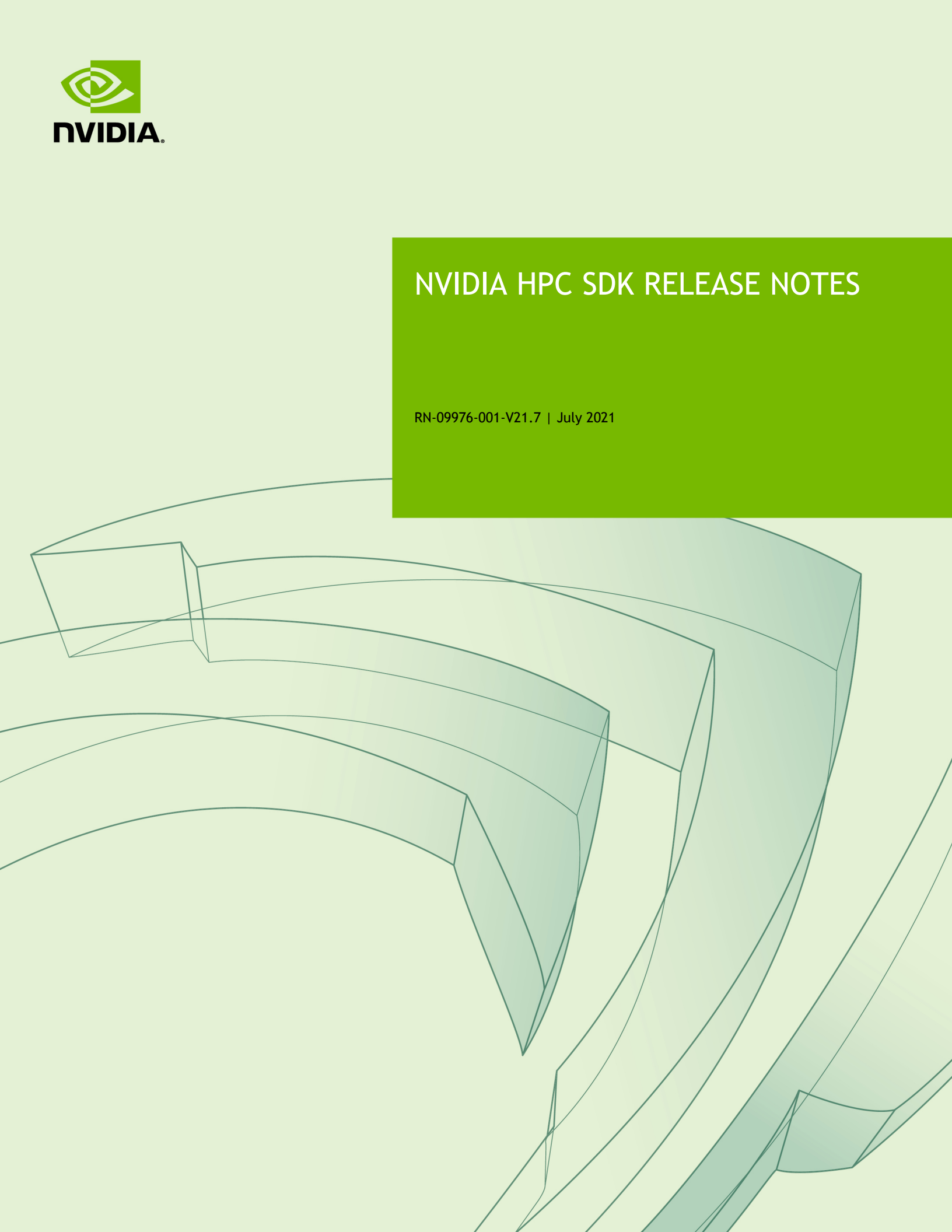

NVIDIA HPC SDK RELEASE NOTES

RN-09976-001-V21.7 | July 2021

TABLE OF CONTENTS

LIST OF TABLES

Chapter 1. WHAT'S NEW

Welcome to the 21.7 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:

- The 21.7 version of the HPC SDK includes full support for the NVIDIA Arm HPC Developer Kit.
- Performance for Arm CPUs is improved by various enhancements to the HPC Compilers that provide additional intrinsics, and optimized math functions and vectorization.
- Reductions for CUDA Fortran "cuf kernels" can be explicitly specified by the user. Refer to the CUDA Fortran Programming Guide for more details.
- The command-line option -Mint128 turns on support for extended integer types __int128 and unsigned __int128 in NVC and NVC++. Note that 128-bit integer types are not supported in GPU code or in OpenMP or OpenACC. In a future release, 128-bit integer types will be enabled by default.
- The -Minfo option for NVC++ has a new suboption, stdpar. When compiling with -stdpar -Minfo=stdpar the compiler will report whether or not calls to C++ standard algorithms with an execution policy were parallelized. Additionally, when compiling with -stdpar, the compiler will now warn about calls to standard algorithms with a parallel execution policy that are not parallelized.
- The NVIDIA HPC SDK now includes versions 11.4, 11.0, and 10.2 of the CUDA toolchain.
- The behavior of the ieee_arithmetic module's ieee_next_after(x,y) runtime library routine has been updated to be consistent with the Fortran specification and other implementations.
- A new option -gpu=[no]implicitsections has been added to direct the HPC Compilers to [not] implicitly treat array element references in a data clause as an array section. The default behavior for this release matches the behavior of previous releases; the default behavior will change in a future release. Please see the HPC Compilers User Guide or the manpages for more information.
- For all targets, -O3 now includes two floating point optimizations that can result in some loss of precision. These include:

