



NSIGHT COMPUTE

v2019.5.0 | December 2019

Release Notes



TABLE OF CONTENTS

Chapter 1. Release Notes	1
1.1. Updates in 2019.5.....	1
1.2. Updates in 2019.4.....	2
1.3. Updates in 2019.3.1.....	3
1.4. Updates in 2019.3.....	4
1.5. Updates in 2019.2.....	5
1.6. Updates in 2019.1.....	5
Chapter 2. Known Issues	7
Chapter 3. Support	9
3.1. Platform Support.....	9
3.2. GPU Support.....	10

LIST OF TABLES

Table 1	Platforms supported by NVIDIA Nsight Compute	9
Table 2	GPU architectures supported by NVIDIA Nsight Compute	10

Chapter 1.

RELEASE NOTES

1.1. Updates in 2019.5

General

- ▶ Added *section sets* to reduce the default overhead and make it easier to configure metric sets for profiling
- ▶ Reduced the size of the installation
- ▶ Added support for CUDA Graphs Recapture API
- ▶ The NvRules API now supports accessing correlation IDs for instanced metrics
- ▶ Added breakdown tables for *SOL SM* and *SOL Memory* in the Speed Of Light section for Volta+ GPUs

NVIDIA Nsight Compute

- ▶ Added a snap-select feature to the Source page heatmap help navigate large files
- ▶ Added support for loading remote CUDA-C source files via SSH on demand for Linux x86_64 targets
- ▶ Charts on the Details page provide better help in tool tips when hovering metric names
- ▶ Improved the performance of the Source page when scrolling or collapsing
- ▶ The charts for Warp States and Compute pipelines are now sorted by value

NVIDIA Nsight Compute CLI

- ▶ Added support for GPU cache control, see `--cache-control`
- ▶ Added support for setting the kernel name base in command line output, see `--kernel-base`
- ▶ Added support for listing the available names for `--chips`, see `--list-chips`
- ▶ Improved the stability on Windows when using `--target-processes all`
- ▶ Reduced the profiling overhead for small metric sets in applications with many kernels

Resolved Issues

- ▶ Reduced the overhead caused by demangling kernel names multiple times
- ▶ Fixed an issue that kernel names were not demangled in CUDA Graph Nodes resources window
- ▶ The connection dialog better disables unsupported combinations or warns of invalid entries
- ▶ Fixed metric *thread_inst_executed_true* to derive from *smsp_not_predicated_off_thread_inst_executed* on Volta+ GPUs
- ▶ Fixed an issue with computing the theoretical occupancy on GV100
- ▶ Selecting an entry on the Source page heatmap no longer selects the respective source line, to avoid losing the current selection
- ▶ Fixed the current view indicator of the Source page heatmap to be line-accurate
- ▶ Fixed an issue when comparing metrics from Pascal and later architectures on the Summary page
- ▶ Fixed an issue that metrics representing constant values on Volta+ couldn't be collected without non-constant metrics

1.2. Updates in 2019.4

General

- ▶ Added support for the Linux PowerPC target platform
- ▶ Reduced the profiling overhead, especially if no source metrics are collected
- ▶ Reduced the overhead for non-profiled kernels
- ▶ Improved the deployment performance during remote launches
- ▶ Trying to profile on an unsupported GPU now shows an "Unsupported GPU" error message
- ▶ Added support for the `%i` sequential number placeholder to generate unique report file names
- ▶ Added support for *smsp__sass_** metrics on Volta and newer GPUs
- ▶ The *launch_occupancy_limit_shared_mem* now reports the device block limit if no shared memory is used by the kernel

NVIDIA Nsight Compute

- ▶ The *Profile* activity shows the command line used to launch `nv-nsight-cu-cli`
- ▶ The heatmap on the Source page now shows the represented metric in its tooltip
- ▶ The *Memory Workload Analysis Chart* on the Details page now supports baselines
- ▶ When applying rules, a message displaying the number of new rule results is shown in the status bar
- ▶ The Visual Profiler Transition Guide was added to the documentation
- ▶ Connection dialog activity options were added to the documentation
- ▶ A warning dialog is shown if the application is resumed without Auto-Profile enabled
- ▶ Pausing the application now has immediate feedback in the toolbar controls
- ▶ Added a *Close All* command to the *File* menu

NVIDIA Nsight Compute CLI

- ▶ The **--query-metrics** option now shows only metric base names for faster metric query. The new option **--query-metrics-mode** can be used to display the valid suffixes for each base metric.
- ▶ Added support for passing response files using the @ operator to specify command line options through a file

