NVIDIA InfiniBand Cluster Operation and Maintenance Guide

Table of Contents

You can download a PDF here.

1 About This Document

This document is intended for network operators responsible for maintaining InfiniBand clusters. The purpose of this document is to outline the necessary automation tools, required tests, and essential information needed when accepting a new cluster. Additionally, the document provides recommendations for monitoring and maintenance routines, along with guidance on how to obtain the necessary inputs for these procedures and how to execute the maintenance operations effectively. The document's content is structured logically to facilitate easy reference and understanding.

In addition, this document provides links to documentation describing how to establish connections between network events and how they are reported by NVIDIA UFM (Unified Fabric Manager). The various scenarios have been categorized based on the anticipated likelihood of their occurrence. For each specific issue, a comprehensive set of UFM alerts that signal its presence are listed, along with the UFM settings that need configuring to receive these alerts. Detailed instances of these alerts are presented, accompanied by thorough explanations of their significance.
It is important to note that this document aligns with the software capabilities as of July 2023. It aims to provide network operators with a comprehensive resource to effectively manage and maintain InfiniBand clusters, utilizing the most up-to-date information and practices available.

2 Related Documentation

NVIDIA UFM Enterprise	NVIDIA UFM Enterprise User Manual NVIDIA UFM Enterprise Quick Start Guide NVIDIA UFM Enterprise REST API Guide
NVIDIA UFM Enterprise Appliance	NVIDIA UFM Enterprise Appliance Software User Manual
NVIDIA UFM Telemetry	NVIDIA UFM Telemetry Documentation
NVIDIA UFM High-Availability	NVIDIA UFM High-Availability User Guide

Document Revision History

For the list of changes made to this document, refer to Document Revision History.

3 Prerequisites

This section describes the required tools for executing the InfiniBand cluster maintenance and operational procedures.

1. UFM - July 2023 SW Version: This entails UFM Enterprise and at least one instance of UFM Telemetry. UFM incorporates an embedded UFM Telemetry instance featuring 120 fundamental debug counters for each port. These counters are collected periodically and are, by default, accessible through an HTTP endpoint. UFM offers multiple mechanisms for pushing (streaming) UFM Telemetry and event streams. Additional information can be found in Retrieving UFM Issues for comprehensive insights.
2. UFM Installation: Refer to the installation instructions according to the desired UFM software.

UFM	Link to Installation Instructions
UFM Enterprise	UFM Enterprise Installation
UFM Enterprise Appliance	UFM Enterprise Appliance Software Upgrade
UFM Telemetry	UFM Telemetry Installation
For those opting to use their own server	UFM Installation Steps

4 From Build to Operation

This chapter describes the required procedure to be executed toward the end of cluster bringup phase, just before the cluster operation. It is also recommended to execute this procedure after every maintenance window. That includes files and logs to be reviewed and kept as reference when the cluster is signed off from the build phase to the operation phase and after performing UFM/OpenSM/Firmware upgrade procedure.

This section outlines the necessary steps that need to be taken as part of the final stages of InfiniBand cluster bring up and initialization, just before the cluster becomes operational. It is recommended to follow these steps after each maintenance window as well. These steps encompass the review and retention of files and logs that serve as references when transitioning the cluster from the construction phase to the operational phase. This practice is particularly relevant after carrying out procedures such as UFM/OpenSM/Firmware upgrades.

Please adhere to the following steps:

Step	Item	Direct Link
1	Verify the operational status of the UFM service	UFM Service Verification
2	Generate the fabric health procedure	Fabric Health Report Generation and Validation
3	Confirm the integrity of the cluster's topology	Cluster Topology Validation
4	Validate the proper functioning of UFM telemetry collection	Telemetry Metrics Collection
5	Evaluate the cluster's performance	Cluster Performance Verification
6	Optional: Check for the presence and functionality of necessary UFM plugins	Review All Unhealthy Nodes
7	Run the quarterly maintenance procedure	Quarterly Maintenance
8	Obtain and securely store UFM system dump using the following command: `/usr/bin/ufm_sysdump.sh` Retain this data for future reference	
9	Contact NVIDIA Networking Support or your designated NVIDIA account team to facilitate a review of your **UFM** and **SM** configurations with NVIDIA. Provide the generated ufm_sysdump data created in the previous step for reference.	

5 Cluster Maintenance Procedure

This section describes the minimal set of operations required to monitor the network service and keep it in good health. We provide a sub-section for activities that should be run every couple of minutes, day, and full. We also describe how each activity is done and observed. Automation to query the UFM API may be required as described in List of Scenarios which also includes details on how each type of alert/issue-found should be handled.
This section outlines the minimal set of operations required to monitor and oversee the network service and uphold its optimal functionality. Also provided are tasks intended for regular intervals in minutes, daily, and full cycles. Additionally, this section describes the execution and observation process for each task. It might be necessary to automate UFM API queries.
It is assumed that UFM Enterprise operates continuously, gathering relevant data and generating alerts that necessitate examination. It is important to diligently monitor and address these alerts.
Cluster maintenance can be performed in the following intervals:

- Maintenance at Regular Intervals of Minutes / Ongoing
- Weekly Maintenance
- Quarterly Maintenance

5.1 Maintenance at Regular Intervals of Minutes / Ongoing

Upon verifying that the UFM service is up and running, the following monitoring measures are automatically activated:

- Validate that the event types outlined in List of Scenarios have not occurred.
- In the event of an occurrence, check the debugging and resolution procedures.

