NVIDIA HPC SDK RELEASE NOTES

RN-09976-001-V21.1 | February 2021

TABLE OF CONTENTS

LIST OF TABLES

Chapter 1.
WHAT'S NEW

Welcome to the 21.1 release 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.

The 21.1 release of the NVIDIA HPC SDK is primarily a bugfix release. Key features that are new in this release for Linux include:

- Diagnostics output from NVC++ is not line-wrapped by default, but can be enabled via the --wrap-diagnostics command line option.
- Support for IEEE 128-bit floating-point operations via software emulation on x86-64 CPUs is enabled via the compiler built-in type __float128.
- The NVIDIA HPC SDK now includes versions 11.2, 11.0, and 10.2 of the CUDA toolchain.
- HPC SDK library and developer tool components have been updated to their newest versions.
- The new command-line option -Msyntax-only has been added to perform only syntax checking, semantic checking, and Fortran module (.mod) file generation.
- The NVFORTRAN option -Minline now allows routines with pass-by-value arguments with the "value" attribute to get inlined when the corresponding argument is a scalar variable.
- New command-line options have been implemented for HPC SDK compilers:

  - The -Minstrument-exclude-file-list option specifies a list of files to exclude from instrumentation.
  - The -Minstrument-exclude-func-list option specifies a list of function names to exclude from instrumentation.

Chapter 2.
RELEASE COMPONENT VERSIONS

The NVIDIA HPC SDK 21.1 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 11.2	CUDA 10.2	CUDA 11.0	CUDA 11.2	CUDA 10.2	CUDA 11.0	CUDA 11.2
nvc++	21.1			21.1			21.1		
nvc	21.1			21.1			21.1		
nvfortran	21.1			21.1			21.1		
nvcc	10.2.89	11.0.228	11.2.0.032	10.2.89	11.0.228	11.2.0.032	N/A	11.0.228	11.2.0.032
NCCL	2.8.3-1	2.8.3-1	2.8.3-1	2.8.3-1	2.8.3-1	2.8.3-1	N/A	N/A	2.8.3-1
NVSHMEM	1.1.3	1.1.3	1.1.3	1.1.3	1.1.3	1.1.3	N/A	N/A	N/A
cuBLAS	10.2.2.89	11.2.0.252	11.2.1.74	10.2.2.89	11.2.0.252	11.2.1.74	N/A	11.2.0.252	11.2.1.74
cuFFT	10.1.2.89	10.2.1.245	10.3.0.74	10.1.2.89	10.2.1.245	10.3.0.74	N/A	10.2.1.245	10.3.0.74
cuRAND	10.1.2.89	10.2.1.245	10.2.2.74	10.1.2.89	10.2.1.245	10.2.2.74	N/A	10.2.1.245	10.2.2.74
cuSOLVER	10.3.0.89	10.6.0.245	11.0.0.74	10.3.0.89	10.6.0.245	11.0.0.74	N/A	10.6.0.245	11.0.0.74
cuSPARSE	10.3.1.89	11.1.1.245	11.2.0.275	10.3.1.89	11.1.1.245	11.2.0.275	N/A	11.1.1.245	11.2.0.275
cuTENSOR	1.2.1	1.2.1	1.2.2	1.2.1	1.2.1	1.2.2	N/A	1.2.1	1.2.2
Nsight Compute	2020.3.0.0-29307467			2020.3.0.0-29307467			2020.3.0.0-29307467		
Nsight Systems	2020.5.1.85 CLI and GUI			2020.5.1.85 CLI			2020.5.1.85 CLI		
OpenMPI	3.1.5			3.1.5			3.1.5		
OpenMPI 4	N/A	N/A	4.0.5	N/A	N/A	4.0.5	N/A	N/A	4.0.5
UCX	N/A	1.10.0	1.10.0	N/A	1.10.0	1.10.0	N/A	1.10.0	1.10.0
OpenBLAS	0.3.10			0.3.10			0.3.10		

	Linux_x86_64			Linux_ppc64le			Linux_aarch64		
	CUDA 10.2	CUDA 11.0	CUDA 11.2	CUDA 10.2	CUDA 11.0	CUDA 11.2	CUDA 10.2	CUDA 11.0	CUDA 11.2
Scalapack		2.1.0			2.1.0			2.1.0	
Thrust	1.9.7	1.9.9	1.9.10	1.9.7	1.9.9	1.9.10	1.9.7	1.9.10	1.9.10
CUB	N/A	1.9.9	1.9.10	N/A	1.9.9	1.9.10	N/A	1.9.9	1.9.10
libcu++	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 16.04, 18.04, 19.10, 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 16.04, 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	RHEL 8.1	C99: 4.8	450.36

Architecture	Linux Distributions	Minimum gcc/ glibc Toolchain	Minimum CUDA Driver
	Ubuntu 18.04	C11: 4.9 C++03: 4.8 C++11: 4.9 C++14: 5.1 C++17: 7.1	

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 21.1 includes the following CUDA toolchain versions:

▸ CUDA 10.2
▸ CUDA 11.0
▸ CUDA 11.2

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

Chapter 4.
KNOWN LIMITATIONS

▶ When invoking MPI executables directly from comm_libs/mpi/bin, the user must explicitly set the LD_LIBRARY_PATH environment variable to include the compilers/lib directory, or use the included modulefiles with the NVIDIA HPC SDK to access the MPI executables.

▶ When compiling for Skylake Core or Skylake Xeon processors, it is recommended that the target processor compiler option -tp skylake be specified. Specifying -tp skylake at link time is optional.

▶ The cuda-gdb debugger is included in this release. Currently, Fortran arrays with non-constant bounds are 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).

Chapter 5.
DEPRECATIONS

- The 21.1 release of the NVIDIA HPC SDK supports NVIDIA GPUs with compute capability 3.5 (cc35) and newer.
- Support for the KNL architecture of multicore CPUs in the NVIDIA HPC SDK is deprecated in the 21.1 release.
- The NVIDIA HPC SDK discontinues support for the Ubuntu operating system on ppc64le architecture systems in the 21.1 release.

NVIDIA HPC SDK