Resolved Issues

- ▶ Fixed an issue that reported the wrong executable name in the Session page when attaching
- ▶ Fixed issues that chart labels were shown elided on the Details page
- ▶ Fixed an issue that caused the cache hitrates to be shown incorrectly when baselines were added
- ▶ Fixed an illegal memory access when collecting *sass_*_histogram* metrics for applications using PyTorch on Pascal GPUs
- ▶ Fixed an issue when attempting to collect all *smsp_** metrics on Volta and newer GPUs
- ▶ Fixed an issue when profiling multi-context applications
- ▶ Fixed that profiling start/stop settings from the connection dialog weren't properly passed to the interactive profile activity
- ▶ Fixed that certain *smsp__warp_cycles_per_issue_stall** metrics returned negative values on Pascal GPUs
- ▶ Fixed that metric names were truncated in the **--page details** non-CSV command line output
- ▶ Fixed that the target application could crash if a connection port was used by another application with higher privileges

1.3. Updates in 2019.3.1

NVIDIA Nsight Compute

- ▶ Added ability to send bug reports and suggestions for features using *Send Feedback* in the *Help* menu

Resolved Issues

- ▶ Fixed calculation of theoretical occupancy for grids with blocks that are not a multiple of 32 threads
- ▶ Fixed intercepting child processes launched through Python's `subprocess.Popen` class
- ▶ Fixed issue of NVTX push/pop ranges not showing up for child threads in NVIDIA Nsight Compute CLI
- ▶ Fixed performance regression for metric lookups on the Source page
- ▶ Fixed description in rule covering the IMC stall reason
- ▶ Fixed cases where baseline values were not correctly calculated in the Memory tables when comparing reports of different architectures
- ▶ Fixed incorrect calculation of baseline values in the Executed Instruction Mix chart
- ▶ Fixed accessing instanced metrics in the NvRules API

- ▶ Fixed a bug that could cause the collection of unnecessary metrics in the Interactive Profile activity
- ▶ Fixed potential crash on exit of the profiled target application
- ▶ Switched underlying metric for **SOL_FB** in the GPU Speed Of Light section to be driven by **dram_throughput.avg.pct_of_peak_sustained_elapsed** instead of **fbpa_throughput.avg.pct_of_peak_sustained_elapsed**

1.4. Updates in 2019.3

General

- ▶ Improved performance
- ▶ Bug fixes
- ▶ Kernel launch context and stream are reported as metrics
- ▶ PC sampling configuration options are reported as metrics
- ▶ The default base port for connections to the target changed
- ▶ Section files support multiple, named Body fields
- ▶ NvRules allows users to query metrics using any convertible data type

NVIDIA Nsight Compute

- ▶ Support for filtering kernel launches using their NVTX context
- ▶ Support for new options to select the connection port range
- ▶ The Profile activity supports configuring PC sampling parameters
- ▶ Sections on the Details page support selecting individual bodies

NVIDIA Nsight Compute CLI

- ▶ Support for stepping to kernel launches from specific NVTX contexts
- ▶ Support for new **--port** and **--max-connections** options
- ▶ Support for new **--sampling-*** options to configure PC sampling parameters
- ▶ Section file errors are reported with **--list-sections**
- ▶ A warning is shown if some section files could not be loaded

Resolved Issues

- ▶ Using the **--summary** option works for reports that include invalid metrics
- ▶ The full process executable filename is reported for QNX targets
- ▶ The project system now properly stores the state of opened reports
- ▶ Fixed PTX syntax highlighting
- ▶ Fixed an issue when switching between manual and auto profiling in NVIDIA Nsight Compute
- ▶ The source page in NVIDIA Nsight Compute now works with results from multiple processes
- ▶ Charts on the NVIDIA Nsight Compute details page uses proper localization for numbers
- ▶ NVIDIA Nsight Compute no longer requires the system locale to be set to English

1.5. Updates in 2019.2

General

- ▶ Improved performance
- ▶ Bug fixes
- ▶ Kernel launch context and stream are reported as metrics
- ▶ PC sampling configuration options are reported as metrics
- ▶ The default base port for connections to the target changed
- ▶ Section files support multiple, named Body fields
- ▶ NvRules allows users to query metrics using any convertible data type

NVIDIA Nsight Compute

- ▶ Support for filtering kernel launches using their NVTX context
- ▶ Support for new options to select the connection port range
- ▶ The Profile activity supports configuring PC sampling parameters
- ▶ Sections on the Details page support selecting individual bodies

