NVIDIA Trusted Computing Solutions

Release Notes

RN-11468-001 | June 2024

Table of Contents

Overview

The NVIDIA® Trusted Computing features of the NVIDIA H100 Tensor Core GPUs have updates and new features in this GA release.

This release consists of the CUDA Toolkit version 12.4 that is paired with NVIDIA® Data Center GPU Drivers version 550.54.15.

This software release features a complete software stack that targets an NVIDIA H100 GPU in passthrough mode with a session key for encryption and authentication and basic use of the developer tools. The code and data will be confidential up to the limits of the NIST SP800-38D AES-GCM standard, and after this point, the VM needs to be restarted.

Before deploying workloads, NVIDIA recommends that users invoke good practices, such as performing regular attestations.

New Features

Early Access (EA) of Protected PCIe is enabled in this release. Supported in an 8-way HGX system, NVLinks will now be enabled to increase

Key Rotation details

Limitations

This section provides a list of the known limitations in this release.

Full Confidential Computing Mode

- If users want to adhere to true CC requirements, only one GPU per VM is allowed. Multiple GPUs that are assigned to a VM to modes other than PPCI, will produce undefined behavior.
- CUDA limits the total number of concurrent CUDA contexts to 18.
  In CC mode (SPT), there is a system-wide limit on the number of secure Copy Engine channels.
- Developer Tools supports profiling only.
  - Nsys CUDA Trace, CUPTI APIs, and GPU Crash Dump **are** supported.
  - Debugging, for example cuda-gdb, modes **are not** supported.
- Video performance placeholder.
  - Video performance that uses small resolutions might have performance reductions.
  - Due to the system-wide limit on the secure Copy Engine channels, the number of concurrent video decode sessions is limited to approximately 89.

HGX Protected PCIe Mode

Peer-to-peer copies using CUDA APIs, for example `cudaMemcpyAsync`, are only supported when peer access is enabled between the source and destination devices using `cudaDeviceEnablePeerAccess` or `cuCtxEnablePeerAccess`.

Confidential Computing and Protected PCIe Modes

- CUDA APIs that use pinned host memory are not supported.
  In the current generation of CC systems, CPU architectures do not allow external entities to access VM memory, and IO devices, such as GPUs, are not allowed to directly access Guest VM memory. To transparently allow applications to operate, use Unified Virtual Memory.
- Graphic applications are not supported in CC modes.
  Hopper CC is targeted towards compute-centric workloads, so graphical interop modes/applications are not supported.
- Ready STate 010
  Generic topic wording on 010 bullet.
- Attestation software run as root will auto-set ready
  Lorem Ipsum wordign on attestation Softwrae running

- Graphic applications are not supported in CC modes.
  Hopper CC is targeted towards compute-centric workloads, so graphical interop modes/applications are not supported.
- Certain CUDA samples are not supported.
  Due to the the samples that use pinned host memory, or are video applications, the following samples will not run on Hopper CC modes:
  - `convolutionTexture`
  - `cudaNvSci`
  - `dct8x8`
  - `lineOfSight`
  - `simpleCubemapTexture`
  - `simpleIPC`
  - `simpleLayeredTexture`
  - `simplePitchLinearTexture`
  - `simpleStream`
  - `simpleTexture`
  - `simpleTextureDrv`
  - `watershedSegmentationNPP`
- Certain CUDA APIs are unsupported.
  - `cudaHostRegister` and cudaHostUnregister return `cudaErrorNotSupported` in HCC mode.
  - Host Memory Registration is not supported in CC mode.
    CUDA programs are expected to query `cudaDevAttrHostRegisterSupported` before invoking the cudaHostRegister API.
  - The CUDA UMD shall return zero value for the following device attributes in HCC mode:
    - `cudaDevAttrHostRegisterSupported`
    - `cudaDevAttrCanUseHostPointerForRegisteredMem` and `CU_DEVICE_ATTRIBUTE_CAN_USE_HOST_POINTER_FOR_REGISTERED_MEM`
    - `cudaDevAttrHostRegisterReadOnlySupported` and `CU_DEVICE_ATTRIBUTE_READ_ONLY_HOST_REGISTER_SUPPORTED`
- GPUDirect RDMA is not supported in CC mode.
  The CUDA UMD returns a zero value for the `cudaDevAttrGPUDirectRDMASupported` device attribute and `CU_DEVICE_ATTRIBUTE_GPU_DIRECT_RDMA_SUPPORTED` in HCC mode.
- CUArray types used in CUDA Memcpy APIs are allowed only in HCC mode when the source and the destination addresses are on the same local GPU device.
  Host-to-Array and Array-to-Host copies are not supported because of the potential to require a conversion between pitch-linear and block-linear access patterns of the CUArray memory type during the secure copy operation.

