NVIDIA

NSIGHT COMPUTE

v2020.1.1 | June 2020

Release Notes

TABLE OF CONTENTS

LIST OF TABLES

Chapter 1. RELEASE NOTES

1.1. Updates in 2020.1.1

General

- Metrics passed to **`--metrics`** on the NVIDIA Nsight Compute CLI or in the respective *Profile* activity option are automatically expanded to all first-level sub-metrics if required. See the documentation on **`--metrics`** for more details.
- Added new rules for detecting inefficiencies of using the sparse data compression on the NVIDIA Ampere architecture.
- The version of the NVIDIA Nsight Compute target collecting the results is shown in the *Session* page.
- Added new **`launch__grid_dim_[x,y,z]`** and **`launch__block_dim_[x,y,z]`** metrics.

NVIDIA Nsight Compute

- The *Break on API Error* functionality has been improved when auto profiling.

NVIDIA Nsight Compute CLI

- The full path to the report output file is printed after profiling.
- Added and corrected metrics in the nvprof *Metric Comparison* table.

Resolved Issues

- Documented the *breakdown:* metrics prefix.
- Fixed handling of escaped domain delimiters in NVTX filter expressions.
- Fixed issues with the occupancy charts for small block sizes.
- Fixed an issue when choosing a default report page in the options dialog.
- Fixed that the scroll bar could overlap the content when exporting the report page as an image.
- Fixed loading the CUDA stub library from the injection library of NVIDIA Nsight Compute on Linux.

1.2. Updates in 2020.1

General

- Added support for the NVIDIA GA100/SM 8.x GPU architecture
- Removed support for the Pascal SM 6.x GPU architecture
- Windows 7 is not a supported host or target platform anymore
- Added a rule for reporting uncoalesced memory accesses as part of the *Source Counters* section
- Added support for report name placeholders %p, %q, %i and %h
- The *Kernel Profiling Guide* was added to the documentation

NVIDIA Nsight Compute

- The UI command was renamed from **`nv-nsight-cu`** to **`ncu-ui`**. Old names remain for backwards compatibility.
- Added support for roofline analysis charts
- Added linked hot spot tables in section bodies to indicate performance problems in the source code
- Added section navigation links in rule results to quickly jump to the referenced section
- Added a new option to select how kernel names are shown in the UI
- Added new memory tables for the L1/TEX cache and the L2 cache. The old tables are still available for backwards compatibility and moved to a new section containing deprecated UI elements.
- Memory tables now show the metric name as a tooltip
- Source resolution now takes into account file properties when selecting a file from disk
- Results in the profile report can now be filtered by NVTX range
- The Source page now supports collapsing views even for single files
- The UI shows profiler error messages as dismissible banners for increased visibility
- Improved the baseline name control in the profiler report header

NVIDIA Nsight Compute CLI

- The CLI command was renamed from **`nv-nsight-cu-cli`** to **`ncu`**. Old names remain for backwards compatibility.
- Queried metrics on GV100 and newer chips are sorted alphabetically
- Multiple instances of NVIDIA Nsight Compute CLI can now run concurrently on the same system, e.g. for profiling individual MPI ranks. Profiled kernels are serialized across all processes using a system-wide file lock.

Resolved Issues

- More C++ kernel names can be properly demangled
- Fixed a **`free(): invalid pointer`** error when profiling applications using pytorch > 19.07

- Fixed profiling IBM Spectrum MPI applications that require PAMI GPU hooks (`--smpiargs="-gpu"`)
- Fixed that the first kernel instruction was missed when computing `sass__inst_executed_per_opcode`
- Reduced surplus DRAM write traffic created from flushing caches during kernel replay
- The *Compute Workload Analysis* section shows the IMMA pipeline on GV11b GPUs
- Profile reports now scroll properly on MacOS when using a trackpad
- Relative output filenames for the Profile activity now use the document directory, instead of the current working directory
- Fixed path expansion of ~ on Windows
- Memory access information is now shown properly for RED assembly instructions on the Source page
- Fixed that user `PYTHONHOME` and `PYTHONPATH` environment variables would be picked up by NVIDIA Nsight Compute, resulting in locale encoding issues.

1.3. Updates in 2019.5.1

General

- Added support for Nsight Compute Visual Studio Integration

1.4. 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

1.5. 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 ncu
- 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

1.6. 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 were 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`**

1.7. 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

1.8. 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

1.9. 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.
- On MacOS 10.14.5 and above, additional steps may be required to launch NVIDIA Nsight Compute since the application is not notarized. For additional information, please refer to https://support.apple.com/en-us/HT202491.
- 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 **`ncu --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

- The Block and Warp Durations histograms in the Launch Statistics section are unavailable for Volta and newer 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).
- In the current release, profiling a kernel while any other GPU work is executing on the same MIG compute instance can result in varying metric values for all units. NVIDIA Nsight Compute enforces serialization of the CUDA launches within the target application to ensure those kernels do not influence each other. See the *Serialization* topic in the *Kernel Profiling Guide* for more details. However, GPU work issued through other APIs in the target process or workloads created by non-target processes running simultaneously in the same MIG compute instance will influence the collected metrics. Note that it is acceptable to run CUDA processes in other MIG compute instances as they will not influence the profiled MIG compute instance.
- Profiling only supports up to 32 device instances, including instances of MIG partitions. Profiling the 33rd or higher device instance will result in indeterminate data.
- Enabling certain metrics can cause GPU kernels to run longer than the driver's watchdog time-out limit. In these cases the driver will terminate the GPU kernel resulting in an application error and profiling data will not be available. Please disable the driver watchdog time out before profiling such long running CUDA kernels.
  - On Linux, setting the X Config option Interactive to false is recommended.
  - For Windows, detailed information on disabling the Windows TDR is available at https://docs.microsoft.com/en-us/windows-hardware/drivers/display/timeout-detection-and-recovery

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*.

- Attempting to use the **`--clock-control`** option to set the GPU clocks will fail when profiling on a GPU partition. Please use **`nvidia-smi`** (installed with nvidia display driver) to control the clocks for the entire GPU. This will require administrative privileges when the GPU is partitioned.

User Interface

- The API Statistics filter in NVIDIA Nsight Compute does not support units.
- File size is the only property considered when resolving source files. Timestamps are currently ignored.
- Scrolling a Profile report on MacOS via the trackpad may be slow if the scrolling is initiated on a table header.
- Terminating or disconnecting an application in the *Interactive Profiling* activity while the API Stream View is updated can lead to a crash.

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	Yes	Windows*, Linux (x86_64)
Windows Subsystem for Linux	No	No
Linux (x86_64)	Yes	Windows*, Linux (x86_64), Linux (ppc64le), Linux (aarch64 sbsa)
Linux (ppc64le)	No	Linux (ppc64le)
Linux (aarch64 sbsa)	No	Linux (aarch64 sbsa)
Linux (x86_64) (Drive SDK)	Yes	Windows*, Linux (x86_64), Linux (aarch64), QNX
MacOSX 10.13+	Yes	Windows*, Linux (x86_64), Linux (ppc64le)
Linux (aarch64)	No	Linux (aarch64)
QNX	No	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.

Only Windows 10 is supported as host and target.

3.2. GPU Support

Table 2 GPU architectures supported by NVIDIA Nsight Compute

Architecture	Support
Kepler	No
Maxwell	No
Pascal	No
Volta GV100	Yes
Volta GV11b	Yes
Turing TU10x	Yes
NVIDIA GA100	Yes

Most metrics used in NVIDIA Nsight Compute 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.

Notice

Trademarks

Copyright

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