NVIDIA HPC SDK RELEASE NOTES

RN-09976-001-V20.7 | August 2020

TABLE OF CONTENTS

LIST OF TABLES

Chapter 1. WHAT'S NEW

This is the first 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 through the interconnect.

Key features of the 20.7 GA release of the NVIDIA HPC SDK for Linux include:

- Support for NVIDIA Ampere Architecture GPUs with FP16, TF32 and FP64 tensor cores, and NVIDIA Volta tensor cores
- Supported on CUDA 11.0, 10.2 and 10.1 for compute capability 3.5 and up
- Cross-platform support for x86-64, OpenPOWER and Arm Server multicore CPUs
- nvc++ ISO C++17 compiler with Parallel Algorithms acceleration on GPUs, OpenACC and OpenMP
- nvfortran ISO Fortran 2003 compiler with array intrinsics acceleration on GPUs, CUDA Fortran, OpenACC and OpenMP
- nvc ISO C11 compiler with OpenACC and OpenMP
- nvcc NVIDIA CUDA C++ compiler
- cuBLAS GPU-accelerated basic linear algebra subroutine (BLAS) library
- cuSOLVER GPU-accelerated dense and sparse direct solvers
- cuSPARSE GPU-accelerated BLAS for sparse matrices
- cuFFT GPU-accelerated library for Fast Fourier Transforms
- cuTENSOR GPU-accelerated tensor linear algebra library
- cuRAND GPU-accelerated random number generation (RNG)
- Thrust GPU-accelerated library of C++ parallel algorithms and data structures
- CUB cooperative threadblock primitives and utilities for CUDA kernel programming
- libcu++ opt-in heterogeneous CUDA C++ Standard Libarary
- NCCL library for fast multi-GPU/multi-node collective communications
- NVSHMEM library for fast GPU memory-to-memory transfers (OpenSHMEM compatible)
- OpenMPI GPU-aware message passing interface library
- NVIDIA Nsight Systems interactive HPC applications performance profiler
- NVIDIA Nsight Compute interactive GPU compute kernel performance profiler
- NVIDIA cuda-gdb for interactive GPU compute kernel performance debugging

- NVIDIA compute-sanitizer for memory safety checks

Chapter 2. RELEASE COMPONENT VERSIONS

The NVIDIA HPC SDK 20.7 release contains the following versions of each component:

Table 1 HPC SDK Release Components

	Linux_x86_64			Linux_ppc64le			Linux_aarch64		
	CUDA 10.1	CUDA 10.2	CUDA 11.0	CUDA 10.1	CUDA 10.2	CUDA 11.0	CUDA 10.1	CUDA 10.2	CUDA 11.0
nvc++	20.7			20.7			20.7		
nvc	20.7			20.7			20.7		
nvfortran	20.7			20.7			20.7		
nvcc	10.1.243	10.2.89	11.0.109	10.1.243	10.2.89	11.0.109	N/A	N/A	11.0.109
NCCL	2.7.3-1	2.7.3-1	2.7.3-1	2.7.3-1	2.7.3-1	2.7.3-1	N/A	N/A	2.7.3-1
NVSHMEM	1.0.0-0	1.0.0-0	1.0.0-0	1.0.0-0	1.0.0-0	1.0.0-0	N/A	N/A	N/A
cuBLAS	10.2.1.243	10.2.2.89	11.1.0.229	10.2.1.243	10.2.2.89	11.1.0.229	N/A	N/A	11.1.0.229
cuFFT	10.1.1.243	10.1.2.89	10.2.0.218	10.1.1.243	10.1.2.89	10.2.0.218	N/A	N/A	10.2.0.218
cuRAND	10.1.1.243	10.1.2.89	10.2.1.218	10.1.1.243	10.1.2.89	10.2.1.218	N/A	N/A	10.2.1.218
cuSOLVER	10.2.0.243	10.3.0.89	10.5.0.218	10.2.0.243	10.3.0.89	10.5.0.218	N/A	N/A	10.5.0.218
cuSPARSE	10.3.0.243	10.3.1.89	11.1.0.218	10.3.0.243	10.3.1.89	11.1.0.218	N/A	N/A	11.1.0.218
cuTENSOR	1.0.2	1.0.2	1.1.0	1.0.2	1.0.2	1.1.0	N/A	N/A	1.1.0
Nsight Compute	2020.1.0.33-28294165			2020.1.0.33-28294165			2020.1.0.33-28294165		
Nsight Systems	2020.3.1.54 CLI and GUI			2020.3.1.54 CLI			2020.3.1.54 CLI		
OpenMPI	3.1.5			3.1.5			3.1.5		
OpenBLAS	0.3.7			0.3.7			0.3.7		
Scalapack	2.1.0			2.1.0			2.1.0		
Thrust	1.9.6-1	1.9.7	1.9.9	1.9.6-1	1.9.7	1.9.9	1.9.6-1	1.9.7	1.9.9

	Linux_x86_64			Linux_ppc64le			Linux_aarch64		
	CUDA 10.1	CUDA 10.2	CUDA 11.0	CUDA 10.1	CUDA 10.2	CUDA 11.0	CUDA 10.1	CUDA 10.2	CUDA 11.0
CUB	N/A	N/A	1.9.9	N/A	N/A	1.9.9	N/A	N/A	1.9.9
libcu++	N/A	1.0.0	2.0.0	N/A	1.0.0	2.0.0	N/A	1.0.0	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 6.6, 6.7, 6.8, 6.9, 6.10 CentOS 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8 CentOS 8.0, 8.1 Fedora 29, 30, 31, 32 OpenSUSE Leap 15.0, 15.1 RHEL 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 6.10 RHEL 7.0, 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8 RHEL 8.0, 8.1 SLES 12SP4, 12SP5, 15SP1 Ubuntu 14.04, 16.04, 18.04, 19.04, 19.10, 20.04	C99: 4.4 C11: 4.9 C++03: 4.4 C++11: 4.9 C++14: 5.1 C++17: 7.1	418.39
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: 4.4 C11: 4.9 C++03: 4.4 C++11: 4.9 C++14: 5.1	410.45

Architecture	Linux Distributions	Minimum gcc/ glibc Toolchain	Minimum CUDA Driver
		C++17: 7.1	
aarch64	RHEL 8.1 Ubuntu 18.04	C99: 4.4 C11: 4.9 C++03: 4.4 C++11: 4.9 C++14: 5.1 C++17: 7.1	450.36

3.2. Supported CUDA Toolkit 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 .

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

The NVIDIA HPC SDK 20.7 GA includes stand-alone support for the following CUDA toolchain versions:

- CUDA 10.1
- CUDA 10.2
- CUDA 11.0

See the NVIDIA HPC Compilers User Guide for information about using the HPC SDK Fortran, C++ and C compilers with alternative versions of the CUDA toolchain.

Chapter 4. KNOWN LIMITATIONS

- When the -stdpar option is used on a machine with multiple GPUs installed with differing compute capabilities, a target compute capability must be selected by the user at compile time by using the flag '-gpu=ccXY' where 'XY' specifies the two-digit compute capability version. This requirement will be lifted in the upcoming 20.9 release.
- 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)

Notice

Trademarks

Copyright