



# NVIDIA HPC SDK RELEASE NOTES

RN-09976-001-V21.11 | December 2021



# TABLE OF CONTENTS

Chapter 1. What's New.....	1
Chapter 2. Release Component Versions.....	3
Chapter 3. Supported Platforms.....	5
3.1. Platform Requirements for the HPC SDK.....	5
3.2. Supported CUDA Toolchain Versions.....	6
Chapter 4. Known Limitations.....	7
Chapter 5. Deprecations and Changes.....	8

## LIST OF TABLES

Table 1 HPC SDK Release Components .....	3
Table 2 HPC SDK Platform Requirements .....	5



# Chapter 1.

## WHAT'S NEW

Welcome to the 21.11 version of the NVIDIA HPC SDK, a comprehensive suite of compilers and libraries enabling developers to program the entire HPC platform, from the GPU foundation to the CPU and out through the interconnect.

Key features that are new in this release of the NVIDIA HPC SDK for Linux include:

- ▶ HPC SDK version 21.11 includes multi-node, multiGPU Math Library functionality, in cuSOLVERMp. Initial functionality will include Cholesky and LU Decomposition, with and without pivoting. Please refer to the [cuSOLVERMp documentation](#) for further details.
- ▶ The HPC SDK now supports Amazon AWS instances that utilize the Graviton 2 processor.
- ▶ The NVFORTRAN compiler now supports the REDUCE clause for use with DO CONCURRENT, as described in the current [working draft of the ISO Fortran Standard](#).
- ▶ The HPC Compilers now include initial support for array reductions for OpenACC and OpenMP loop. More details can be found in the user guide.
- ▶ The HPC Compilers now support the `--gcc-toolchain` option, similarly to the clang-based compilers. This is provided in addition to the existing `rcfile` method of specifying non-default GNU Compiler Collection (GCC) versions for use with the HPC Compilers.
- ▶ The HPC Compilers now include several GCC-compatible command line flags for specifying x86-64 target architecture details. Please refer to the `-tp` option for details.
- ▶ The HPC SDK includes the NVTX Fortran module, which enables easier use with the NVIDIA Tools Extension Library (NVTX) for performance and profiling studies with NVIDIA Nsight.
- ▶ The HPC SDK now includes the cufftXt Fortran module, which allows HPC applications written in Fortran to directly use cufftXt, NVIDIA's highly-optimized multi-GPU FFT library that can be called directly from host code.
- ▶ To allow application packagers and developers to more seamlessly integrate their code with the NVIDIA HPC SDK, CMake config files are now included that define CMake targets for the various components of the HPC SDK.
- ▶ The NVLAMATH linear algebra wrappers now include support for 64-bit integers.

- ▶ CUDA Fortran users can now specify that `cudaMallocManaged` should be used for allocating device data.
- ▶ Important performance enhancements and bug fixes are included in all components updated for the 21.11 release of the HPC SDK.
- ▶ The NVIDIA HPC SDK now includes versions 11.5 Update 1, 11.0, and 10.2 of the CUDA toolchain.

# Chapter 2.

## RELEASE COMPONENT VERSIONS

The NVIDIA HPC SDK 21.11 release contains the following versions of each component:

Table 1 HPC SDK Release Components

	Linux_x86_64			Linux_ppc64le			Linux_aarch64		
	CUDA 10.2	CUDA 11.0	CUDA 11.5	CUDA 10.2	CUDA 11.0	CUDA 11.5	CUDA 10.2	CUDA 11.0	CUDA 11.5
nvc++	21.11			21.11			21.11		
nvc	21.11			21.11			21.11		
nvfortran	21.11			21.11			21.11		
nvcc	10.2.89	11.0.221	11.5.117	10.2.89	11.0.228	11.5.117	N/A	11.0.228	11.5.117
NCCL	2.11.4	2.11.4	2.11.4	2.11.4	2.11.4	2.11.4	N/A	2.11.4	2.11.4
NVSHMEM	2.2.1	2.2.1	2.2.1	2.2.1	2.2.1	2.2.1	N/A	N/A	N/A
cuBLAS	10.2.2.89	11.2.0.252	11.7.4.6	10.2.2.89	11.2.0.252	11.7.4.6	N/A	11.2.0.252	11.7.4.6
cuFFT	10.1.2.89	10.2.1.245	10.6.0.107	10.1.2.89	10.2.1.245	10.6.0.107	N/A	10.2.1.245	10.6.0.107
cuRAND	10.1.2.89	10.2.1.245	10.2.7.107	10.1.2.89	10.2.1.245	10.2.7.107	N/A	10.2.1.245	10.2.7.107
cuSOLVER	10.3.0.89	10.6.0.245	11.2.1.48	10.3.0.89	10.6.0.245	11.2.1.48	N/A	10.6.0.245	11.2.1.48
cuSOLVER	N/A	N/A	0.2.0	N/A	N/A	N/A	N/A	N/A	N/A
cuSPARSE	10.3.1.89	11.1.1.245	11.7.0.107	10.3.1.89	11.1.1.245	11.7.0.107	N/A	11.1.1.245	11.7.0.107
cuTENSORFLOW	1.4.0	1.4.0	1.4.0	1.4.0	1.4.0	1.4.0	N/A	1.4.0	1.4.0
Nsight Compute	2021.3.0			2021.3.0			2021.3.0		
Nsight Systems	2021.4.1.73			2021.4.1.73			2021.4.1.73		
OpenMPI	3.1.5			3.1.5			3.1.5		
HPC-X	N/A	2.10b	2.10b	N/A	N/A	N/A	N/A	2.10b	2.10b
UCX	N/A	1.12.0	1.12.0	N/A	1.12.0	1.12.0	N/A	1.12.0	1.12.0

	Linux_x86_64			Linux_ppc64le			Linux_aarch64		
	CUDA 10.2	CUDA 11.0	CUDA 11.5	CUDA 10.2	CUDA 11.0	CUDA 11.5	CUDA 10.2	CUDA 11.0	CUDA 11.5
OpenBLAS	0.3.13			0.3.13			0.3.13		
Scalapack	2.1.0			2.1.0			2.1.0		
Thrust	1.9.7	1.9.9	1.13.1	1.9.7	1.9.9	1.13.1	1.9.7	1.9.10	1.13.1
CUB	N/A	1.9.9	1.13.1	N/A	1.9.9	1.13.1	N/A	1.9.9	1.13.1
libc++	1.0.0	2.0.0	2.0.0	1.0.0	2.0.0	2.0.0	1.0.0	2.0.0	2.0.0



# Chapter 3. SUPPORTED PLATFORMS

## 3.1. Platform Requirements for the HPC SDK

Table 2 HPC SDK Platform Requirements

Architecture	Linux Distributions	Minimum gcc/ glibc Toolchain	Minimum CUDA Driver
x86_64	CentOS 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8 CentOS 7.9, 8.0, 8.1, 8.2 Fedora 29, 30, 31, 32 OpenSUSE Leap 15.0, 15.1 RHEL 7.0, 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9 RHEL 8.0, 8.1, 8.2 SLES 12SP4, 12SP5, 15SP1 Ubuntu 18.04, 20.04	C99: 4.8 C11: 4.9 C++03: 4.8 C++11: 4.9 C++14: 5.1 C++17: 7.1	440.33
ppc64le	RHEL 7.3, 7.4, 7.5, 7.6, 7.7, 8.0, 8.1 RHEL Pegas 7.5, 7.6 Ubuntu 18.04	C99: 4.8 C11: 4.9 C++03: 4.8 C++11: 4.9 C++14: 5.1 C++17: 7.1	440.33
aarch64	CentOS 8.1, 8.2, 8.3, 8.4	C99: 4.8 C11: 4.9	450.36

Architecture	Linux Distributions	Minimum gcc/ glibc Toolchain	Minimum CUDA Driver
	RHEL 8.1, 8.2 Ubuntu 18.04, 20.04	C++03: 4.8 C++11: 4.9 C++14: 5.1 C++17: 7.1	

Programs generated by the HPC Compilers for x86\_64 processors require a minimum of AVX instructions, which includes Sandy Bridge and newer CPUs from Intel, as well as Bulldozer and newer CPUs from AMD. POWER 8 and POWER 9 CPUs from the POWER architecture are supported. For the Arm architecture, the minimum required version is Arm v8.1.

