neigh.ib0.unres_qlen=3"
    a. This sets the qlen variable back to the default.

11.4.2 SFTP procedure

SFTP procedures require an SFTP server to be configured on each machine in the partner pair. An SFTP client needs to be available on each machine as well. The default RHEL install includes both.
A 4 MB file will be SFTP'd to the partner and then SFTP'd back and binary compared to the original file, this will be done in each direction and then bidirectional using every SM available.

Setup

1.) Make sure vsftpd is installed on each node for SFTP application.
2.) A special account for this should be created as follows:
    a. Username: Interop
    b. Password: openfabrics

Procedure

Run SFTP server on all nodes.

1.) Start an SM (all SM's will need to be tested) and let it initialize
    a. Verify that the running SM is the intended one to be started.
2.) SFTP:
    a. Connect an HCA pair via SFTP on IPoIB using the specified user name and password.
    b. Put the 4MB file to the /tmp dir on the remote host.
    c. Get the same file to your local dir again.
    d. Compare the file using the command *cmp tfile tfile.orig.*
        i. The two must be identical
3.) Repeat the procedure with a different SM.

Note: Every node must SFTP the 4MB file to all others using all SM's and the files must be identical as determined by the binary compare in order for the device to pass 10.4.5 overall.

Note: Sections 10.4.4 and 10.4.5 must pass using the configuration determined by sections 10.4.1, 10.4.2, and 10.4.3 for the device to pass IPoIB Datagram mode overall.

11.5 IB SM Failover and Handover Procedure using OFED

Setup

1.) Connect HCAs per the selected topology.
2.) In this test, all active SMs on the fabric which are going to be tested, must be from the same vendor. They will be tested pairwise; two at a time.

Procedure

1.) Disable all SMs in the cluster then start a SM on either machine in a chosen pair.
2.) Run "saquery" on a node in the fabric.
a. Verify that all nodes in the cluster are present in the output
3.) Using the ibdiagnet tool with the -r option, verify that the running SM is the master.
4.) Start a SM on the second machine in the current pair.
5.) Verify that the SMs behave according to the SM priority rules. Use "# ibdiagnet -r" again.
    a. SM with highest numerical priority value is master and the other is in standby.
    b. If both SMs have the same priority value then the SM with the smallest guid is master and the other is in standby.
6.) Run "saquery" on either machine in the current pair.
a. Verify that all nodes in the cluster are present in the output.
7.) Shutdown the master SM.
8.) Verify the other active SM goes into the master state using "# ibdiagnet -r" again.
9.) Run "saquery" on either machine in the current pair.
a. Verify that all nodes in the cluster are present in the output.
10.) Start the SM that was just shutdown.
    a. Verify that the newly started SM resumes it's position as master while the other goes into standby again.
11.) Run "saquery" on either machine in the current pair.
    a. Verify that all nodes in the cluster are present in the output.
12.) Shutdown the standby SM.
13.) Verify that the previous master SM is still the master.
14.) Run "saquery" on either machine in the current pair.
    a. Verify that all nodes in the cluster are present in the output.
15.) Repeat steps 1-14 above 2 more times, ensuring that the below criteria is met (total of 3 tests per pair which can be run in any order):
    a. First SM to be started having highest numerical priority value.
    b. Second SM to be started having highest numerical priority value.
    c. Both SMs having equal numerical priority values.

Repeat steps 1-15 until all possible SM pairs from identical vendors in the cluster have been tested.

Note: All of the "saquery" commands must return the expected list of nodes in order for the SMs in this test to receive a passing grade.

11.6 IB SRP using OFED

Setup

1.) Edit the file srp_daemon.conf and make sure it contains the following line
   a. “a max_sect=65535,queue_size=128”
2.) Connect the HCAs and switches as per the Architected Network and make sure that no SM is running on the Fabric.

Note: As of the April 2012 Interop events, one SRP target (i.e.target port) should present 2 or more volumes. All other target ports may be limited to one volume per port. This decision was made in order to reduce the amount of time required to run the tests.

Note: As of October 2012, the SRP Extended Procedure is a Beta test

Note: NetApp targets only support writing and reading from one controller at a time. Therefore only one controller per device should be attached to the test fabric. The controller that accepts writes is on a per LUN basis, the controller that owns the volume.

SRP Core Procedure - Mandatory

1.) Start an SM (all SM's will need to be tested) and let it initialize
   a. Verify that the running SM is the intended one to be started
2.) Choose a node to work with
3.) Unload the srp module
4.) Load srp module with “cmd_sg_entries=64 allow_ext_sg=1 indirect_sg_entries=512”
   a. **Example**: “# modprobe ib_srp cmd_sg_entries=64 allow_ext_sg=1 indirect_sg_entries=512”
   b. Let it initialize
5.) Verify that the module loaded correctly
   a. **Example**: “$ lsmod | grep ib_srp”
6.) Load srp_daemon with -e -o -n options
   a. **Example**: “$ srp_daemon -e -o -n”
   b. Let it initialize
7.) Tune all volumes or block devices.
   a. This allows larger IO sizes, faster testing, and more closely simulates a basic HPC file system environment.
   b. $drive = SCSI block device or volume (sdc, sdd, etc).
   c. # echo 4096 > /sys/block/$drive/queue/max_sectors_kb
      - Increases max SCSI command size to 4MB.
   d. # echo deadline > /sys/block/$drive/queue/scheduler
      - Faster IO scheduler
   e. # echo 0 > /sys/block/$drive/queue/nomerges
      - Enable IO merging requests
   f. # echo 4096 > /sys/block/$drive/queue/read_ahead_kb

- Increase maximum read ahead for block device
- Also increases IO size for cached read requests

8.) Find all volumes from all targets
    a. Use lsscsi

Note: As of April 2012, the OFILG mandated that the target only include two volumes when doing mandatory testing.

Note: For Ubuntu, lsscsi is not installed by default. Please do “# apt-get install lsscsi” to install it.

9.) Perform 20GB read from srp volume to null
    a. **Example**: “# dd if=$drive of=/dev/null count=2000 bs=10M”
10.) Perform 20GB write from zero to srp volume
    a. **Example**: “# dd if=/dev/zero of=$drive count=2000 bs=10M”
11.) Perform 20GB read from srp volume to null using Direct IO, 32M request size
    a. **Example**: “# dd if=$drive of=/dev/null count=640 bs=33554432 iflag=direct”
12.) Perform 20GB write from zero to srp volume using Direct IO, 32M request size
    a. **Example**: “# dd if=/dev/zero of=$drive count=640 bs=33554432 oflag=direct”
13.) Perform 1M read from srp volume’s SCSI Generic device to null, 20GB total length
14.) Perform 1M write from zero to srp volume’s SCSI Generic device, 20GB total length
15.) Perform 4M read from srp volume’s SCSI Generic device to null, 20GB total length
16.) Perform 4M write from zero to srp volume’s SCSI Generic device, 20GB total length
17.) Perform 8M read from srp volume’s SCSI Generic device to null, 20GB total length
18.) Perform 8M write from zero to srp volume’s SCSI Generic device, 20GB total length
19.) Perform 16M read from srp volume’s SCSI Generic device to null, 20GB total length
20.) Perform 16M write from zero to srp volume’s SCSI Generic device, 20GB total length

The following example shows a method in bash for performing steps 12 through 19. Proper block alignment must be used when accessing SCSI Generic devices.

```
# Total transfer length, 20GB
DD_XFER_SIZE=21474836480
SGP_DD="sgp_dd"
VENDOR=<target_vendor_name>
# IO sizes to test
SGP_DD_SIZE='1048576 4194304 8388608 16777216'
# Use sgp_dd to perform IO to each SCSI Generic device (sg)
for block_device in $(lsscsi | grep $VENDOR | grep "disk" | awk '{print $(NF)}'); do
        # Some volumes are not 512b sectors, get the proper block size for alignment
        physical_blocksize=`sudo su -c "blockdev --getss ${block_device}"`;
        # Get the sg device for the block device
        sg_device=$(lsscsi -g | grep $VENDOR | grep "disk" | grep ${block_device} | awk '{print
        $(NF)}');
        for IO_SIZE in $SGP_DD_SIZE ; do
                # Calculate BPT based on IO size and physical blocksize
                BPT=$((${IO_SIZE}/$physical_blocksize));
                # sgp_dd read
                sudo $SGP_DD if=$sg_device of=/dev/null bs=$physical_blocksize bpt=$BPT thr=8
                count=$(($DD_XFER_SIZE/$physical_blocksize)) time=1
                # sgp_dd write
                sudo $SGP_DD if=/dev/zero of=$sg_device bs=$physical_blocksize bpt=$BPT thr=8
                count=$(($DD_XFER_SIZE/$physical_blocksize)) time=1
        done
done
```

21.) Perform steps #8 and #9 for both volumes found from each target as determined by step #7
22.) Unload srp module
23.) Repeat steps 2 through 9 for all HCAs
24.) Reboot all devices in the fabric and repeat the procedure using a different SM.