For more information, refer to UFM Events Fluent Streaming (EFS) Plugin.

5.2 Weekly Maintenance

- Follow the steps outlined in Maintenance at Regular Intervals of Minutes / Ongoing.
- Monitor trends in link monitoring key indicators. Refer to Link Monitoring Key Indicators.
- Validate the integrity of the Cluster topology as instructed in Cluster Topology Validation.
- Execute Fabric Health Validation tests is instructed in Fabric Health Report Generation and Validation.
- Verify network performance key indicators in accordance with Cluster Performance Verification.
- Perform maintenance for the cooling system: review temperature differentials as detailed in Cooling System Maintenance and address any identified issues as instructed in Inadequate Control of Cluster Temperature.

5.3 Quarterly Maintenance

- Follow the steps outlined in Weekly Maintenance.

- Examine the most recent NVIDIA firmware and software release notes as detailed in . It is recommended to perform regular updates of the cluster software, at a minimum of once per year, aligning with the LTS release schedule.
- Even if a software upgrade cannot be carried out, it is strongly recommended to familiarize yourself with documented known issues that have been resolved through releases Refer to UFM SW Release Notes and User Manual.
- Conduct an annual review of NVIDIA network health – Contact NVIDIA Networking Support or your designated NVIDIA contact.

6 Operations Procedures

This page describes guidelines and working methods for NDR clusters and later.

6.1 UFM Service Verification

Confirm the operational status of the UFM service.

- If you are using UFM Enterprise Appliance, execute the following command via the command-line interface (CLI) after logging into the UFM appliance.

  ```
  show ufm status
  ```

  For more information, refer to UFM General Commands
- If you are using your own server, refer to Showing UFM Processes Status.
- If you prefer using the web user interface:
  - Navigate to the "System Health" tab in the left menu.
  - Under the "UFM Health" section, click on "Create New Report."
  - Confirm that all fields are displaying green indicators.
    For detailed instructions, refer to UFM Health Tab

It is also recommended to conduct a remote test of the REST API by querying the "UFM Health" report. For instructions, refer to Reports REST API.

6.2 Fabric Health Report Generation and Validation

To generate fabric health report and verifying all sections are green, perform the following steps using Web UI:

- Access the "System Health" tab on the left menu
  - Click on "Run New Report" under the "Fabric Health" section
  - Confirm that all fields are indicating green status
  - For detailed instructions, refer Fabric Health Tab
  - Additionally, within the "System Health" tab:
- Run the available tests under "Fabric Validation"
  - Verify the outcomes as either "Pass" or "Completed with No Errors"
  - For detailed instructions, refer Fabric Validation Tab.
  - Furthermore, it is recommended to conduct remote REST API tests from a remote node. This can be done using the REST APIs described in the following links:
- Reports REST API
- Fabric Validation Tests REST API

6.3 Cluster Topology Validation

Once the InfiniBand cluster is built, it is essential to create a Master Topology. This Master Topology serves as a reference during cluster operation, enabling the detection of any network configuration changes. It is noteworthy that the actual cluster topology may be different from the initially planned specifications. Detecting and validating these discrepancies in topology is crucial to ensure

the cluster's proper functionality.
As an example, even in cases where a known TOR switch is defected due to hardware malfunction and is planned for RMA process, the cluster can still operate, albeit with some degradation in performance and anticipated capacity.
For a more comprehensive details, refer to Topology Compare REST API.

6.4 Telemetry Metrics Collection

To collect InfiniBand ports, PHY and cables telemetry metrics, perform the following.
Access the embedded UFM Telemetry instance through an HTTP End Point using the following URL to your browser address bar:
+http://ufm_ip:9001/labels/enterprise+
Please remember to replace your UFM IP according to your IP address, for example:
http://10.209.44.100:9001/labels/enterprise
Expected Results:

- `PortXmitDataExtended{device_name="",device_type="host",fabric="compute",hostname="swx-snap3",level="server",node_desc="swx-snap3 mlx5_0",peer_level="server",port_id="248a0703009a15fa_1"} 228011616 1648987628390`
- `PortRcvDataExtended{device_name="",device_type="host",fabric="compute",hostname="swx-snap3",level="server",node_desc="swx-snap3 mlx5_0",peer_level="server",port_id="248a0703009a15fa_1"} 228011616 1648987628390`
- `PortXmitPktsExtended{device_name="",device_type="host",fabric="compute",hostname="swx-snap3",level="server",node_desc="swx-snap3 mlx5_0",peer_level="server",port_id="248a0703009a15fa_1"} 791707 1648987628390`
- `PortRcvPktsExtended{device_name="",device_type="host",fabric="compute",hostname="swx-snap3",level="server",node_desc="swx-snap3 mlx5_0",peer_level="server",port_id="248a0703009a15fa_1"} 791707 1648987628390`
- `SymbolErrorCounterExtended{device_name="",device_type="host",fabric="compute",hostname="swx-snap3",level="server",node_desc="swx-snap3 mlx5_0",peer_level="server",port_id="248a0703009a15fa_1"} 0 1648987628390`

For a more compact CSV data format, access the following endpoint:
http://ufm_ip:9001/labels/csv/metrics (http://ufm_ip:9001/labels/csv/metrics)
Remember to replace "ufm_ip" with your actual UFM IP address. Example:
http://10.209.44.100:9001/labels/csv/metrics

6.5 Link Monitoring Key Indicators

The following table lists the link monitoring key indicators and provides their descriptions, pass/fail criteria and monitoring intervals.