Table 1. Memcopy API behaviour on CUDA Arrays in HCC Mode

API	Behavior in CC mode

cuMemcpyHToA(Async) cuMemcpyAToH(Async) cudaMemcpy2DToArray(Async) cudaMemcpy2DFromArray(Async)	Returns CUDA_ERROR_NOT_SUPPORTED.
cudaMemcpy3D(Async)	When srcArray or dstArray is passed into cudaMemcpy3DParms, the API returns CUDA_ERROR_NOT_SUPPORTED.
cuMemcpy2D(Async) cuMemcpy2DUnaligned() cuMemcpy3D(Async)	The following combinations in the pCopy parameter return CUDA_ERROR_NOT_SUPPORTED: ○ srcMemoryType == CU_MEMORYTYPE_HOST and dstMemoryType == CU_MEMORYTYPE_ARRAY ○ srcMemoryType == CU_MEMORYTYPE_ARRAY and dstMemoryType == CU_MEMORYTYPE_HOST

The kind parameter that is accepted by some of the above memory copy APIs do not change the API behavior regarding array copies. Passing CUDA arrays into the CUDA memory copy APIs with kind set to any value might lead to undefined behavior.

Memset APIs on 2D and 3D arrays are supported in HCC mode because these APIs use GPU kernels to initialize a CUArray in device memory. Since the device memory that is mapped under the CUDA UMD belongs to CPR in the HCC mode, these operations are not changed from the non-CC mode implementation.

Known Issues

- A key rotation feature is missing.
  A sophisticated attacker with physical,or logical superuser, access to the system might be able to act as a passive adversary to capture the ciphertext and execute an attempt to break the ciphertext or the key.

  Workaround
  Users should review the latest research on the effects of extreme usage of AES keys and the cryptographic wear out to determine their requirements for an attacker advantage. To create a new set of encryption keys, users must terminate and relaunch their CVMs.

- IV exhaustion will crash the application.
  The H100 CC modes use a 96-bit deterministic IV for each virtual copy engine used to transfer data between the GPU and CPU. When this IV space is exhausted, transfers will fail to complete.

  Workaround
  CVM must be restarted.

- GPU-Ready bit is set when devtools mode is enabled.

  Workaround
  When in full CC-on modes, the driver will not accept any workloads until the Attestation SDK, or users, manually enable a GPU-Ready bit. Devtools mode will automatically have this bit enabled.

  Users should use best practices by attesting the GPU before performing any work. The GPUs booted in devtools mode will be clearly identified, and the attestation will fail.

- NVIDIA Performance Primitives (NPP) might not work.

  Workaround
  NPP uses optimized coding to extract maximum performance from commonly used transforms/calculations. Part of these leverage pinned host memory, which is unsupported in CC.
- In Protected PCIe mode, when the source or destination operand are imported GPU memory allocations on a device that is not visible to the process, the host-to-device or device-to-host copies might fail asynchronously with `cudaErrorLaunchFailure`.

- In Protected PCIe mode, using `cooperative_groups::multi_grid_group::sync` in kernels launched with `cudaLaunchCooperativeKernelMultiDevice` results in the kernel failing with `cudaErrorIllegalAddress`.

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. 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.

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 are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com