Note: An HCA must successfully complete all DD and SGP_DD operations to and from all volumes on all targets using all available SM's in order to pass SRP testing.

11.7 IB iSER using OFED

Setup:

1) Connect HCA’s and switches as per the topology
2) Start an SM (all SM's will need to be tested) and let it initialize
   - Verify that the running SM is the intended one to be started
3) Install iscsi-initiator-utils and device-mapper-multipath packages on all hosts under test
   - # yum install iscsi-initiator-utils device-mapper-multipath -y
4) Edit /etc/iscsi/iscsid.conf file with the proper parameters:
   - # echo -e "node.startup = automatic_replacement_timeout = 20" | tee /etc/iscsi/iscsid.conf
5) Run the following command to generate the iSER configuration file:
   - # iscsiadm -m iface -I iser | tee /var/lib/iscsi/ifaces/iface-ib0
6) Make a file called multipath.conf in the /etc/ directory:
   - # touch /etc/multipath.conf (NOTE: contents of file are empty)
7) Restart the multipath service:
   - # service multipathd restart
8) Restart the iSCSI service:
   - # service iscsid restart
9) Load the iSER driver:
   - # modprobe ib_iser

Testing Procedure:

1) Discover targets with iscsadm:
   - # iscsiadm -m discovery -t st -p <target_ip_address> -I iser
2) Log into discovered targets:
   - # iscsiadm -m node -p <target_ip_address> -l
3) Verify the iSER connection:
   - # iscsiadm -m session | grep iser
4) Find the mount point:
   - # multipath -ll
5) Make a partition on the target:
   - # parted -a optimal -s -- /dev/mapper/<drive> mklabel gpt mkpart primary ext4 0% 100%
6) Make the file system on the partition:
   - # mkfs.ext4 <drive>
7) Make a mount point to mount the drive to:
   - $ mkdir /tmp/<directory_name>
8) Change permissions of the directory just created:
   - # chmod 777 -R /tmp/<directory_name>
9) Mount the drive :
   - # mount /dev/mapper/<drive> /tmp/<diectory_made>
10) Do a 6GB write from /dev/zero to a file on the partition:
   - # head -c 6G < /dev/zero > /tmp/netapp/<test_file_name>.tmp

11.8 IB Ethernet Gateway using OFED

Procedure

1.) Connect the HCA of the IB host to the IB fabric. Connect the Ethernet Gateway to the IB fabric. Connect the Ethernet gateway to the Ethernet network or Ethernet device. Start the SM to be used in this test.
2.) Determine which ULP the ethernet gateway uses and be sure that ULP is running on the host (VNIC or IPoIB).
3.) Restart the ULP or using the tool provided by the ULP, make sure that the host “discovers” the Ethernet Gateway. Configure the interfaces and make sure they are up.
4.) Run ping from the host to the Ethernet device. While the ping is running, kill the master SM. Verify that the ping data transfer is unaffected.
5.) Reboot the Ethernet Gateway. After the Ethernet Gateway comes up, verify that the host can discover the Ethernet Gateway as it did before and able to configure the interfaces.
6.) Restart the ULP used by Ethernet Gateway and verify that after the ULP comes up, the host can discover the Ethernet Gateway and able to configure the interfaces.
7.) Unload the ULP used by Ethernet Gateway and check that the Ethernet Gateway shows it disconnected. Load the ULP and verify that the Ethernet gateway shows the connection.
8.) Repeat step 4 by using ssh and scp instead of ping.

11.9 IB Fibre Channel Gateway using OFED

Procedure

1.) Connect the HCA of the IB host to the IB fabric.
    a. Connect the FC Gateway to the IB Fabric (how to do this is determined by the FC Gateway vendor).
    b. Connect the FC Gateway to the FC network or FC device.
    c. Start the SM to be used in this test.
2.) Configure the FC Gateway appropriately (how to do this is vendor specific).
3.) Use ibsrpdm tool in order to have the host "see" the FC storage device.
    a. Add the storage device as target.
4.) Run basic dd application from the SRP host to the FC storage device.
5.) Run basic dd application from the SRP host to the FC storage device.
    a. While the test is running, kill the master SM. Verify that the test completes properly.
6.) Unload the SRP host / SRP Target (target first/host first) and check that the SRP connection is properly disconnected.
7.) Load the SRP host / SRP Target. Using ibsrpdm, add the target.
8.) Run basic dd application from the SRP host to the FC storage device.
9.) Reboot the FC Gateway.
    a. After FC Gateway comes up, verify using ibsrpdm tool that the host see the FC storage device. Add the storage device as target.
10.) Run basic dd application from the SRP host to the FC storage device.
11.) Follow steps 1-10 above with each SM to be tested and with each HCA to be tested, until each HCA and each SM has been tested with the FC Gateway.

12 Ethernet Specific Interop Procedures using

12.1 iWARP Link Initialize using OFED

Purpose

The iWARP Link Initialize test is a validation that all iWARP devices receiving the OFA Logo can link and pass traffic under nominal (unstressed) conditions.

Resource Requirements

1.) Gigabit or 10Gigabit iWARP RNIC,
2.) Gigabit or 10Gigabit Ethernet Switch
3.) Compliant Cables

Discussion

The validation of the underlying transport infrastructure is essential to the end-users experience of the operation of the OFED software stack. To this end, this test confirms that iWARP devices receiving the OFA Logo can suitably link and pass traffic in any configuration. Exhaustive compliance testing of BER performance of the channel or electrical signaling of the ports is not performed; however, successful completion of this test provides further evidence of the robustness of the OFA logo bearing device.

Procedure

1.) Connect the two link partners together utilizing compliant cables.
2.) Check all relevant LEDs on both ends of the link.
3.) Verify that basic IP connectivity can occur by driving minimum size ICMP echo requests and replies across the link or equivalent traffic (including RDMA traffic if readily configured, in which case an additional RNIC responder station is required). To verify that an RDMA link has been initialized between Host A and Host B run the following commands:
    a. Start a persistent server in verbose mode on Host A:
        i. “$ rping -svP”
    b. Start a client on Host B to ping the server.
        i. “$ rping -cv -a *ServerHostName*-iw*”*
    c. Optional Command for the client
        i. “$ rping -cv -a Host A *RNIC_IP_Address* -C 4 -S 50”

Note: The optional command sends a count of 4 pings and character strings of size 50.

4.) Repeat steps 1-3 for all combinations of 2 RNICs to switches, switch to switch, and RNIC to RNIC link partner combinations. Previously tested combinations resident in the OFILG cluster may be omitted.

Observable results

1.) Link should be established on both ends of the channel.
2.) Traffic should pass in both directions. Error rates of 10e-5 or better should be readily confirmed (no lost frames in 10,000).

Possible Problems

1.) Traffic directed to a switches IP management address may not be processed at high speed, in such cases, traffic should be passed across the switch to a remote responder.

12.2 RCA Basic Connectivity

Purpose

The RoCE Link Initialize test is a validation that all RoCE devices receiving the OFA Logo can link and pass traffic under nominal (unstressed) conditions.

Resource Requirements

1.) 10 or 40 Gigabit RoCE Channel Adapter (RCA)
2.) 10 or 40 Gigabit RoCE Switch (DCB Enabled)
3.) Compliant Cables

Discussion

The validation of the underlying transport infrastructure is essential to the end-users experience of the operation of the OFED software stack. To this end, this test confirms that RoCE devices receiving the OFA Logo can suitably link and pass traffic in any configuration. Exhaustive compliance testing of BER performance of the channel or electrical signaling of the ports is not performed; however, successful completion of this test provides further evidence of the robustness of the OFA logo bearing device.

A flow control mechanism must be agreed upon prior to the start of testing, and configured appropriately. In the future, a flow control mechanism may be mandated.

Procedure

1.) Connect RCA's per the selected topology utilizing compliant cables.
2.) Check all relevant LEDs on both ends of the link.
    a. Verify connectivity between hosts by sending ICMP echo requests between hosts.
3.) Verify that basic IP connectivity can occur by driving minimum size ICMP echo requests and replies across the link or equivalent traffic (including RDMA traffic if readily configured, in which case an additional RoCE responder station is required). To verify that an RDMA link has been initialized between Host A and Host B run the following commands:
    a. Start a server in verbose mode on Host A:
        i. "$ rping -sv -a Host A *RCA_IP_Address*"
    b. Start a client on Host B to ping Host A.
        i. "$ rping -cv -a Host A *RCA_IP_Address* -C 4 -S 50"
4.) Repeat steps 1-3 for all combinations of 2 RCAs to switches, switch to switch, and RCA to RCA link partner combinations. Previously tested combinations resident in the OFILG cluster may be omitted.