Parameter	Description	Evaluation Criteria	Monitoring Interval
Link State			
Phy_state	Physical link state	Verify link up	ongoing
Logical_state	Logical link state	Verify link in ACTIVE mode	ongoing
speed_active	Active link speed	Verify expected speed	ongoing
width_active	Active link width	Verify expected width 4x, Or for split cable - 2x	ongoing
Link Quality			

<table>
<tr>
<td>NDR Link Quality</td>
<td>Link Quality criteria depend of error correction scheme type.</td>
<td>
<table>
<tr><th rowspan="2">ErrorCorrectionSchemeTYPE</th><th rowspan="2">MediaType</th><th colspan="3">Post-FEC</th><th colspan="3">Symbol</th></tr>
<tr><th>Normal</th><th>Warning</th><th>Error</th><th>Normal</th><th>Warning</th><th>Error</th></tr>
<tr><td><b><i>Default for DAC/ACC/Active < 100m Low_Latency_RS_FEC_PLR</i></b></td><td>DAC/LACC/Active</td><td>1.00E-12</td><td>5.00E-12</td><td>1.00E-11</td><td>1.00E-15</td><td>5.00E-15</td><td>1.00E-14</td></tr>
</table>
</td>
<td>ongoing</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td><table>
<tr><th rowspan="2">ErrorCorrectionSchemeTYPE</th><th rowspan="2">MediaType</th><th colspan="3">Post-FEC</th><th colspan="3">Symbol</th></tr>
<tr><th>Normal</th><th>Warning</th><th>Error</th><th>Normal</th><th>Warning</th><th>Error</th></tr>
<tr><td><b><i>Default for DAC/ACC/Active > 100m KP4_Standard_RS_FEC</i></b></td><td>Active</td><td>1.00E-15</td><td>5.00E-15</td><td>1.00E-14</td><td>1.00E-15</td><td>5.00E-15</td><td>1.00E-14</td></tr>
</table>
Note: Minimum port up time for BER measurement - 125 minutes.</td><td></td></tr>
<tr><td><b>PHY Errors</b></td><td></td><td></td><td></td></tr>
<tr><td>Symbol_Errors</td><td>Errors after FEC and PLR</td><td>Defined by Symbol BER</td><td>ongoing</td></tr>
</table>

Link_Down counter	Total number of link down occurred as a result of involuntary link shutdown.	If delta from last sample > 0: • Trace the event and include switch, port, date and time, link down counter. • If same switch and port has at least 2 link down occurrences within 24 hours, further investigation required. • Note: • Make sure link down was due to involuntary port down from the partner side (e.g. not due to partner server reboot). • The criteria intends to catch major link down events.	ongoing
LInkErrorRecoveryCounter	The number of times the Port Training state machine has successfully completed the link error recovery process.	Clean, no errors	ongoing
Chip temperature	Temperature in C	If temperature reached max threshold FW will do protective thermal shutdown.	ongoing
Device FW version	Switch / HCA FW ver	Verify approved version is the last released version by NVIDIA, Need to see the cluster have similar versions	Days
Cables Information			
PN	Part number	No check required	Days
SN	Serial number	No check required	Days
FW ver	FW version	Verify approved version is the last released version by NVIDIA	Days
Module temperature	Optic module only	There is an alarm and threshold for each transceiver. <ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="d1bfc8cb-81de-4fd3-b7ed-43fd19d36b26"><ac:plain-text-body><![CDATA[Usually Warning [70c, 0c] and Alarm [80c, -10c]	ongoing
Rx power Tx power per lane	Optic module only	There is an alarm and threshold for each transceiver.	Minutes
Packet Discard			
PortRcvErrors	Total number of packets containing an error that were received on the port.	< 10 per second (perform 2 successive samples)	Minutes

PortXmitDiscards	Total number of outbound packets discarded by the port because the port is down or congested.	< 10 per second (perform 2 successive samples)	Minutes

6.6 Cluster Performance Verification

The tool used for validating cluster performance is known as ClusterKit, an integral component of the HPC-X Software Toolkit.

NVIDIA® HPC-X® presents a comprehensive software bundle encompassing MPI and SHMEM communication libraries. Within this package, various acceleration components are included, enhancing the performance and scalability of applications that operate on top of these libraries. Notably, UCX (Unified Communication X) accelerates the underlying send/receive (or put/get) messages. Also included, HCOLL, which accelerates the underlying collective operations used by the MPI/PGAS languages.

For detailed documentation, along with instructions for downloading and installing HPC-X, refer to HPC-X Documentation.