NVIDIA Nsight Compute CLI

- ▶ Support for stepping to kernel launches from specific NVTX contexts
- ▶ Support for new `--port` and `--max-connections` options
- ▶ Support for new `--sampling-*` options to configure PC sampling parameters
- ▶ Section file errors are reported with `--list-sections`
- ▶ A warning is shown if some section files could not be loaded

Resolved Issues

- ▶ Using the `--summary` option works for reports that include invalid metrics
- ▶ The full process executable filename is reported for QNX targets
- ▶ The project system now properly stores the state of opened reports
- ▶ Fixed PTX syntax highlighting
- ▶ Fixed an issue when switching between manual and auto profiling in NVIDIA Nsight Compute
- ▶ The source page in NVIDIA Nsight Compute now works with results from multiple processes
- ▶ Charts on the NVIDIA Nsight Compute details page uses proper localization for numbers
- ▶ NVIDIA Nsight Compute no longer requires the system locale to be set to English

1.6. Updates in 2019.1

General

- ▶ Support for CUDA 10.1
- ▶ Improved performance

- ▶ Bug fixes
- ▶ Profiling on Volta GPUs now uses the same metric names as on Turing GPUs
- ▶ Section files support descriptions
- ▶ The default sections and rules directory has been renamed to *sections*

NVIDIA Nsight Compute

- ▶ Added new profiling options to the options dialog
- ▶ Details page shows rule result icons in the section headers
- ▶ Section descriptions are shown in the details page and in the sections tool window
- ▶ Source page supports collapsing multiple source files or functions to show aggregated results
- ▶ Source page heatmap color scale has changed
- ▶ Invalid metric results are highlighted in the profiler report
- ▶ Loaded section and rule files can be opened from the sections tool window

NVIDIA Nsight Compute CLI

- ▶ Support for profiling child processes on Linux and Windows x86_64 targets
- ▶ NVIDIA Nsight Compute CLI uses a temporary file if no output file is specified
- ▶ Support for new **--quiet** option
- ▶ Support for setting the GPU clock control mode using new **--clock-control** option
- ▶ Details page output shows the NVTX context when **--nvtx** is enabled
- ▶ Support for filtering kernel launches for profiling based on their NVTX context using new **--nvtx-include** and **--nvtx-exclude** options
- ▶ Added new **--summary** options for aggregating profiling results
- ▶ Added option **--open-in-ui** to open reports collected with NVIDIA Nsight Compute CLI directly in NVIDIA Nsight Compute

Resolved Issues

- ▶ Installation directory scripts use absolute paths
- ▶ OpenACC kernel names are correctly demangled
- ▶ Profile activity report file supports a relative path
- ▶ Source view can resolve all applicable files at once
- ▶ UI font colors are improved
- ▶ Details page layout and label elision issues are resolved
- ▶ Turing metrics are properly reported on the Summary page
- ▶ All byte-based metrics use a factor of 1000 when scaling units to follow SI standards
- ▶ CSV exports properly align columns with empty entries
- ▶ Fixed the metric computation for `double_precision_fu_utilization` on GV11b
- ▶ Fixed incorrect 'selected' PC sampling counter values
- ▶ The SpeedOfLight section uses 'max' instead of 'avg' cycles metrics for Elapsed Cycles

Chapter 2.

KNOWN ISSUES

Installation

- ▶ The Visual Studio 2017 redistributable is not automatically installed by the NVIDIA Nsight Compute installer. The workaround is to install the x64 version of the 'Microsoft Visual C++ Redistributable for Visual Studio 2017' manually. The installer is linked on the main download page for Visual Studio at <https://www.visualstudio.com/downloads/> or download directly from <https://go.microsoft.com/fwlink/?LinkId=746572>.
- ▶ On platforms other than Windows, NVIDIA Nsight Compute must not be installed in a directory containing spaces or other whitespace characters.
- ▶ The installer might not show all patch-level version numbers during installation.

Launch and Connection

- ▶ Launching applications on remote targets/platforms is not supported for several combinations. See [Platform Support](#) for details. Manually launch the application using command line `nv-nsight-cu-cli --mode=launch` on the remote system and connect using the UI or CLI afterwards.
- ▶ In the NVIDIA Nsight Compute connection dialog, a remote system can only be specified for one target platform. Remove a connection from its current target platform in order to be able to add it to another.
- ▶ Terminating an application profiled via *Remote Launch* is not supported. NVIDIA Nsight Compute only disconnects from the remote process. This also applies when cancelling remote-launched *Profile* activities.
- ▶ Loading of CUDA-C sources via SSH requires the remote connection to be still configured, and for the hostname in the connection settings and in the report session details to match. For example, prefer `my-machine.my-domain.com`, instead of `my-machine`, even though the latter resolves to the same.

