GPU Management and Deployment

XID Errors

Release r555

NVIDIA Corporation

Sep 24, 2024

Contents

Xid Errors

This document explains what Xid messages are, and is intended to assist system administrators, developers, and FAEs in understanding the meaning behind these messages as an aid in analyzing and resolving GPU-related problems.

Chapter 1. What Is an Xid Message

The Xid message is an error report from the NVIDIA driver that is printed to the operating system's kernel log or event log. Xid messages indicate that a general GPU error occurred, most often due to the driver programming the GPU incorrectly or to corruption of the commands sent to the GPU. The messages can be indicative of a hardware problem, an NVIDIA software problem, or a user application problem.

These messages provide diagnostic information that can be used by both users and NVIDIA to aid in debugging reported problems.

The meaning of each message is consistent across driver versions.

Chapter 2. What Is an SXid Message

NVIDIA drivers for NVSwitch report error conditions relating to NVSwitch hardware in kernel logs through a similar mechanism to Xids. These “Switch Xids”, or SXids and guidelines for their usage are documented separately in the Fabric Manager User Guide.

Chapter 3. How to Use Xid Messages

Xid messages are intended to be used as debugging guides. Because many problems can have multiple possible root causes, it's not always feasible to understand each issue from the Xid value alone.

For example, an Xid error might indicate that a user program tried to access invalid memory. But, in theory, memory corruption due to PCIE or frame buffer ("FB") problems could corrupt any command and thus cause almost any error. Generally, the Xid classifications listed below should be used as a starting point for further investigation of each problem.

The GPU Debug Guidelines manual provides additional guidance for debugging GPU problems, including advice for interpreting Xids and provides guidance for next steps to handle common Xids.

Chapter 4. Working with Xid Errors

4.1. Viewing Xid Messages

Under Linux, the Xid error messages are placed in the location `/var/log/messages`.

Grep for “NVRM: Xid”to find all the Xid messages.

The following is an example of a Xid string:

```
[...] NVRM: GPU at 0000:03:00: GPU-b850f46d-d5ea-c752-ddf3-c4453e44d3f7

[...] NVRM: Xid (0000:03:00): 14, Channel 00000001
```

- The first Xid in the log file is preceded by a line that contains the GPU GUID and device IDs.

  In the above example:

  - “GPU-b850f46d-d5ea-c752-ddf3-c4453e44d3f7” is the GUID.

    The GUID is a globally unique, immutable identifier for each GPU.
  - “0000:03:00” is the device ID.
- Each subsequent Xid line contains the device ID, the Xid error, and information about the Xid.

  In the above example:

  - “0000:03:00” is the device ID.
  - “14” is the Xid error identifier .
  - “Channel 00000001” is data specific to that Xid error.

4.2. Tools That Provide Additional Information about Xid Errors

NVIDIA provides three additional tools that may be helpful when dealing with Xid errors.

- **nvidia-smi** is a command-line program that installs with the NVIDIA driver. It reports basic monitoring and configuration data about each GPU in the system. `nvidia-smi` can list ECC error counts (Xid 48) and indicate if a power cable is unplugged (Xid 54), among other things. Please see the `nvidia-smi` man page for more information. Run `nvidia-smi -q` for basic output.

- **NVIDIA Data Center GPU Manager (DCGM)** is a suite of tools for managing and monitoring NVIDIA datacenter GPUs in cluster environments. It includes active health monitoring, comprehensive diagnostics, system alerts and governance policies including power and clock management. DCGM diagnostics is a health checking tool that can check for basic GPU health, including the presence of ECC errors, PCIe problems, bandwidth issues, and general problems with running CUDA programs.

  DCGM is documented and downloadable at https://developer.nvidia.com/dcgm

- **nvidia-bug-report.sh** is a script that installs with the NVIDIA driver. It collects debug logs and command outputs from the system, including kernel logs and logs collected by the NVIDIA driver itself. The command should be run as root:

  ```
  sudo nvidia-bug-report.sh
  ```

  The output of this tool is a single compressed text file, `nvidia-bug-report.log.gz`, that can be included when reporting problems to NVIDIA.