6.6.1 HPC-X is Functionality Verification

To ensure the correct operation of HPC-X, a straightforward MPI test program bundled with HPC-X can be employed. Use the following procedure:

1. Set the HPCX_HOME environment variable to point to the HPCX installation directory:

   ```
   % export HPCX_HOME=<HPCX Directory>
   ```

2. Initialize HPC-X environment variables:

   ```
   % source $HPCX_HOME/hpcx-init.sh
   % hpcx_load
   ```

3. Execute the precompiled MPI test program hello_c. The MPI program can be executed using either of the following methods:
   a. Inside a SLURM allocation or job, run:

      ```
      % mpirun $HPCX_MPI_TESTS_DIR/examples/hello_c
      ```

   b. Without SLURM using SSEH and explicitly setting hosts to run on:

      ```
      % mpirun --host <host1,host2,…,hostN> $HPCX_MPI_TESTS_DIR/examples/hello_c
      ```

      Alternatively, you can put all hostnames into a single file (hostfile) and pass that file to mpirun (see mpirun(1) man page for details):

      ```
      % mpirun --hostfile <hostfile> $HPCX_MPI_TESTS_DIR/examples/hello_c
      ```

4. The output should contain one line for every MPI process that was executed. Each line indicates the MPI rank of the process, the total number of processes, and the version of OpenMPI bundled with HPC-X. For instance:

```
Hello, world, I am 90 of 168, (Open MPI v4.1.5rc2, package: Open MPI root@hpc-kernel-03 Distribution,
ident: 4.1.5rc2, repo rev: v4.1.5rc1-16-g5980bac633, Unreleased developer copy, 150)
The number of lines should match the number of cores in the allocation
```

5. Check that the ClusterKit script (clusterkit.sh) is available. Run:

```
cpde ls -l $HPCX_CLUSTERKIT_DIR/bin/run_clusterkit.sh
```

6. Check that the file $HPCX_CLUSTERKIT_DIR/bin/run_clusterkit.sh exists and is executable.

6.6.2 Running ClusterKit

Prior to executing ClusterKit, it is important to have HPC-X properly set up with initialized environment variables. Additionally, ensure that the ClusterKit script (clusterkit.sh) is accessible, as instructed in the preceding section.
ClusterKit can be run inside SLURM allocation or job or without SLURM. When operating within a SLURM allocation, employ the following command:

```
$HPCX_CLUSTERKIT_DIR/bin/clusterkit.sh -d mlx5_4:1 -x "-d bw"
```

Where `-d adapter:port` selects which InfiniBand adapter and port to use and `-x` "`-d bw`" sets which test to run (bandwidth test).
If running outside SLURM allocation, use:

```
$HPCX_CLUSTERKIT_DIR/bin/clusterkit.sh -f hostfile -d mlx5_4:1 -x "-d bw"
```

Where `-f hostfile` sets hostfile to use. The hostfile contains the list of nodes to use (see mpirun(1) man page for details).

You can add `-D <output dir>` switch to set the output directory for the run. Without it, the output will be saved into the directory composed of date and time of the run (e.g., `20230731_154932`).

In the output directory two files are create, `bandwidth.json` and `bandwidth.txt`. `bandwidth.json` can be used for automatic processing of the results which is out of scope of this document. In `bandwidth.txt` see the last 3 line of text which look like:

```
Minimum bandwidth: 24869.6 MB/s between node14 and node28
Maximum bandwidth: 25208.7 MB/s between node02 and node13
Average bandwidth: 25002.5 MB/s
```

The results are in decimal Megabytes per second (10^6 Bytes per second).

6.6.3 Results Verification

Your cluster's performance is satisfactory when the minimum achieved result is at least 95% of the maximum available bandwidth, as illustrated in the table below.
For your convenience, the technology of your cluster interconnect is shown in the header of the `bandwidth.txt` file.
Expected InfiniBand Performance (for 4x Connections)

Technology	Speed, Gb/s	95% performance, MB/s
EDR	100	11 515
HDR	200	23 030
NDR	400	46 060

6.7 Review All Unhealthy Nodes

Once the UFM examines the behavior of subnet nodes, including switches and hosts, and identifies a node as "unhealthy" based on internal conditions, this node is displayed in the "Unhealthy Ports" list. Once a node is declared as "unhealthy," the Subnet Manager either ignores, reports, isolates, or disables the node. Users hold the authority to control the executed actions and the criteria that categorize a node as "unhealthy." Furthermore, the user can "clear" nodes previously labeled as "unhealthy".

To navigate through these functionalities using the Web User Interface, refer to Unhealthy Ports Window. to review all unhealthy nodes using Web UI. Alternatiely, use the REST API from a remote node via the Unhealthy Ports REST API.

6.8 Congestion Monitoring with UFM Telemetry

Since InfiniBand is lossless, the network does not drop packets which may cause network congestion. The metric XmitWaitPerc provides the percentage of time in which ports had data to send but could not progress due to congestion. This metric can be obtained per topology layer (distance from the source hosts towards the destination hosts) or for each link separately.

If a switch port connected to a host is showing >5% of XmitWaitPerc, then the most probably cause is that the host PCIe or its memory is not healthy.