Observable results

1.) Link should be established on both ends of the channel.
2.) Traffic should pass in both directions. Error rates of 10e-5 or better should be readily confirmed (no lost frames in 10,000).

12.3 RoCE Fabric Init using OFED

(**Note:** This test will be developed for a future version of the test plan.)

12.4 RoCE IPoCE

Setup

Connect RCAs and switches as per the Fabric Diagram.

IPoCE Network Configuration

1.) Configure IPoCE addresses so that all hosts reside on the same subnet.
   a. Set interfaces to 10.3.X.X/16 via ifconfig eth2 10.3.X.X netmask 255.255.0.0

12.4.1 Ping Procedures

Step A

1.) Power cycle all switches in the fabric.
2.) Visually verify that all devices are in the active state by inspecting link LEDs.
3.) Examine the arp table (via $ ip neigh show) and remove the destination node's eth0 address from the sending node's arp table (via # ip neigh flush <destination>).
4.) Ping every RCA in the topology except localhost with packet sizes of 511, 1025, 2044, 8192, 32768, and 34741.
   a. "$ ping -i 0.2 -t 3 -c 10 -s <payload size> <destination>"
      i. "-i" - interval 0.2 seconds
      ii. "-t" - IP Time to Live equals 3 seconds
      iii. "-c" - count equals 10
      iv. "-s" - size of the ping
      v. "destination" - the IP address of the IPoIB interface being pinged.
5.) Clear the arp table before each ping. Each payload size is a separate ping command.
6.) In order to pass Step A a reply must be received for every ping sent, without losing a single packet

Informative test: ping every HCA except localhost with packet size of 65507

Step B

1.) Physically disconnect a single RCA in the topology (node0).
2.) Check for ping response from all nodes (all-to-all ping test).
3.) A response from the disconnected RCA (node0) should not be returned.
4.) Physically disconnect a second RCA from the topology (node1).
5.) Ping the newly disconnected node (node1) from all other nodes.
    a. A response from the disconnected RCA (node 1) should ***not*** be returned.
6.) Reconnect node0 to the fabric and check for ping responses from all other RCAs in the fabric.
    a. A response from the reconnected RCA (node0) should be returned.
7.) Connect node1 to a different switch in the topology.
8.) Ping node1 from all other nodes.
    a. A response from the reconnected RCA (node 1) should be returned.
9.) Repeat step B, this time by logically disabling the interfaces using `ifdown eth2` instead of physically disconnecting the nodes. There should be no change in behavior from steps 1-7.

Note: Steps A and B must pass in order for the device to pass 12.4 overall.

12.4.2 SFTP Procedure

A 4MB file will be SFTP'd from the DUT to a link partner, and then back to the DUT. The file will then be compared to the original as fidelity verification. SFTP procedures require that SFTP server and client programs be installed on all hosts in the fabric.

Setup

1.) Ensure that sftp is installed and running on all nodes in the topology.

Procedure

1.) Create a 4MB file on the host to test.
    a. “$ dd if=/dev/urandom of=temp_file.orig bs=4194304 count=1”
2.) Put the file in the /tmp directory of the destination host.
    a. “$ echo "put temp_file.orig /tmp/temp_file" > put.in”
    b. “$ sftp -b put.in <destination address>”
3.) Get the same file back to the DUT.
    a. “$ echo "get /tmp/temp_file /tmp/<client>.temp_file" > get.in”
    b. “$ sftp -b get.in <destination address>”
4.) Compare the files via `cmp temp_file.orig /tmp/<client>.temp_file`.
    a. The files must be identical

Notes: Every node must correctly SFTP the 4MB file to all other nodes in order for the device to pass 12.4.2 overall. Sections 12.4.3 and 12.4.4 must pass using the configuration determined by the setup section in 12.4.1 in order for the device to pass IPoCE overall.

12.5 RoCE InfiniBand Gateway

(Note: This test will be developed for a future version of the test plan.)

12.6 RoCE Fibre Channel Gateway

(Note: This test will be developed for a future version of the test plan.)

13 Transport Independent Interop Procedures using OFED

13.1 TI iSER using OFED

IB Setup
Connect initiator/target to switch as well as run one or more SMs (embedded in the switch or host based). If more than one SM, let the SMs split into master and slave.

Optional: In the procedures below, an IB analyzer can be inserted in the appropriate link to obtain traces and validate the aspects of the procedures specifically detailed below in subsequent sections.

iWARP Setup
Connect iSER host initiator and target RNICs to an 10GbE switch.

RoCE Setup
Connect iSER host initiator and target RCA to a 10/40 GbE switch which is DCB Enabled.

Procedure

1.) Load iSER target and iSER initiator to hosts from OpenFabrics tree, check iSER connection.
2.) Run basic dd application from iSER initiator host connected to target.
3.) [IB Specific Test] Run basic dd application from iSER initiator host connected to target. Kill the master SM while test is running and check that it completes properly.
4.) Unload iSER initiator from a Host and check iSER connection properly disconnected on a target host.
5.) Unload iSER target from a Host and check iSER connection properly disconnected on an initiator host.
6.) [IB Specific Test] Repeat steps 2-5 now with the previous slave SM (we did not actually stop the target).

13.2 TI NFS over RDMA

NFS over RDMA Test Procedure

1.) **Note**: This step is for IB Only - the rest of the steps apply to all transports.)
   a. Start a Subnet Manager
2.) Server setup
   a. Add nfs rdma server support to the running kernel if not already present.
      i. "# modprobe svcrdma"
   b. Clean up any existing mount paths and create a new one
      i. "# rm -rf /tmp/nfsordma"
      ii. "$ mkdir -p /tmp/nfsordma/srv"
      iii. "$ chmod -R 777 /tmp/nfsordma/srv"
   c. Start the server
      i. "# /etc/init.d/nfs restart"

      or

      ii. "# service nfs restart"
   d. Tell the server to listen for rdma connection requests on port 20049
      i. "# echo 'rdma 20049' | tee -a /proc/fs/nfsd/portlist"
3.) Client setup
   a. Add nfs rdma client support to the running kernel if not already present.
      i. "# modprobe xprtrdma"
   b. Mount the servers export using rdma
      i. "$ mkdir -p /tmp/nfsordma/<server>"
      ii. "# mount -t nfs <server>-<interface>:/tmp/nfsordma/srv /tmp/nfsordma/<server> -o rdma,port=20049"
   c. Verify that the mount is using the rdma protocol
      i. "$ cat /proc/mounts | grep rdma" (and check the "proto" field for the given mount.)
   d. Set the NFSTESTDIR
      i. "$ export NFSTESTDIR=/tmp/nfsordma/<server>/<client>
4.) Obtain the updated Connectathon2 library from UNH-IOL which includes the updated /general tests which are required to run in Scientific Linux 7.
   a. From the main Connectathon2 directory
      i. "$ cd basic && ./runtests"
      ii. "$ cd ../general && ./runtests"
      iii. "$ cd ../lock && ./runtests"
5.) Repeat steps 2-4 using a new client-server pair until all nodes have acted as both a server and a client.
6.) Repeat steps 2-5 using a new SM until all registered SM's have been used.
7.) All tests run by the connectathon runtests binary must pass on all client nodes rdma mount points from all server nodes using all SM's in order for the device to pass **NFS over RDMA test procedure** over all.

13.3 TI Reliable Datagram Service (RDS) using OFED

13.3.1 RDS-Ping Procedure

Note: RDS does not support iWARP

1.) Use the command "# modprobe rds_rdma" to add RDS support to the kernel
2.) Verify that the kernel supports RDS by issuing the "$ rds-info" command.
  a. The rds-info utility presents various sources of information that the RDS kernel module maintains. When run without any optional arguments rds-info will output all the information it knows of.

Note: Package rds-tools 1.4.1-OFED-1.4.2-1 is required to run rds-info on Ubuntu. Also the rdstcp module needs to be loaded – "# modprobe rds-tcp"

3.) [**For IB**] Start one of the Subnet Managers in the cluster

Note: RDS is IP based and a host address needs to be provided either through an out of band Ethernet connection or through IPoIB. RDS also requires the LIDs to be set in an InfiniBand Fabric and therefore an SM must be run.

Note: All SMs in the fabric should be tested.