  - potentially rewriting floating point division as a multiply by reciprocal (x/y => x*1/y). This behavior can be enabled or disabled by -M[no-]recip-div.
  - factorization of floating point types for increased symbolic cancelation. This behavior can be enabled or disabled by -M[no-]factorize.
- On Arm, -O3 and -Mfprelaxed are more aggressive at finding FMA opportunities.
- On Arm and x86, -Mfprelaxed allows more floating point optimization.
- For Fortran codes, -Minline has been improved to increase inlining opportunities.

Chapter 2. RELEASE COMPONENT VERSIONS

The NVIDIA HPC SDK 21.7 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 11.4**	**CUDA 10.2**	**CUDA 11.0**	**CUDA 11.4**	**CUDA 10.2**	**CUDA 11.0**	**CUDA 11.4**
nvc++	21.7			21.7			21.7		
nvc	21.7			21.7			21.7		
nvfortran	21.7			21.7			21.7		
nvcc	10.2.89	11.0.228	11.4.43	10.2.89	11.0.228	11.4.43	N/A	11.0.228	11.4.43
NCCL	2.10.3	2.10.3	2.10.3	2.10.3	2.10.3	2.10.3	N/A	2.10.3	2.10.3
NVSHMEM	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2	N/A	N/A	N/A
cuBLAS	10.2.2.89	11.2.0.252	11.5.4.3	10.2.2.89	11.2.0.252	11.5.4.3	N/A	11.2.0.252	11.5.4.3
cuFFT	10.1.2.89	10.2.1.245	10.5.0.43	10.1.2.89	10.2.1.245	10.5.0.43	N/A	10.2.1.245	10.5.0.43
cuRAND	10.1.2.89	10.2.1.245	10.2.5.43	10.1.2.89	10.2.1.245	10.2.5.43	N/A	10.2.1.245	10.2.5.43
cuSOLVER	10.3.0.89	10.6.0.245	11.2.0.43	10.3.0.89	10.6.0.245	11.2.0.43	N/A	10.6.0.245	11.2.0.43
cuSPARSE	10.3.1.89	11.1.1.245	11.6.0.43	10.3.1.89	11.1.1.245	11.6.0.43	N/A	11.1.1.245	11.6.0.43
cuTENSOR	1.3.1	1.3.1	1.3.1	1.3.1	1.3.1	1.3.1	N/A	1.3.1	1.3.1
Nsight Compute	2021.1.1			2021.1.1			2021.1.1		
Nsight Systems	2021.2.1.58			2021.2.1.58			2021.2.1.58		
OpenMPI	3.1.5			3.1.5			3.1.5		
HPC-X	N/A	2.8.1	2.8.1	N/A	2.8.1	2.8.1	N/A	2.8.1	2.8.1
UCX	N/A	1.10.0rc1	1.10.0rc1	N/A	1.10.0rc1	1.10.0rc1	N/A	1.10.0rc1	1.10.0rc1
OpenBLAS	0.3.13			0.3.13			0.3.13		

	Linux_x86_64			Linux_ppc64le			Linux_aarch64		
	CUDA 10.2	CUDA 11.0	CUDA 11.4	CUDA 10.2	CUDA 11.0	CUDA 11.4	CUDA 10.2	CUDA 11.0	CUDA 11.4
Scalapack	2.1.0			2.1.0			2.1.0		
Thrust	1.9.7	1.9.9	1.12.0	1.9.7	1.9.9	1.12.0	1.9.7	1.9.10	1.12.0
CUB	N/A	1.9.9	1.12.0	N/A	1.9.9	1.12.0	N/A	1.9.9	1.12.0
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 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	RHEL 8.1	C99: 4.8 C11: 4.9	450.36

Architecture	Linux Distributions	Minimum gcc/ glibc Toolchain	Minimum CUDA Driver
	Ubuntu 18.04	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 21.7 includes the following CUDA toolchain versions:

- CUDA 10.2
- CUDA 11.0
- CUDA 11.4

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

Chapter 4. KNOWN LIMITATIONS

- The cuda-gdb debugger is included in this version. 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).
- 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

- 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.
- 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

Trademarks

Copyright