  `nvidia-bug-report.sh` will typically run quickly, but in rare cases may run slowly. Allow up to one hour for it complete. If the command remains hung, run the command with additional arguments as:

  ```
  nvidia-bug-report.sh --safe-mode --extra-system-data
  ```

  This will collect alternative logs, in such a way that it should avoid common causes of hangs during debug collection.

4.3. Analyzing Xid Errors

The following table lists the recommended actions to take for various issues encountered.

Issue	Recommended Action
Suspected User Programming Issues	Run the debugger tools. Refer to the Compute Sanitizer "memcheck" tool and CUDA-GDB documentation.
Suspected Hardware Problems	Contact the hardware vendor. They can run through their hardware diagnostic process.
Suspected Driver Problems	File a bug with NVIDIA, including output of the command `nvidia-bug-report.sh`. Refer to the document GPU Debug Guidelines for guidance on gathering additional information to provide to NVIDIA and troubleshooting common Xid causes.

Chapter 5. Xid Error Listing

The following table lists the Xid errors along with the potential causes for each.

Xid	Failure
1	Invalid or corrupted push buffer stream
2	Invalid or corrupted push buffer stream
3	Invalid or corrupted push buffer stream
4	Invalid or corrupted push buffer stream
	GPU semaphore timeout
5	Unused
6	Invalid or corrupted push buffer stream
7	Invalid or corrupted push buffer address
8	GPU stopped processing
9	Driver error programming GPU
10	Unused
11	Invalid or corrupted push buffer stream
12	Driver error handling GPU exception
13	Graphics Engine Exception
14	Unused
15	Unused
16	Display engine hung
17	Unused
18	Bus mastering disabled in PCI Config Space
19	Display Engine error
20	Invalid or corrupted Mpeg push buffer
21	Invalid or corrupted Motion Estimation push buffer

Table

Xid	Failure
22	Invalid or corrupted Video Processor push buffer
23	Unused
24	GPU semaphore timeout
25	Invalid or illegal push buffer stream
26	Framebuffer timeout
27	Video processor exception
28	Video processor exception
29	Video processor exception
30	GPU semaphore access error
31	GPU memory page fault
32	Invalid or corrupted push buffer stream
33	Internal micro-controller error
34	Video processor exception
35	Video processor exception
36	Video processor exception
37	Driver firmware error
38	Driver firmware error
39	Unused
40	Unused
41	Unused
42	Video processor exception
43	GPU stopped processing
44	Graphics Engine fault during context switch
45	Preemptive cleanup, due to previous errors – Most likely to see when running multiple cuda applications
46	GPU stopped processing
47	Video processor exception
48	Double Bit ECC Error
49	Unused
50	Unused
51	Unused
52	Unused

Table

Xid	Failure
53	Unused
54	Auxiliary power is not connected to the GPU board
55	Unused
56	Display Engine error
57	Error programming video memory interface
58	Unstable video memory interface detected
	EDC error - clarified in printout
59	Internal micro-controller error (older drivers)
60	Video processor exception
61	Internal micro-controller breakpoint/warning (newer drivers)
62	Internal micro-controller halt (newer drivers)
63	ECC page retirement or row remapping recording event
64	ECC page retirement or row remapper recording failure
65	Video processor exception
66	Illegal access by driver
67	Illegal access by driver
68	NVDEC0 Exception
69	Graphics Engine class error
70	CE3: Unknown Error
71	CE4: Unknown Error
72	CE5: Unknown Error
73	NVENC2 Error
74	NVLINK Error
75	CE6: Unknown Error
76	CE7: Unknown Error
77	CE8: Unknown Error
78	vGPU Start Error
79	GPU has fallen off the bus
80	Corrupted data sent to GPU
81	VGA Subsystem Error
82	NVJPG0 Error