4.) Choose a host and use "$ rds-pinghost" to communicate with every other end point in the fabric.

  Note: Be sure that the correct host is identified when using the command *rds-ping host*.

  a. "$ rds-ping" is used to test whether a remote node is reachable over RDS. Its interface is designed to operate in a similar way to the standard ping(8) utility, even though the way it works is pretty different.
  b. "$ rds-ping" opens several RDS sockets and sends packets to port 0 on the indicated host. This is a special port number to which no socket is bound; instead, the kernel processes incoming packets and responds to them.

5.) Verify that all nodes respond without error.

  Note: To avoid losing packets, do not run this while RDS-Stress is running.

13.3.2 RDS-Stress Procedure

1.) Choose a host and start a passive receiving session for the RDS Stress test. It only needs to be told what port to listen on.
    a. “$ rds-stress -p 4000”
2.) Chose a second host and start an active sending instance giving it the address and port at which it will find a listening passive receiver. In addition, it is given configuration options which both instances will use.
    a. “$ rds-stress -T 5 -s recvhost -p 4000 -t 1 -d 1”

Note: If repeating the test in less than one minute, the error message "Cannot assign requested address" may show since the port numbers are not immediately reusable. Either wait or change the port number using the *-p* option

Note: The *-t* option is for the number of tasks (child processes), which defaults to 1 so "-t 1" is optional. The *-d* option is for the message queue depth, which also defaults to 1 so "-d 1" is optional.

3.) Every second, the parent process will display statistics of the ongoing stress test. If the -T option is given, the test will terminate after the specified time and a summary is printed.
4.) Verify that the test completes without error.
5.) Repeat steps 1-4 until all end points in the cluster have been tested.

13.4 TI uDAPLTEST Commands using OFED

(**Note:** Server Command is “$ dapltest -T S -D <ia_name>”)

Setup

1.) The /etc/dat.conf needs to be verified to be sure that the correct interface is used. By default the dapl interface for IB is ib0 and for iWARP is eth2. If these are not correct for the current cluster then errors will occur.
2.) It is also important to verify that the desired dapl library is being used.
3.) **[For IB]** an SM needs to be running.
4.) **[For iWARP hosts with Chelsio RNICs]** Ensure that /sys/module/iw_cxgb3/parameters/peer2peer contains '1' on all hosts.

Group 1: Point-to-Point Topology

1.) Connection and simple send/recv:
    a. “$ dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 1 -R BE client SR 256 1 server SR 256 1”
2.) Verification, polling, and scatter gather list:
    a. “$ dapltest -T T -s <sever_name> -D <ia_name> -i 100 -t 1 -w 1 -V -P -R BE client SR 1024 3 -f server SR 1536 2 -f”

Group 2: Switched Topology

InfiniBand Switch:	Any InfiniBand switch
iWARP Switch:	10 GbE Switch
RoCE Switch:	10/40 GbE DCB Enabled switch

1.) Verification and private data:
    a. “$ dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 1 -V -P -R BE client SR 1024 1 server SR 1024 1”
2.) Add multiple endpoints, polling, and scatter gather list:
    a. “$ dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 10 -V -P -R BE client SR 1024 3 server SR 1536 2”
3.) Add RDMA Write:
    a. “$ dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 1 -V -P -R BE client SR 256 1 server RW 4096 1 server SR 256 1”
4.) Add RDMA Read:
    a. “$ dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 1 -V -P -R BE client SR 256 1 server RR 4096 1 server SR 256 1”

Group 3: Switched Topology with Multiple Switches
(**Note**: This test is **not applicable to RoCE** for the October 2012 Events)

1.) Multiple threads, RDMA Read, and RDMA Write:
   a. “$ dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 4 -w 8 -V -P -R BE client SR 256 1 server RR 4096 1 server SR 256 1 client SR 256 1 server RW 4096 1 server SR 256 1”
2.) Pipeline test with RDMA Write and scatter gather list:
   a. “$ dapltest -T P -s <server_name> -D <ia_name> -i 1024 -p 64 -m p RW 8192 2”
3.) Pipeline with RDMA Read:
   a. **InfiniBand**: “$ dapltest -T P -s <server_name> -D <ia_name> -i 1024 -p 64 -m p RR 4096”
   b. **iWARP**: “$ dapltest -T P -s <server_name> -D <ia_name> -i 1024 -p 64 -m p RR 4096 1”
4.) Multiple switches:
   a. “$ dapltest -T T -s <server_name> -D <ia_name> -i 100 -t 1 -w 10 -V -P -R BE client SR 1024 3 server SR 1536 2”

13.5 TI RDMA BASIC Interop

Purpose

To demonstrate the ability of endpoints to exchange core RDMA operations across a simple network path. This test procedure validates the operation of endpoints at the RDMA level, in a simple network configuration.

The Basic RDMA interop test identifies interoperability issues in one of four ways:

- The inability to establish connections between endpoints
- The failure of RDMA operations to complete
- Incorrect data after the completion of RDMA exchanges
- Inconsistent performance levels.

General Setup

The RDMA interop procedure can be carried out using the OFA Verbs API to create RDMA Connections and send RDMA operation. A flow control mechanism must be agreed upon prior to the start of testing, and configured appropriately. In the future, a flow control mechanism may be mandated.

Topology

Connect topology as shown in topology diagram for test event.

IB Setup

Connect endpoints to switch and run one or more SMs (embedded in the switch or host based).

iWARP Setup

Connect iWARP RDMA endpoints to a 10 or 40 GbE switch.

RoCE Setup

Connect RoCE RCAs to a 10 or 40 GbE switch which is DCB Enabled.

RDMA Connectivity Setup

Each of the tests described below must be run twice with Host A being the server and then Host B being the server. This ensures that the different semantics associated with active and passive sides of the connection are exercised. This way each RDMA interface tested will be sending RDMA data (Requestor) in one test and receiving RDMA data (Target) in the next.

Small RDMA READ Procedure

Note: When testing *between vendors*, the “***-o #*** ” (max 32) flag may be needed to sync the number of outstanding reads.

1.) Select the two devices that will be tested:
2.) On the server device issue the following command on command line:
    a. [**For IB & RoCE**] “$ ib_read_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048 -F”
    b. [**For iWARP**] “$ ib_read_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048 -R -x 0 ***-o 4*** -F”
3.) On the client device issue the following command on command line:
    c. [**For IB & RoCE**] “$ ib_read_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048 <Server IP Address> -F”
    d. [**For iWARP**] “$ ib_read_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048 -R -x 0 ***-o 4*** <RNIC_IP_Address> -F”
4.) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

Large RDMA READ Procedure

Note: When testing *between vendors*, the “ ***-o #*** ” (max 32) flag may be needed to sync the number of outstanding reads.

1.) Select the two devices that will be tested:
2.) On the server device issue the following command on command line:
    a. [**For IB & RoCE**] “$ ib_read_bw -d <dev_name> -i <port> -s 1000000 -n 300 -m 2048 -F”
    b. [**For iWARP**] “$ ib_read_bw -d <dev_name> -i <port> -s 1000000 -n 300 -m 2048 -R -x 0 ***-o 4*** -F”
3.) On the client device issue the following command on command line:
    a. [**For IB & RoCE**] “$ ib_read_bw -d <dev_name> -i <port>-s 1000000 -n 300 -m 2048 <Server IP Address> -F”
    b. [**For iWARP**] “$ ib_read_bw -d <dev_name> -i <port>-s 1000000 -n 300 -m 2048 -R -x 0 ***-o 4*** <RNIC IP Address> -F”
4.) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

Small RDMA Write Procedure

1.) Select the two devices that will be tested:
2.) On the server device issue the following command on command line:
   a. [**For IB & RoCE**] “$ ib_write_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048 -F”
   b. [**For iWARP**] “$ ib_write_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048 -R -x 0 -F”
3.) On the client device issue the following command on command line:
   a. [**For IB & RoCE**] “$ ib_write_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048 <Server IP Address> -F”
   b. [**For iWARP**] “$ ib_write_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048 -R -x 0 <RNIC IP Address> -F”
4.) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

Large RDMA Write Procedure

1.) Select the two devices that will be tested:
2.) On the server device issue the following command on command line:
   a. [**For IB & RoCE**] “$ ib_write_bw -d <dev_name> -i <port> -s 1000000 -n 300 -m 2048 -F”
   b. [**For iWARP**] “$ ib_write_bw -d <dev_name> -i <port> -s 1000000 -n 300 -m 2048 -R -x 0 -F”
3.) On the client device issue the following command on command line:
   a. [**For IB & RoCE**] “$ ib_write_bw -d <dev_name> -i <port> -s 1000000 -n 300 -m 2048 <Server IP Address> -F”
   b. [**For iWARP**] “$ ib_write_bw -d <dev_name> -i <port>-s 1000000 -n 300 -m 2048 -R -x 0 <RNIC IP Address> -F”
4.) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

Small RDMA SEND Procedure

1.) This procedure may fail due to the inability of an endpoint to repost the consumed buffers.
2.) Select the two devices that will be tested:
3.) On the server device issue the following command on command line:
   a. [**For IB & RoCE**] “$ ib_send_bw -d <dev_name> -i <port> -s 1 -n 25000-m 2048 -F”
   b. [**For iWARP**] “$ ib_send_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048 -R -x 0 -r 510 -F”
4.) On the client device issue the following command on command line:
   a. [**For IB & RoCE**] “$ ib_send_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048 <Server IP Address> -F”
   b. [**For iWARP**] “$ ib_send_bw -d <dev_name> -i <port> -s 1 -n 25000 -m 2048 -R -x 0 -r 510 <RNIC IP Address> -F”
5.) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

Large RDMA SEND Procedure

This procedure may fail due to the inability of a endpoint to repost the consumed buffers.