If XmitWaitPerc >5% on links/layer that are not driving a host, then that is most probably caused by traffic that exceeds the capacity of that layer. This is normal for over-subscribed networks where the total number of cables connecting the switches of that layer to the next one is smaller than the number of cables connected to previous layers. But if the network is not over-subscribed, a high XmitWaitPerc can be strong sign that adaptive routing is not used, or some many-to-one traffic patterns are used by the applications. Or that some missing (unhealthy links) makes a specific switch over-subscribed.

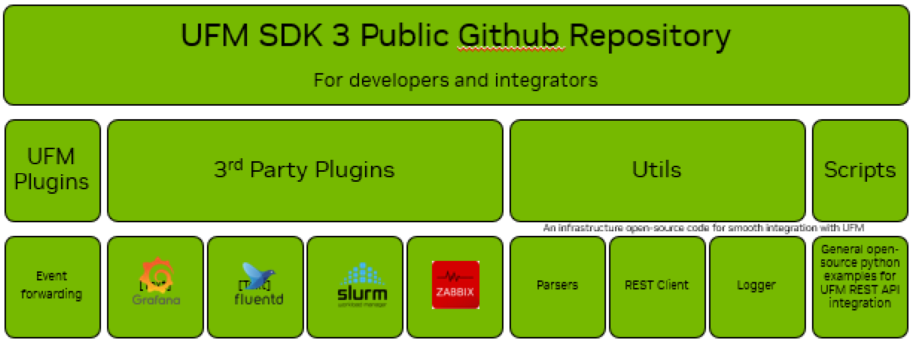
For more information, refer to "Top X Telemetry Sessions REST API" under Telemetry REST API.

6.9 Monitoring Systems Integrations

The monitoring system includes UFM Telemetry and optional streaming of its results into customer specific Network Management Systems. UFM Telemetry is responsible for collecting the vital networking metrics and forwarding them to a data-lake or other customer data analysis tools.

- A comprehensive array of plugins, scripts, recipes and tools to facilitate UFM integration with third-party network management systems can be accessed via a publicly available GitHub repository: UFM SDK Repository.

- Furthermore, the UFM Software Development Kit (SDK) allows extension of the capabilities of the UFM platform with additional tools.


UFM SDK 3 Public Github Repository
For developers and integrators
UFM Plugins
3rd Party Plugins
Utils
Scripts
An infrastructure open-source code for smooth integration with UFM
Event forwarding
Grafana
fluentd
slurm
ZABBIX
Parsers
REST Client
Logger
General open-source python examples for UFM REST API integration

The following are links for instruction detailing installation and usage of Telemetry Streaming/ Forwarding Plugins:

- FluentD Telemetry Streaming Plugin
- FluentD Events Streaming Plugin
- GRPC Telemetry Streaming Plugin
- Packet Mirroring Collector Plugin
- Packet Drop Rate (PDR) - Link Quality Check Capabilities

6.10 Latest SW Updates

- For a catalog of the validated configuration products and their respective versions that have been rigorously tested together and endorsed by NVIDIA, please refer to Quantum-2 Clusters. This page provides the exact version per InfiniBand product (e.g., Switch, HCA, UFM, Transceiver, etc.), that was released and tested as a bundle.
- In case your cluster is running Long Term Support (LTS) releases, it is recommended to check NVIDIA LTS web page for the latest LTS release. To gain insights into critical bug resolutions, it is recommended to visit the release notes for each specific product: Long-Term Support (LTS) Releases.
- Please note that, currently, a complete maintenance window is required for device firmware upgrades.
- For UFM and OpenSM upgrades, a staged approach can be adopted: begin by upgrading the secondary UFM, transition to it as the master, and subsequently proceed with upgrading the prior master UFM.

6.11 Cooling System Maintenance

- For cooling system maintenance using web UI, run the check temperature test from fabric validation tab. For more information, refer to Fabric Validation Tab.
- For cooling system maintenance using the REST APIs, issue a POST request using the following URL:

  ```
  POST /ufmRest/fabricValidation/tests/CheckTemperature
  ```

  For more information, refer to Fabric Validation Tests REST API.

7 List of Scenarios

This section provides a set of UFM API queries used to gather information about events identified by UFM. The events/scenarios are organized by their likelihood. For each event, comprehensive details are supplied, including a description, relevant fabric state, methods to retrieve the specific UFM event, corresponding event IDs, and suggested courses of action for remediation.

7.1 Bad Link/Port

<table>
<tr><th>Scenario</th><th>Bad Link/Port</th></tr>
<tr><td><b>Description</b></td><td>A port can drop too many packets</td></tr>
<tr><td><b>Fabric State</b></td><td>Any fabric state</td></tr>
<tr><td><b>How to Detect</b></td><td>Via UFM health reports, UFM events\alerts</td></tr>
<tr><td><b>UFM Alerts</b></td><td>
<table>
<tr><th>Event ID</th><th>Event Name</th></tr>
<tr><td>915</td><td>Critical BER reported</td></tr>
<tr><td>916</td><td>High BER reported</td></tr>
</table>
</td></tr>
</table>