Table

Xid	Failure
83	NVDEC1 Error
84	NVDEC2 Error
85	CE9: Unknown Error
86	OFA Exception
87	Reserved
88	NVDEC3 Error
89	NVDEC4 Error
90	Reserved
91	Reserved
92	High single-bit ECC error rate
93	Non-fatal violation of provisioned InfoROM wear limit
94	Contained ECC error
95	Uncontained ECC error
96	NVDEC5 Error
97	NVDEC6 Error
98	NVDEC7 Error
99	NVJPG1 Error
100	NVJPG2 Error
101	NVJPG3 Error
102	NVJPG4 Error
103	NVJPG5 Error
104	NVJPG6 Error
105	NVJPG7 Error
106	SMBPBI Test Message
107	SMBPBI Test Message Silent
108	Reserved
109	Context Switch Timeout Error
110	Security Fault Error
111	Display Bundle Error Event
112	Display Supervisor Error
113	DP Link Training Error

Table

Xid	Failure
114	Display Pipeline Underflow Error
115	Display Core Channel Error
116	Display Window Channel Error
117	Display Cursor Channel Error
118	Display Pixel Pipeline Error
119	GSP RPC Timeout
120	GSP Error
121	C2C Link Error
122	SPI PMU RPC Read Failure
123	SPI PMU RPC Write Failure
124	SPI PMU RPC Erase Failure
125	Inforom FS Failure
126- 139	Reserved
140	Unrecovered ECC Error
141	Reserved
142	Reserved
143	GPU Initialization Failure

For the comprehensive list of XIDs, please refer to https://github.com/NVIDIA/open-gpu-kernel-modules/blob/main/src/common/sdk/nvidia/inc/nverror.h.

Chapter 6. Common Xid Errors

This section provides more information on some common Xid errors.

6.1. Xid 13: GR: SW Notify Error

This event is logged for general user application faults. Typically this is an out-of-bounds error where the user has walked past the end of an array, but could also be an illegal instruction, illegal register, or other case.

In rare cases, it's possible for a hardware failure or system software bugs to materialize as XID 13.

When this event is logged, NVIDIA recommends the following:

1. Run the application in cuda-gdb or the Compute Sanitizer `memcheck` tool , or
2. Run the application with `CUDA_DEVICE_WAITS_ON_EXCEPTION=1` and then attach later with cuda-gdb, or
3. File a bug if the previous two come back inconclusive to eliminate potential NVIDIA driver or hardware bug.

Note: The Compute Sanitizer `memcheck` tool instruments the running application and reports which line of code performed the illegal read.

6.2. Xid 31: FIFO: MMU Error

This event is logged when a fault is reported by the MMU, such as when an illegal address access is made by an applicable unit on the chip. Typically these are application-level bugs, but can also be driver bugs or hardware bugs.

When this event is logged, NVIDIA recommends the following:

1. Run the application in cuda-gdb or the Compute Sanitizer `memcheck` tool, or
2. Run the application with CUDA_DEVICE_WAITS_ON_EXCEPTION=1 and then attach later with cuda-gdb, or
3. File a bug if the previous two come back inconclusive to eliminate potential NVIDIA driver or hardware bug.

Note: The Compute Sanitizer `memcheck` tool instruments the running application and reports which line of code performed the illegal read.

6.3. Xid 32: PBDMA Error

This event is logged when a fault is reported by the DMA controller which manages the communication stream between the NVIDIA driver and the GPU over the PCI-E bus. These failures primarily involve quality issues on PCI, and are generally not caused by user application actions.

6.4. Xid 43: Reset Channel Verif Error

This event is logged when a user application hits a software induced fault and must terminate. The GPU remains in a healthy state.

In most cases, this is not indicative of a driver bug but rather a user application error.

6.5. Xid 45: OS: Preemptive Channel Removal

This event is logged when the user application aborts and the kernel driver tears down the GPU application running on the GPU. Control-C, GPU resets, sigkill are all examples where the application is aborted and this event is created.

In many cases, this is not indicative of a bug but rather a user or system action.

6.6. Xid 48: DBE (Double Bit Error) ECC Error

This event is logged when the GPU detects that an uncorrectable error occurs on the GPU. This is also reported back to the user application. A GPU reset or node reboot is needed to clear this error.