1.) Select the two devices that will be tested:
2.) On the server device issue the following command on command line:
    a. [**For IB & RoCE**] “$ ib_send_bw -d <dev_name> -i <port> -s 1000000 -n 300 -m 2048 -F”
    b. [**For iWARP**] “$ ib_send_bw -d <dev_name> -i <port> -s 1000000 -n 300 -m 2048 -R -x 0 -F”
3.) On the client device issue the following command on command line:
    a. [**For IB & RoCE**] “$ ib_send_bw -d <dev_name> -i <port>-s 1000000 -n 300 -m 2048 <Server IP Address> -F”
    b. [**For iWARP**] “$ ib_send_bw -d <dev_name> -i <port>-s 1000000 -n 300 -m 2048 -R -x 0 <RNIC IP Address> -F”
4.) Verify that the operation completed without error and the level of performance achieved is reasonable and as expected.

Additional IB Notes

1.) Alternate read commands available
   a. Server command: “$ ib_read_bw -m 2048”
   b. Client command (small): “$ ib_read_bw -s 1 -n 25000 IPoIB Address for server -m 2048”
   c. Client command (large):”$ ib_read_bw -s 1000000 -n 300 IPoIB Address for server -m 2048”

2.) Alternate write commands available
   a. Server command: “$ ib_write_bw -m 2048”
   b. Client command (small): “$ ib_write_bw -s 1 -n 25000 IPoIB Address for server”
   c. Client command (large): “$ ib_write_bw -s 1000000 -n 300 IPoIB Address for server -m 2048”

3.) Alternate send commands available
   a. Server command: “$ ib_send_bw -m 2048”
   b. Client command: “$ ib_send_bw -s 1 -n 25000 IPoIB Address for server -m 2048”
   c. Client command (large): “$ ib_send_bw -s 1000000 -n 300 IPoIB Address for server -m 2048”

4.) Explanation of parameters
   a. "-d" allows specifying the device name which may be obtained from the command line: ibv_devinfo
   b. "-i" allows specifying the port number. This may be useful when are running the tests consecutively because a port number is not immediately released, and this will allow specifying another port number to run the test.
   c. "-m" - this specifies the IB PMTU size. As of 10/3/2011 some devices did not support greater than 2048
   d. "-n" - this it the number of operations to complete.
   e. "-R" - Connect QPs with rdma_cm and run test on those QPs
   f. "-s" - this is the size of the operation to complete
   g. "-t" - Size of tx queue (default 128)
   h. "-x 0" - Test uses GID with GID index (Default: IB - no GID. ETH - 0)

IB Example:
DevInfo - Server

```
hca_id:                 mthca0
        fw_ver:                 1.2.0
        node_guid:              0002:c902:0020:b4dc
        sys_image_guid:         0002:c902:0020:b4df
        vendor_id:              0x02c9
        vendor_part_id:         25204
        hw_ver:                 0xA0
        board_id:               MT_0230000001
        phys_port_cnt:          1
                port:                   1
                        state:                  PORT_ACTIVE (4)
                        max_mtu:                2048 (4)
                        active_mtu:             2048 (4)
                        sm_lid:                 1
                        port_lid:               2
                        port_lmc:               0x00
```

Command Line: “$ ib_read_bw -d mthca0 -i 1”

DevInfo - Client

```
hca_id:                 mlx4_0
        fw_ver:                 2.2.238
        node_guid:              0002:c903:0000:1894
        sys_image_guid:         0002:c903:0000:1897
        vendor_id:              0x02c9
        vendor_part_id:          25418
        hw_ver:                 0xA0
        board_id:                MT_04A0110002
        phys_port_cnt:           2
                port:                   1
                        state:                  PORT_ACTIVE (4)
                        max_mtu                 2048 (4)
                        active_mtu:             2048 (4)
                        sm_lid:                 1
                        port_lid:               1
                        port_lmc:               0x00
```

Command Line: ib_send_bw -d mlx4_0 -i 1 10.0.0.1 -s 1 -n 300

13.6 TI RDMA Stress Test

Note: This test cannot be run on Ubuntu 12-4 or 12-10 Server due to the lack of supported packages for Ubuntu

Purpose

This test is designed to identify problems that arise when RDMA operations are performed over interconnection devices in the fabric. The test is not designed to measure the forwarding rate or switching capacity of a device, but does use performance measures to identify failures.

Test failures are identified by the following events:

- The inability to establish connections between endpoints
- The failure of RDMA operations to complete
- Incorrect data after the completion of RDMA exchanges
- Inconsistent performance levels.

Topology

This test does not define a detailed topology and can be used either on a single switch or across a RDMA fabric that may include gateways to and from other technologies. The test configuration depends on the number of endpoints available to perform the testing.

Switch Load

The switch load test validates proper operation of a switch when processing a large number of small RDMA frames. This test is analogous to normal switch testing.

1.) Attach a device to each port on the switch.
2.) Select two ports on the switch to test (This will be the control stream)
3.) Generate RDMA WRITE Operations of size 1024 bytes 100, 000 times on each device by issuing the following commands
    a. On the server device issue the following command on command line:
        i. [**For IB & RoCE**] “$ ib_write_bw -d <dev_name> -i <port>-s 1024 -m 2048”
        ii. [**For iWARP**] “$ ib_write_bw -d <dev_name> -i <port>-s 1024 -m 2048 -R -x 0”
    b. On the client device issue the following command on command line:
        i. [**For IB & RoCE**] “$ ib_write_bw -d <dev_name> -i <port> -s 1024 -n 25000 -m 2048 <Server IP Address>”
        ii. [**For iWARP**] “$ ib_write_bw -d <dev_name> -i <port> -s 1024 -n 25000 -m 2048 -R -x 0 <RNIC IP Address>”

(**Note:** This must be done on both devices at the same time.)

4.) On all other pairs generate RDMA WRITE Operations of size 1 byte continuously until the control stream completes.
5.) Repeat above steps until all port pairs are tested.
6.) Repeat the above steps with all endpoint pairs, except the control stream changed such that the size of the RDMA WRITE operation is 1,000,000 bytes (~1 MB)

Switch FAN in

The switch fan in test attempts to validate proper operation of RDMA exchanges in the presence of traffic loads that exceed the forwarding capacity of the switch. The test requires a minimum of two switches that are interconnected by one port pair.

1.) Connect all possible endpoint pairs such that data exchanges between pairs must traverse the pair of ports interconnecting the switch. The control connections must be across the interconnect network.
2.) Select two ports such that it has to cross both switches. (This will be the control stream)
3.) Generate RDMA WRITE Operations of size 1024 bytes 25,000 times on each device by issuing the following commands
   a. On the server device issue the following command on command line:
      i. [**For IB & RoCE**] “$ ib_write_bw -d <dev_name> -i <port>-s 1024 -m 2048”
      ii. [**For iWARP**] “$ ib_write_bw -d <dev_name> -i <port>-s 1024 -m 2048 -R -x 0”
   b. On the client device issue the following command on command line:
      i. [**For IB & RoCE**] “$ ib_write_bw -d <dev_name> -i <port> -s 1024 -n 25000 -m 2048 <Server IP Address>”
      ii. [**For iWARP**] “$ ib_write_bw -d <dev_name> -i <port> -s 1024 -n 25000 -m 2048 -R -x 0 <RNIC IP Address>”
4.) This must be done on both devices at the same time.
5.) On all other pairs generate RDMA WRITE Operations of size 1 byte continuously until the control stream completes.
6.) Repeat above steps until all port pairs are tested.
7.) Repeat the above steps with all endpoint pairs, except the control stream changed such that the size of the RDMA WRITE operation is 1,000,000 bytes (~1 MB)

RoCE Simultaneous Stress Test

Stress the RoCE Adapter (RCA) by simultaneously transmitting both RoCE/IB traffic and IP level Ethernet traffic.