Scenario	Bad Link/Port

Action Required	Review the firmware version of both the device and cable to ensure they are up to date. Fixing a bad link involves a sequence of actions to be followed on both ends of the link. Each of these actions should be executed in sequence, while verifying if the issue has been resolved. These actions are:
	1. Reset the link using NOS/mlx command:
	`interface ib <port number> clear counters`
	Example:
	`interface ib 1/1/1 clear counters`
	2. Pullout, clean, and reinsert the fiber.
	3. Consider replacing the transceiver - it is recommended to use two spare transceivers on both sides of the link. If this resolves the issue, determine in a controlled lab environment which of the removed transceivers was responsible for the problem. This minimizes downtime on the operational floor.
	4. Try using a new or spare fiber between the source and destination.
	5. If the problem persists, the Switch or NIC are likely malfunctioning. Replace them with new hardware.
	6. If the issue remains unresolved, contact NVIDIA Networking Support. To determine a link's viability, a short period of high Bit Error Rate (BER) or more complex criteria may disqualify it. Conversely, a significant monitoring duration of approximately 30 minutes is essential to confirm its reliability.
	The complete procedure for fixing a link involves isolating the link, applying manual fixes from the above list, waiting for confirmation of resolution, and then de-isolating the link:
	7. If cable temperature exhibits considerable drift or surpasses established thresholds, isolate it without attempting to repair. Allow the temperature to stabilize before de-isolating.
	8. Identify problematic links using the UFM report of bad links. Alternatively, consider links in the INIT state due to firmware AutoDetect or monitor changes in link counters.
	9. Employ the UFM port isolation routine to isolate the link, transitioning it to the INIT state.
	10. Apply the fixes outlined in actions (a to f above), addressing one issue at a time.
	11. Await the Cable Validation tool to report port health for the impacted ports.
	12. De-isolate the link using the UFM port de-isolation routine.
	13. Verify that the link has successfully transitioned to the ACTIVE state.

Additional Reading	Refer to Cable Validation Tool and Reports REST API
	`{`
	`"cables": true,`
	`"symbol_ber_check:": true,`
	`"effective_ber_check:": true`
	`}`

7.2 Wrong Connection

<table>
<tr><th>Scenario</th><th>Wrong Connection</th></tr>
<tr><td><b>Description</b></td><td>A cable that is not connected according to the topology can act as a bottleneck.</td></tr>
<tr><td><b>Fabric State</b></td><td>Bring-up/maintenance</td></tr>
<tr><td><b>How to Detect</b></td><td>Set master topology scheduled topodiff to run once every 3 hours using Topology Compare REST API.
Turn on UFM topology compare capability.</td></tr>
<tr><td><b>UFM Alerts</b></td><td><table><tr><th>Event ID</th><th>Event Name</th></tr><tr><td>1316</td><td>Topo Config Subnet Mismatch</td></tr></table></td></tr>
<tr><td><b>Action Required</b></td><td>Fix Misconnection</td></tr>
</table>

7.3 Host is Hanging

<table>
<tr><th>Scenario</th><th>Host is Hanging</th></tr>
<tr><td><b>Description</b></td><td>Most commonly due to software issue or overload</td></tr>
<tr><td><b>Fabric State</b></td><td>Any</td></tr>
<tr><td><b>How to Detect</b></td><td>SM alerts about unresponsive host.</td></tr>
<tr><td><b>UFM Alerts</b></td><td><table><tr><th>Event ID</th><th>Event Name</th></tr><tr><td>331</td><td>Node is Down</td></tr></table></td></tr>
<tr><td><b>Action Required</b></td><td>Fix stuck host</td></tr>
</table>

7.4 Low Bandwidth

Scenario	Low Bandwidth
Description	In the event that certain network cables remain disconnected and unfixed, the network may lose bandwidth. While application performance decline can occur due to various factors, the most common factor is losing network links.
Fabric State	Bring-up/maintenance
UFM API Query	GET /ufmRest/app/events
How to Detect	Customer complaints. Examine UFM Congestion dashboard.

<table>
<tr><th>Scenario</th><th>Low Bandwidth</th></tr>
<tr><td><b>UFM Alerts</b></td><td><table><tr><th>Event ID</th><th>Event Name</th></tr><tr><td>122</td><td>Congested Bandwidth (%) Threshold Reached</td></tr><tr><td>134</td><td>T4 Port Congested Bandwidth</td></tr></table></td></tr>
<tr><td><b>Action Required</b></td><td>Assess the network's bandwidth while considering the impact of the disconnected links, as this could explain the decrease in bandwidth. If the calculations align, it is crucial to replace the affected cables. In cases where the figures do not match, a deeper analysis may be necessary, including an analysis of the application's traffic pattern and the utilized transport methods.</td></tr>
</table>

7.5 SHARP AM Issue

<table>
<tr><th>Scenario</th><th>SHARP AM Issue</th></tr>
<tr><td><b>Description</b></td><td>SHARP job failure</td></tr>
<tr><td><b>Fabric State</b></td><td>Any</td></tr>
<tr><td><b>UFM API Query</b></td><td>GET /ufmRest/app/events</td></tr>
<tr><td><b>How to Detect</b></td><td>Review UFM events</td></tr>
<tr><td><b>UFM Alerts</b></td><td><table><tr><th>Event ID</th><th>Event Name</th></tr><tr><td>1523</td><td>Job Start Failed</td></tr><tr><td>1524</td><td>Job Error</td></tr><tr><td>1532</td><td>SHARP is not Responding</td></tr></table></td></tr>
<tr><td><b>Action Required</b></td><td><ul><li>Check SHARP AM configuration.</li><li>Check job scheduler configuration.</li></ul></td></tr>
</table>