Profiling and Metrics

- ▶ Real texture traffic is not captured in *First-Level Cache* table for Pascal chips.
- ▶ The Block and Warp Durations histograms in the Launch Statistics section are unavailable for Volta and Turing architectures.
- ▶ Profiling kernels executed on a device that is part of an SLI group is not supported. An "Unsupported GPU" error is shown in this case.

- ▶ Profiling a kernel while other contexts are active on the same device (e.g. X server, or secondary CUDA or graphics application) can result in varying metric values for L2/FB (Device Memory) related metrics. Specifically, L2/FB traffic from non-profiled contexts cannot be excluded from the metric results. To completely avoid this issue, profile the application on a GPU without secondary contexts accessing the same device (e.g. no X server on Linux).

Compatibility

- ▶ Reports collected on Windows might show invalid characters for file and process names when opened in NVIDIA Nsight Compute on Linux.
- ▶ Applications calling blocking functions on std input/output streams can result in the profiler to stop, until the blocking function call is resolved.
- ▶ On QNX, when using the `--target-processes all` option, profiling shell scripts may hang after the script has completed. End the application using *Ctrl-C* on the command line or in the UI Terminate command in that case.
- ▶ NVIDIA Nsight Compute can hang on applications using RAPIDS in versions 0.6 and 0.7, due to an issue in cuDF.
- ▶ Profiling child processes launched from Python using `os.system()` cannot be profiled.
- ▶ Profiling of Cooperative Groups kernels launched with `cuLaunchCooperativeKernelMultiDevice` is not yet supported.
- ▶ On Linux systems, when profiling *bsd-csh* scripts, the original application output will not be printed. As a workaround, use a different C-shell, e.g. *tcsh*.

User Interface

- ▶ The API Statistics filter in NVIDIA Nsight Compute does not support units.
- ▶ Changing the application theme may result in incorrect rendering of some UI components. Restarting the application will resolve this issue.

Chapter 3.

SUPPORT

Information on supported platforms and GPUs.

3.1. Platform Support

Host denotes the UI can run on that platform. Target means that we can instrument applications on that platform for data collection. Applications launched with instrumentation on a target system can be connected to from most host platforms. The reports collected on one system can be opened on any other system.

Table 1 Platforms supported by NVIDIA Nsight Compute

	Host	Targets
Windows x86_64	Yes	Windows x86_64*, Linux x86_64, Linux PowerPC
Windows Subsystem for Linux	No	No
Linux x86_64	Yes	Windows x86_64*, Linux x86_64, Linux PowerPC
Linux PowerPC	No	Linux PowerPC
Linux x86_64 (Drive SDK)	Yes	Windows x86_64*, Linux x86_64, DRIVE OS Linux, DRIVE OS QNX
MacOSX 10.13+	Yes	Windows x86_64*, Linux x86_64*
DRIVE OS Linux	No	DRIVE OS Linux
DRIVE OS QNX	No	DRIVE OS QNX

Target platforms marked with * do not support remote launch from the respective host. Remote launch means that the application can be launched on the target system from the host UI. Instead, the application must be launched from the target system.

On all Linux platforms, NVIDIA Nsight Compute requires GLIBC version 2.15 or higher.

3.2. GPU Support

Table 2 GPU architectures supported by NVIDIA Nsight Compute

Architecture	Support	Metrics*
Kepler	No	
Maxwell	No	
Pascal GP100	No	
Pascal GP10x	Yes	Group A
Volta GV100	Yes	Group B
Volta GV11b	Yes	Group B
Turing TU10x	Yes	Group B

* NVIDIA Nsight Compute uses different sets of metric names for the different GPU architectures. This is due to the underlying measurement libraries that are used on those architectures. Within each metric name group (Group A, Group B), names are identical, with the exception of some metrics being only available on some specific architectures. The metrics of Group B are identical to those of the [PerfWorks Metrics API](#). A comparison between the metrics used in nvprof and their equivalent in NVIDIA Nsight Compute can be found in the [NVIDIA Nsight Compute CLI User Manual](#).

When using the default sections and rules installed with NVIDIA Nsight Compute, the difference in metric names is handled automatically. When manually selecting metric names for profiling or writing your own sections or rules, the correct metric group must be picked for the respective target architecture.

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 and the NVIDIA logo are trademarks 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

© 2018-2019 NVIDIA Corporation. All rights reserved.

This product includes software developed by the Syncro Soft SRL (<http://www.sync.ro/>).