1.) Establish two connections using a two RoCE adapters with two ports on each adapter. The connections may use a switch but can also be connected directly.
    a. Channel 1 should be established and configured as an Ethernet connection.
    b. Channel 2 should be established and configured as an RDMA over Converged Ethernet connection.
2.) Simultaneously transmit traffic on both channels
    a. Transmit Ethernet traffic on Channel 1 using uperf (www.uperf.org).
    b. Transmit RDMA traffic on Channel 2 using dapltest or the OFED utilities such as ib_write_bw.
3.) Validate that the RCA is able to sustain traffic on both channels such that the traffic on one channel does not interfere with the traffic on the other channel.

13.7 TI RSockets using OFED

General Setup

The librspreload.so library should be loaded prior to running the tests and then removed after all tests are run. Please complete the following steps:

1.) "$ echo $LD_PRELOAD"
    a. This will show what the system already has loaded.
2.) "$ export LD_PRELOAD=$LD_PRELOAD:/usr/lib64/rsocket/librspreload.so"
    a. This will add the rsocket library to the LD_PRELOAD variable.
3.) "$ echo $LD_PRELOAD"
    a. This verifies that the library was added. The result should include but may not be limited to the following:
    b. :/usr/lib64/rsocket/librspreload.so
4.) To remove the setting for the LD_PRELOAD variable, give one of these commands:
    a. "$ export LD_PRELOAD="
    b. "$ unset LD_PRELOAD"

Note: if the first command indicated that there was already a PRELOAD before you added the rsocket PRELOAD, then you should restore that PRELOAD.

Ethernet Setup

Connect RSocket Channel Adapters to a 10 or 40 GbE switch.

IB Setup

Connect the HCAs and switches as per the Fabric Diagram and make sure that no SM is running on the Fabric (verify using the command sminfo). Start OpenSM on one of the HCAs

Installation Requirements

Make sure that the OFA ULP "RSockets" is installed on all nodes.

RSockets Test procedure

There are five separate test procedures to be run on each client/server pair. All nodes should be tested.

Socket Procedure

1.) Start an rstream server on a system:
   a. "$ rstream -T s -S all"
2.) For each client, run socket tests for all sizes
   a. "$ rstream -s <server-ip-address> -T s -S all"

Asynchronous Procedure

1.) Start an rstream server on a system:
   a. "$ rstream -T a -S all"
2.) For each client, run socket tests for all sizes
   a. "$ rstream -s <server-ip-address> -T a -S all"

Blocking Procedure

1.) Start an rstream server on a system:
   a. "$ rstream -T b -S all"
2.) For each client, run socket tests for all sizes
   a. "$ rstream -s <server-ip-address> -T b -S all"

Non-blocking Procedure

1.) Start an rstream server on a system:
   a. "$ rstream -T n -S all"
2.) For each client, run socket tests for all sizes
   a. "$ rstream -s <server-ip-address> -T n -S all"

Verified Transfers Procedure

1.) Start an rstream server on a system:
   a. "$ rstream -T v -S all"
2.) For each client, run socket tests for all sizes
   a. "$ rstream -s <server-ip-address> -T v -S all"

13.8 TI MPI - Open MPI using OFED

The following values are used in examples below:

- $MPIHOME: The absolute directory location of the Open MPI installation that is common to all systems under test.
- $NP: The number of MPI processes to use in the test.
- $HOSTFILE: The absolute filename location of the hostfile
- $IMBHOME: The absolute directory location of the Intel MPI Benchmark (IMB) tools installation that is common to all systems under test.

Cluster setup

1.) Network configuration requirements
   a. All systems must be reachable by each other over IP (IPoIB for IB).
   b. All nodes must agree on the IPoIB IP addresses of all systems (e.g., via /etc/hosts, DNS, or some other mechanism).
2.) The same version of OFED must be installed in the same filesystem location on all systems under test.
3.) The same version of the Intel MPI Benchmark (IMB) tools must be installed in the same filesystem location on all systems under test.
   a. IMB can be used from the OFED installation or, if a later version of Open MPI is to be used, IMB can be downloaded from Intel's web site (with the “Accept” button at the bottom of the page):

      http://software.intel.com/en-us/articles/intel-mpi-benchmarks/?wapkw=intel%20mpi%20benchmarks

4.) The same version of Open MPI must be available in the same file system location on all systems under test.
   a. Open MPI can be used from the OFED installation, or, if a later version is required, can be downloaded and installed from the main Open MPI web site:

      http://www.open-mpi.org/

      i. If building Open MPI from source, and if the OpenFabrics libraries and headers are installed in a non-default location, be sure to use the --with-openib=<dir> option to configure to specify the OpenFabrics filesystem location.
      ii. Open MPI can be installed once on a shared network filesystem that is available on all nodes, or can be individually installed on all systems. The main requirement is that Open MPI's filesystem location is the same on all systems under test.

iii. If Open MPI is built from source, the --prefix value given to configure should be the filesystem location that is common on all systems under test. For example, if installing to a network filesystem on the filesystem server, be sure to specify the filesystem location under the common mount point, not the "native" disk location that is only valid on the file server.

Note that Open MPI is included in some Linux distributions and other operating systems. Multiple versions of Open MPI can peacefully co-exist on a system as long as they are installed into separate filesystem locations (i.e., configured with a different --prefix argument). All MPI tests must be built and run with a single installation of Open MPI.

iv. Ensure that the Open MPI installation includes OpenFabrics support:
$MPIHOME/bin/ompi_info | *grep openib*

Example output should look like this:
MCA btl: openib (MCA v1.0, API v1.0.1, Component v1.4)

Note: The exact version numbers displayed will vary depending on your version of Open MPI.
The important part is that a single "btl" line appears showing the openib component.

b. Basic Open MPI run-time functionality can first be verified by running simple non-MPI applications. This ensures that the test user's rsh and/or ssh settings are correct, etc. The following example uses the ‘hostname’ command:
$MPIHOME/bin/mpirun -np $NP --hostfile $HOSTFILE hostname

Note: The output should show the hostname of each host listed in the hostfile; the hostname should appear as many times as there are lines in the hostfile. The list of hostnames may appear in random order; this is normal**.** Any serial application can be run; "hostname" is a good, short test that clearly identifies specific hosts were used. Verify the version of mpi installed by doing “$ mpirun -version” just as a sanity check.

5.) All systems must be setup with at least one identical user account. This user must be able to SSH or RSH to all systems under test from the system that will launch the Open MPI tests with no additional output to stdout or stderr (e.g., all SSH host keys should already be cached, no password/passphrase prompts should be emitted, etc.).
6.) The lockable memory limits on each machine should be set to allow unlimited locked memory per process.
7.) The underlying OpenFabrics network used in the test should be stable and reliable.
8.) No other fabric interoperability tests should be running during the Open MPI tests.
9.) Whenever possible the MPI tests should be run across 5 or more separate systems to stress the OpenFabrics network. If only one single system is available, one can run in loopback mode with the addition of '--mca btl openib,self' to the mpirun command.

Install Open MPI for OFED 3.5 and Later

1.) Download the latest stable version of Open MPI here:
http://www.openmpi.org/software/ompi/
   a. Change to the directory where the tar ball was unpacked
   b. Invoke the command:
   ./configure --prefix=/usr/local/openmpi -enable-orterun-prefix-by-default

Note: you can build without the prefix because the default directory is /usr/local. But the OFA Cluster at UNH-IOL uses /tmp because there is no write access to /usr/local and therefore the prefix is needed.

   c. Invoke the command:
   make all install

Note: OFA Cluster at UNH-IOL requires # *make all install*

2.) Now you must build IMB-MPI1: you can download it here:
http://software.intel.com/en-us/articles/intel-mpi-benchmarks/
   a. Unpack the IMB tar file and cd to the unpacked directory and go to the subdirectory 'src'
   b. Open the make_ict file and change ‘CC = mpiicc’ to ‘CC = mpicc’ or alternately, run this command:
   sed -i -e ‘s/mpiicc/mpicc/g’ make_ict
   c. While still in the 'src' directory, invoke: *make all*
   d. Copy IMB-MPI1 which has just been built to the directory /usr/local/bin

Note: The OFA Cluster at UNH-IOL copies IMB-MPI1 to /usr/local/openmpi/bin rather than the default location and requires super user privileges to change to this directory

Configuring and building Open MPI 1.8.X for PowerLinux systems

These are the instructions for configuring and building Open MPI on a Power Linux system.