7.6 Inadequate Control of Cluster Temperature

Scenario	Inadequate Control of Cluster Temperature
Description	The cluster's temperature could escalate rapidly or surpass a designated threshold, potentially risking equipment damage.
Fabric State	Any
How to Detect	UFM temperature alert. Via REST API: Request URL: POST /ufmRest/fabricValidation/tests/CheckTemperature For more information, refer to Fabric Validation Tests REST API.

<table>
<tr><th>Scenario</th><th>Inadequate Control of Cluster Temperature</th></tr>
<tr><td><b>UFM Alerts</b></td><td><table>
<tr><th>Event ID</th><th>Event Name</th></tr>
<tr><td>912</td><td>Module Temperature High Threshold Reached</td></tr>
<tr><td>919</td><td>Cable Temperature High</td></tr>
<tr><td>1400</td><td>High Ambient Temperature</td></tr>
</table></td></tr>
<tr><td><b>Action Required</b></td><td>Fix the fans/cooling/air conditioning systems.</td></tr>
</table>

7.7 Non-Responsive Switch

<table>
<tr><th>Scenario</th><th>Non-Responsive Switch</th></tr>
<tr><td><b>Description</b></td><td>switch is not responding due to software/hardware issues</td></tr>
<tr><td><b>Fabric State</b></td><td>Any</td></tr>
<tr><td><b>How to Detect</b></td><td>Examine the UFM generated alerts for such cases</td></tr>
<tr><td><b>UFM Alerts</b></td><td><table>
<tr><th>Event ID</th><th>Event Name</th></tr>
<tr><td>907</td><td>Switch is Down</td></tr>
<tr><td>909</td><td>Director Switch is Down</td></tr>
<tr><td>1312</td><td>Suspected switch Reboot</td></tr>
</table></td></tr>
<tr><td><b>Action Required</b></td><td>Isolate and reboot the switch, then de-isolate it. If problem persists, keep it isolated.</td></tr>
<tr><td><b>Additional Reading</b></td><td>Refer to Actions REST API.</td></tr>
</table>

7.8 Infinite Switch Reboots due to Switch HW Malfunction

Scenario	Infinite Switch Reboots due to Switch HW Malfunction
Description	While it is not highly probable, switch's software or hardware malfunction may lead to multiple switch reboots. Should this occur, the fabric might experience packet loss transmitted through that particular switch. Consequently, the SDN infrastructure could become busy managing these incidents and reconfiguring the network.
Fabric State	Any
How to Detect	UFM alerts about unhealthy switch

<table>
<tr><th>Scenario</th><th>Infinite Switch Reboots due to Switch HW Malfunction</th></tr>
<tr><td><b>UFM Alerts</b></td><td><table><tr><th>Event ID</th><th>Event Name</th></tr><tr><td>907</td><td>Switch is Down</td></tr><tr><td>909</td><td>Director Switch is Down</td></tr><tr><td>1312</td><td>Suspected switch Reboot</td></tr></table></td></tr>
<tr><td><b>Action Required</b></td><td>Isolate and reboot the switch, then de-isolate it. If problem persists, keep it isolated.</td></tr>
<tr><td><b>Additional Reading</b></td><td>Refer to Actions REST API.</td></tr>
</table>

7.9 UFM Server Failover

<table>
<tr><th>Scenario</th><th>UFM Server Failover</th></tr>
<tr><td><b>Description</b></td><td>UFM server failure causes UFM HA failover to standby host.</td></tr>
<tr><td><b>Fabric State</b></td><td>Any</td></tr>
<tr><td><b>How to Detect</b></td><td>Monitor UFM events</td></tr>
<tr><td><b>UFM Alerts</b></td><td><table><tr><th>Event ID</th><th>Event Name</th></tr><tr><td>602</td><td>UFM Server Failover</td></tr></table></td></tr>
<tr><td><b>Action Required</b></td><td>Execute a UFM health report and validate the successful execution of UFM failover. Fix any hardware failures and restore the UFM High Availability (HA) cluster. Ensure the collection of system dumps open a support ticket through NVIDIA Networking Support.</td></tr>
</table>

7.10 SM Not Responding

<table>
<tr><th>Scenario</th><th>SM Not Responding</th></tr>
<tr><td><b>Description</b></td><td>The OpenSM process hangs.</td></tr>
<tr><td><b>Fabric State</b></td><td>Any</td></tr>
<tr><td><b>How to Detect</b></td><td>Monitor UFM events</td></tr>
<tr><td><b>UFM Alerts</b></td><td><table><tr><th>Event ID</th><th>Event Name</th></tr><tr><td>545</td><td>SM is not responding</td></tr></table></td></tr>
<tr><td><b>Action Required</b></td><td>Automated SM restart must be handled automatically within UFM. Conduct a UFM health report to verify the UFM's ability to restart OpenSM. Remember to gather sysdump data and proceed to initiate a support ticket through NVIDIA Networking Support.</td></tr>
</table>