The tool `nvidia-smi` can provide a summary of ECC errors. See *Tools That Provide Additional Information about Xid Errors*.

6.7. Xid 63, 64: ECC Page Retirement or Row Remapping

These events are logged when the GPU handles ECC memory errors on the GPU.

On GPUs that support row remapping, starting with NVIDIA® Ampere archtecture GPUs, these events provide details on row remapper activity. For more information row remapper Xids, refer to https://docs.nvidia.com/deploy/a100-gpu-mem-error-mgmt/index.html#row-remapping.

On earlier GPUs that support dynamic page retirement, these events provide details on dynamic page retirement activity. For more information on dynamic page retirement Xids, refer to https://docs.nvidia.com/deploy/dynamic-page-retirement/index.html.

6.8. Xid 74: NVLink Error

This event is logged when the GPU detects that a problem with a connection from the GPU to another GPU or NVSwitch over NVLink. A GPU reset or node reboot is needed to clear this error.

This event may indicate a hardware failure with the link itself, or may indicate a problem with the device at the remote end of the link. For example, if a GPU fails, another GPU connected to it over NVLink may report an Xid 74 simply because the link went down as a result.

The `nvidia-smi nvlink` command can provide additional details on NVLink errors, and connection information on the links.

If this error is seen repeatedly and GPU reset or node reboot fails to clear the condition, contact your hardware vendor for support.

6.9. Xid 79: GPU has fallen off the bus

This event is logged when the GPU driver attempts to access the GPU over its PCI Express connection and finds that the GPU is not accessible.

This event is often caused by hardware failures on the PCI Express link causing the GPU to be inaccessible due to the link being brought down. Reviewing system event logs and kernel PCI event logs may provide additional indications of the source of the link failures.

This event may also be cause by failing GPU hardware or other driver issues.

6.10. Xid 93: Non-fatal violation of provisioned InfoROM wear limit

This event is logged when the GPU driver fails to update the InfoROM due to violation of the provisioned InfoROM wear limit that was set for the GPU using NVFlash using `nvflash -=elsessionstart`.

In most cases this is not indicative of a driver or flash failure, but rather the intentional use of the InfoROM wear protection feature as set by NVFlash.

6.11. Xid 94, 95: Contained/uncontained ECC errors

These events are logged when GPU drivers handle ECC memory errors in GPUs that support ECC Error containment, starting with NVIDIA® A100 GPUs.

Detailed documentation for these Xids are covered in the A100 Memory error management manual: https://docs.nvidia.com/deploy/a100-gpu-mem-error-mgmt/index.html#row-remapping

6.12. Xid 110: Security fault error

This event should be uncommon unless there is a hardware failure. To recover, revert any recent system hardware modifications and cold reset the system. If this fails to correct the issue, contact your hardware vendor for assistance.

6.13. Xid 119, 120: GSP RPC Timeout / GSP Error

One or both of these events may be logged when an error occurs in code running on the GSP core of the GPU and/or a timeout occurs while waiting for the GSP core of the GPU to respond to an RPC message. A GPU reset or node power cycle may be needed if the error persists. If this problem reoccurs after a power cycle, follow the NVIDIA GPU Debug Guidelines document for additional debugging steps.

6.14. Xid 121: C2C Link corrected error

This event may occur when the GPU driver has observed corrected errors on the C2C NVLink connection to a Grace CPU. These errors are corrected by the system and have no operational impact. Resetting the GPU at an available service window will allow the GPU to retrain the link.

6.15. Xid 140: ECC unrecovered error

This event may occur when the GPU driver has observed uncorrectable errors in GPU memory, in such a way as to interrupt the GPU driver's ability to mark the pages for dynamic page offlining or row remapping. Reset the GPU, and if the problem persists, contact your hardware vendor for support.

Chapter 7. Notices

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. NVIDIA Corporation (“NVIDIA”) makes no 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.

7.1. Trademarks

Copyright