1.) From the command line:
```
./configure --prefix=/usr/local/openmpi-1.8.X
--with-platform=../../contrib/platform/ibm/optimized-ppc64-gcc && make -j 16 &&
make install
```

Note: The setting of the prefix will depend on where you usually do the installs. With RHEL6.x, we typically use the module command from the environment-modules rpm to dynamically adjust the environment to pick up a specific MPI build. But, mpi-selector will work as well. Also, the value given to -j on the make is dependent on the number of available cores. For example, on the P7 system at UNH-IOL, 16 should work fine.

Configuring and building Open MPI 1.8.X for PowerLinux systems (continued)

2.) The main specifications are in the associated platform file. The one item of import for IB/RoCE testing is the line: ***with_verbs=/usr*** **(***with_openib=/usr* in Open MPI 1.6X) This will ensure that the IB transport is supported. It also assumes that the RDMA stack is installed in the standard place. If not, then this parameter will have to be adjusted accordingly.

- enable_mem_debug=no
- enable_mem_profile=no
- enable_debug=no
- enable_contrib_no_build=libnbc,vt
- enable_ft_thread=no
- with_verbs=/usr
- enable_shared=yes
- enable_static=no
- CXXFLAGS=-m64
- CCASFLAGS=-m64
- FCFLAGS=-m64
- FFLAGS=-m64
- CFLAGS=-m64
- with_wrapper_cflags=-m64
- with_wrapper_cxxflags=-m64
- with_wrapper_fflags=-m64
- with_wrapper_fcflags=-m64

3.) MPI Executable

a. *mpirun --bind-to-core -np 16 --host <sys1>,<sys2> ./IMB-MPI1*

b. The MPI executables should find the RDMA adapters and then figure out the appropriate connection method. The --bind-to-core can improve performance, but is certainly optional.

Test Setup

1.) Create a hostfile ($HOSTFILE) listing the hostname of each system that will be used in the test. If a system under test can run more than one MPI process (such as multiprocessor or multicore systems), list the hostname as many times as MPI processes are desired. For example, for two systems named node1.example.com and node2.example.com that are each able to run 4 processes:

```
$ cat hostfile.txt
node1.example.com
node1.example.com
node1.example.com
node1.example.com
node2.example.com
node2.example.com
node2.example.com
node2.example.com
```

2.) Determine the number of Open MPI processes ($NP) that are to be run determined by the number of host entries in the created hostfile.

3.) Open MPI defaults to probing all available networks at run-time to determine which to use. The mpirun parameter, --mca btl openib,self, willl force all traffic over the RDMA fabric for iWARP, InfiniBand, and RoCE. Also, it means that processes on the same system will use the OFA stack for communication rather than shared memory. This is also how you do "loopback" to force the use of an RDMA adapter on a single system. For OpenFabrics testing for iWarp, InfiniBand and RoCE, add this extra command line parameter.

   --mca btl openib,self

4.) It has been discovered that the following Open MPI command line parameter is required to facilitate multi RDMA adaptor vendor MPI rings; both iWarp and InfiniBand:

   --mca pml ob1 --mca btl_openib_flags 306

5.) It has been discovered that the following Open MPI command line parameter is required to facilitate multi RNIC adaptor vendors MPI rings; iWarp specific:

   --mca btl_openib_receive_queues P,65536,256,192,128

Test Procedure

1.) Create a hostfile listing the MPI ring nodes, process distribution, and total number of processes to use as indicated in steps 1 and 2 of section 13.8.4. The filesystem location of this hostfile is irrelevant.
2.) Locate the "mpirun" binary that will be used. This determines the version of Open MPI that will be used.
3.) Locate the "IMB-MPI1" IMB binary. This must have been built against the version of Open MPI selected above. If using an OFED distribution this build process has already been performed.
4.) For **InfiniBand**:
    a. Verify that a subnet manager has configured the fabric. If not, start one.
5.) Verify that all hosts present within the hostfile are online and accessible.
6.) Run the IMB-MPI1 benchmarks
7.) For **InfiniBand**:

Note: All IMB benchmarks must pass successfully using all subnet managers under test in order for the devices under test defined within the hostfile pass.

Method of implementation for all Linux OS's

1.) For **InfiniBand**:
    a. To perform step 4 of section 13.8.5 use ibdiagnet -r from a host defined in the mpi hostfile and look for an SM - Master entry in the output
2.) To perform step 5 of section 13.8.5 ping the IP address (IPoIB for InfiniBand) address of all hosts defined in the mpi hostfile from a host defined in said hostfile.
3.) To perform step 6 of section 13.8.5 use the following command from a host that can access all hosts defined within the hostfile; this host can be part of the hostfile
    a. For **InfiniBand**:
        i. $MPIHOME/bin/mpirun --mca btl openib,self,sm --mca pml ob1 -mca btl_openib_flags 306 --mca btl_openib_cpc_include rdmacm -np $NP -hostfile $HOSTFILE $IMBHOME/IMB-MPI1”
    b. For **iWarp**:
        i. $MPIHOME/bin/mpirun --mca btl openib,self,sm --mca pml ob1 --mca btl_openib_flags 306 --mca btl_openib_receive_queues P,65536,256,192,128 --mca btl_openib_cpc_include rdmacm -np $NP -hostfile $HOSTFILE $IMBHOME/IMB-MPI1
    c. For **RoCE**:
        i. $MPIHOME/bin/mpirun --mca btl openib,self,sm --mca pml ob1 --mca btl_openib_flags 306 --mca btl_openib_receive_queues P,65536,120,64,32 --mca btl_openib_cpc_include rdmacm -np $NP -hostfile $HOSTFILE $IMBHOME/IMB-MPI1
    d. For **PowerLinux Systems:**
        i. $ mpirun --mca btl openib,self --allow-run-as-root --bind-to core:overload-allowed -np 16 --host <sys1>,<sys2> /usr/local/openmpi/bin/IMB-MPI1

14 Bug reporting methodology during pre-testing

The following bug reporting methodology will be followed during the execution of interoperability pre-testing at UNH-IOL.

1.) UNH-IOL and the OEMs (e.g. Chelsio, Data Direct, Intel, NetApp, Mellanox) will assign a focal point of contact to enable fast resolution of problems.
2.) Bug reports will include:
    a. Detailed fail report with all relevant detail (Test/Application, Topology.).
    b. **[For IB]** IB trace if needed.
    c. **[For iWARP]** iWARP, TCP and SCTP traces if needed.
3.) Bug reports will be sent via email by UNH-IOL to the focal point assigned by the OEM
4.) Bug reports and suggested fixes will be sent to the OpenFabrics development community - OFA Bugzilla. When such reports are communicated, UNH-IOL will ensure that confidentiality between UNH-IOL and the OEM will be maintained. Bug reports will be generalized and not include any company specific proprietary information such as product name, software name, version etc.
5.) All bug fixes/issues that are found during testing will be uploaded to the OpenFabrics repository. Documentation related to fixes will not mention any company specific proprietary information.

Note: This test plan does not cover how bugs will be reported by IBTA/CIWG or IETF iWARP during or after interoperability testing at plugfests.

15 Results Summary

15.1 InfiniBand Specific Test Results

Please add a check mark whenever a test case passes and when the system is behaving according to the criteria mentioned below. Otherwise indicate a failure along with a comment explaining the nature of the failure.