## 3.2. Supported CUDA Toolchain Versions

The NVIDIA HPC SDK uses elements of the CUDA toolchain when building programs for execution with NVIDIA GPUs. Every HPC SDK installation package puts the required CUDA components into an installation directory called `[install-prefix]/[arch]/[nvhpc-version]/cuda`.

An NVIDIA CUDA GPU device driver must be installed on a system with a GPU before you can run a program compiled for the GPU on that system. The NVIDIA HPC SDK does not contain CUDA Drivers. You must download and install the appropriate [CUDA Driver from NVIDIA](#), including the [CUDA Compatibility Platform](#) if that is required.

The `nvaccelinfo` tool prints the CUDA Driver version in its output. You can use it to find out which version of the CUDA Driver is installed on your system.

The NVIDIA HPC SDK 21.11 includes the following CUDA toolchain versions:

- ▶ CUDA 10.2
- ▶ CUDA 11.0
- ▶ CUDA 11.5 update 1

The minimum required CUDA driver versions are listed in the table in Section 3.1.

## Chapter 4.

# KNOWN LIMITATIONS

- ▶ Some users may experience a bug when using OpenBLAS that causes segmentation faults at job startup; increasing the user's data segment limit (`ulimit -d`) will work around this issue. This issue has recently been addressed upstream, and the fix will be included when OpenBLAS is updated in a future release of the HPC SDK.
- ▶ Debug information for Fortran arrays with non-constant bounds is not handled correctly, and querying values will yield incorrect results. Stepping through CUDA Fortran and OpenACC kernels is partially supported, but incorrect line numbers are displayed. For additional general limitations with `cuda-gdb`, please refer to its documentation.
- ▶ When using `-stdpar` to accelerate C++ parallel algorithms, the algorithm calls cannot include virtual function calls or function calls through a function pointer, cannot use C++ exceptions, can only dereference pointers that point to the heap, and must use random access iterators (raw pointers as iterators work best).
- ▶ When `nvc++ -stdpar=multicore` is used to generate parallel code, OpenMP pragmas in the same translation unit will also be enabled.

## Chapter 5. DEPRECATIONS AND CHANGES

- ▶ Starting with the 21.11 version of the NVIDIA HPC SDK, the HPC-X package is no longer shipped as part of the packages made available for the POWER architecture.
- ▶ The current default of `-gpu=implicitsections` will change in a future release to `-gpu=noimplicitsections` to adhere to the OpenACC specification.
- ▶ Starting with the 21.5 version of the NVIDIA HPC SDK, the `-cuda` option for NVC++ and NVFORTRAN no longer automatically links the NVIDIA GPU math libraries. Please refer to the `-cudalib` option.
- ▶ HPC Compiler support for the Kepler architecture of NVIDIA GPUs was deprecated starting with the 21.3 version of the NVIDIA HPC SDK.
- ▶ Support for the KNL architecture of multicore CPUs in the NVIDIA HPC SDK was deprecated in the HPC SDK version 21.3.

## Notice

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.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

## Trademarks

NVIDIA, the NVIDIA logo, CUDA, CUDA-X, GPUDirect, HPC SDK, NGC, NVIDIA Volta, NVIDIA DGX, NVIDIA Nsight, NVLink, NVSwitch, and Tesla 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.

## Copyright

© 2013-2021 NVIDIA Corporation. All rights reserved.