Scenario	SM Not Responding
Additional Reading	Fabric Validation Tests REST API

7.11 UFM Management Interface is Down

<table>
<tr><th>Scenario</th><th>UFM Management Interface is Down</th></tr>
<tr><td><b>Description</b></td><td>UFM server InfiniBand management interface is down.</td></tr>
<tr><td><b>Fabric State</b></td><td>Any</td></tr>
<tr><td><b>How to Detect</b></td><td>Monitor UFM events</td></tr>
<tr><td><b>UFM Alerts</b></td><td><table><tr><th>Event ID</th><th>Event Name</th></tr><tr><td>546</td><td>Management interface is down</td></tr><tr><td></td><td></td></tr></table></td></tr>
<tr><td><b>Action Required</b></td><td>Verify the connectivity of the management interface. Run the "ibstat" command to confirm that the State: Active and Physical state: LinkUp. If the issue persists, retrieve sysdump information and proceed to create a support ticket through NVIDIA Networking Support.</td></tr>
</table>

7.12 UFM Server Disk Utilization

<table>
<tr><th>Scenario</th><th>Threshold for UFM Server Disk Utilization Exceeded</th></tr>
<tr><td><b>Description</b></td><td>UFM server disk utilization threshold is reached and UFM is not able to free disk space.</td></tr>
<tr><td><b>Fabric State</b></td><td>Any</td></tr>
<tr><td><b>How to Detect</b></td><td>Monitor UFM events</td></tr>
<tr><td><b>UFM Alerts</b></td><td><table><tr><th>Event ID</th><th>Event Name</th></tr><tr><td>525</td><td>Disk utilization threshold reached</td></tr></table></td></tr>
<tr><td><b>Action Required</b></td><td>Clean UFM server disk from third party data. If the problem persists, collect sysdump and open support ticket in NVIDIA Networking Support.</td></tr>
</table>

7.13 Duplicated GUIDs

Scenario	Duplicated GUIDs
Description	Cards or switches can be accidentally provisioned with duplicated GUIDs. This is a rare case.

Scenario	Duplicated GUIDs
Fabric State	Bring-up/maintenance
How to Detect	Subnet Manager detects and reports duplicated GUIDs via UFM
UFM Alerts	<table><tr><th>Event ID</th><th>Event Name</th></tr><tr><td>1310</td><td>Duplicated node GUID was detected</td></tr><tr><td>1311</td><td>Duplicated port GUID was detected</td></tr></table>
Action Required	• Collect sysdump and open support ticket in NVIDIA Networking Support. • Turn off switch/host.
Additional Reading	Actions REST API

8 Retrieving UFM Issues

The following are the two methods to retrieve UFM issues:

8.1 Using Push Mechanism (over FluentBit)

To retrieve events using the push mechanism over FluentBit, follow the instructions provided in the UFM Telemetry Forwarder.

8.2 Using Pull Mechanism (over REST API)

To retrieve events using the push mechanism over REST APU, follow the instructions provided in the Events REST API available in UFM Enterprise REST API Guide.

8.2.1 Get All Events REST API

Description	This API allows you to retrieve information about all events currently running in the fabric or get information about a specific event using its ID.
Request URL	GET /ufmRest/app/events
Request Content Type	Application/json

Response	`{`
	`"category": "Logical Model",`
	`"severity": "Info",`
	`"timestamp": "2017-09-19 10:49:03.018",`
	`"counter": null,`
	`"object_name": "Grid",`
	`"object_path": "Grid",`
	`"name": "Network Added",`
	`"write_to_syslog": false,`
	`"type": "352",`
	`"id": 227,`
	`"description": "Network management is added"`
	`},`

	`{`
	`"category": "Fabric Notification",`
	`"severity": "Info",`
	`"timestamp": "2017-09-19 10:49:11.520",`
	`"counter": null,`
	`"object_name": "Grid",`
	`"object_path": "Grid",`
	`"name": "Fabric Configuration Started",`
	`"write_to_syslog": false,`
	`"type": "901",`
	`"id": 228,`
	`"description": "Fabric Configuration started."`
	`}`

Possible Filters	• object_name - filters by object name • type - filters by type • category - filters by category • severity - filters by severity • group - filters events by the group that has caused the event
Status Code	• 200 - OK

9 UFM Dashboards Customization

For customers using UFM Web UI, a range of UFM widgets are available to enable customization of the UFM dashboard. This customization is particularly useful for unattended usage in Service Operation Centers. For more detailed insights, refer to Dashboard Views and Panel Management.

10 Document Revision History

Version	Date	Description of Change
1.2	March 20, 2024	Updated Operations Procedures
1.1	December 11, 2023	Updated Latest SW Updates
1.0	August 15, 2023	First Release

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. Neither NVIDIA Corporation nor any of its direct or indirect subsidiaries and affiliates (collectively: “NVIDIA”) make any representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice. Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer (“Terms of Sale”). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, “MATERIALS”) ARE BEING PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA, the NVIDIA logo, and Mellanox are trademarks and/or registered trademarks of NVIDIA Corporation and/or