IB Link Initialize

Test #	Test	Pass	Fail	Comment
1	Phy link up all ports			

IB Fabric Initialization

Test #	Test	Pass	Fail	Comment
1	Verify that all ports are in Armed or Active state			

IB IPoIB - Connected Mode (CM)

Test #	Test	Pass	Fail	Comment
1	Ping all to all - Ping using SM 1			
2	Ping all to all - Ping using SM 2			
3	Ping all to all - Ping using SM 3			
4	Ping all to all - Ping using SM 4			
5	Ping all to all - Ping using SM 5			

6	Ping all to all - Ping using SM 6			
7	Ping all to all - Ping using SM x			
8	Connect/Disconnect Host			
9	FTP Procedure			

IB IPoIB - Datagram Mode (DM)

Test #	Test	Pass	Fail	Comment
1	Ping all to all - Ping using SM 1			
2	Ping all to all - Ping using SM 2			
3	Ping all to all - Ping using SM 3			
4	Ping all to all - Ping using SM 4			
5	Ping all to all - Ping using SM 5			
6	Ping all to all - Ping using SM 6			
7	Ping all to all - Ping using SM x			
8	Connect/Disconnect Host			
9	FTP Procedure			

IB SM Failover/Handover

Test #	Test	Pass	Fail	Comment
1	Basic sweep test			
2	SM Priority test			
3	Failover test - Disable SM1			
4	Failover test - Disable SM2			

IB SRP

Test #	Test	Pass	Fail	Comment
1	Basic dd application			
2	IB SM kill			

Fibre Channel Gateway - (IB Specific)

Test #	Test	Pass	Fail	Comment
1	Basic Setup			
2	Configure Gateway			
3	Add Storage Device			
4	Basic dd application			
5	IB SM kill			
6	Disconnect Host/Target			

7	Load Host/Target			
8	dd after SRP Host and Target reloaded			
9	Reboot Gateway			
10	dd after FC Gateway reboot			

Ethernet Gateway - (IB Specific)

Test #	Test	Pass	Fail	Comment
1	Basic Setup			
2	Start ULP			
3	Discover Gateway			
4	SM Failover			
5	Ethernet gateway reboot			
6	ULP restart			
7	Unload/load ULP			

15.2 Ethernet Specific Test Results

iWARP Link Initialize

Test #	Test	Pass	Fail	Comment
1	Phy link up all ports			
2	Verify basic IP connectivity			

RoCE Link Initialize

Test #	Test	Pass	Fail	Comment
1	Phy link up all ports			
2	Verify basic IP connectivity			

15.3 Transport Independent Test Results

TI iSER

Test #	Test	Pass	Fail	Comment
1	Basic dd application			
2	IB SM kill			
3	Disconnect Initiator			
4	Disconnect Target			
5	Repeat with previous SM Slave			

TI NFS Over RDMA

Test #	Test	Pass	Fail	Comment
1	File and directory creation			
2	File and directory removal			
3	Lookups across mount point			
4	Setattr, getattr, and lookup			
5	Read and write			
6	Readdir			
7	Link and rename			
8	Symlink and readlink			
9	Statfs			

TI RDS

Test #	Test	Pass	Fail	Comment
1	rds-ping procedure			
2	rds-stress procedure			

TI uDAPL

Test #	Test	Pass	Fail	Comment
1	P2P - Connection & simple send receive			
2	P2P - Verification, polling & scatter gather list			
3	Switched Topology -Verification and private data			
4	Switched Topology - Add multiple endpoints, polling, & scatter gather list			
5	Switched Topology - Add RDMA Write			
6	Switched Topology - Add RDMA Read			
7	Multiple Switches - Multiple threads, RDMA Read, & RDMA Write			
8	Multiple Switches - Pipeline test with RDMA Write & scatter gather list			
9	Multiple Switches - Pipeline with RDMA Read			
10	Multiple Switches - Multiple switches			

TI RDMA Basic Interop

Test #	Test	Pass	Fail	Comment
1	Small RDMA READ			
2	Large RDMA READ			
3	Small RDMA Write			
4	Large RDMA Write			
5	Small RDMA SEND			
6	Large RDMA SEND			
7	Small RDMA Verify			
8	Large RDMA Verify			

TI RDMA Stress Tests

Test #	Test	Pass	Fail	Comment
1	Switch Load			
2	Switch Fan In			

RSockets

Test #	Test	Pass	Fail	Comment
1	Socket calls			
2	Asynchronous calls			
3	Blocking calls			
4	Non-blocking calls			
5	Verified transfers			

15.4 Open MPI Test Results

TI MPI - Open MPI

Test #	Test Suite	Pass	Fail	Comment
Phase 1: "Short" tests				
2	OMPI built with OpenFabrics support			
3	OMPI basic functionality (hostname)			
4.1	Simple MPI functionality (hello_c)			
4.2	Simple MPI functionality (ring_c)			
5	Point-to-point benchmark (NetPIPE)			
6.1.1	Point-to-point benchmark (IMB PingPong multi)			
6.1.2	Point-to-point benchmark (IMB PingPing multi)			

Phase 2: "Long" tests				
6.2.1	Point-to-point benchmark (IMB PingPong)			
6.2.2	Point-to-point benchmark (IMB PingPing)			
6.2.3	Point-to-point benchmark (IMB Sendrecv)			
6.2.4	Point-to-point benchmark (IMB Exchange)			
6.2.5	Collective benchmark (IMB Bcast)			
6.2.6	Collective benchmark (IMB Allgather)			
6.2.7	Collective benchmark (IMB Allgatherv)			
6.2.8	Collective benchmark (IMB Alltoall)			
6.2.9	Collective benchmark (IMB Reduce)			
6.2.10	Collective benchmark (IMB Reduce_scatter)			
6.2.11	Collective benchmark (IMB Allreduce)			
6.2.12	Collective benchmark (IMB Barrier)			
6.3.1	I/O benchmark (IMB S_Write_Indv)			
6.3.2	I/O benchmark (IMB S_IWrite_Indv)			
6.3.3	I/O benchmark (IMB S_Write_Expl)			
6.3.4	I/O benchmark (IMB S_IWrite_Expl)			
6.3.5	I/O benchmark (IMB P_Write_Indv)			
6.3.6	I/O benchmark (IMB P_IWrite_Indv)			

6.3.7	I/O benchmark (IMB P_Write_Shared)			
6.3.8	I/O benchmark (IMB P_IWrite_Shared)			
6.3.9	I/O benchmark (IMB P_Write_Priv)			
6.3.10	I/O benchmark (IMB P_IWrite_Priv)			
6.3.11	I/O benchmark (IMB P_Write_Expl)			
6.3.12	I/O benchmark (IMB P_IWrite_Expl)			
6.3.13	I/O benchmark (IMB C_Write_Indv)			
6.3.14	I/O benchmark (IMB C_IWrite_Indv)			
6.3.15	I/O benchmark (IMB C_Write_Shared)			
6.3.16	I/O benchmark (IMB C_IWrite_Shared)			
6.3.17	I/O benchmark (IMB C_Write_Expl)			
6.3.18	I/O benchmark (IMB C_IWrite_Expl)			
6.3.19	I/O benchmark (IMB S_Read_Indv)			
6.3.20	I/O benchmark (IMB S_IRead_Indv)			
6.3.21	I/O benchmark (IMB S_Read_Expl)			
6.3.22	I/O benchmark (IMB S_IRead_Expl)			
6.3.23	I/O benchmark (IMB P_Read_Indv)			
6.3.24	I/O benchmark (IMB P_IRead_Indv)			
6.3.25	I/O benchmark (IMB P_Read_Shared)			
6.3.26	I/O benchmark (IMB P_IRead_Shared)			
6.3.27	I/O benchmark (IMB P_Read_Priv)			

6.3.28	I/O benchmark (IMB P_IRead_Priv)			
6.3.29	I/O benchmark (IMB P_Read_Expl)			
6.3.30	I/O benchmark (IMB P_IRead_Expl)			
6.3.31	I/O benchmark (IMB C_Read_Indv)			
6.3.32	I/O benchmark (IMB C_IRead_Indv)			
6.3.33	I/O benchmark (IMB C_Read_Shared)			
6.3.34	I/O benchmark (IMB C_IRead_Shared)			
6.3.35	I/O benchmark (IMB C_Read_Expl)			
6.3.36	I/O benchmark (IMB C_IRead_Expl)			
6.3.37	I/O benchmark (IMB Open_Close)			

Remarks

General Remarks: Comments about the set-up, required updates to the TD, and any other issues that came up during the testing.

Test Results Key

Key	Meaning	Interpretation
Pass	Pass	The Device Under Test (DUT) was observed to exhibit compliant behavior.
PWC	Pass with Comments	The Device Under Test (DUT) was observed to exhibit compliant behavior, however changes were made to the normal test procedure or the behavior observed requires additional comments
QP	Qualified Pass	The DUT was generally observed to exhibit compliant behavior; however there are some known issues or defects which are outlined below.
Fail	Fail	The Device Under Test (DUT) was observed to exhibit non-compliant behavior.
Info	Informative Result	This test is designed for informational purposes only. The results may help provide the interoperability information about the DUT to the end user, but the publication of these results is not required. Note: The reporting of Beta test results is optional and at the sole discretion of the vendor.
Warn	Warning	The DUT was observed to exhibit behavior that is not recommended.
N/S	Not Supported	The DUT was not observed to support the necessary functionality required to perform these tests.
N/T	Not Tested	This test was not performed and therefore this is not a complete test report. Please see the comments for additional reasons.
UA	Unavailable	The test was not performed due to limitation of the test tool(s) or interoperable systems, or the test methodology is still under development.
RTC	Refer to Comments	From the observations, a valid pass or fail was not determined. An additional explanation of the situation is included.